פרוייקט גמר 5 יחידות לימוד

התמחות – תכנון ותכנות מערכות Deep Learning

זיהוי לוגואים של חברות מכוניות

מגיש: נדב ברוך

תז: 213123623

בית ספר: מקיף י"א ראשונים

ראשון לציון

מנחה: דינה קראוס

תאריך הגשה: 18.6.2020

תוכן עניינים:

1. מבוא....................................................................................ע"מ 3
2. מדריך למשתמש.................................................................ע"מ 4-9
3. מדריך למפתח................................................................ע"מ 10-11
4. מסקנות הרצת המודל.......................................................ע"מ 12-13
5. נספחים ..............................................................................ע"מ 14

**מבוא**

מטרת הפרויקט

לזהות סוגים שונים של לוגואים של חברות מכוניות מתוך תמונה.

המודל מסוגל לזהות 40 לוגואים שונים של חברות שונות והן כדלהלן:

"Alfa Romeo" "Audi" "BMW" "Chevrolet" "Citroen" "Dacia" "Daewoo"

"Dodge" "Ferrari" "Fiat" "Ford" "Honda" "Hyundai" "Jaguar" "Jeep"

"Kia" "Lada" "Lancia" "Land Rover" "Lexus" "Maserati" "Mazda"

"Mercedes" "Mitsubishi" "Nissan" "Opel" "Peugeot" "Porsche" "Renault"

"Rover" "Saab" "Seat" "Skoda" "Subaru" "Suzuki" "Tata" "Tesla"

"Toyota" "Volkswagen" "Volvo"

ה-dataset של המודל מכיל כעשרים אלף תמונות, 90% מהתמונות משומשות לאימון המודל, ו10% מהתמונות משומשות לבדיקת המודל.

דרישות להרצה של הפרוייקט:

גרסת פייטון: python 3.7.7

מערכת הפעלה: windows

ספריות פייטון דרושות: numpy, keras, tensorflow, tensorflow-gpu, pillow, matplotlib

כרטיס מסך:

שימוש בכרטיס מסך של חברת nvidia מאיץ את תהליך האימון של המודל

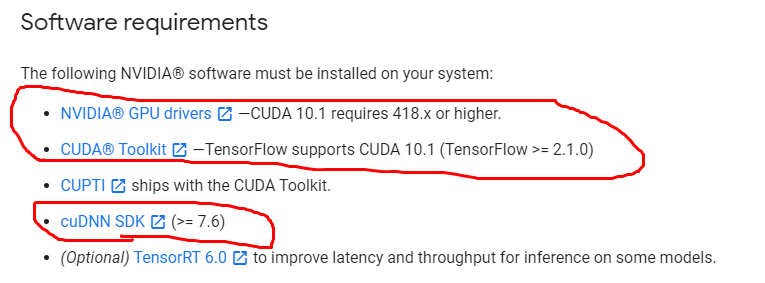
**מדריך למשתמש**

התקנת תמיכה לכרטיס מסך:

כדי להוריד תמיכה לכרטיסי מסך יש לעקוב אחרי ההנחיות הבאות:

יש להיכנס לאתר הבא ולהוריד את כל התוכנות והקבצים שבקישורים הבאים המוקפים באדום:

<https://www.tensorflow.org/install/gpu>



לאחר שהורדתם את כל התוכנות הנדרשות, אליכם להפעיל את הקובץ בשם setup.py דרך ממשק שורת הפקודה.

במהלך הרצת הקובץ תישאלו האם הקובץ בשם "cudnn64\_7.dll" קיים במחשב שלכם, אליכם לענות כן ובבקשה הבאה לקלט אליכם לשלוח את הנתיב אליו

הורדתם את הקובץ "cudnn64\_7.dll" בהורדה של ה cuDNN SDK והוא נמצא באחת מתתי התיקיות בתוכה.

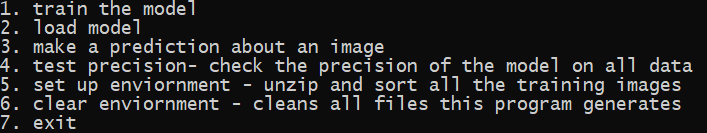
**הוראות שימוש ל main.py**

הקובץ main.py הוא הקובץ הראשי של התוכנה והוא מכיל את הפונקציות החשובות שלה.

