פרויקט גמר 5 יחידות לימוד התמחות – תכנון ותכנות מערכות Deep Learning

Glasses-No glasses Classification

מגיש: עמית עובדיה

תעודת זהות: 326613346

בית הספר: מקיף י"א ראשונים, ראשון לציון

כיתה: י"ב 4

מורה מנחה: דינה קראוס

תאריך הגשה: 16.6.2022

תוכן

מבוא 3
ארכיטקטורה 4
הDataset: 4
שלב בנייה ואימון המודל 5
תיאור גרפי של המודל שעליו בוצע האימון 5
הסבר על סוגי השכבות השונים ברשת 6
תיאורUML 6
גרפים תוצאות שלב האימון 6
Hyper parameters 7
לשיפור תוצאות האימוניHyper parametersשינויים של ה 7
הסבר של פונקציית השגיאה 7
תיעוד והסבר של ייעול ההתכנסות 8
תיעוד ההתמודדות עם הטיה ושונות ..... **Error! Bookmark not defined.**
software development 8
תרשיmUML 8
תיאור הטכנולוגיה שעל פיה מומש ממשק המשתמש 8
מדריך למפתח 9
Amit_glasses.pyקובץ 9
קובץSoftware.py 11
קובץgraph_Accuracy_Loss 13
קובץImage_Selector_and_Prediction 14
קובץsample.py 14
קובץvars.py 15
מדריך למשתמש 16
הוראות התקנה 16
הרצת התוכנית 17
רפלקציה 21
ביבליוגרפיה 22
נספחים ..... **Error! Bookmark not defined.**

מבוא

למידת מכונה (Machine Learning) היא תת-תחום במדעי המחשב והבינה המלאכותית (AI) העוסק במגוון משימות חישוביות בהן התכנות הקלאסי לא ישים. המאפיין המרכזי של למידת מכונה הוא בפיתוח אלגוריתמים שאינם מבוססים על סט חוקים מוגדר מראש, אלא לומדים מתוך מצבור דוגמאות. המטרה המרכזית של למידת מכונה היא טיפול ממוחשב בנתונים, על ידי מידול, חיזוי או גילוי עובדות מן העולם האמיתי, עבור בעיות שלא ניתן לכתוב להן תכנת מחשב "קלאסית".

כלומר, אם בתכנות הקלאסי כתיבת התוכנה מתבסת על לוגיקה ברורה שמורכבת מהגדרת תנאים מסוימים (בהם לולאות, דרכי פעולה במצבים שונים ועוד), אזי למידה מכונה (Machine Learning)
מאופיינת באלגוריתמים שמטרתם לאפשר למחשב ללמוד את התנאים ולהסיק מידע באופן עצמאי מתוך כמות גדולה של דוגמאות. יכולת זו יעילה מאוד בסיטואציות מורכבות בהן קשה להגדיר מראש סט חוקים או תנאים - מנהיגה אוטונומית, דרך אבחון גידולים סרטניים, ועד לחיזוי העדפות משתמש.

למידת מכונה (Learning Machine) היא טכנולוגיה המאפשרת לפתור בעיות תוכנה, שקשה מאוד עד כדי כך שלא ניתן היה לפתור אותם בעזרת תכנות מסורתי הכולל משפטי if ולולאות. for טכנולוגיה זו מבוססת על היכולת לשפר באופן אוטומטי את ביצועי המחשב במשימות מורכבות. למידת מכונה משתמשת בשיטות מתמטיות/סטטיסטיות כדי לשפר את ביצועי האלגוריתם תוך כדי שימוש בנתונים העוברים דרך האלגוריתם.

מטרת הפרויקט היא למיין תמונות של פרצופים של אנשים הלובשים משקפיים או לא.
קהל היעד של פרויקט זה יכול להיות כל אדם אפשרי אשר לובש משקפיים או לא.
הפרויקט ממיין תמונות של אנשים לבין אם לובשים משקפיים או לא.
בחרתי בפרויקט זה מכיוון שעניין אותי לדעת כיצד מכונת יכולה ללמוד להבדיל בין פריטים אשר אנשים לובשים על פניהם.

אחד האתגרים המרכזיים היו ללמוד את החומר של דיפ לרנינג כמעט הכל לבד בעזרת חברים וטיפה עזרה מהמורה מכיוון שזהו תחום חדש של למידה החומר עליו באינטרנט מופיע כמעט רק באנגלית ובמאמרים מתקדמים עם מתמטיקה גבוהה שיכולה להיות קשה להבנה.
התגברתי על אתגר זה בכך שלמדתי עם חברים את החומר ועזרנו אחד לשני.
הפרויקט עונה על צורך של זיהוי משקפיים על אנשים ללא עזרת האדם אלא רק בעזרת מכונה.

ארכיטקטורה

הDataset:
https://www.kaggle.com/datasets/jeffheaton/glasses-or-no-glasses

התמונות נלקחו מהאתרKaggle התמונות חולקות לתתי תיקיות עם הבדל בין משקפיים, התמונות של האנשים הם אינם אנשים אמיתיים אלא אנשים שנוצרו על ידי מכונת שמייצרת פרצופים של אנשים לא אמיתיים. תוך כדי הפרויקט אני משנה חלקים קטנים בתמונות(augmentation) כדי לגרום למכונה ללמוד טוב יותר ולשפר את הDataset.
לאחר שינויים קלים הDataset החדש נמצא להורדה אצלי בkaggle ושאר הקבצים נמצאים בgithub.

לינק לkaggle:

https://www.kaggle.com/datasets/amitov/glasses-no-glasses-classification-dataset-amit

לינק לgithub:

https://github.com/amit126787/Project-Glasses-No-glasses-clasification

תיאור וניתוח הנתונים הגולמיים

הDataset הגיע בתיקייה המחולקת מראש לאנשים עם משקפיים ואנשים בלי וחילקתי את התמונות לtrain ולvalidation ב70%/30%.

