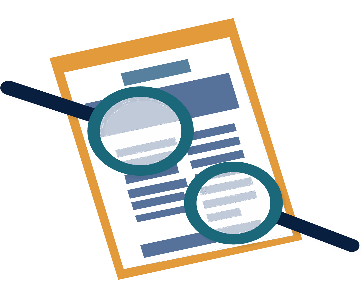
**המחלקה להנדסת תוכנה**

**פרויקט גמר - תש"פ**

אימות מחבר המבוסס על ניתוח כתב יד

Author verification based on handwritten text analysis

**מאת**

**דניאל גבאי - 311223598**

**שחר ישראלי - 203713094**

**מנחה אקדמי: דר' יהודה חסין אישור: תאריך:**

**רכז הפרויקטים: דר' אסף שפיינר אישור: תאריך:**

**מערכות ניהול הפרויקט:**

|  |  |  |
| --- | --- | --- |
| **#** | **מערכת** | **מיקום** |
| 1 | מאגר קוד | <https://github.com/DanielGabay/Author-verification-by-handwriting-samples> |
| 2 | יומן | <https://trello.com/b/7jKnxmLL/author-verification-by-handwriting-samples> |
| 3 | סרטון גרסת אלפא | <https://drive.google.com/file/d/1gAy8cAt62CRi6B0vDs2cZ9trAWreiXq9/view> |
| 4 | סרטון גרסה סופית |  |

*נאום המעלית*

בפרויקט זה ביצענו מחקר למציאת אלגוריתם שבעזרתו, פיתחנו מערכת ממוחשבת המקבלת שני טקסטים סרוקים הכתובים בכתב יד, ומחזירה כפלט את הסיכוי (באחוזים) שהטקסטים השונים נכתבו על ידי אותו אדם. המוצר הסופי מיועד לשימוש ע"י המרכז הארצי לבחינות והערכה כדי לזהות רמאות בבחינות, אך הוא יכול להוות פתרון במגוון רחב של תחומים (לדוגמה: עבור מחלקת הזיהוי הפלילי במשטרה). בשלב זה הפרויקט מותאם לשפה העברית אך הוא נכתב בצורה גנרית וניתן להרחיבו לשפות נוספות.

*מבוא*

בחינת הכניסה הפסיכומטרית לאוניברסיטאות היא כלי לחיזוי סיכויי ההצלחה בלימודים במוסדות להשכלה גבוהה. הבחינה משמשת ככלי מיון לכניסה לאוניברסיטאות ולמכללות השונות. הבחינה נבנית על ידי "המרכז הארצי לבחינות ולהערכה" (מאל"ו) ומתקיימת בשפות: עברית, ערבית, רוסית, צרפתית ובנוסח משולב של אנגלית ועברית.

הבחינה הפסיכומטרית בודקת את יכולות המועמד בשלושה תחומים: חשיבה מילולית, חשיבה כמותית ובאנגלית. בנוסף, ישנה מטלת כתיבה (חיבור) המהווה 10% מציון כלל הבחינה ובה הפרויקט שלנו מתמקד. מטלת הכתיבה היא החלק הראשון בבחינה, בה נדרש הנבחן לכתוב חיבור באורך של 25-50 שורות. המטלה נכתבת בעיפרון, על דף בן 50 שורות המיועד לכך, והיא המטלה היחידה במבחן בה נדרש הנבחן לכתוב בכתב ידו.

המרכז הארצי לבחינות ולהערכה נוקט באמצעים ומאמצים רבים על מנת להבטיח את טוהר הבחינה ולמנוע רמאות מכל סוג ככלל, והעתקות וזיופים בפרט. למרות האמצעים אשר ננקטים כדי למנוע זיופים, עדיין ישנם מקרים של התחזות, בהם מצליחים מועמדים לקבל ציון גבוה בבחינה באמצעות אדם אחר אשר מבצע את הבחינה במקומם.

בעקבות כך, המרכז הארצי לבחינות והערכה מעסיק מומחים לזיהוי כתבי יד, בכדי לנסות לבצע השוואה בין שני מועדי בחינה של נבחנים מסוימים אשר מוגדרים כחשודים, על ידי בדיקה של כתב ידו של הנבחן במטלת החיבור. ההשוואה אמורה לאמת/להפריך חשד עבור נבחן שרימה בבחינה.

השוואת כתב ידו של נבחן בין שני המועדים מתבססת על ההנחה שכתב ידו של כל אדם הוא ייחודי ושניתן על פיו לזהות את כותבו. בנוסף יש הסכמה כי אין לשני בני אדם כתב יד זהה (לחלוטין) ואף האדם אינו יכול לכתוב בשנית באופן טבעי דברים שכתב בעבר בצורה זהה לגמרי. עם זאת, סגנון הכתיבה של האדם מאופיין, מוגדר, אישי וייחודי לכל אדם. פעולת הכתיבה היא תוצאה של שיתוף פעולה בין מערכות אחדות בגוף, כגון הראייה, התפיסה ועיבוד הנתונים, וכן פעולות מוטוריות של שרירים רבים. שיתוף כל אלה לפעולות שגרתיות המתבצעות מדי יום, מפתח מיומנות שהיא הבסיס לסגנון כתב היד על תכונותיו הייחודיות.

פרויקט מחקרי זה הינו ביוזמה של המרכז הארצי לבחינות והערכה, בהנחייתו של דר' יהודה חסין. מתוך ההנחה כי קשה להכריע ששני כתבי יד זהים או שונים, נרצה לבנות מערכת שמקבלת כקלט שני חיבורים סרוקים של אותו נבחן (ממועדים שונים). מערכת זו תקבע את הסיכוי שהחיבורים שייכים לשני אנשים שונים ככלי עזר לזיהוי רמאות בבחינה.

*תיאור הבעיה*

מידי שנה ניגשים כ- 70,000 נבחנים לבחינה הפסיכומטרית, כשליש מתוכם ניגשים לבחינה בשנית. על מנת להבטיח את טוהר הבחינה, מאל"ו מעסיק מומחים לזיהוי כתב יד אשר מבצעים בדיקה ידנית של מאות ואלפי בחינות.

במטרה לייעל את תהליך אימות הנבחן ולצמצם את מספר הבדיקות הנעשות ע"י המומחים, מאל"ו זקוק למערכת ממוחשבת שתייעל את תהליך הבדיקה.

באמצעות המערכת ניתן יהיה לצמצם את כמות הבדיקות אשר מגיעות לידי המומחים כך שרק הנבחנים שזוהו ע"י המערכת כחשודים, יגיעו לידיהם לבדיקה ידנית מעמיקה.

## הבעיה מבחינת הנדסת תוכנה

על מנת לבצע השוואה בין שני כתבי יד ואימות המחבר, נדרש תהליך המורכב משלושה חלקים מרכזיים איתם נתמודד בפרויקט: **גילוי, זיהוי ואימות**, עליהם נפרט בתיאור הפתרון.

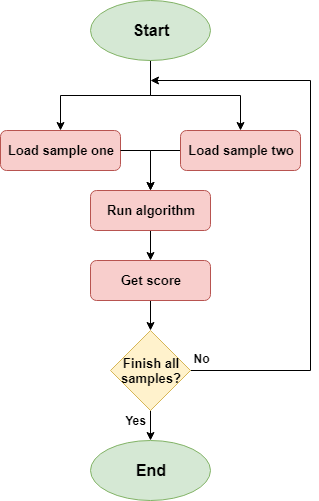
בניגוד לזיהוי כתב של תמונה עם טקסט מודפס, הנחשב קל יחסית לביצוע, התוצאות לגילוי וזיהוי

כתב יד הן חלקיות בלבד. בנוסף, אין מסד נתונים בשפה העברית של כתבי יד שבו ניתן להשתמש כדי לאמן מכונה, לעומת השפה האנגלית בה המידע נגיש ונפוץ יותר ברחבי האינטרנט.

האתגר הקשה ביותר עמו נאלץ להתמודד בפרויקט הוא היכולת לקבוע בסבירות גבוהה האם שני כתבי יד שייכים לאותו אדם או לא. אתגר זה נובע מכמה סיבות:

* בכתב יד, בניגוד לכתב מודפס, המרווחים בין האותיות, המילים והשורות אינם קבועים, דבר המקשה בגילוי המילים והאותיות אשר נשתמש בהם להשוואה בין כתבי היד.
* כל בדיקה של שני טקסטים היא בדיקה אינדיבידואלית. כמות המידע (הטקסט) שיש ברשותנו על כל נבחן מוגבלת ויכולה להשפיע על תוצאות הבדיקה.
* מאל"ו דורש כי הפרויקט יתמקד תחילה בחיבורים הכתובים בשפה העברית, אך שיהיה ניתן לבצע הרחבה לחיבורים בשפות נוספות כמו השפה הערבית. כלומר עלינו לבצע את הפרויקט בצורה גנרית ככל שניתן כדי להתאימו לשפות נוספות כך שעיקר העבודה תהיה באיסוף ה- data של השפה הדרושה ואימון דומה של המודלים.
* מבדיקה ראשונית שביצענו בספרות עולה כי הבעיה איתה אנו מתמודדים מורכבת והניסיונות לפתור אותה הניבו תוצאות חלקיות בלבד.

