

פרויקט גמר 5 יחידות לימוד

התמחות – למידת מכונה

Deep Learning

נושא הפרויקט

זיהוי וסיווג מכונות בתמונות

Cars classification in images

בית ספר: "מקיף י"א ראשונים"

שם התלמיד: עידן גלנץ

תעודת זהות: 326611274

שם המנחה: דינה קראוס

תאריך הגשה: 16.6.2022



תוכן

מבוא	3
בסיס תאורטי	4
מדריך למשתמש	7
ארכיטקטורה של הפרויקט	14
main.....	18
Options	19
Windows	20
Prepare_data	21
Train	22
test	25
רפלקציה.....	26
ביבליוגרפיה	27

מבוא

פרויקט הגמר שלי עוסק בהתמחות אשר למדנו בבית ספר שהיא בנושא "למידה עמוקה"

בתחום זה עוסקים ללמד את המחשב כיצד לחשוב כמו בן אדם ולנתח המון נתונים ולמצוא את המשותף ביניהם המכונה לומדת לנתח תהליכי לימוד כך שיהיו דומים ככל האפשר ליכולות החשיבה והניתוח של מוח אנושי. מוח האדם בנוי מרשת נוירונים בהם נקלט המידע. שיתוף הפעולה בין אלפי הנוירונים המורכב מפעולה קטנה של כל אחד מהם, מאפשר למוח לעשות פעולה גדולה ומורכבת. ובעזרת התקשורת שבין כל תא עצב

לאילו שבקרבתו נוצרת היכולת ללמוד דברים חדשים גם בהתבסס על מידע קיים ולזכור את מה שלמדנו. על רעיון זה מתבססים הפרויקטים בהתמחות זו בכלל וכן הפרויקט שלי בפרט. רשת נוירונים של "למידה עמוקה" עובדת באופן דומה, שכן היא מורכבת ממשקלים ונוירונים שכל אחד מהם מבצע פעולה

מתמטית פשוטה, ובסופו של דבר התוכנה מסוגלת ללמוד באופן דומה למוח האנושי. בתחום learningDeep המתכנת אינו כותב את הקוד לזיהוי האובייקט הנלמד באופן ידני, אלא הוא כותב תוכנה

שמסוגלת ללמוד בעצמה באמצעות התמונות שברשותה, באמצעות מודל. בפרויקט זה בחרתי לבנות תוכנה שמטרתה לזהות מכונות הנמצאות בתמונות. המודל לומד המון תמונות של מכונות ותמונות שאין בהן מכונות ולאחר תהליך למידה הוא יוכל לנתח תמונות שונות וכאשר ימצא מכונת באחת התמונות הוא ידע להגיד. בחרתי בנושא זה מכיוון שראיתי בו כשימושי ביותר ואתגר מעניין, הפרויקט יכול להיות שימושי מאוד לחברות המתעסקות בנייה של רכבים אוטונומיים או בחברות המתעסקות בצילומי כבישים על מנת לדעת מתי יש רכבים על הכביש ומתי אין ולייצר מכך סטטיסטיקות שונות המנתחות את המצב הזה ואף ניתן להתקין מצלמות אלו על רמזורים ובכך לייצר מערכת אשר תדע איזה רמזורים להפעיל ואיזה לא.

בסקירה של המצב הקיים בשוק מצאתי כי ישנן תוכנות שונות אשר מצליחות לזהות מכונות ברמות דיוק גבוהות, ואכן ישנן תוכנות ומוצרים (כגון: Mobileye) מבוססות machine learning בעלות היכולת לזהות מכונות

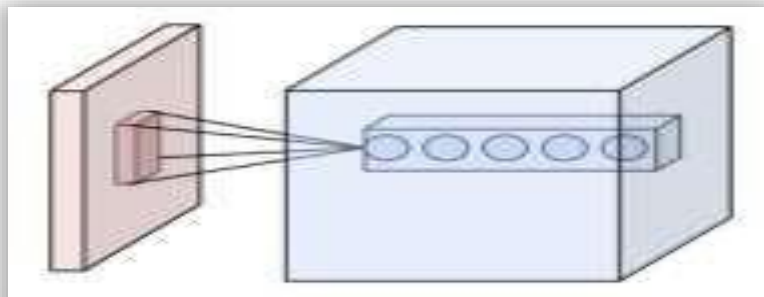
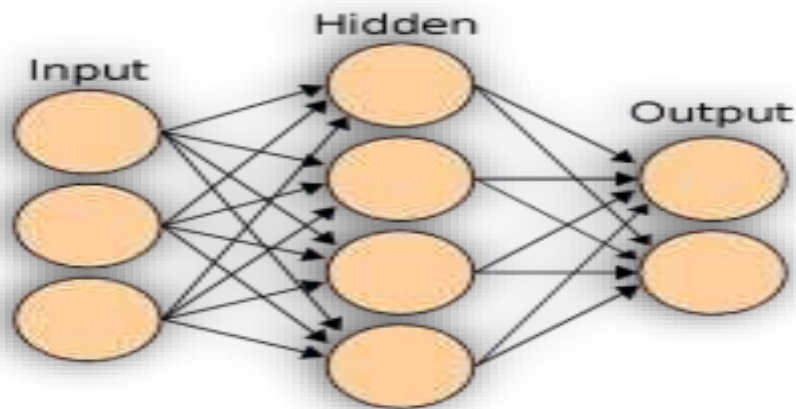
שלם ומפורט של העלאת dataset, אימון המודל, בחינה של המודל, ואף זיהוי של תמונות ספציפיות. במסגרת ביצוע פרויקט זה הרחבתי את הידע שלי בתחום הבינה המלאכותית AI בכלל ובתחום deep learning בפרט. למדתי מושגים תיאורטיים הקשורים לנושא, כמו רשת נוירונית, משקלים, מודל, פונקציות פעולה, ועוד. כמו כן, הבנתי לעומק איך העולם הזה עובד ומהי החשיבות שלו. עם זאת, במהלך ביצוע הפרויקט בהתמחות הנ"ל נתקלתי במספר אתגרים מרכזיים. ראשית כל, האתגר המשמעותי ביותר נבע מכך שהתחום הזה היה חדש לי ולא הכרתי אותו לפני תחילת העבודה. היה עליי ללמוד היטב את כל הרקע התיאורטי לנושא והרעיונות מאחורי deep learning ובמקביל ללמוד את הדרך המעשית ליצירת פרויקט המתבסס על נושא זה. יתרה מכך, זהו נושא שיחסית חדש בלימודים

התיכונים בכלל ובבית הספר שלי בפרט, לכן גם עבור המורה, גם עבור בית הספר, וגם עבור התלמידים הוא חדש יחסית. כלומר, התבקשנו לכתוב תוכנה המבצעת דבר שמלכתחילה לא ידענו כיצד לבצעו ועל כן למדנו תוך כדי פעולה. התגברתי על אתגר זה בכך שאני וחבריי להתמחות למדנו את הנושא תוך כדי ביצוע הפרויקט באמצעות מדריכים וספרים שונים, מורתנו דינה, האינטרנט ומקורות נוספים.

בסיס תאורטי

רשת עצבית - רשת נוירונים או רשת קשרית הוא מודל מתמטי חישובי, שפותח בהשראת תהליכים מוחיים או קוגניטיביים המתרחשים ברשת עצבית טבעית ומשמש במסגרת למידת מכונה. רשת מסוג זה מכילה בדרך כלל מספר רב של יחידות מידע (קלט ופלט) המקושרות זו לזו, קשרים שלעיתים קרובות עוברים דרך יחידות מידע "חבויות"

רשת קונבולוציה-היא סוג של רשת נוירונים המשתמשת בפעולת הקונבולוציה במקום בכפל מטריצות כללי לפחות באחת מהשכבות שלה. סוג זה של רשת נוירונים משמש בעיקר לעיבוד תמונה וראיה ממוחשבת, אך יש לו שימושים גם במערכות המלצה, עיבוד שפה טבעית וממשק מוח-מחשב



מונחי בסיס

Accuracy – אחוז ההצלחה של המודל בחיזוי הקטגוריות של המכוניות (נכון או לא נכון).

Loss – מגדיר כמה קרובות היו תוצאות חיזוי השירים לקטגוריה האמיתית שאליה השיר שייך.

Train data – השירים אותן המודל לומד בעת תהליך האימון, מאגר זה הוא הגדול מבין שלושת תתי המאגרים.

Test data – התמונות אותן המודל אינו לומד, אלא מנסה לזהות בתום תהליך הלמידה.

Validation data – validation הוא דמוי test שנערך במהלך תהליך הלמידה. מטרתו להציג בפני המשתמש את אחוזי ההצלחה של המודל כבר בעת הלמידה שלו. על כן validation data אלו שירים שהמודל אינו לומד, אלא מנסה לזהות במהלך האימון שלו.

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

שתי בעיות שעלולות להיווצר הן: overfitting ו-fitting under:

Overfitting – "התאמת יתר" היא מצב בו המודל מותאם יתר על המידה למאגר אותו הוא לומד, ולכן

פחות מצליח בביצוע תחזיות. ניתן לזהות מצב כזה כאשר ה-validation accuracy גבוה משמעותית מן

ה-training accuracy או כאשר ה-validation loss קטן משמעותית מן ה-training loss.

Under fitting – ההפך מ-overfitting, מצב בו מאגר הלמידה פשוט מידי ולא כולל מספיק מגוון של תמונות שונות, או כאשר יש מיעוט בפרמטרים המגדירים את המודל. במצב זה המודל אינו מצליח ללמוד את התמונות. ניתן לזהות מצב כזה כאשר ה-validation accuracy נמוך משמעותית מן ה-training

accuracy או כאשר ה-validation loss גבוה משמעותית מן
ה-training loss.

epoch - חלוקה ל-epochs הינה הגדרה של מספר פעמים שתהליך הלמידה
יתבצע מחדש.

מדריך למשתמש

הוראות התקנה:

בפרק זה אסביר על הפרויקט כאשר הנמען הוא המשתמש. ראשית, אעבור על הדרישות

להרצת התוכנית, הוראות ההתקנה והקבצים הנדרשים. יש להוריד **3.7** - <https://www.python.org/downloads/> Python

אם יש לך **Python** על המחשב ואתה לא בטוח איזו גרסה מותקנת, תוכל לבדוק מהי גרסתו

באמצעות הפקודה: **python -V**

יש להוריד מספר ספריות קוד בהן הפרויקט

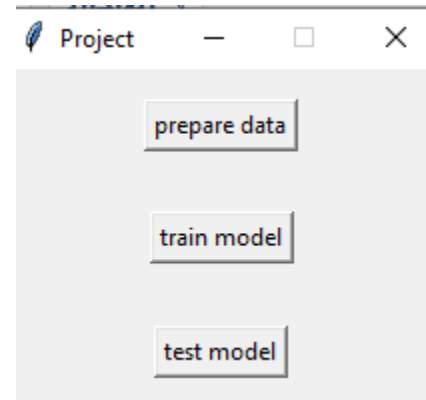
שם הספרייה	פקודת התקנה	קישור
keras	pip install keras	https://pypi.org/project/Keras/
tensorflow	pip install tensorflow	https://pypi.org/project/tensorflow/
matplotlib	pip install matplotlib	https://pypi.org/project/matplotlib/
numpy	pip install numpy	https://pypi.org/project/numpy/
glob	pip install glob	https://pypi.org/project/glob2/

