

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 משמשות לאימון המודל, ו-10% מהתמונות משמשות לבדיקת המודל.

דרישות להרצה של הפרויקט:

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>

Software requirements

The following NVIDIA® software must be installed on your system:

- [NVIDIA® GPU drivers](#) –CUDA 10.1 requires 418.x or higher.
- [CUDA® Toolkit](#) –TensorFlow supports CUDA 10.1 (TensorFlow >= 2.1.0)
- [CUPTI](#) ships with the CUDA Toolkit.
- [cuDNN SDK](#) (>= 7.6)
- (Optional) [TensorRT 6.0](#) to improve latency and throughput for inference on some models.

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

```
1. train the model
2. load model
3. make a prediction about an image
4. test precision- check the precision of the model on all data
5. set up enviornment - unzip and sort all the training images
6. clear enviornment - cleans all files this program generates
7. exit
```

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 הפרוייקט.

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:

```
enter an integer number in the range [1, 7]
1
Found 18717 images belonging to 40 classes.
Found 2061 images belonging to 40 classes.
```

הרבה מאוד שורות דומות לאלו יודפסו לאורך אימון המודל:

```
Epoch 1/160
- 10s - loss: 0.0813 - accuracy: 0.9726 - val_loss: 2.1324 - val_accuracy: 0.7690
Epoch 2/160
- 10s - loss: 0.0923 - accuracy: 0.9717 - val_loss: 2.1307 - val_accuracy: 0.7807
Epoch 3/160
- 10s - loss: 0.0755 - accuracy: 0.9732 - val_loss: 2.1165 - val_accuracy: 0.7802
Epoch 4/160
- 9s - loss: 0.0711 - accuracy: 0.9752 - val_loss: 2.1227 - val_accuracy: 0.7792
Epoch 5/160
```

סוף אימון המודל:

```
Epoch 156/160
- 11s - loss: 0.0506 - accuracy: 0.9833 - val_loss: 2.3489 - val_accuracy: 0.7851
Epoch 157/160
- 9s - loss: 0.0513 - accuracy: 0.9819 - val_loss: 2.4358 - val_accuracy: 0.7807
Epoch 158/160
- 11s - loss: 0.0480 - accuracy: 0.9826 - val_loss: 2.3299 - val_accuracy: 0.7734
Epoch 159/160
- 10s - loss: 0.0547 - accuracy: 0.9807 - val_loss: 2.4753 - val_accuracy: 0.7792
Epoch 160/160
- 9s - loss: 0.0460 - accuracy: 0.9846 - val_loss: 2.5731 - val_accuracy: 0.7817
finished training model
```

```
finished training model...
saving model weights to disk...
```

דוגמת הרצה של אופציה מספר 2:

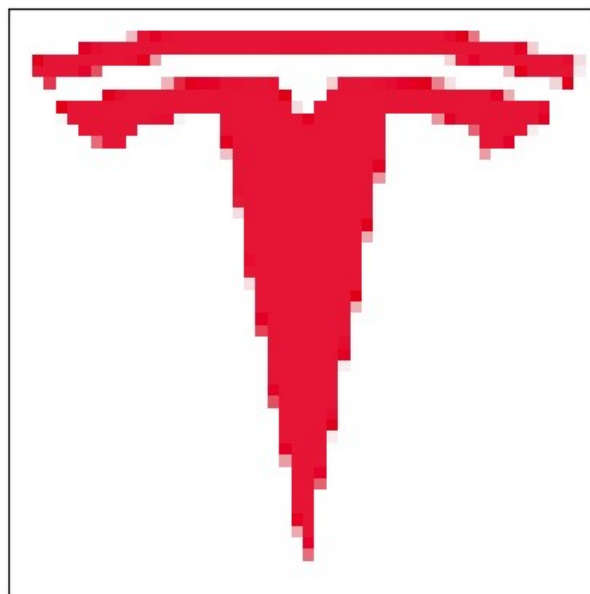
```
enter an integer number in the range [1, 7]  
2  
model loaded
```

דוגמת הרצה של אופציה מספר 3:

```
enter an integer number in the range [1, 7]  
3  
enter an image path:   
enter file path:  
D:\Downloads\tesla1.png
```

(המשתמש צריך להכניס את השורה האחרונה)

ואז מופיע התמונה יחד עם החיזוי של המודל לגביה



prediction: Tesla

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.

```
enter an integer number in the range [1, 7]
4
evaluating training data
Found 18717 images belonging to 40 classes.
```

```
18717/18717 [=====] - 9693s 518ms/step
evaluating test data
Found 2061 images belonging to 40 classes.
2061/2061 [=====] - 2853s 1s/step
evaluating all data
Found 20778 images belonging to 40 classes.
20778/20778 [=====] - 5235s 252ms/step
train images evaluation: loss=3.002733319590334e-05, accuracy=0.9991450905799866
test images evaluation: loss=2.573124647140503, accuracy=0.7816593647003174
all images evaluation: loss=0.10515732318162918, accuracy=0.9775723218917847
```

תוצאה של הרצה של אופציה מספר 5:

```
enter an integer number in the range [1, 7]
5
extracting the zip file with all images...
finished extractin zip file
organzing the data set...
finished organzing the data set
```

תוצאה של הרצה של אופציה מספר 6:

```
enter an integer number in the range [1, 7]
6
started cleaning the files the program generated
finished cleaning
```


הוראות שימוש ל setup.py

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

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

```
setup script for machine learning project  
  
beware, this script will update all of pip packages!  
this script will modify the system path  
do you wish to continue?  
enter a boolean value:  
for true value write: true, t  
for false value write: false, f
```

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:

```
started script...  
write if cudnn64_7 exists in your computer  
enter a boolean value:  
for true value write: true, t  
for false value write: false, f
```

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

```
write if cudnn64_7 exists in your computer  
enter a boolean value:  
for true value write: true, t  
for false value write: false, f  
t  
enter cudnn64_7.dll directory path if it exists  
enter directory path:
```

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 אחת 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.dll למשתנה הסביבה PATH
 count_files_in_dir – סופר כמה קבצים יש בתיקייה
 count_imgs_in_dataset – סופר כמה קבצים יש בתיקייה 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:

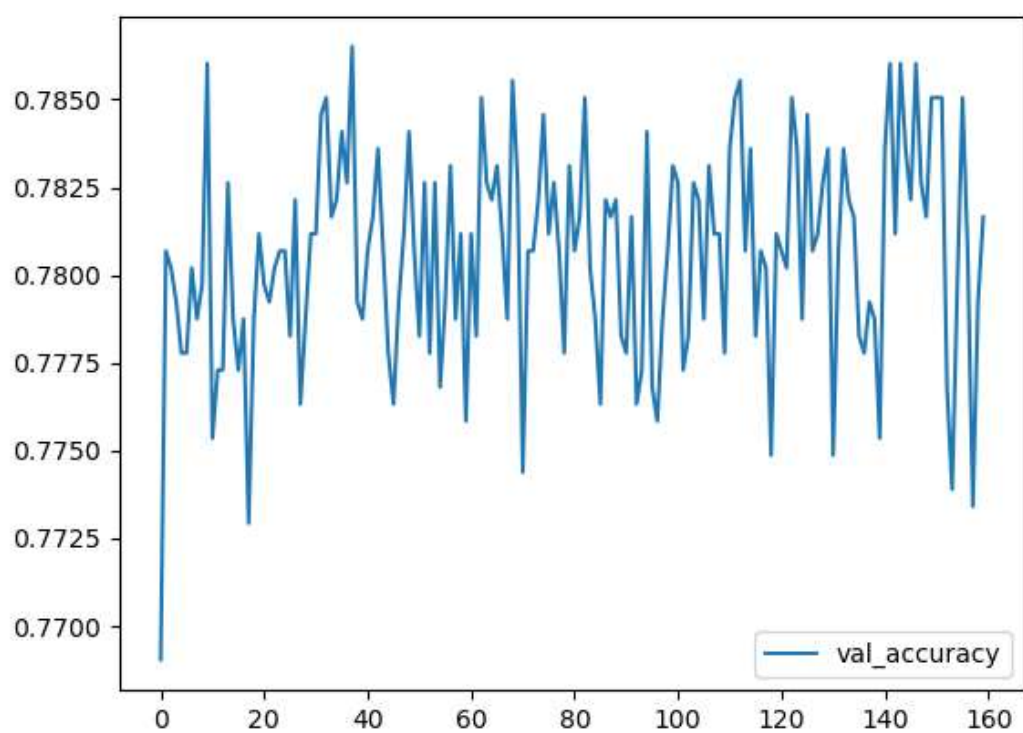
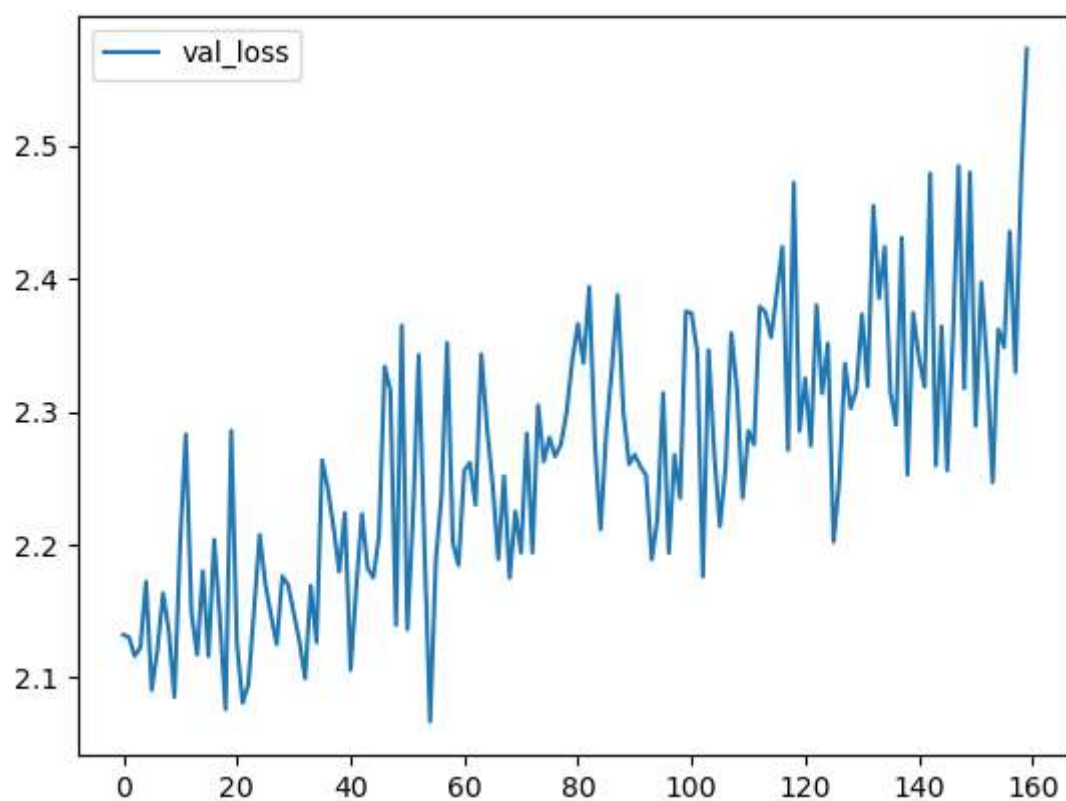
main – הפונקציה הראשית של התוכנה

