פרויקט גמר 5 יחידות לימוד במחשבים DL



שם בית הספר: מקיף י"א ראשונים

שם עבודה: זיהוי כלבים על ידי שימוש ב deep learning

ת.ז תלמיד: 325049237

שם מנחה: דינה קראוס

שם החלופה: פרויקט במחשבים בהתמחות deep learning

תאריך הגשה: 08/06/2020





תוכן עניינים

מבוא:..............................................................................3

מדריך למשתמש:...............................................................6

מדריך למפתח:.................................................................11

מסקנות הרצת המודל.........................................................19

רפלקציה/סיכום אישי:.........................................................23

ביבליוגרפיה:.....................................................................23

נספחים:...........................................................................23

מבוא

רקע לפרויקט:

בתחילת השנה התבשרנו כי פרויקט 5 יח"ל במחשבים שלנו יעסוק בdeep learning כלומר מטרת הפרויקט היא "ללמד" את המחשב לזהות דברים.

מטרת הפרויקט:

מטרת הפרויקט הוא להגיע למצב שהמחשב יודע לזהות כלב. כלומר, מצב שבו המשתמש נותן(כקלט) למחשב תמונה של כלב, והמחשב צריך להוציא הודעת פלט שאכן זוהה כלב (למשל- it's a dog!), בדיוק של לפחות 80%.

כמו כן על המחשב לזהות גם תמונות שאינן כלב, למשל שהמשתמש נותן כקלט למחשב תמונה של חתול, או של גבר על המחשב להוציא הודעת פלט למשתמש שהתמונה שהוא קיבל אינה של כלב(דוגמה- it isn't a dog) כאמור בדיוק של לפחות 80%.

דרישות הרצת הפרויקט:

בעת ביצוע הפרויקט, השתמשתי בגרסת פייטון 3.7 במערכת הפעלה של 64 ביט. **יש לציין כי במחשבים ישנים אשר עובדים במערכת הפעלה של 32 ביט, ייתכן כי לא יהיה ניתן להוריד חלק מהספריות ההכרחיות להרצת הפרויקט**(tensorflow). הפרויקט משתמש בגרסה של 1.15 של tensorflow.

במידה והמחשב עומד בדרישות ההרצה, על המשתמש להוריד ספריות אשר הכרחיות להרצת הפרויקט- tensorflow ו keras.

כדי לבצע הורדות של ספריות אלה, על המשתמש לעשות פעולות אלה:

1. לפתוח את anaconda prompt
2. לכתוב(משמאל לימין) **pip install keras** ולחכות עד סיום ההורדה
3. לכתוב(משמאל לימן) **pip install dll-diagnostics** ולחכות לסיום ההורדה
4. לכתוב(משמאל לימן) **pip install tensorflow==1.15** ולחכות לסיום ההורדה

לאחר ביצוע שלבים אלה, המשתמש הוריד את כל הספריות לביצוע הפרויקט.

שלבי הרצת הפרויקט:

הרצת הפרויקט תתבצע ב anaconda prompt,ע"י גרירת הpath של הפרויקט למשל:

(base) C:\Users\User>python C:\Users\User\.spyder-py3\DLproject.py

מדריך למציאת path של קובץ מסוים- <https://www.youtube.com/watch?v=G4qyzclix10>. יש לשים לב כי יש לכתוב את המילה python לפני כתיבת ה path של הקובץ.

לאחר כתיבת שורה זו למחשב, ייפתחו בפני המשתמש כמה אופציות(עליהם ארחיב בפרק הבא). לאחר בחירה בכל אופציה, המשתמש יקבל אל המסך הודעה בהתאם.

כמו כן לאחר סיום של כל אופציה, המשתמש יקבל הודעה בהתאם. אורך ההרצה של כל אופציה בפרויקט שונה אך לא עולה על 20 דקות(במחשב סטנדרטי).

**ספריות הdata**

ספריות הdata הן רכיב חשוב בפרויקט, ספריות אלה מכילות את התמונות שבאמצעותן המחשב לומד לזהות כלבים(ולא כלבים) ובכך לממש את מטרת הפרויקט.

בפרויקט זה, יש 2 ספריות data ששמן dog וnot dog.

כלומר, בספרייה ששמה dog - ימצאו תמונות של כלבים, באמצעות ספרייה זאת ילמד המחשב ע"י מודל לזהות כלבים.

ובספרייה השנייה ששמה not dog- ימצאו תמונות לא של כלבים, כדי בעצם לבצע בקרה, ללמד את המחשב כי לא כל תמונה היא כלב. בהיעדר ספרייה שכזו, כל תמונה שהמשתמש יביא כפלט למחשב, המחשב יזהה תמונה זו כ-כלב. משום שבספריית ה-לא כלב לא ניתן לשים תמונה מכל אובייקט שקיים שהוא לא כלב, שמתי בספרייה זו תמונות של אובייקטים אשר **זהים לכלב**- למשל חתול, בני אדם או חיות אחרות כמו סוס, כבשה כדי לבחון את יעילות הפרויקט.

דוגמה לתמונה מספריית dog



דוגמה לתמונה מספריית not dog:



**הוראות להורדת ספריות הdata**

בטרם הורדת קבצי התמונות, על המשתמש לבצע את הפעולות הבאות:

1. לפתוח בכונן C תיקייה(folder) ולקרוא לה DeepLproject.
2. בתוך התיקייה הנ"ל לפתוח 2 תיקיות- train וtest-.
3. בתוך תיקיית ה-train יש לפתוח 2 תיקיות נוספות ששמן dog ו- not dog
4. בתוך תיקיית ה-train יש לפתוח תיקייה נוספת ששמה train

* **כל התיקיות שיש לפתוח הן תיקיות ריקות**.

לאחר פתיחת תיקיות אלה, יש להוריד את קבצי התמונות. כדי להוריד את קבצי התמונות הללו יש להיכנס לקישור הבא: <https://www.kaggle.com/yonatanerwr/project-data>

לאחר שנכנסים לקישור, יש ללחוץ על כפתור הdownload ולשמור את תיקיית ה zip שהורדה בכונן c.

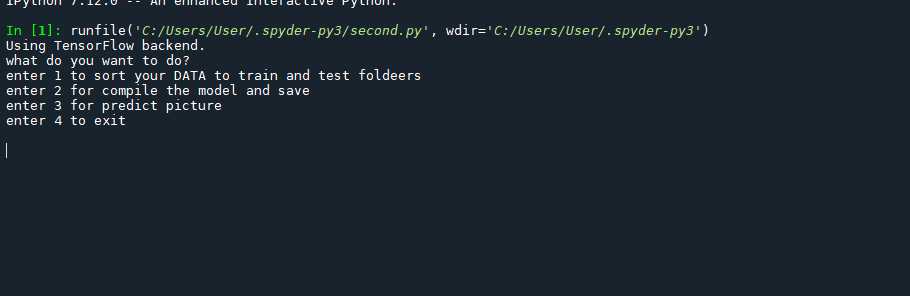
הערה: יש להוריד את התמונות בקישור דרך המחשב.

הערה נוספת: אל הקישור הנ"ל יש להיכנס דרך דפדפן גוגל כרום

**מדריך למשתמש**

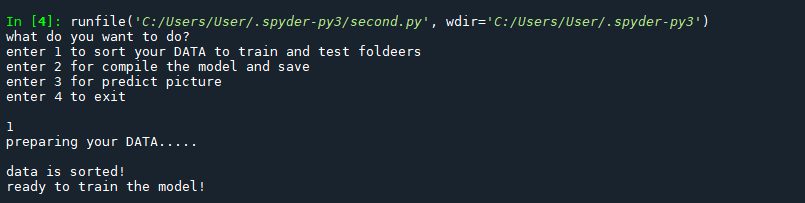
לאחר שבוצעו המשימות שתוארו בחלק הקודם, בטרם ביצוע אימון הפרויקט יש למיין בין התמונות(מהקישור בעמוד הקודם) שהורדנו אל התיקיות החדשות שיצרנו. כדי לטפל בבעיה זו, נכתב קוד שלאחר הורדת התמונות לכונן c ופתיחת התיקיות בכונן c ממיין את התמונות.

כדי לבצע פעולה זו, על המשתמש להריץ את הפרויקט כפי שהסברנו בפרק המבוא(גרירת ה path). לאחר ביצוע פעולה זו, ייפתח בפני המשתמש כמה אופציות כמתואר בצילום המסך הבא: (הצילום נלקח תוך הרצת הפרויקט בspyder- אך הרצה בanaconda prompt תראה אותו דבר)



הפעולה הראשונה שעל המשתמש לעשות לאחר שממשק האפשרויות נפתח הוא ללחוץ 1- בלחיצה על כפתור זה, הפרויקט ימיין בין התמונות שהורדנו לבין התיקיות שיצרנו ויאפשר את ההרצה של הפרוייקט.

לאחר שנלחץ על אופציה זו, נקבל אישור על כך שהפעולה אכן בוצעה- שהתמונות מויינו בתיקיות שיצרנו:



כך נקבל תיקיית train של כלבים- המכילה תמונות של כלבים, נשתמש בתיקייה זו כדי "לאמן" את המחשב ללמוד לזהות כלבים. בנוסף נקבל תיקיית train של לא-כלב, תיקייה המכילה תמונות למשל של חתולים ובני אדם. תיקייה זו משמשת כבקרה. כמו כן ניצור גם תיקיית test התכיל תמונות של כלבים כדי לבחון את המחשב- לבדוק באיזה אחוזים המחשב מצליח לזהות תמונות של כלבים.

**פתירת תקלות(אם נוצרו לאחר ביצוע פעולה 1)**

אם סידור התמונות לתיקיות כשל(כאשר הרצנו את הקוד אך קיבלנו הודעת שגיאה), יש לבצע את הפעולות הבאות:

1. למחוק את כל התיקיות שיצרנו

2. לפתוח תיקייה חדשה בכונן c ששמה deepLproject

2. להיכנס לקישור הבא: <https://www.kaggle.com/yonatanerwr/sorted-data> (יש ללחוץ על כפתור הdownload)

3. להיכנס לתיקייה שהורדה(ימצאו שם 2 תיקיות, train וtest)

4. יש להעביר תיקיות אלה לתיקייה שפתחנו בפעולה מספר 2.

5. ניתן להמשיך כעת לביצוע האפשרויות הבאות.

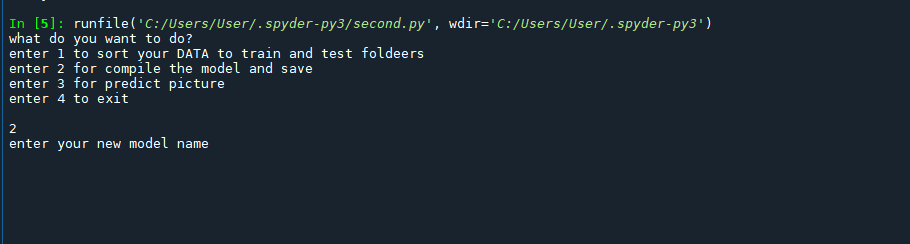
**הסבר על האפשרויות הנוספות**

כפי שניתן לראות בתמונה ובהקשר לנאמר בפרק הראשון, הפרויקט מציע מספר אפשרויות שניתן לבצע אותן בסדר מסוים.

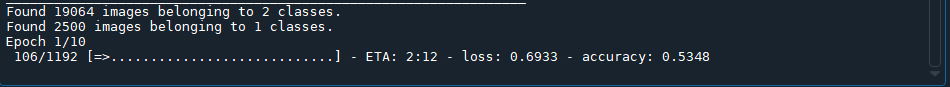
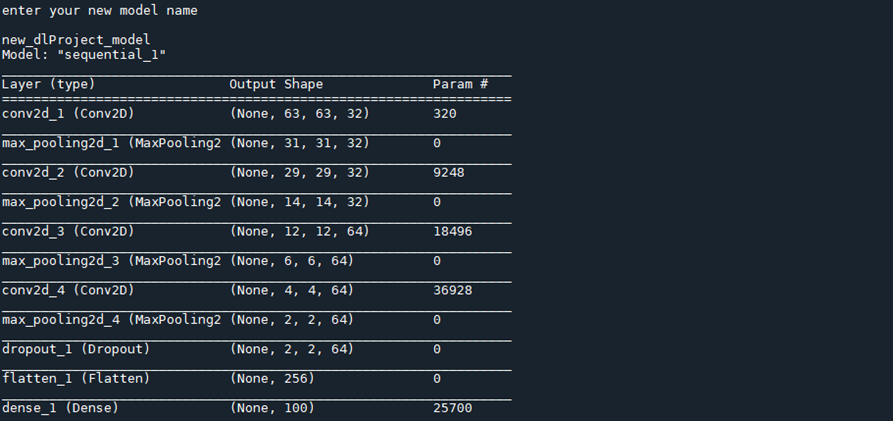
לאחר שלחצנו 1- והתמונות סודרו בתיקיות שלהן, המשתמש קיבל כפלט את ההודעה "ready to train your model". ומשמעותה היא שכעת ניתן לבצע את האפשרות הבאה והעיקרית שמציע הפרויקט והיא ליצור מודל כלומר, לבצע את תהליך הלמידה של המחשב- שבסיומו יבצע את מטרת הפרויקט- ידע לזהות תמונות של כלבים בדיוק של לפחות 80%. כמו כן, אפשרות זו שומרת את המודל שהורץ(על חשיבות הדבר נרחיב בהמשך).

**אופציה 2**

כדי לבצע אפשרות זו, בשעה שבה נפתח ממשק האפשרויות על המשתמש ללחוץ על הספרה 2. לאחר הלחיצה ייפתח בפני המשתמש המסך הבא:



על המשתמש כעת לכתוב את שם המודל כדי שהמחשב ידע לזהות את הרצת הפרויקט ולשמור את המודל(ואת תוצאות הזיהוי) לפי שם(יש צורך לכך כאשר המשתמש ירצה לבחור באופציה 3). לאחר שהמשתמש בחר שם כלשהו(**עדיף באנגלית**) ייפתח המסך הבא:

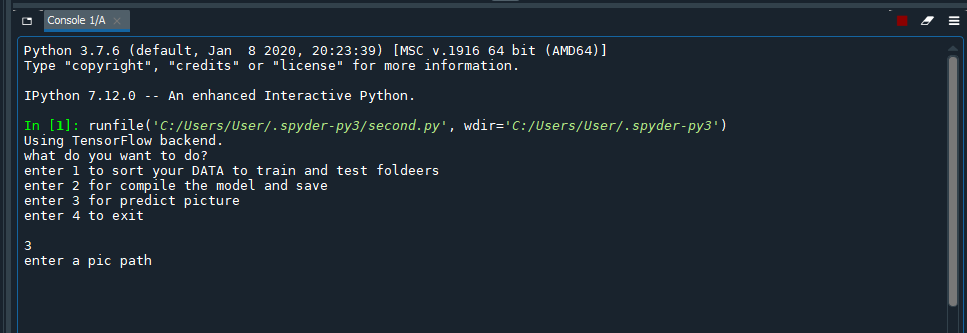


בחלק העליון של התמונה ניתן לראות את ה model summery כלומר, את שכבות המודל, ממה המודל מורכב. בחלק התחתון של התמונה ניתן לראות את תחילת האימון של המודל, כלומר תחילת תהליך הלמידה של המחשב. בפרק הבא ארחיב על משמעות המספרים המצויים בחלק זה של התמונה וכיצד ניתן לראות את תהליך הלמידה של המחשב.

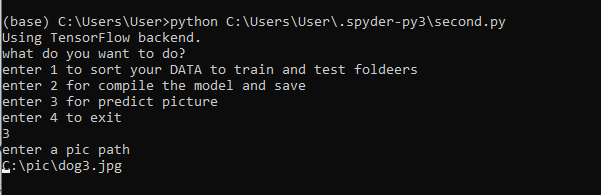
**אופציה 3**

לאחר סיום שלב זה, המשתמש יוכל לבחור באופציה 3 של הפרויקט בממשק המשתמש.

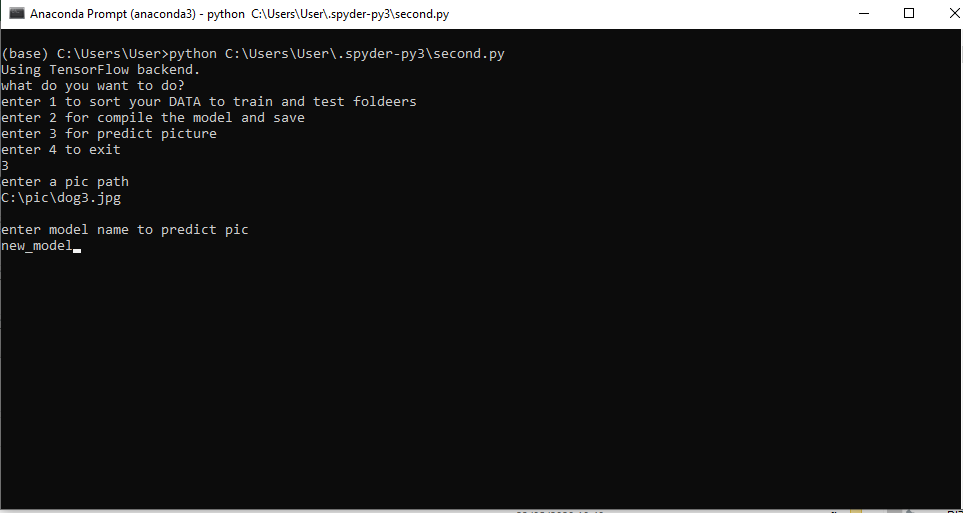
אופציה זו היא אופציית הpredict, כלומר פעולה שבוחנת את תהליך הלמידה של המחשב. באופציה זו, המשתמש ייתן למחשב path של תמונה של כלב/לא כלב והמחשב יחזיר כפלט, לאחר שסיים את תהליך הלמידה, את התוצאה:



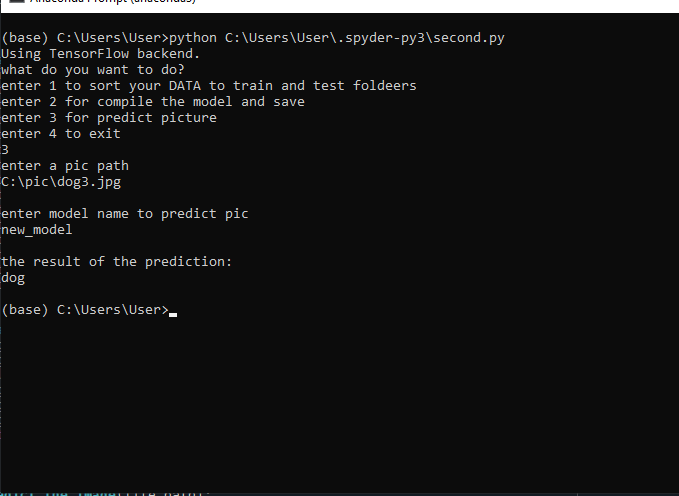
לאחר לחיצה על 3 כאמור יש לתת למחשב כקלט path של תמונה



לאחר הבאת הקלט, יהיה על המשתמש לתת כקלט נוסף למחשב את שם המודל שנשמר(השם שניתן באופציה 2 כאשר המשתמש בחר את שם המודל- נניח כי השם בתמונה הבאה(new\_model) הוא מודל שהורץ באופציה 2):



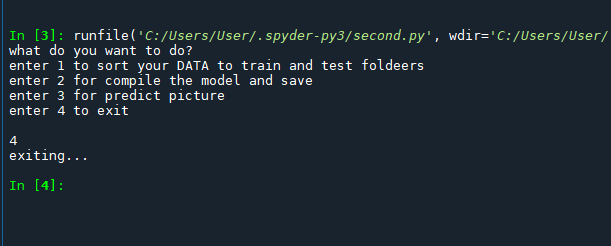
לאחר מכן, ניתן לראות ליד התמונה שהמחשב קיבל כקלט את תוצאת הזיהוי שלו:



במקרה זה ניתן לראות זיהוי נכון של המחשב- הובא לידו כקלט path שלתמונה של כלב("dog3.jpg") ותוצאת הזיהוי שלו הייתה אכן כלב.

**אופציה 4**

האופציה הרביעית היא כאשר המשתמש רוצה לצאת ממשק המשתמש/ רוצה לצאת מהרצת הפרויקט. לחיצה על ספרה 4 ובכך בחירה באופציה הרביעית והאחרונה יביא לסיום הרצת הפרויקט:



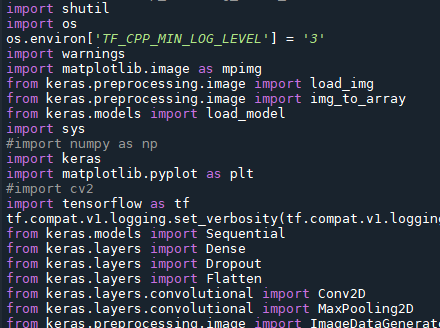
**מדריך למפתח**

בפרק זה ארחיב על פעולות הפרויקט(ה-def השונים) ועל מחלקות הפרויקט. על מטרת הפעולות/מחלקות ומה תפקידם בפרויקט. כלומר, חלק התכנות בפרויקט.

**הפעולות השונות(def):**

**1. ספריות הפרויקט**

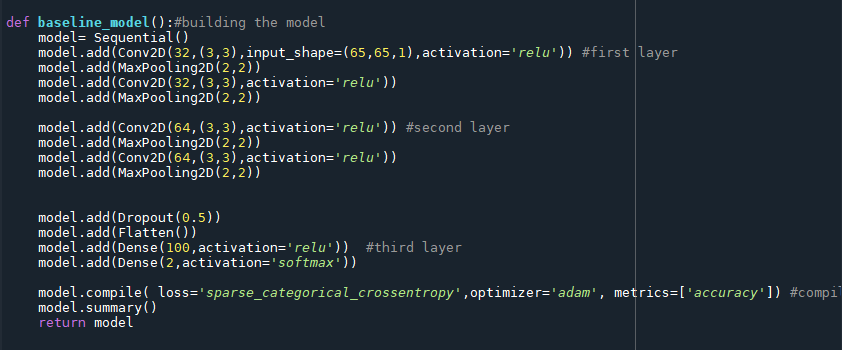
קטע הקוד הבא אינו פעולה, אך חשוב להסביר את תפקידו:



כדי שיהיה ניתן להריץ את הפרויקט יש לייבא את הספריות הללו. ייבוא ספריות אלו הכרחי כיוון שהם מאפשרות ביצוע פעולות כמו למשל הדפסת תמונה. ללא ייבוא כל הספריות הללו יתבצע בקטע הקוד שגיאת קומפילציה(הידור).

**2. פעולת בניית המודל**

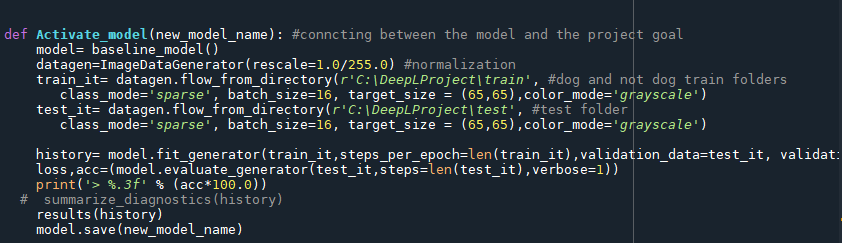
פעולה זו היא החשובה ביותר בפרויקט והיא פעולת יצירת המודל.



בעצם, יצירת המודל היא יצירת תהליך הלמידה של המחשב לזיהוי הכלבים ולא כלבים(במקרה זה של הפרויקט). ככל המודל יהיה טוב יותר, כך תהליך הלמידה של המחשב לזיהוי כלבים ולא כלבים יהיה יותר טוב כלומר, אחוז הדיוק של המחשב בזיהוי התמונות יהיה גבוה יותר(ידע להבדיל בין תמונות של כלב ושלא לא-כלב יותר טוב).

**3. פעולת הפעל המודל**

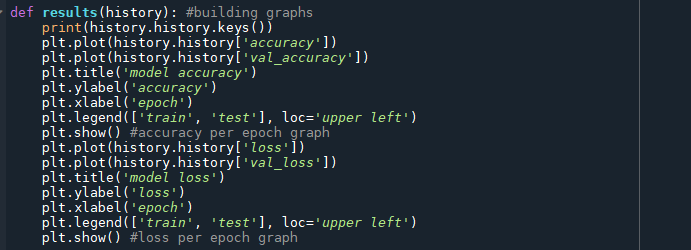
פעולה זו מקשרת בין התמונות שלנו(של הכלבים ושל הלא-כלבים) לבין המודל. כלומר, מפעילים את המודל על התמונות הללו ובכך המחשב מתחיל את תהליך הלמידה שלו בהתאם לתמונות שמובאות לידו בקטע הקוד הבא.



פעולה זו מזמנת את הפעולה הקודמת שכתבתי עלייה(יצירת המודל) בשורה הראשונה ו"מלבישה" על מודל זה את התמונות הרצויות(של הכלבים ולא-כלבים) ע"י הבאת ה-path של תיקיות התמונות. כמו כן, פעולה זו מתאימה את גודל התמונות(שמצויות בתיקיות) לגודל התמונה שהמודל מקבל(למשל בפרויקט זה, גודל התמונה שנקבע הוא (65,65). בנוסף לכך, פעולה זו מחשבת את הדיוק הסופי של הפרויקט(ארחיב על כך בפרק הבא).

**3. עיבוד התוצאות**

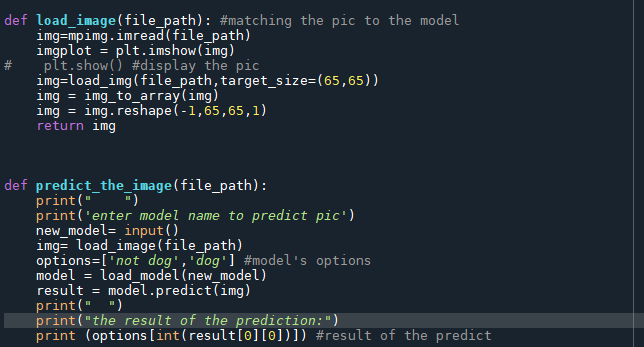
מטרת הפעולה הבאה היא ליצור גרפים המסכמים את תוצאות הפרויקט לידי המשתמש.



הגרפים שנוצרים הם אחוז הדיוק של הפרויקט במהלך תהליך הלמידה(תהליך הלמידה מחולק ל-10 חלקים שלאחר כל אחד מהם מבוצעת בדיקה אשר מראה את אחוז הדיוק הנוכחי של הפרויקט בזיהוי התמונות).

**4. יצירת פעולת הpredict(אופציה 3 בממשק המשתמש)**

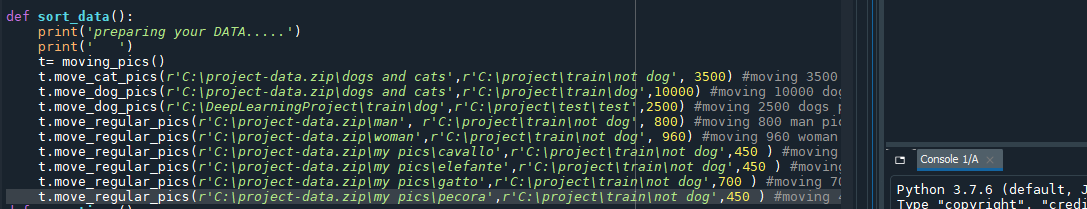
על פעולת ה- predict בממשק המשתמש הסברנו בפרק הקודם. ליצרית האופציה הזו יש צורך בכתיבת שתי הפעולות הבאות:



כאמור, בפעולה זו, המשתמש מביא למחשב כ-קלט path של תמונה, והמחשב צריך לקבוע האם התמונה שהובאה לידו הינה כלב או לא כלב. הפעולה העליונה מטרתה להתאים את גודל התמונה שקיבל המחשב כקלט לגודל התמונה שהמודל שיצרנו עובד אתו(בפרויקט זה כאמור גודל התמונה שנקבע הוא (65,65). מטרת הפעולה התחתונה היא לקשר בין התמונה שקיבל המחשב כpath לבין מודל שהצרנו ושמרנו(אופציה 2 בממשק המשתמש). בעצם, הפעולה התחתונה מפעילה את המודל שכבר יצרנו על התמונה היחידה הזו והמשתמש מקבל הודעה עם תוצאת הזיהוי של המחשב.

**5. מיון התמונות לתיקיות(אופציה 1 בממש' המשתמש)**

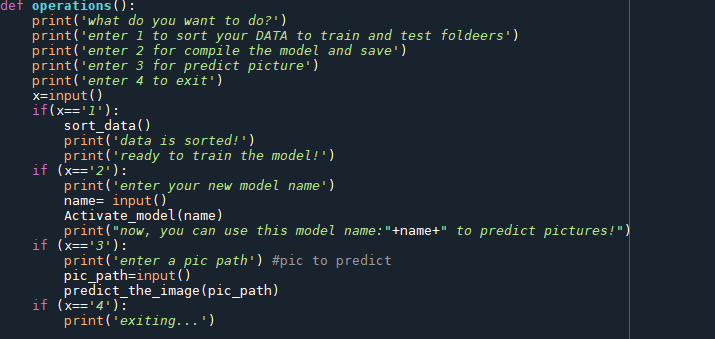
בפרק הקודם הסברנו על אופציה 1 של המשתמש בממשק המשתמש. הפעולה הראשונה שעל המשתמש לבצע בטרם הרצת האופציות האחרות של הפרויקט. פעולה זו מתבצעת בשל קטע הקוד הבא.



בנוסף, קטע קוד זה מדפיס הודעות בהתאם למשתמש- שמטרת הפעולה אכן בוצעה והתמונות סודרו בתיקיות שלהן.

**6. פעולת יצירת ממשק המשתמש**

בקטע הקוד שבפעולה זו, הפעולה מזמנת את כל הפעולות שכתבנו בפרק זה ובכך יוצרת את ממשק המשתמש.

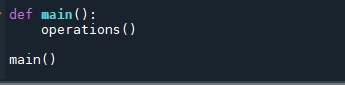


ניתן תחילה לראות כי פעולה זו מדפיסה את האפשרויות שעומדות בפני המשתמש ולאחר מכן מזמן את הפעולה הרצויה בהתאם לבחירת האופציה של המשתמש.

למשל, אם המשתמש בחר באופציה מספר 1(סידור התמונות בתיקיות השונות) אז הפעולה מזמנת את פעולת סידור מיון התמונות לתיקיות אשר מדפיסה הודעות למשתמש בהתאם כדי שידע מה מתרחש.

**7. הפעולה שב- main**

פעולת הmain בעצם מזמנת רק את פעולת יצירת ממשק המשתמש.



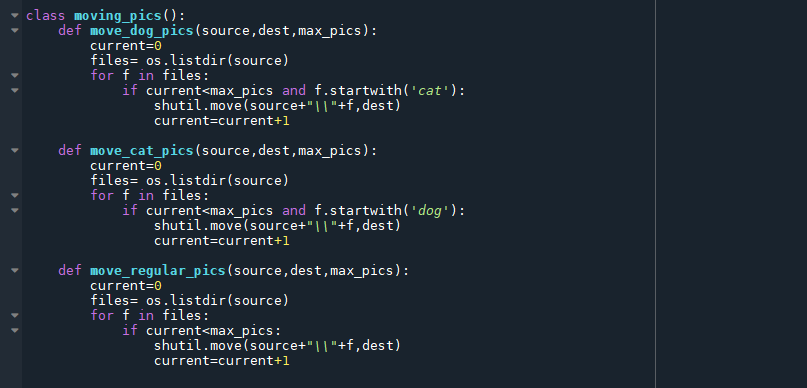
**מחלקת הפרויקט**

לאחר שהסברנו על הפעולות השונות אשר מצויות בפרויקט כעת אסביר על המחלקה שבפרויקט.

מטרת המחלקה שבפרויקט היא לדאוג לסידור התמונות לתיקיות(אופציה 1 בממשק המשתמש/פעולה מספר 5 כפי שהסברנו בפרק זה). פעולה מספר 5 בפרויקט זה מזמנת מחלקה זו והיא הפעולה היחידה אשר מזמנת ישירות את המחלקה.

קטע הקוד של המחלקה:

**שם המחלקה: moving\_pics**



נסביר את פעולות המחלקה באמצעות ממשק המחלקה הבא:

|  |  |  |  |
| --- | --- | --- | --- |
| שם הפעולה | מטרת המחלקה | הפרמטרים שהפעולה מקבלת | כיצד הפעולה מבצעת את מטרתה |
| Move\_dog\_pics(source,dest,max\_pics) | להעביר קובצי תמונות כלבים מתיקייה לתיקייה | **Source:**  תיקיית המקור של התמונות  **Dest**:  התיקייה שרוצים להעביר אליה את קובצי התמונות  **Max\_pics:**  מספר התמונות המקסימלי שרוצים להעביר | הפעולה מקבלת path של תיקייה המכילה תמונות. הפעולה מעבירה תמונות ששם התמונה שלהם מתחיל עם המילה dog(למשל dog1.jpg) לתיקיית יעד ריקה. למשתמש יש אופציה לבחור מספר מקסימלי של תמונות שניתן להעביר- **למשל** להעביר מתיקייה המכילה 25,000 תמונות של כלבים וחתולים 12,500 תמונות של כלבים. |
| Move\_cat\_pics(source,dest,max\_pics) | להעביר קובצי תמונות חתולים מתיקייה לתיקייה | **Source:**  תיקיית המקור של התמונות  **Dest**:  התיקייה שרוצים להעביר אליה את קובצי התמונות  **Max\_pics:**  מספר התמונות המקסימלי שרוצים להעביר | הפעולה מקבלת path של תיקייה המכילה תמונות. הפעולה מעבירה תמונות ששם התמונה שלהם מתחיל עם המילה cat (למשל cat1.jpg) לתיקיית יעד ריקה. למשתמש יש אופציה לבחור מספר מקסימלי של תמונות שניתן להעביר- **למשל** להעביר מתיקייה המכילה 25,000 תמונות של כלבים וחתולים 12,500 תמונות של חתולים. |
| Move\_regular\_pics(source,dest,max\_pics) | להעביר תמונות מתיקייה אחת לאחרת | **Source:**  תיקיית המקור של התמונות  **Dest**:  התיקייה שרוצים להעביר אליה את קובצי התמונות | הפעולה מקבלת path של תיקייה המכילה תמונות וpath של תיקייה נוספת. הפעולה מעבירה מספר תמונות(שנקבע ע"י המשתמש) מהתיקייה הראשונה לשנייה. |

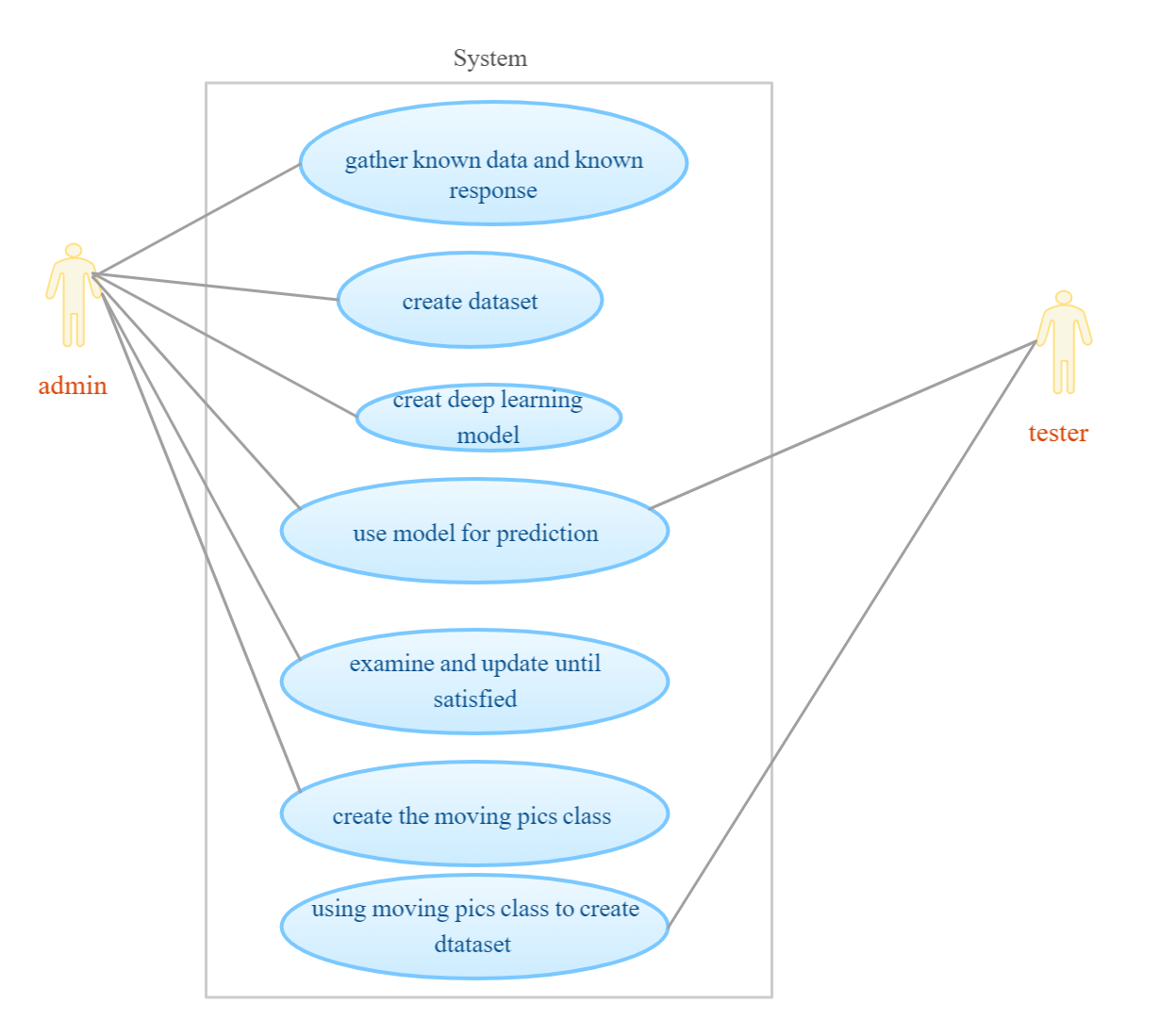
**הערות לגבי המחלקה**

* בשתי הפעולות הראשונות- אשר מעבירות תמונות של כלבים/חתולים מתיקיית המקור לתיקייה אחרת:

המחשב יודע להעביר תמונה של כלב/חתול לפי שם התמונה. בdataset שבו השתמשתי, תמונות החתולים נקראו cat1.jpg, cat2.jpg וכך גם הכלבים ולכן ניתן לדעת איזה תמונה היא של כלב/חתול ולהעבירה לתיקייה הנדרשת.

* בשלוש הפעולות שבמחלקה, ישנו המשתנה max\_pics שמטרתו לתת למשתמש לקבוע כמה תמונות להעביר מתיקייה אחת לשנייה. יש לציין כי לפעמים המספר התמונות שירצה המשתמש להעביר מתיקייה אחת לשנייה לא יהיה מספר התמונות שיעברו בפעול כיוון שלפעמים לא קיים המספר התמונות שהוא רוצה להעביר בתיקייה. למשל- המשתמש רוצה להעביר מתיקייה אחת לאחרת 10,000 תמונות של כלבים, אך ייתכן כי בתיקיית המקור(התיקייה שממנה מעבירים את התמונות) אינה מכילה כלל 10,000 תמונות של כלבים, לכן הפעולה תעביר את מספר תמונות הכלבים הקרוב ביותר ל-10,000. לדוגמה, אם המשתמש רוצה להעביר מתיקייה אחת לאחרת 10,000 תמונות של כלבים אך התיקייה מכילה רק 8,500 תמונות של כלבים, הפעולה תעביר 8,500 תמונות של כלבים מתיקיית המקור.

**סיכום פעולות משתמש ופעולות יוצר הפרויקט**



לפי התרשים:

**פעולות יוצר הפרויקט:**

1. לאסוף תמונות המתאימות לפרויקט

2. ליצור את הdata מכל התמונות שנאספו(ליצור מאגר תמונות של כלבים ולא-כלבים)

3. לפתח את המודל שעליו מבסס המחשב את תהליך הלמידה שלו.

4. לבחון את אופציה 3 בממשק המשתמש(ה-predict)

5. לשנות ולעדכן את המודל עד הגעה לתוצאה המתאימה

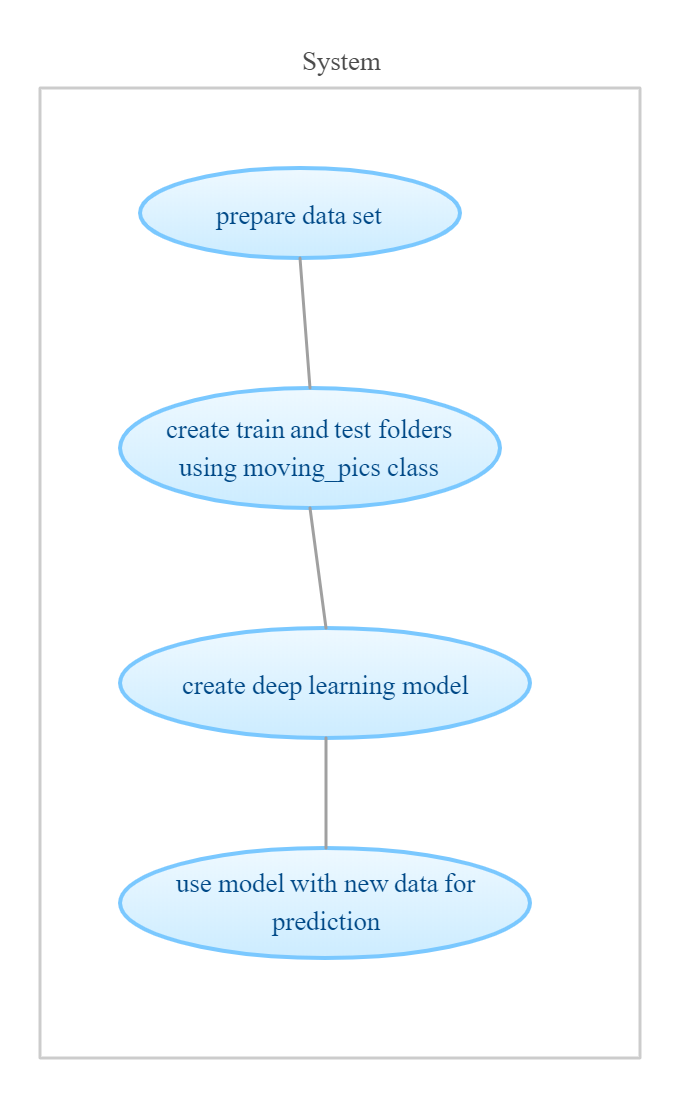
6. ליצור את מחלקת moving\_pics (שעלייה דיברנו קודם)

**פעולות בוחן הפרויקט:**

1. לבחון את המודל שנוצר ע"י שימוש באופציה 3 בממשק המשתמש

2. ליצור את הdata באמצעות שימוש במחלקת moving\_pics

**תרשים זרימה של הפרויקט(תהליך יצירת הפרויקט):**



הסבר: יצירת dataset---לסדר את התמונות בתיקיות הtrain וtest---לפתח את מודל הלמידה---לבחון את יעילות המודל ע"י שימוש באופציה 3 של ממשק המשתמש.

**מסקנות הרצת המודל**

פרק זה יעסוק ביעילות המודל האחרון שנוצר וההגעה למודל זה.

כדי להגיע למודל אופטימלי, יש להגיע לאיזון בין כמה גורמים:

1. **גודל התמונה**. ככל שגודל התמונה שנקבע בפרויקט גדול יותר, כך זמן הרצת הפרויקט יהיה גדול יותר אך אחוז הדיוק של המחשב במהלך תהליך הלמידה עשוי להיות גבוה יותר.

2. **התאמת המודל לתמונות**. גורם חשוב נוסף הוא להתאים את המודל שיצרנו לתמונות. כלומר, שהמחשב יבצע תהליך למידה טוב על הdataset

3. **אחוז דיוק הסופי לאחר תהליך הלמידה**. הגורם האחרון והחשוב ביותר הוא האחוז הדיוק הסופי של הפרויקט. אחוז זה נקבע לאחר שהמחשב סיים את תהליך הלמידה שלו וכעת עובר על כל התמונות אשר הקצנו לבדיקת המודל. תוצאת הבדיקה הזו היא המשקפת ביותר ולכן לה חשיבות גדולה.

לפני ההגעה למודל הסופי, נערכו מספר מודלים ותיקון טעויות.

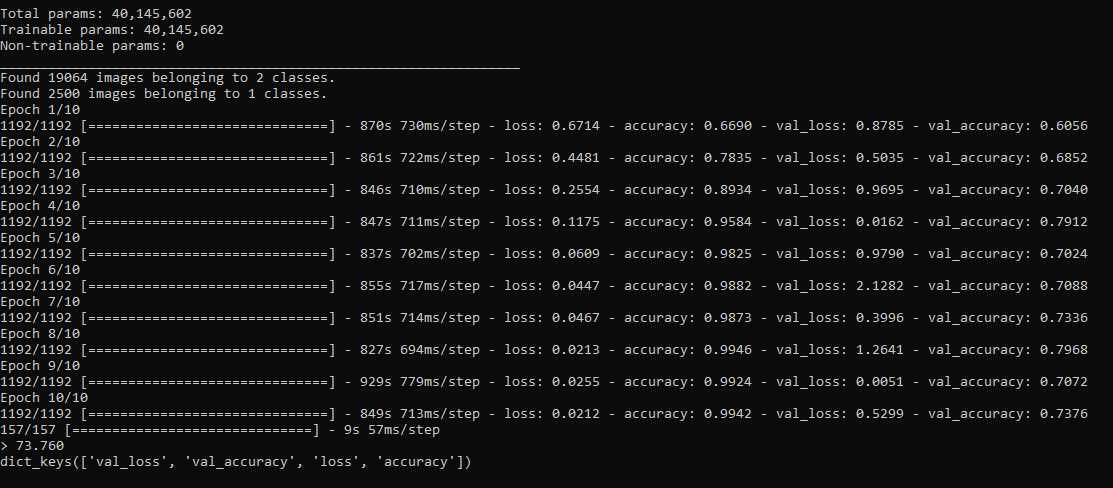
מבנה המודלים השונים שיוצגו בפרק זה יופיעו בנספחים.

**המודל הראשון שנוצר**

ניתן לראות כי תהליך הלמידה חולק ל10 חלקים(טעינות). לאחר כל חלק, המחשב מציג כפלט למשתמש את זמן ההרצה של כל חלק בשניות (הטור הראשון בתמונה), ואת אחוז הדיוק של כל טעינה(הטור השלישי בתמונה).

**בתחתית כל תמונה ניתן לראות את אחוז הדיוק הסופי לאחר תהליך הלמידה.(נמצא לאחר השורה שכתוב בה 157/157)**

המודל הראשון שנוצר היה מודל שעובד עם גודל תמונה של (200,200) ובו היו מספר בעיות:



1. **זמן ההרצה**

כאמור בטור הראשון ניתן לראות את זמן ההרצה של כל חלק בשניות. במודל הראשון כל חלק ארך בערך כ-850 שניות שהם כ-14 דקות. כלומר כל הרצת הפרויקט במודל הראשון אורכת כ-140 דקות. זמן הרצה זה מהווה בעיה קשה- קושי ביכולת שינוי במודל ובבדיקת תוצאת השינוי ובנוסף זמן ההרצה הארוך עלול להוות גם קושי לבודק הפרויקט.

**2**. **חוסר הלימה בין אחוז דיוק לאחר כל טעינה לבין אחוז הדיוק הסופי של הפרויקט**.

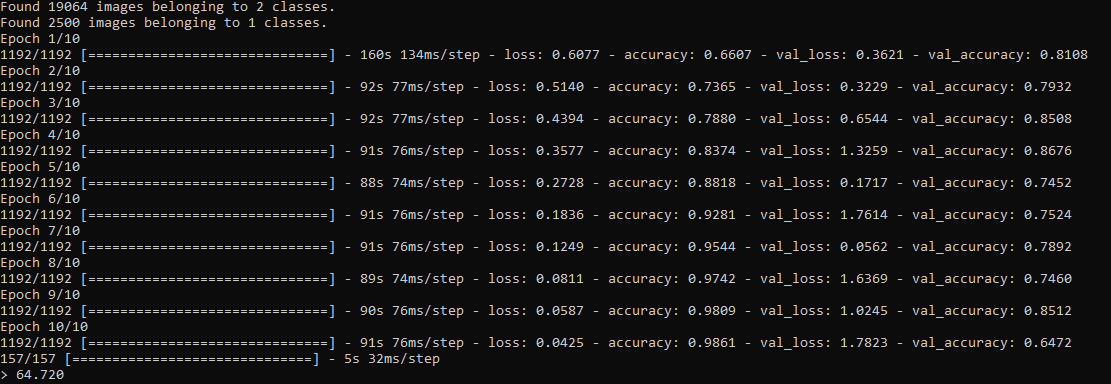
ניתן לראות בתמונה המצורפת, כבר מהחלק הרביעי בתהליך הלמידה, אחוז הדיוק הוא 95.8 ומגיע עד 99.4 בחלק האחרון של תהליך הלמידה אך ניתן לראות בתחתית התמונה את אחוז הדיוק הסופי לאחר סיום תהליך הלמידה.

במודל זה אחוז זה הוא 73.76 בלבד, לעומת אחוז גבוה ביותר בתהליך הלמידה.

לאחר הרצת המודל הראשון, הבנתי כי קודם יש להפחית את זמן ההרצה של הפרויקט, ולשם כך נוצר:

**המודל השני**

לאחר חיפוש באינטרנט, הסקתי כי שינוי גודל התמונה שאיתה המודל עובד עשוי להפחית משמעותית את זמן הרצת הפרויקט מבלי לפגוע באחוז הדיוק של הפרויקט בצורה דרסטית. לכן, שיניתי את גודל התמונה שאיתה המודל עובד ל(65,65) במקום (200,200) וקיבלתי את התוצאה הבאה(**יש להדגיש כי מבנה המודל לא שונה**):



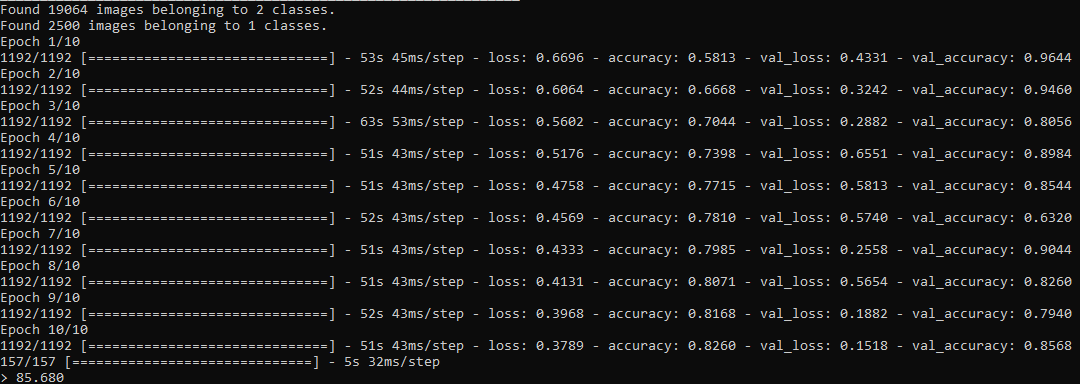
בהרצה זו ניתן לראות כי זמן ההרצה של כל חלק עמד על בערך 90 שניות כלומר דקה וחצי. לכן זמן ההרצה הכולל של הפרויקט הוא כעת 15 דקות לעומת 140 דקות קודם.

אך גם במודל השני שנוצר עדיין קיימת בעיית חוסר ההלימה בין אחוז הדיוק של החלקים(בחלק האחרון, המודל מגיע לדיוק של **98.61**) לבין אחוז הדיוק הסופי לאחר סיום תהליך הלמידה(**64.78** בלבד).

כדי לפתור בעיה זו הבנתי כי יש לשנות את מבנה המודל שהשתמשתי בו עד כה ולכן נעזרתי באינטרנט לחיפוש אחר מודל אחר.

לאחר הרבה הרצות שונות במודלים שונים הגעתי למודל אשר עומד בגרומים אשר הצגתי בתחילת הפרק

להלן המודל:



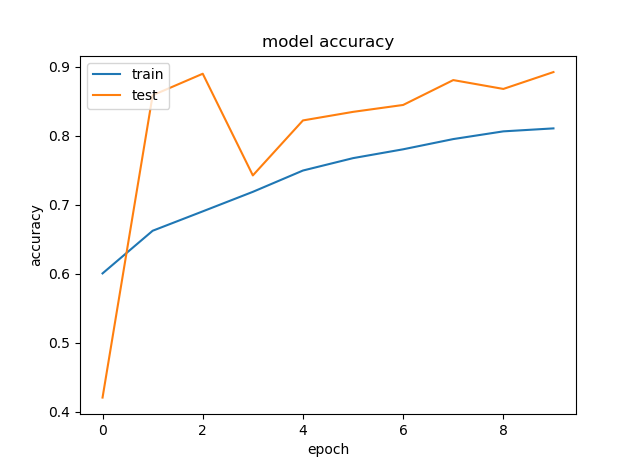
**מדוע המודל האחרון יעיל**

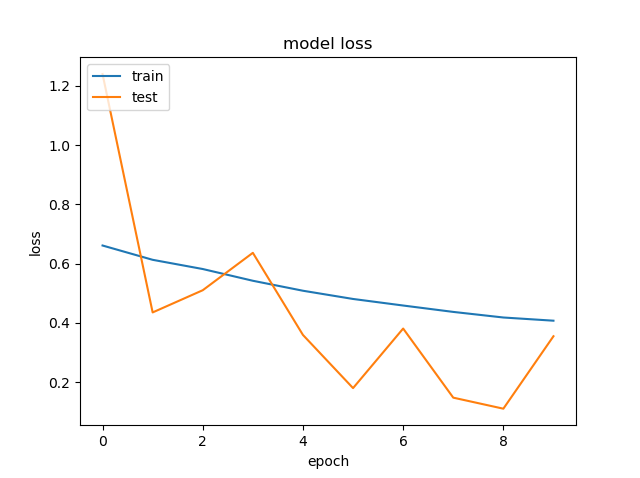
1. זמן הרצת המודל כלומר, זמן תהליך הלמידה של המחשב הוא קצר ביותר. הוא מורכב מ10 חלקים שאורך כל אחד מהם הוא בערך 50 שניות, כלומר כל אורך הרצת הפרויקט קטן מ10 דקות.

2. ניתן לראות כיצד אחרי כל חלק(מתוך ה10) אחוז הדיוק של המודל עולה(ניתן לראות זאת הטור השני). אחוז הדיוק לאחר החלק הראשון בתהליך הלמידה הוא 58.13 אחוז ואילו לאחר סיום החלק העשירי, אחוז הדיוק הוא 82.6. כלומר, ניתן להסיק מכך שהמחשב מצוי בתהליך הרצוי- הוא לומד, ולכן משפר את אחוז הדיוק שלו.

3. ניתן לראות כעת במודל האחרון את ההלימה בין אחוז הדיוק של החלקים לבין אחוז הדיוק הסופי לאחר סיום תהליך הלמידה, דבר שלא התרחש במודלים הקודמים. בסיום החלק האחרון במודל זה, אחוז הדיוק של המודל הוא 82.6 אחוז ואילו אחוז הדיוק הסופי לאחר סיום תהליך הלמידה הוא 85.68 ולכן ניתן להסיק כי המודל יפעל ויזהה תמונות לפי אחוז הדיוק שהמשתמש רואה כפלט.

**תוצאות הגרפים של המודל האחרון**

****

****

בגרף הראשון ניתן לראות כיצד האחוז דיוק של המודל עולה כתלות ב epoch(כל epoch הוא 1 מתוך ה10 החלקים המרכיבים את תהליך הלמידה). ובגרף השני ניתן לראות כיצד הmodel loss יורד כתלות ב epoch. זה מראה כי אין בעיה של overfitting במודל.

**רפלקציה**

נהנתי לבצע את הפרויקט השני כתחלופה של 5 יח"ל.

העשרתי את הידע שלי בתכנות ופיתחתי יכולות שונות אשר עזרו לי לבצע את הפרויקט

ביצוע הפרויקט היה מאתגר, אשר כלל הרבה שעות עבודה של ביצוע, חיפוש אחר dataset מתאים, מקורות מידע וניסיות לשיפור המודל.