תיאור תהליך הכנת ה Dataset לאימון כולל הסבר אודות שיטת נרמול הנתונים

לפני הרצת המודל, אנו מבצעים נרמול לנתונים(לתמונות כולל חלק מהתמונות שעברו augmentation בעזרת הפעולה הבנויה של Keras שהיא ImageDataGenerator ובעזרתה נעשה נירמול לנתונים.
נרמול נתונים הוא התהליך של להפוך את כל הפיקסלים בתמונה לערכים בין 0-1, כש-0 הוא הערך הקטן ביותר ו-1 הוא הגדול ביותר. זה משפר את יכולת האימון של המודל, ונותן לו לעבוד עם טווח קטן יותר של מספרים .יש כמה שיטות לנרמול, אך השיטה שאנו משתמשים בה משתמשת בנוסחה הבאה-

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

במקרה שלנו, כיוון שהפיקסל המינימלי הוא 0 והגבוה ביותר ההוא 255, אנו יודעים שזה הטווח של המספרים כך שבפועל אנו מחלקים את כל ה dataset-ב- 255 כיוון ש x/255 =(x-0)/(255-0) -כך נקבל את התמונות בטווח של 0-1 כשפיקסל של 255 הוא 1 ופיקסל של 0 הוא 0, וכל השאר נמצאים ביניהם.

שלב בנייה ואימון המודל

תיאור גרפי של המודל שעליו בוצע האימון

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 198, 198, 32)      896

 activation (Activation)     (None, 198, 198, 32)      0

 max_pooling2d (MaxPooling2D  (None, 99, 99, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 97, 97, 32)        9248

 activation_1 (Activation)   (None, 97, 97, 32)        0

 max_pooling2d_1 (MaxPooling  (None, 48, 48, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 46, 46, 64)        18496

 activation_2 (Activation)   (None, 46, 46, 64)        0

 max_pooling2d_2 (MaxPooling  (None, 23, 23, 64)       0
 2D)

 flatten (Flatten)           (None, 33856)             0

 dense (Dense)               (None, 64)                2166848

 activation_3 (Activation)   (None, 64)                0

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

 activation_4 (Activation)   (None, 1)                 0
```

```
=================================================================
Total params: 2,195,553
Trainable params: 2,195,553
Non-trainable params: 0
_________________________________________________________________
```

הסבר על סוגי השכבות השונים ברשת

שכבה 1 Conv2D filters-32 kernel size-3x3

שכבה 2 MaxPooling2D pool_size = 2x2

שכבה 3 Conv2D filters-32 kernel size-3x3

שכבה 4 MaxPooling2D pool_size = 2x2

שכבה 5 Conv2D filters-64 kernel size-3x3

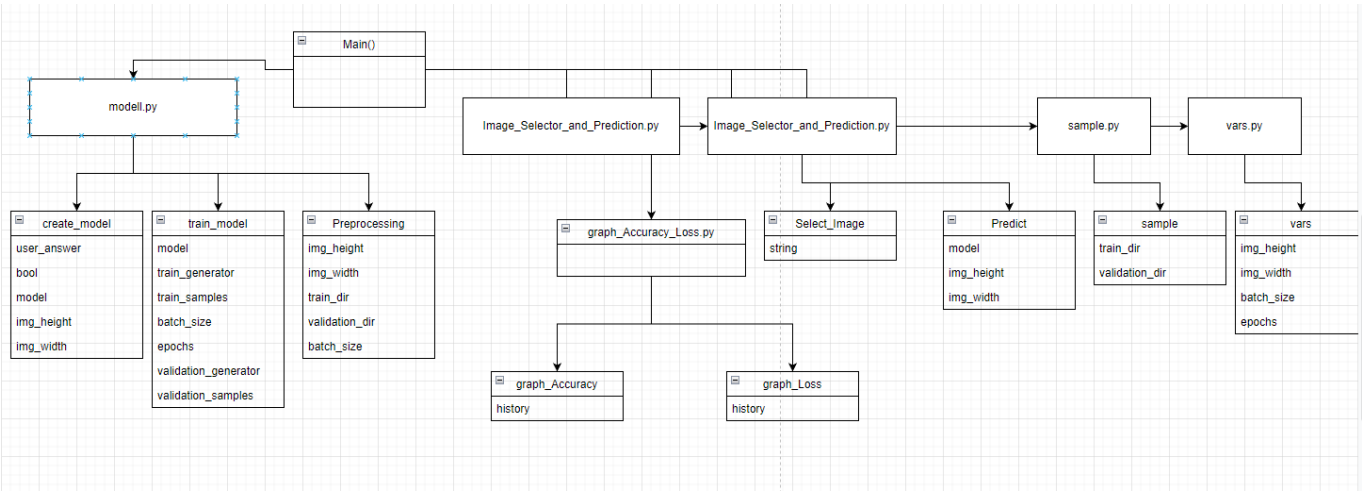
שכבה 6 MaxPooling2D pool_size = 2x2

שכבה 7 Dense=64=number of neurons

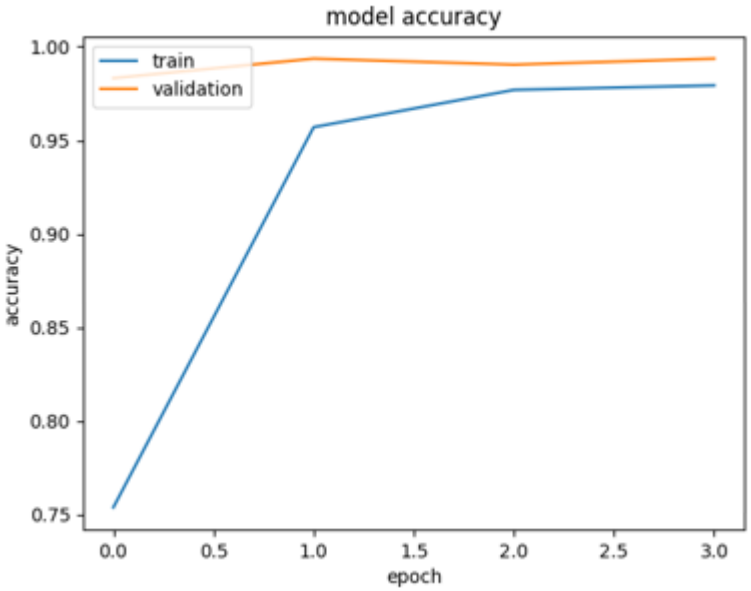
Dropout = 0.5

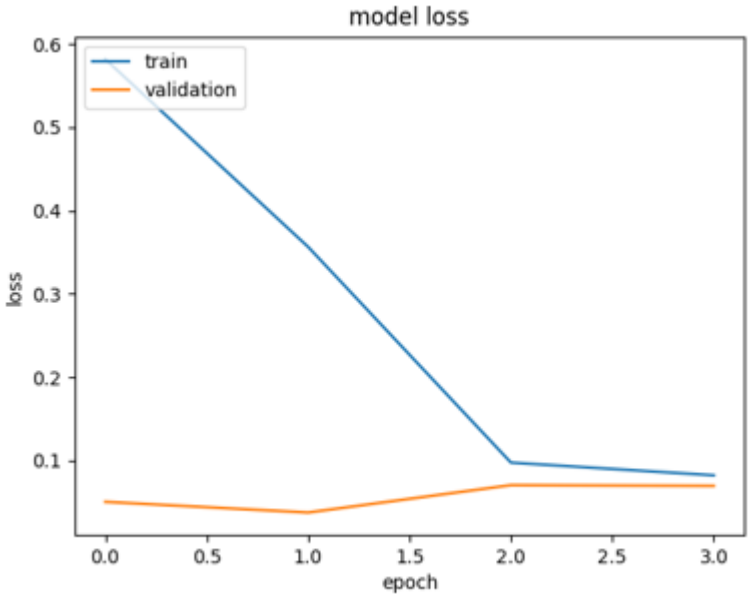
שכבה 8 Dense=1=number of neurons

תיאור UML



גרפים תוצאות שלב האימון





Hyper parameters

PARAMS = {'batch_size': 32,

'shuffle': True,

'activation': sigmoid&relu,

'dense_units': 64,

'dropout': 0.5,

'learning_rate': 0.001,

'early_stopping': 2,

'optimizer': rmsprop, }

שינויים של הHyper parameters לשיפור תוצאות האימון

Dence משתנה ב1 כל פעם

Dropout משתנה ב0.5 כל פעם

הסבר של פונקציית השגיאה

הסבר על הLoss Function (הגרף השמאלי): ה-Loss Function הופכת ערכים של משתנה אחד או כמה למספר ממשי המייצגים עלות. בפונקציית הLoss אנחנו מוצאים כמה המכונה קרובה בתוצאות לחיזוי התמונות לקטגוריות המתאימות להן. ככל שה-Loss יותר נמוך, כך המכונה קרובה יותר לחיזוי. כמו שאפשר לראות בגרף השמאלי, ניתן לראות כי ה-Loss יורד, כך שהמכונה מזהה טוב את התמונות.

Binary Cross Entropy

הפונקציה משווה כל "חיזוי" של המודל לתשובה הנכונה האמיתית ולאחר מכן מחשבת את כמות ה"שגיאה" של הפונקציה כלומר כמה היא הייתה קרובה לתשובה האמיתית.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

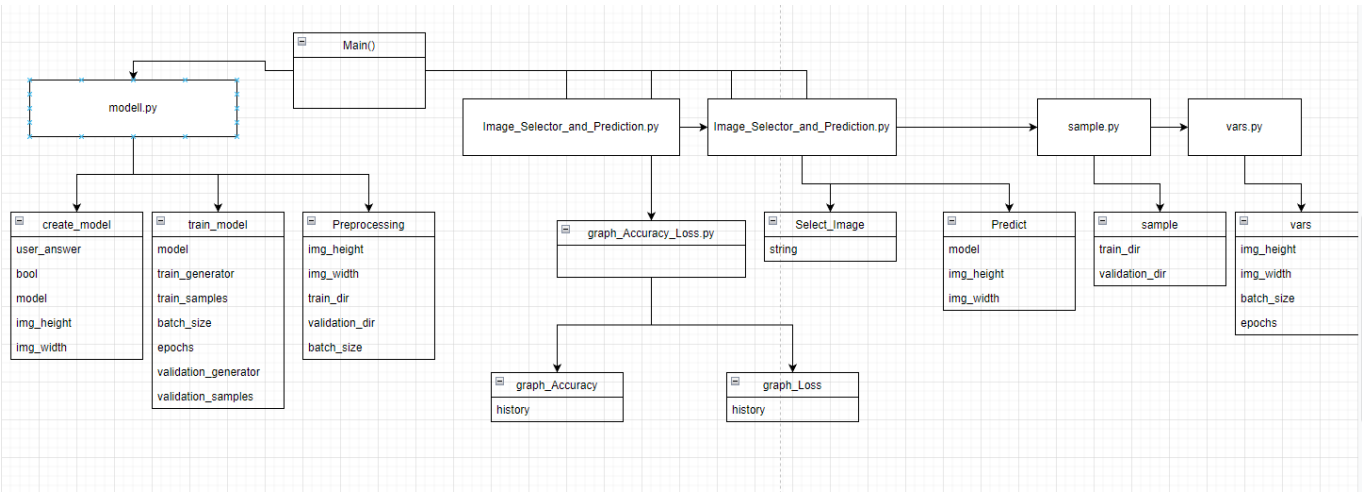
תיעוד והסבר של ייעול ההתכנסות

ה-Optimization מייצר פונקציית מינימום ופונקציית מקסימום בשביל להבין איך המודל עובד. בעזרתו נאמן את המודל שלנו עבור כל epoch כך שככל שמספר ה-epoch גדל, כך מקבלים תוצאות טובות יותר. כמו שאפשר לראות בגרפים, פונקציית ה-Accuracy עולה ככל שמספר ה-epoch גדל, כך שהמכונה מדייקת בחיזוי שלה לתמונות בקטגוריה. ובגרף השני , פונקציית ה-Loss יורדת ככל שמספר ה-epoch גדל, כך שהטעות שהמכונה עושה באימון המודל יורדת בכל פעם, וכך היא חוזה יותר טוב את התמונות.

software development

המודל משתמש ביישום השואל שאלות את המשתמש בcommand line, המשתמש עונה בתשובה בכתב yen/no(y/n) וכך המודל קולט את רצונות המשתמש ומעביר את המודל לפעולות הנכונות בהתאם לתשובותיו של המשתמש.

תרשים UML



תיאור הטכנולוגיה שעל פיה מומש ממשק המשתמש

שורת הפקודה Command Prompt, באנגלית או CMD בקיצור, הינו ממשק מערכת הפעלה של Windows והוא משמש להריץ פקודות שהוזנו על ידי המשתמש.

ממשק המשתמש משתמש באפליקציה הבנויה במחשב הנקראת cmd(command prompt) וכך המשתמש מקליד את רצונותיו לפי הדרישות של המודל וההוראות הבנויות בפייתון וכך המודל מתקדם בעבודתו.

מדריך למפתח

קובץ Amit_glasses.py

תפקידו: זהו הקוד הראשי של הפרויקט

מיקום: scripts תקייה

משתנים:

מגדירים משתנים עיקריים בסיסיים שבהם נשתמש לאורך כל הפרויקט img_width, img_height)מימדי התמונות הקבועים שבהם נשתמש בכל קטע בו יש שימוש בתמונות(, batch_size)גודל קבוצת תמונות, בו נשתמש בחלק זה ובתהליכי אימון המודל(, epochs)מספר האפוקים, שבו נשתמש בתהליכי אימון המודל(. בנוסף, הגדרנו באמצעות הספרייה warnings שהתוכנית תסנן אזהרות, מכיוון שהן לא קשורות למשתמש ורק יפריעו למשתמש בכך שישימו הערות לא שימושיות למשתמש)אין לכך השפעה על התוכנית עצמה.
Answer המשתנה הוא תשובה של y/n מהמשתמש לגבי הוצאת קבצים מzip או לא. במידה וכן הפעולה תזמן את הפעולה open_zip מהקובץ zip_extract.py, אם לא התוכנית תמשיך ללא הוצאת קבצים ותרוץ כרגיל בהנחה שכבר הקבצים מותקנים על המחשב.

הגדרת כתובות תיקיות הפרויקט שבהן נעשה שימוש. אנו עושים זאת בעזרת הגדרת המשתנה main_dir שיכיל את הכתובת לתיקייה הראשית
Glasses_No_glasses_project דרך הפעולה Select_directory שנמצאת בקובץ Software.py.
אחרי שבmain_dir נמצאת כתובת התיקייה הראשית, המשתנים data_dir, train_dirוvalidation_dir מוגדרים כאשר המחרוזת שהם מכילים מייצגת את המשך הכתובת main_dir לפי שמות התיקיות שבאות עם הפרויקט. תהליך זה מתרחש בעזרת הפעולהpath.join של הספרייה os, המאפשרת לנו להגדיר משתנים ככתובות המשך של כתובת.
מכינים את מאגרי התמונות לאימון המודל. מגדירים את המשתנים train_gen וvalidation_gen ובתוכם שמים את הערכים שיוחזרו עם הפעולה Preprocessing בקובץ modell.py.
המשתנים train_samples וvalidation_samples הם משתנים שמכילים בתוכם את מספר התמונות הכולל שנמצא בתיקייה train וvalidation בהתאמה. המשתנים האלו נחוצים עבור תהליך אימון המודל – לצורך חישוב steps_per_epoch. steps_per_epoch הם מספר הקבוצות)batches(שבהן מתאמן המודל לכל אפוק)epoch(אחד. המשתנים train_sampleוvalidation_samples נחוצים עבור חישוב הsteps_per_epoch בכך שמחלקים את סך התמונות של כל אחד בגודל קבוצה)batch_size(. כך בעצם בכל אפוק)epoch(המודל
מאומן על ידי כל התמונות שבמאגר התמונות, כמה פעמים.

train_samples וvalidation_samples מקבלים את הערך שלהם דרך הפעולה Sample שבקובץ sample.py שמחשבת את סך כל התמונות בtrain וvalidation ומחזירה את הערכים בתור המשתנים.
model הוא המודל שכל הפרויקט סובב סביבו. את המודל אנו יוצרים, מאמנים/ מכניסים ערכים מוכנים מראש, ובו אנו משתמשים על מנת לחזות את התמונות שהמשתמש בוחר.
את המודל אנו יוצרים ומגדירים דרך ספריית הקוד keras, דרכה אפשר ליצור מודל ולהגדיר את

המבנה שלו בצורה מאוד פשוטה, כפי שניתן לראות בפעולה create_model בקובץ modell.py.
Load_answer הוא משתנה של תשובה מהמשתמש של y/n במידה וכן y הפעולה תטען מודל בנוי ולא תעבור על אימון המודל ובידה ולא n הפעולה תאמן את המודל.
המשתנה history הוא משתנה השומר בתוכו את היסטוריית אימון המודל.
המטרה העיקריתשל השמירה על היסטוריית המודל היא עבור הדפסת גרפים המראים את השינוי בדיוק המודל Accuracy ואת השינוי בLoss של המודל.
המשתנה bool מתפקד כמשתנה בוליאני המופיע בלולאה של חיזוי התמונות. בסוף התוכנית המשתמש נשאל האם הוא רוצה לבחור תמונה בשביל לתת למודל לחזות את סוגה. אחרי כל פעם שהמשתמש מסיים את תהליך הבחירה וחיזוי התמונה, הוא נשאל את השאלה הזו שוב)במטרה לתת לו אפשרות לבחון את המודל כמה פעמים שיירצה(. אם המשתמש בוחר שלא להמשיך, התוכנית תסתיים.
המשתנה bool משמש כתנאי לצאת מהלולאה כאשר המשתמש בוחר לסיים את התהליך.
)מוגדר כTRUE לפני הלולאה ומוגדר כFalse כאשר מסתיימת הלולאה, מכיוון שהלולאה היא while(bool=True)
בפונקציה check_user_answer שבקובץ Sofyware.py.
המשתנה user_answer, כמו המשתנה bool, מופיע בלולאה של חיזוי התמונות.
המשתנהמשמש לבחירה בין שתי אופציות: המשתמש מתבקש לבחור האם לחזות תמונה או לסייאת התהליך)לא לחזות תמונה(. כאשר המשתמש מקיש n ואז Enter(, הלולאה מסתיימת וכך גם מסתיימת התוכנה. כאשר המשתמש מקיש y)ואז Enter(, הוא יעבור לתהליך בחירה וחיזוי התמונה. לאחר חיזוי התמונה, התוכנית משנה את ערך המשתנה user_answer לערך שאינו n ואינו y על מנת לתת למשתמש לבחור שוב אם לחזות תמונה. כלומר, אם המשתמש יקיש ערך שאינו y ואינו n, הוא יישאל שוב עד שתשובתו תהיה תקינה.

User_answer מכיל את הinput של המשתמש לשאלה שמוצגת בלולאה.

קובץ Software.py

תפקידו: זהו הקוד שמכיל את האפליקציה של המודל.

מיקום: scripts תקייה

פעולה check_user_answer

המשתמש בוחר לאם הוא רוצה לחזות תמונה

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

לא מחזירה כלום

פעולה check_load_answer

המשתמש בוחר האם הוא רוצה לטעון מודל בנוי או לאמן את המודל מההתחלה.

המשתנים הוסברו בקובץ Amit_glasses.py.

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

מחזירה משתנה history המכיל את היסטורית הגרפים וערכים

פעולה Select_directory

הפעולה מקבלת מחרוזת ששואלת את המשתמש איפה נמצא קובץ הפרויקט ולאחר מכן מחזירה את המיקום שהוכתב על ידי המשתמש.

Directory_path המשתנה הוא מחרוזת שהמשתמש מקליד של היכן נמצא קובץ הפרויקט.

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

לא מחזירה כלום

קובץ zip_extract.py

תפקידו: תפקיד הקוד הוא לעשות extract לקובץ הzip עם הDataset ושאר הקבצים.

מיקום: scripts תקייה

פעולה open_zip

תפקיד הפעולה הוא לעשות extract לקובץ הzip עם הDataset ושאר הקבצים.

File_name משתנה הוא מחרוזת שהמשתמש מקליד של המיקום של קובץ הzip.

הפעולה מקבלת:

-Filename משתנה שמכיל את מיקום קובץ הzip

הפעולה מחזירה:

לא מחזירה כלום.

קובץ modell.py

תפקידו: זהו הקוד בניית המודל ואימונו.

מיקום:scripts תקייה

פעולה create_model

הפעולה מקבלת כפרמטרים את img_height וimg_width, מימדי התמונה שבהם אנו משתמשים לאורך כל הפרויקט. בתחילת הפעולה אני משתמש בספרייה K בשביל לבדוק מהו פורמט התמונות שינתנו למודל שלנו, כך שיהיה ניתן לכתוב את צורת הinput כנכונה – כך לא יהיו לנו בעיות עם הקשר בין התמונות למודל בתוכנית.
לאחר שהגדרנו את צורת הinput נוכל להגדיר את המודל ואת המבנה שלו.תחילה אנו מגדירים את המודל עצמו. אנו משתמשים בפעולה ()Sequential מהספרייה keras שמגדירה עבורינו מודל שיהיה ניתן להוסיף לו שכבות בקלות. לאחר הגדרת המודל, כל השורות האחרות בפעולה מגדירות את המבנה שלו שכבות.

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

model הוא המודל.

פעולה train_model

הפעולה מקבלת כפרמטרים את המודל, את הגנרטורים שהגדרנו קודם לכן, הכמות הכוללת של התמונות בtrain וvalidation)שהגדרנו קודם לכן(, את גודל קבוצות המידע)batch_size(ואת כמות האפוקים)epochs(, שהגדרנו בתחילת התוכנית.

בתחילת הפעולה אנו מגדירים את earlyStopping שהיא שיטה שמונעת Over Fitting על ידי עצירת אימון המודל כאשר ביצועיו מגיעים לשיאם)הסבר מלא בבסיס הנתונים(.

לאחר מכן אנו מגדירים את history שיהיה שווה לmodel.fit, פעולה מהספרייה keras שמאמנת את המודל עם הגנרטורים שיצרנו, על כמות מסוימת של 'צעדים' עבור train וvalidation)הסבר מלא בבסיס הנתונים(, והגדרת מתודות שניתן להוסיף לתהליך האימון)כמו earlyStopping שהגדרנו לפני כן(. השוואת הhistory לאימון המודל מאפשרת לנו לשמור את היסטוריית אימון המודל, למטרות כמו גרפים שמראים את התקדמות המודל עם תהליך האימון. בשורה הזו בעצם אנו מאמנים את המודל ושומרים את תיעוד התהליך

בhistory.

הפעולה מסתיימת ומחזירה את model וhistory.

פעולה Preproccecing

בפעולה זו המטרה העיקרית היא ליצור עצמים מסוג Generator של הספרייה kera על מנת להכין את מאגרי המידע לקראת אימון המודל. בחרתי להשתמש בGenerator מכיוון שניתן לבצע Data Augmentation)שינויים בתמונות, כמו שינוי זווית, zoom in, שינוי גודל וכו'(באופן פשוט מאוד כבר ביצירה שלו. השימוש בData Augmentation מאפשר לנו למנוע מקרה של Over Fitting במודל, וכך ליצור מודל טוב יותר.

תחילה אנו יוצרים את המשתנה train_datagen בעזרת הפעולה ImageDataGenerator, עסכמה הגדרות שמפעילות את הData Augmentation, ומיד לאחר מכן אנו יוצרים את המשתנה train_datagen בעזרת אותה פעולה, אך עם פחות אפשרויות כמו train_datagen,מהסיבה הפשוטה שהתמונות בvalidation צריכות פחות שינויים מהתמונות בtrain)באימון חשוב שהמודל ייחשף למגוון רחב מאוד של תמונות מכל הסוגים ובכל הצורות(.

לאחר מכן אנו ממירים את הגנרטורים שיצרנו לtrain_generator וvalidation_generator)בהתאמה(בעזרת הפעולה ()flow_from_directory שמקשרת את הגנרטורים למאגר התמונות)דרך התיקיות ששלחנו כפרמטרים לפעולה(, משתמשת בחלוקת התיקיות בשביל לתת לכל תמונה במאגר תווית שמסמנת עבור כל תמונה את סוגה)משקפיים/בלי משקפיים,במקרה זה(, נותנת לתמונות במאגר גודל קבוע)שאת מימדיו שלחנו לפעולה Preprocessing כפרמטרים(, ומגדירה את סוג הסיווג של התמונות)בפרויקט הזה, הסיווג הוא בינארי, מכיוון שיש לנו שני קלאסים בלבד(

התוכנית מגיעה לסוף הפעולה, והגנרטורים החדשים שיצרנו)validation_generator, train_generator(נשלחו בחזרה על ידי הפעולה.

קובץ graph_Accuracy_Loss

תפקידו: זהו הקוד לבניית גרפים של דיוק והפסד המודל.

מיקום:scripts תקייה

פעולה graph_Accuray

פעולה בשביל גרף דיוק המודל

History משתנה היסטוריית אימון המודל

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

לא מחזירה כלום.

פעולה graph_Accuray

פעולה בשביל גרף הפסד המודל

History משתנה היסטוריית אימון המודל

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

לא מחזירה כלום.

קובץ Image_Selector_and_Prediction

תפקידו: זהו הקוד לחיזוי תמונות ובחירת של התמונות על ידי המשתמש.

מיקום:scripts תקייה

פעולה Select_Image

הפעולה מקבלת את מחרוזת string שהגדרנו בפעולה predict ומבקשת מהמשתמש

להקליד את מיקום של תמונה שהיא png/jpeg בלבד.

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

הפעולה מחזירה את המיקום של התמונה.

פעולה Predict

הפעולה מקבלת כפרמטרים את המודל, ואת ממדי התמונות שבהם השתמשנו לאורך כל התוכנית img_height,img_width וגם model.

אנו מגדירים את המשתנה string שיהיו מחרוזת שתוצג עבור המשתמש.לאחר מכן אנו

מגדירים את img_path הכתובת לתמונה הנבחר(כמשתנה שהפעולה Select_Image

באותו הקובץ.

הפעולה מקבלת:

המשתנים הוסברו בקובץ Amit_glasses.py

הפעולה מחזירה:

לא מחזירה כלום.

קובץ sample.py

תפקידו: זהו הקוד לבניית גרפים של דיוק והפסד המודל.

מיקום:scripts תקייה

הפעולה מקבלת כפרמטרים את כתובות התיקיות של train וvalidation שאותן הגדרנו בחלק

,train_samples

1 (הכנת מאגרי המידע.) הפעולה מחזירה את הפרמטרים

validation_samples שמכילים בתוכם את כמות התמונות הכוללת שיש בtrain וכמות

התמונות הכוללת שיש ב,validation בהתאמה. הפרמטרים האלו חשובים לנו ביצירת

המודל, הסבר בהמשך (וגם בבסיס הנתונים.)

,glasses_train_dir, no_glasses_train_dir

בפעולה אנו מגדירים את המשתנים

glasses_validation_dir no_glasses_validation_dir, שכל אחד מהם מכיל כתובת של תיקייה שמכילה תמונות לפי מחלקה, בtrain או בvalidation (בהתאמה.) בשביל לקבל את כמות התמונות בכל תיקייה, השתמשתי בפעולה listdir של הספרייה ,os שמכלילה את כל הקבצים

שבdirectory בתוך רשימה .)list(כאשר אנו משתמשים בlen בשביל לבדוק את אורך הרשימה, זהו גם כמות הקבצים (תמונות) שבתיקייה. כך מצאתי את מספר התמונות בכל

תיקייה ולחבר אותם בהתאמה בשביל לקבל את train_samples ו,validation_samples

שאותם הפעולה מחזירה.

קובץ vars.py

<u>תפקידו:</u> תפקידו של הקוד הוא לשמור את המשתנים העיקריים והבסיסיים שנשתמש בהם לאורך כל הפרויקט.

<u>מיקום:</u>scripts תקייה

המשתנים הוסברו בקובץ Amit_glasses.py.

קובץ saved_history.npy

<u>תפקידו:</u> תפקידו של קובץ זה הוא לשמור את המשתנה history כדי שאינינו נצטרך לעבור ולאמן את המודל בכל פעם מחדש והוא ישמור את המשתנים המתאימים לפרויקט.

<u>מיקום:</u>Glass_No_glasses_project תקייה

קובץ saved_weights.h5

<u>תפקידו:</u> תפקידו של קובץ זה הוא לשמור את המשתנים שמצאנו שמתאימים לפרויקט שלנו ה"משקלים" כדי שאינינו נצטרך לעבור ולאמן את המודל בכל פעם מחדש.

<u>מיקום:</u>Glass_No_glasses_project תקייה

מדריך למשתמש

לפני שאציג את האופן שבו הפרויקט מתממש, אציג מדריך המראה למשתמש כיצד לתפעל את התוכנית:

הוראות התקנה

1 יש להוריד python 3.7 או גרסה עדכנית יותר:

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

אם למשתמש יש python מותקן על המחשב. יש לוודא שהגרסה היא 3.7 או יותר, בעזרת הפקודה **python -V**

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

2 יש להתקין במחשב את סביבת העבודה Anaconda
לינק: https://www.anaconda.com/products/individual

3 יש להוריד את ספריות הקוד הבאות שבהן משתמש הפרויקט:

שם ספרייה	הוראת התקנה
tensorflow	pip install tensorflow
keras	pip install keras
numpy	pip install numpy
matplotlib	pip install matplotlib
Cv2	pip install opencv-python
Zipfile	Pip install hazm

4 יש להוריד מחשבון הgithub שלי את קובץ הzip שמכיל בתוכו את הפרויקט.

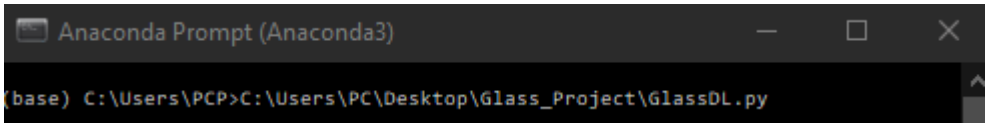
לינק:

לאחר הורדת קובץ הzip יש לחלץ את התיקייה Glass_Project אל directory שאינו מכיל אותיות עבריות

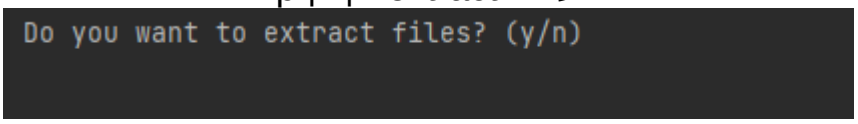
אין לערוך שינויים בקבצים למעט הוצאתם מקובץ הZIP!

הרצת התוכנית

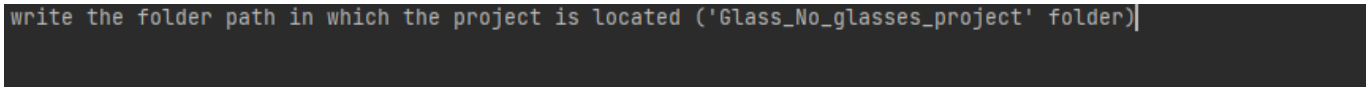
יש להריץ בCommand Line של סביבת העבודה Anaconda)Anaconda Prompt(את קובץ הpython: GlassDL.py ביחד עם המיקום המדויק שלו במחשב המשתמש.
דוגמה: C:\Users\PC\Desktop\Glass_Project\GlassDL.py



המשתמש בוחר האם לעשות extract לקובץ zip שהוריד.



המשתמש מתבקש לכתוב את הכתובת המלאה של התיקייה של הפרויקט Glass_No_glasses_project



המשתמש חייב לכתוב את התיקייה הנכונה, אחרת התוכנית לא תעבוד בצורה תקינה.

לאחר בחירת התיקייה, המשתמש יצטרך לבחור בין שתי אפשרויות:

1. לאמן את המודל בעצמו, התהליך הוא הארוך יותר משתי האופציות. אם ייבחר באפשרות הזו, המשתמש יוכל לראות כמה זמן לוקח לתוכנית לאמן את המודל עבור כל epoch ויצטרך לחכות עד שהתהליך יסתיים על מנת להמשיך. על מנת לבחור באפשרות זו, המשתמש צריך להקיש את האות n ואז ללחוץ Enter.

```
Do you wish to load a pre-trained model, and skip the training process? (y=yes/n=no)n
2021-06-13 22:29:58.026672: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes are enabled (registered 2)
Epoch 1/10
2021-06-13 22:30:20.898702: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
2021-06-13 22:30:22.182455: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 80289792 exceeds 10% of free system memory.
2021-06-13 22:30:22.242184: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
2021-06-13 22:30:24.827678: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
2021-06-13 22:30:25.945659: W tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 80289792 exceeds 10% of free system memory.
79/79 - 352s - loss: 0.5819 - accuracy: 0.7540 - val_loss: 0.0506 - val_accuracy: 0.9832
Epoch 2/10
79/79 - 327s - loss: 0.3566 - accuracy: 0.9570 - val_loss: 0.0378 - val_accuracy: 0.9936
Epoch 3/10
79/79 - 334s - loss: 0.0977 - accuracy: 0.9769 - val_loss: 0.0707 - val_accuracy: 0.9904
Epoch 4/10
79/79 - 325s - loss: 0.0825 - accuracy: 0.9793 - val_loss: 0.0698 - val_accuracy: 0.9936
 1/40 [..............................] - ETA: 1:11 - loss: 3.8102e-06 - accuracy: 1.0 2/40 [>.............................] - ETA: 1:11 - loss: 1.9232e-05 -
 accuracy: 1.0 3/40 [=>............................] - ETA: 1:05 - loss: 3.6640e-05 - accuracy: 1.0 4/40 [==>...........................] - ETA: 1:03 - loss
: 0.0019 - accuracy: 1.0000 40/40 [==============================] - 77s 2s/step - loss: 0.0691 - accuracy: 0.9937
```

)בתמונה הנ"ל ניתן לראות את תהליך האימון של המודל, לפי ההתקדמות בepochs

2. להשתמש בערכים מוכנים מראש. האופציה מאפשרת לדלג על אימון המודל ועל הזמן הארוך יותר שהוא לוקח, כך שהמודל כבר מאומן מראש. האפשרות הזו קצרה בהרבה מהאפשרות האחרת. על מנת לבחור באפשרות זו, המשתמש צריך להקיש את האות y ואז ללחוץ Enter

