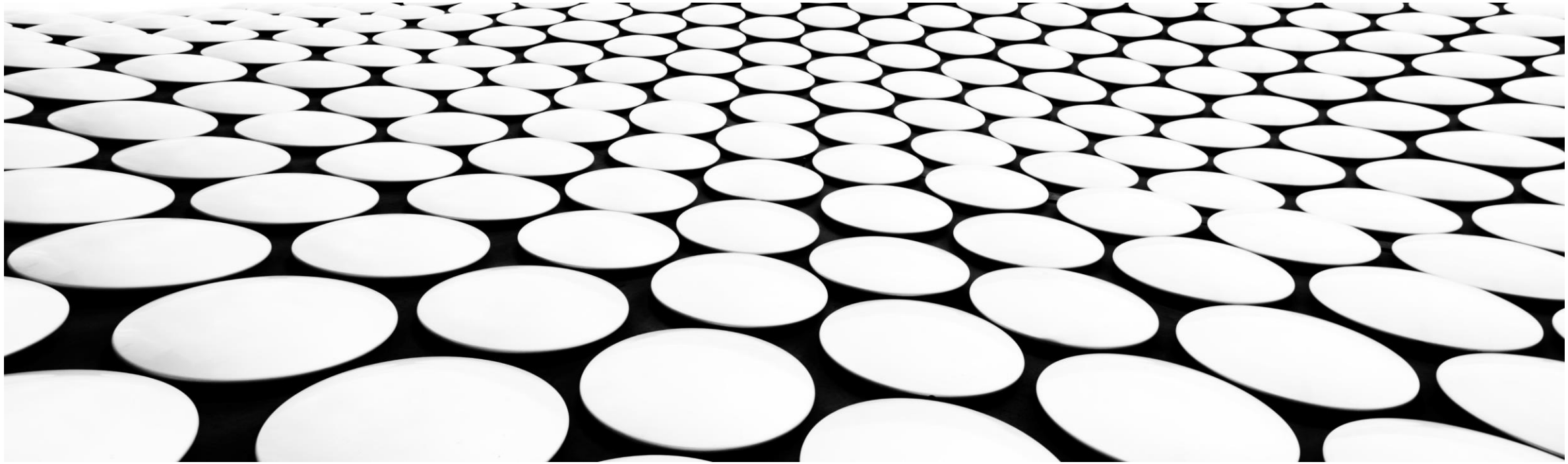
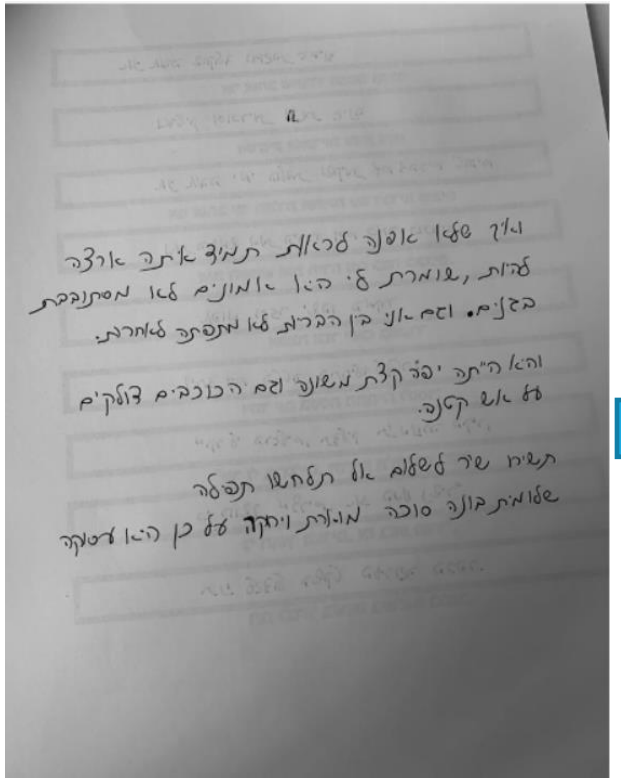
מתמונה לטקסט - חילוץ כתב יד בעברית
SHECODES

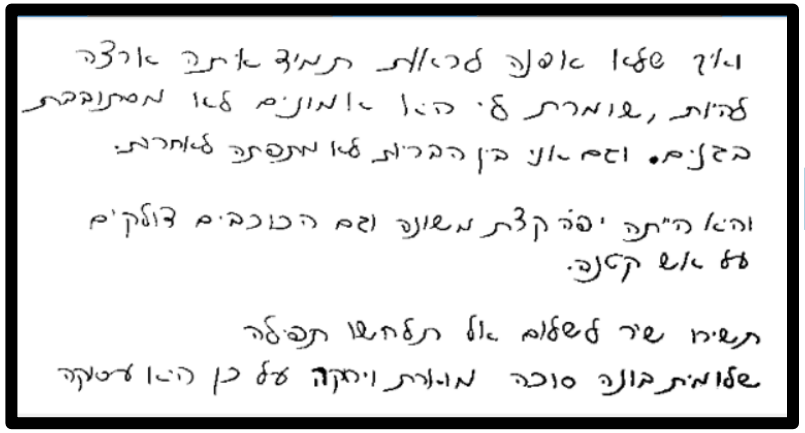
עדי רוזנטל, אוקטובר 2020

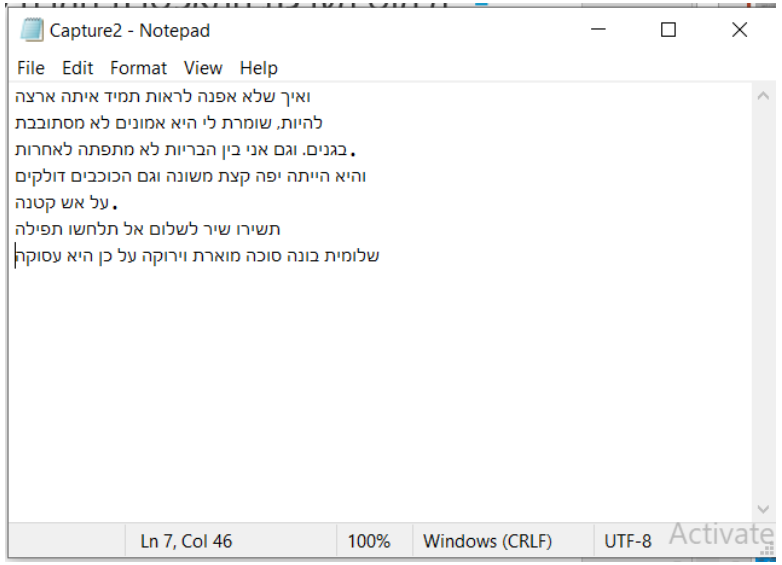


מטרת הפרוייקט

מימוש מערכת המאפשרת המרה של תמונה המכילה טקסט בעברית (בכתב יד / דפוס) לכדי כתב מחשב אותו ניתן לערוך ולעצב בצורה נוחה.