*תיאור הפתרון*



תרשים 1 - high level design

*בתמונה - High level design*

הפתרון הוא יצירת מערכת ממחושבת, שמקבלת כקלט מהמשתמש שני חיבורים סרוקים של כתב יד בעברית. עם קבלת פקודה, התוכנה תבצע ניתוח והשוואה של כתבי היד, ותחזיר כפלט את הסיכוי שהחיבורים נכתבו על ידי אותו אדם.

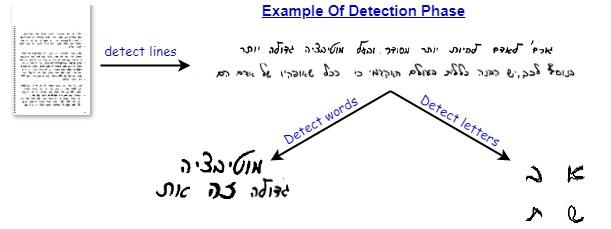
בפרויקט זה השתמשנו בחלק מהאלגוריתמים שמומשו ע"י הסטודנטים מורן זרגרי ואיתי חפץ בפרויקט גמר משנה שעברה שבוצע בהנחייתו של דר' יהודה חסין, אשר חקרו את נושא זיהוי כתב יד והמרתו לטקסט ממוחשב. מטרת הפרויקט הייתה לתת למשתמש כלי להמרה של טקסטים בכתב יד לטקסט ממוחשב, על מנת לאפשר לו את היכולת לסדר ולשתף את המסמכים השונים וכן לבצע שינויים ומניפולציות על הטקסט. בפרויקט זה תוצאות זיהוי הטקסט היו חלקיות בלבד ולא מספיק בכדי להמיר את כל הטקסט מכתב יד לכדי כתב ממוחשב.

בכדי לבצע את תהליך הזיהוי של כתב היד מהתמונה, הבעיה הופרדה לשני שלבים מרכזיים: גילוי (detection) וזיהוי (recognition). שלבים אלו יהיו גם חלק מארכיטקטורת המערכת שלנו עם התאמות ושינויים לצרכים שלנו. יתר על כן, הפרויקט שלנו מצריך שלב נוסף: שלב האימות. שלב זה הוא החלק המרכזי והמורכב בפרויקט שהוא ביצוע ההשוואה בין שני החיבורים ואימות המחבר. בכדי שנוכל לבצע את ההשוואה, נרצה משלב הגילוי והזיהוי לחלץ מילים ואותיות שישמשו אותנו כגורמי השוואה.

פירוט השלבים:

1. **שלב הגילוי** -(detection)– שלב זה מורכב מניתוח התמונה ומציאה של החלקים המכילים את הטקסט אותו יש לפענח. תהליך הגילוי מורכב ממספר תתי שלבים עיקריים:
2. עיבוד מקדים- את החיבורים אנו מקבלים בפורמט tiff/tif/png ממאל"ו. כל חיבור מכיל עמוד אחד או שניים, לכן בחרנו לאחד את העמודים לכדי תמונה אחת. לאחר מכן, אנו מבצעים חיתוך של קצוות התמונה במטרה ללכוד רק את הטקסט הנכתב על ידי הנבחן.
3. גילוי שורות – מציאת השורות בטקסט מתוך כתב היד בתמונה. גילוי השורות נעשה ע"י מציאת הרווחים בין השורות באמצעות אלגוריתם שעובר בצורה רוחבית על התמונה וסוכם את הפיקסלים של כל שורה. מבחינה גרפית, מתקבלת פונקציה בעלת הרבה 'פיקים' כך שנקודות המינימום הקרובות לכל פיק משמאל ומימין, הן הנקודות שהאלגוריתם זיהה לתחילת השורה וסוף השורה (כלומר הרווח שמעל ומתחת לשורה).
4. גילוי מילים – הפרדת השורות למילים.
5. גילוי אותיות – הפרדת השורות לאותיות.

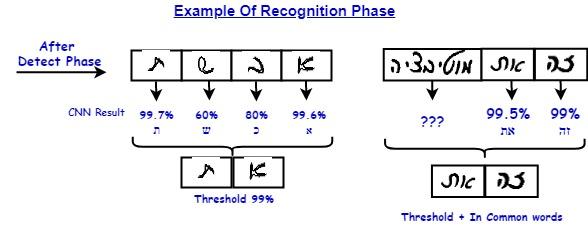
בדומה לגילוי השורות, גילוי האותיות והמילים מתבצע ע"י מציאת הרווחים בין מילה למילה, ובין אות לאות. ראה דוגמא לשלב הגילוי בתרשים 2.



תרשים

1. **שלב הזיהוי** (recognition)– לאחר שלב הגילוי שבאמצעותו מצאנו את המילים והאותיות בטקסט, נרצה לזהות אילו מילים ואותיות גילינו:
2. זיהוי אותיות: לאחר הפרדת השורות, אנו משתמשים במודל (רשת נוירונים) שאומן לזהות את 27 האותיות בשפה העברית (מהפרויקט של שנה שעברה). נרצה לשמור לשלב הבא רק אותיות שהאלגוריתם זיהה בהסתברות גבוהה (כ- 99%). רמת הבטחון בזיהוי האות חשובה לנו כיוון שבשלב ההשוואה נרצה להשוות בין אותיות **זהות בלבד**.
3. זיהוי מילים: אנו נסתפק בזיהוי של חלק מהמילים בחיבור ולא כולן. כלומר, בשלב זה נרצה לסנן את המילים אשר באמצעותן נערוך את תהליך האימות. בחרנו לפעול בדרך זאת על מנת שנוכל "לאמן מכונה" לזהות מספר מצומצם של מילים נבחרות, אשר בדקנו כי חוזרות על עצמן בסבירות הכי גבוהה בקרב החיבורים. בכך נייעל את תהליך איסוף ה- data שישמש אותנו באימון רשת נוירונים. [הסבר על אופן בחירת המילים בנספחים].

ראה דוגמא לשלב הזיהוי בתרשים 3.



תרשים

לפני שנתאר את שלב האימות, חשוב לציין מה הסיבות שהקשו על השגת מלוא המטרות של הפרויקט משנה שעברה:

* אלגוריתם לגילוי השורות התקשה לבצע הפרדה מדויקת של טקסט בעל שורות עם נטייה אלכסונית חדה ולא אחידה.
* אלגוריתמי הפרדת השורות למילים ולאותיות התקשו להפריד כתבי יד מחוברים, צפופים ומקושקשים ולכן חיתוך האותיות והמילים לא תמיד היה במיקום מדויק.
* אחוז הדיוק של רשת הנוירונים היה יחסית גבוה. עם זאת, ישנן אותיות מסוימות שהרשת לא הצליחה לזהות באחוזים גבוהים ולכן זה לא היה מספיק לצרכי המערכת כדי לספק כלי אמין.

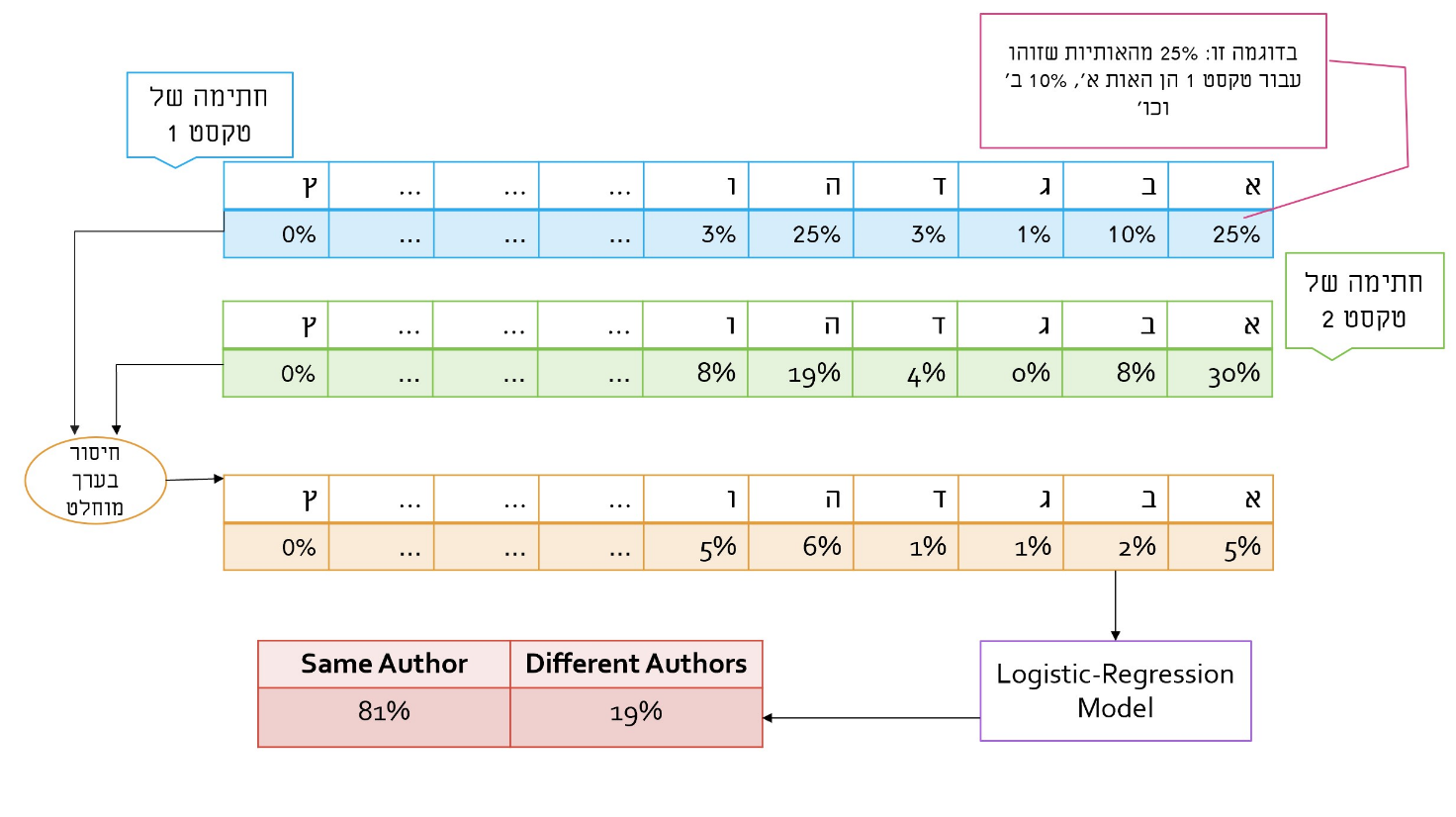
כעת נשאלת השאלה מדוע בכל זאת החלטנו להשתמש בחלקים מהפרויקט הקודם לטובת הפרויקט שלנו? ובכן, הסיבה המרכזית היא שמטרת הפרויקט שלנו שונה ממטרת הפרויקט הקודם. למערכת שלנו אין צורך לזהות ב- 100% את כל האותיות והמילים על מנת לבצע את ההשוואה בין שני הטקסטים. מספיק שנזהה בביטחון גבוה רק חלק מהאותיות והמילים שבטקסט (ואת השאר "נזרוק") כדי להיות מסוגלים לבצע השוואה. את הנחה זו ביססנו על המאמר Hybrid Feature Learning for Handwriting Verification [ראה סקר שוק] שהציג תוצאות גבוהות באימות המחבר ע"י **2 דגימות בלבד** של המילה 'and' בשפה האנגלית ואנו נממש חלק מהארכיטקטורה שהוצגה בו.