/https://pypi.org/project/Pillow	pip install Pillow	PIL
https://pypi.org/project/pathlib/	Pip install pathlib	pathlib
https://pypi.org/project/pytest-shutil/	Pip install shutil	shutil
https://www.geeksforgeeks.org/how-to-install-tkinter-in-windows/	pip install tk	tkinter

מדריך למשתמש:

את קובץ **project** יש להעביר לאחסון המחשב ולחלץ בתוך הקובץ ניתן למצוא את הדאטהסט ואת המודלים אשר בהם נשתמש על מנת להריץ את הפרויקט. כאשר כל המודלים נמצאים באותו עורך טקסט אשר יכול להריץ פייתון.

מריצים את הפעולה **main** בהרצת המודל יקפוץ החלון הבא:

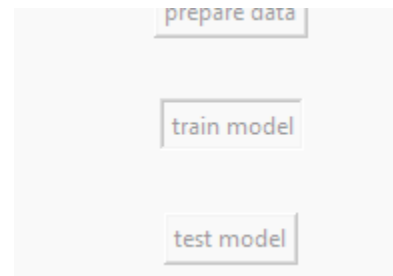


קודם כל נצטרך לעשות **prepare data** על מנת שכל הקבצים יגיעו למקומות הדרושים (**path**)

כאשר שהפעולה תסתיים תופיע החלונית הבאה ובכך נדע שניתן לאמן או לבחון את המודל

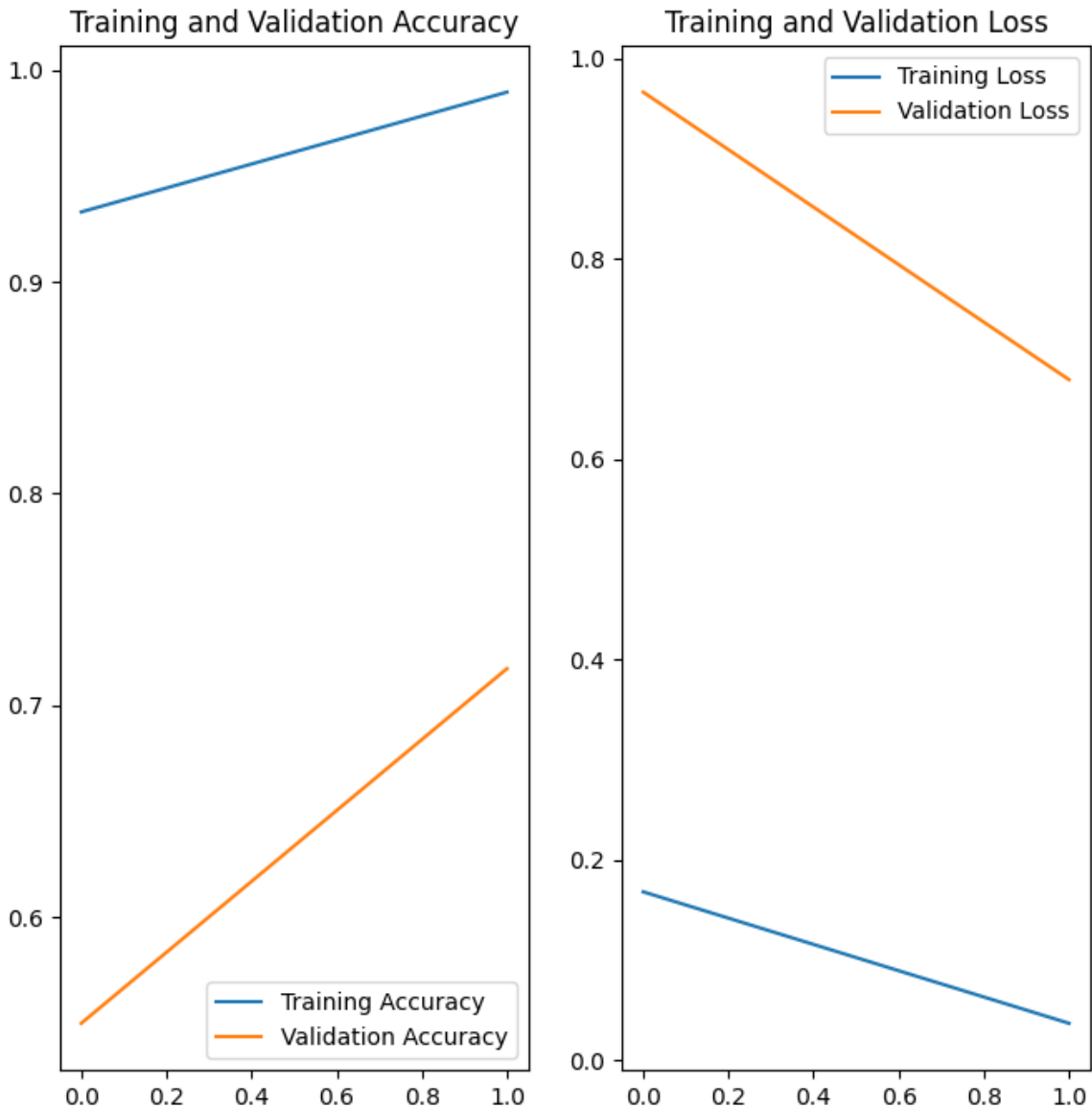
```
Preparing data
Finished preparing data
```

אחר כך נלחץ על כפתור **train model** אשר מתחיל לאמן את המודל בעזרת הדאטהסט



```
dense_1 (Dense)          (None, 2)          258
=====
otal params: 7,470,754
rainable params: 7,470,754
on-trainable params: 0
=====
ound 12277 files belonging to 2 classes.
ound 1755 files belonging to 2 classes.
poch 1/2
23/123 [=====] - 382s 3s/step - loss: 0.1679 - accuracy: 0.9330 - val_loss: 0.9665 - val_accuracy: 0.5499
poch 2/2
84/123 [=====>.....] - ETA: 1:57 - loss: 0.0425 - accuracy: 0.9882
```

כאשר המודל מאומן יעלו התוצאות בצורה של גרף אשר מראה את השינוי



לבסוף תעלה החלונית הבאה והמודל ישמר כפי שניתן לראות

```
Done training  
Saved model to saved_model.h5 file
```

לאחר מכן נוכל לבדוק את אימון המודל בעזרת הפעולה **test model**
נוכל לבדוק את יעילות המודל על תמונות שונות אשר נבחרו באופן רנדומלי
מבין כל התמונות

```
Found 3508 files belonging to 2 classes.  
110/110 [=====] - 15s 133ms/step - loss: 0.7378 - accuracy: 0.6836  
Test loss: 0.7378156781196594  
Test accuracy: 0.6835803985595703  
Found 3508 files belonging to 2 classes.  
1/1 [=====] - 0s 89ms/step  
1/1 [=====] - 0s 20ms/step  
1/1 [=====] - 0s 17ms/step  
1/1 [=====] - 0s 17ms/step  
1/1 [=====] - 0s 17ms/step  
1/1 [=====] - 0s 16ms/step  
1/1 [=====] - 0s 17ms/step  
1/1 [=====] - 0s 22ms/step  
1/1 [=====] - 0s 16ms/step
```

ונראה תוצאות של התמונות

car - 73 %



ארכיטקטורה של הפרויקט

איסוף הנתונים:

את הנתונים אספתי מכמה דאטהסטים שונים מאתר (**keggle**) לאחר מכן עשיתי אוגמנטיישן לתמונות בעזרת הקודים הבאים:

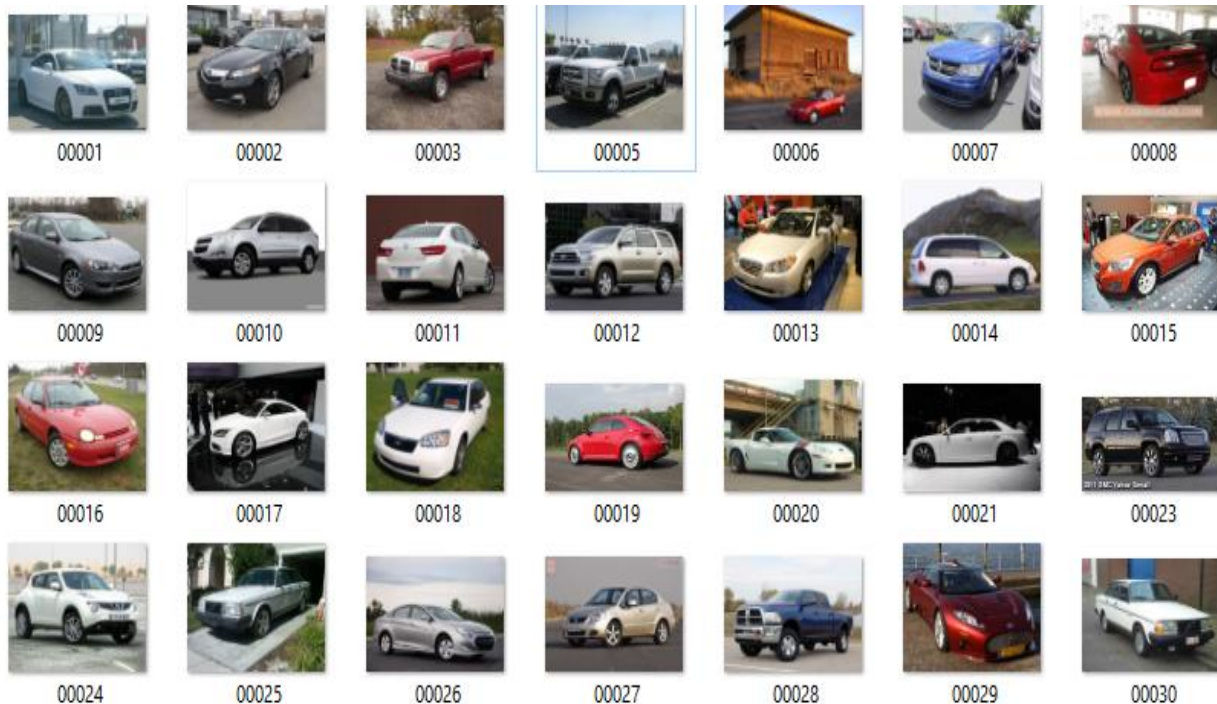
*יש לציין שבקודים הבאים איני משתמש בפרויקט עצמו אלא השתמשתי בהם לפני על מנת לסדר לעצמי את הדאטהסט

Change name to file

הקוד לוקח את התמונות בקובץ ומשנה את שמן

-zero filled number מוסיף אפסים במידת הצורך לשם התמונה

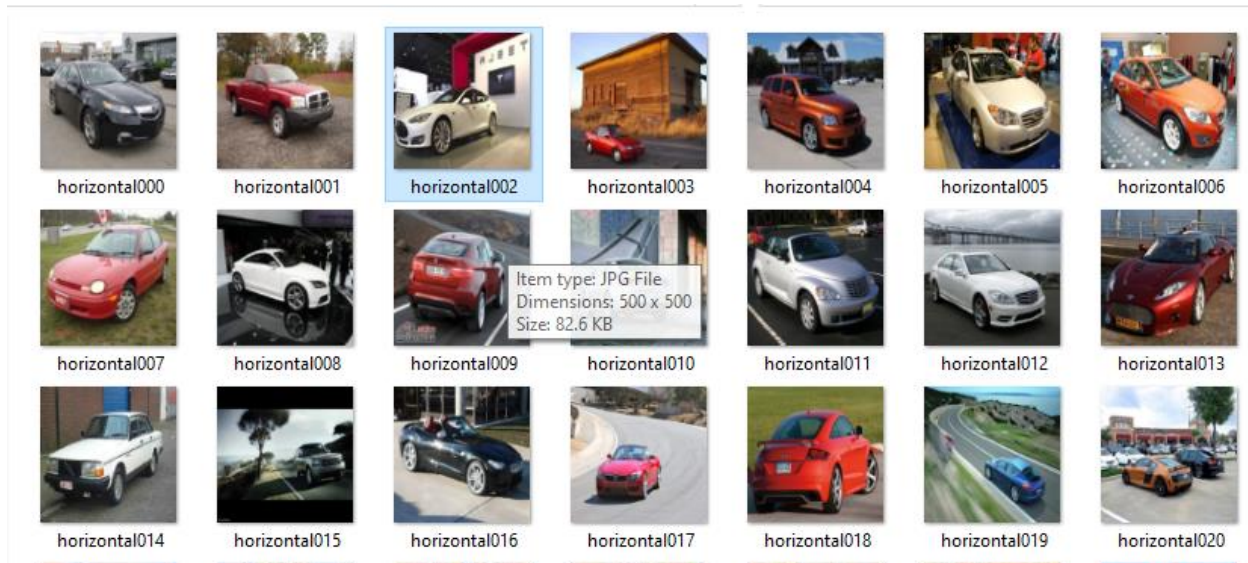
הקוד הבא לוקח את כל התמונות בקובץ הנתון ומשנה את שמם למסודר לפי אפסים דוגמא:



Image_flip_horizontal

הקוד לאחר מכן הופך את התמונות בתמונות מראה ושומר אותם מחדש

cv2.flip - הפעולה הופכת אותה לתמונת מראה



Rotate image

הקוד משנה התמונות מבחינת הזווית (15 מעלות ימינה ושמאלה)

cv2.getRotationMatrix2D(center=(h//2, w//2), angle = 15, scale = 1) - הפעולה משנה את גובה ורוחב ואת הזווית של התמונה

וכך התמונות נראו לאחר השימוש בקוד:

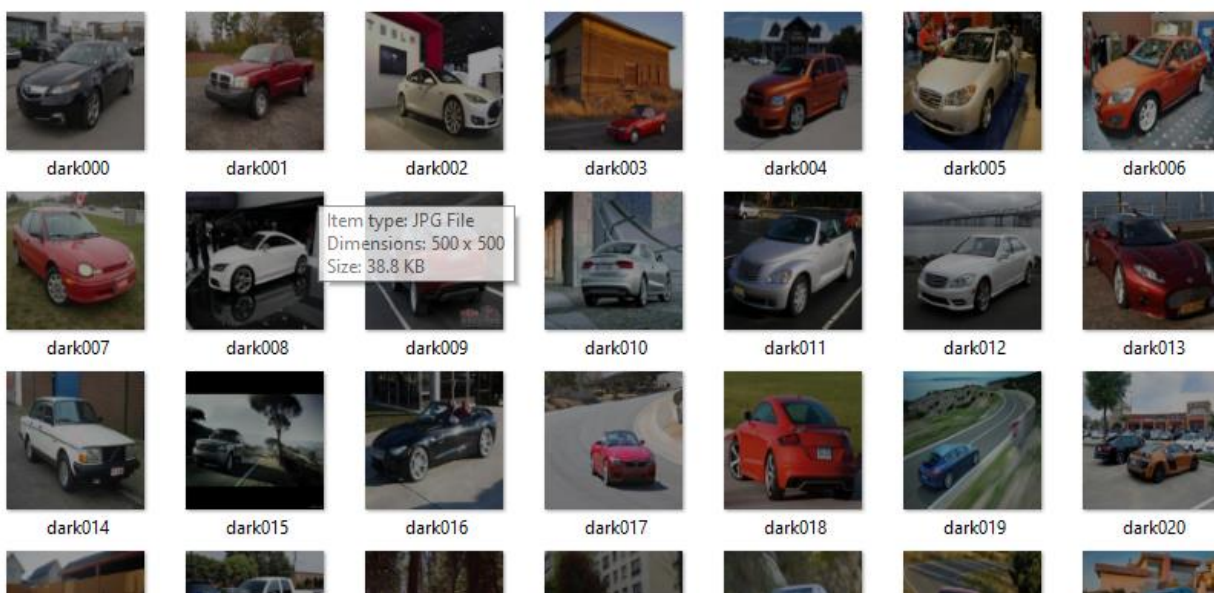


ובעזרת הקוד הבא

Brightness change

`enhancer = ImageEnhance.Brightness(im)`
והפקטור קובע את הבהירות (שבר יוצר יותר קהירות ומעל 1 גורם לתמונה להיות בהירה יותר)
`factor = 0.6`

שיניתי את הבהירות של התמונות:

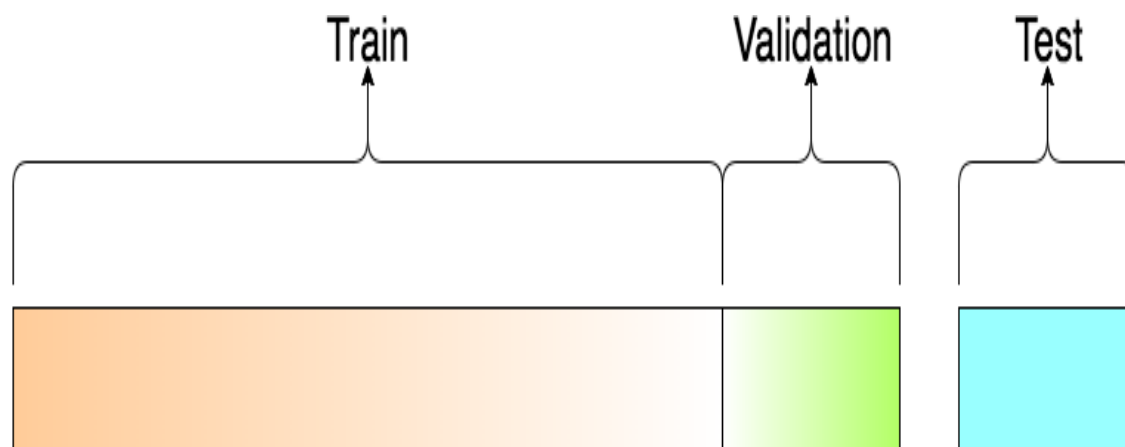


עידן גלנץ/ זיהוי מכוניות בתמונות



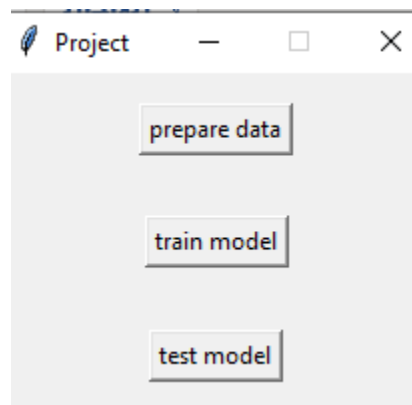
לאחר מכן שילבתי את כל התמונות לדאטה סט גדול וחילקתי את התמונות לסט, טריין (train) וולדיינג

Validation



main

דרך הפעולה הזאת נפעיל את כל הפרויקט ויקפוץ החלון הבא



Options

דרך מודל זה נוכל לשנות את האופציות שבהם המודל ישתמש

התיקיה שבה ימצאו הדאטהסט שבו ישתמש המודל על מנת להתאמן	DATA_DIR
כמות המחלקות שמודל יסווג	NUM_CLASSES
אורך התמונות שיהיו בדאטהסט	IMAGE_HEIGHT
רוחב התמונות שיהיו בדאטהסט	IMAGE_WIDTH
כמות התמונות שהמודל יתאמן עליהם בכל אימון	BATCH_SIZE
כמות הפעמים שבהם המודל יאמן את עצמו	EPOCHS
היחס בו התמונות מהדאטהסט יחולקו לאימון	TRAIN_SIZE
היחס בו התמונות מהדאטהסט יחולקו לבדיקת התוצאות	TEST_SIZE

Windows

המודל האחראי על אינטראקציית המשתמש הראשונית (gui) בעזרת חבילת tkinter

אחראי על גודל החלון הראשוני שנראה אשר יכיל בתוכו כפתורים	tk_root.geometry
השם של החלון אשר יעלה	tk_root.title
פעולה שדרכה ניתן לשנות את גודל החלון הראשוני	tk_root.resizable
פעולה שדרכה ניתן לסדר את השורות לפי מרווח מסויים	tk_root.columnconfigure
פעולה שדרכה ניתן לסדר את הטורים לפי מרווח מסויים	tk_root.rowconfigure
מייצרת כפתור שלחיצה עליו מפעילה פעולה אחרת	tk.Button
מציבה את הכפתורים במיקום מסויים בחלון	train_btn.grid

Prepare_data

המודל שבו כל הכנות הנתונים יתקיים דרך הנתונים שנכנסו במודל options

data_dir	יקבל את כל המידע מהמודל אופציות
dataset_dir	יקבל את הדאטהסט הראשוני
train_size	יקבל את האופציה של גודל חלוקת התמונות לאימון
test_size	יקבל את האופציה של גודל חלוקת התמונות לבדיקה
train_dir	יכין תיקיה שבה יהיו את קבצי האימון
test_dir	יכין תיקיה שבה יהיו את קבצי הבדיקה
val_dir	יכין תיקיה שבה יהיו את קבצי האימות
os.makedirs	יצור את התיקיות בתוך המחשב

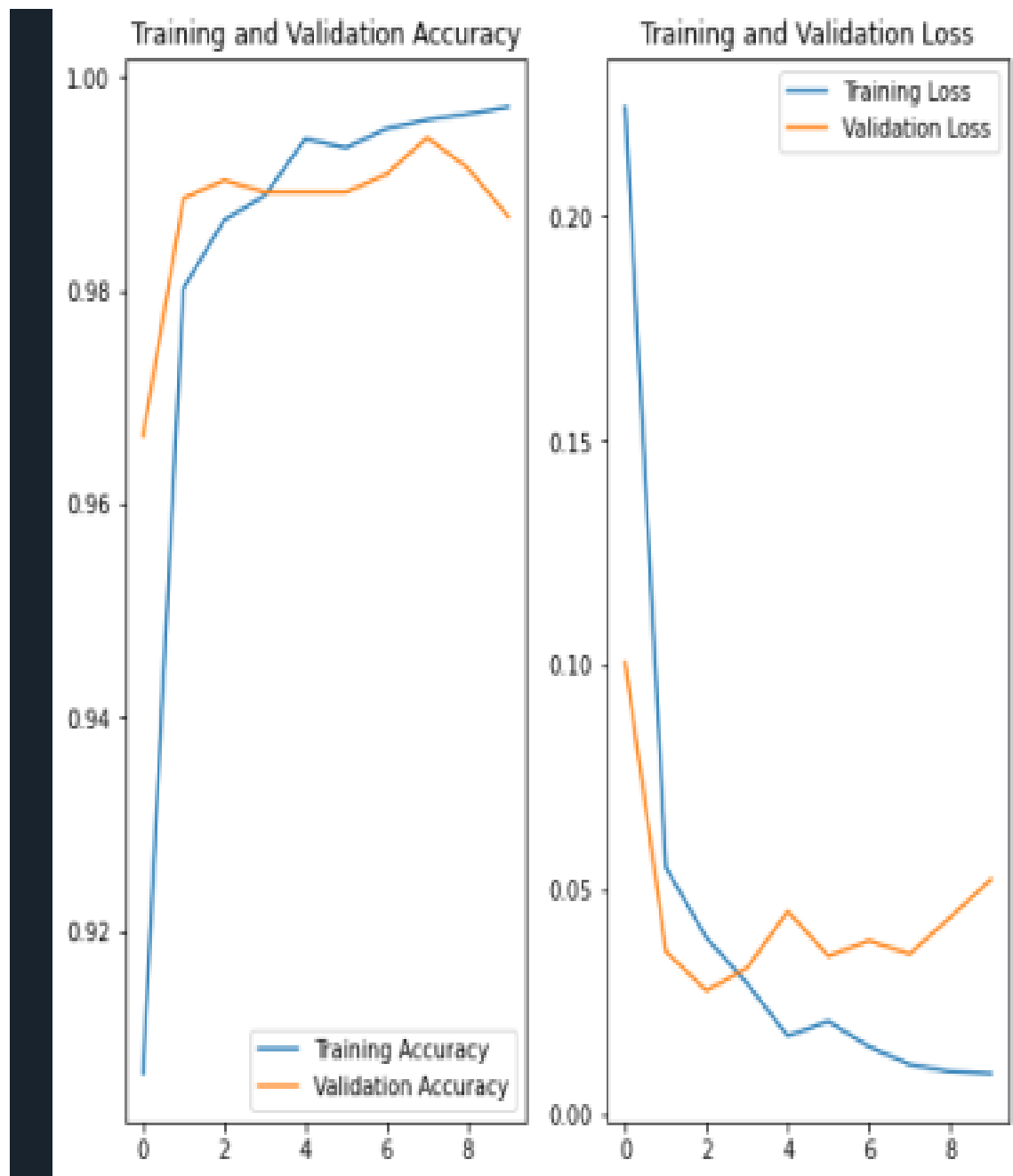
Train

המודל דרכו נאמן את המודל בעזרת הדאטהסט, המודל יופעל בעזרת הכפתור train the model

סוג המודל בשבו השתמשתי על מנת לאמן את המודל	model = Sequential
פעולה זאת גורמת לייצור של כמה תוצאות לכל תמונה ובכך משפרת את המודל ואת התוצאות הסופיות לכל איפוצ'	layers.experimental.preprocessing.Rescaling(1./255, input_shape=(image_height, image_width, 3))
במודל יש 3 שכבות של לימוד בעלות פילטרים {16,32,64} כל שכבה בעצם מעלה את כמות הסינונים של החישובים	layers.Conv2D(16, 3, padding='same', activation='relu')
שם בכל צד של התוצאה את אותו מספר של אפסים על מנת שהתוצאה מכל שכבה תצא אותו מספר של ספרות	'padding='same
פונקציית ההפעלה שלפיה מחושבת השכבה.	, activation='relu'
לוקח את התוצאות המתקבלות לאחר חישוב שכבה אחת ומעגל אותה ולוקח את התוצאה הטובה ביותר	layers.MaxPooling2D()
מעביר את התוצאות המתקבלות ממערך מסויים של מספרים למספר אחד מחושב	()layers.Flatten
לוקח את התוצאות הסופיות ומוריד מהן את המספרים העשרוניים לכל הפחות 2 מספרים אחרי הנקודה	layers.Dense(num_classes)
מסכם את תוצאות המודל	()model.summary

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 216, 216, 3)	0
conv2d (Conv2D)	(None, 216, 216, 16)	448
max_pooling2d (MaxPooling2D)	(None, 108, 108, 16)	0
conv2d_1 (Conv2D)	(None, 108, 108, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 54, 54, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 27, 27, 64)	0
flatten (Flatten)	(None, 46656)	0
dense (Dense)	(None, 128)	5972096
dense_1 (Dense)	(None, 2)	258
Total params: 5,995,938		
Trainable params: 5,995,938		
Non-trainable params: 0		

בתוך המודל יש קוד אשר מייצר גרפים אשר מראים את תוצאות האימון:



לבסוף המודל המאומן נשמר בתור קובץ

test

המודל ישתמש במודל המאומן ויבדוק את התוצאות השונות, המודל יופעל בעזרת לחיצת על כפתור test the model
הפעולה בודקת בהתחלה האם יש מודל מאומן מלכתחילה ואם אין היא מחזירה
'Cant load the saved_model.h5 file'
לאחר מכן היא תיקח 10 תמונות רנדומליות ותבחן את התוצאות עליהן
ולבסוף תראה את התמונה ואת אחוזי ההצלחה בעזרת המודל המאומן

car - 73 %



רפלקציה

העבודה על הפרויקט הייתה קשה אך מעניינת ומסקרנת מכיוון שאת החומר הנלמד אין מאמרים או מקורות מידע בשפה העברית והייתי צריך ללמוד הכל מאפס מכיוון שהנושא הוא נושא חדש יחסית גם בעולם התכנות כולו ולכן קשה יותר למצוא נתונים אך מאוד הנושא מאוד עניין ומשך אותי ללמוד עוד.

במהלך הפרויקט למדתי המון מושגים ומונחים חדשים בעולם התכנות ואף פיתחתי דרכי חשיבה חדשות אשר לא הייתי מפתח בתכנות "רגיל" למדתי המון על עולם הבינה המלאכותית וכיצד היא עובדת, בנוסף איך הטכנולוגיות באות לידי שימוש במיוחד בשנים האחרונות עם התפתחות הנושא ויצירת אפשרויות חדשות.

הכלים שאני לוקח לעצמי מעבודה על הפרויקט הם איך לאסוף מידע לבד וללמוד נושא שלם מאפס בלי עזרה מיוחדת, למדתי כיצד לנהל את סדר וארגון העבודה שלי על מנת שאוכל לעמוד בזמנים.

הקשיים שחוויתי במהלך הפרויקט היו למצוא תמונות רבות על מנת שהדאטה שלי יהיה מספיק גדול על מנת שאוכל לאמן את המודל שלי ושיגיע לתוצאות טובות בנוסף לכך במהלך הפרויקט חוויתי מקרה של **overfitting** אשר גרם לכך שהתוצאות שלי לא יהיו הגיוניות ולאחר מחקר מעמיק גיליתי שיש לי כפילויות של תמונות רבות בקבצי התמונות שלי ולאחר שהצלחתי לאתר את כולן ולמחוק אותן התוצאות שלי היו טובות ומדויקות.

בנוסף לקח לי הרבה זמן עד שהצלחתי לבנות את המודל המושלם עבור הפרויקט שלי על מנת שאוכל לקבל את התוצאות הטובות ביותר שאני יכול לקבל.

המסקנות שלי מהפרויקט הן שעולם התכנות עובר מהפכה מיוחדת שתשנה את פני העולם בעוד כמה שנים, עולם הבינה המלאכותית מתפתח בקצב מסחרר ולשם פנינו מועדות.

אם הייתי מתחיל את הפרויקט היום הייתי משנה את סדרי העדיפויות שלי במהלך העבודה על הפרויקט ומתעמק יותר בפיתוח המודל שלי ופחות ביצירת הדאטה המורחב, וגם הייתי מנהל את הזמן שלי יותר טוב על מנת שהייתי מסיים את הפרויקט לפני.

העבודה הייתה יותר יעילה עבורי אם היה לי יותר ידע מקדים על הנושא ובכך הייתי יכול לפתח יותר את השאלה של הפרויקט שלי ולעשות פרויקט יותר מעניין אך מכיוון שהייתי צריך ללמוד הכל מאפס לא היה לי מספיק ידע לעשות פרויקט מדרגת קושי גבוהה יותר.

ביבליוגרפיה

- BARIS DINCER Vehicle Detection Image(2020,3 june).kaggle
<https://www.kaggle.com/datasets/brsdincer/vehicle-detection-image-set>
- Amos, D. Python GUI Programming With Tkinter. Real Python.
[/https://realpython.com/python-gui-tkinter](https://realpython.com/python-gui-tkinter)
- Chollet, F (2016, June 5). Building powerful image classification models using very little data. The Keras Blog.
<https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html>
- Image data preprocessing. K Keras. <https://keras.io/api/preprocessing/image/>
- Narkar, M (2019, August 6). Image classification with Convolution Neural Networks (CNN)with Keras. Medium <https://medium.com/@manasnarkar/image-classification-with-convolution-neural-networks-cnn-with-keras-dbd71c05ed2a>
- Rosebrock, A (2017, December). Image classification with Keras and deep learning. Pyimagesearch. <https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deep-learning/>
- Shorten, C (2018, October 15). Image Classification Keras Tutorial: Kaggle Dog Breed Challenge. Towards Data Science <https://towardsdatascience.com/image-classification-python-keras-tutorial-kaggle-challenge-45a6332a58b8>
- Stack Overflow. <https://stackoverflow.com/>
- Tarang, S (2017, December 6). About Train, Validation and Test Sets in Machine Learning. Towards data science. <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>