OpenCV


Tesseract OCR


OCR
TXT
Image
Text


Tk&


git


GitHub


docker


Linux


difflib
Python 3 Standard Library

שלושה תהליכים עיקריים בפרוייקט:

1. איסוף וחילוץ של דאטה מתוייג בכתב יד בעברית
2. אימון רשת נוירונים tesseract המבוסס OCR
3. יצירת ממשק נוח למשתמש לחילוץ טקסט מתוך תמונה ואיסוף דאטה עבור אימון הרשת

ארכיטקטורה

Main.py

הרכיב בו נמצא הGUI.
מארגן את המידע
שמתקבל מהמשתמש.

controller.py

מרכז את המידע שנקלט
ומכוון איזה תהליך ירוץ

ModelTesseract.py

רכיב האחראי על מודל החילוץ–
מבצע חילוץ, חישוב accuracy
למודל חדש ומתן ציון לתוצאת
החילוץ ע"י השוואת מחרוזות.

config.py

קובץ תצורה – מכיל
פרמטרים להתאמה
מקומית של תוכנה.

HandwrittenDoc.py

מכיל את כל הפונקציות
הנחוצות ומילון התיוגים
עבור חילוץ כתב יד מתוך
המסמכים הסרוקים.

DataManager.py

רכיב המנהל את המאגר –
קורא / מכניס / מוחק /
עורך את המאגר.

ImageProcessing.py

פונקציות לעיבוד תמונה +
מבנה נתונים שמכיל את
התמונות והמאפיינים הנחוצים
בתהליכי הפרויקט.

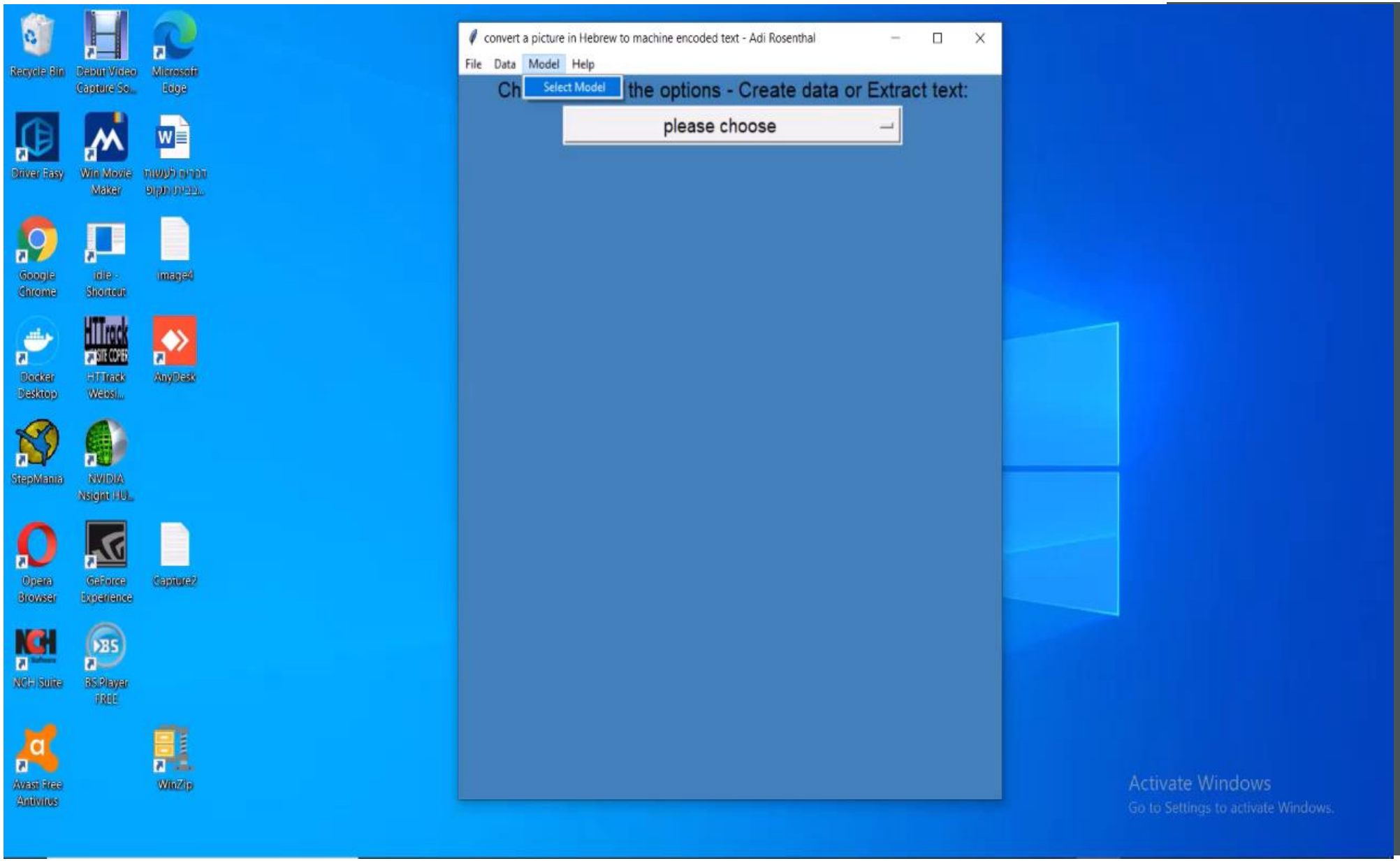
Make_fonts_data.py

מכיל פונקציות לייצור דאטה
מתויג בצורה מאסיבית דרך
פונטים בעברית.

מבנה נתונים (DB)

תהליך טיוב המכונה לחילוץ טקסט דורש אימון של הרשת נוירונים עם כמות גדולה של דאטה.

- תצורת הדאטה: קובץ תמונה (TIF) + קובץ טקסט (.txt)
- הדאטה צריך להיות מאורגן כך שבכל תמונה מופיע שורה אחת של כתב יד + קובץ טקסט בעל שם זהה המכיל את טקסט שכתוב בתמונה
- הדאטה צריך להיות מדוייק ובאיכות גבוהה
- כדאי לבצע גיוון בכתבי היד (כדי למנוע overfitting)



וידיאו

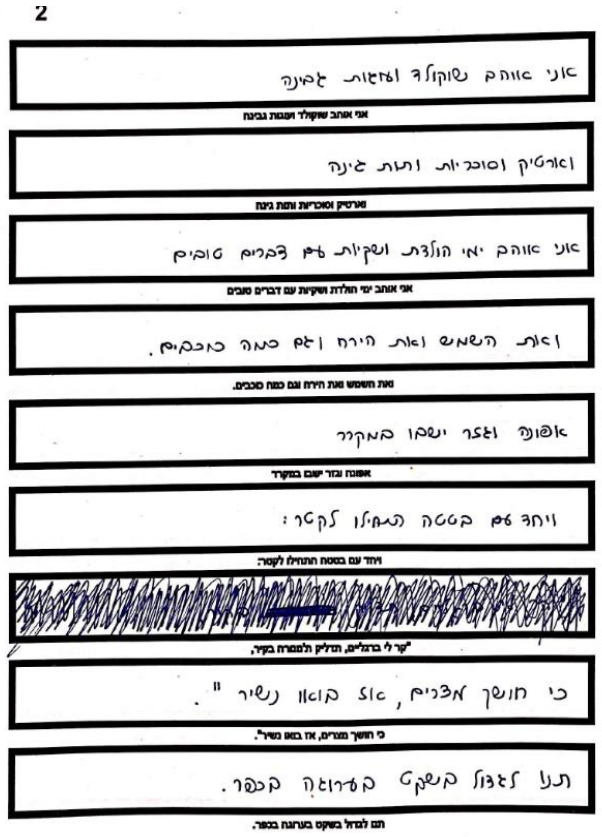
בסיס הנתונים

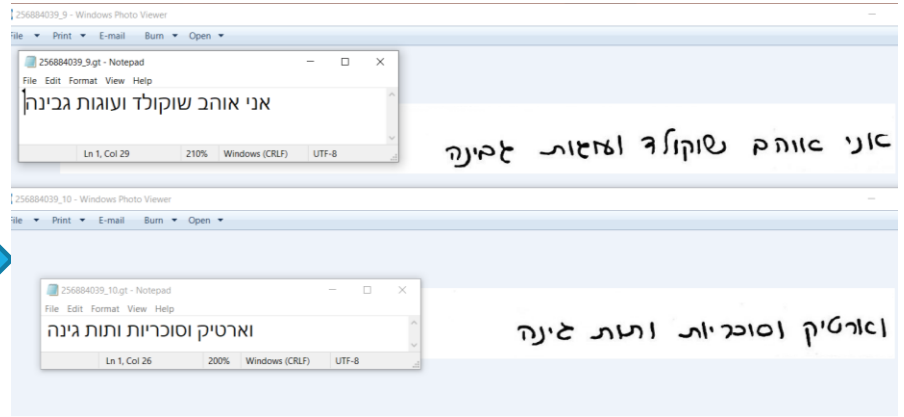
יצירה והבניה של מידע

- **שלוש שיטות עיקריות בהן השתמשתי ליצירת דאטה בפרויקט:**

1. סריקת וחילוץ של שורות טקסט מתוך דפים מוכנים ("Template") אליהם היו צריכים המשתמשים להעתיק שורות של שירים בכתב ידם לפי ההנחיות בדף.
2. סריקה של תמונה עם טקסט ותיוג התמונה בתוך ממשק המערכת.
3. איסוף פונטים בכתב יד בעברית וחילוץ שורות טקסט ותיוגם.

סריקת וחילוץ של שורות טקסט מתוך דפים מוכנים



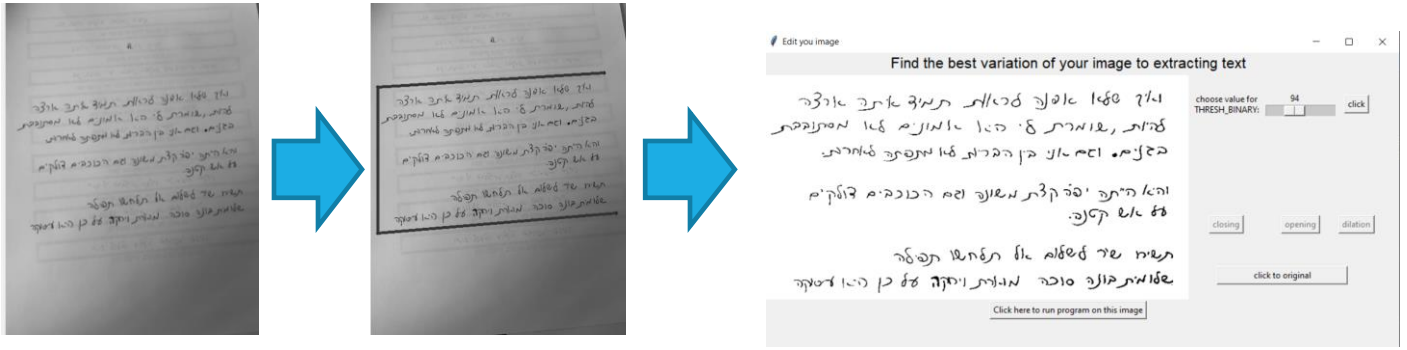


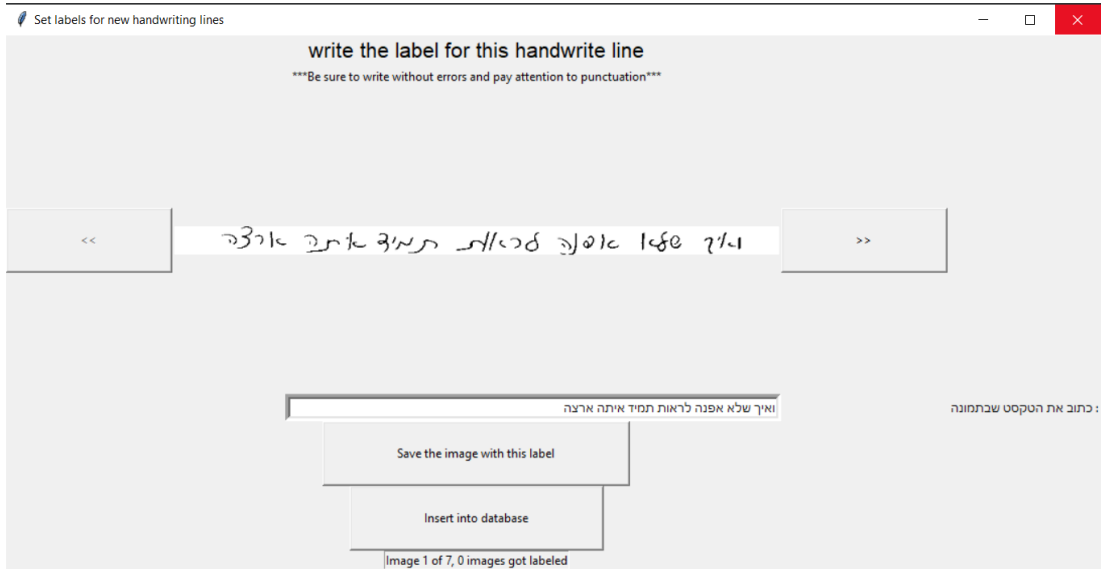
תהליך :

- קבלת קובץ PDG / קובץ תמונה
  - במידה ומתקבל PDF מתבצעת המרה לPNG.
- עיבוד קבצי התמונה ע"י המשתמש – יישור וניקוי רעשים.
- חילוץ מלבנים מהתמונה
- ניקוי "רעשים" בתיחומי המלבנים לפי גודל
- ניקוי של מלבנים מושחרים
- חיתוך של התמונה לפי תיחום המלבן
- תיוג התמונה לפי מילון מוגדר :

(page, line) -> label

תיוג תמונות על ידי המשתמש

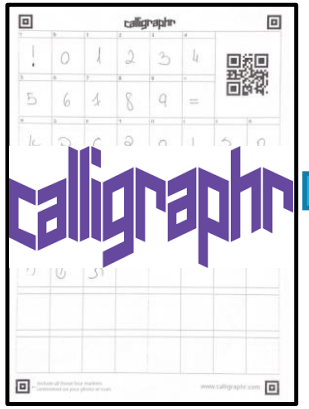


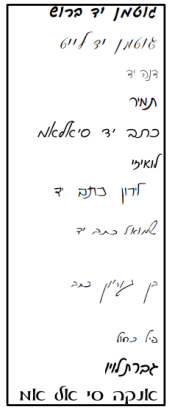


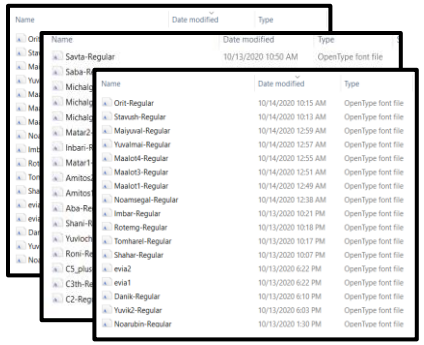
תהליך :

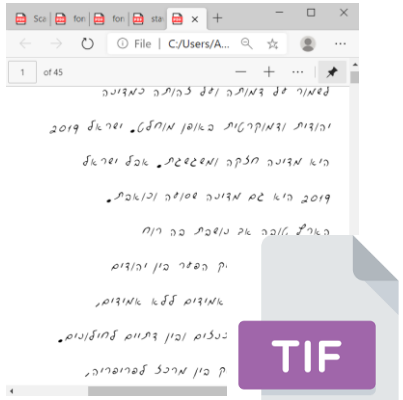
- קבלה כקלט תמונה עם טקסט בכתב יד
- עיבוד קבצי התמונה ע"י המשתמש – יישור וניקוי רעשים.
- הפרדה של הטקסט לשורות על ידי ערך מינימלי של צבע בכל שורה
- הצגה של השורות בממשק תיוג עבור המשתמש

איסוף דאטה בעזרת פונטים







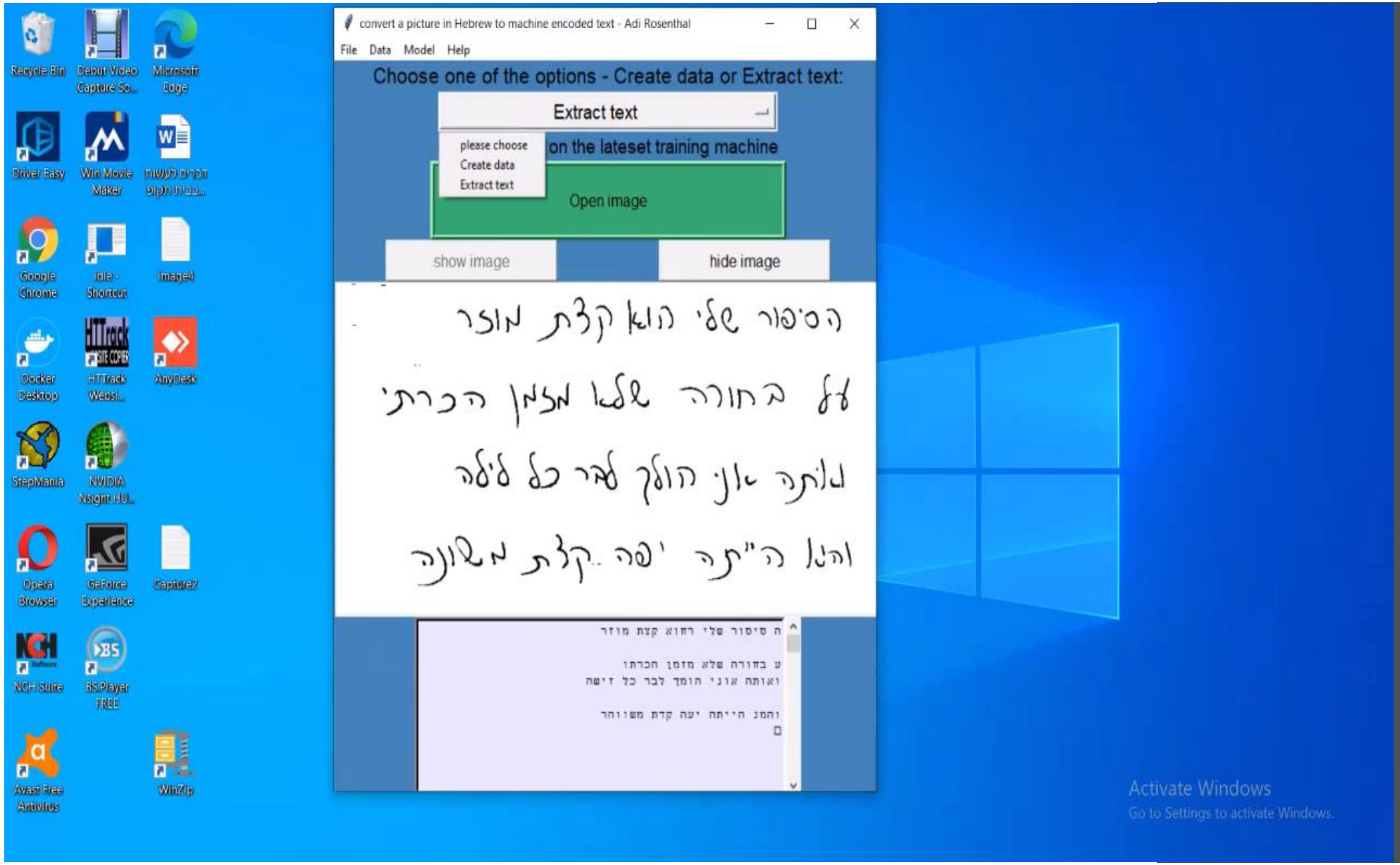



```
def convert_wordDocx_to_x_words_in_line_txtFile(x, doc, outputPathForTextFile):
    j=0
    f = open(outputPathForTextFile, "w+", encoding="utf-8")
    for para
        list
        list
        numw
        whil
def convert_pdf_to_tif_images(folder):
    files = os.listdir(folder)
    for
def extract_ines_of_text_from_images_and_match_labels(fo
    font_name = os.path.basename(folder)[:-8] #remove "d
    font_origin = font_name
    if len
def GetLineBounds(img):
        for
        wh
            lineBounds = []
            minValImage = np.amin(img, axis=1)
            smallThanTHRESHOLDTIGHT = minValImage < THRESHOLD
            row = 1
    names_
    files
            startL = 0
            endL = 0
            imgCopy = img.copy()
            width = img.shape[1]
```