1. **שלב האימות (**verification**)**– שלב זה הוא השלב המרכזי בו מתקבלת ההחלטה האם שני החיבורים נכתבו על ידי אותו אדם או לא. כדי להשיג מטרה זו מימשנו לאחר ביצוע מחקר וניסויים, שלושה אלגוריתמים:
2. אלגוריתם קוף
3. השוואת אותיות-
   1. Auto-Encoder
   2. SSIM

לבסוף, אנו משלבים את תוצאות שלושת האלגוריתמים לכדי תוצאה אחת.

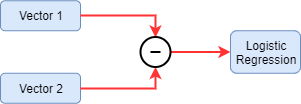
נרצה לציין כי שלושת האלגוריתמים מבצעים השוואה בין החיבורים ברמת האות, כלומר גורמי השוואה הינם האותיות אותן זיהינו בטקסט. במהלך הפרויקט רצינו לבצע השוואה גם ברמת המילה, כלומר השוואה בין אותן מילים משני החיבורים, אך כיוון זה לא צלח ובהמשך נפרט את הסיבות לכישלון [ראה תהליך בניית שלב האימות].

1. אלגוריתם קוף – במהלך המחקר, לאחר שהרצנו מספר בדיקות של זיהוי האותיות, גילינו כי המודל שבו אנו משתמשים מצליח לזהות בהסתברות גבוהה אותיות שונות עבור אנשים שונים. כלומר, שמנו לב כי למשל עבור נבחן א' נזהה בדיוק רב את האותיות א', ח', ע', ש', ועבור נבחן ב' נזהה בדיוק רב אותיות אחרות כמו ב', ג', ח', ה', ק'. מתוך תוצאות אלו, הסקנו כי ניתן לנצל את 'חוסר הדיוק' של המודל. כלומר, אם עבור כתב יד של נבחן א' זיהינו בדיוק רב סט כלשהו של אותיות (ואותיות אחרות לא זיהינו), הרי שהאלגוריתם שלנו ידע לזהות את סט אותיות אלו גם בחיבור אחר שנכתב על ידי אותו אדם. אם בחיבור הנוסף האותיות שזוהו אינן תאומות, או תואמות חלקית, הרי שזה יכול להוות אינדיקציה על כך שהחיבורים לא נכתבו על ידי אותו אדם. אנו מגדירים סט אותיות אלו כ"חתימה" של הנבחן. קראנו לאלגוריתם זה אלגוריתם 'קוף' מכיוון שאנו מנצלים את חוסר היכולת של המודל לזהות בדיוק של 100% את כל האותיות. כלומר, לא נזדקק 'לעבוד קשה' ולבצע השוואה אמיתית בין הגורמים השונים בכדי לקבל אינדיקציה האם מדובר באותו אדם או לא ורק נשווה בין החתימות השונות. ראה דוגמא לאלגוריתם קוף בתרשים 4.



תרשים – אלגוריתם קוף

עבור כל חיבור, נגדיר וקטור בגודל 27 (כמספר האותיות בשפה העברית). בכל תא בוקטור נשמור את כמות המופעים (באחוזים) מכל אות שהמודל הצליח לזהות (תא 0 האות א' וכו'). לאחר מכן, נבצע חיסור בערך מוחלט בין שני הווקטורים משני החיבורים. את וקטור החיסור נעביר למודל סיווג נוסף (Logistic Regression). המודל אומן לזהות וקטורי חיסור של חיבורים שנכתבו על ידי אותו אדם ועל ידי אנשים שונים.



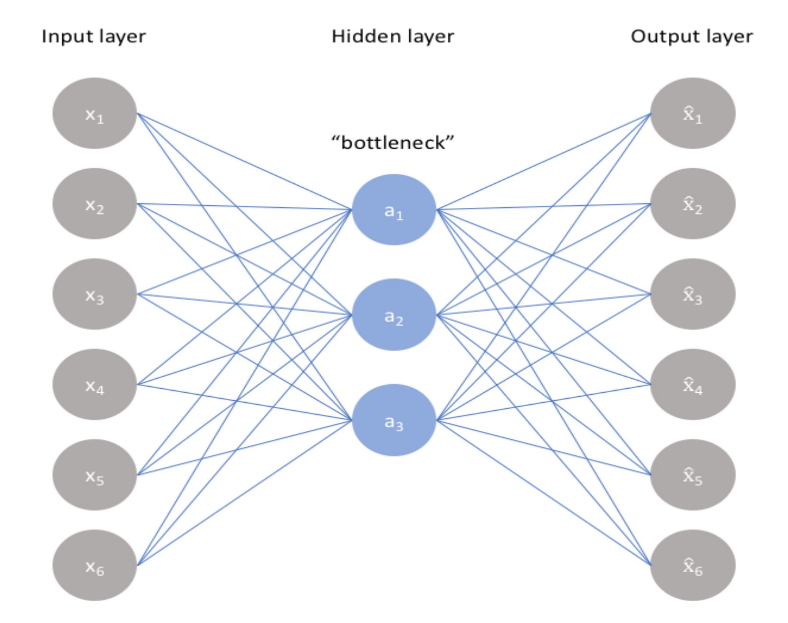
תרשים 5 – monkey high level

1. השוואת אותיות - לאחר שלב הזיהוי, נשארנו עם גורמי ההשוואה משני החיבורים (האותיות שזוהו). כעת נרצה לבצע השוואה בין האותיות השוות משני החיבורים (א' מול א', ב' מול ב' וכו')

ההשוואה בפועל מתבצעת בעזרת שימוש בשני אלגוריתמים: Auto-Encoder, SSIM

2.1אלגוריתם Auto-Encoder

מודל למידה לא מפוקח (unsupervised learning) המבוסס על רשת נוירונים, כדי לבצע representation learning.



תרשים 6 – Auto-Encoder neural network

המטרה העיקרית של שימוש בסוג כזה של רשת נוירונים היא כיווץ קלט מסוים שממנו מתקבל מספר קטן יותר של פיצ'רים שבעזרתם ניתן יהיה לשחזר את הקלט המקורי.

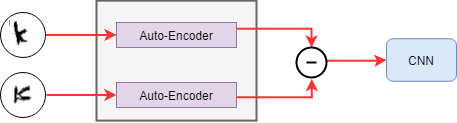
הרשת יוצרת "צוואר בקבוק" שבאמצעותו מבוצע כיווץ של הקלט המקורי לכדי מספר קטן של פיצ'רים. תהליך זה יכול להתבצע עבור קלטים בהם יש קשר בין הפיצ'רים השונים (כלומר יש תלויות, מבנה כלשהו) אחרת זו הייתה משימה כמעט בלתי אפשרית.

הפיצ'רים שנוצרים בשכבה של צוואר הבקבוק הם הפיצ'רים החשובים של הקלט המאפיינים אותו. במקרה שלנו הקלטים הן תמונות של אותיות.

תיאור האלגוריתם-

כל תמונה של אות שזוהתה נשמרת בגודל 28\*28 פיקסלים, כלומר התמונה מורכבת סה"כ מ- 784 פיקסלים (פיצ'רים). באמצעות שימוש ב Auto-Encoder, אנו יכולים לחלץ את 32 הפיצ'רים החשובים ביותר (כלומר "חתימה" של האות). ההנחה היא שעבור 2 אותיות (לדוגמא שתי אותיות א') שנכתבו על ידי אותו כותב, הפיצ'רים יהיו דומים. עבור כל זוג חיבורים אנו מבצעים השוואות בין אותיות שוות כפי שמתואר בתרשים 7. רשת הנוירונים מחזירה ערך בתחום [0,1] כך שאחד מתאר התאמה מושלמת.

הקביעה הסופית של האלגוריתם (החיבור נכתב ע"י אותו אדם או לא) מתקבלת ע"י ממוצע התוצאות שהתקבלו בין כל ההשוואות שבוצעו.



28

28

28

28

תרשים 7 – Auto-Encoder high level

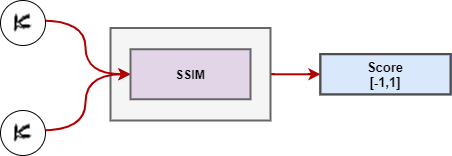
Neural Network

2.2 אלגוריתם SSIM - Structural Similarity Index

SSIM הוא אלגוריתם שפותח ע"י Want et el ובאמצעותו ניתן להעריך את רמת הדמיון בין שתי תמונות. עבור שתי תמונות האלגוריתם מחזיר מספר בתחום [1,1-] כך שאחד מתאר התאמה מושלמת. בשונה מאלגוריתמים אחרים המבצעים השוואה בין שתי תמונות, כמו לדוגמה MSE Mean Squared Error)) שעוברים על כל פיקסל בין שתי התמונות ומבצעים השוואה, SSIM עובד בצורה שונה. ניתן להגיד כי SSIM מדמה פעולה של העין האנושית ומסתכל על קבוצות של פיקסלים במקום על פיקסלים בודדים וכך יכול להבחין בשוני ובדמיון בין התמונות בצורה יותר חכמה.

ההנחה היא שאם נבצע השוואה בין מספר רב של אותיות משני החיבורים, נקבל בהסתברות גבוהה כמות גדולה יותר של זוגות אותיות שקיבלו תוצאה גבוהה כאשר החיבורים נכתבו ע"י אותו אדם, וכמות גדולה יותר של זוגות אותיות שקיבלו תוצאה נמוכה אאשר החיבורים נכתבו ע"י אנשים שונים.

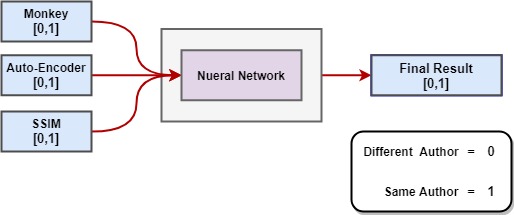
בדומה למה שתיארנו ב- Auto-Encoder וכפי שניתן לראות בתרשים 8, כל זוג אותיות זהות משני החיבורים, עובר דרך SSIM שמחזיר לנו תוצאה. הקביעה הסופית של האלגוריתם (החיבור נכתב ע"י אותו אדם או לא) מתקבלת ע"י ממוצע התוצאות שהתקבלו בין כל ההשוואות.



תרשים 8 – SSIM high level

שילוב שלושת האלגוריתמים לכדי תוצאה אחת

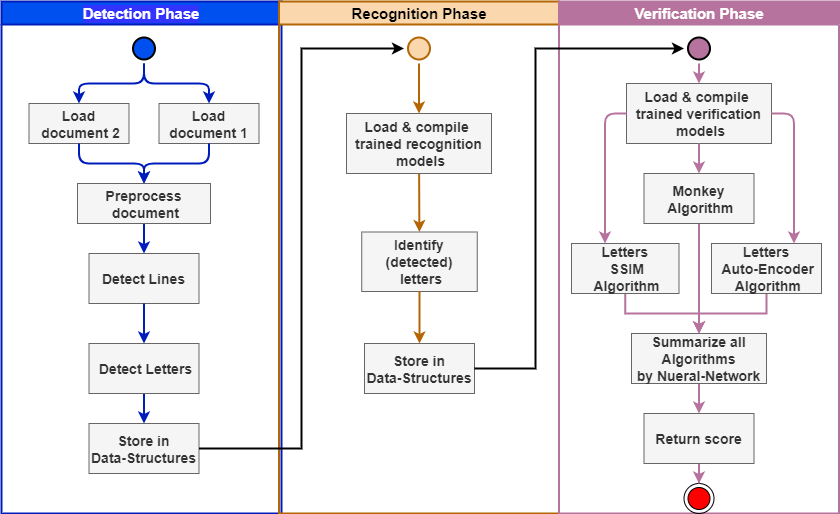
עד כה תיארנו את שלושת האלגוריתמים שמבצעים את ההשוואה בפועל. מטרת הפרויקט היא להחזיר תשובה בינארית באחוזים, האם החיבורים נכתבו על ידי אותו אדם או לא. לכן כפי שניתן לראות בתרשים 9, אימנו רשת נוירונים נוספת המקבלת כקלט את תוצאות שלושת האלגוריתמים ומחזירה כפלט כמה באחוזים החיבורים דומים (נכתבו ע"י אותו אדם) וכמה שונים (נכתבו ע"י אנשים שונים).



תרשים 9 - Final score

## ארכיטקטורת המערכת

תרשים 10 מתאר את ארכיטקטורת המערכת לפי החלוקה של שלושת השלבים המרכזיים: גילוי, זיהוי, אימות.



תרשים 10 – System Architecture

## טכנולוגיות

את ה-backend של הפרויקט בחרנו לממש בשפת python. בחירה זו נעשתה מכיוון שאנו משתמשים בעיבוד תמונה, למידת מכונה, חישובים מתמטיים ועוד. לכל אלו קיימות ספריות רבות, חזקות ונוחות לשימוש בשפה זו (NumPy, sklearn, pandas, matplotlib, keras PIL, TensorFlow ועוד) מה שהפך את הבחירה למובנת מאליה.

בנוסף את ה-frontend של הפרויקט כתבנו בעזרת HTML, CSS, Javascript. החלטנו להשתמש בטכנולוגיה זו כדי להשיג חווית משתמש נוחה ויפה. בכדי לחבר את ה-frontend אל ה-backend, השתמשנו בספרייה בשם eel אשר בעזרתה ניתן לחשוף פונקציות בין הסקריפטים של ה-python לבין ה-javascript.

*תהליך בניית שלב האימות*

כפי שתיארנו בדוח אלפא, התכנון המקורי היה לבצע השוואה באמצעות שלושה אלגוריתמים, שניים ברמת האות (Letters Auto-Encoder, Monkey) ואחד ברמת המילה (Words Auto-Encoder).

כדי שנוכל לזהות את המילים בטקסט לאחר שלב הגילוי, נדרשנו לאמן מודל לזיהוי מילים [ראה הסבר על אופן בחירת 'המילים הנפוצות' בנספחים]. מכיוון שלא היה ברשותנו מאגר של מילים בכתב יד, אספנו data שישמש אותנו לאימון מודל לזיהוי המילים הנפוצות באמצעות חלוקת טפסים לכ-300 אנשים שונים אשר רשמו את המילים בכתב ידם (כלומר, אספנו בסה"כ 3,600 מופעים מכל מילה) [ראה בנספחים דוגמא של טופס איסוף המילים].

מהטפסים שאספנו, חילצנו בצורה ממוינת את המילים (11 מילים שונות) וביצענו אימון של בconvolutional neural network. על מילים אלו הפעלנו data augmentation שבאמצעותו ניתן להגדיל את ה-data עליו אנחנו לומדים באמצעות שכפול של התמונות עם שינויים קטנים (הזזה בזווית, הגדלה, הקטנה וכו'). לאחר ניסיונות רבים הצלחנו להגיע לאחוזי דיוק של 96 אחוזים (על קבוצת ה-test מה-data שאספנו). למרות זאת, כשניסינו להשתמש במודל המאומן עבור חיבור אמיתי, המודל נאלץ לסווג גם מילים שלא אומן עליהם (מילים שלא חלק מהמילים הנפוצות). גילינו כי המודל מסווג חלק ממילים אלו בצורה לא נכונה, כלומר נותן אחוזים גבוהים שהמילה שייכת לאחת מהמילים הנפוצות. כלומר, קיבלנו זיהוי נכון של חלק מהמילים בשילוב עם קבוצת מילים "זבל".

החלטנו לנסות לשפר עוד את המודל, בכך שהוספנו מחלקה נוספת, מחלקת "זבל" בכדי לנסות לגרום למילים שאינן חלק מ'המילים הנפוצות' להיות מסווגות אליה, ובכך לבצע סינון טוב יותר של המילים. מהבדיקות שעשינו אכן הוספת מחלקה זו שיפר את תוצאות הזיהוי אך לא מספיק בכדי שנוכל לסמוך על כך שבשלב ההשוואה, בו אנו נשווה בין מילים זהות משני הטקסטים.

תוך כדי תהליך בניית המודל, הבנו כי קיימת בעיה בשימוש מודל לזיהוי מילים המבוסס על 11 מילים בלבד (לצרכי המערכת שלנו). הרעיון הראשוני ליצור מודל לזיהוי 'המילים הנפוצות' נבע מכך שהקלט עליו אנו עובדים הוא כמעט אינסופי (כלומר יש כמעט אינסוף מילים אפשריות). מכיוון שאין באפשרותנו להשיג מאגר המכיל את כל המילים לצורך אימון, בחרנו להתמקד במילים הנפוצות. למעשה רק בתהליך מימוש המודל גילינו כי הדרך בה בחרנו לפעול קשה (ואולי גם לא אפשרית במסגרת זמן הפרויקט) ולכן החלטנו לוותר על השוואת המילים ולהתמקד בהשוואה ברמת האות בלבד.

מכיוון שהחלטנו לוותר על כיוון זה, התחלנו בחיפוש אחר אלגוריתם נוסף לשלב ההשוואה שיוכל לשמש אותנו ברמת האות. אלגוריתם נוסף הוא הכרחי מכיוון שבשימוש בשני אלגוריתמים יכולות להתקבל תוצאות 'קונפליקט', כלומר ששני האלגוריתמים להשוואה קובעים תשובה שונה, מצב שבו אין לנו אפשרות לקבל החלטה סופית. לכן כדי לפתור זאת, החלטנו להשתמש ב- SSIM ברמת האות, אשר תרם לפתירת בעיית הקונפליקטים, מכיוון שבעזרתו יהיו לפחות שתי תשובות זהות (כלומר נקבל רוב).

*שלבי בניית ומימוש האלגוריתמים*

1. שלב א' - תכנון, הכנה ואימון

1.0.1 ה-data לשלב האימון

* + קיבלנו בתחילת הפרויקט כ-500 חיבורים סרוקים השייכים לאנשים שונים אשר ישמשו אותנו בשלבי הלמידה.
  + בהמשך הפרויקט קיבלנו כ- 106 חיבורים נוספים אשר מהווים השלמה לזוגות (מתוך 500 החיבורים הראשונים). כלומר יש ברשותנו 106 זוגות חיבורים השייכים לאותו אדם (ממועדים שונים).

1.1 אלגוריתם קוף

* בתחילת הפרויקט כאשר היו בידינו 500 חיבורים השייכים לאנשים שונים, היינו צריכים למצוא דרך שבה נוכל לאמן את המודל לזהות חיבורים השייכים לאותו אדם. מכיוון שלא היו ברשותנו בשלב זה זוגות חיבורים של אותו אדם, החלטנו לחלק כל חיבור לשני חצאים (לפי השורות שמצאנו בכל חיבור), ובעצם כל חצי מהווה "חיבור" של אותו כותב. כלומר, יצרנו 500 באופן מלאכותי זוגות של חיבורים השייכים לאותו אדם. בנוסף לקחנו עוד כ-500 חצאי חיבורים השייכים לאנשים שונים ועליהם ביצענו את הלמידה.
* מכל "חיבור" חילצנו וקטור בגודל 27 [ראה תיאור אלגוריתם קוף]. ביצענו חיסור בין הווקטורים ואת התוצאה העברנו למודל סיווג נוסף (Logistic regression) שילמד לזהות וקטורי חיסור השייכים לאנשים שונים, ווקטורי חיסור השייכים לאותו אדם.
* לאחר שקיבלנו את 106 החיבורים המהווים השלמה לזוגות של חיבורים בעלי אותו מחבר, הוספנו אותם ללמידה ויצרנו מודל חדש ומשופר.

1.2 אלגוריתמי השוואת אותיות (Auto-Encoder, SSIM)

תכנון והכנה של ה-data לאלגוריתמים אלו כללו מספר שלבים:

* כאמור, כדי לחלץ את האותיות להשוואה לצורכי האימון, השתמשנו במודל לזיהוי אותיות אשר פותח בפרויקט של מורן זרגרי ואיתי חפץ. המודל הצליח לזהות את האותיות, אך לא בדיוק גבוה, ולכן השתמשנו ב- threshold של כ- 99.5%. רק במידה שהזיהוי הצליח מעבר לכך, שמרנו את האות בתוך מאגר האותיות שזוהו בהצלחה.
* למרות שהשתמשנו ב- threshold גבוה, חלק מהאותיות לא זוהו בצורה טובה. כמו כן, חלק מהאותיות בשפה זוהו בכמות נמוכה יחסית לשאר האותיות.
* בשביל לאמן את ה Auto-Encoder נאלצנו למיין ידנית אותיות שזיהינו מהחיבורים באמצעות המודל לזיהוי אותיות הקיים, דבר שדרש מאיתנו הרבה זמן (המודל חילץ כ- 3,000-25,000 תמונות של כל אות בשילוב עם אותיות 'זבל').
* לאור הבעיות שהוזכרו לעיל, החלטנו לצמצם את מספר האותיות עליהן נבצע השוואה. בבחירת האותיות היו שלושה פרמטרים עיקריים:
  + המודל הקיים מזהה את האות באחוזי דיוק גבוהים.
  + כמות המופעים של האות שחולצו מהחיבורים הייתה גבוהה (לדוגמא המודל אינו מזהה את האותיות ג', ז', ץ' בצורה טובה).
  + אותיות בעלות צורה ייחודית באופן יחסי (לדוגמא האותיות ו', י', ן' די דומות פרט לאורך האות).
* האותיות שנבחרו: ב', ד',ה', ח', מ', ל', פ', ס', ם'.
* למרות זאת, עדיין חלק מהאותיות סווגו בצורה לא נכונה (לדוגמה, חלק ממילה סווג כאחת האותיות). על מנת לפתור בעיה זו, אימנו מודל נוסף אשר נועד לוודא את סיווג האותיות. המודל החדש אומן לזהות את האותיות הנ"ל (באמצעות ה-data שאספנו מהחיבורים) ובנוסף גם ערכי 'זבל', כלומר תמונות שחולצו וסווגו לא נכון ואינן חלק מתהליך ההשוואה שברצוננו לנקות (כגון חלקי אותיות, קשקושים על הדף, חלקי מחיקות וכו'). באמצעות הסינון הנ"ל הצלחנו לדאוג לך שכ- 99% מהאותיות שזוהו אכן נכונות ובכך שלב ההשוואה יהיה יותר אמין (כדי להבטיח שהאותיות שאנו משווים ביניהן אכן שוות).
* לאחר מיון האותיות אימנו מודל Auto-Encoder שכאמור מחלץ עבור כל אות את 32 הפיצ'רים החשובים.
* השלב האחרון היה לבצע אימון לסיווג בין זוגות של אותיות השייכים לאותו אדם וזוגות השייכים לאנשים שונים. מכל אות בזוג חילצנו באמצעות ה- Auto-Encoder שאומן, וקטור באורך 32 המייצג את הפיצ'רים החשובים וביצענו חיסור בין שני הווקטורים (תוצאת החיסור היא גם וקטור באורך 32). את וקטור החיסור העברנו למודל סיווג נוסף (רשת נוירונים) שילמד לזהות האם וקטור זה מאפיין זוג אותיות אשר נכתבו ע"י אותו אדם או לא.

1. שלב ב' - בניית המערכת

* לאחר שביצענו אימון לכל המודלים הנ"ל, מימשנו את ה- flow של המערכת כפי שמתואר בארכיטקטורה.

1. שלב ג' - בדיקת המערכת

3.1 תכנון הניסוי

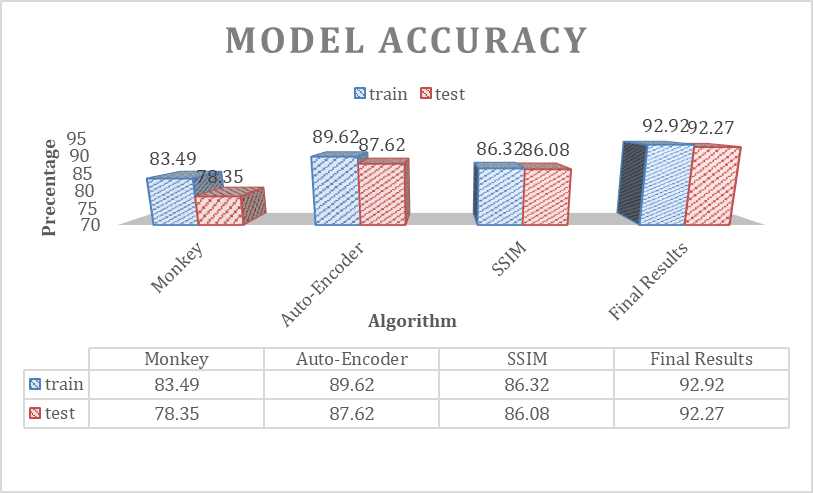
* כאמור, יש בידינו 106 זוגות חיבורים השייכים לאותו אדם, שאיתם ביצענו גם את האימון לאלגוריתמים (קבוצת ה- train).
* לקראת סוף הפרויקט, קיבלנו בנוסף 97 זוגות חיבורים שעליהם לא בוצעו שלבי למידה כלל (מהווים קבוצת test נקייה).
* את הניסוי ביצענו בנפרד על 106 זוגות החיבורים (קבוצת ה-train) ועל 97 זוגות החיבורים (קבוצת ה-test)
* בכל ניסוי לקחנו מספר שווה של זוגות השייכים לאותו אדם, וזוגות רנדומליים השייכים לאנשים שונים.

3.2 מהלך הניסוי

* עבור כל זוג חיבורים, הרצנו בדיקה של המערכת. בכל בדיקה שמרנו את הנתונים הבאים:
  + תוצאת כל אחד משלושת אלגוריתמי ההשוואה
  + עבור כל אלגוריתם, האם הוא צדק או לא.
  + תוצאה סופית משולבת של שלושת האלגוריתמים
  + האם התוצאה הסופית נכונה או לא.

3.3 תוצאות הניסוי

תרשים 11 מתאר את אחוזי הדיוק של כל אחד משלושת האלגוריתמים בנפרד ובעמודה Final Results את התוצאה המתקבלת כאשר המערכת משכללת את תוצאות שלושת האלגוריתמים לכדי תוצאה אחת. התוצאות המופיעות בגרף מתארות את קבוצות ה-train וה-test.



תרשים 11

כפי שניתן לראות, אחוזי הדיוק של קבוצת ה-train מעט גבוהים יותר מקבוצת ה-test כצפוי, אך לא בפער משמעותי בייחוד עבור התוצאה הסופית. למעשה בניסוי על קבוצת ה-test המערכת צדקה ב- 92.27% מההשוואות שנעשו, כלומר 179 מתוך 194 סווגו נכון.

בטבלה בעמוד הבא, מפורטות תוצאות הניסויים באמצעות confusion matrix באופן הבא:

|  |  |  |
| --- | --- | --- |
| **Predicted Different** | **Predicted Same** |  |
| **FN =**  **False Negative** | **TP =**  **True Positive** | **Actual Same** |
| **TN =**  **True Negative** | **FP =**  **False positive** | **Actual Different** |

**TP-** System predicted 'Same' <->True label is 'Same'

**TN-** System predicted 'Negative' <-> True label is 'Different'

**FN-** System predicted 'Negative' <-> True label is 'Same'

**FP-** System predicted 'Same' <-> True label is 'Different'

למעשה האלכסון שבצבע כחול מתאר את כמות הבדיקות בהן המערכת צדקה ובאדום את כמות הבדיקות בהן המערכת שגתה. חישוב Model Accuracy נעשה באופן הבא:

או במילים פשוטות, כמות הבדיקות שבהן המערכת צדקה לחלק לסה"כ הבדיקות.

|  |  |  |
| --- | --- | --- |
| **Test set**  (97 pairs of same, 97 pairs of different) | **Train set**  (106 pairs of same, 106 pairs of different) |  |
| |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **15** | **82** | **Actual Same** | | **70** | **27** | **Actual Different** |   **Model Accuracy: 78.35%** | |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **6** | **100** | **Actual Same** | | **77** | **29** | **Actual Different** |   **Model Accuracy: 83.49%** | **Monkey results** |
| |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **7** | **90** | **Actual Same** | | **80** | **17** | **Actual Different** |   **Model Accuracy: 87.62%** | |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **2** | **104** | **Actual Same** | | **86** | **20** | **Actual Different** |   **Model Accuracy: 89.62%** | **Auto-Encoder results** |
| |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **21** | **76** | **Actual Same** | | **91** | **5** | **Actual Different** |   **Model Accuracy: 86.08%** | |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **19** | **87** | **Actual same** | | **96** | **10** | **Actual Different** |   **Model Accuracy: 86.32%** | **SSIM results** |
| |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **11** | **86** | **Actual Same** | | **93** | **94** | **Actual Different** |   **Model Accuracy: 92.27%** | |  |  |  | | --- | --- | --- | | **Predicted Different** | **Predicted Same** |  | | **5** | **101** | **Actual Same** | | **96** | **10** | **Actual Different** |   **Model Accuracy: 92.92%** | **Final results** |

*סיכום / מסקנות*

בפגישתנו הראשונה עם דר' יהודה חסין, הוצגו לנו מספר פרויקטים אפשריים. בסופו של דבר, החלטנו לבחור בפרויקט השוואת כתבי יד כיוון שנושא זה גרם לנו לעניין רב מצד אחד ואתגר גדול מצד שני. ממחקר ראשוני שביצענו בספרות, גילינו כי הבעיה איתה אנו מתמודדים מורכבת מאוד, קיימות מספר דרכים שונות לתקוף אותה ואין תשובה חד משמעית כיצד לפעול. בנוסף בנימה אישית, היינו מעט פסימיים לגבי היכולת שלנו לפתור בעיה מורכבת זו עם אחוזי דיוק גבוהים.

בחלקו הראשון של הפרויקט ביצענו מחקר מעמיק, קראנו מאמרים, חפשנו מידע על זיופי חתימות, למדנו את הטכנולוגיות איתן נעבוד (python, image processing, machine learning) ובמקביל ביצענו התאמות מהפרויקט שבוצע בשנה שעברה ע"י מורן זרגרי ואיתי חפץ לפרויקט שלנו. לאחר שהתחלנו לחשוב על דרכים אפשריות לפתור את הבעיה, התחלנו לאסוף data של המילים הנפוצות ובדקנו מספר אלגוריתמים נוספים עבור שלב הגילוי.

לאורך הפרויקט ביצענו חלוקה של הקוד למודולים שונים בעלי תחומי אחריות מוגדרת, השתדלנו לבצע refactor לאורך הזמן ולשמור על הקוד נקי ויציב. כל זאת על מנת לעמוד בדרישת מאל"ו כי הפרויקט יבוצע בצורה גנרית ככל האפשר, כדי לאפשר התאמה שלו לשפות נוספות (בעיקר ערבית).

אתגר מרכזי שליווה אותנו לאורך הפרויקט היה מחסור בדאטה לצורך אימון המודלים. בתחילת הפרויקט הסברנו לנציגי המאל"ו כי לכמות ה-data (כלומר זוגות של חיבורים השייכים לאותו אדם) בו נשתמש לאימון המודלים יש מרכיב מרכזי בהצלחת הפרויקט. בעקבות מגפת הקורונה, פעילות מאל"ו התעכבה וקיבלנו מספר קטן יותר של data ממה שציפינו. בתחילת הפרויקט קיבלנו ממאל"ו 500 חיבורים השייכים לאנשים שונים וברוב שלבי הפרויקט חיבורים אלו היו היחידים שברשותנו. לכן, היינו צריכים לבצע התאמות ולחשוב על דרכים בהן נוכל לעבוד במצב הקיים [כמפורט בתהליך בניית שלב האימות].

לאחר שסיימנו לבנות את המערכת המשלבת את כל אלגוריתמי ההשוואה, ביצענו את הניסויים כפי שתיארנו בבדיקת המערכת. בסופו של דבר לאחר שיפורים וכוונונים של האלגוריתמים הצלחנו להגיע לכ-92% הצלחה בדיוק סיווג החיבורים, תוצאה אשר גבוהה ממה שציפינו בתחילת הפרויקט. עם זאת, אנו סבורים כי ניתן לשפר עוד את התוצאות ע"י אימון המודלים באמצעות כמות גדולה יותר של data וכן ע"י אימון מודלים ברמת המילה שכאמור במסגרת זמן הפרויקט לא הצלחנו לממש.

בנוסף, בנינו desktop application לעטיפת המערכת. באמצעות התוכנה ניתן להריץ את השוואת החיבורים בצורה נוחה וידידותית למשתמש. התוכנה מאפשרת למשתמשים לבחור שני חיבורים להשוואה וכן הרצה של תיקייה שלמה המכילה זוגות של חיבורים. תוצאות ההשוואה מוצגות בגרף המתאר את רמת הביטחון שזוג החיבורים נכתב ע"י אותו אדם או לא. בנוסף, ניתן לשמור את כל התוצאות בטבלת excel [ראה תיאור מסך ה- GUI בנספחים].

בהזדמנות זו, נרצה להודות למנחה הפרויקט דר' יהודה חסין, אשר ליווה אותנו לאורך הפרויקט, הקדיש הרבה מעבר לזמן הנדרש והיה חלק ניכר מהצלחתנו. יהודה עזר לנו רבות במחקר, תכנון האלגוריתמים המרכזיים וכן בניתוח התוצאות ושיפורן.

*סקירת ספרות*

Hybrid Feature Learning for Handwriting Verification

המאמר העיקרי עליו אנו מבססים את ארכיטקטורת השוואת המילים והאותיות (שלב האימות). המאמר מתמקד בזיהוי מחבר על ידי מילה בודדת בשפה האנגלית – and.

במאמר מוסבר כי הבחירה במילה זאת נובעת מכך שהיא המילה הרביעית השכיחה ביותר באנגלית ומכך שלמילה זאת קיים מאגר מידע גדול אשר אפשר להשתמש בו כדי לאמן את האלגוריתם למידה. ההשוואה נעשית ע"י שילוב של רשת נוירונים, Auto-Encoder ואלגוריתם SIFT (שנכון להיום בחרנו עוד לא לממש לצרכים שלנו) לקביעת אחוזי ההתאמה בין 2 מילים.

<https://arxiv.org/pdf/1812.02621.pdf>

Writer verification based on a single handwriting word sample

המאמר עוסק בבעיית זיהוי של מחבר של טקסט בכתב יד. המאמר מתמקד באימות המחבר ומציע גישה חדשה לאימות בהתבסס על מילה אחת בלבד (ללא צורך בהרבה נתונים).

המאמר מתאר את הדמיון בין הבעיה הנתונה לבין זיהוי חתימה המשתמש ב-Levenshtein edit distance, וכן בפתרון המוצע יש שימוש ב- Wagner-Fisher algorithm.

אלגוריתם זה נותן הערכה לעלות השינוי של הפיכת תמונה של מילה מסוימת לתמונה של מילה נוספת (יש לציין כי שתי התמונות של מייצגות אותה המילה) באמצעות הערכת העלות השינוי בין האלמנטים הבסיסיים.

<https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-016-0139-0>

Matching Handwritten Document Images

המאמר עוסק בסכמה של פעולות כדי לבדוק התאמה בין שני כתבי יד מקבצים שונים.

ע"פ המאמר, הבדיקה מבוצעת בצורה המיטבית כאשר במסמכים יש מספר גדול ביותר של התאמות מילים. זיהוי המילים הדומות נעשה בעזרת רשת נוירונים וכמה מאגרי מידע גדולים כדי לאמן אותה. מתוארים בו הקריטריונים אשר שימשו לצורך החלטה האם מדובר באותו הכתב יד: צורת כתב היד, שכיחות המילים, סדר המילים ואוצר המילים.

במאמר מדגישים את הצורך במאגר נתונים גדול כדי לבצע את ההשוואה בצורה הטובה ביותר.

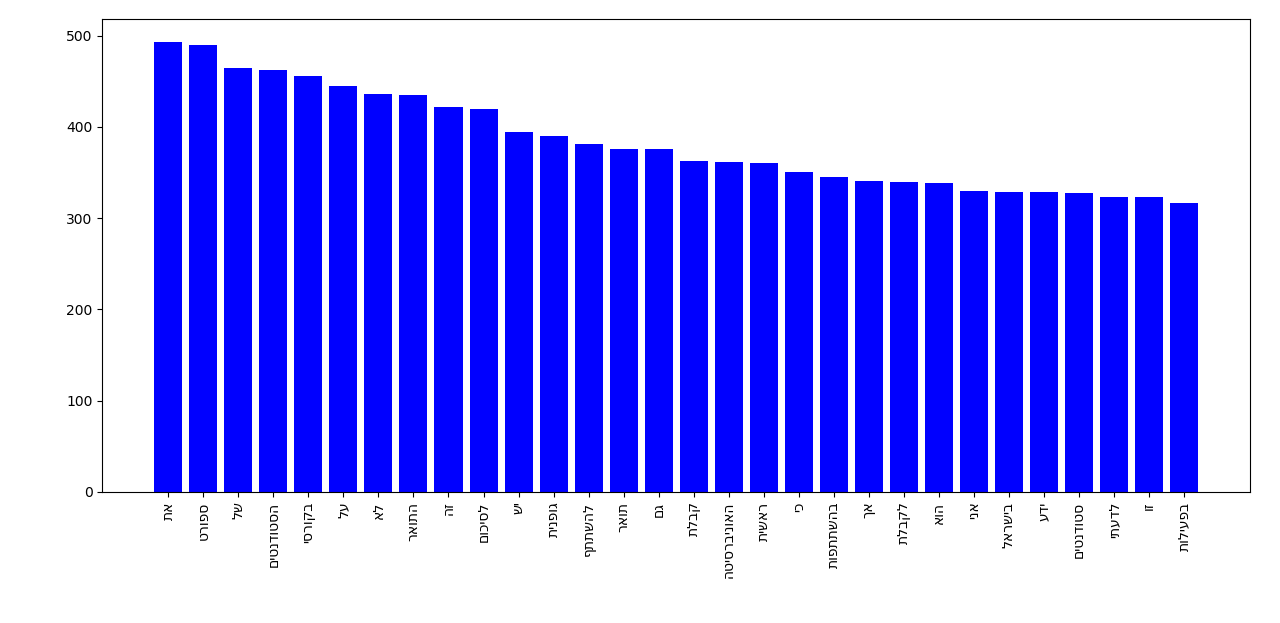
<https://arxiv.org/pdf/1605.05923.pdf>

*נספחים*

\*באישור המנחה, לא בוצעו החלקים שאינם נדרשים מפרויקטים מחקריים.

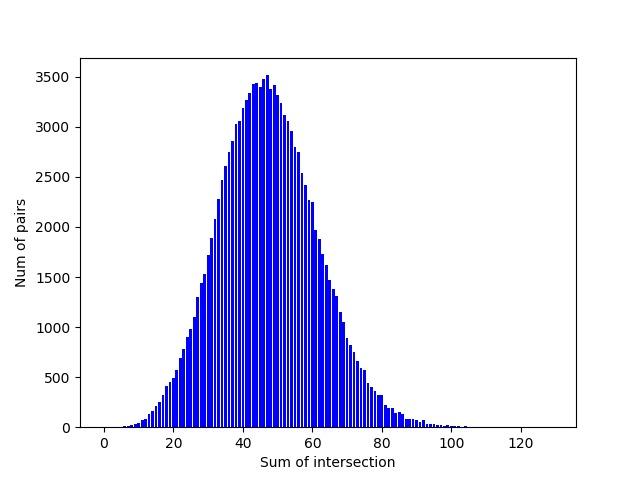
**המשך תיאור הפתרון – אופן בחירת המילים הנפוצות**

בכדי לבחור את המילים הנפוצות, קיבלנו מהמרכז הארצי לבחינות והערכה כ-500 חיבורים מוקלדים, ובעזרת תוכנית שכתבנו חילצנו את המילים שחוזרות הכי הרבה בכל החיבורים. יש לציין כי החיבורים שקיבלנו הם מאותו נושא בחינה, לכן מילים כמו 'אוניברסיטה', 'ספורט', 'סטודנטים' הן מילים ששייכות לנושא החיבור ולכן לא בחרנו אותן כמילים הנפוצות שנבחר לעבוד איתן. בהתבסס על התוצאות, המילים שנבחרו הן: של, לא, את, גם, לסיכום, כי ,זה, זו ,יש ,לדעתי, אני.



תרשים -most common words frequencies

לאחר מכן, כדי לוודא שאכן יש 'חיתוך' בין טקסטים שונים לבין המילים המופיעות בטקסט (כלומר לוודא שנוכל לבצע השוואה בין מילים שוות בין שני טקסטים), כתבנו תוכנית נוספת שמבצעת את הבדיקה הבאה: מכל טקסט חילצנו את המילים הנפוצות שמופיעות בו (המילים שבחרנו מהפסקה הקודמת), ביצענו חיתוך עם כל שאר הטקסטים וספרנו את מספר המילים השוות.

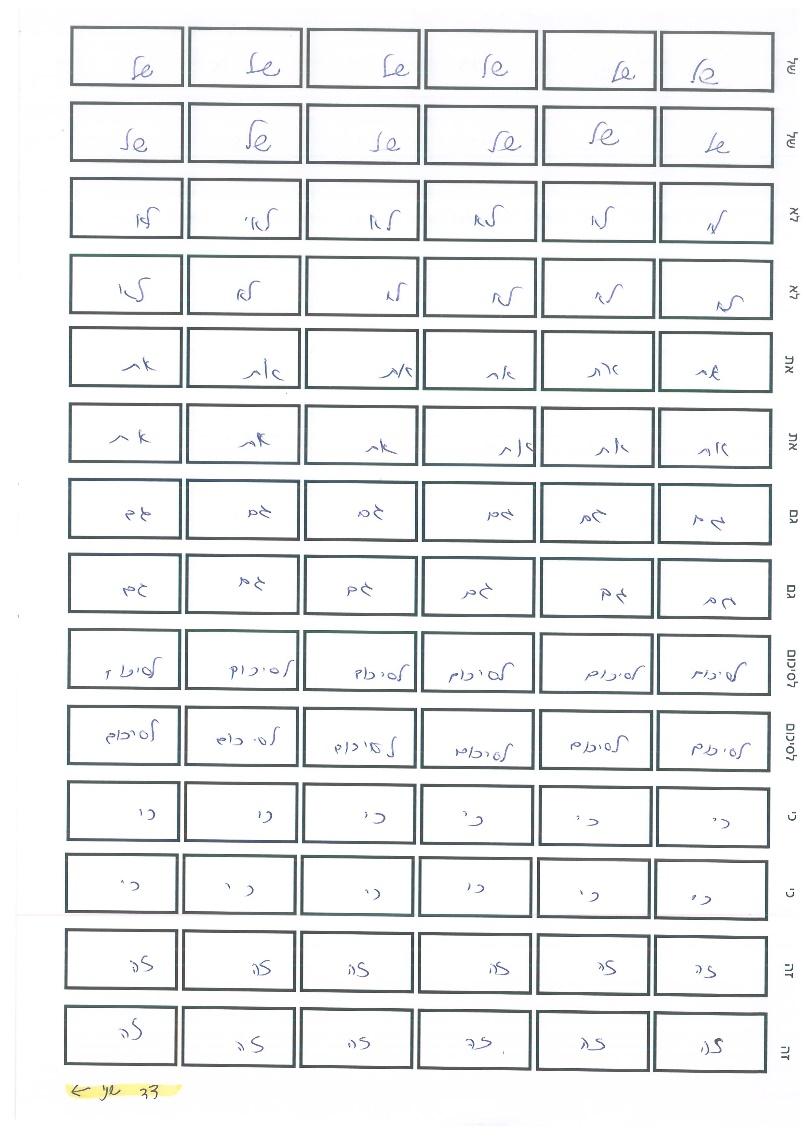


תרשים

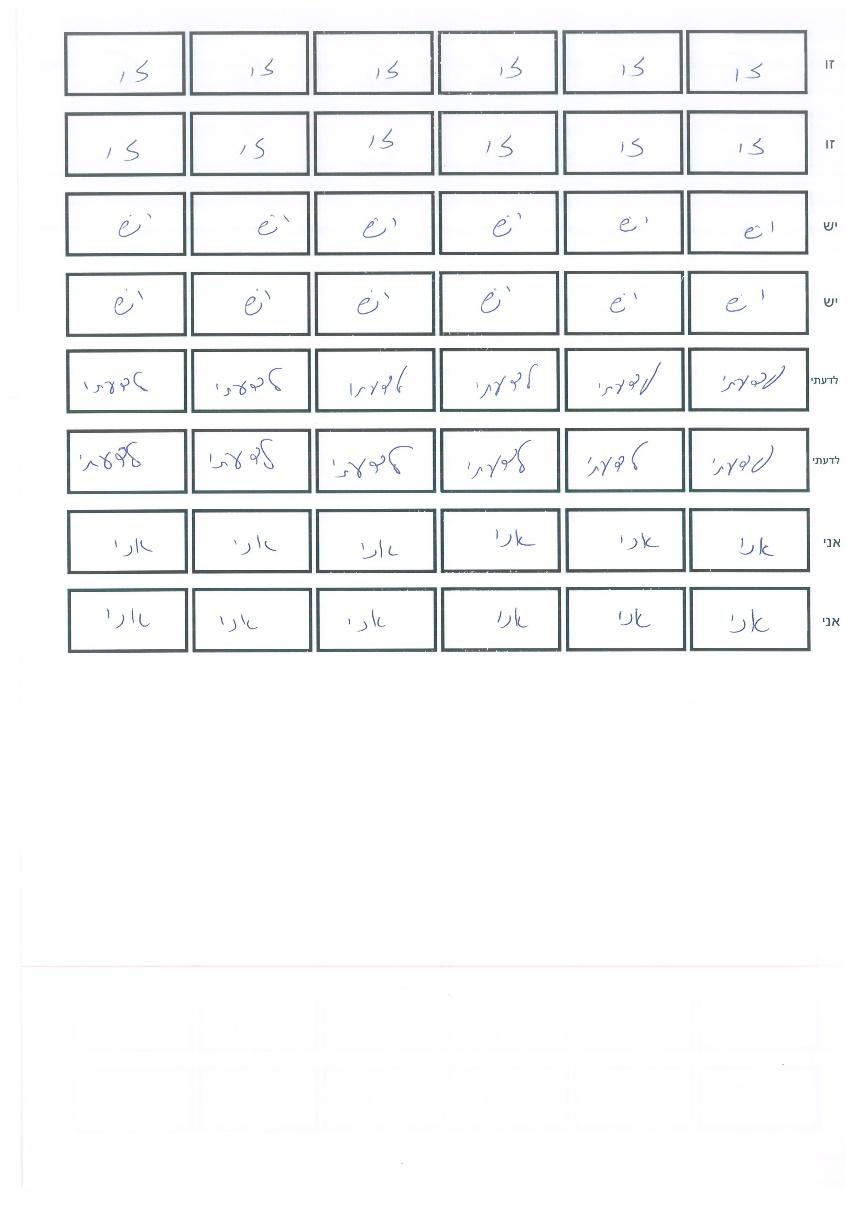
ביצענו כ-  *השוואות (כל הזוגות האפשריים עבור 500 החיבורים) וקיבלנו כי בממוצע יש חיתוך של כ- 45 מילים כפי שניתן לראות בתרשים 13.*

בכדי לאמן מכונה לזהות את המילים שבחרנו, אספנו כ-300 דפים של מילים בכתב יד (3600 מופעים מכל מילה). לאחר שנבצע אימון לרשת נוירונים, כל מילה שהפרדנו בשלב הגילוי נעביר למודל בכדי לבדוק האם היא חלק מהמילים הנפוצות. את המילים שזיהינו נשמור במבנה נתונים לשלב האימות.

דוגמא של הטופס לאיסוף המילים בתרשימים 14,15



תרשים



תרשים

## **טבלת סיכונים -- Risk assessment**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | **PRIORITIZE** | | | **PLAN TO REDUCE IMPACT** |
| **Sort by priority** | **Description of risk** | **Owner** | **Possible Impact** | **Prob (%)** | **Impact (L, M, H, VH)** | **Risk Code (color per table above)** | **Mitigation Plan / Contingency Plan** |
| 7 | אי עמידה בזמני ההגשות של פרויקט הגמר | שחר ודניאל | אי הגשה של המשימות, הורדה בציון. | 15% | VH |  | בניית לוח זמנים עבור משימות הפרויקט ומעקב אחר סיום המשימות בזמן. |
| 8 | המרכז הארצי לא יעביר בזמן הדרוש את הנתונים הדרושים להתקדמות בפרויקט | שחר ודנאל | עיכוב בהתקדמות הפרויקט | 15% | H |  | להקדים את בקשות הנתונים ככל שניתן. |
| 4 | שילוב טכנולוגיות חדשות בפרויקט הגמר שחברי הפרויקט לא עבדו איתם בעבר | שחר ודניאל | עיכוב בהתקדמות הפרויקט | 40% | H |  | הקצאה של חלק מהזמן ללמוד את הטכנולוגיות החדשות. |
| 3 | חתונה של דניאל בפברואר | דניאל | פחות זמן עבודה על הפרויקט בזמן הארגונים לחתונה | 60% | M |  | חלוקת משימות ועבודה יותר חזקה בתקופה שלפני. |
| 1 | תקופת מבחנים + פרויקטים אחרים | שחר ודניאל | פחות זמן עבודה על הפרויקט והתמקדות במבחנים/פרויקטים | 60% | H |  | חלוקת משימות ועבודה יותר אינטנסיבית בתקופה שלפני. |
| 6 | התחלת עבודה חדשה | שחר | קושי בחלוקת הזמנים בין הפרויקט, העבודה והקורסים בסמסטר. | 30% | M |  | קביעת לוח זמנים מסודר, בנוסף יש לשחר קורס אחד פחות בסמסטר ב' ויוכל להקדיש יותר זמן לפרויקט |
| 5 | כתב יד לא מובן של הנבחן ,קשקושים על מחברת הבחינה אשר לא יאפשרו לבצע את הקריטריונים ההשוואה. | שחר ודניאל | קושי בניתוח החיבור וקביעה חדש משמעית לגבי אימות הנבחן | 50% | M |  | אלגוריתם 'קוף' כפי שתואר יוכל אולי לטפל בבעיה זו. |
| 2 | קושי בקביעת קריטריונים להשוואה בין החיבורים | שחר ודניאל | עיכוב בפרויקט, חוסר הצלחה באימות הנבחן | 50% | VH |  | נקיטה בכמה דרכים לפתרון ששילובם יניב את התוצאה הטובה ביותר |

**טבלת דרישות (User Requirements Document)**

מכיוון שאנו מבצעים פרויקט מחקרי, כרגע עוד לא הוחלט על אופן הגשת המוצר (במידה ותהיה הצלחה) ואנו מתמקדים בפתרון הבעיה.

עם זאת הדרישות היחידות כרגע מהלקוח הן:

* המערכת תקבל החלטה בדיוק רב ככל שניתן אם המבחנים נכתבו ע"י אותו אדם או לא.
* תכנון הפרויקט בצורה גנרית ככל שניתן כך שיהיה ניתן בהמשך לבצע התאמות לשפות נוספות כגון ערבית.

**תכנון הפרויקט**

ניתן לגשת ליומן בקישור שבעמוד הראשון

**ביבליוגרפיה**

זרגרי מורן, איתי חפץ (2019). Handwriting detection recognition.

<https://github.com/moranzargari/Handwriting-detection-recognition>

**Building Autoencoders in Keras**

By: Francois Chollet

Publication Date: May 14, 2016

**Hybrid Feature Learning for Handwriting Verification**

By: Mohammad Abuzar Shaikh , Mihir Chauhan , Jun Chu and Sargur Srihari

Publication Date: November 19, 2018

**National institute for testing & evaluation**

<https://www.nite.org.il/>

**Matching Handwritten Document Images**

By: Praveen Krishnan, C.V Jawahar

Publication Date: May 19, 2016

**Writer verification based on a single handwriting word samples**

By: Ameur Bensefia, Thierry Paquet

Publication Date: 2016

**How-To: Python Compare Two Images**

By: Adrian Rosebrock

Publication Date: September 15, 2014

<https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/>