```
 1/40 [..............................] - ETA: 12:19 - loss: 9.6982e-06 - accuracy: 1.00002021-06-13 23:46:31.388322: W
 tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
 2/40 [>.............................] - ETA: 1:10 - loss: 1.3274e-04 - accuracy: 1.0000 2021-06-13 23:46:33.363142: W
 tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
 3/40 [=>............................] - ETA: 1:11 - loss: 1.4790e-04 - accuracy: 1.00002021-06-13 23:46:35.450134: W
tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
 4/40 [==>...........................] - ETA: 1:10 - loss: 1.4499e-04 - accuracy: 1.00002021-06-13 23:46:37.116017: W
tensorflow/core/framework/cpu_allocator_impl.cc:80] Allocation of 160579584 exceeds 10% of free system memory.
40/40 [==============================] - 87s 2s/step - loss: 0.0653 - accuracy: 0.9948
```

)בתמונה הנ"ל ניתן לראות את התהליך של הטענת ערכי המודל המוכן לתוך מודל התוכנית.(

ההבדל העיקרי בין שתי האפשרויות הוא הזמן שלוקח התהליך. אין שינוי במודל או בתפקודו.

בחלק השלישי של התוכנית, המשתמש יכול להביא למודל תמונה ולתת לו לסווג אותה לפי האדם בתמונה - מרכיב משקפיים או לא.

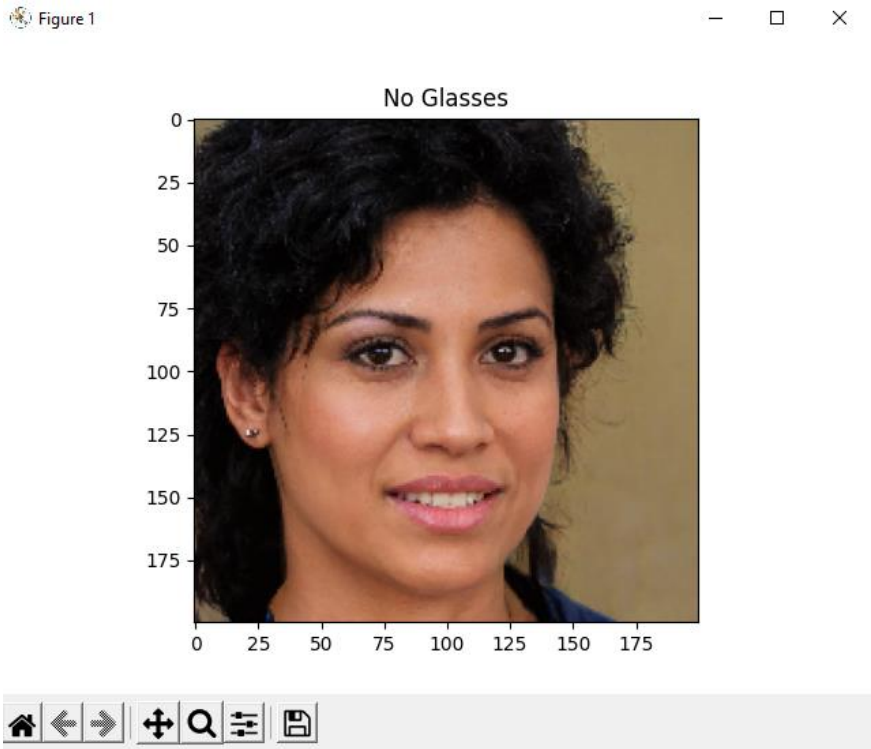
התוכנית תציג שאלה השואלת את המשתמש האם הוא רוצה 'לעשות חיזוי', כלומר לבחור תמונה ולבחון בעזרתה את המודל.

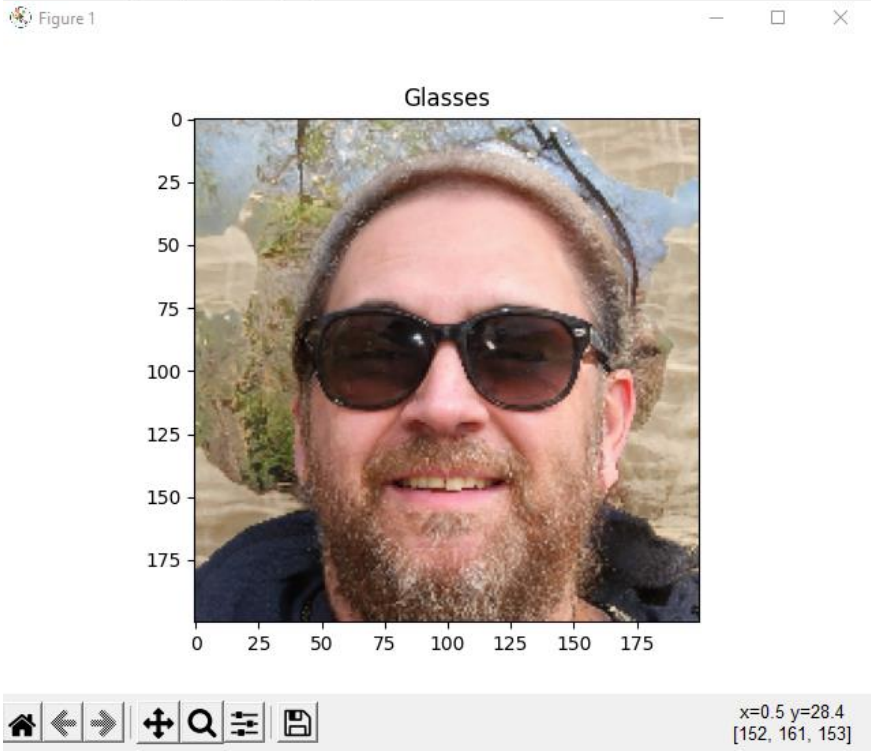
```
Do you want to make a prediction on an image? (y=yes/n=no) y
write an image path to test (PNG/JPEG ONLY)
```

אם המשתמש מקיש את האות y ואז לוחץ Enter, הוא ייבחר לבחון את המודל עם תמנה.

עכשיו המשתמש צריך לבחור תמונה שירצה לתת למודל ולכתוב את הכתובת המלאה שלה עם השם שלה ניתן לבחור תמונות מהסוגים: PNG, JPEG. כדאי לבחור תמונות מתיקיית הValidation שנמצאת בתוך תיקיית הDataset שבתוך תיקיית הפרויקט Glass_No_glasses_project.

לאחר שהמשתמש בוחר את התמונה, המשתמש יצטרך לכתוב את המיקום של התמונה שהוא רוצה לחזות, המודל יחזיר האם האדם בתמונה מרכיב משקפיים או לא.
בשביל להמשיך בתוכנה, המשתמש צריך לסגור את חלון התמונה שנפתח לו.





אם המשתמש מקיש את האות n ואז לוחץ Enter, לא יהיה חיזוי תמונה והתוכנה תיכבה (סיום התוכנה).

```
Do you want to make a prediction on an image? (y=yes/n=no) n

Process finished with exit code 0
```

המשתמש יכול לחזות כמה תמונות שירצה: **החלק השלישי** יחזור על עצמו עד שהמשתמש יקיש את האות n בבחירה שהוצגה קודם. אם המשתמש רוצה לחזות כמה תמונות, הוא צריך כל פעם להקיש y בבחירה הנ"ל ולעבור את תהליך בחירת התמונה מחדש.

רפלקציה

העבודה על הפרויקט היה מעניין וחשובה לפיתוח הידע שלי במדעי המחשב, אני נהנתי לעבוד על הפרויקט בגלל שזה עניין אותי ללמוד חומר חדש. קיבלתי מהעבודה כלים לתחום המחשבים ודרכים ללמוד לבד ועצמאית את החומר הנלמד.

הכלים שאני לוקח עם עצמי לעתיד מהעבודה הם כלים שאי אפשר להשיג בשום דרך אחרת חוץ מלהתאמן כמו למידה על חומר חדש או כתיבת קוד ומוטיבציה לסיים את מה שהתחלתי.

היו המון קשיים בנוגע לעבודה אחד מהקשיים הגדולים היו ללמוד את חומר חדש כמעט עצמאי עם עזרה קטנה ומהמורה וחברים, עוד קושי שהיה לי היה ניהול זמן נכון, נתקעתי על דברים הרבה דברים למשך המון זמן וזה עיכב לי את סיום הפרויקט.

המסקנות שלי מהעבודה הן שתחום הדיפ לרנינג הוא עולם גדול מאוד וחשוב מאוד ללמוד אותו מכיוון שזהו העתיד.

אילו הייתי מתחיל היום לעבוד על הפרויקט הייתי משתדל ללמוד יותר ביסודיות את החומר הנלמד ואיך הוא עובד בדיוק כדי שאוכל לייעל את עבודתי בפרויקט ואת הזמן שלי.

במידה והייתי מנהל את זמני נכון עבודתי הייתה יכולה להיות הרבה יותר יעילה.

ביבליוגרפיה

Keras Blog, Francois Chollet, Building powerful image classification models using very little data

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

Machine Learning Mastery, Jason Brownlee, How to Visualize a Deep Learning Neural Network Model in Keras

https://machinelearningmastery.com/visualize-deep-learning-neural-network-/model-keras

How to Choose Loss Functions When Training Deep Learning Neural Networks

https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-/deep-learning-neural-networks

Display Deep Learning Model Training History in Keras

https://machinelearningmastery.com/display-deep-learning-model-training-history-/in-keras

TensorFlow Tutorials, Image classification

https://www.tensorflow.org/tutorials/images/classification

Towards Data Science, Binh Phan, 10 Minutes to Building a CNN Binary Image Classifier in TensorFlow

https://towardsdatascience.com/10-minutes-to-building-a-cnn-binary-image-classifier-in-tensorflow-4e216b2034aa