קובץ setup.py:

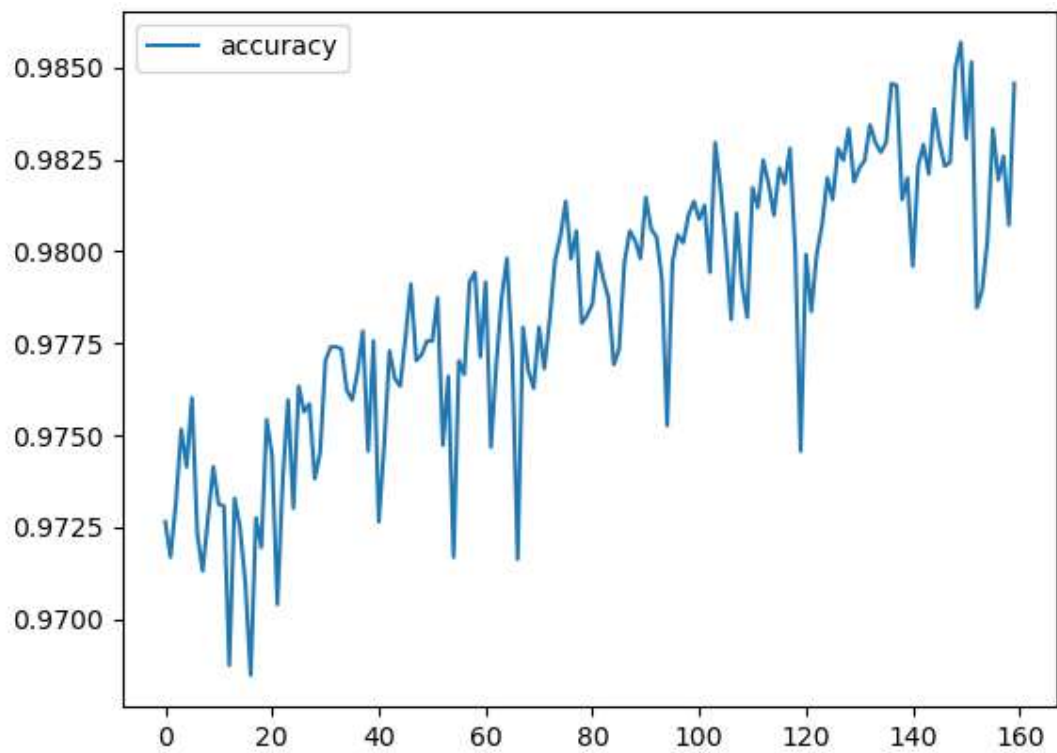
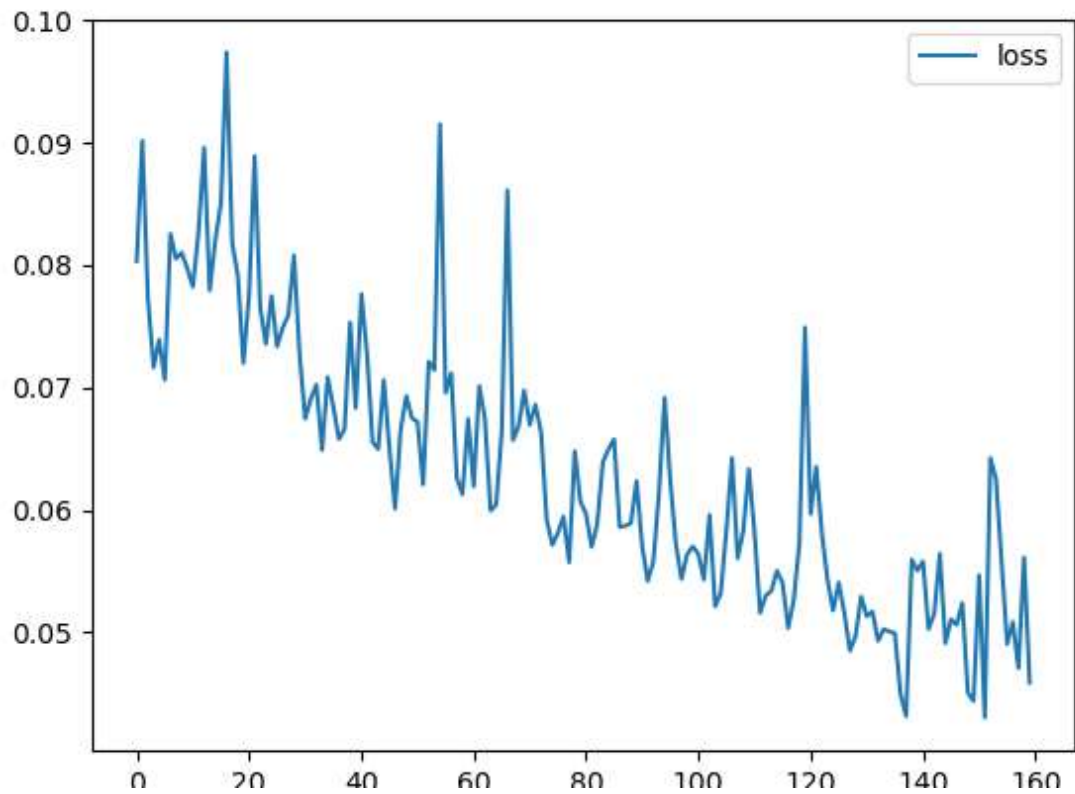
install_packages – מעדכן/מוריד את הספריות של פייטון הנדרשות להרצת הפרוייקט
 update_packages – מעדכן את כל הספריות של סביבת הפייטון
 main – מעדכן את הספריות הנדרשות להרצת הפרוייקט ומבקש מהמשתמש את הנתיב לספרייה
 cudnn64_7.dll ושומר אותה בקובץ

מסקנות הרצת המודל

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



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



נספחים:

שכבות המודל:

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	896
conv2d_2 (Conv2D)	(None, 46, 46, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 23, 23, 64)	0
dropout_1 (Dropout)	(None, 23, 23, 64)	0
flatten_1 (Flatten)	(None, 33856)	0
dense_1 (Dense)	(None, 128)	4333696
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 40)	5160
Total params: 4,358,248		
Trainable params: 4,358,248		
Non-trainable params: 0		

דיאגרמת uml – תרשים זרימה של התוכנה:

