

ספר פרויקט 5 יחידות למידת מכונה-

Sign Language's digits identifier



שם המגישה: עמית כספי

תעודת זהות: 214302564

שם בית הספר: מקיף י"א ראשונים ראשון לציון

כיתה: י"ב – 4

שם המורה: דינה קראוס

תאריך הגשה: 19.6.2020

תוכן עניינים:

2-5.....	מבוא
6-20.....	מדריך למשתמש
21-36.....	מדריך למפתח
37	רפלקציה וסיכום אישי
38-39.....	מסקנות מהרצת המודל
40.....	ביבליוגרפיה
41-43.....	נספחים

מבוא:

במסגרת לימודי בתיכון מקיף י"א ראשונים החלטתי ללמוד במגמת הנדסת תוכנה ולהגביר 5 יחידות לימוד בהתמחות – תכנון ותכנות מערכות Deep Learning- למידה עמוקה, אשר כחלק ממנה בחרתי לעבוד על פרויקט גמר בנושא Sign Language's digits identifier.

למידה עמוקה היא למעשה תת תחום בתוך שדה שנקרא למידת מכונה Machine Learning, הנמצא תחת תחום הגג הגדול של בינה מלאכותית. למידה עמוקה, היא מודל חישובי המכיל קבוצת אלגוריתמים, ונשען על תשתית של רשתות נוירונים מלאכותיים (פרספטרונים) המחקים את תהליך הלמידה במוח האנושי, אשר מורכב מרשת נוירונים. ברשת הנוירונים שבמוח האנושי נקלט המידע ובעזרת התקשורת שבין כל תא עצב לאלו שבקרבתו בני האדם מסוגלים לחשוב וכן להיות בעלי יכולת ללמוד דברים חדשים ולזכור את מה שלמדו. התחום, שעתידי להפוך לחלק אינטגרלי מתעשיית ההיי-טק העולמית, מצליח לא כי הוא דמיוני, אלא להפך, מכיוון שהוא מייצר פתרונות לבעיות ולתהליכים שגרתיים ויומיומיים: מזיהוי כתב, זיהוי דיבור בסמארטפונים, ועד לשדרוג של חלקים מעולם הרפואה וההתקדמות לעבר תחבורה אוטונומית, תעשייה יעילה וחכמה יותר וכלים פיננסיים.

בפועל, הלמידה העמוקה במחשבים נועדה לחקות כמה שיותר את פעולת הלמידה במוח האנושי, והיא נעשית באמצעות עיבוד של מידע (דאטה) רב ברשתות נוירונים מלאכותיים בצורה סימולטנית, תוך אימון של המערכת. כל שכבה עושה עוד עיבוד על השכבה שלפני, והכל ביחד מבצע חישוב מורכב. המטרה של הטכנולוגיה היא לזהות דפוסים חוזרים, מתוך כמויות גדולות של דאטה שהוזנו לתוכה.

המוטיבציה שלי לפרויקט הינה גבוהה ביותר שכן החלטתי לבחור בפרויקט בנושא זיהוי ספרות בשפת הסימנים לאחר שנתקלתי באנשים רבים בחיי בעלי מוגבלויות שונות וכן באנשים דוברי שפת הסימנים (למשל חרשים) הנתקלים בבעיות רבות בחייהם ובבעיות תקשורת עם העולם והאנשים שמסביבם, על כן לפעמים אנשים אלו יכולים להתפס כזרים ומשונים, ולעיתים אף לא מובנים, אך בעיניי אין כל סיבה שכך יהיה הדבר, וכאמור- ניתן לחשוב על פתרונות יצירתיים לאתגרים אלו ולסיוע בתקשורת בין אנשים שונים. על כן, החלטתי לנסות לחשוב על דרך לסייע ולתרום לתקשורת זו וכך קיבלתי את הרקע לרעיון הפרויקט שלי - זיהוי ספרות בשפת הסימנים, שאני מאמינה בחשיבותו הרבה גם בהיבט המעשי וגם בהיבט התיאורטי- כל אחד שונה בדרכו, לא צריך להיבהל מהשונה אלא ללמוד להכיר ולהבין את השונה, ועל כן המוטיבציה הרבה שלי לעשיית הפרויקט.

בפרויקט שבחרתי- המזהה ספרות בשפת הסימנים, בחרתי לכתוב תוכנה אשר לאחר תהליך למידה משמעותי תוכל לנתח תמונות שונות המייצגות ספרות בשפת הסימנים ולזהות מהי הספרה המיוצגת בתמונה. בחרתי לבצע פרויקט זה כדי ליצור את התשתית הבסיסית שתסייע לתקשורת של 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- ספרות מ0 עד 9. תוכנה זו למעשה מתרגמת את סימוני הידיים המבטאים ספרות בשפת הסימנים לספרה מספרית (חשוב להדגיש כי הספרות בשפת הסימנים החל מן הספרה חמש אינן זהות לספירה שרבים רגילים

אליה, וכן כי כל הספרות בשפת הסימנים מ-0-9 מתבטאות בסימונים באמצעות יד אחת בלבד). בעתיד פיתוח תוכנה זו והרחבת אוצר מילים הניתן לזיהוי תעצים ותחזק את הפוטנציאל הגלום תחת פרויקט זה ואת השפעתו המשמעותית והחיובית בעולם.

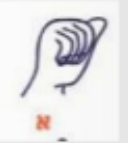
במסגרת כתיבת הפרויקט נתקלתי בקשיים ובאתגרים רבים. ראשית, על מנת לבצע הליך למידה משמעותי אשר ייתן תוצאות טובות יש צורך במאגרי תמונות עצומים של כאלפי תמונות המתארות ספרות בשפת הסימנים. תחילה נבהלתי מגודל הdata שהתבקשתי להשיג, וכן תהיתי האם המחשב שלי (יחסית ישן) יוכל להכיל כמות כזו של מידע. אך לאחר שפניתי מקום במחשב וכן לאחר חיפוש רבים ברשת האינטרנט הצלחתי למצוא ולהשיג data set ראוי שיתאים לצרכי הפרויקט, שכן ראיתי את החשיבות הגדולה שבו ועל כן בחרתי שלא לוותר על נושא זה אשר זיהיתי בו את היכולת לתרום ולהוות מעבר לפרויקט שממנו אלמד גם פרויקט שימושי בעל חשיבות גבוהה.

בנוסף לכך, התחום של למידה עמוקה Deep Learning והרעיונות העומדים מאחוריו, היו חדשים, שונים וזרים לי. לכן, למידת הנושא, בסיס התחום והרעיונות העומדים בו במקביל ותוך כדי כתיבת הפרויקט- היוו בעבורי אתגר משמעותי. על מנת להתגבר על מכשול זה, נעזרתי במורתי דינה, בחברים, ובעיקר מחומרים שמצאתי באינטרנט, מדריכים, ספרים שונים, סרטונים מיוטיוב ואף מפורומים שונים וניסיונות של אנשים (אציין כי אחד הדברים האהובים עליי בשפת python הוא העובדה שהשפה הזו כל כך נפוצה ומוכרת ברחבי האינטרנט כך שכמעט כל קושי או שגיאה שנתקלתי בה במהלך הפרויקט- אנשים אחרים נתקלו בה לפניי ובכך הצלחתי ללמוד מניסיונם של אחרים ולהתגבר על שגיאות רבות), ובכך למדתי את הנושא והתגברתי על אתגרים רבים כחלק מתהליך העבודה על הפרויקט וכתיבתו.

על רעיונות אלו מתבססים פרויקטים רבים בתחום ובשימושים שונים, ואף הפרויקט שלי, המזהה ספרות בשפת הסימנים.

כמה עובדות על שפת הסימנים!

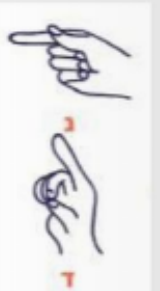
שפת הסימנים אינה שפה אוניברסלית



שפת הסימנים אינה פנטומימה



שפת הסימנים היא שפה דינמית ומתפתחת



כל אחד יכול ללמוד את שפת הסימנים!

מניפים את היד באוויר
לשני הצדדים.



שלום



מניפים את היד באוויר
לשני הצדדים

אהבה



מסמנים את הסימן על גוף המין.

מסמנים את הסימן עם
הידיים על החזה.



מה שלומך?



היד ממוקמת במקביל ללב
מסמנים בתנועה סיבוב
פנימה.

היד ממוקמת במקביל ללב
מסמנים בתנועה סיבוב
פנימה.



שמח

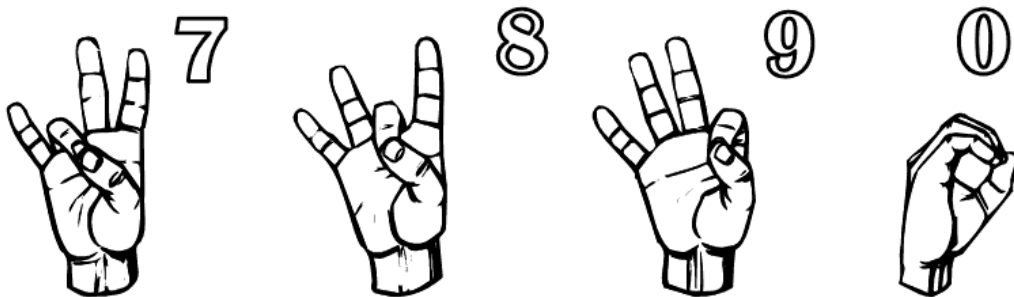


מזיזים את שתי הידיים
מעלה ומטה בתנועות
מהירות שוב ושוב.

מזיזים את שתי הידיים
מעלה ומטה בתנועות
מהירות שוב ושוב.



מקרא ספרות 0-9 בשפת הסימנים:



מדריך למשתמש:

מדריך למשתמש- כיצד ניתן להשתמש בתוכנית?

הוראות התקנה:

1. יש להוריד Python 3.7 או גרסה עדכנית יותר -

<https://www.python.org/downloads/>

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

```

C:\> שורת הפקודה
Microsoft Windows [Version 10.0.19041.1052]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amit>python -V
Python 3.7.4
  
```

2. יש להתקין במחשב את סביבת העבודה PyCharm (או אחרת לבחירתכם).

לינק להורדת PyCharm - <https://www.jetbrains.com/pycharm>

3. יש להוריד מספר ספריות קוד אשר הפרויקט משתמש בהן:

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
https://pypi.org/project/Kivy/	pip install Kivy	kivy

4. יש להוריד מן חשבון ה GitHub שלי את הקבצים הבאים ובאופן הארגון הבא:

קישור לחשבון Github שלי - <https://github.com/amit-caspi>

- יש ליצור ולרכז בתיקיה אחת ולהוריד אליה: את כל קבצי הpython (9) עליהם מתבסס המודל ובנוי הפרויקט + להוריד אליה את קובץ ה kivy (קובץ בסיומת kv. אשר הספרייה של kivy יודעת לעבוד מול הפורמט של הקובץ לצורך GUI) + לשמור בתיקיה זו את 6 התמונות המשמשות את הGUI שבפרויקט לפי שמן של התמונות (Submit, Predict, TakeImage, Train, Random, GoBack).

- יש ליצור תיקייה נוספת ולהוריד אליה: את התיקייה המכילה את מאגר התמונות + את התיקייה המכילה את מאגר התמונות באופן ממזין + את תיקיית התמונות החדשות ובה menu תמונות חדשות שאינן ממאגר התמונות שהמחשב לומד + את קובץ המודל השמור + את קובץ התוויות (ניתן לשמור קבצים אלו לפי בחירה אישית במחשב במפוזר אך אמליץ שלא).

אין לשנות את תוכן הקבצים!

כל התיקיות והקבצים בפרויקט יאוכלסו בdirectories שאינם מכילים אותיות עבריות!

5. יש לעדכן את הקוד בהתאם לdirectories החדשים לפי המחשב: ישנן שתי אפשרויות להרצה שעליהן יפורט בהמשך-
 - אפשרות ההרצה הראשונה ללא GUI: בפונקציית ה menu() בקובץ menu.py יש לשנות את ערכי המשתנים המחזיקים את הכתובות הרלוונטיות-
 - data_set = directoryn של מאגר כלל התמונות הלא ממזין.
 - sorted_data_set = directoryn של מאגר התמונות הממזין בתיקיות לפי ספרות מ0 עד 9 כולל.
 - model_path = directoryn של המודל השמור.
 - labels_path = directoryn של קובץ התוויות של התמונות.
 - new_images_folder = directoryn של תיקייה ובה תמונות חדשות שהמודל אינו נחשף אליהן בתהליך הלימוד (לא בtest ולא בtrain).
- תמונה להמחשה של הפונקציה, כאשר יוכנסו הנתיבים במקום ה-XXXX לפי הכתובות הרלוונטיות במחשב-

```

30
31 def menu():
32     flag = True
33     options()
34     """
35     default directories
36     this directoties change according to the user activity
37     """
38
39     data_set = r"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
40     sorted_data_path = r"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
41     model_path = r"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
42     labels_path = r"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
43     new_images_folder = r"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
44

```


- אפשרות ההרצה השנייה עם מסך GUI: בקובץ newApp.py יש לשנות את ערכי המשתנים המחזיקים את הכתובות הרלוונטיות-

`global_data_set = directory` של מאגר כלל התמונות הלא ממוין.
`global_sorted_data_set = directory` של מאגר התמונות הממוין בתיקיות לפי ספרות מ 0 עד 9 כולל.
`global_model_path = directory` של המודל השמור.
`global_labels_path = directory` של קובץ התוויות של התמונות.
`global_new_images_folder = directory` של תיקייה ובה תמונות חדשות שהמודל אינו נחשף אליהן בתהליך הלימוד (לא בtest ולא בtrain).

תמונה להמחשה, כאשר יוכנסו הנתיבים במקום ה-XXXX לפי הכתובות הרלוונטיות במחשב-

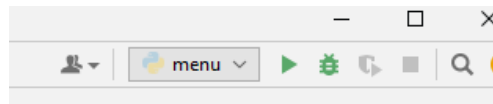
```

18 import os
19 import random
20
21 global_data_set = r"XXXXXXXXXXXXXXXXXXXX"
22 global_sorted_data_path = r"XXXXXXXXXXXXXXXXXXXX" #sorted data direction
23 global_model_path = r"XXXXXXXXXXXXXXXXXXXX" #model direction
24 global_labels_path = r"XXXXXXXXXXXXXXXXXXXX" #Labels (binari) direction
25 global_new_images_folder = r"XXXXXXXXXXXXXXXXXXXX"
26
27 class MainWindow(Screen):
28     pass
  
```

מדריך למשתמש- כיצד ניתן להריץ את הפרויקט?

הוראות להרצת התוכנית- אפשרות 1 דרך הקובץ menu.py בסביבת העבודה PyCharm וללא GUI:

על מנת להריץ את התוכנית יש להשתמש בסביבת העבודה PyCharm- כלומר יש להריץ את קובץ ה Python: **menu.py** באמצעות לחצן ההרצה בתוכנה מצד ימין למעלה (משולש ירוק)-



הרצת קובץ זה תריץ את כל התוכנית. (יש לציין כי הדוגמאות וההסברים שיוצגו בספר זה הן לפי הרצה דרך Anaconda Prompt של סביבת העבודה Anaconda).

אחר הרצת התוכנית יוצג התפריט הבא:

```
Anaconda Prompt - python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py
(base) C:\Users\ilano>python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py
Using TensorFlow backend.
2021-06-18 12:47:58.316001: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'C:\Windows\System32\user32.dll'; dlerror: user32.dll not found
2021-06-18 12:47:58.316225: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror
*****
*          USER INTERFACE          *
*          *                       *
* Enter 1 --> train the model      *
* Enter 2 --> predict an image     *
* Enter 3 --> predict random images *
* Enter 4 --> take an image        *
* Enter space bar --> exit         *
*          *                       *
*****
--> Your Choice:
Enter: _
```

מוצג למשתמש תפריט מסודר המציג למשתמש את האפשרויות העומדות לפניו לרשותו ואת הקלט שיש להזין עבור כל אפשרות שיחליט לבחור-

מוצעות חמש אפשרויות:

הכנס 1 כדי לאמן את המודל

הכנס 2 כדי לחזות את הקטגוריה של תמונה שהמשתמש ישלח כקלט (נתיב)

הכנס 3 כדי לחזות את הקטגוריה של תמונה שתבחר באופן רנדומלי

הכנס 4 כדי לצלם תמונה חדשה ולשמור אותה במחשב (אם קיימת מצלמה במחשב)

הכנס רווח ליציאה מן התוכנית (יש להקיש רווח ואחרי Enter)

עבור כל קלט אחר התוכנית תמשיך לבקש קלט.

המשתמש יכול לבצע חיזוי ללא אימון- התוכנה תשתמש בקבצים שבהם שמורים הנתונים אותם המשתמש התקין במחשבו מראש (באמצעות default directories) ולפי ההוראות שהצגתי קודם לכן (קובץ שבו נשמר המודל וכן הלאה). יש לציין כי התוכנה 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).

מעבר על ארבעת האופציות השונות באפשרות ההרצה הראשונה (ללא GUI):

יש לציין כי:

הדפסות בצבע ירוק – מנחות את המשתמש בבקשות הקלט.

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

הדפסות בצבע אדום – מסמלות שגיאה של קלטיים אי תקינים.

יש לציין כי ישנן בדיקות תקינות קלט לאורך כל התוכנית!

- במצב בו נכניס directory שאינו קיים נקבל את הודעת השגיאה **Error - directory is not valid** – כלומר שהכתובת ששלחנו כקלט אינה תקינה והתוכנה מבקשת מאתנו קלט חדש.
- במצב מסוים בו ננסה להכניס כתובת אשר מכילה אותיות עבריות (אשר לאחר ניסוי ותהייה גיליתי כי בשלב טעינת תמונה באמצעות opencv ישנה שגיאה כשאר ה directory של התמונה מכיל אותיות עבריות) נקבל את הודעת השגיאה **Error – path contains Hebrew letters** האומרת שהכתובת מכילה אותיות עבריות והתוכנה מבקשת מאתנו קלט חדש.
- במצב בו נשלח כקלט directory אשר הינו אכן קיים אך ללא שם התיקייה החדשה, נקבל שגיאה האומרת ש directory זה הינו קיים, משמע הוא אינו חדש, אך אינו כולל את שם התיקייה **Error – this directory is allready exists** והתוכנה מבקשת מאתנו קלט חדש.

אופציה ראשונה- אימון המודל:

```
Anaconda Prompt - python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py

(base) C:\Users\ilano>python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py
Using TensorFlow backend.
2021-06-18 12:47:58.316001: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_11.0.dll'; this might be because the library is not placed in the expected location relative to the module.
2021-06-18 12:47:58.316225: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dl
*****
*          USER INTERFACE          *
*          *                       *
* Enter 1 --> train the model       *
* Enter 2 --> predict an image     *
* Enter 3 --> predict random images *
* Enter 4 --> take an image        *
* Enter space bar --> exit         *
*          *                       *
*****
--> Your Choice:
Enter: 1
Enter the full path (with the name) to output model:
Enter: C:\Users\ilano\OneDrive\Desktop\amitModel.model
Enter the full path (with the name) to output label binarizer:
Enter: C:\Users\ilano\OneDrive\Desktop\amitLB.pickle
Enter the folder directory to output accuracy/loss plot:
Enter: C:\Users\ilano\OneDrive\Desktop\MyGraphs
[INFO] Loading images...
```

יוצגו בפני המשתמש הנחיות לבקשת קלטים שונים אחת אחרי השנייה, כאשר המשתמש יראה את הבקשה להזין את הקלט הבא רק לאחר שהזין קלט תקין.

לאחר שהמשתמש הקיש 1-

הקלט ה1 שמתבקש הינו ה directory המלא עם שם הקובץ החדש שיווצר שבו נרצה לשמור את המודל לאחר ביצוע האימון, כאשר סיומת הקובץ חייבת להיות model.

לשם הדוגמה מכניסים את ה directory:

C:\Users\ilano\OneDrive\Desktop\amitModel.model

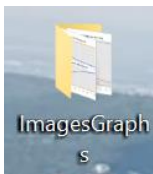
הקלט ה2 שמתבקש לאחר מכן הינו ה directory המלא עם שם הקובץ שיווצר שבו נשמור את קובץ התוויות הבינארי (קובץ זה ישמש לביצוע הלמידה ולחיזוי קטגוריה של תמונה בהמשך), כאשר סיומת הקובץ חייבת להיות pickle או txt.

יש לציין כי בחלק זה ישנה בדיקת קלט נוספת והודעת שגיאה אפשרית. אם נכניס את אותו ה directory של הקובץ הקודם אחד הקבצים ידרוס את השני או שהתוכנית עלולה לקרוס, ולכן כדי למנוע זאת נמנעה מהמשתמש אפשרות זו. למשל, אם כעת ננסה להכניס קלט את ה directory הקודם שהזנו נקבל הודעת שגיאה ובקשה לקלט מחדש:

Error - file will be override
Enter the full path (with the name) to output label binarizer:

הקלט ה3 שמתבקש הינו ה directory של תיקייה חדשה שתיווצר ובה ישמרו גרפי תוצאות הלמוד (כקבצי png).

לאחר הכנסת כל הקלטים מתחיל לרוץ תהליך האימון ומוצגים לפנינו פלטי אינפורמציה על תתי התהליכים המתרחשים אחד אחרי השני לביצוע האימון, אשר לוקח זמן. 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. בסיום האימון אנו מקבלים הודעה כי כעת השימוש יתבצע בקובץ המודל החדש ולא בדיפולטיבי שהוגדר קודם לכן (נבחין כי directories שהזנו אכן נותרו הקבצים בהתאמה וכי נוצרה תיקייה ובה שני קבצי תמונה שבהם גרפי מהלך האימון). לבסוף, אנו יכולים לבחור לבצע אפשרות חדשה.



(יש לציין כי תהליך האימון לוקח זמן- 45 דקות)

```
Anaconda Prompt - python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py

[INFO] Training network...
Epoch 1/15
120/120 [=====] - 96s 301ms/step - loss: 2.3657 - accuracy: 0.2969 - val_loss: 0.2799 - val_accuracy: 0.1826
Epoch 2/15
120/120 [=====] - 96s 283ms/step - loss: 1.5385 - accuracy: 0.4633 - val_loss: 2.1593 - val_accuracy: 0.3929
Epoch 3/15
120/120 [=====] - 94s 293ms/step - loss: 1.2294 - accuracy: 0.5647 - val_loss: 1.0572 - val_accuracy: 0.6998
Epoch 4/15
120/120 [=====] - 92s 287ms/step - loss: 1.0881 - accuracy: 0.6165 - val_loss: 0.9323 - val_accuracy: 0.6726
Epoch 5/15
120/120 [=====] - 93s 291ms/step - loss: 0.9558 - accuracy: 0.6686 - val_loss: 0.5857 - val_accuracy: 0.7928
Epoch 6/15
120/120 [=====] - 92s 287ms/step - loss: 0.8715 - accuracy: 0.6998 - val_loss: 0.6748 - val_accuracy: 0.7684
Epoch 7/15
120/120 [=====] - 91s 285ms/step - loss: 0.7929 - accuracy: 0.7158 - val_loss: 0.5723 - val_accuracy: 0.8736
Epoch 8/15
120/120 [=====] - 97s 304ms/step - loss: 0.7271 - accuracy: 0.7397 - val_loss: 0.7398 - val_accuracy: 0.7468
Epoch 9/15
120/120 [=====] - 95s 297ms/step - loss: 0.6714 - accuracy: 0.7597 - val_loss: 0.4831 - val_accuracy: 0.8533
Epoch 10/15
120/120 [=====] - 83s 268ms/step - loss: 0.6145 - accuracy: 0.7861 - val_loss: 0.4821 - val_accuracy: 0.8583
Epoch 11/15
120/120 [=====] - 87s 272ms/step - loss: 0.5661 - accuracy: 0.8089 - val_loss: 0.2689 - val_accuracy: 0.9025
Epoch 12/15
120/120 [=====] - 87s 273ms/step - loss: 0.5577 - accuracy: 0.8022 - val_loss: 0.3674 - val_accuracy: 0.8666
Epoch 13/15
120/120 [=====] - 88s 274ms/step - loss: 0.5058 - accuracy: 0.8246 - val_loss: 0.2708 - val_accuracy: 0.9052
Epoch 14/15
120/120 [=====] - 80s 258ms/step - loss: 0.4967 - accuracy: 0.8274 - val_loss: 0.3151 - val_accuracy: 0.8939
Epoch 15/15
120/120 [=====] - 82s 258ms/step - loss: 0.4658 - accuracy: 0.8408 - val_loss: 0.4967 - val_accuracy: 0.8338

# Evaluate on test data
253/2563 [=====] - 5s 2ms/step
test loss 0.49578377868822935 , test acc 0.8337885141372681
[INFO] Serializing network...
[INFO] Serializing label binarizer...
[INFO] Using trained model
--> Your Choice:
Enter:
```

אופציה שנייה- חיזוי תמונה:

```
Anaconda Prompt - python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py

(base) C:\Users\ilano>python C:\Users\ilano\OneDrive\Desktop\Amit\
Using TensorFlow backend.
2021-06-19 16:20:10.969425: W tensorflow/stream_executor/platform/
found
2021-06-19 16:20:10.969749: I tensorflow/stream_executor/cuda/cuda
*****
*          USER INTERFACE          *
*                                  *
* Enter 1 --> train the model      *
* Enter 2 --> predict an image     *
* Enter 3 --> predict random images *
* Enter 4 --> take an image        *
* Enter space bar --> exit         *
*                                  *
*****
--> Your Choice:
Enter: 2
Enter the image path:
Path Must be in English!
Enter: ????
Error - path contains hebrew letters
Enter the image path:
Path Must be in English!
Enter: 2
Error - no such file or directory
Enter the image path:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\num4_new.png
```

לאחר שהמשתמש הקיש 2-

הקלט 1 שמתבקש הינו ה directory המלא לתמונה שאותה הוא רוצה שהמודל המאומן יזהה. נבחין כי מאחר שמדובר בתמונה שיש לטעון עם הספרייה opencv ה directory בו שמורה התמונה חייב להיות באנגלית בלבד (אחד מדרישות תקינות הקלט) כאשר נכניס אותיות בעברית נקבל את הודעת השגיאה - **Error - path contains Hebrew letters**. בנוסף, הקלט צריך להיות תקין וקיים. אם יינתן directory שאינו קיים (למשל בדוגמה המוצגת- כשמכניסים את הספרה 2) נקבל הודעת שגיאה-

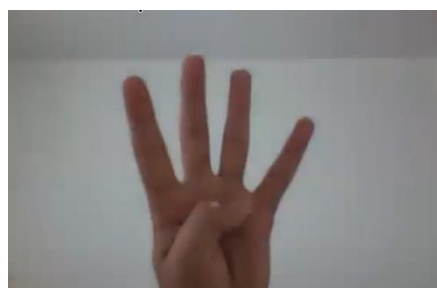
Error - no such file or directory.

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

למשל, נכניס את ה directory:

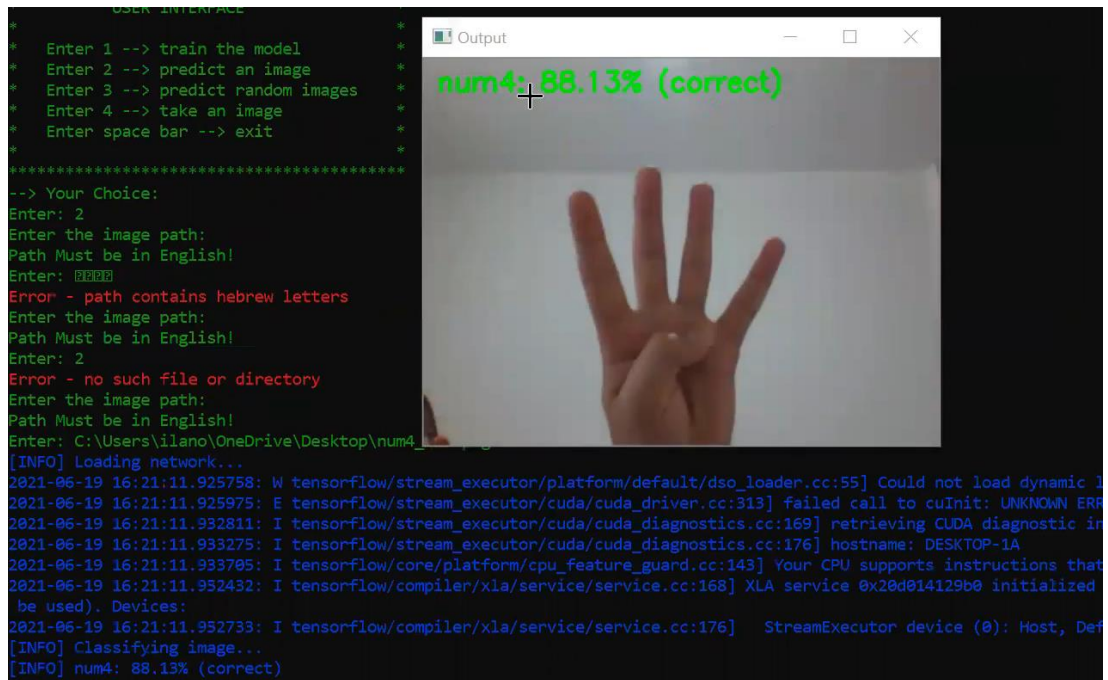
C:\Users\ilano\OneDrive\Desktop\num4_new.png

כתובת זו נמצאת התמונה הבאה אשר מסמלת בשפת הסימנים את הספרה 4:



נקבל מיד הודעה על כך שהמודל השמור נטען בידי התוכנית, ומיד לאחר מכן נקבל את הודעת אינפורמציה שהתוכנית 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).

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.



```

USER INTERFACE
* Enter 1 --> train the model
* Enter 2 --> predict an image
* Enter 3 --> predict random images
* Enter 4 --> take an image
* Enter space bar --> exit
*
*****
--> Your Choice:
Enter: 2
Enter the image path:
Path Must be in English!
Enter: 0000
Error - path contains hebrew letters
Enter the image path:
Path Must be in English!
Enter: 2
Error - no such file or directory
Enter the image path:
Path Must be in English!
Enter: C:\Users\ilano\OneDrive\Desktop\num4_1.jpg
[INFO] Loading network...
2021-06-19 16:21:11.925758: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'C:\Windows\System32\user32.dll'; falling back to other platform-specific dso loaders.
2021-06-19 16:21:11.925975: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-06-19 16:21:11.932811: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for all devices.
2021-06-19 16:21:11.933275: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: DESKTOP-1A
2021-06-19 16:21:11.933705: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX-512. To enable them, please compile TensorFlow with CPU flags enabled.
2021-06-19 16:21:11.952432: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x20d014129b0 initialized.
2021-06-19 16:21:11.952733: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default
[INFO] Classifying image...
[INFO] num4: 88.13% (correct)

```


אופציה שלישית- חיזוי תמונה רנדומלית:

```

*****
*                               *
*      USER INTERFACE          *
*                               *
*   Enter 1 --> train the model *
*   Enter 2 --> predict an image *
*   Enter 3 --> predict random images *
*   Enter 4 --> take an image  *
*   Enter space bar --> exit   *
*                               *
*****
--> Your Choice:

```

לאחר שהמשתמש הקיש 3-

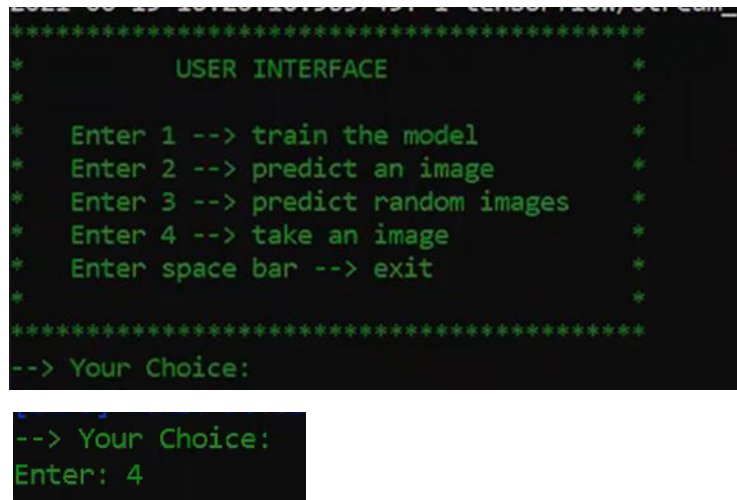
נקבל מיד הודעות אינפורמציה על כך שמתחיל תהליך החיזוי לתמונות שנלמדו והינן מהמאגר ועל כך שהמודל השמור נטען בידי התוכנית, ומיד לאחר מכן המשתמש מקבל הודעת אינפורמציה שהתוכנית כעת חוזרת את הקטגוריה שאליה שייכת התמונה הרנדומלית ולבסוף ניתן לראות שהמודל זיהה נכונה את התמונה ולקבל כפלט את סוג התמונה (0-9)-(בעזרת קובץ התוויות התוכנית אף יודעת לומר האם היא חזתה את הקטגוריה נכון או לא). בנוסף להדפסה זו, נפתח בפני המשתמש מסך חדש ובו התמונה הרנדומלית שזוהתה ועליה תוויות עם תוצאת החיזוי (מודפסת בירוק). רק לאחר שסיגור את החלון הנ"ל, התהליך יחזור על עצמו פעם נוספת כך שיזוהו חמשת תמונות רנדומליות ובסוף כל הזיהויים התוכנית תבקש מהמשתמש לבחור אופציה חדשה. יש לציין כי לאחר חיזוי של מספר תמונות רנדומליות מהמאגר ישנו מעבר לחיזוי תמונות חדשות רנדומליות שאינן מהמאגר.

```

--> Your Choice:
Enter: 3
[INFO] Learned images...
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num6: 99.84% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num5: 98.76% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num0: 100.00% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num7: 99.59% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num7: 72.61% (correct)
[INFO] New images...
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num6: 88.52% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num0: 100.00% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num2: 66.11% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num4: 96.44% (correct)
[INFO] Loading network...
[INFO] Classifying image...
[INFO] num1: 96.81% (correct)
--> Your Choice:

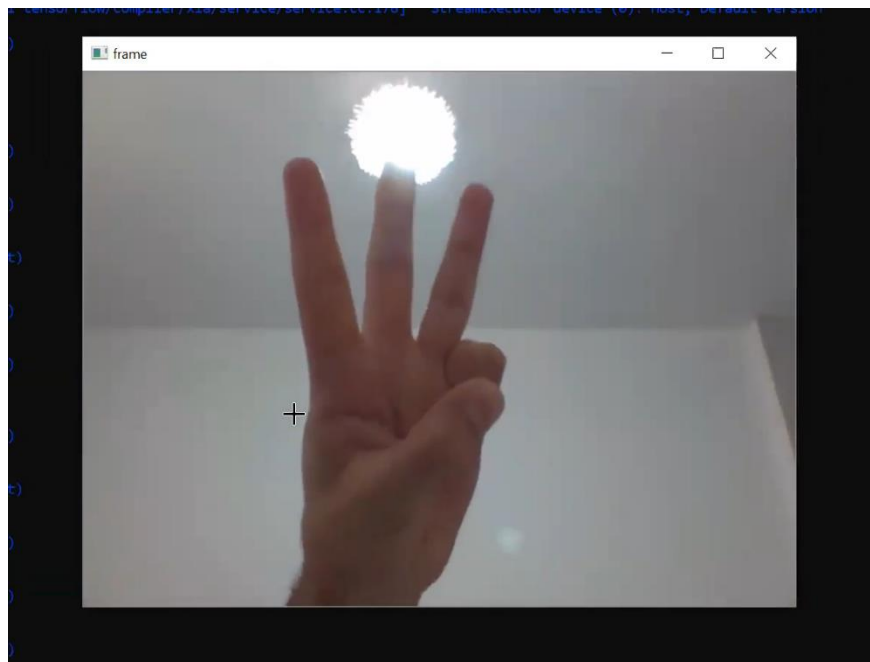
```

אופציה רביעית- צילום תמונה חדשה (במידה וקיימת מצלמה):



לאחר שהמשתמש הקיש 4-

נפתח מיד חלון חדש של מצלמה המאפשר למשתמש לצלם תמונה חדשה. בדוגמא זו המשתמש בוחר לצלם תמונה חדשה לסמן את הספרה 3 בשפת הסימנים. מיד לאחר מכן, יפתח למשתמש חלון חדש המאפשר לבחור מיקום במחשב ושם לקובץ התמונה החדש שנוצר. כלומר, יצרנו תמונה חדשה לבחירתנו ושמרנו אותה במחשב, וכעת ניתן למשל להשתמש בנתיב התמונה החדשה שנשמרה, לבחור באופציה 2 בתפריט ולבצע חיזוי של התמונה.



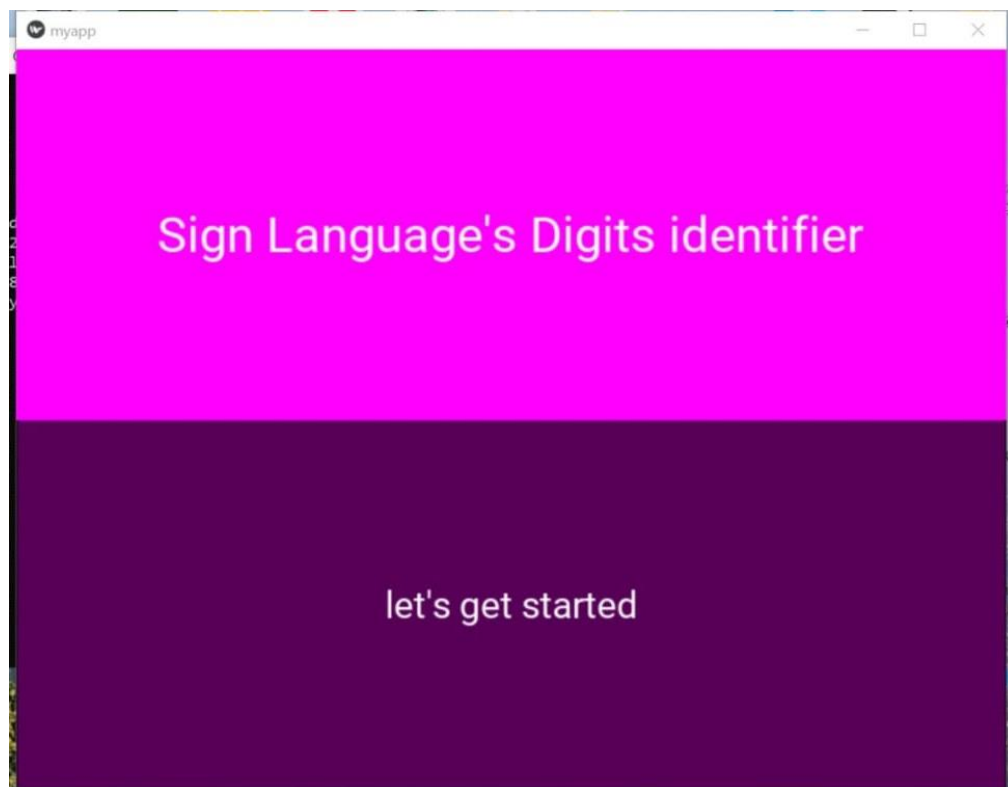
הוראות להרצת התוכנית- אפשרות 2 דרך הקובץ newApp.py בסביבת העבודה PyCharm ועם GUI:

ישנה אפשרות הרצה נוספת באמצעות GUI -Graphical User Interface-

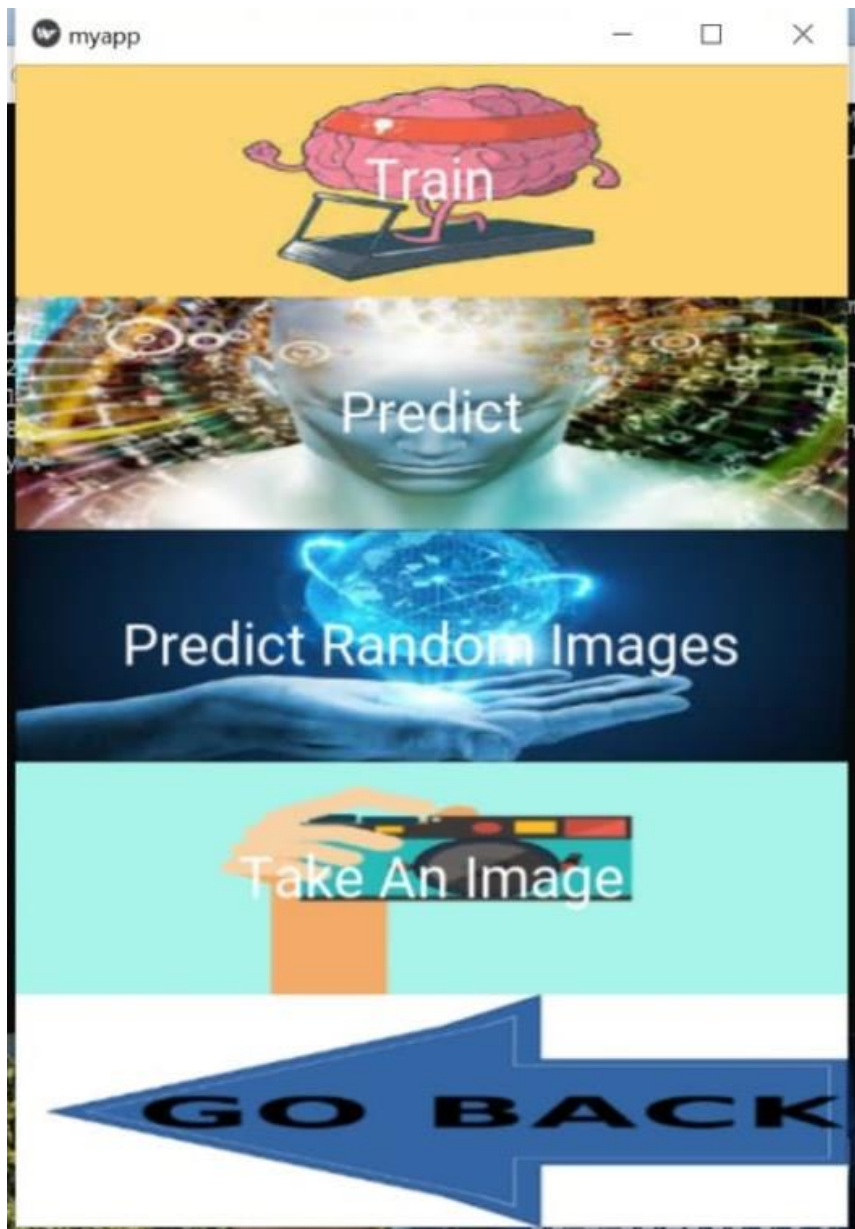
ממשק משתמש גרפי (באנגלית: GUI) הוא ממשק משתמש לתוכנה או לאתר אינטרנט, המבוסס על עיצוב גרפי (להבדיל מעיצוב טקסטואלי בלבד) של המסך המוצג למשתמש. מסוף שנות ה-80 נפוץ ממשק משתמש גרפי, שבו משלבים אלמנטים גרפיים כחלונות, צלמיות, פקדים כדוגמת תפריטים וכפתורים ואמצעי הצבעה. הממשקים הגרפיים מאפשרים תצוגה גמישה יותר, במיוחד בצבעים וגופנים, הבהרת כוונת הממשק והשימוש בה, באמצעות סמלים וצלמיות, ואינטראקציה קלה ושימושית יותר במחשב. בפרויקט שלי החלטתי להיעזר בממשק גרפי (שימוש בkivy file) על מנת להציג את אפשרויות הבחירה של המשתמש בתצוגה נוחה, פשוטה, ונעימה יותר לעין המשתמש, וזאת באמצעות כפתורים עליהם המשתמש לוחץ לפי החלטתו (במקום הקלדת ספרה לפי האפשרויות המוצגות למשתמש כפלט כפי שהסברתי קודם לכן).

על מנת להריץ את התוכנית באופן זה יש להשתמש בסביבת העבודה PyCharm - כלומר יש להריץ את קובץ ה Python: **newApp.py** באמצעות לחצן ההרצה (משולש ירוק בצד ימין למעלה בתוכנה). (גם באפשרות הרצה זו ישנן בדיקות תקינות קלט. כמו כן, ישנן הודעות אינפורמציה למשתמש המוצגות כפלט טקסט כמו באפשרות ההרצה ה-1 ללא GUI).

הרצת קובץ זה תריץ את כל התוכנית. לאחר הרצת התוכנית יוצג המסך הבא כאשר החלק התחתון הינו כפתור:



לאחר לחיצה על הכפתור התחתון מוצג המסך הבא:



כלומר, מוצג למשתמש תפריט מסודר, צבעוני וויזואלי (באמצעות תמונות) המציג למשתמש את חמשת האפשרויות העומדות לרשותו.

התוכן והסיבה לכל כפתור זהה לאמור בסוג ההרצה הקודם, כלומר כל כפתור מקביל לאחת מחמשת האפשרויות שהוצגו כפלט טקסט בסוג ההרצה הקודמת. אך בכל זאת ישנם שינויים וויזואלים קלים בסגנון ההגשה למשתמש אשר יכולים לשפר את חווייתו.

כאשר נלחץ על כפתור ה Train נפתח המסך הבא ובו ניתן להכניס את הקלטים הרצויים בהתאם (לאישור נלחץ על הכפתור האמצעי submit ולחזרה אחורה נלחץ על הכפתור Go Back):

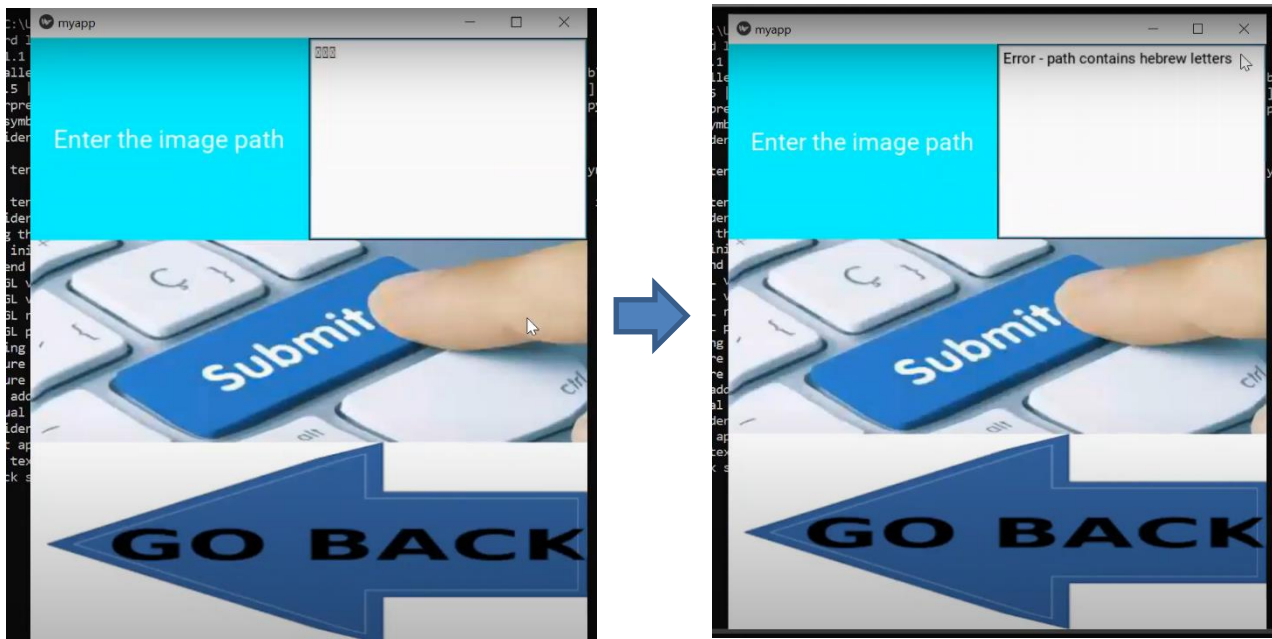


כאשר נלחץ על כפתור ה Predict נפתח המסך הבא ובו ניתן להכניס את הקלט הרצוי לתמונה (בריבוע הלבן), לאישור נלחץ על הכפתור האמצעי submit ולחזרה אחורה נלחץ על הכפתור Go Back:



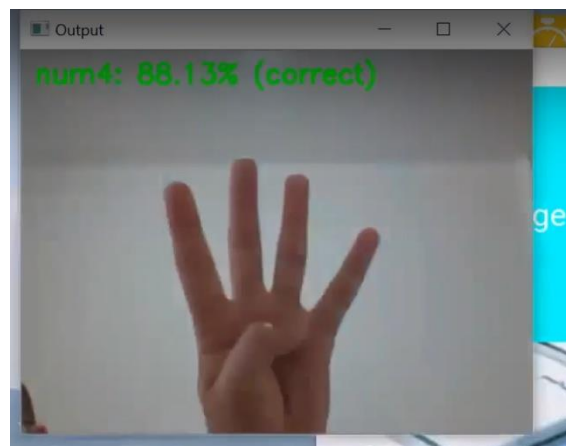
כאשר נכניס כקלט למשל אותיות עבריות, נקבל הודעת שגיאה-

"Error – path contains Hebrew letters"



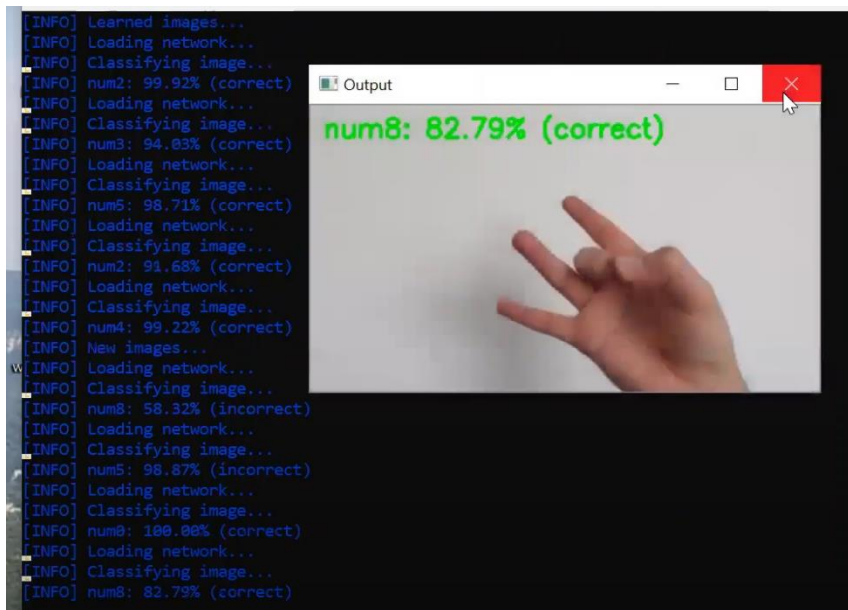
כאשר המשתמש יכניס כקלט נתיב תקין של תמונה, יפתח למשתמש מסך חדש בו התמונה אותה שלחנו כקלט ועליה תווית עם תוצאת החיזוי (מודפסת בירוק).

בנוסף לכך, מוצגות למשתמש הודעות האינפורמציה המעדכנות על התהליך המתרחש. המשתמש יקבל הודעה על תהליך חיזוי התמונה ולאחר מכן יוצג כפלט טקסט זיהוי התמונה וסוגה עם אחוזי הצלחה של החיזוי.



```
2021-06-19 16:10:43.847144: I tensorflow/compiler/xla/service/se
platform Host (this does not guarantee that XLA will be used). D
2021-06-19 16:10:43.847305: I tensorflow/compiler/xla/service/se
Version
[INFO] Classifying image...
[INFO] num4: 88.13% (correct)
```

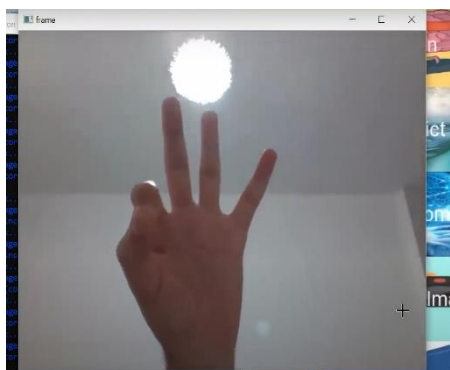
כאשר נלחץ על כפתור ה-Predict Random Images:



לאחר הלחיצה במסך על כפתור זה, שאר התהליך חוזר למסך בו מתקבלות הודעות האינפורמציה ונקבל מיד הודעות אינפורמציה בטקסט במסך זה על כך שמתחיל תהליך החיזוי לתמונות שנלמדו והינן מהמאגר ועל כך שהמודל השמור נטען בידי התוכנית. מיד לאחר מכן המשתמש מקבל הודעת אינפורמציה שהתוכנית כעת חוזרת את הקטגוריה שאלה שייכת התמונה הרנדומלית ולבסוף ניתן לראות שהמודל זיהה נכונה את התמונה ולקבל כפלט טקסט את סוג התמונה (0-9) - (בעזרת קובץ התוויות התוכנית אף יודעת לומר האם היא חזתה את הקטגוריה נכון או לא). בנוסף להדפסה זו, נפתח בפני המשתמש מסך חדש ובו התמונה הרנדומלית שזוהתה ועליה תוויות עם תוצאת החיזוי (מודפסת בירוק). רק לאחר שסיגור את החלון הנ"ל, התהליך יחזור על עצמו פעם נוספת כך שיזוהו מחדש תמונות רנדומליות ובסוף כל הזיהויים התוכנית תבקש מהמשתמש לבחור אופציה חדשה. יש לציין כי לאחר חיזוי של מספר תמונות רנדומליות מהמאגר ישנו מעבר לחיזוי תמונות חדשות רנדומליות שאינן מהמאגר.

כאשר נלחץ על כפתור ה-Take An Image:

לאחר לחיצה על כפתור זה במסך Gui ובעצם לאחר הלחיצה, מתרחש אותו התהליך שצוין קודם לכן באפשרות ההרצה ה-1 כאשר המשתמש מקיש את הספרה 4 לצילום תמונה חדשה. נפתחת המצלמה (אם קיימת) וישנה אפשרות לצלם תמונה חדשה ולשמור אותה במחשב במיקום ובשם שהמשתמש בוחר.



מדריך למפתח:

עבדתי על הפרויקט שלי כך שהוא מחולק למספר קבצי קוד אשר לכל אחד ישנו תחום אחריות שונה וחלק אחר בפרויקט בו הקוד מטפל. החלוקה של הקוד לפי החלקים השונים המרכיבים את הפרויקט השלם, מארגנת ומסדרת את הקוד, מסייעת במציאת שגיאות ובאגים שונים ובעיקר תורמת לכתיבת הפרויקט ומקלה על התהליך.

שמות הקבצים השונים ותחום האחריות שהוטל על כל אחד מהם:

menu.py	קובץ זה מנהל את כל התוכנית לפי בחירות המשתמש – זהו הקובץ הראשי (הקובץ הראשי לאפשרות ההרצה ה-1- הצגת פלטים וקבלת קלטים באמצעות טקסט וללא GUI)
newApp.py	קובץ זה מנהל את כל התוכנית לפי בחירות המשתמש ובאמצעות GUI – (הקובץ הראשי לאפשרות ההרצה ה-2- שימוש בממשק גרפי GUI, בכפתורים והצגה וויזואלית עם תמונות)
train_model.py	קובץ זה אחראי על אימון המודל (משתמש במודל שהוגדר בקובץ Model.py ובמאגר התמונות הממוין לפי הקטגוריות השונות (ספרות 0-9))
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 (שייך לאפשרות ההרצה ה-1 ללא GUI). ההתייחסות הינה לקלטי המיקום של קבצים ותיקיות שונות במחשב.
Cheak_Dir_GUI_Input.py	קובץ זה אחראי על תקינות קלטי המשתמש (שייך לאפשרות ההרצה ה-2 עם שימוש בממשק גרפי GUI - kivy). ההתייחסות הינה לקלטי המיקום של קבצים ותיקיות שונות במחשב.
Model.py	קובץ זה אחראי על הגדרת שכבות המודל.
PrintsForUser.py	קובץ זה אחראי על ההדפסות השונות המוצגות למשתמש
Camera.py	קובץ זה אחראי על צילום תמונה חדשה ושמירתה במחשב (במידה וישנה מצלמה)

הסבר על הקבצים השונים ועל הפונקציות הכלולה בהם:

קובץ ראשי – menu.py:

כאמור, קובץ זה מנהל את מהלך התוכנית לפי קלטי המשתמש ובחירותיו (לפי אפשרות הרצה 1- ללא GUI), וכן הקובץ משתמש בשאר הקבצים כדי לבצע זאת. בקובץ זה מצויה הפונקציה הראשית (ששמה menu()) המהווה פונקציית שירות עבור המשתמש. בהרצת קובץ פייתון זה, יודפס למסך תפריט ובו 5 אפשרויות העומדות לרשותו של המשתמש.

```
Anaconda Prompt - python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py
(base) C:\Users\ilano>python C:\Users\ilano\OneDrive\Desktop\Amit\menu.py
Using TensorFlow backend.
2021-06-18 12:47:58.316001: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'cudart64_110.dll'; this might mean you forgot to include the dynamic library in your library path.
2021-06-18 12:47:58.316225: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dl
*****
*           USER INTERFACE           *
*                                     *
* Enter 1 --> train the model          *
* Enter 2 --> predict an image         *
* Enter 3 --> predict random images    *
* Enter 4 --> take an image            *
* Enter space bar --> exit             *
*                                     *
*****
--> Your Choice:
Enter: _
```

כפי שצוין, מוצעות חמש אפשרויות:

הכנס 1 כדי לאמן את המודל

הכנס 2 כדי לחזות את הקטגוריה של תמונה שהמשתמש ישלח כקלט (נתיב)

הכנס 3 כדי לחזות את הקטגוריה של תמונה שתבחר באופן רנדומלי

הכנס 4 כדי לצלם תמונה חדשה ולשמור אותה במחשב (אם קיימת מצלמה במחשב)

הכנס רווח ליציאה מן התוכנית (יש להקיש רווח ואחריו Enter)

(עבור כל קלט אחר התוכנית תמשיך לבקש קלט).

כאמור, המשתמש יכול לבצע חיזוי ללא אימון – התוכנה תשתמש בקבצים שבהם שמורים הנתונים אותם המשתמש התקין במחשבו מראש (באמצעות default directories) ולפי ההוראות שהצגתי קודם לכן. יש לציין כי התוכנה יודעת להשתמש בקבצים העדכניים עבור כל הרצה חדשה.

אופן מימוש הקובץ:

```

"""
this is the main python file that manage the program according to the user choicess
"""
import os
import train_model
import classify
import Cheak_Dir
import PrintsForUser
import random
import camera

def options():
    """
    This function prints for the user his options (UI)
    """
    PrintsForUser.printOptions("*****")
    PrintsForUser.printOptions("USER INTERFACE")
    PrintsForUser.printOptions("Enter 1 --> train the model")
    PrintsForUser.printOptions("Enter 2 --> predict an image")
    PrintsForUser.printOptions("Enter 3 --> predict random images")
    PrintsForUser.printOptions("Enter 4 --> take an image")
    PrintsForUser.printOptions("Enter space bar --> exit")
    PrintsForUser.printOptions("*****")

```

הפונקציה	תפקידה
Options()	<p>תחום אחריות:</p> <p>פונקציה זו אחראית על ההדפסה של אפשרויות המשתמש למסך.</p> <p>אופן ביצוע:</p> <p>הפונקציה עושה שימוש בפונקציה מתוך הקובץ PrintsForUser.py.</p>
case_One(sorted_data_path)	<p>תחום אחריות:</p> <p>אם המשתמש הקיש 1 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 ממיונות לפי קטגוריות, משתמשת בפונקציות אלוליות שבקובץ שציינתי ומחזירה את מיקום הקובץ שבו שמור המודל, ואת מיקום הקובץ שבו שמורות תוויות התמונות.</p>
case_Two(model_path, labels_path)	<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 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 בפונקציות אלוליות שבקובץ שציינתי ומדפיסה את תוצאת החיזוי.</p>
case_Three(model_path, labels_path, data_set, new_images_folder)	<p>תחום אחריות:</p> <p>אם המשתמש הקיש 3 הפונקציה הראשית menu() תקרא לפונקציה זו. פונקציה זו אחראית על חיזוי תמונות באופן רנדומלי מכל תיקייה.</p>

	<p>אופן ביצוע:</p> <p>פונקציה זו מקבלת את המיקום במחשב של מיקום הקובץ שבו שמור המודל, את מיקום הקובץ שבו שמורות תוויות התמונות, את מאגר התמונות ה data set ותיקיה עם תמונות שהמחשב לא למד, וחזרה 5 תמונות שהמודל למד ו5 תמונות חדשות לגמרי.</p>
cheakPredict(model_path, labels_path, data_set, num)	<p>תחום אחריות:</p> <p>פונקציה זו אחראית לחיזוי predict לnum תמונות רנדומליות שנמצאות בתיקיה משותפת כלשהי (data_set)</p> <p>אופן ביצוע:</p> <p>פונקציה זו משתמשת בלולאת for החוזרת על עצמה num פעמים.</p>
menu()	מנהלת בלולאת while את התוכנית

הלולאה הראשית שבפונקציית ה menu():

```

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

    if choice == '1':
        """
        if the use enter 1 -> the directory of the model and the labeld will be updated
        """
        model_path,labels_path = case_One(sorted_data_path)
        PrintsForUser.printProcess("[INFO] Using trained model")

    if choice == '2':
        """
        if the use enter 3 -> the program will use the updated directory and predict the image
        """
        case_Two(model_path,labels_path)

    if choice == '3':
        case_Three(model_path,labels_path, data_set, new_images_folder)

    if choice == '4':
        camera.takeImage()

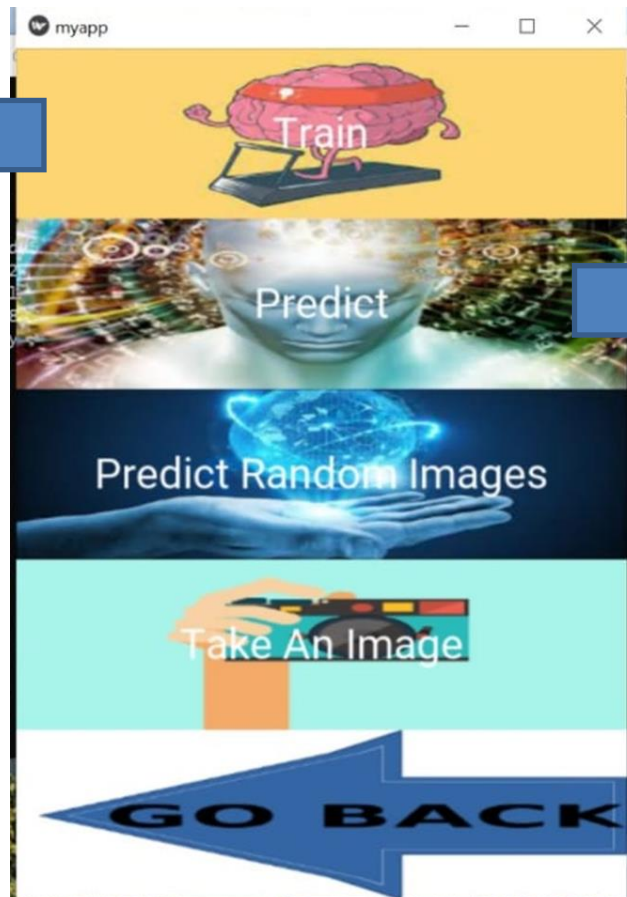
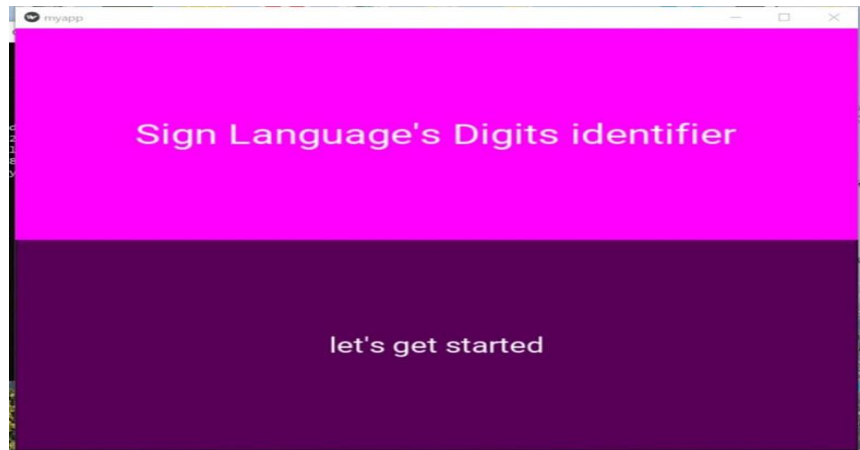
    if choice == ' ':
        PrintsForUser.printProcess("[INFO] Exiting...")
        flag = False

if __name__ == "__main__":
    menu()

```

קובץ ראשי – newApp.py:

כאמור, קובץ זה מנהל את מהלך התוכנית לפי קלטי המשתמש ובחירותיו (לפי אפשרות הרצה 2- באמצעות ממשק גרפי GUI), וכן הקובץ משתמש בשאר הקבצים כדי לבצע זאת. כאמור, בפרויקט שלי החלטתי להיעזר בממשק גרפי (שימוש ב-kivy file- קובץ בסיומת ky אשר נועד לעיצוב ושימוש בממשק גרפי- מסך וויזואלי ובו כפתורים המארגנים את אפשרויות המשתמש) על מנת להציג את אפשרויות הבחירה של המשתמש בתצוגה נוחה, פשוטה, ונעימה יותר לעין המשתמש, וזאת באמצעות כפתורים עליהם המשתמש לוחץ לפי החלטתו. בהרצת קובץ פייתון זה, יפתח מסך ובו יוצג תפריט הכפתורים (האפשרויות) הבא:



מוצגים מספר כפתורים עיקריים:

Train - כדי לאמן את המודל

Predict - כדי לחזות את הקטגוריה של תמונה שהמשתמש ישלח כקלט (נתיב)

Predict Random Image - כדי לחזות את הקטגוריה של תמונה שתבחר באופן רנדומלי

Take An Image - כדי לצלם תמונה חדשה ולשמור אותה במחשב (אם קיימת מצלמה במחשב)

Go Back - לחזרה אחורה

בנוסף ישנו כפתור **submit**

ניתן להסיק כי כל כפתור מבצע משימה מסוימת וכן כי לכל כפתור ישנו תחום עליו הוא אחראי לבצעו ולקשרו אל הקוד המממש את תכליתו.

בקובץ newApp.py ישנן מספר מחלקות הקשורות זו לזו (ניתן לומר כי כל חלון מהווה מחלקה) ומרכיבות יחד את הקובץ ותחום אחריותו-

- myapp(App) – מחלקה ובה נמצאת הפונקציה הבונה build(self)
- MainWindow(Screen) – מחלקה האחראית על המסך הראשי בממשק הגרפי
- WindowManager(ScreenManager)
- PredictWindow(Screen) – מחלקה האחראית על כפתור החיזוי במסך הממשק הגרפי וקישורו לקוד המבצע את החיזוי (ובה הפונקציה Submit(self)).
- TrainWindow(Screen) - מחלקה האחראית על כפתור האימון במסך הממשק הגרפי וקישורו לקוד המבצע את האימון של המודל בפרויקט (ובה הפונקציה Submit(self)).
- OptionsWindow(Screen) - מחלקה האחראית על מסך הממשק הגרפי והצגת האפשרויות העומדות לרשות המשתמש (ובה הפונקציות PredictRandom(self), cheakPredict(self, model_path, labels_path, data_set, num) על חיזוי התמונות ו takeImage(self) האחראית לצילום ושמירת תמונה חדשה במחשב)

Model.py - קובץ הגדרת מבנה המודל:

```
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](#)

[הנוכחי:](#)

```
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'))
```

[המודל הקודם:](#)

בקובץ Model.py יישנה מחלקה בשם MyModel אשר מגדירה את מבנה המודל הפרויקט שבאמצעותו מתבצעת הלמידה. במחלקה זו ישנה מטודה סטטית אחת בשם build אשר מקבלת את ממדי התמונות, מספר הקטגוריות והactivation function (שתבצע בשכבה האחרונה באופן דיפולטיבי- softmax function) ובונה את שכבות המודל.

ניתן לומר כי שכבות המודל מהוות יחדיו את המוח המלאכותי אותו יוצרים בפרויקט- בכל שכבה ישנו מספר שונה של נוירונים ("פרספטרונים") השקולים לתאי העצב במוח האנושי. במוח האדם תאי העצב מחוברים לתאי העצב הסמוכים אליהם באמצעות שלוחות התא, אלו מעבירות אותות חשמליים שמשמשים להעברת מידע בין התאים והקישוריות הזאת בין התאים יוצרת רשת אחת גדולה ומסועפת שמהווה את המוח האנושי. במודל ישנן מספר שכבות אשר בכל אחת מהן מספר שונה של נוירונים מלאכותיים. כמו כן, הנוירונים בכל

שכבה יכולים "לתקשר" עם הנוירונים שבשכבות הסמוכות, ואופן 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.

כלומר, ישנם מודלים רבים של רשתות עצביות, והמשותף לכולם הוא קיומן של יחידות עיבוד בדידות (המקבילה במודל לנוירונים הביולוגיים) הקשורות ביניהן בקשרים רבים. כלומר, רשתות נוירונים מורכבות ממספר רב של יחידות עיבוד פשוטות הנקראות נוירונים, אשר מחוברות באופן היררכי ומובנות בשכבות. הפרטים- מספר הנוירונים, מספר הקשרים ומבנה הרשת (סידור בשכבות, מספר השכבות)- משתנים ממודל למודל. מקובל להשתמש בכלים מתחום האלגברה הליניארית כגון מטריצות ווקטורים על מנת לייצג את הנוירונים והקשרים ביניהם, והפעולה שמבצע כל נוירון על הקלט שלו בדרך כלל מיוצגת על ידי פונקציה. בעצם, רשת נוירונים מאופיינת על ידי: חיבורים - אופן החיבור בין הנוירונים ברשת, משקלים- השיטה הקובעת את משקלי החיבורים בין הנוירונים ופונקציית האקטיבציה, העשויה להיות שונה בכל שכבה.

בעת זימון פונקציה זו- תחילה ערך המשקלים הללו הינו רנדומלי, ורק בעת אימון המודל הם משתנים בהתאם למערכי התמונות שרצים על גביהם. בחרתי במודל זה אשר ישנן בו 28 שכבות (כאשר החשובות שבהן הינן הראשונה והאחרונה), מכיוון שלאחר הרצת התמונות על גביו הוא הניב את אחוז ההצלחה הגבוה ביותר. השכבה הראשונה היא שכבת הקלט המגדירה את ממדי התמונה- input_shape (בפרויקט שלי (50,50,1)), והשכבה האחרונה היא שכבת הפלט אשר באופן דיפולטיבי (במידה ואין קלט אחר) משתמשת בפונקציית softmax. (יש לציין שבשכבה אחת לפני הגדרנו את מספר הקטגוריות האפשרויות – מספר נוירוני הפלט). 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 הגבוה ביותר שהוא הנכון -> וזה בדיוק מה שפונקציה זו עושה -> בוחרת את נוירון הפלט אשר הינו המתאים ביותר עבור אותה התמונה.

קובץ אימון המודל – train_model.py :

זהו קובץ האחראי על אימון המודל בפרויקט. הקובץ כולל מחלקה בשם TrainModel אשר בה מטודה אחת פומבית המשתמשת במספר מטודות פרטיות במטרה לבצע את אימון המודל. ניתן להסיק כי הפונקציות השונות זקוקות לפרמטרים זהים, ועל כן נוצר הצורך ליצירת המחלקה.

נסתכל על בנאי המחלקה, בו כל ה data members הם בהרשאת -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
```

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

נסתכל על המטודות השונות בקובץ-

במחלקה זו זה ישנה מטודה פומבית המנהלת את מהלך האימון (לכן הינה הפונקציה הפומבית היחידה במחלקה) והינה אחראית לבצע את אימון המודל באמצעות המטודות הפרטיות העומדות לרשותה. מטודה ראשית זו נקראת `handle_train()`.

שם הפונקציה	תפקיד הפונקציה
<code>__loading_Images()</code>	מטודה פרטית זו אחראית על טעינת התמונות מן התיקייה הראשית והכנתן לאימון המודל. 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). בנוסף פונקציה זו מחלצת מן שם התיקייה שבה כל תמונה נמצאת את הקטגוריה שלה ומתאימה ביניהן באמצעות שתי רשימות המהוות חלק מתכונות המחלקה- האחת מאכלסת את מערכי התמונות (<code>__data</code>) והשנייה את התוויות (<code>__labels</code>) - סוג התמונה (<code>num</code> -). ההוספה לרשימות נעשית ביחד בלולאה, ולכן ישנה התאמה במספר האינדקס בין שתי הרשימות ובכך ניתן לקשר בין כל תמונה לקטגוריה שלה. השינויים מתבצעים על תכונות האובייקט במחלקה ועל כן מטודה זו אינה מחזירה ערך.
<code>__scale_Pixels()</code>	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> .
<code>__train()</code>	מטודה פרטית זו מבצעת את אימון המודל בפועל- ראשית, מטודה זו שומרת את רשימת התוויות בקובץ בינארי אשר ישמש באימון המודל וכן בחיזוי קטגוריות של תמונות. בנוסף, מטודה זו מחלקת את ה <code>data set</code> לשני חלקים: <code>test - 20%</code> - בדיקת אחוז הצלחת המודל <code>train - 80%</code> - אימון המודל לבסוף מטודה זו מחזירה את ההיסטוריה של לימוד המודל. בנוסף, מטודה זו שומרת את למידת המודל בקובץ ואף את קובץ התוויות הבינארי, זאת על מנת שיהיה ניתן להשתמש בלמידה הנוכחית גם בהרצות אחרות של התוכנית ואף בשביל שיהיה ניתן לבצע <code>predict</code> מבלי להריץ בכל פעם את המודל מחדש.
<code>Graph1(history, plot_path)</code>	מטודה פרטית זו מקבלת כפרמטר את ההיסטוריה וה- <code>directory</code> בו נשמור את פלט המטודה. <code>directory = __plot_dir+ r"\plot1.png"</code> מטודה זו משתמשת בהיסטוריה ויוצרת קובץ תמונה מסוג <code>png</code> בו משורטט גרף הלימוד של המודל.
<code>graph2(history, plot_path)</code>	מטודה פרטית זו מבצעת גרף נוסף (דומה למטודה <code>Graph1</code>) אך באופן מעט שונה. <code>directory = __plot_dir+ r"\plot2.png"</code>

handle_train()	<p>הפונקציה case_Two(sorted_data_path) אשר בקובץ ה menu.py יוצרת אובייקט מסוג TrainModel ומזמנת את המטודה .handle_train()</p> <p>זו המטודה הפומבית היחידה של המחלקה ועל כן ניתן לקרוא למטודה זו רק מתוך אובייקט מאותחל של המחלקה (לכן יש בנאי עם פרמטרים).</p> <p>כאמור מטודה זו דואגת 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>
----------------	--

קובץ ביצוע ה predict – classify.py:

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

- The saved model directory
- The images labels binary file's directory

במחלקה זו ישנה מטודה פומבית אחת- **handle_classify(self, image_path)** את ה directory לתמונה אותה רוצים לחזות ומזמנת מטודות פרטיות (כך ניתן להשתמש באובייקט אחד כדי לחזות מספר תמונות שונות) באמצעותן היא מממשת את מטרת הקובץ. ניצור אובייקט מסוג המחלקה וניתן לו את הפרמטרים הדרושים לו. במחלקה ישנה מטודה הטוענת את המודל השמור במחשב ומטודה נוספת המריצה על גבי המודל את התמונה שהתקבלה כקלט מן המשתמש. מטודה אחרת מוצאת את הקטגוריה אשר לה הסיכוי הגבוה ביותר להיות הנכונה, ובעזרת הקובץ הבינארי ששמרנו בעת אימון המודל אף יוצג למשתמש כפלט אם הקטגוריה אותה זיהה היא נכונה או לא (נערכת השוואה בין התווית המתאימה לתמונה לבין תוצאת החיזוי).

נסתכל 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
```

שם התכונה	ערך	תפקיד התכונה
__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.

נסתכל על המטודות של המחלקה-

שם הפונקציה	תפקיד הפונקציה
__load_Model()	מטודה פרטית הטוענת את המודל השמור, קוראת את הקובץ הבינארי ומאתחלת את תכונות המחלקה: __model, __lb.
__load_Image()	מטודה פרטית המתאימה את התמונה השמורה ב directory של __image_path, לריצה על המודל. כפי שבוצע על תמונות לפני האימון: ישנו שינוי של ממדי

	<p>התמונה לגודל אחיד שנקבע(1,54,96), המרת התמונה מ RGB ל GrayScale, המרת התמונה למערך והקטנת טווח הפיקסלים שלה מ [0,255] ל [0,1].</p> <p>לבסוף המטודה מחזירה את המערך המייצג את התמונה והעתק של התמונה כפי שהתקבלה כקלט.</p>
__predict_Image(image_arr, output)	<p>מטודה פרטית זו אחראית על ביצוע החיזוי עצמו לתמונה על גבי המודל שנטען- היא מקבלת כקלט את הפלט של המתודה הפרטית __load_image() - המערך שמייצג את התמונה והעתק של התמונה.</p> <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 במחלקה וזו שאחראית לזמן את שאר המטודות בסדר הנכון לשם תהליך החיזוי וקבלת הפלט הסופי.</p> <p>מטודה זו מקבלת כפרמטר את הdirectory לתמונה אותה המשתמש בחר לחזות, ומאתחלת את תכונתה image_path.</p> <p>(בקובץ ה menu.py נוצר אובייקט מסוג PredictAnImage וממנו מזמנים את המטודה handle_classify() מהקובץ classify.py).</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>

קובץ לבדיקת תקינות קלט של directory – Cheak_Dir.py:

ישנו שימוש בקובץ זה בכל פונקציה בקוד בה בוצע קלט של directory מן המשתמש, וזאת לצורך בדיקת תקינות הקלט. בקובץ זה ישנה מחלקה בשם GetDirectory המכילה מטודות סטטיות אשר בודקות את תקינות הקלט הנדרש. בהכנסת קלט לא תקין- יוצגו הודעות שגיאה רלוונטיות בהתאם לשגיאה, והתוכנית תדרוש מן המשתמש קלט חדש עד שיוזן לה קלט תקין.

נסתכל על המטודות של המחלקה בקובץ-

שם המטודה	תפקיד המטודה
@staticmethod is_Exists(message, flagLang = False)	מטודה פומבית וסטטית המשמשת לביצוע קלט של directory הקיים במחשב, מן המשתמש. 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 כתקין גם אם הוא כן קיים במחשב. המטודה תמשיך ותבקש מן המשתמש קלט תקין כל עוד לא התקבל אחד כזה, ולבסוף תחזיר את הקלט. <pre> @staticmethod def is_Exists(message, flagLang = False): PrintsForUser.printOptions(message) path = input("Enter: ") while not GetDirectory.__cheak_Exists_Dir(path, flagLang): PrintsForUser.printOptions(message) path = input("Enter: ") return path </pre>
@staticmethod def __cheak_Launguge(path)	מטודה סטטית ופרטית (מטודת עזר) ש"בודקת את השפה" ומחזירה True אם path שהתקבל כקלט אינו מכיל אותיות עבריות, אחרת תחזיר False.
@staticmethod def __cheak_Exists_Dir(path, flagLang)	מטודת סטטית ופרטית (מטודת עזר) אשר בהינתן path ודגל בודקת האם הוא אכן קיים, ואם הדגל הינו True אז בודקת גם את תווים המרכיבים את הכתובת. המטודה תחזיר True אם הכתובת תקינה בהתאם לדרישות אחרת תחזיר False.
@staticmethod __cheak_New_Dir(path, flagLang)	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.
@staticmethod def get_New_Dir(message, flagLang = False)	מטודה סטטית ופומבית זו משמשת לביצוע קלט מן המשתמש של directory שהינו תקין ואינו קיים כבר במחשב. פונקציה זו תחשיב directory כתקין אם הוא מכיל רצף של כתובת קיימת שבסופה תיקייה חדשה או שם קובץ חדש מטודה זו מקבלת את אותן הפרמטרים כמו המטודה

	<p>is_Exists(message, flagLang = False) והיא פועלת באופן זהה מלבד כך שקלט תקין ייחשב כ directory חדש ולא כאחד שכבר קיים. כלומר מטודת העזר שבודקת את ה Path שמתקבל בכל פעם קלט 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>
--	---

קובץ לבדיקת תקינות קלט של directory – Cheak_Dir_GUI_Input.py:

לקובץ זה פונקציונאליות הדומה לפונקציונאליות ולתחום האחריות של הקובץ Cheak_Dir, שכן שני הקבצים מטפלים בבדיקות תקינות קלט המתקבל מן המשתמש, רק שהקובץ הנוכחי Cheak_Dir_GUI_Input.py אחראי לבדיקות הקלט באפשרות ההרצה ה-2 באמצעות תפריט הכפתורים והממשק הגרפי GUI- ובפרט kivy.

בקובץ זה ישנה מחלקה בשם GetDirectory המכילה מטודות סטטיות אשר בודקות את תקינות הקלט הנדרש. בהכנסת קלט לא תקין- יוצגו הודעות שגיאה רלוונטית בהתאם לשגיאה, והתוכנית תדרוש מן המשתמש קלט חדש עד שיוזן לה קלט תקין.

נסתכל על המטודות של המחלקה בקובץ-

שם המטודה	תפקיד המטודה
@staticmethod def __cheak_Launguge(path)	מטודה סטטית ופרטית (מטודת עזר) ש"בודקת את השפה" ומחזירה True אם ה path שהתקבל כקלט אינו מכיל אותיות עבריות, אחרת תחזיר False.
@staticmethod def __cheak_Exists_Dir(path, flagLang)	מטודת סטטית ופרטית (מטודת עזר) אשר בהינתן path ודגל בודקת האם הוא אכן קיים, ואם הדגל הינו True אז בודקת גם את תווים המרכיבים את הכתובת. המטודה תחזיר True אם הכתובת תקינה בהתאם לדרישות אחרת תחזיר False.
@staticmethod def __cheak_New_Dir(path, flagLang)	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.

הקובץ האחראי על הפליטים למשתמש – PrintsForUser.py:

קובץ זה אחראי על ביצוע פלט- ההדפסות למשתמש. ישנו קובץ מיוחד המטפל בהדפסות מכיוון שסיווגתי את ההדפסות לשלושה סוגים שונים:

- הדפסות המבקשות קלט – יודפסו בצבע **ירוק**
- הדפסות שנותנות אינפורמציה על המתרחש בזמן ריצה – יודפסו בצבע **כחול**
- הדפסות של הודעות שגיאה – יודפסו בצבע **אדום**

כל אחת מן הפונקציות הללו מקבלת הודעה להציג למשתמש וקובעת צבע הדפסה שונה בהתאם-

```
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
```

הקובץ האחראי על צילום תמונה חדשה ושמירתה במחשב – camera.py:

בקובץ זה ישנה הפונקציה `takeImage()`, האחראית על צילום תמונה חדשה (אם קיימת מצלמה) אשר נשמרת במחשב במקום ובשם לפי בחירת המשתמש, וכך ניתן להשתמש בה לאחר מכן (באמצעות הנתיב שלה) ולבצע לה חיזוי-

```
import cv2

def takeImage():
    cap = cv2.VideoCapture(0)

    # Define the codec and create VideoWriter object
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('output.avi', fourcc, 20.0, (640, 480))

    while(cap.isOpened()):
        ret, frame = cap.read()
        if ret==True:
            frame = cv2.flip(frame,1)

            # write the flipped frame
            out.write(frame)

            cv2.imshow('frame',frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
        else:
            break

    # Release everything if job is finished
    cap.release()
    out.release()
    cv2.destroyAllWindows()
```

רפלקציה וסיכום אישי:

לאחר מספר שנים שלמדתי מחשבים במסגרת בית הספר בעיקר, התרגשתי מאוד לקראת למידת התחום של למידה עמוקה והכרת המושגים השונים מעולמות הבינה המלאכותית, כן נושאים אלו תמיד סיקרנו ומשכו את תשומת ליבי. אך, עם כל ההתרגשות, במהלך התהליך ועד לעשיית הפרויקט וספר הפרויקט נתקלתי באתגרים ומכשולים רבים שעמדו בדרכי. ראשית, לא תיארת לעצמי עד כמה זהו תחום ענק, ותמיד יש מה לחדש, ללמוד ולהעשיר את הידע בו. נהייתי "לטרופ" את מרחבי האינטרנט עם כל כמויות המידע בו, המדריכים, הספרים, הבלוגים ונדהמתי לגלות עד כמה גדולה קהילת המתכנתים ברשתות ועד כמה חשוב להיעזר אנשים האחד בשני. כל שגיאה שניתן להיתקל בה בעשיית הפרויקט- אדם אחד כנראה נתקל בה לפניי, וכנראה שאלפים ניסו למצוא לבעיה פתרון.

במהלך עשיית הפרויקט העשרתי רבות את הידע שלי, הבנתי את חשיבות הלמידה העצמית ושיפרתי את יכולותיי ועצמאותי בהיבט זה, גיליתי סקרנות רב לגבי החומר הנלמד, ועל כן גם אשמח להעשיר את הידע שלי בתחום גם בעתיד ואף לחקור על פרויקטים נוספים שעשויים לתרום לקהילה ולעולם בכלל. הבנתי עד כמה ניתן להשפיע ואף לשנות חיים של אחרים אפילו באמצעות **קוד קטן ורעיון גדול**, ואני מסופקת מהרעיון שנושא הפרויקט שלי הוא בעל חשיבות וערך משמעותי עבור האחר. לפי השקפתי, לתחום Deep Learning ישנו כוח ופוטנציאל אדיר להביא לשינויים, חידושים טכנולוגיים ופריצות דרך משמעותיים בעולם. פעמים רבות חשתי ייאוש מקטע קוד שלא צלח, ממחסור בזמן שהיה לי או מחוסר מציאת data מתאים, אך על כל אלו הצלחתי להתגבר והיום אני יכולה להגיד שהאתגרים בדרך חישלו אותי וללא שום ספק גם שיפרו את הפרויקט שלי ואת משמעותו עבורי. ניתן גם לומר שבאופן כיתתי נתקלנו באתגר משותף, ועל כן נעזרנו אחד בשני, שיפרנו את התקשורת בינינו והתגבשנו זה עם זה. אני מאמינה גם כי שהעזרה ההדדית והאחוזה שנוצרה בין חברי כיתתי היא התוצר הטוב ביותר שהתקבל מעשיית הפרויקטים תחת נושא מאתגר זה. לכן, בין הכלים שאקח איתי מעשיית הפרויקט- **ההבנה שהכל אפשרי ואין דבר שאני לא מסוגלת לעשות, ההבנה שלכל אחד יש הכוח לשנות וכמובן שאקח איתי לעתיד גם את הידע הרחב שצברתי ואת הניסיון ויכולות הלמידה העצמית שפיתחתי**. בסופו של דבר נהניתי מאוד מעשיית הפרויקט ומפיתוחו, ואני גאה בתוצר הסופי שהשגתי.



מסקנות מהרצת המודל:

ישנם מספר מושגים הנדרשים בתחום:

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 הצלחת המודל לזהות נכון את הקטגוריות אליהן שייכות תמונות אלו קובעת אם המודל אכן למד כראוי את התמונות מן מאגר הtrain.

Accuracy – אחוז ההצלחה של המודל בחיזוי הקטגוריות של התמונות.

Loss – בניגוד לaccuracy אינו פועל באופן בינארי (חיזוי נכון או לא – 0/1) אלא קובע כמה קרובות היו תוצאות חיזוי התמונות, לקטגוריה האמיתית שאליה שייכות.

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

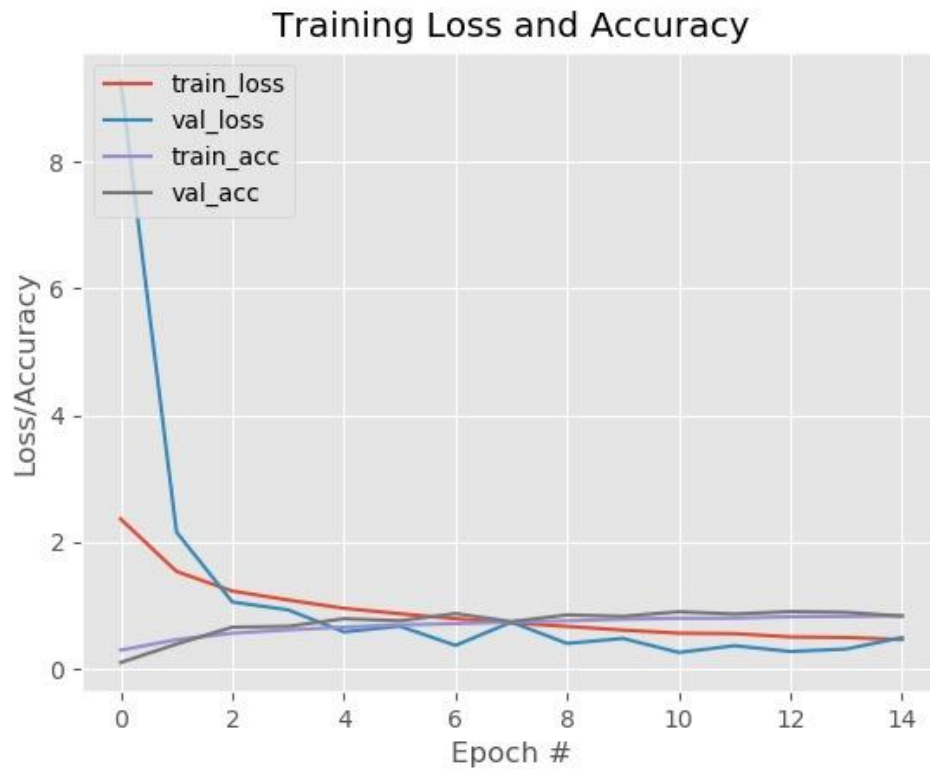
ישנן שתי בעיות אשר יכולות להיווצר אשר ניתן להסיק אותן מן הערכים הללו: overfitting ו under fitting.

-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.

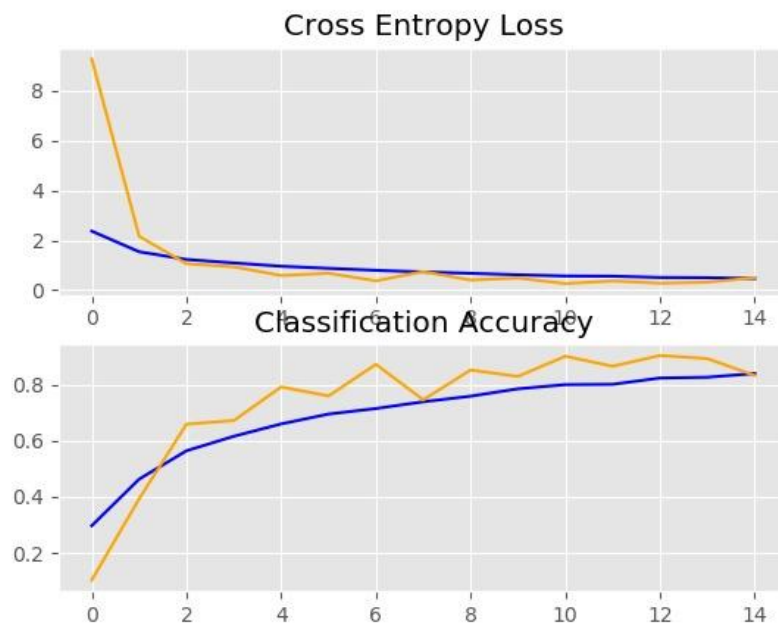
Num epoch – מספר הפעמים בהם יתבצע תהליך הלמידה מחדש.



להלן חלק מגרפי התוצאות של הלמידה מן מאגר התמונות שיצרתי:

לפי תוצאות הגרף ניתן לראות שאכן התבצעה למידה- ניתן לראות שה accuracy וה loss של שני חלקי המאגר התקדמו מ epoch ל epoch באופן דומה, כאשר אין הבדלים משמעותיים ביניהם.

גרף נוסף של תוצאות הרצת המודל (בגרף זה ערכי הloss והaccuracy מופיעים בנפרד):



ביבליוגרפיה:

ספרים:

Deep_learning_for_computer_vision

Practical Python and OpenCV, 4th Edition

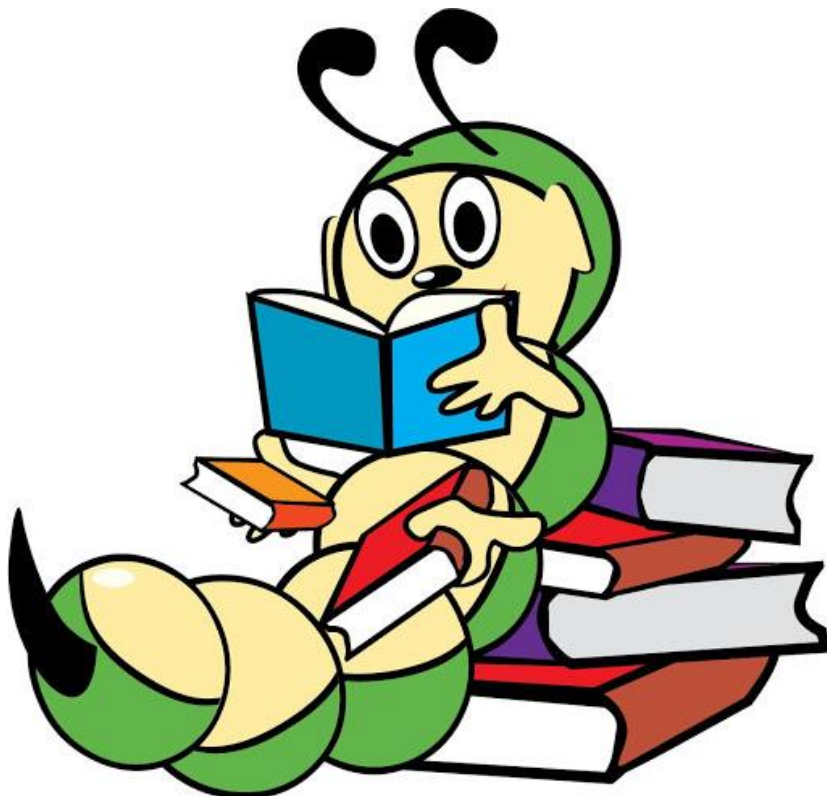
- [/https://stackoverflow.com](https://stackoverflow.com) - שאלות ותשובות בתחומי המחשוב ופיתוח תוכנה וחומרה.

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

The keras blog - <data.html>

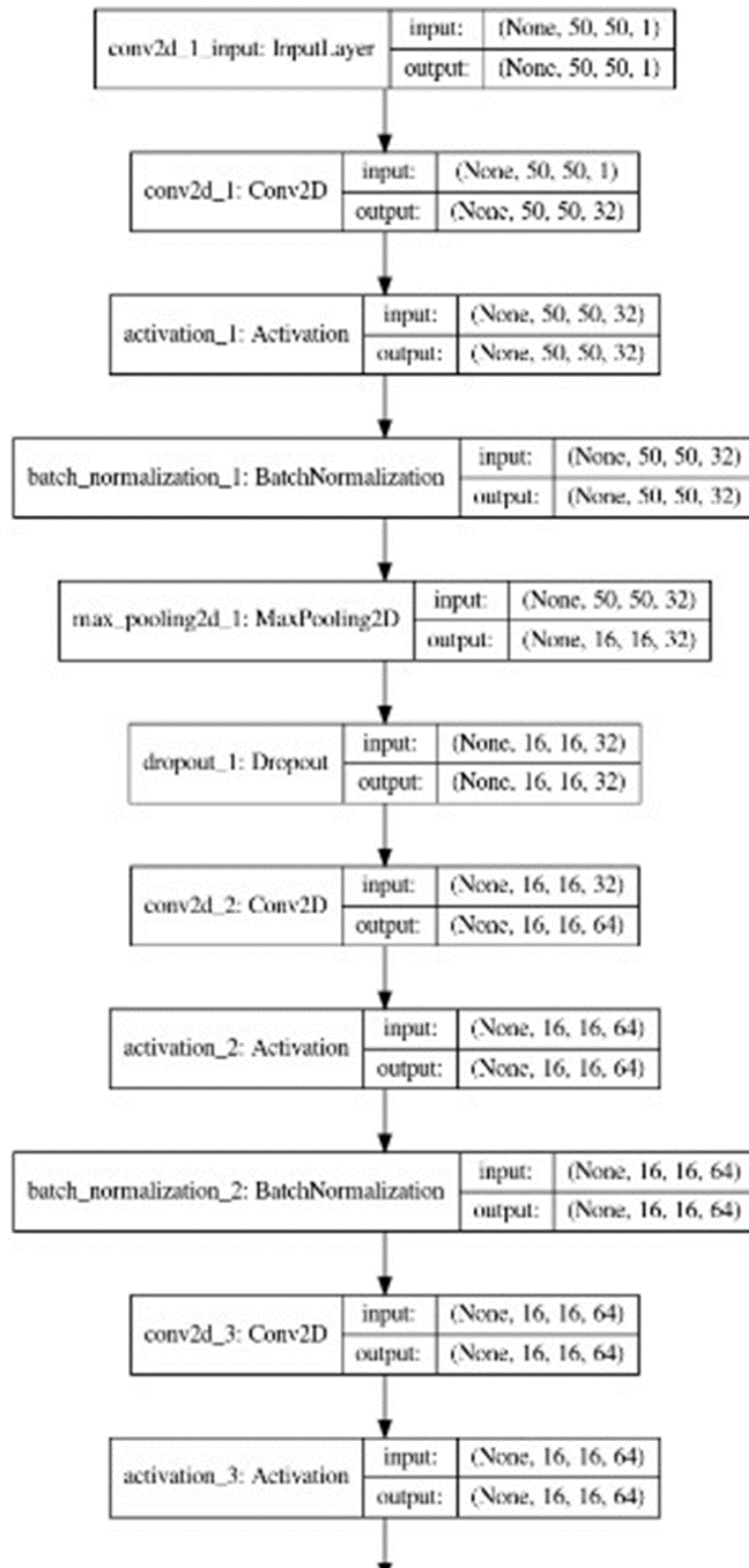
- [/https://pypi.org](https://pypi.org)

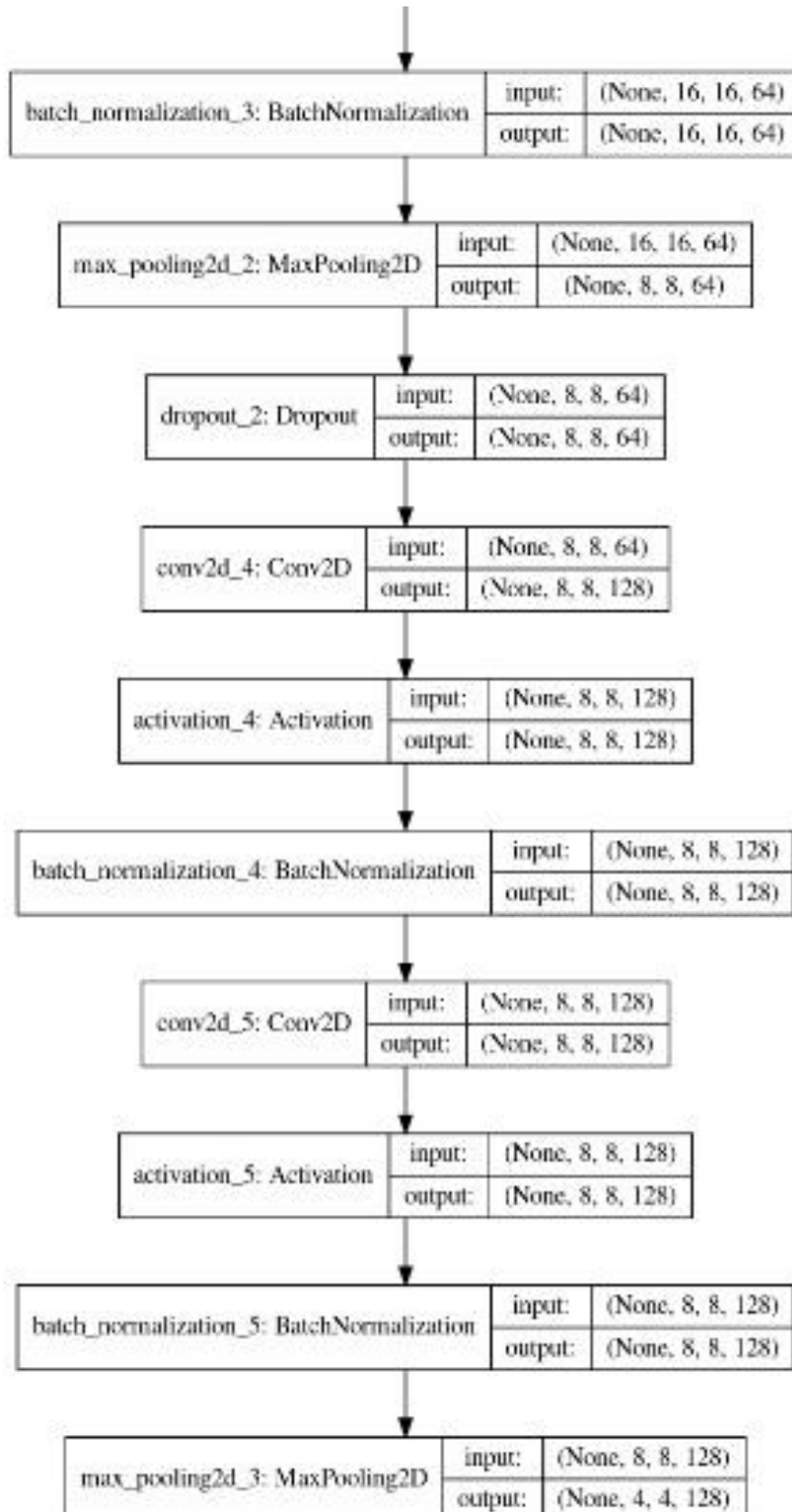
- [/https://www.kaggle.com](https://www.kaggle.com)

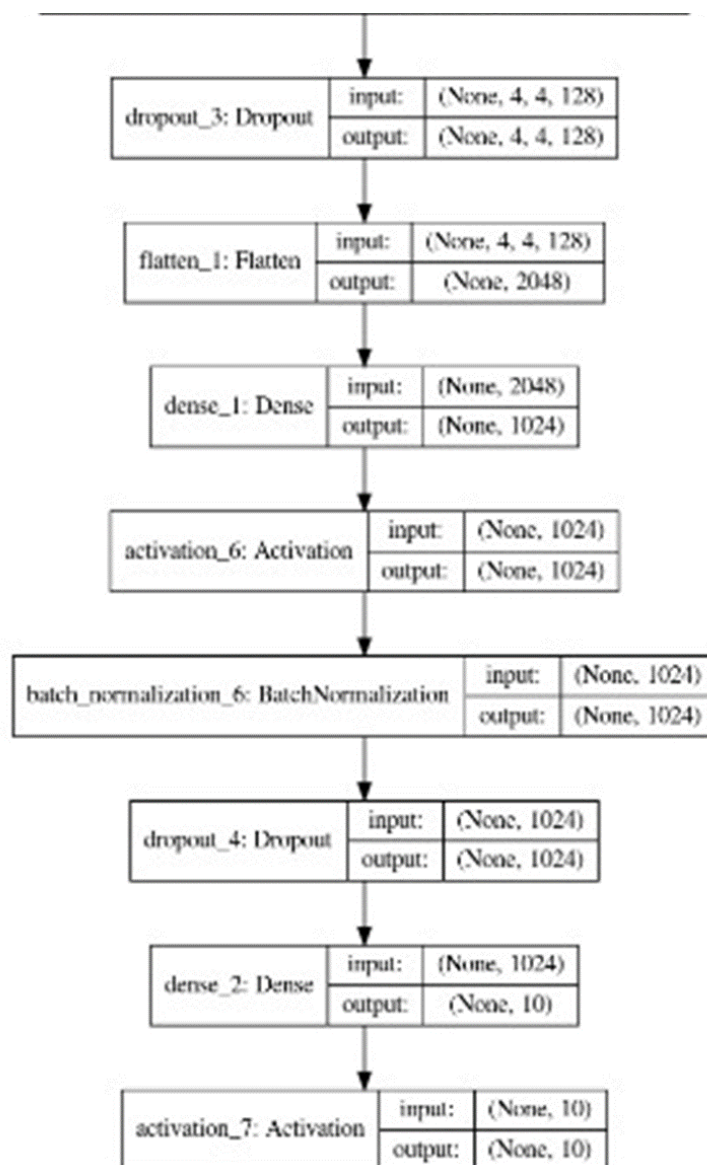


נספחים:

שכבות המודל:







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

Class diagram זו מציגה את המחלקות השונות בפרויקט-