הקובץ main.py מניח שכל הקבצים והספריות שהיה צריך להוריד כבר הותקנו לפני ההרצה.

כדי להריץ את הקובץ יש לכתוב בשורת פקודה (בהנחה ואתה נמצא בספרייה של (main.py "python ./main.py"

זהו התפריט למשתמש שמופיע כאשר מריצים את הקובץ main.py



ברגע שאתה מפעיל את התוכנה, ישנם כמה פעולות שאתה יכול לבחור מהן:

1. אימון המודל ושמירתו לדיסק
2. טעינת המודל מהדיסק לתוכנה
3. לעשות חיזוי לגבי תמונה
4. לבדוק את אחוזי ההצלחה של המודל
5. מכין את הסביבה להרצה של הפרוייקט
6. מוחק את הסביבה מהקבצים שהפרוייקט יצר
7. יציאה מהתוכנה

בכדי להפעיל את אופציה מספר 1: אופציה מספר 5 חייבת להיבחר קודם לכן, לווא דווקא באותו הרצה של הפרויקט.

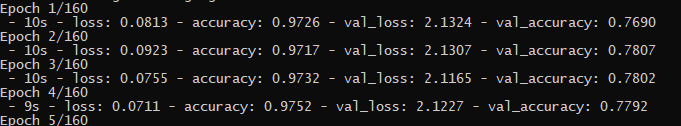
בכדי להפעיל את אופציות מספר 3 או 4: מודל חייב להיות טעון לתוכנה- בכדי לאפשר זאת ניתן לבחור באופציות מספר 1 או 2.

אופציה מספר 3 מקבלת את הנתיב לתמונה שאותה המשתמש רוצה לעשות חיזוי לגביה. התוכנה מדפיסה את התמונה ואת השם של החברה של הלוגו.

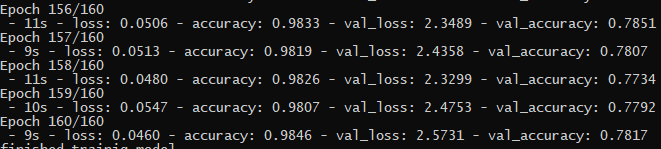
דוגמת הרצה של אופציה מספר 1:



הרבה מאוד שורות דומות לאלו יודפסו לאורך אימון המודל:



סוף אימון המודל:

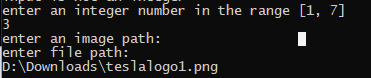




דוגמת הרצה של אופציה מספר 2:

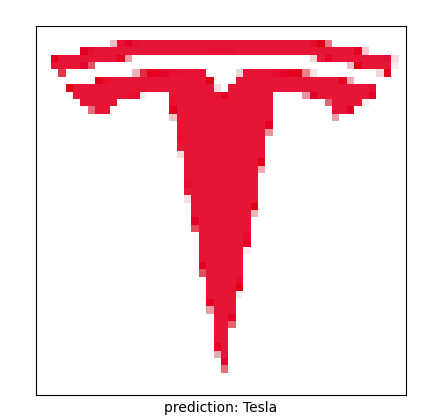


דוגמת הרצה של אופציה מספר 3:



(המשתמש צריך להכניס את השורה האחרונה)

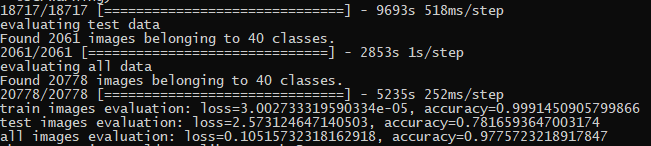
ואז מופיע התמונה יחד עם החיזוי של המודל לגביה



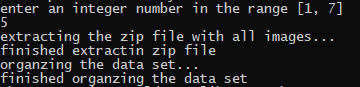
תוצאה של הרצה של אופציה מספר 4:

אורך הרצה של אופציה זאת בעזרת כרטיס מסך לוקחת שעות לבצע. ייתכן שיהיה כתוב שזמן ההרצה יהיה גדול בצורה אבסורדית, אך לאחר כמה דקות המספר קטן בצורה משמעותית.





תוצאה של הרצה של אופציה מספר 5:



תוצאה של הרצה של אופציה מספר 6:

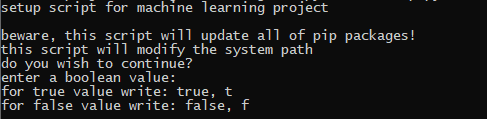


הוראות שימוש ל setup.py

setup.py מעדכן/מוריד ספריות פייטון הדרושות להרצת הפרוייקט וגם שומר את ההגדרות הנדרשות לשימוש בכרטיס מסך בפרויקט.