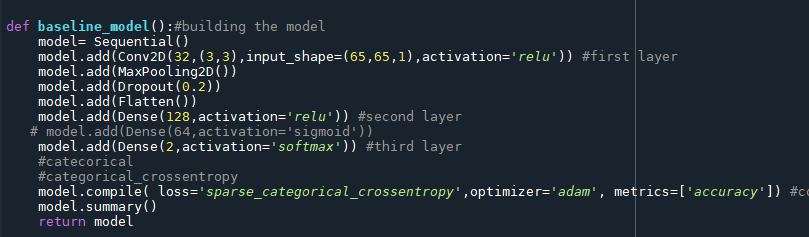
**ביבליוגרפיה**

נעזרתי באתר kaggle- רוב התמונות המהוות את הdataset נלקחו מאתר זה.

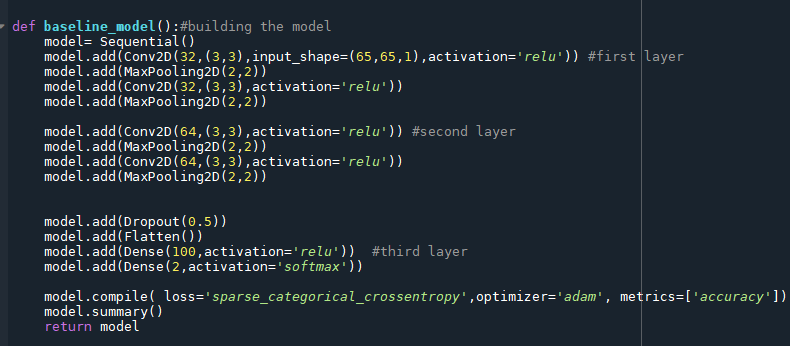
בנוסף נעזרתי בספר Deep Learning for computer vision-image classifaction כדי לבצע חלק מפעולות הפרויקט.

**נספחים**

מבנה המודל הראשון והשני(אשר מתוארים בפרק הקודם):

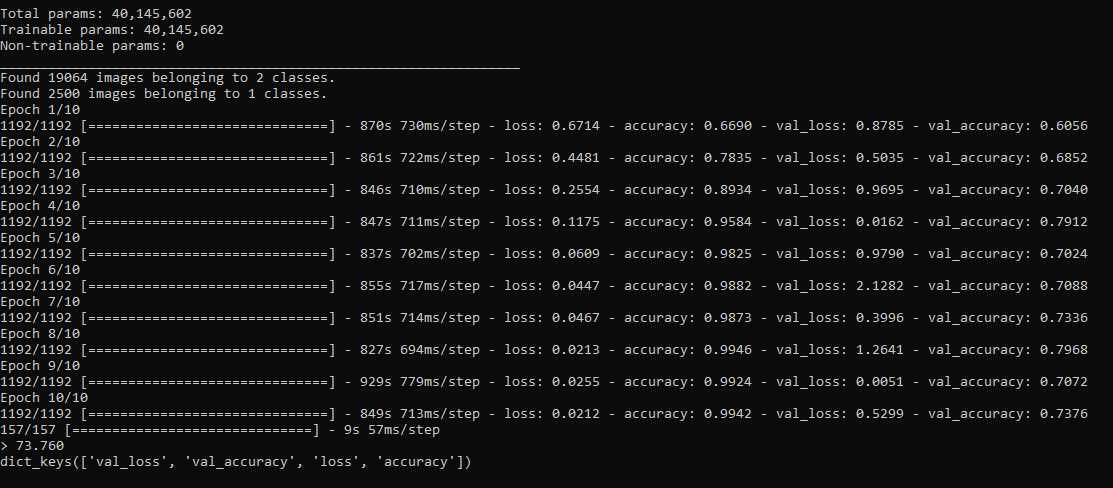


מבנה המודל האחרון:

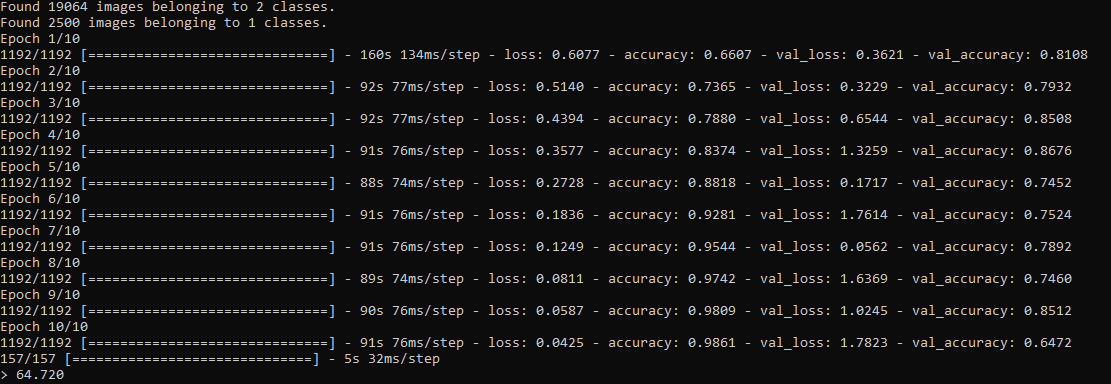


תוצאות הרצת המודלים

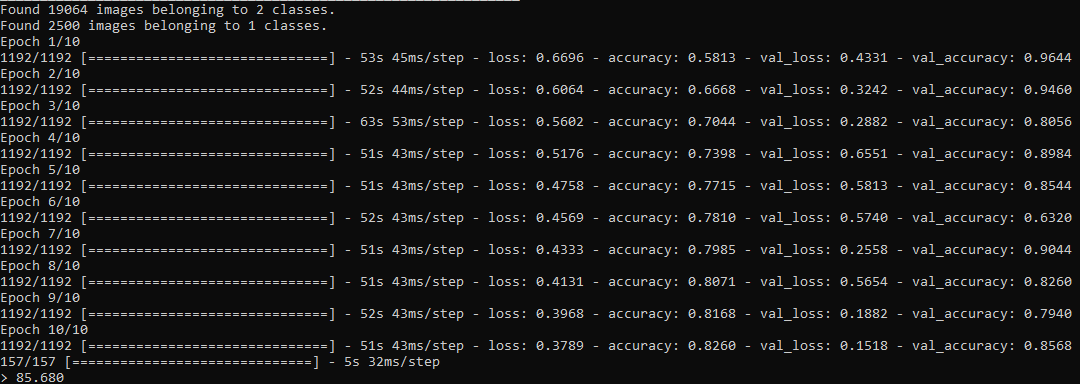
ה1:



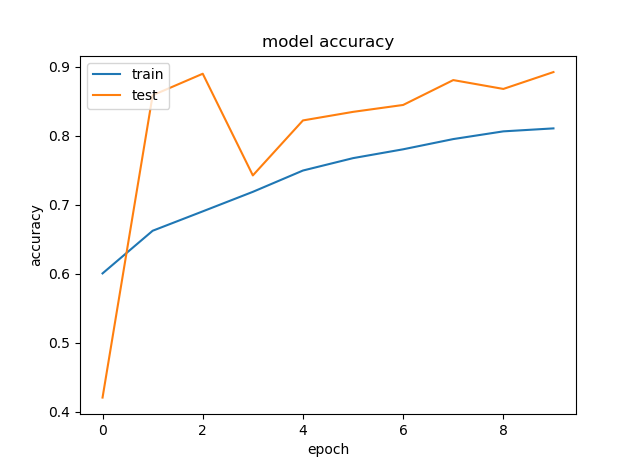
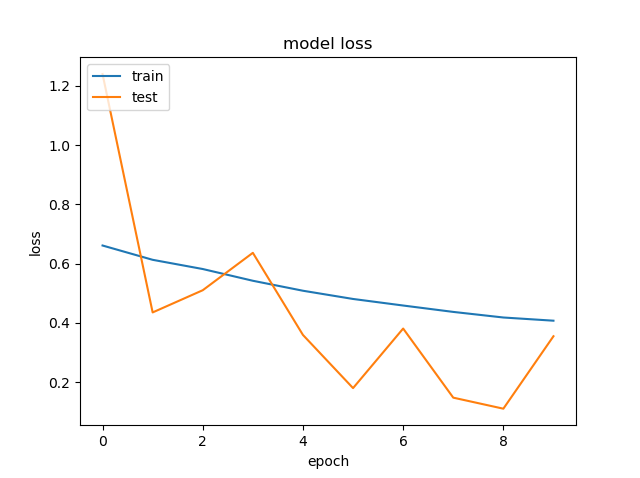
ה2:



האחרון:



גרפים של המודל האחרון:

****