Name	Date modified	Type
fonts_text_679	10/14/2020 10:36 AM	TIF File
fonts_text_680.gt	10/14/2020 10:36 AM	Text Document
fonts_text_680	10/14/2020 10:36 AM	TIF File
fonts_text_681.gt	10/14/2020 10:36 AM	Text Document
fonts_text_681	10/14/2020 10:36 AM	TIF File
fonts_text_682.gt	10/14/2020 10:36 AM	Text Document
fonts_text_682	10/14/2020 10:36 AM	TIF File
fonts_text_683.gt	10/14/2020 10:36 AM	Text Document
fonts_text_683	10/14/2020 10:36 AM	TIF File
fonts_text_684.gt	10/14/2020 10:36 AM	Text Document
fonts_text_684	10/14/2020 10:36 AM	TIF File
fonts_text_685.gt	10/14/2020 10:36 AM	Text Document
fonts_text_685	10/14/2020 10:36 AM	TIF File
fonts_text_686.gt	10/14/2020 10:36 AM	Text Document
fonts_text_686	10/14/2020 10:36 AM	TIF File
fonts_text_687.gt	10/14/2020 10:36 AM	Text Document
fonts_text_687	10/14/2020 10:36 AM	TIF File

תהליך :

- ריכוז פונטים של כתב יד:
  - חיפוש פונטים שקיימים באינטרנט
  - איסוף כ-30 כתבי יד מאנשים שונים ויצירה של פונטים ע"י אתר caligraphr
- ריכוז טקסט ארוך בעברית
- הפרדה של הטקסט לשורות של כ-6 מילים בשורה
- השמה של הפונטים במסמך וורד, המרה לPDF ובסוף לקובץ TIF + יצירת קובץ טקסט מותאם לתמונות
- חילוץ שורות טקסט מתוך התמונות
- תיוג השורות לפי מספר שורה במסמך טקסט



וידיאו

יצירת דאטה לאימון

אימון הרשת - TESSERACT

שלושה שלבים עיקריים במהלך הפרוייקט עבור תהליך אימון הרשת:

1. **מחקר כלים ואפשרויות לאימון הרשת + יצירת סביבה מתאימה לאימון**
2. **תהליך האימון**
3. **בדיקה של הרשת המאומנת**

תהליך האימון

- ארגון של הדאטה והעברתו לסביבה המתאימה
  - DOCKER
  - UBUNTO 20
  - Virtual machine
- שימוש בקוד פתוח Tesstrain שעוזר להכין את הדאטה עבור אימון
  - .txt + tif -> .box -> .lstm
- Tesseract training tools
- העברה של הרשת המאומנת (hebX.traineddata) לתקית התקנה של Tesseract

docker_stepps.txt
~/Documents/shecodes/train/train-tesseract

```
OUT DOCKER:
cd ~/Documents/shecodes/train/train-tesseract
sudo docker-compose -f docker.dev.yml up
sudo docker exec -ti train-ocr-heb bash

IN DOCKER:
cd /app/src/tesstrain/data
rm -r heb?-ground-truth
mkdir hebX-ground-truth
cp -a /app/data/hebX-ground-truth/* /app/src/tesstrain/data/hebX-ground-truth/.
cd /app/src/tesstrain/
make training MODEL_NAME=heb? START_MODEL=heb LANG_TYPE=RTL PSM=7 TESSDATA=/usr/-
local/share/tessdata
cp /app/src/tesstrain/data/heb?.traineddata /usr/local/share/tessdata/

OUT DOCKER:
sudo docker container ls
sudo docker cp CONTAINERID:/app/src/tesstrain/
share/tesseract-ocr/4.00/tessdata
```

root@dd8575c1527d: /app/src/tesstrain

```
root@dd8575c1527d:/app/src/tesstrain# make training MODEL_NAME=heb9 START_MODEL=
heb7 LANG_TYPE=RTL PSM=7 TESSDATA=/usr/local/share/tessdata
find data/heb9-ground-truth -name '*.gt.txt' | xargs cat | sort | uniq > "data/h
eb9/all-gt"
combine_tessdata -u /usr/local/share/tessdata/heb7.traineddata  data/heb7/heb9
Extracting tessdata components from /usr/local/share/tessdata/heb7.traineddata
Wrote data/heb7/heb9.lstm
Wrote data/heb7/heb9.lstm-unicharset
Wrote data/heb7/heb9.lstm-recoder
Wrote data/heb7/heb9.version
Version string:4.1.0
17:lstm:size=3039635, offset=192
21:lstm-unicharset:size=4653, offset=3039827
22:lstm-recoder:size=724, offset=3044480
23:version:size=5, offset=3045204
unicharset_extractor --output_unicharset "data/heb9/my.unicharset" --norm_mode 3
 "data/heb9/all-gt"
Bad box coordinates in boxfile string!
```

בדיקת הרשת

- הרצה של הרשת החדשה על הדאטה לבדיקות (בתקיית validation)
  - הדאטה מתויג אך הרשתלא אומנה עליו
- השוואה של תוצאת הרשת לתיוג "אמת"
  - שימוש בפונקציית SequenceMatcher שמבוסס על אלגוריתמים להשוואת מחרוזות LCS – longest contiguous matching
  - האלגוריתם לא מתייחס למשקלים של אותיות – כמו למשל אותיות דומות וכו'
  - יצירת דוח
  - השוואה בין תוצאות הרשת המאומנת על כל שורה בvalidation לתוצאת הרשת על אותם המשפטים לפני האימון והערכה של אחוז שיפור

תוצאות של אימוני הרשת

his PC > Documents > shecodes_finalProject > models-trained

Name	Date modified	Type
test_models	10/14/2020 8:14 AM	File folder
heb.traineddata	7/31/2020 5:04 PM	TRAINEDDATA File
heb3.traineddata	10/8/2020 9:23 AM	TRAINEDDATA File
heb4.traineddata	10/11/2020 4:22 PM	TRAINEDDATA File
heb-fonts3.traineddata	10/7/2020 9:28 AM	TRAINEDDATA File
heb-fonts4.traineddata	10/7/2020 2:42 PM	TRAINEDDATA File
instructtion train on docker	10/7/2020 9:37 AM	Text Document

דו"ח בדיקות רשת

heb7_13_10_17_52 - Notepad

File Edit Format View Help

```
DATE : 13_10 17_52
Model : heb7

Validation folder: C:\Users\Adi Rosental\Documents
\shecodes_finalProject\models-trained\test_models, length
data: 32.0

Algorithem of compare strings : SQ

text: ואיך שלא אפנה לראות תמיד איתה ארצה
Output: ואיך שלא אפנה לראות תמיך אייתה ארצה
Befor: ה_20 ש_ר ויכ /ᗰ])0)© |<«6 ך
Percent coincidence after train: 92.96%
diff between befor train and after: 71.75%


text: להיות, שומרת לי היא אמונים לא מסתובבת
Output: להיות, שומרת זי היא אמונים לא מסתובבת
Befor: תלתנבלר |6& ©עאקי/[< |ש% &/ כר/#122, ות; _
Percent coincidence after train: 94.74%
diff between befor train and after: 70.05%


text: בגנים. וגם אני בין הבריות לא מתפתה לאחרות.
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

```
from difflib import SequenceMatcher as SQ
```

```
def Check_model_tesseract(self, folder_validation, folder_output_txtfile, psm=7, compare_methods = "SQ"):
```

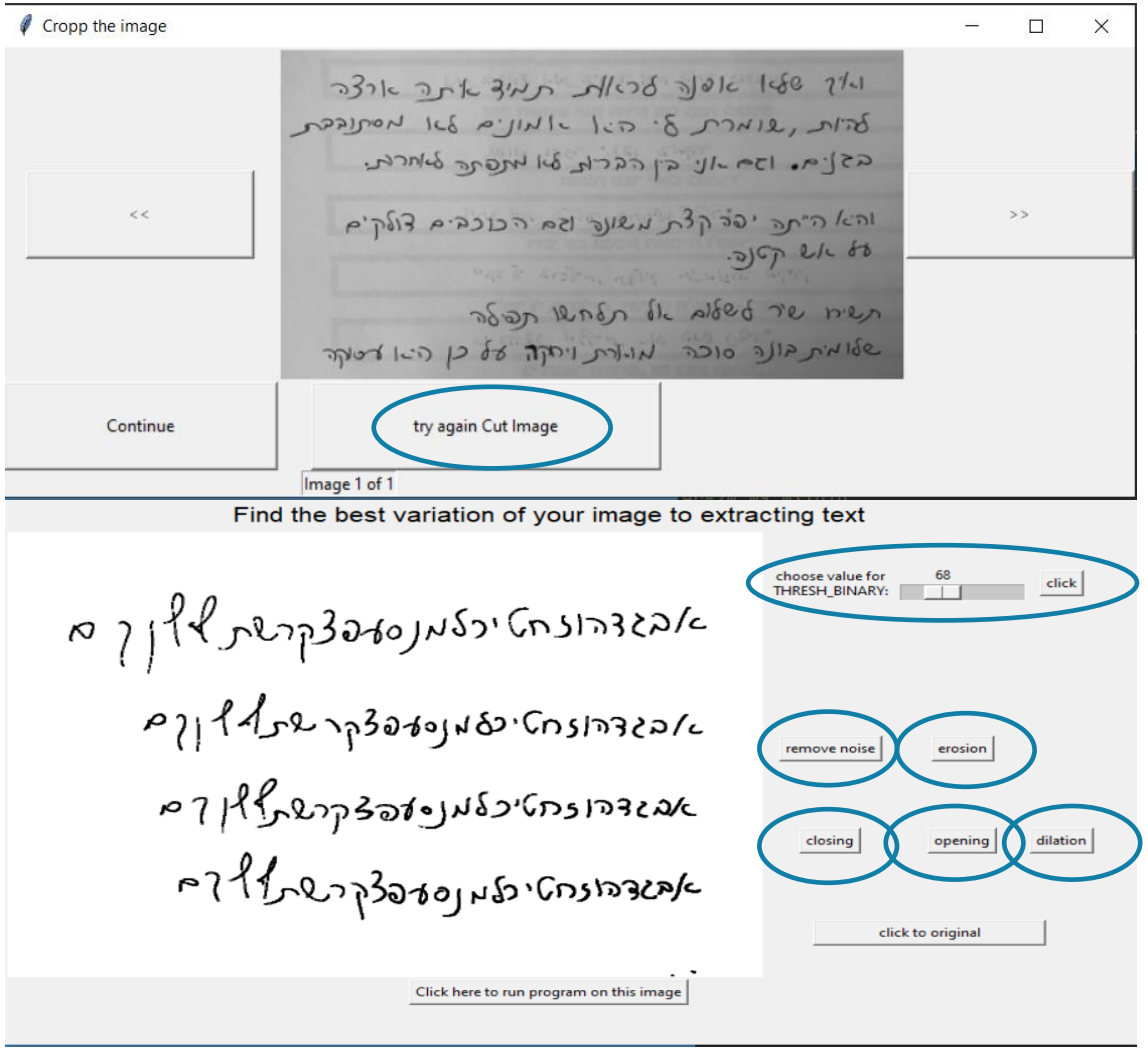
וידיאו

אימון מודל

חילוץ טקסט מתוך תמונה

- עיבוד תמונה לפני חילוץ
- Pytesseract
- השוואת תוצאות התוצאה הרשת ל"אמת"

עיבוד תמונה לפני חילוץ



כדי שהחילוץ יהיה אופטימלי כדאי לבצע עיבוד תמונה לפני שימוש ברשת נוירונים

בממשק ישנן מספר אפשרויות להטיב עם תוצאת הרשת :

1) מרכוז של הטקסט על ידי סימון האזור בו מצוי הטקסט אותו אנחנו רוצים לחלץ.

2) ניקוי רעשים בעזרת פונקציות לעיבוד תמונה (OpenCV):

Binary – thresholding

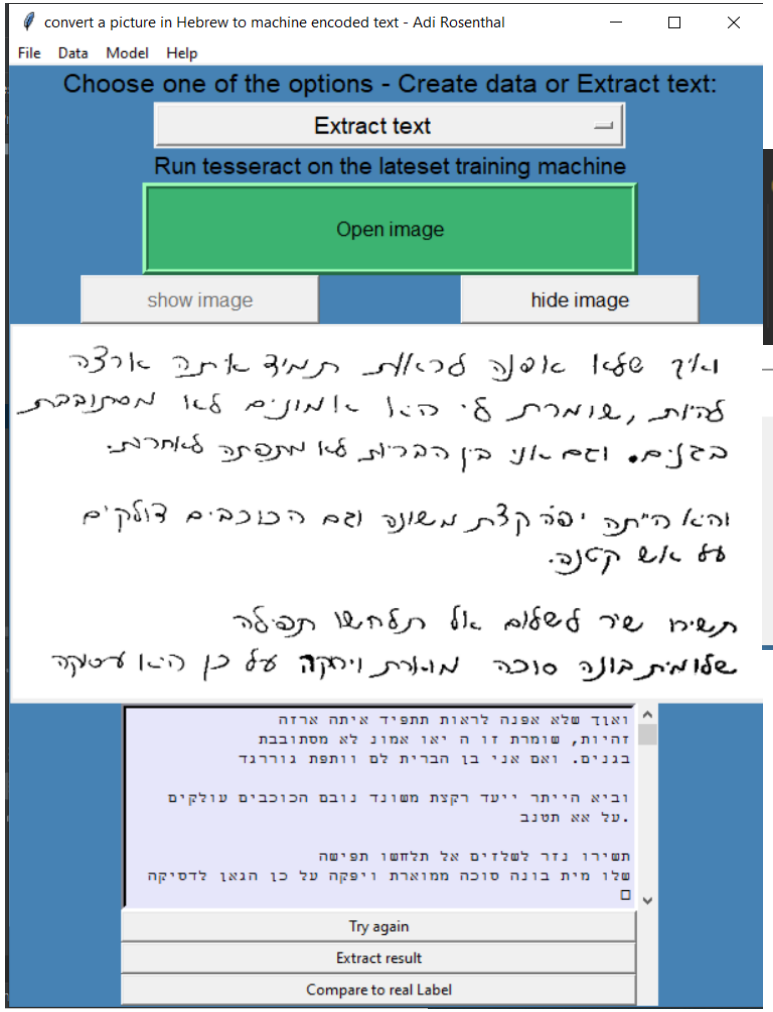
medianBlur

Dilate

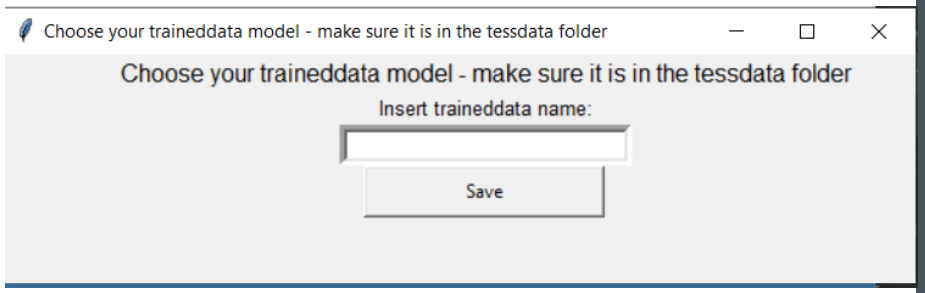
Erosion

dilate

חילוץ טקסט



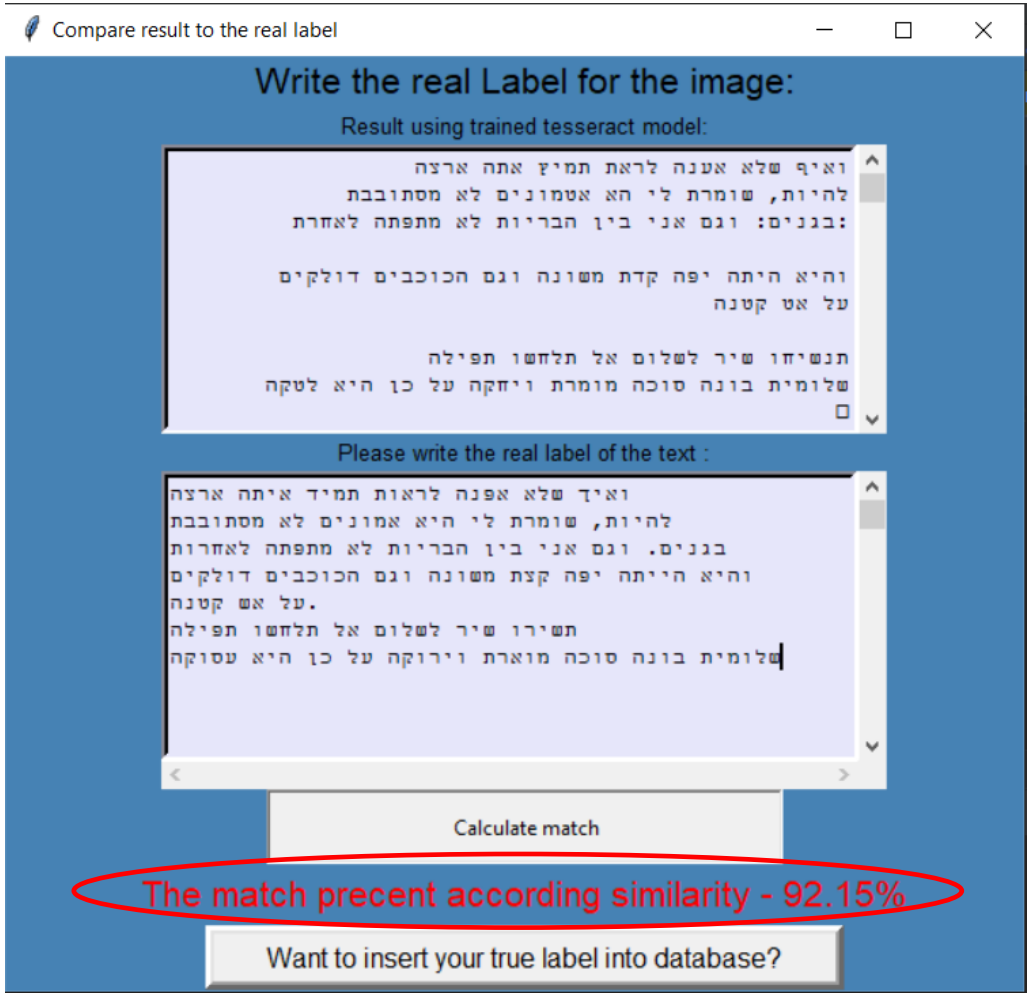

```python
def ExportTextTesseract(self, image):
    print(self.lang)
    str = pytesseract.image_to_string(image, lang=self.lang)
    return str
```



תהליך חילוץ הטקסט:

- אופציה לבחירת רשת נוירונים בעזרתה נרצה לחלץ
  - הרשת הנבחרת צריכה להיות שמורה בתקיית tessdata
  - באופן דיפולטי הרשת מוגדרת להיות הרשת הכי איכותית
- החילוץ עצמו מתבצע ע"י שימוש בספריה pytesseract שמתממשקת לתוכנה של tesseract – מקבלת תמונה ושם של רשת
- שמירת תוצאות הרשת כקובץ טקסט (Exstract result)

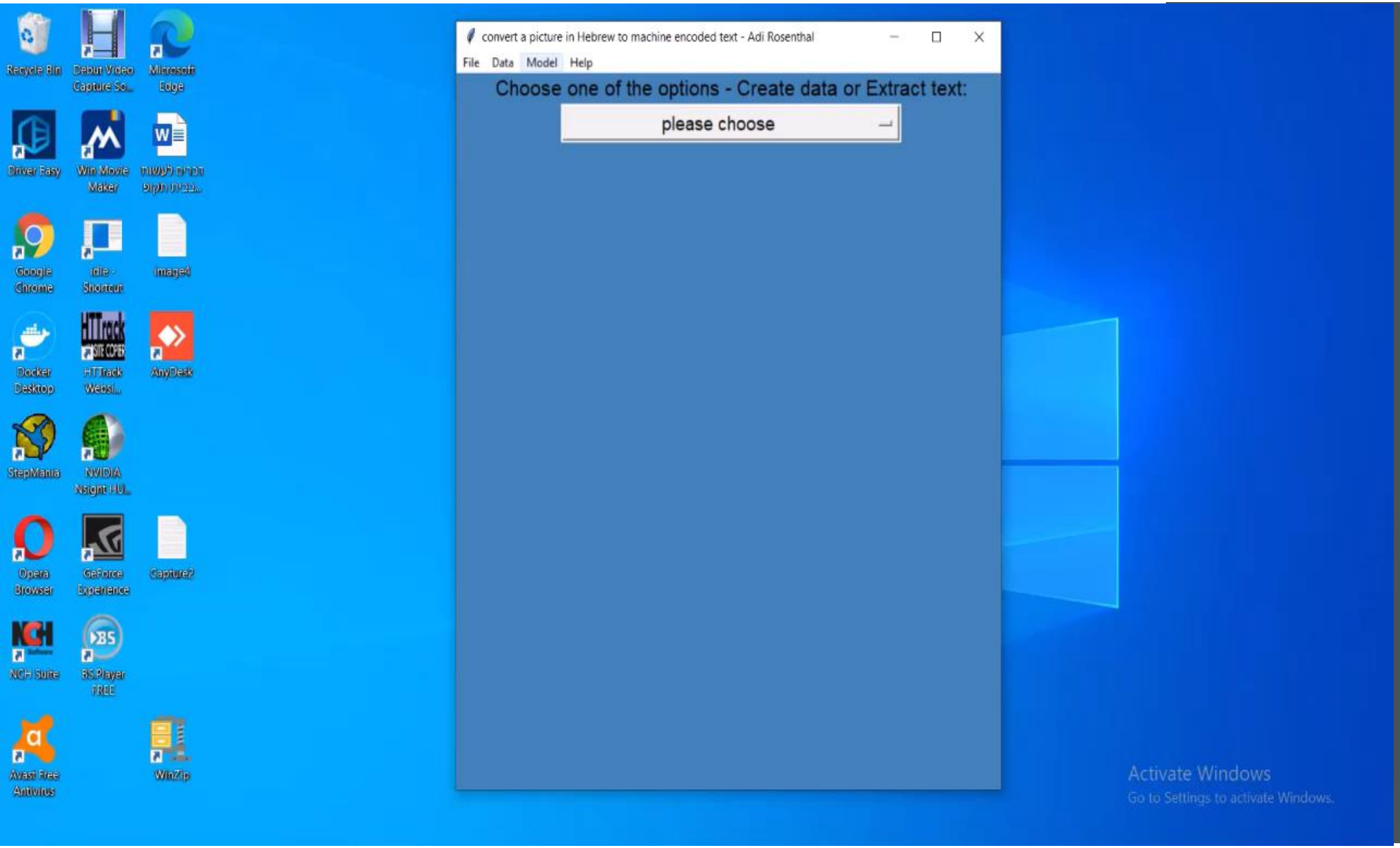
השוואת תוצאות



כדי לבחון את טיב התוצאה שהתקבלה בחילוץ קיימת אופציה עבור המשתמש להכניס את הטקסט הנכון של התמונה – ה"אמת".

ההשואה בין הטקסטים מתקבלת באחוזים על ידי פונקציית "SequenceMatcher"

(לאחר הכנסת התוצאות אמת ניתן גם להכניס את תמונת הקלט עם הטקסט שהמשתמש הכניס בבדיקה לתוך בסיס הנתונים)



וידיאו

חילוץ טקסט

מה למדתי בפרוייקט?

- Creating Virtual machine
- Using Linux
- Work with Dockers
- Git + Github
- Tesseract / OCR
- Design + Architecture
- Using open source
- OpenCV
- Tkinter
- PIL (Image)
- Difflib
- Training tesseract
- Similarity between strings
- Designing and planning a GUI

צעדים להמשך...

- שיפור תוצאות הרשת בשיטות נוספות כמו:
  - NLP
  - שימוש בתיקון על פי מילון
  - סינון של מילים ואותיות- בשילוב עם התניות בשפה העברית
  - https://github.com/NLPH/NLPH_Resources
- בניית אתר אינטרנט / אפליקציה נוחה למשתמש
- הבנה מעמיקה יותר של אלגוריתם tesseract ותהליך האימון כדי להשפיע יותר על האימון
- בדיקה יותר מעמיקה של תוצאות הרשת.

קישורים

- Tesseract-ocr : https://github.com/tesseract-ocr/tesstrain
- https://medium.com/@guiem/how-to-train-tesseract-4-ebe5881ff3b7
- Train-tesseract and docker files for train :https://github.com/guiem/train-tesseract
- Tesstrain : https://github.com/tesseract-ocr/tesstrain
- https://tesseract-ocr.github.io/tessdoc/ImproveQuality
- https://www.makeuseof.com/tag/create-virtual-machine-using-windows-10-hyper-v/
- Learn OpenCV: https://www.youtube.com/watch?v=N81PCpADwKQ&t=6764s
- Learn TKinter: https://www.youtube.com/watch?v=YXPyB4XeYLA
- Learn docker: https://www.youtube.com/watch?v=i7ABlHngi1Q
- SequenceMarcher in python : https://towardsdatascience.com/sequencematcher-in-python-6b1e6f3915fc

שאלות?