

פרויקט גמר 5 יחידות לימוד
התמחות – תכנון ותכנות מערכות Deep Learning
Sign Language's digits identifier



מגיש: אוהד עומרד

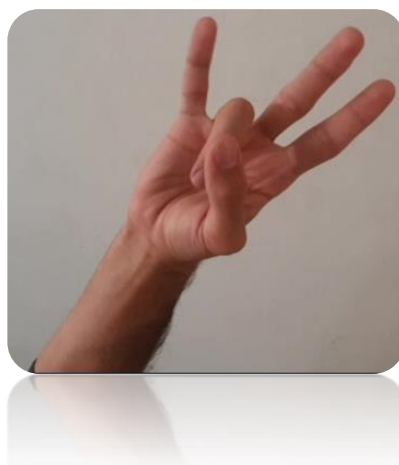
תעודת זהות: 212332191

בית הספר: מקיף י"א ראשונים ראשון לציון

כיתה: י"ב 4

מורה: דינה קראוס

תאריך בחינה: 18.6.2020



תוכן עניינים:

מבוא.....	3
מדריך למשתמש.....	4-15
מדריך למפתח.....	16-37
מסקנות הרצת המודל.....	38-40
רפלקציה / סיכום אישי.....	41
ביבליוגרפיה.....	42
נספחים.....	43-47

מבוא

השנה בבית ספרי במסגרת מסלול פיילוט התבקשנו לבצע את פרויקט הגמר שלנו בנושא "למידה עמוקה" (Deep Learning Computer Vision). תחום זה הינו תחום רק גוני העוסק ביכולת לדמות באופן מלאכותי את הליך הלמידה האנושית. מוח האדם בנוי מרשת ניורונים אשר בהם נקלט המידע ובעזרת התקשורת שבין כל תא עצב לאילו שבקרבתו אנו חושבים ובאופן ספציפי יותר אנו בעליי היכולת ללמוד דברים חדשים ולזכור את מה שלמדנו. על רעיון זה מתבססים כל הפרויקטים בתחום זה ואף הפרויקט שלי.

בפרויקט זה אני בחרתי לכתוב תוכנה אשר לאחר תהליך למידה תוכל לנתח תמונות שונות המייצגות ספרות בשפת הסימנים ולזהות מהי הספרה המיוצגת בתמונה. בחרתי לבצע פרויקט זה כדי ליצור את התשתית הבסיסית שתאפשר תקשורת אנשים דוברי שפת הסימנים לבין אנשים שאינם דוברים שפה זו. מאחר ואין בידי הכלים והאמצעים ליצור תוכנה אשר תוכל לזהות את כל אוצר המילים בשפת הסימנים בחרתי באוצר מילים מצומצם מתוך השפה. תוכנה זו למעשה מתרגמת את סימוני הידיים המבטאים ספרות בשפת הסימנים לספרה מספרית (חשוב להדגיש כי הספרייה בשפת הסימנים החל מן הספרה חמש אינה זהה לספירה שאנו רגילים אליה). בעתיד פיתוח תוכנה זו שכתבתי והרחבת אוצר מילים הניתן לזיהוי תעצים ותחזק את הפוטנציאל הגלום תחת פרויקט זה.

בסקירה של המצב בשוק ישנן תוכנות שונות המנסות לתת מענה לבעיה זו ואכן ישנם תוכנות מבוססות machine learning, בעלות היכולת לתרגם אוצר מילים נכבד של מילים בשפת הסימנים. עם זאת, אף תוכנות אלו אינן מפותחות מספיק במטרה ליצור תקשורת זורמת בין שני אנשים כאשר אחד מהם דובר את שפת הסימנים.

במסגרת ביצוע פרויקט זה נתקלתי במספר אתגרים מרכזיים. ראשית כל, על מנת לבצע את הליך הלמידה שציינתי יש צורך במאגר של כאלפי תמונות המתארות ספרות בשפת הסימנים. אך בכל חיפושיי במרשתת לא הצלחתי למצוא data set מתאים. למרות קושי זה, בחרתי שלא לוותר על נושא זה מכיוון שזיהיתי בו את היכולת לתרום ולהוות מעבר לפרויקט שממנו אלמד גם פרויקט שימושי. פתרתי את בעיה זו באמצעות צילום סרטוני ווידאו של כפות ידיים של חבריי, אשר לפי הנחיותי ביטאו את כל הספרות בשפת הסימנים. מסרטונים אלו באמצעות תוכנה בשם Debut Video Capture Software המרתי את סרטוני הווידאו למאגר תמונות מאוזן אשר היוו את ה"דוגמאות" מהן תלמד התוכנה בהמשך.

בנוסף לאתגר זה נתקלתי באתגרים שונים במהלך ביצוע הפרויקט, אך האתגר המשמעותי ביותר היה למידת הנושא והרעיונות מאחורי DL ובמקביל לבצע פרויקט המתבסס על נושא זה. כפי שציינתי בתחילת הפרק נושא זה של DL היה חדש עבורנו, ולמעשה התבקשנו לכתוב תוכנה המבצעת דבר שאינינו ידענו כיצד לבצעו ולמעשה למדנו את הנושא תוך כדי ביצוע הפרויקט באמצעות מדריכים וספרים שונים, מורתנו דינה ומקורות נוספים. מאחר שביצעתי את הפרויקט תוך כדי למידה לפרויקט שביצעתי ישנן גרסאות שונות אשר כל אחת עולה על השנייה עד לקבלת הגרסה הסופית. ההבדלים בין הגרסאות הינו המודל עליו התבססתי ואף במאגר התמונות (ארחיב על כך בפרק מסקנות המודל). כפי שציינתי תוכנה זו ייצרה מוח מלאכותי ולמעשה האתגר הגדול היה לכתוב ולהשתמש במודל (המדמה את שכבות הניורונים במוח האנושי) אשר יניב אחוז הצלחה גבוה כלל הניתן. הפרויקט הסופי שלי משתמש במודל המבוסס על 27 שכבות מסוגים שונים אשר קשורים ביניהם.

מדריך למשתמש

לפני שאציג את האופן בו מומש הפרויקט, אציג מדריך אשר ינחה את המשתמש כיצד להשתמש בתוכנית.

הוראות התקנה:

- יש להוריד Python 3.7 או גרסה עדכנית יותר - <https://www.python.org/downloads/>

```
שורת הפקודה
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ilano>python -V
Python 3.7.3
```

אם יש לך Python על המחשב, בדוק מהי גרסתו באמצעות הפקודה: python -V

- יש להתקין במחשב את סביבת העבודה Anaconda או כל IDE אחר. לינק להורדת Anaconda - <https://www.anaconda.com/products/individual>

- יש להוריד מספר ספריות קוד אשר בהן הפרויקט משתמש:

Link	Installation command	library name
https://pypi.org/project/Keras/	pip install keras	keras
https://pypi.org/project/tensorflow/	pip install tensorflow	tensorflow
https://pypi.org/project/matplotlib/	pip install matplotlib	matplotlib
https://pypi.org/project/imutils/	pip install imutils	imutils
https://pypi.org/project/numpy/	pip install numpy	numpy
https://pypi.org/project/opencv-python/	pip install opencv-python	opencv

- יש להוריד מן חשבון ה GitHub שלי את הקבצים הבאים:

- את קבצי ה python אשר עליהם מתבסס המודל
- את התיקייה המכילה את מאגר התמונות – ניתן לשמור שקובץ zip (רק תיקייה ז).
- את התיקייה המכילה את מאגר התמונות באופן ממזין
- את קובץ המודל השמור
- את קובץ התוויות.

***אין לשנות את תוכן הקבצים**

5. יש לארגן את הקבצים במחשב באופן הבא:

- א. קבצי ה python ירוכזו בתיקייה אחת.
- ב. התיקיות המכילות את מאגר התמונות באופן ממוין ובאופן שאינו ממוין שיאוכלסו ב directories שאינם מכילים אותיות עבריות.
- ג. המודל השמור וקובץ התוויות יכולים לשבת בכל directory במחשב.

6. עדכון הקוד בהתאם ל directories החדשים:

בפונקציית ה menu() בקובץ menu.py יש לשנות את ערכי המשתנים המחזיקים את הכתובות הרלוונטיות.
data_set – ה directory של מאגר התמונות הלא ממוין (במידה והינו קובץ zip יש להוסיף סיומת zip בקוד - יוצג בהמשך).
sorted_data_set – ה directory של מאגר התמונות הממוין.
model_path – ה directory של המודל השמור.
labels_path – ה directory של קובץ התוויות של התמונות.

```
98 def menu():
99     flag = True
100     options()
101     """
102     default directories
103     this directoties change according to the user activity
104     """
105     #directories_file.create_directories_file()
106
107     data_set = r"C:\Users\ilano\OneDrive\Desktop\images"
108     sorted_data_path = r"C:\Users\ilano\OneDrive\Desktop\my_dataset"
109     model_path = r"C:\Users\ilano\OneDrive\Desktop\result\Model.model"
110     labels_path = r"C:\Users\ilano\OneDrive\Desktop\result\lb.txt"
```

הרצת התוכנית:

יש להריץ ב Command Line של סביבת העבודה Anaconda את קובץ ה Python: menu.py.

הרצת קובץ זה תריץ את כל התוכנית. לאחר הרצת התוכנית יוצג התפריט הבא:

```
Anaconda Prompt - python "C:\Users\ilano\OneDrive\Desktop\Projects\python\Deep Learning\Final Project\menu.py"
(base) C:\Users\ilano>python "C:\Users\ilano\OneDrive\Desktop\Projects\python\Deep Learning\Final Project\menu.py"
Using TensorFlow backend.
*****
*          USER INTERFACE          *
*                                  *
* Enter 1 --> create sorted data set *
* Enter 2 --> train the model       *
* Enter 3 --> predict an image      *
* Enter space bar --> exit          *
*                                  *
*****
--> Your Choice:
Enter:
```

תפריט זה מציג למשתמש את האפשרויות העומדות לרשותו ואת הקלט שיש להזין עבור כל אפשרות.

כפי שניתן לראות מוצעות ארבע אפשרויות:

הכנס 1 כדי ליצור מאגר תמונות ממזין

הכנס 2 כדי לאמן את המודל

הכנס 3 כדי לחזות את הקטגוריה של תמונה שהמשתמש ישלח כקלט

הכנס רווח לציאה מן התוכנית (יש להקיש רווח ואחריו Enter).

עבור כל קלט אחר התוכנית תמשיך לבקש קלט.

המשתמש יכול לבצע חיזוי ללא אימון או אימון ללא מיון – התוכנה תשתמש בקבצים שבהם שמורים הנתונים – אותם המשתמש התקין במחשבו לפי ההוראות שהצגתי קודם לכן (קובץ אשר ממזין מראש, קובץ שבו נשמר המודל וכן הלאה).

אחד הפיצ'רים של התוכנה היא שהיא יודעת להשתמש בקבצים העדכניים עבור כל הרצה חדשה. לדוגמה: אם המשתמש בחר לאמן את המודל ואחר כך לבצע חיזוי של תמונה התוכנית תשתמש במודל שנשמר לאחר האימון שבוצע. הרצה חוזרת של התוכנית תחזור להשתמש בקובץ הדיפולטיבי (המותקן על המחשב מראש).

כעת נעבור על שלושת האפשרויות השונות.

בכל אפשרות אציג את קלטי המשתמש הנדרשים, את ההדפסות המתארות את התהליך המתבצע ואת הודעות השגיאה השונות אשר יציינו את השגיאה וידרשו קלט תקין בהתאם עד לקבלתו.

הדפסות בצבע ירוק – מנחות את המשתמש בבקשת קלט.

הדפסות בצבע כחול – מספקות אינפורמציה על התהליך המתרחש בכל רגע.

הדפסות בצבע אדום – מסמלות שגיאה של קלטיים אי תקינים.

האופציה הראשונה – מיון מאגר התמונות:

המשתמש הקיש 1

יוצגו לפניו ההדפסות הבאות:

```
*****
*          USER INTERFACE          *
*                                  *
* Enter 1 --> create sorted data set *
* Enter 2 --> train the model       *
* Enter 3 --> predict an image      *
* Enter space bar --> exit          *
*                                  *
*****
--> Your Choice:
Enter: 1
[INFO] Got a zip file...
Choose the path to extract the files (with a new folder name)
Enter:
```

1. הודעת אינפורמציה בכחול האומרת שהתוכנה קיבלה קובץ רגיל. הקובץ המדובר הינו התיקיה בה מאוכלסות התמונות באופן לא ממזין. במחשב שלי:
C:\Users\ilano\OneDrive\Desktop\images

במידה ותיקיה זו הייתה מקובץ zip היינו מקבלים הודעת אינפורמציה שונה והיה מתבצע תהליך של חילוץ הקבצים, אך אותו אציג בהמשך.

2. התוכנית מבקשת מן המשתמש לקבל כקלט את ה directory של תיקיה שתהווה תיקיה ראשית, שבה ישמרו כל התמונות באופן ממזין. ההודעה אף דורשת ששם ה directory יהיה באנגלית בלבד.

נכניס קלטיים שאינם תקינים ובנחנו את הודעות השגיאה השונות.

קלט ראשון: C:\Users\ilano\OneDrive\Desktop\Ohad's Project\MainFolder

(חשוב להדגיש שהתיקיה Ohad's Project אינה קיימת ב directory זה).

וקיבלתי הודעת שגיאה **Error - directory is not valid**

קיבלנו הודעת שגיאה שאומרת שהכתובת ששלחנו כקלט אינה תקינה והתוכנה מבקשת מאתנו קלט חדש.

```
--> Your Choice:
Enter: 1
[INFO] Got ordinary file...
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\Ohad's Project\MainFolder
Error - directory is not valid
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter:
```

קלט שני: C:\Users\ilano\OneDrive\documents\MainFolder

במחשב שלי המילה documents כתובה בעברית "מסמכים"

קיבלנו הודעת שגיאה האומרת שהכתובת מכילה אותיות עבריות והתוכנה מבקשת מאתנו קלט חדש.

*אותיות עבריות נחשבות כשגיאה מאחר שלאחר ניסוי ותהייה גיליתי כי בשלב טעינת תמונה באמצעות opencv ישנה שגיאה כשאר ה directory של התמונה מכיל אותיות עבריות.

```
--> Your Choice:
Enter: 1
[INFO] Got ordinary file...
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\Ohad's Project\MainFolder
Error - directory is not valid
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\שולחן העבודה\MainFolder
Error - path contains hebrew letters
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter:
```

קלט שלישי: C:\Users\ilano\OneDrive\Desktop

נשלח כקלט directory אשר הינו קיים וללא שם התיקיה החדשה שבה יאוכלסו התמונות

```
--> Your Choice:
Enter: 1
[INFO] Got ordinary file...
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\Ohad's Project\MainFolder
Error - directory is not valid
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\שולחן העבודה\MainFolder
Error - path contains hebrew letters
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop
Error - this directory is already exists
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter:
```

קיבלנו שגיאה האומרת ש directory זה הינו קיים, משמע הוא אינו חדש – אינו כולל את שם התיקיה והתוכנה מבקשת מאתנו קלט חדש.

קלט תקין: C:\Users\ilano\OneDrive\Desktop\MainFolder

הקלט תקין ולכן התוכנית מתחילה את תהליך יצירת התיקיות הייעודיות ואת אתחולן בתמונות.

כפי שניתן לראות קיבלנו שתי הודעות אינפורמציה:

```
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\MainFolder
[INFO] Creating folders...
[INFO] Initializing the folders...
```

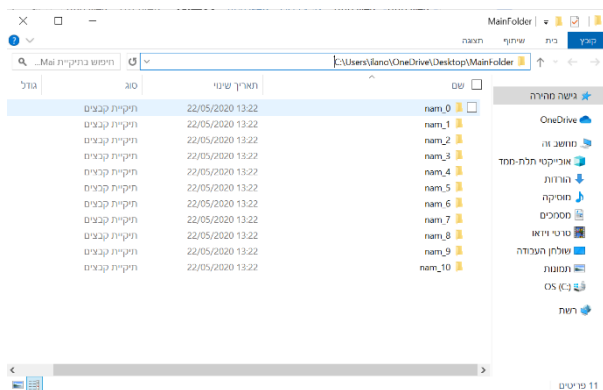
הראשונה שהתוכנה יוצרת את התיקיות ואחריה (לאחר שהתיקיות נוצרו) הודעה המעדכנת שהחל שלב מיון התמונות בתיקיות החדשות.

בסוף ההליך נקבל הודעת אינפורמציה על כך שהתוכנית כעת תשתמש במאגר התמונות הממוין החדש שנוצר.

ולאחר מכן התוכנית שוב מצפה לקלט מספרי 1/2/3/4 או רווח לפי בחירת המשתמש.

```
Please enter the path that you want for your main data set folder:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\MainFolder
[INFO] Creating folders...
[INFO] Initializing the folders...
[INFO] Using new data set
--> Your Choice:
Enter:
```

הפלט – תיקייה בה התמונות ממוינות.



במידה ומאגר התמונות היה שמור בקובץ zip, ואף ה directory של מאגר התמונות בקוד היה של אותו קובץ zip היה מתווסף לתהליך זה שהצגתי תת תהליך נוסף.

```
98 def menu():
99     flag = True
100     options()
101     """
102     default directories
103     this directoties change according to the user activity
104     """
105     #directories_file.create_directories_file()
106
107     data_set = r"C:\Users\ilano\OneDrive\Desktop\imagesimagesZipFolder.zip"
108     sorted_data_path = r"C:\Users\ilano\OneDrive\Desktop\my_dataset"
109     model_path = r"C:\Users\ilano\OneDrive\Desktop\result\Model.model"
110     labels_path = r"C:\Users\ilano\OneDrive\Desktop\result\lb.txt"
```

במחשב שלי ישנו קובץ zip המכיל בתוכו את כל התמונות (ללא תת תיקיות). אשנה בקוד את ה directory הרלוונטי (שורה 107 בתמונה).

לפני שאריץ מחדש את התוכנית ואבחר באופציה מספר 1 שוב, אסיים את התוכנית הקודמת בהקשה על רווח והכנס.

```
--> Your Choice:
Enter:
[INFO] Exiting...

(base) C:\Users\ilano>
```

```

*****
*          USER INTERFACE          *
*          *                       *
*  Enter 1 --> create sorted data set *
*  Enter 2 --> train the model       *
*  Enter 3 --> predict an image     *
*  Enter space bar --> exit         *
*          *                       *
*****
--> Your Choice:
Enter: 1
[INFO] Got a zip file...
Choose the path to extract the files:
Enter:

```

הרצה חדשה:

כפי שניתן לראות קיבלנו הודעת אינפורמציה שמאגר התמונות שמור בקובץ zip. בנוסף לכך, קיבלנו הודעה המבקשת מאתנו directory שאליו יחולצו הקבצים.

directory זה ישמר במהלך התוכנית כה – directory בו שמור מאגר התמונות ולא ה directory שך קובץ ה zip. (כמובן שבכל הרצה חדשה קובץ ה zip יחזור לשמש ככתובת דיפולטיבית) אך אל דאגה יש לנו אפשרות להשתמש במאגר התמונות הממין מראש).

כאן ניתן לראות שהקלט יכול להכיל אותיות עבריות (שכן התמונות שעליהן נרוץ – הנמצאות בקובץ הממין ישבו ב directory שאינו מכיל אותיות עבריות).

אכניס קלט תקין (גם כאן יש בדיקת קלט):

C:\Users\ilano\OneDrive\Desktop\ExtractedImagesHere

```

--> Your Choice:
Enter: 1
[INFO] Got a zip file...
Choose the path to extract the files (with a new folder name)
Enter: C:\Users\ilano\OneDrive\Desktop\ExtractedImagesHere
[INFO] Extracting the images from the zip file...

```

נקבל הודעת אינפורמציה שהתהליך החל

```

--> Your Choice:
Enter: 1
[INFO] Got a zip file...
Choose the path to extract the files (with a new folder name)
Enter: C:\Users\ilano\OneDrive\Desktop\ExtractedImagesHere
[INFO] Extracting the images from the zip file...
[INFO] Finished extracting process...
Please enter the path that you want for your main data set folder (with the folder name)
Path Must be in English!
Enter:

```

ואחריה הודעה שהתהליך החילוץ הסתיים.

לאחר סיום תת תהליך זה שאר השלבים זהים למקודם.

אופציה שנייה – אימון המודל:

המשתמש הקיש 2

לאחר מכן יוצג בפניו מספר בקשות קלט אחת אחת השנייה.

המשתמש יראה את הבקשה להזין את הקלט הבא רק לאחר שהזין קלט תקין.

בדיקות תקינות הקלט זהות לקודמות, מלבד אחת נוספת שאציג כעת.

```
*****
*          USER INTERFACE          *
*          *                        *
* Enter 1 --> create sorted data set *
* Enter 2 --> train the model       *
* Enter 3 --> predict an image     *
* Enter space bar --> exit         *
*          *                        *
*****
--> Your Choice:
Enter: 2
Enter the full path (with the name) to output model:
Enter: _
```

נכניס את ה directory המלא עם שם הקובץ החדש שיווצר שבו נרצה לשמור את המודל לאחר ביצוע האימון. סיומת הקובץ חייבת להיות model

לשם הדוגמה אכניס את ה directory:
C:\Users\ilano\OneDrive\Desktop\Model.model

```
--> Your Choice:
Enter: 2
Enter the full path (with the name) to output model:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Enter the full path (with the name) to output label binarizer:
Enter: _
```

לאחר מכן התוכנה תבקש ממני שוב directory מלא עם שם הקובץ שיווצר שבו נשמור את קובץ התוויות הבינארי (את חשיבות קובץ זה נבין בהמשך, אך לעת עתה אציין שקובץ זה ישמש לביצוע הלמידה ולחיצוי קטגוריה של תמונה בהמשך). סיומת הקובץ חייבת להיות pickle או .txt

אם אכניס את אותו ה directory של הקובץ הקודם קבצים אחת הקבצים ידרוס את השני או שהתוכנית עלולה לקרוס, ולכן כדי למנוע זאת מנעתי ממך המשתמש את אפשרות זו. אם כעת ננסה להכניס כקלט את ה directory הקודם שהזנו נקבל הודעת שגיאה.

```
--> Your Choice:
Enter: 2
Enter the full path (with the name) to output model:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Enter the full path (with the name) to output label binarizer:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Error - file will be override
Enter the full path (with the name) to output label binarizer:
Enter: _
```

אכניס את ה directory:

C:\Users\ilano\OneDrive\Desktop\lb.pickle

```
--> Your Choice:
Enter: 2
Enter the full path (with the name) to output model:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Enter the full path (with the name) to output label binarizer:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Error - file will be override
Enter the full path (with the name) to output label binarizer:
Enter: C:\Users\ilano\OneDrive\Desktop\lb.pickle
Enter the folder directory to output accuracy/loss plot:
Enter: _
```

לאחר קלט זה התוכנית מבקשת directory של תיקייה חדשה שתיווצר שבה ישמרו גרפי תוצאות הלימוד (כקבצי png).

אכניס את ה directory:

C:\Users\ilano\OneDrive\Desktop\ImagesGraphs

לאחר הכנסת כל הקלטים מתחיל לרוץ תהליך האימון ומוצגים לפנינו פלטי אינפורמציה על תתי התהליכים המתרחשים אחד אחרי השני לביצוע האימון.

בתחילה התוכנית טוענת מן ה directory של מאגר התמונות הממוין (העדכני- פר ריצה) את התמונות.

תהליך זה לוקח זמן (כ 10 דקות)

```
--> Your Choice:
Enter: 2
Enter the full path (with the name) to output model:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Enter the full path (with the name) to output label binarizer:
Enter: C:\Users\ilano\OneDrive\Desktop\Model.model
Error - file will be override
Enter the full path (with the name) to output label binarizer:
Enter: C:\Users\ilano\OneDrive\Desktop\lb.pickle
Enter the folder directory to output accuracy/loss plot:
Enter: C:\Users\ilano\OneDrive\Desktop\ImagesGraphs
[INFO] Loading images...
```

לאחר מכן נקבל הודעות אינפורמציה על הגודל במגה בתים של רשימת כל התמונות כמערכים (הסבר במדריך למפתח), הודעה על כך שהמודל התקמפל וישר לאחר מכן הודעה על כך שהחל האימון.

נבחין כי לאחר לימוד כל התמונות שלוש פעמים אחוז ההצלחה עומד על כ 68 אחוזים.

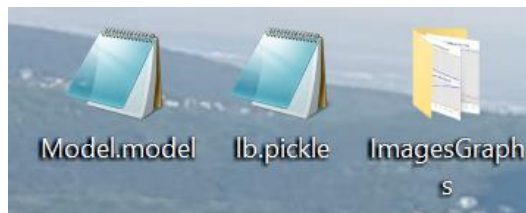
בסיום האימון אנו מקבלים הודעה כי כעת השימוש יתבצע בקובץ המודל החדש ולא בדיפולטיבי.

ולבסוף אנו יכולים לבחור לבצע אפשרות חדשה.

```
[INFO] Training network...
Epoch 1/10
320/320 [=====] - 92s 287ms/step - loss: 2.3608 - accuracy: 0.2954 - val_loss: 2.4482 - val_accuracy: 0.2439
Epoch 2/10
320/320 [=====] - 74s 232ms/step - loss: 1.5396 - accuracy: 0.4786 - val_loss: 2.1602 - val_accuracy: 0.4155
Epoch 3/10
320/320 [=====] - 89s 278ms/step - loss: 1.1815 - accuracy: 0.5850 - val_loss: 0.7049 - val_accuracy: 0.7698
Epoch 4/10
320/320 [=====] - 94s 293ms/step - loss: 0.9600 - accuracy: 0.6610 - val_loss: 1.2805 - val_accuracy: 0.6574
Epoch 5/10
320/320 [=====] - 94s 294ms/step - loss: 0.8096 - accuracy: 0.7081 - val_loss: 0.5933 - val_accuracy: 0.7850
Epoch 6/10
320/320 [=====] - 83s 259ms/step - loss: 0.6765 - accuracy: 0.7625 - val_loss: 0.4481 - val_accuracy: 0.8353
Epoch 7/10
320/320 [=====] - 73s 229ms/step - loss: 0.6204 - accuracy: 0.7763 - val_loss: 0.4321 - val_accuracy: 0.8486
Epoch 8/10
320/320 [=====] - 73s 229ms/step - loss: 0.5524 - accuracy: 0.8015 - val_loss: 0.2854 - val_accuracy: 0.8947
Epoch 9/10
320/320 [=====] - 73s 228ms/step - loss: 0.5098 - accuracy: 0.8196 - val_loss: 0.2701 - val_accuracy: 0.9071
Epoch 10/10
320/320 [=====] - 73s 229ms/step - loss: 0.4690 - accuracy: 0.8330 - val_loss: 0.3748 - val_accuracy: 0.8685
```

נבחין כי ב directories שהזנו אכן נותרו הקבצים בהתאמה.

(צילום מסך של שולחן העבודה שלי).



התיקיה ובה שני קבצי תמונה שבהם גרפי מהלך האימון.

אופציה שלישית – חיזוי תמונה:

המשתמש הקיש 3

התוכנית מבקשת מן המשתמש את ה directory לתמונה שאותה הוא רוצה שהמודל המאומן יזהה.

נבחין כי גם כאן מאחר שמדובר בתמונה שיש לטעון עם הספרייה opencv ה directory בו שמורה התמונה חייב להיות באנגלית בלבד.

```
*****
*          USER INTERFACE          *
*                                  *
*   Enter 1 --> create sorted data set *
*   Enter 2 --> train the model      *
*   Enter 3 --> predict an image     *
*   Enter space bar --> exit         *
*                                  *
*****
--> Your Choice:
Enter: 3
Enter the image path:
Path Must be in English!
Enter:
```

כמובן שה directory שניתן חייב להיות קיים ולכן אם יינתן directory שאינו קיים נקבל הודעת שגיאה.

לשם המחשת שגיאה זו אכניס כקלט את ה directory:

C:\Users\ilano\OneDrive\Desktop\number5.jpg

```
--> Your Choice:
Enter: 3
Enter the image path:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\number5.jpg
Error - no such file or directory
Enter the image path:
Path Must be in English!
Enter:
```

והנה קיבלנו הודעת שגיאה שמציינת בפנינו שאין קובץ או directory כמו זה שהזנו כקלט ואכן אין קובץ כזה ב directory הנ"ל.

נבחר את התמונה אותה אנו רוצים לחזות מתוך מאגר התמונות ונשלח את ה directory שלה כקלט.

אכניס את ה directory :C:\Users\ilano\OneDrive\Desktop\num5_H2_R.jpg

בכתובת זו נמצאת התמונה הבאה:

תמונה זו כפי שניתן לראות מסמלת בשפת הסימנים את הספרה 5.



נקבל מיד הודעה על כך שהמודל השמור נטען בידי התוכנית.

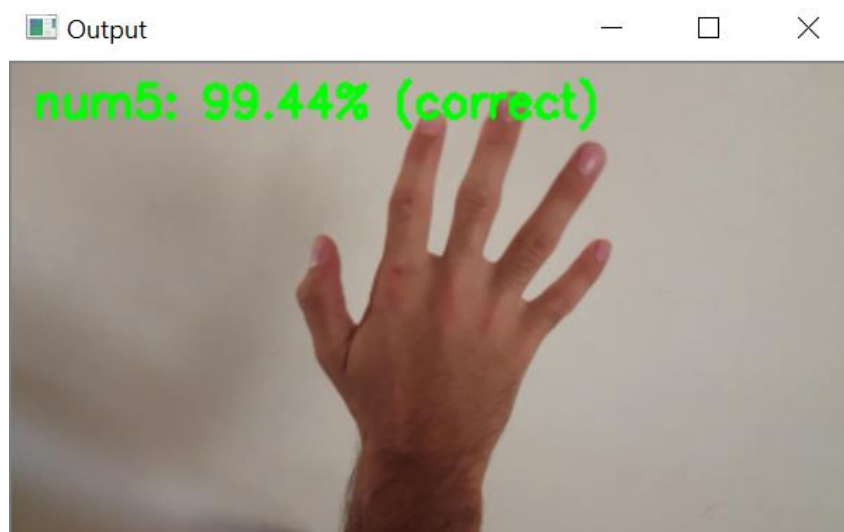
```
>>> Your Choice:
Enter: 3
Enter the image path:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\number5.jpg
Error - no such file or directory
Enter the image path:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\num5_H2_R.jpg
[INFO] Loading network...
2020-05-22 16:57:56.278284: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
```

ומיד לאחר מכן נקבל את הודעת אינפורמציה שתוכנית כעת חוזרת את הקטגוריה שאליה שייכת התמונה ולבסוף ניתן לראות שהמודל זיהה נכונה את התמונה.

קיבלנו כפלט שהתמונה היא מסוג num5. בעזרת קובץ התוויות אותו ציינתי לפני כן התוכנית אף יודעת לומר האם היא חזתה את הקטגוריה נכון או לא.

```
Enter the image path:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\num5_H2_R.jpg
[INFO] Loading network...
2020-05-22 16:57:56.278284: I tensorflow/core/platform
[INFO] Classifying image...
[INFO] num5: 99.44% (correct)
```

בנוסף להדפסה זו נפתח בפנינו מסך חדש ובו התמונה אותה ששלחנו כקלט ועליה תווית עם תוצאת החיזוי (מודפסת בירוק). רק לאחר שנסגור את החלון הנ"ל התוכנית תבקש מאיתנו לבחור אופציה חדשה.



מדריך למפתח:

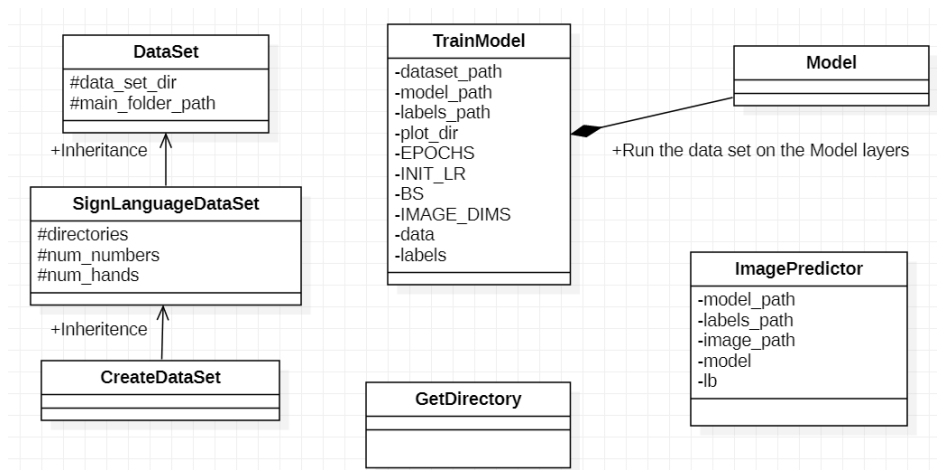
ראשית כל, הפרויקט שלי מחולק למספר קבצי קוד אשר לכל אחד ישנו תחום אחריות שונה. חלוקה זו בין חלקי הקוד השונים העוסקים בחלקים שונים בפרויקט מאפשרת ארגון קוד, ממזערת באגים למיניהם ואף איפשרה לי לבצע את הפרויקט ביתר קלות.

כעת אציין את שמות הקבצים השונים ותחום האחריות שהוטל על כל אחד מהם:

menu.py	קובץ זה מנהל את כל התוכנית לפי בחירות המשתמש – זהו הקובץ הראשי
Sort_Data_Set.py	בקובץ זה אחראי על יצירת מאגר תמונות ממוין לפי הקטגוריות השונות. הוא מקבל כקלט את התמונות באופן לא ממוין.
Extract_Zip_File.py	אם התוכנית קיבלה כקלט את התמונות בקובץ zip קובץ זה אחראי על חילוץ התמונות לתיקייה.
Model.py	קובץ זה אחראי על הגדרת שכבות המודל.
train_model.py	קובץ זה אחראי על אימון המודל (משתמש במודל שהוגדר בקובץ Model.py ובמאגר התמונות שיצר הקובץ Sort_Data_Set.py
classify.py	קובץ זה אחראי על חיזוי קטגוריה של תמונה שהתקבלה כקלט – The Predict Module
Cheak_Dir.py	קובץ זה אחראי על תקינות קלטי המשתמש. ההתייחסות הינה לקלטי המיקום של קבצים ותיקיות שונות במחשב.
PrintsForUser.py	קובץ זה אחראי על ההדפסות השונות המוצגות למשתמש

בחלק מן הקבצים הפונקציונאליות רוכזה במחלקות והפונקציות השונות בקובץ ה menu.py יצרו מופעים שונים שלהן או שפשוט השתמשו במטודות הסטטיות של אותה מחלקה.

Class diagram זו מציגה את המחלקות השונות בפרויקט (בנספחים ניתן למצוא את הדיאגרמה המלאה הכוללת גם את המטודות השונות).



כעת נעבור על הקבצים השונים ועל הפונקציות הכלולה בהם:

קובץ ראשי – menu.py :

כפי שציינתי בטבלה קובץ זה מנהל את מהלך התוכנית לפי קלטי המשתמש. על מנת לבצע את הפונקציות שבה המשתמש בחר, קובץ זה משתמש בשאר הקבצים. בקובץ זה מצויה הפונקציה הראשית (ששמה menu()) המהווה פונקציית שירות עבור המשתמש. כאשר המשתמש מריץ את קובץ פייטון זה, מודפסת למסך רשימה הכוללת בתוכה את האפשרויות שהמשתמש יכול לבצע.

```
Anaconda Prompt - python "C:\Users\ilano\OneDrive\Desktop\Projects\python\Deep Learning\Final Project\menu.py"

(base) C:\Users\ilano>python "C:\Users\ilano\OneDrive\Desktop\Projects\python\Deep Learning\Final Project\menu.py"
Using TensorFlow backend.
*****
*          USER INTERFACE          *
*                                  *
* Enter 1 --> create sorted data set *
* Enter 2 --> train the model       *
* Enter 3 --> predict an image      *
* Enter space bar --> exit          *
*                                  *
*****
--> Your Choice:
Enter: _
```

כפי שניתן לראות מוצעות ארבע אפשרויות:

1 – ליצור מאגר תמונות ממזין

2 – לאמן את המודל

3 – לחזות את הקטגוריה של תמונה שהמשתמש ישלח כקלט

רווח – ליציאה מן התוכנית (יש להקיש רווח ואחריו Enter).

עבור כל קלט אחר התוכנית תמשיך לבקש קלט.

המשתמש יכול לבצע חיזוי ללא אימון או אימון ללא מיון – התוכנה תשתמש בקבצים שבהם שמורים הנתונים – (קובץ אשר ממזין מראש, קובץ שבו נשמר המודל וכן הלאה). אימון המודל לדוגמה וביצוע חיזוי של תמונה אחריו ישתמש במודל העדכני.

לפני שניכנס לאופציות השונות אציג את האופן בו מומש קובץ זה.

```
6 """
7 this is the main python file that manege the program according to the user choicess
8 """
9
10 import Sort_Data_Set
11 import Extract_Zip_File
12 import train_model
13 import classify
14 import Cheak_Dir
15 import PrintsForUser
16
17 def options():
18     pass
19
20 def case_One(data_set):
21     pass
22
23 def case_Two(sorted_data_path):
24     pass
25
26 def case_Three(model_path, labels_path):
27     pass
28
29 def menu():
30     pass
31
32
33 if __name__ == "__main__":
34     menu()
```

תפקידה	הפונקציה
<p>תחום אחריות:</p> <p>פונקציה זו אחראית על ההדפסה של אפשרויות המשתמש למסך (היא עושה שימוש בפונקציה מתוך הקובץ PrintsForUser.py (הסבר בהמשך).</p>	Options()
<p>תחום אחריות:</p> <p>אם המשתמש הקיש 1 הפונקציה הראשית menu() תקרא לפונקציה זו. פונקציה זו אחראית על השימוש בקובץ Sort_Data_Set.py, כלומר אחראית על הקישור בין הקובץ הראשי לפונקציונאליות שבקובץ זה – מיון מאגר התמונות.</p> <p>אופן ביצוע:</p> <p>פונקציה זו מקבלת את המיקום של תיקיית התמונות במחשב, מחלצת את התמונות מן הקובץ לתיקייה שהמשתמש הזין כקלט (במידה והתמונות שמורות בקובץ zip), משתמשת בקובץ שציינתי לשם ביצוע המיון ומחזירה את מיקום התיקייה (שהמשתמש בחר לשמור בה את המיון) של מאגר הנתונים הממוין.</p>	case_One(data_set)
<p>תחום אחריות:</p> <p>אם המשתמש הקיש 2 הפונקציה הראשית menu() תקרא לפונקציה זו. פונקציה זו אחראית על הקישור בין הקובץ הראשי לבין הקובץ האחראי על אימון המודל.</p> <p>אופן ביצוע:</p> <p>פונקציה זו מקבלת את המיקום במחשב של התיקייה שבה התמונות ממוינות לפי קטגוריות, משתמשת בפונקציונאליות שבקובץ שציינתי ומחזירה את מיקום הקובץ שבו שמור המודל, ואת מיקום הקובץ שבו שמורות תוויות התמונות (הסבר בקובץ train_model.py).</p>	case_Two(sorted_data_set)
<p>תחום אחריות:</p> <p>אם המשתמש הקיש 3 הפונקציה הראשית menu() תקרא לפונקציה זו. פונקציה זו אחראית על הקישור בין הקובץ הראשי לבין הקובץ האחראי על חיזוי תמונה.</p> <p>אופן ביצוע:</p> <p>פונקציה זו מקבלת את המיקום במחשב של מיקום הקובץ שבו שמור המודל, ואת מיקום הקובץ שבו שמורות תוויות התמונות, משתמשת בפונקציונאליות שבקובץ שציינתי ומדפיסה את תוצאת החיזוי (הסבר בקובץ classify.py).</p>	case_Three(model_path, labels_path)
מנהלת בלולאה את התוכנית	menu()

הלולאה הראשית שבפונקציית ה `.menu()`

```
data_set = r"C:\Users\ilano\OneDrive\Desktop\images"
sorted_data_path = r"C:\Users\ilano\OneDrive\Desktop\my_dataset"
model_path = r"C:\Users\ilano\OneDrive\Desktop\result\Model.model"
labels_path = r"C:\Users\ilano\OneDrive\Desktop\result\lb.txt"

while(flag):
    PrintsForUser.printOptions("--> Your Choice: ")
    choice = input("Enter: ")

    if choice == '1':
        """
        if the use enter 1 -> the directory of the data will be updated
        """
        data_set, sorted_data_path = case_One(data_set)
        PrintsForUser.printProcess("[INFO] Using new data set")

    if choice == '2':
        """
        if the use enter 2 -> the directory of the model and the labeld will be updated
        """
        model_path, labels_path = case_Two(sorted_data_path)
        PrintsForUser.printProcess("[INFO] Using trained model")

    if choice == '3':
        """
        if the use enter 3 -> the program will use the updated directory and predict the image
        """
        case_Three(model_path, labels_path)

    if choice == ' ':
        PrintsForUser.printProcess("[INFO] Exiting...")
        flag = False
```

כעת נעבור על האפשרויות השונות (cases) הנקראים לפי קלטי המשתמש.

Case 1

קובץ חילוץ התמונות – Extract_Zip_File.py:

בקובץ זה מומשה הפונקציונאליות הנדרשת לחילוץ קבצים מתוך תיקייה zip במידה וזו אכן תיקייה מסוג זה. בקובץ זה ישנה פונקציה אחת בלבד בשם extract_Zip(path) המקבלת path כלשהוא כפרמטר. במידה וזהו אכן קובץ zip הפונקציה תודיע על כך למשתמש ותדרוש ממנו directory חדש בו תשמור את הקבצים שתחלץ מן ה path שקיבלה כפרמטר.

אם ה path אינו מצביע על קובץ zip יוחזר אותו path אחרת יוחזר ה path החדש שהמשתמש הזין כקלט.

```
def extract_Zip(zip_path):
    path = zip_path
    """
    ls is a list of put substring thar split by '.'
    """
    ls = zip_path.split(".")
    """
    we cheak if the file/folder is zip by its ending - file type
    """
    if ls[len(ls)-1] == "zip":
        PrintsForUser.printProcess("[INFO] Got a zip file...")
        path = Cheak_Dir.GetDirectory.get_New_Dir("Choose the path to extract the files (with a new folder name)")
        """
        extract the folder to new location that the user chose
        """
        with ZipFile(zip_path, 'r') as zipObj:
            # Extract all the contents of zip file in different directory
            PrintsForUser.printProcess("[INFO] Extracting the images from the zip file...")
            zipObj.extractall(path)

        PrintsForUser.printProcess("[INFO] Finished extracting process...")
    else:
        PrintsForUser.printProcess("[INFO] Got ordinary file...")
    return path
```

קובץ יצירת המאגר הממוין – Create_Data_Set.py:

בקובץ זה הפונקציונאליות מומשה באמצעות ריכוזה במחלקות שונות אשר ביניהן קשרי ירשה.

ישנן שלושה מחלקות:

המחלקה הבסיסית ביותר הינה המחלקה האבסטרקטית **DataSet**. מחלקה זו הגדירה את הטיפוס הבסיסי היוצר מאגר נתונים של תמונות. כעת נעבור על הפונקציונאליות שהגדירה מחלקה זו.

```
21 class DataSet(ABC):
22
23     @abstractmethod
24     def __init__(self, data_set, main_folder):
25
26         self._data_set_dir = data_set
27         self._main_folder_path = main_folder
28
29     def _make_Folder(self, path):
30         try:
31             os.mkdir(path)
32
33         except:
34             PrintsForUser.printError("Error - Creation of the directory %s failed: " + path)
35
36     @abstractmethod
37     def make_All_Folders(self):
38         pass
39
40     @abstractmethod
41     def initialization_Data(self):
42         pass
```

תפקיד	ערך	תכונה
מחזיקה את מיקום התיקייה שבה נמצא מאגר התמונות הלא ממוין.	Data set directory	_data_set_dir
מחזיקה את המיקום שבו בחר המשתמש לשמור את התמונות לאחר שיעברו מיון לפי קטגוריות.	input	_main_folder_path

תפקידה	המטודה
הבנאי של המחלקה. בנאי זה מגדיר את התכונות המשותפות (בעלות הרשאת protected) כל האובייקטים מסוג זה של המחלקה (נדגיש כי המחלקה אבסטרקטית – כלומר ניתן ליצור אובייקטים מטיפוס הירש את מחלקה זו).	__init__(self, data_set, main_folder)
מטודה בעלת הרשאת protected. בהינתן מיקום המטודה תיצור תיקייה. מטודה זו תשרת את המטודה הפומבית .make_All_Folders() מימוש: המטודה תנסה ליצור תיקייה ואם תיכשל תזרוק חריגה והודעה למשתמש. (בהמשך נראה כי אינה כל אפשרות שתזרוק חריגה – הקובץ Ceak_Dir.py מונע זאת.	_make_Folder(self, path)

@abstractmethod make_All_Folders(self)	מטודה פומבית זו הינה אבסטרקטית. כלומר חסרת מימוש אך כל מחלקה יורשת תחויב לממשה. מטודה זו אחראית על יצירת כל התיקיות
@abstractmethod initialization_Data(self)	מטודה פומבית זו הינה אבסטרקטית. כלומר חסרת מימוש אך כל מחלקה יורשת תחויב לממשה. מטודה זו אחראית לאתחל את התיקיות שיצרה המטודה make_All_Folders(self) בתמונות, אופן ממוין.

את מחלקה זו ירשה המחלקה האבסטרקטית **SignLanguageDataSet** אשר הוסיפה פונקציונאליות ותכונות על קודמתה. למעשה מחלקה זו הגדירה את הייחודיות שבמאגר נתונים שבו תמונות המייצגות ספרות בשפת הסימנים (כמובן בהתאמה למאגר הנתונים שאתו אני עובד).

```
class SignLanguageDataSet(DataSet):
    def __init__(self, data_set, main_folder):
        """
        Sign_Language_Data_Set Constactor:
        Get: data_set = path to the all images
            main_folder = path to the folder which the user chose to sort the images
        """
        super(SignLanguageDataSet, self).__init__(data_set, main_folder)
        self._directories = [] # a list of all the direction that we will create in our "make_All_Folders" method
        self._num_categories = 10 #the num of folders
```

תכונה	ערך	תפקיד
_directories	[]	רשימה שתאכלס את כל המיקומים שבהם ייווצרו התיקיות השונות.
_num_categories	10	מספר הקטגוריות שעל בסיסן אנו ממיינים.

מטודה	תפקידה
__init__(self, data_set, main_folder)	בנאי המחלקה. הפרמטרים שבנאי המחלקה מקבל משמשים לאתחול החלק מסוג DataSet באובייקט. כלומר יש כאן קריאה לבנאי המחלקה שממנה ירשנו באמצעות super(). בנוסף בנאי זה הוסיף תכונות נוספות בהרשאת .protected
_get_Image_Category(self, image_name)	מטודה זו מחלצת משם התמונה את הקטגוריה אליה היא שייכת. numx_Hy index מאחר שכל הפרמטרים המייצגים את התמונה מופרטים במכף תחתון ניתן להפריד בין המחרוזות ופענח כל פרמטר בנפרד. במטרה לפענח את השם של כל תמונה (שהרי הוא מחרוזת), מטודה זו משתמשת במטודה נוספת.

	<p>מטודה זו מחזירה את המספר המיוצג</p> <p>הערה חשובה: מאחר ואני יצרתי את מאגר התמונות בכוחות עצמי קבעתי כי כל התמונות באותו הסוג יקראו num_x_Hy index – שם זה מייצג את התמונה.</p> <p>לדוגמה: 13 nam1_H1 – בתמונה זו ישנה הספרה אחת של איש מספר אחד. (אין משמעות למספר הסידורי).</p>
_num_Switcher(self, s)	<p>מטודה זו בהינתן מחרוזת בפורמט numx תחזיר את x כמספר.</p> <p>מטודה זו משתמשת במילון (מפה) אשר ממפה בין קבוצת המפתחות numx (מחרוזות) למספרים מ 0 עד 10 (x).</p>
@abstractmethod make_All_Folders(self)	<p>מטודה פומבית זו הינה אבסטרקטית. כלומר חסרת מימוש אך כל מחלקה יורשת תחויב לממשה. מטודה זו אחראית על יצירת כל התיקיות</p>
@abstractmethod initialization_Data(self)	<p>מטודה פומבית זו הינה אבסטרקטית. כלומר חסרת מימוש אך כל מחלקה יורשת תחויב לממשה. מטודה זו אחראית לאתחל את התיקיות שיצרה המטודה make_All_Folders(self) בתמונות, אופן ממוין.</p>

```

114 class CreateDataSet(SignLanguageDataSet):
115
116     def __init__(self, data_set, main_folder):
117         """
118         Create_Data_Set Constactor:
119         Get: data_set = path to the all images
120             main_folder = path to the folder which the user chose to sort the images
121         """
122         super(CreateDataSet, self).__init__(data_set, main_folder)
123
124     def make_All_Folders(self):
125         """
126         -- mainFolder:
127         -- num0
128         -- num1
129         -- num2
130         -- num3
131         -- num4
132         -- num5
133         -- num6
134         -- num7
135         -- num8
136         -- num9
137         -- num10
138         """

```

המחלקה **CreateDataSet** יורשת מן המחלקה **SignLanguageDataSet** והיא כשמה יוצרת את מאגר הנתונים הממוין בפועל. מחלקה זו מימשה את המטודות האבסטרקטיות שירשה מאבותיה.

מטודה	תפקידה
<code>__init__(self, data_set, main_folder)</code>	בנאי המחלקה. הפרמטרים שבנאי המחלקה מקבל משמשים לאתחול החלק מסוג <code>DataSet</code> באובייקט. כלומר יש כאן קריאה לבנאי המחלקה שממנה ירשנו באמצעות <code>.super()</code> .
<code>make_All_Folders(self)</code>	מטודה זו יוצרת את התיקיות תחת אותו <code>directory</code> ראשי שאותו בחר המשתמש. מטודה זו יוצרת למעשה את "עץ" התיקיות הבא: תיקייה ראשית בהתאם למיקום המוחזק בתכונה <code>self._main_folder_path</code> ותחתיה יוצרת 11 תיקיות כמספר הקטגוריות השונות בפרויקט ה"ל". (ניתן לראות את המבנה בתיעוד של הפונקציה). את מיקומי (ה <code>directories</code>) של התיקיות המטודה הוסיפה לרשימה השמורה כתכונה <code>self._derictions</code> אשר בה יעשה שימוש בעת אתחול התיקיות.
<code>initialization_Data(self)</code>	מטודה זו אחראית על אתחול התיקיות בתמונות הרלוונטיות. באמצעות המטודה <code>self._get_Image_Properties(self, image_name)</code> המטודה יודעת מהו סוג כל תמונה ומעתיקה אותה למיקום הרלוונטי שברשימה <code>self._directions</code>

בכדי להבין באופן מיטבי כיצד נעשה שימוש בעץ ירושה זה נבחן שוב את הפונקציה `case_One(data_set)` שבקובץ `menu.py`.

```

1 import Sort_Data_Set
2 import Extract_Zip_File
3 import train_model
4 import classify
5 import Cheak_Dir
6 import PrintsForUser
7
8
9 def case_One(data_set):
10     extracted_data = Extract_Zip_File.extract_Zip(data_set) # return the folder path after extract (if it was a zip file at first)
11     sorted_data_path = Cheak_Dir.get_New_Dir("Please enter the path that you want for your main data set folder:\nPath Must be in English!", True)
12     data_obj = Sort_Data_Set.CreateDataSet(extracted_data, sorted_data_path)
13     data_obj.make_All_Folders() # this method creat the folders tree of drictories
14     data_obj.initialization_Data() #this method initializing the folder by the current images
15     return extracted_data, sorted_data_path

```


מספר שורה	הפקודה
10	בשורה זו אנו קוראים לפונקציה <code>extract_Zip</code> מתוך הקובץ <code>Extract_Zip_File.py</code> . במידה והתיקייה שקיבלנו שבה מאוחסן מאגר התמונות הינה מסוג <code>zip</code> פונקציה זו תדאג לחלץ את הקבצים מן התיקייה ל <code>directory</code> שהמשתמש יבחר (הרחבה בהמשך). פונקציה זו מחזירה את ה <code>directory</code> החדש, ואם התיקייה אינה הייתה מסוג <code>zip</code> יוחזר מיקום זהה לזה שהתקבל כפרמטר.
11	בשורה זו אנו קוראים לפונקציה <code>get_New_Dir</code> מתוך הקובץ <code>Cheak_Dir.py</code> . בשורה זו אנו קולטים מן המשתמש את ה <code>directory</code> שבו הוא רוצה שיווצר ה <code>data set</code> הממוין. (אופן הפעולה של פונקציית קלט זו בהמשך).
12	בשורה זו אנו יוצרים אובייקט מסוג <code>CreateDataSet</code> ושולחים לבנאי את שני הפרמטרים הנדרשים: <code>extracted_data</code> ו- <code>sorted_data_set</code> .
13	אנו מזמנים את המטודה יצירת התיקיות
14	אנו מזמנים את המטודה לאתחול התיקיות
15	ולבסוף מחזירים את שני הפרמטרים במטרה לעדכן את ערכיהם בפונקציה הראשית <code>menu()</code>

Case 2

קובץ הגדרת מבנה המודל - Model.py:

בקובץ זה יצרתי מחלקה בשם MyModel אשר מגדירה את מבנה המודל שעל גביו נבצע את למידת התמונות.

במחלקה זו ישנה מטודה סטטית אחת בשם build.

מטודה זו בהינתן ממדי התמונות, מספר הקטגוריות וה- activation function (שתבצע בשכבה האחרונה – באופן דיפולטיבי softmax function) תבנה את שכבות המודל.

אז מה בעצם מגדירות שכבות המודל?

שכבות המודל מהוות יחדיו את המוח המלאכותי אותו יצרנו. בכל שכבה ישנו מספר שונה של נוירונים השקולים לתאי העצב במוח האנושי. במוח האדם תאי העצב מחוברים לתאי העצב הסמוכים אליהם באמצעות שלוחות התא. הקישוריות הזאת בין התאים יוצרת רשת אחת גדולה ומסועפת שמהווה את המוח האנושי. השלוחות שבין התאים מעבירות אותו חשמליים שמשמשים להעברת מידע בין התאים. תקשורת זו היא שמאפשרת לאדם לחשוב ולפעול. באופן דומה כך בנוי גם המודל. במודל ישנן מספר שכבות אשר בכל אחת מהן מספר שונה של נוירונים מלאכותיים. הנוירונים בכל שכבה יכולים "לתקשר" עם הנוירונים שבשכבות הסמוכות. אופן התקשורת בין השכבות אינו נעשה כמובן כפי שהוא נעשה במוח האנושי אלא על סמך משקלים שמהווים חישוב סטטיסטי אשר יניב את אחוז ההצלחה הגבוה ביותר.

בעת זימון פונקציה זו ערכי המשקלים הללו הינו רנדומלי ורק בעת אימון המודל הם משתנים בהתאם למערכי התמונות שרצים על גביהם.

בחרתי במודל זה מכיוון שלאחר הרצת התמונות על גביו הוא הניב אחוז הצלחה גבוהה מזה שהניב המודל הקודם.

במודל זה 27 שכבות כאשר החשובות שבהן הינן הראשונה והאחרונה.

השכבה הראשונה היא שכבת הקלט המגדירה את ממדי התמונה - input_shape בפרויקט שלי (1,50,50), והשכבה האחרונה היא שכבת הפלט. חשוב לציין שבשכבה אחת לפנייה הגדרנו את מספר הקטגוריות האפשרויות – מספר נוירוני הפלט.

השכבה האחרונה באופן דיפולטיבי (במידה ואין קלט אחר) משתמשת בפונקציית softmax.

כאשר מריצים תמונה על גבי המודל המאומן היא צפויה להגיע אל אחד מנוירוני הפלט. מכיוון שמדובר בחישובים כאלו ואחרים התשובה אינה חד משמעית, ולכן נבחר באפשרות שישנו הסיכוי הגבוהה ביותר שהוא הנכון -> וזה בדיוק מה שפונקציה זו עושה -> בוחרת את נוירון הפלט אשר הינו המתאים ביותר עבור אותה התמונה.

המודל הנוכחי:

```
class MyModel:
    @staticmethod
    def build(width, height, depth, classes, finalAct="softmax"):

        model = Sequential()
        inputShape = (height, width, depth)
        chanDim = -1

        if K.image_data_format() == "channels_first":
            inputShape = (depth, height, width)
            chanDim = 1

        model.add(Conv2D(32, (3, 3), padding="same", input_shape=inputShape))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(MaxPooling2D(pool_size=(3, 3)))
        model.add(Dropout(0.25))
        model.add(Conv2D(64, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(Conv2D(64, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))
        model.add(Conv2D(128, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(Conv2D(128, (3, 3), padding="same"))
        model.add(Activation("relu"))
        model.add(BatchNormalization(axis=chanDim))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))
        model.add(Flatten())
        model.add(Dense(1024))
        model.add(Activation("relu"))
        model.add(BatchNormalization())
        model.add(Dropout(0.5))
        model.add(Dense(classes))
        model.add(Activation(finalAct))

    return model
```

המודל הקודם:

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(96, 54, 1)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(11, activation='softmax'))
```

קובץ אימון המודל – train_model.py :

כפי שציינתי קודם לכן קובץ זה אחראי על אימון המודל. בקובץ זה ישנה מחלקה בשם TrainModel אשר בה מטודה אחת פומבית המשתמשת במספר מטודות פרטיות במטרה לבצע את אימון המודל. הצורך במחלקה נבע מכך ששלל הפונקציות הנדרשות זקוקות לפרמטרים זהים - < דבר המוביל ליצירת מחלקה.

ראשית כל נעבור את בנאי המחלקה:

כל ה data members הם הרשאות private.

```
class TrainModel():
    def __init__(self, dataset_path, model_path, labels_path, plot_dir):
        self.__dataset_path = dataset_path # the data set directory
        self.__model_path = model_path # the directory the user chose to save the trained model
        self.__labels_path = labels_path # the directory the user chose to save the images labels
        self.__plot_dir = plot_dir # the directory to the folder that the user chose to save the graph images

        self.__EPOCHS = 10 #number of epochs
        self.__INIT_LR = 1e-3 #Learning rate
        self.__BS = 32 #batch size
        self.__IMAGE_DIMS = (50, 50, 1) #image dimensions
        self.__data = [] # List of all the images as arrays
        self.__labels = [] #List Labels of all the images
```

תפקיד	ערך	המשתנה
מחזיק את הכתובת שבה נמצא מאגר התמונות הממוין	Input	__dataset_path
ה directory שבו בחר המשתמש לשמור את המודל	Input	__model_path
ה directory שבו בחר המשתמש לשמור את תוויות התמונות.	Input	__labels_path
ה directory לתיקייה בה בחר המשתמש לשמור את תוצאות הגרפים (קבצי png).	Input	__plot_dir
משתנה זה קובע את מספר הפעמים שבו נריץ את כל התמונות על גבי המודל לצורך ביצוע האימון (הלמידה).	10	__EPOCHES
	1*e - 3	__INIT_LR
מספר התמונות שבזמן אימון (לימוד) המודל יעבד במקביל.	32	__BS
ממדי התמונות שהמודל יריץ (width, height, color)	(50,50,1)	__IMAGE_DIMS
רשימה שתחזיק את המערכים שייצגו את התמונות. הסבר הצורך: הפונקציה loading_Images(dataset_path) שבה מתבצעת טעינת התמונות. פונקציה זו תמיר את התמונות למערכים ותשמור אותם ברשימה זו. במקביל פונקציה זו תשמור את הקטגוריה של כל תמונה ברשימת ה labels. כך עבור כל תמונה נשמור את הקטגוריה שלה.	[]	__Data
רשימה שתחזיק את התוויות של כל התמונות.	[]	__labels

כעת נעבור על המטודות השונות. במחלקה זו זה ישנה מטודה פומבית אחת המהווה את הפונקציה הראשית בשם `handle_train()`. מטודה זו האחראית לבצע את אימון המודל באמצעות המטודות הפרטיות העומדות לרשותה. כלומר תפקידה של מטודה זו היא לנהל את מהלך האימון, ולכן היא הפונקציה הפומבית היחידה במחלקה.

כעת נעבור על המטודות השונות בקובץ זה:

תפקידה	הפונקציה
<p>מטודה פרטית זו אחראית על טעינת התמונות מן התיקייה הראשית והכנתן לאימון המודל.</p> <p>אופן המימוש:</p> <p>מטודה זו משנה את גודלי ממדי התמונה לגודל אחיד שנקבע בתכונה <code>__IMAGE_DIMS</code>, ממירה את התמונות מ RGB ל GrayScale ואף ממירה אותן למערכים (המחזיקים את הפיקסלים של התמונות).</p> <p>בנוסף פונקציה זו מחלצת מן שם התיקייה שבה כל תמונה נמצאת את הקטגוריה שלה ומתאימה ביניהן באמצעות שתי רשימות המהוות חלק מתכונות המחלקה.</p> <p>רשימה אחת מאכלסת את מערכי התמונות (<code>__data</code>) והשנייה את התוויות (<code>__labels</code> - סוג התמונה - <code>num</code>). מכיוון שההוספה לרשימות נעשית ביחד בלולאה ישנה התאמה במספר האינדקס בין שתי הרשימות ובכך ניתן לקשר בין כל תמונה לקטגוריה שלה.</p> <p>מטודה זו אינה מחזירה ערך שכן השינוי נעשה על ה- <code>data</code> members של המחלקה (על תכונות האובייקט).</p>	<code>__loading_Images()</code>
<p>מטודה פרטית זו מקטינה את טווח הפיקסלים של המערכים השמורים ב <code>__data</code> מן התחום <code>[0,255]</code> לתחום <code>[0, 1]</code>.</p>	<code>__scale_Pixels()</code>
<p>מטודה פרטית זו מבצעת את אימון המודל בפועל. נבחן את הפונקציה במבט מלמעלה:</p> <p>ראשית כל, מטודה זו שומרת את רשימת התוויות בקובץ בינארי אשר ישמש באימון המודל וכן בחיזוי קטגוריות של תמונות בהמשך.</p> <p>בנוסף מטודה זו יוצרת טרנספורמציה מן הקובץ הבינארי שתשמש לאימון המודל.</p> <p>מטודה זו מחלקת את ה <code>data set</code> לשני חלקים <code>train</code> ו <code>test</code> ביחס של 80:20. כלומר 80% מן התמונות ישמשו לאימון המודל ו 20% מן התמונות ישמשו לבדיקת אחוז ההצלחה של המודל – לבדיקתו.</p> <p>לבסוף מטודה זו מחזירה את ההיסטוריה של לימוד המודל. בנוסף, מטודה זו שומרת את למידת המודל בקובץ ואף את קובץ התוויות הבינארי כדי שיהיה ניתן להשתמש בלמידה</p>	<code>__train()</code>

	הנוכחית גם בהרצות אחרות של התוכנית ואף בשביל שיהיה ניתן לבצע predict מבלי בכל פעם להריץ את המודל מחדש.
Graph1(history, plot_path)	<p>מטודה פרטית זו מקבלת כפרמטר את ההיסטוריה וה- directory בו נשמור את פלט המטודה. <code>directory = __plot_dir+ r"\plot1.png"</code></p> <p>מטודה זו משתמשת בהיסטוריה ויוצרת קובץ תמונה מסוג png שבו היא משרטטת את גרף הלימוד את המודל.</p>
graph2(history, plot_path)	<p>מטודה פרטית זו מבצעת באופן מעט שונה את אותו הדבר כפי הפונקציה הראשונה. (גרף נוסף). <code>directory = __plot_dir+ r"\plot2.png"</code></p>
handle_train()	<p>המטודה הפומבית היחידה של המחלקה. ניתן לקרוא למטודה זו רק מתוך אובייקט מאותחל של המחלקה (לכן יש בנאי עם פרמטרים). ניתן להפעיל מתון אובייקט של המחלקה את המטודה זו בלבד והיא זו שדואגת לביצוע האימון -> בעזרת המטודות הפרטיות.</p> <pre> def handle_train(self): """ this public method manage the train section return the train model path and the images labels path """ self.__loading_Images() self.__scale_Pixels() history = self.__train() self.__graph1(history, self.__plot_dir+ r"\plot1.png") self.__graph2(history, self.__plot_dir +r"\plot2.png") </pre>

הפונקציה case_Two(sorted_data_path) אשר בקובץ ה menu.py קולטת מן המשתמש:

1. The model directory
2. The binary labels file directory
3. A folder directory to save their graphs images.

הפונקציה תמשיך לדרוש קלט מן המשתמש במידה והזין directory הזהה לאחד מן ה directories הקודמים שהתן כקלט.

לאחר מכן הפונקציה הזו יוצרת אובייקט מסוג TrainModel ומזמנת את המטודה handle_train()

לבסוף הפונקציה מחזירה את ה directory בו נשמר המודל ואת ה directory בו נשמר הקובץ הבינארי של התוויות (של התמונות).

```

def case_Two(sorted_data_path):

    model_path = Cheak_Dir.GetDirectory.get_New_Dir("Enter the full path (with the name) to output model: ")
    lb_path = Cheak_Dir.GetDirectory.get_New_Dir("Enter the full path (with the name) to output label binarizer: ")

    while model_path == lb_path:
        PrintsForUser.printError("Error - file will be override")
        lb_path = Cheak_Dir.GetDirectory.get_New_Dir("Enter the full path (with the name) to output label binarizer: ")

    plot_dir = Cheak_Dir.GetDirectory.get_New_Dir("Enter the folder directory to output accuracy/loss plot: ")

    while model_path == plot_dir or lb_path == plot_dir:
        PrintsForUser.printError("Error - file will be override")
        plot_dir = Cheak_Dir.GetDirectory.get_New_Dir("Enter the folder directory to output accuracy/loss plot: ")

    os.mkdir(plot_dir)
    |
    train_obj = train_model.TrainModel(sorted_data_path, model_path, lb_path, plot_dir)

    train_obj.handle_train()

    return model_path, lb_path

```

Case 3

קובץ ביצוע ה predict – classify.py:

קובץ זה אחראי על ביצוע ה case השלישי – בהינתן תמונה הקובץ יחזה לאיזו קטגוריה היא שייכת. בקובץ זה ישנה מחלקה בשם ImagePredictor הדורשת בבנאי שלה שני פרמטרים:

1. The saved model directory
2. The images labels binary file's directory

במחלקה הנ"ל יש מטודה פומבית אחת המזמנת מטודות פרטיות ובאמצעותן מממשת את מטרת הקובץ. מטודה זו מקבלת את ה directory לתמונה אותה רוצים לחזות. בכך למעשה ניתן להשתמש באובייקט אחד כדי לחזות מספר תמונות שונות.

לשם ביצוע מטלה זו יצרתי אובייקט מסוג המחלקה ונתתי את הפרמטרים הדרושים. אחת ממטודות המחלקה טוענת את המודל השמור אצלי לוקאלית במחשב ואחרת ומריצה על גביו את התמונה שהתקבלה כקלט מן המשתמש. מטודה אחרת תחמצא את הקטגוריה אשר לה הסיכוי הגבוה ביותר להיות הנכונה, ובעזרת הקובץ הבינארי ששמרנו בעת אימון המודל הפונקציונאליות שבקובץ אף תציג למשתמש כפלט אם הקטגוריה אותה זיהה היא נכונה או לא (הוא למעשה ישווה בין התווית המתאימה לתמונה לבין תוצאת החיזוי).

ראשית כל נעבור על בנאי המחלקה:

```
class ImagePredictor():  
    def __init__(self, model_path, labels_path):  
        self.__model_path = model_path  
        self.__labels_path = labels_path  
        self.__image_path = ""  
        self.__model = None  
        self.__lb = None
```

תכונה	ערך	תפקיד
__model_path	input	מכיל את ה directory שבו שמור המודל שהורץ.
__labels_path	input	מכיל את ה directory בו שמור הקובץ הבינארי המכיל את התוויות של התמונות.
__image_path	""	מכיל את ה directory של התמונה שהמשתמש שלח כקלט.
__model	None – a default value	אל תכונה זו נטען את המודל השמור.
__lb	None – a default value	אל תכונה זו נקרא את הקובץ הבינארי השמור.

כעת נעבור על המטודות של המחלקה אשר ממשות יחדיו את תחום האחריות עליו ממונה הקובץ.

פונקציה	תפקידה
<code>__load_Model()</code>	מטודה פרטית טוענת את מודל השמור וקוראת את הקובץ הבינארי ומאתחל את תכונו המחלקה: <code>__model</code> , <code>__lb</code> .
<code>__load_Image()</code>	מטודה פרטית זו מתאימה את התמונה השמורה ב <code>directory</code> של <code>__image_path</code> , לריצה על המודל. כפי שבוצע על תמונות לפני האימון: משנה את ממדי התמונה לגודל אחיד שקבעתי (96,54,1) ממירה את התמונה מ RGB ל GrayScale, לאחר מכן ממירה את התמונה למערך ואף מקטינה את טווח הפיקסלים שלה מ [0,255] ל [0,1]. לבסוף המטודה מחזירה את המערך המייצג את התמונה והעתק של התמונה כפי שהתקבלה כקלט.
<code>__predict_Image(image_arr, output)</code>	מטודה פרטית זו אחראית על ביצוע החיזוי עצמו. היא מקבלת כקלט את הפלט של המתודה הפרטית <code>__load_image()</code> : 1. המערך שמייצג את התמונה 2. העתק של התמונה המטודה מבצעת <code>predict</code> לתמונה על גבי המודל שנטען ומדפיסה את ההעתק התמונה של גביו היא כתבה כתווית את אחוז הסיכוי שהמודל זיהה נכון את התמונה (לפי חישוביו...) והאם הזיהוי נכון או לא נכון (זאת על ידי השוואה בין התווית של התמונה שמצויה בשם הקובץ לבין התווית שהמודל זיהה (התווית מציינת את הקטגוריה אליה שייכת התמונה). המטודה: <pre> def __predict_Image(self, image_arr, output): PrintsForUser.printProcess("[INFO] Classifying image...") proba = self.__model.predict(image_arr)[0] idx = np.argmax(proba) label = self.__lb.classes_[idx] filename = self.__image_path[self.__image_path.rfind(os.path.sep) + 1:] correct = "correct" if filename.rfind(label) != -1 else "incorrect" label = "{}: {:.2f}% ({}).".format(label, proba[idx] * 100, correct) output = imutils.resize(output, width=400) cv2.putText(output, label, (10, 25), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2) PrintsForUser.printProcess("[INFO] {}".format(label)) cv2.imshow("Output", output) cv2.waitKey(0) </pre>

handle_classify(image_path)	<p>המטודה הפומבית היחידה שמהווה למעשה המטודה הראשית האחראית לזמן את שאר המטודות בסדר הנכון לשם קבלת הפלט הסופי.</p> <p>מטודה זו מקבלת כפרמטר את ה directory לתמונה אותה המשתמש בחר לחזור, והמטודה מאתחלת את תכונתה, image_path בפרמטר הזה.</p> <pre>def handle_classify(self, image_path): self.__load_Model() self.__image_path = image_path image_arr, output = self.__load_Image() self.__predict_Image(image_arr, output)</pre>
-----------------------------	---

הפונקציה case_Three(model_path, labels_path) אשר בקובץ ה menu.py מבקשת מן המשתמש לקבל כקלט את ה directory של התמונה אותה הוא רוצה לחזור, יוצרת אובייקט מסוג PredictAnImage וממנו מזמנת את המטודה handle_classify() מהקובץ .classify.py

את ה directories של המודל השמור הפונקציה ושל קובץ התוויות הפונקציה case_Three(model_path, labels_path) , מקבלת מן פונקציית ה menu().

כעת אציג את הקבצים האחראים על הקלט והפלט:

קובץ לבדיקת תקינות של Cheak_Dir – directory input

בכל פונקציה בקוד בה בוצע קלט של directory מן המשתמש הפונקציה הזו השתמשה בפונקציות אנאליות שבקובץ זה. קובץ זה הגדיר מחלקה בשם GetDirectory. מחלקה זו מכילה מטודות סטטיות אשר בודקות את תקינות הקלט הנדרשת באופן עיקש, כלומר במקרה של קלט לא תקין, יופיעו הודעות שגיאה רלוונטיות אודות אי התקינות והתוכנית תדרוש מן המשתמש קלט חדש עד שיוזן לה קלט תקין.

נעבור על המטודות של המחלקה המוגדרת בקובץ זה:

מטודה	תפקידה
@staticmethod is_Exsists(message, flagLang = False)	<p>מטודה פומבית וסטטית זו משמשת לביצוע קלט של directory הקיים במחשב, מן המשתמש.</p> <p>הפונקציה מקבלת את ההודעה שיש להציג למשתמש (מבקשת את הקלט הרלוונטי), ו flagLang אשר ערכו הדיפולטיבי הינו false. אם הדגל נשלח עם ערך True הדבר אומר ש directory שמכיל אותיות עבריות אינו ייחשב כ directory תקין גם אם הוא כן קיים במחשב.</p> <p>המטודה תמשיך ותבקש מן המשתמש קלט תקין כל עוד לא התקבל אחד כזה, ולבסוף תחזיר את הקלט.</p> <pre> @staticmethod def is_Exsists(message, flagLang = False): PrintsForUser.printOptions(message) path = input("Enter: ") while not GetDirectory.__cheak_Exsists_Dir(path, flagLang): PrintsForUser.printOptions(message) path = input("Enter: ") return path </pre>
@staticmethod def __cheak_Launguge(path)	<p>מטודה סטטית ופרטית זו מחזירה True אם ה path שהתקבל כקלט אינו מכיל אותיות עבריות אחרת תחזיר False.</p> <p>זוהי מטודת עזר.</p>
@staticmethod def __cheak_Exsists_Dir(path, flagLang)	<p>מטודת סטטית ופרטית אשר בהינתן path ודגל בודקת האם הוא אכן קיים, ואם הדגל הינו True אז בודקת גם את תווים המרכיבים את הכתובת.</p> <p>המטודה תחזיר True אם הכתובת תקינה בהתאם לדרישות אחרת תחזיר False.</p> <p>מטודת פרטית זו הינה מטודת עזר עבור הפונקציה הפומבית, is_Exsists(message, flagLang = False), אשר מזמנת אותה בלולאה עבור כל קלט של path שהיא מקבלת מן המשתמש.</p>

<pre>@staticmethod def get_New_Dir(message,flagLang = False)</pre>	<p>מטודה סטטית ופומבית זו משמשת לביצוע קלט של directory שהינו תקין ואינו קיים כבר במחשב מן המשתמש. פונקציה זו תחשיב directory כתקין אם הוא מכיל רצף של כתובת קיימת שבסופה תיקייה חדשה או שם קובץ חדש לדוגמה: C:\Users\ilano\OneDrive\Desktop – לא תקין שכן כבר קיים, C:\Users\ilano\OneDrive\Desktop\oha d.txt – (תקין) מטודה זו מקבלת את אותן הפרמטרים כמו המטודה is_Exists(message, flagLang = False) והיא פועלת באופן זהה אליו מלבד כך שקלט תקין ייחשב כ directory חדש ולא אכן שכבר קיים. כלומר מטודת העזר שבודקת את ה Path שמתקבל בכל פעם כקלט הינה שונה.</p> <pre>@staticmethod def get_New_Dir(message,flagLang = False): path = "" while not GetDirectory.__cheak_New_Dir(path, flagLang): PrintsForUser.printOptions(message) path = input("Enter: ") return path</pre>
<pre>@staticmethod __cheak_New_Dir(path, flagLang)</pre>	<p>מטודת סטטית ופרטית אשר בהינתן path ודגל בודקת האם ה path הינו חדש, ואם הדגל הינו True אז בודקת גם את תווים המרכיבים את הכתובת. המטודה תחזיר True אם הכתובת תקינה בהתאם לדרישות אחרת תחזיר False. מטודת פרטית זו הינה מטודת עזר עבור הפונקציה הפומבית get_New_Dir(message,flagLang = False), אשר מזמנת אותה בלולאה עבור כל קלט של path שהיא מקבלת מן המשתמש.</p>

הקובץ האחראי על הפליטים למשתמש – PrintsForUser.py

קובץ זה אחראי על ביצוע ההדפסות למשתמש. ישנו קובץ מיוחד המטפל בהדפסות מכיוון שסיווגתי את ההדפסות שלושה סוגים:

1. הדפסות המבקשות קלט – יודפסו בצבע **ירוק**
2. הדפסות שנותנות אינפורמציה על המתרחש בזמן ריצה – יודפסו בצבע **כחול**
3. הדפסות של הודעות שגיאה – יודפסו בצבע **אדום**

כל אחת מן הפונקציות הללו מקבלת הודעה להציג למשתמש כקלט וקובעת צבע הדפסה שונה (את הודעות אלו נכיר בפרק המדריך למשתמש).

```
from colorama import init, Fore, Back, Style
```

```
def printError(message):  
    init(convert=True)  
    print(Fore.RED + message)  
    Style.RESET_ALL
```

```
def printOptions(message):  
    init(convert=True)  
    print(Fore.GREEN + message)  
    Style.RESET_ALL
```

```
def printProcess(message):  
    init(convert=True)  
    print(Fore.BLUE + message)  
    Style.RESET_ALL
```

מסקנות הרצת המודל:

לפני שאנתח את תוצאות המודל ואתאר את הניסיונות שהביאו לתוצאות הסופיות אסביר מספר מושגים נדרשים:

Data set – מאגר כל התמונות. נשתמש במאגר ממיון של התמונות לפי קטגוריות ואת מאגר זה נחלק לשני תתי מאגרי מידע: train and test/validation.

Train data – התמונות אותן המודל לומד. מאגר זה גדול מן מאגר ה test (אצלי 80% מסך כל התמונות).

Test\ validation data – התמונות אותן המודל אינן לומד, אך הצלחתו לזהות נכון את הקטגוריות אליהן שייכות תמונות אלו קובעת אם המודל אכן למד את התמונות מן מאגר ה train.

Accuracy – אחוז ההצלחה של המודל בחיזוי הקטגוריות של התמונות.

Loss – בניגוד ל accuracy אינו פועל באופן בינארי (חזה נכון או לא – 0/1) אלא קובע כמה קרובות היו תוצאות חיזוי התמונות, לקטגוריה האמתית שאליה שייכים.

לכל אחד מן שני חלקי המאגר יש ערכי accuracy ו loss. כך ניתן לבחון האם המודל אכן מבצע למידה ואם כן, כמה הוא מצליח.

ישנן שתי בעיות אשר יכולות להיווצר אשר ניתן להסיק אותן מן הערכים הללו: overfitting ו under fitting.

Overfitting – "התאמת יתר" היא מצב בו המודל מותאם יתר על המידה למאגר אותו הוא לומד (אצלנו מאגר תמונות) ופחות מצליח בביצוע תחזיות של המאגר אותו הוא לומד. אנו נזהה מצב זה כאשר ה validation accuracy גבוה משמעותית מן ה accuracy או כאשר ה validation loss קטן משמעותית מן ה training loss.

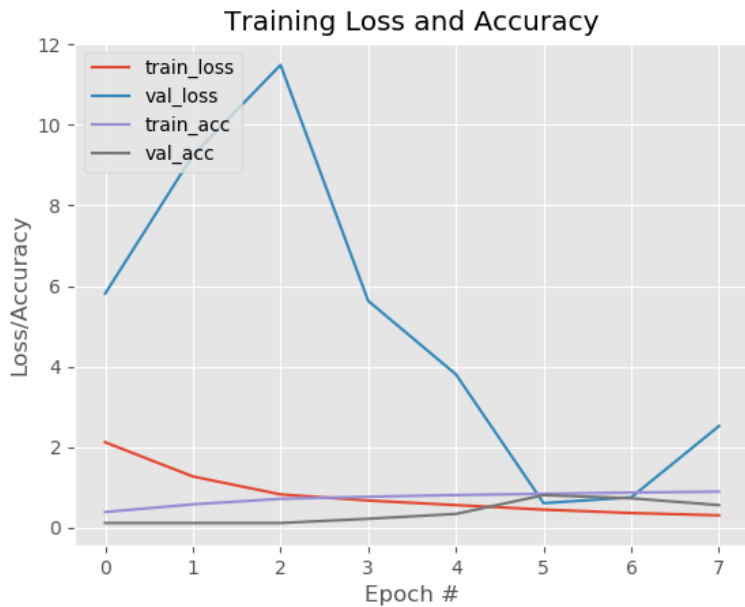
Under fitting – מצב ההפוך מ overfitting, אשר בו מאגר הלמידה הינו פשוט מידי ואינו כולל מספיק תמונות שונות או כאשר אין בתמונות מספר מספיק את תכונות הלמידה. במצב זה המודל אינו מצליח ללמוד את התמונות שכן הוא בלתי אפשרי ללמידה. נזהה מצב זה כאשר ה validation accuracy נמוך משמעותית מן ה training accuracy או כאשר ה validation loss גבוה משמעותית מן ה training loss.

Num epoch – מספר הפעמים בהם יתבצע תהליך הלמידה מחדש.

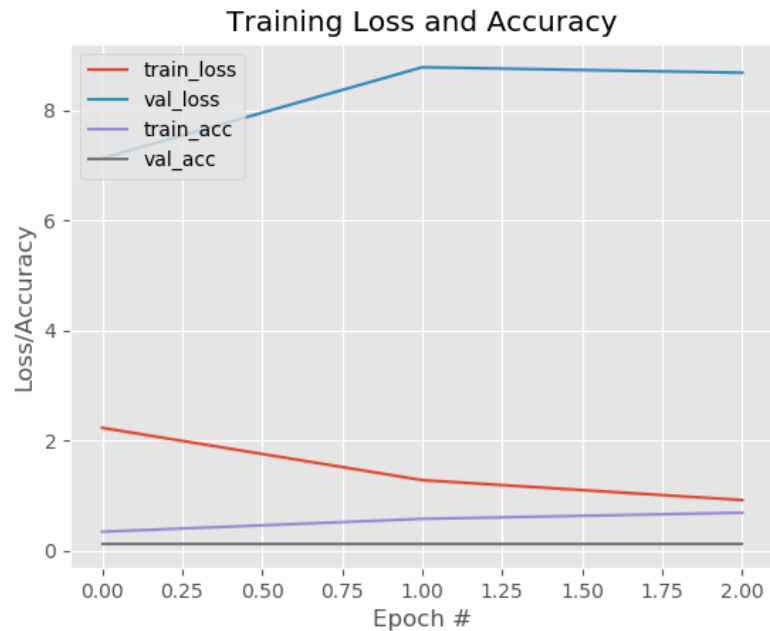
בפרויקט שלי נתקלתי בבעיה של under fitting. בעיה זו נוצרה מכך שמאגר התמונות שלי הכיל תמונות די דומות אשר אינן אפשרו למידה. חשוב לציין שאני יצרתי את מאגר התמונות בעצמי באמצעות חיתוך פריימים מן סרטונים שצילמתי. ולמעשה התדירות הגבוהה בה חתכתי פריימים יצרה מצב בו התמונות דומות יתר על המידה ואינן מאפשרות למידה.

להלן חלק מגרפי התוצאות של הלמידה מן מאגר התמונות שיצרתי:

גרף 2



גרף 1 (מתאר את התחלת הלמידה)



ניתן להבחין שלא התבצעה בהרצה זו למידה שכן ה validation loss גבוה משמעותית מן ה training loss (גם לאחר שירד שב למגמת עליית) וגם ה- validation accuracy היה נמוך משמעותית מן ה training accuracy (עלה מעט ושב לרדת).

תחילה לא הבנתי שהבעיה הינה במאגר וניסיתי להעלות את מספר ה epoch המתבצע או לשנות מעט את המודל, להגדיל את המאגר אך ללא הועיל. כשהבנתי שהבעיה היא במאגר שאינו מאפשר למידה עלה במוחי רעיון. יצרתי מן הסרטונים שיצרתי פחות תמונות כדי לצמצם את מספר התמונות הדומות והוספתי למאגר הנתונים שני מאגרי נתונים נוספים מן האינטרנט ובכך יצרתי מאגר אחד גדול יותר המכיל תמונות שונות יותר.

המאגר הראשוני כלל בתוכו 4400 תמונות ואחר יצרתי מאגר כפול של 8800 אך לבסוף המאגר הסופי כולל כ 12,000 תמונות. הדבר אכן פתר את בעיה ה Under fitting והביא לקבלת התוצאות שלהלן:



ניתן לראות שה accuracy וה loss של שני חלקי המאגר התקדמו מ epoch ל epoch באופן דומה, כאשר אין הבדלים משמעותיים ביניהם.

ההרצה המוצגת כאן הניבה תוצאות סופיות של: validation accuracy 89%, training accuracy 94%, validation loss 0.14 ו training loss 0.33.

```
[INFO] Training network...
Epoch 1/10
230/230 [=====] - 71s 308ms/step - loss: 2.0921 - accuracy: 0.4025 - val_loss: 4.0781 - val_accuracy: 0.1043
Epoch 2/10
230/230 [=====] - 67s 291ms/step - loss: 1.2107 - accuracy: 0.6016 - val_loss: 1.0273 - val_accuracy: 0.5644
Epoch 3/10
230/230 [=====] - 52s 226ms/step - loss: 0.8741 - accuracy: 0.7079 - val_loss: 0.6634 - val_accuracy: 0.7550
Epoch 4/10
230/230 [=====] - 52s 225ms/step - loss: 0.6807 - accuracy: 0.7642 - val_loss: 2.3365 - val_accuracy: 0.4910
Epoch 5/10
230/230 [=====] - 52s 226ms/step - loss: 0.5684 - accuracy: 0.8022 - val_loss: 0.2261 - val_accuracy: 0.9207
Epoch 6/10
230/230 [=====] - 52s 225ms/step - loss: 0.4895 - accuracy: 0.8320 - val_loss: 0.3174 - val_accuracy: 0.8751
Epoch 7/10
230/230 [=====] - 52s 226ms/step - loss: 0.4638 - accuracy: 0.8464 - val_loss: 0.1812 - val_accuracy: 0.9321
Epoch 8/10
230/230 [=====] - 52s 226ms/step - loss: 0.3910 - accuracy: 0.8674 - val_loss: 0.1595 - val_accuracy: 0.9419
Epoch 9/10
230/230 [=====] - 52s 226ms/step - loss: 0.3644 - accuracy: 0.8831 - val_loss: 0.1322 - val_accuracy: 0.9598
Epoch 10/10
230/230 [=====] - 52s 225ms/step - loss: 0.3336 - accuracy: 0.8906 - val_loss: 0.1406 - val_accuracy: 0.9484
```


רפלקציה/סיכום אישי:

ביצוע פרויקט זה אינו היה מטלה פשוטה כפי שציינתי בפרקי המבוא ומסקנות הרצת המודל. באופן כיתתי נדרשנו ללמוד את הנושא באופן די מהיר ובכוחות עצמנו (יחד עם עזרתה של דינה כמובן), ובאופן אישי נתקלתי בקושי בהבנת הבעיה מולה אני עומד לאחר הרצות הראשוניות של המודל (under fitting) ובמציאת הפתרון. אך, בסיכומי של דבר נהייתי מאוד מביצוע פרויקט זה. בעיני התחום של Deep Learning הינו מרתק ובעל כוח להביא לחידושים טכנולוגיים אדירים. אני גאה בתוצר הסופי שהצלחתי להשיג.

חשוב לי להוסיף שעשיית הפרויקט ליכדה את חברי הכיתה כולה שכן עבור כולנו הייתה מטרה משותפת – להבין את הנושא עליו אנו נדרשים לעשות פרויקט, ואני חושב שהעזרה ההדדית והאחוזה שנוצרה בין חברי כיתתי היא התוצר הטוב ביותר שהתקבל מעשיית הפרויקטים תחת נושא זה.

ביבליוגרפיה:

pyimagesearch

<https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deep-learning/>

<https://www.pyimagesearch.com/2019/12/30/label-smoothing-with-keras-tensorflow-and-deep-learning/>

analyticsvidhya

<https://www.analyticsvidhya.com/blog/2019/04/build-first-multi-label-image-classification-model-python/>

The keras blog

<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>

Medium

<https://medium.com/@manasnarkar/image-classification-with-convolution-neural-networks-cnn-with-keras-dbd71c05ed2a>

<https://towardsdatascience.com/image-classification-python-keras-tutorial-kaggle-challenge-45a6332a58b8>

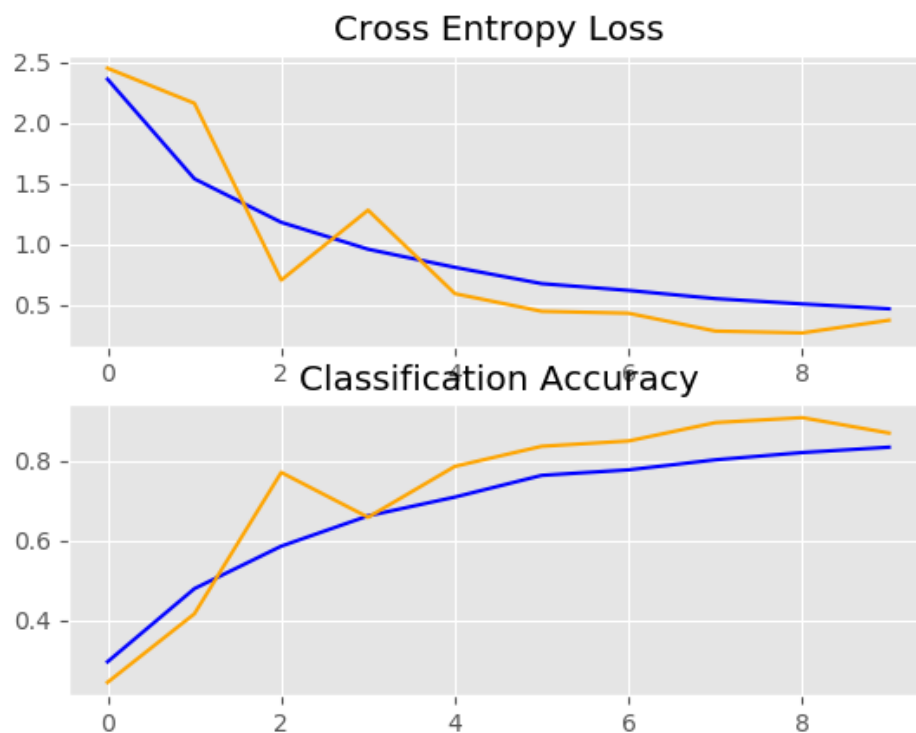
ספרים:

Deep_learning_for_computer_vision

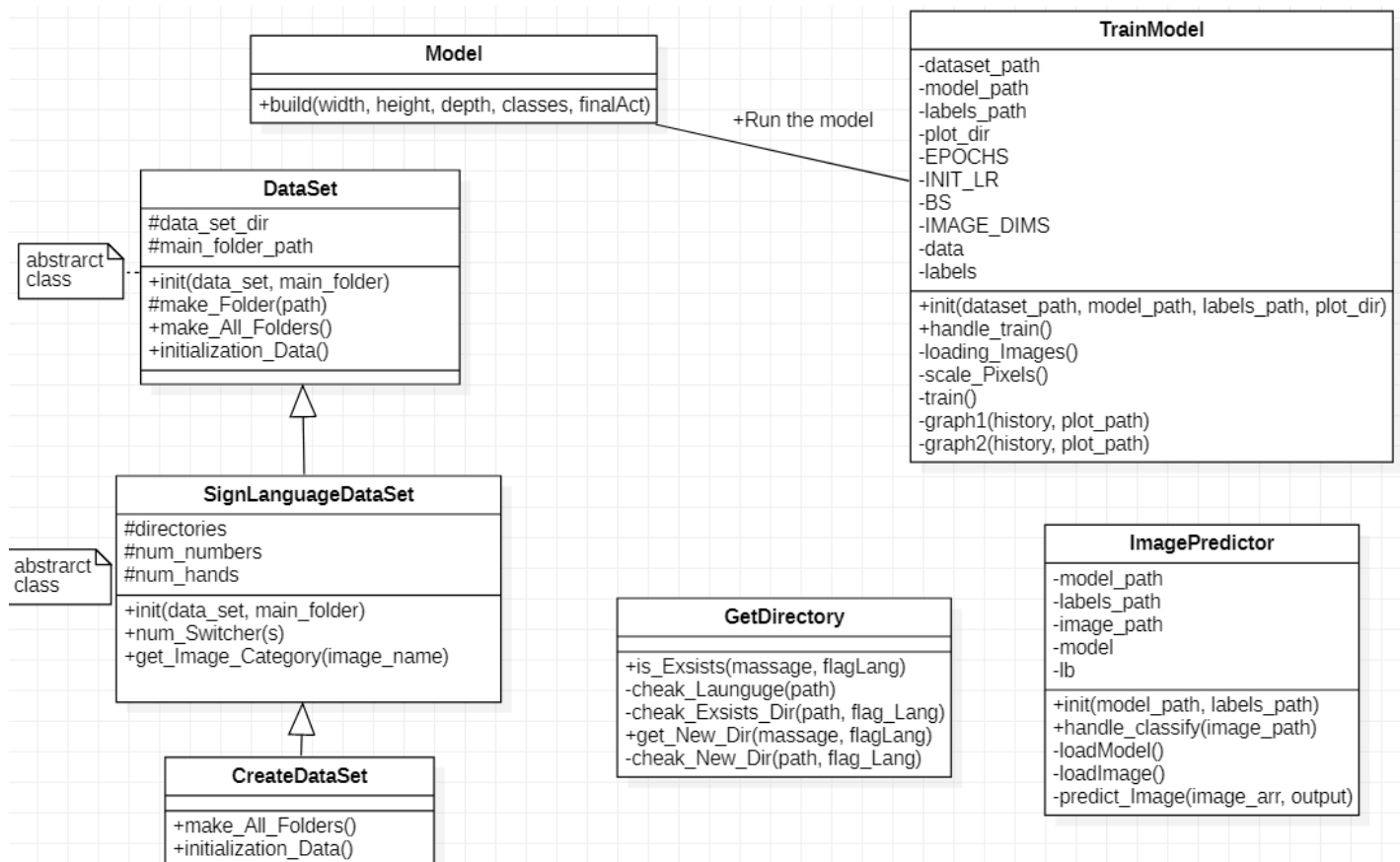
Practical Python and OpenCV, 4th Edition

נספחים:

גרף נוסף של תוצאות הרצת המודל (בגרף זה ערכי ה loss וה accuracy מופיעים בנפרד).



UML class diagram



UML Actors diagram