כדי להריץ את הקובץ יש לכתוב בשורת פקודה (בהנחה ואתה נמצא בספרייה של (setup.py "python ./setup.py"

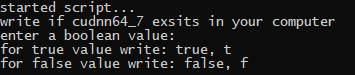
זה המסך שמופיע כאשר מריצים את setup.py



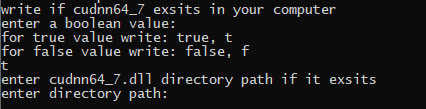
בכדי שהסקריפט ימשיך לרוץ, יש לכתוב או "true" או "t"

בכדי לסיים אותו, יש לכתוב "false" או "f"

אם המשתמש ממשיך להמשיך עם התוכנה, יופיע לו המסך הבא:



אם ברצון המשתמש להוסיף תמיכה לשימוש בכרטיס מסך, אליו לכתוב "true" או "t".



על המשתמש להכניס את הנתיב לקובץ cudnn64\_7.dll

גם אם המשתמש בוחר false, וגם אם בוחר true, והוא מסיים לטפל באופציה זו המשתמש יגיע אל המסך הבא שבו הסקריפט יתקין את הספריות הנדרשות.

**מדריך למפתח:**

קובץ globalvars.py:

מכיל הגדרות לתוכנה

קובץ sort.py:

מכיל פונקציות שמסדרות את התמונות הנדרשות לאימון המודל בתיקיות המתאימות להם

make\_folders - יוצרת את התיקיות הדרושות לdataset

copy\_files - מעתיקה את הקבצים של ה dataset לתיקיות המתאימות

sort\_all\_dirs – קוראת ל make\_folders ו copy\_files

קובץ neuralnetworks.py:

מכיל את הפעולות של הלמידת מכונה

print\_fit\_statistics – מדפיסה את הגרפים והסטטיסטיקה של האימון של המודל

define\_model – יוצר אובייקט מודל חדש

train\_model – מאמן את המודל

single\_prediction – עושה prediction אחד לגבי תמונה אחת

קובץ :utils.py

מכיל פעולות עזר

RepresentsInt – בודק אם מחרוזת מייצגת מספר שלם

get\_integer – קולט מחרוזות עד שמקבל מחרוזת שמייצגת מספר שלם ומחזיר את המספר השלם הזה

RepresentsBool – בודק אם מחרוזת מייצגת משתנה בוליאני

GetBool – הופך מחרוזת למשתנה בוליאני

get\_bool – קולט מחרוזות עד שמקבל מחרוזת שמייצגת משתנה בוליאני, מחזיר את המשתנה הבוליאני

get\_file\_path – קולט מחרוזות עד שמקבל נתיב לקובץ ומחזיר את הנתיב לקובץ

get\_dir\_path – קולט מחרוזות עד שמקבל נתיב לתיקיה ומחזיר את הנתיב לתיקיה

extract\_zip – מחלץ קובץ zip

build\_path – מחבר רשימה של מחרוזות שכל אחד מהם מייצג חלק מ path ל path אחד שלם ותקני.

get\_cudnn64\_7\_dir\_path – מחזיר את הנתיב לתיקיה שמכילה את קובץ ה cudnn64\_7.dll

save\_path\_vars – שומר את הנתיב של הקובץ cudnn64\_7.dll בקובץ הגדרות

setup\_path\_vars – מוסיף את הנתיב לתיקייה של הקובץ cudnn64\_7.dl למשתנה הסביבה PATH

count\_files\_in\_dir – סופר כמה קבצים יש בתיקייה

count\_imgs\_in\_dataset – סופר כמה קבצים יש בתקייה של dataset

list\_substraction – מחסר רשימה מרשימה

load\_photo טוען קובץ תמונה מהדיסק לתוכנה כאובייקט תמונה

print\_prediction – מדפיס prediction אחת של המודל לגבי תמונה אחת

קובץ ui.py:

מכיל את הפעולות שהמשתמש בוחר בהם

ui\_train\_model – מאמן את המודל ושומר אותו לדיסק

ui\_load\_model – טוען את המודל מהדיסק

ui\_test\_precision – מחשב את אחוזי ההצלחה של המודל

ui\_setup\_env – מכין את הסביבה לאימון המודל, כלומר שם את התמונות בתיקיות המתאימות

ui\_clean\_env – מוחק את כל הקבצים שהתוכנה יצרה

קובץ main.py:

main – הפונקציה הראשית של התוכנה

קובץ setup.py:

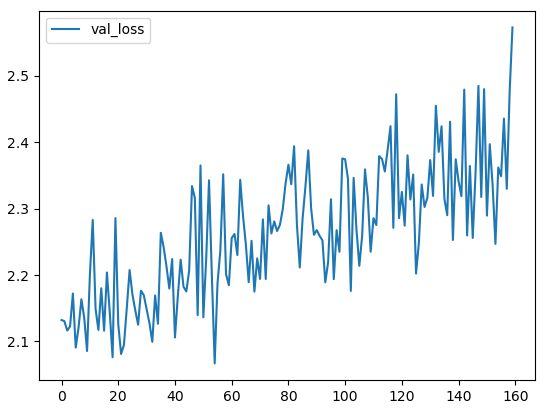
install\_packages – מעדכן/מוריד את הספריות של פייטון הנדרשות להרצת הפרוייקט

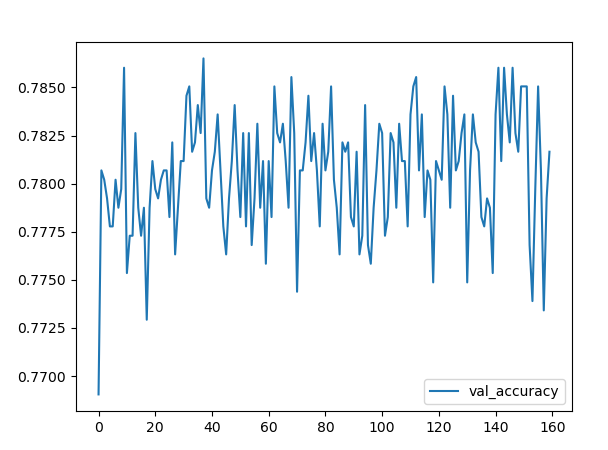
update\_packages – מעדכן את כל הספריות של סביבת הפייטון

main – מעדכן את הספריות הנדרשות להרצת הפרוייקט ומבקש מהמשתמש את הנתיב לספרייה cudnn64\_7.dll ושומר אותה בקובץ

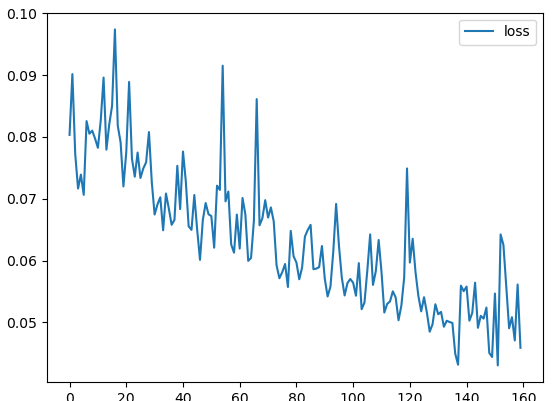
**מסקנות הרצת המודל**

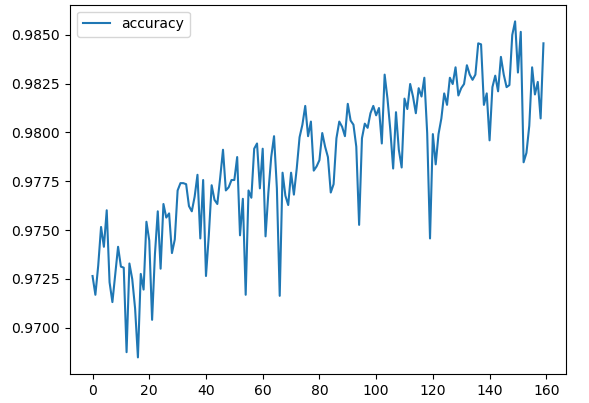
גרפים של ה validation וה accuracy של ה validation data set במהלך האימון:





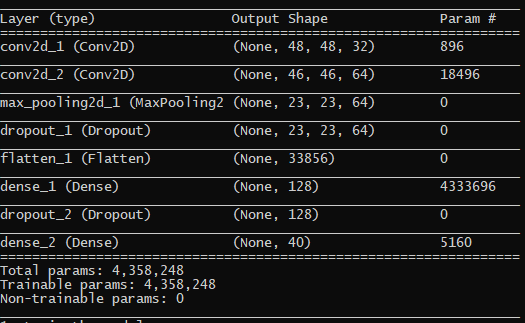
גרפים של ה validation וה accuracy של ה training data set במהלך האימון:

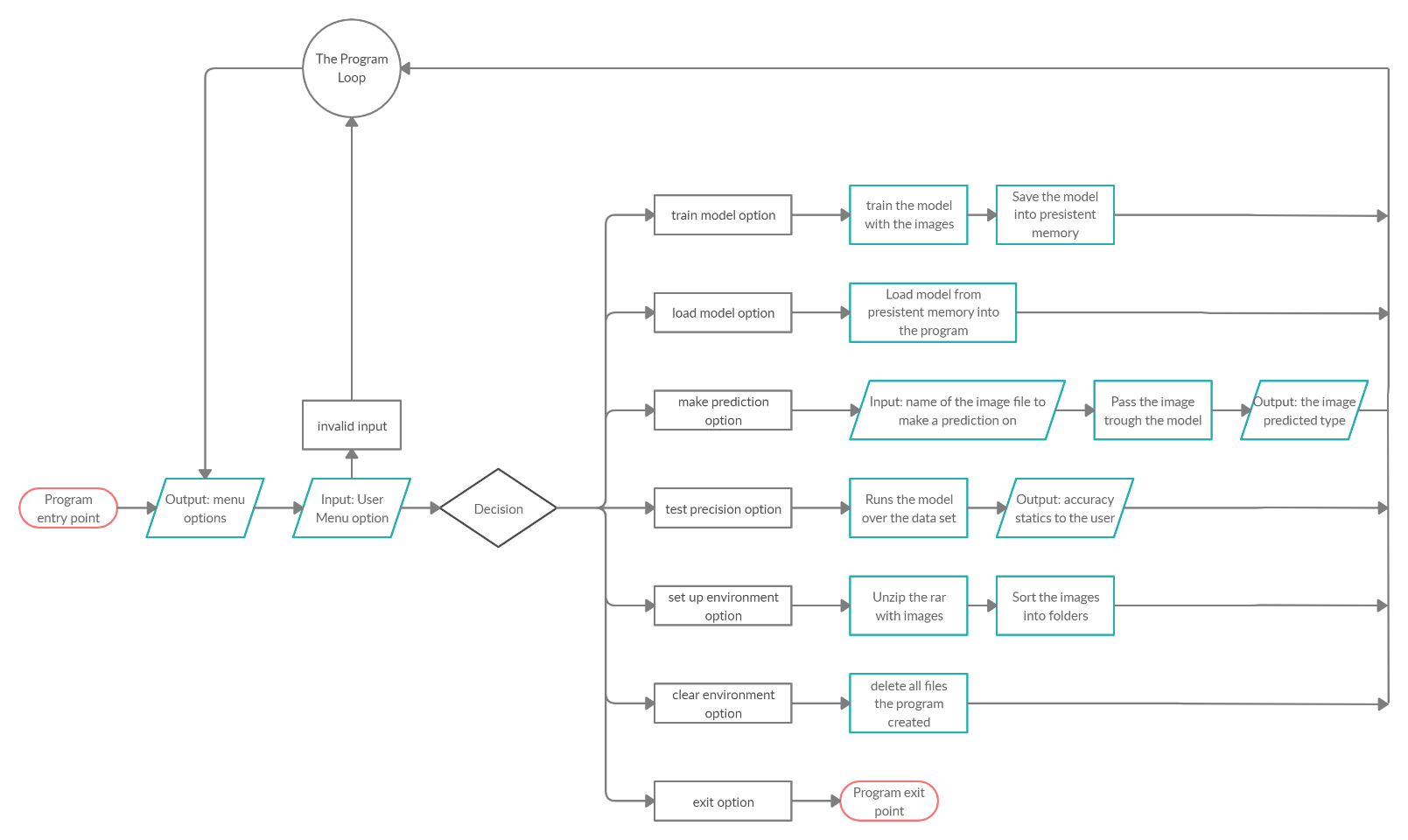




**נספחים:**

שכבות המודל:



דיאגרמת uml – תרשים זרימה של התוכנה: