**פרויקט גמר 5 יחידות לימוד**

**התמחות – למידת מכונה**

**Deep Learning**

**נושא הפרויקט**

**זיהוי וסיווג מכוניות 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**

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

תוכן

[מבוא 3](#_Toc107420727)

[בסיס תאורטי 4](#_Toc107420728)

[**מדריך למשתמש** 7](#_Toc107420729)

[**ארכיטקטורה של הפרויקט** 14](#_Toc107420730)

[main 18](#_Toc107420731)

[**Options** 19](#_Toc107420732)

[Windows 20](#_Toc107420733)

[**Prepare\_data** 21](#_Toc107420734)

[**Train** 22](#_Toc107420735)

[**test** 25](#_Toc107420736)

[רפלקציה 26](#_Toc107420737)

[ביבליוגרפיה 27](#_Toc107420738)

# 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 Deep learning 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 מכוניות הנמצאות בתמונות**. המודל לומד המון תמונות של מכוניות ותמונות שאין בהן מכוניות ולאחר תהליך למידה הוא יוכל לנתח תמונות שונות וכאשר ימצא מכונית 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 מכוניות ברמות דיוק גבוהות, ואכן ישנן תוכנות ומוצרים (כגון:Mobileye (מבוססות learning machine בעלות היכולת לזהות מכוניות

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 נויטרונית, 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 מאחוריי learning deep ובמקביל ללמוד את הדרך המעשית ליצירת פרויקט המתבסס על נושא זה. יתרה מכך, זהו נושא שיחסית חדש בלימודים

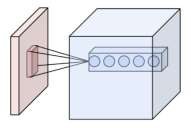
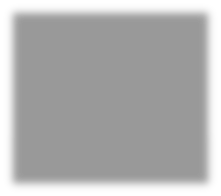
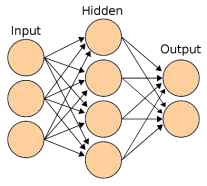
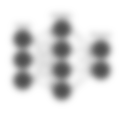
התיכוניים בכלל ובבית הספר שלי בפרט, לכן גם עבור המורה, גם עבור בית הספר, וגם עבור התלמידים הוא חדש יחסית. כלומר, התבקשנו לכתוב תוכנה המבצעת דבר שמלכתחילה לא ידענו כיצד לבצעו ועל כן למדנו תוך כדי פעולה. התגברתי על אתגר זה בכך שאני וחבריי להתמחות למדנו את הנושא תוך כדי

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 קונבלוציה-היא סוג של רשת נוירונים המשתמשת בפעולת הקונבלוציה 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 data validation 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 וfitting- :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 הaccuracy- validation גבוה משמעותית מן

הaccuracy- training או כאשר הloss- validation קטן משמעותית מן הloss- .training

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

accuracy או כאשר הloss- validation גבוה משמעותית מן הloss- .training

-**epoch** חלוקה לepochs- 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 -Python** [**https://www.python.org/downloads/**](https://www.python.org/downloads/)

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

**באמצעות הפקודה: -V python\**

**יש להוריד מספר ספריות קוד בהן הפרויקט**

|  |  |  |
| --- | --- | --- |
| **קישור** | **פקודת התקנה** | **שם הספרייה** |
| [**https://pypi.org/project/Keras/**](https://pypi.org/project/Keras/) | **pip install keras** | **keras** |
| [**https://pypi.org/project/tensorflow/**](https://pypi.org/project/tensorflow/) | **pip install tensorflow** | **tensorflow** |
| [**https://pypi.org/project/matplotlib/**](https://pypi.org/project/matplotlib/) | **pip install matplotlib** | **matplotlib** |
| [**https://pypi.org/project/numpy/**](https://pypi.org/project/numpy/) | **pip install numpy** | **numpy** |
| [**https://pypi.org/project/glob2/**](https://pypi.org/project/glob2/) | **pip install glob** | **glob** |
| [**/https://pypi.org/project/Pillow**](https://pypi.org/project/Pillow/) | **pip install Pillow** | **PIL** |
| [**https://pypi.org/project/pathlib/**](https://pypi.org/project/pathlib/) | **Pip install pathlib** | **pathlib** |
| [**https://pypi.org/project/pytest-shutil/**](https://pypi.org/project/pytest-shutil/) | **Pip install shutil** | **shutil** |
| [**https://www.geeksforgeeks.org/how-to-install-tkinter-in-windows/**](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:**

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

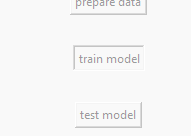
**קודם כל נצטרך לעשות prepare data על מנת שכל הקבצים יגיעו למקומות הדרושים (path)**

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

**תמונה שמכילה טקסט

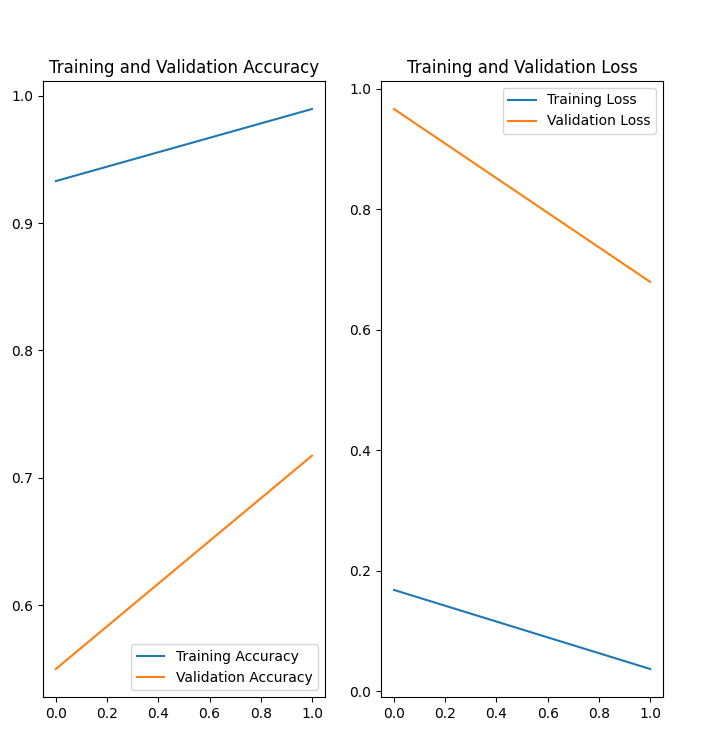
התיאור נוצר באופן אוטומטי**

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

****

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

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

**לבסוף תעלה החלונית הבאה והמודל ישמר כפי שניתן לראות**

****

**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**

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

**ונראה תוצאות של התמונותתמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

# **ארכיטקטורה של הפרויקט**

**איסוף הנתונים:**

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

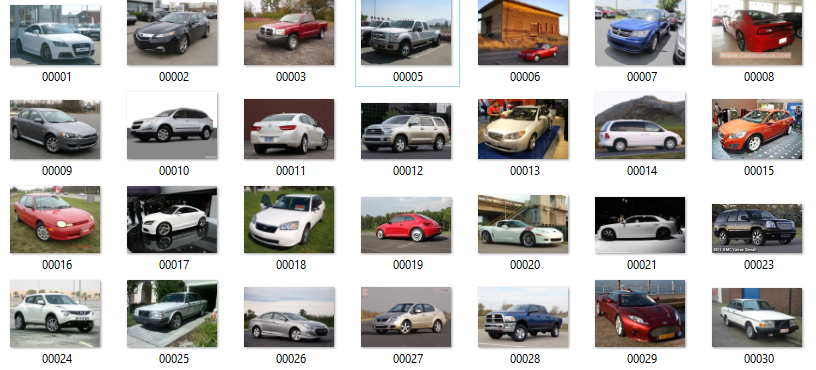
**\*יש לציין שבקודים הבאים איני משתמש בפרויקט עצמו אלא השתמשתי בהם לפני על מנת לסדר לעצמי את הדאטהסט**

Change name to file

הקוד לוקח את התמונות בקובץ ומשנה את שמן

zero\_filled\_number- מוסיף אפסים במידת הצורך לשם התמונה

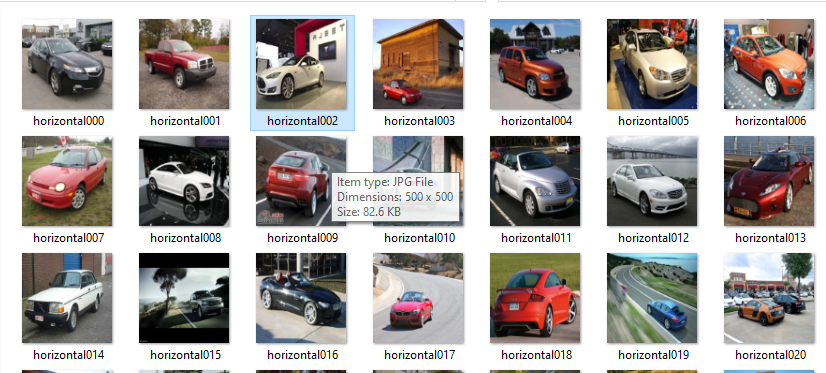
הקוד הבא לוקח את כל התמונות בקובץ הנתון ומשנה את שמם למסודר לפי אפסים דוגמא:



Image\_flip\_horizontal

הקוד לאחר מכן הופך את התמונות בתמונת מראה ושומרת אותם מחדש

- הפעולה הופכת אותה לתמונת מראהcv2.flip

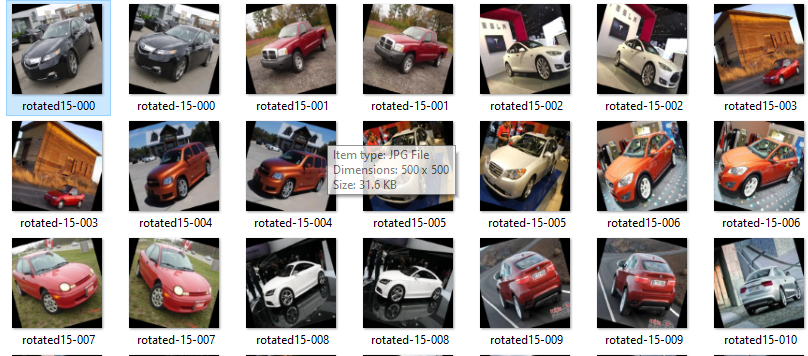


Rotate\_image

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:

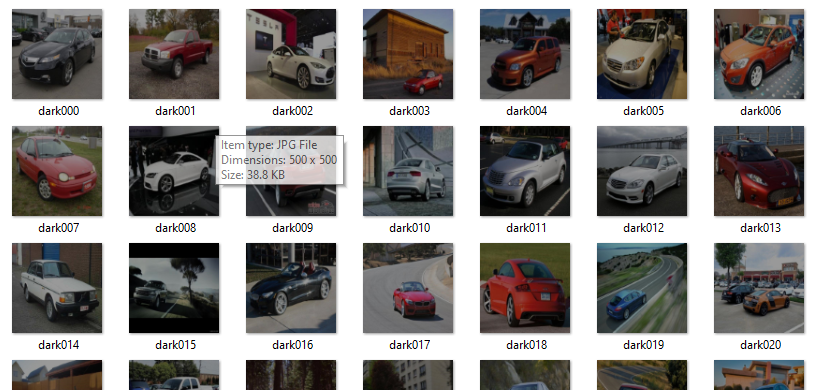


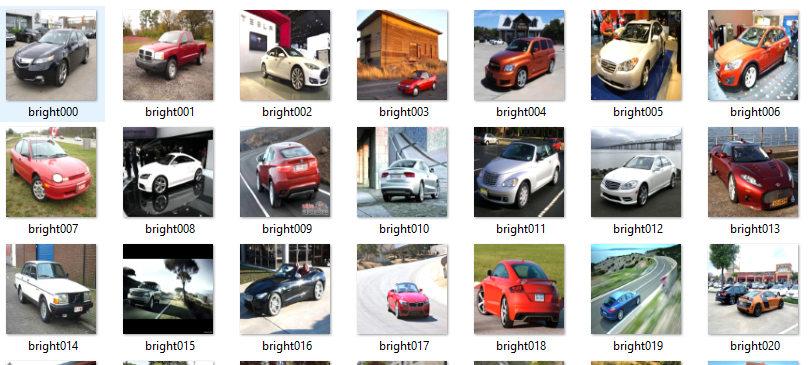
ובעזרת הקוד הבא

Brightness change

enhancer = ImageEnhance.Brightness(im)- יוצר אפשרות לשינוי בהירות בתמונה הנתונה והפקטור קובע את הבהירות(שבר יוצר יותר קהירות ומעל 1 גורם לתמונה להיות בהירה יותר)  
factor = 0.6

שיניתי את הבהירות של התמונות:





לאחר מכן שילבתי את כל התמונות לדאטה סט גדול וחילקתי את התמונות לטסט, טריין( trainוולדיישן

Validation



# main

דרך הפעולה הזאת נפעיל את כל הפרויקט ויקפוץ החלון הבא

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

# **Options**

דרך מודל זה נוכל לשנות את האופציות שבהם המודל ישתמש

|  |  |
| --- | --- |
| DATA\_DIR | התיקייה שבה ימצאו הדאטהסט שבו ישתמש המודל על מנת להתאמן |
| NUM\_CLASSES | כמות המחלקות שמודל יסווג |
| IMAGE\_HEIGHT | אורך התמונות שיהיו בדאטהסט |
| IMAGE\_WIDTH | רוחב התמונות שיהיו בדאטהסט |
| BATCH\_SIZE | כמות התמונות שהמודל יתאמן עליהם בכל אימון |
| EPOCHS | כמות הפעמים שבהם המודל יאמן את עצמו |
| TRAIN\_SIZE | היחס בו התמונות מהדאטהסט יחולקו לאימון |
| TEST\_SIZE | היחס בו התמונות מהדאטהסט יחולקו לבדיקת התוצאות |

# Windows

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

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

# **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**

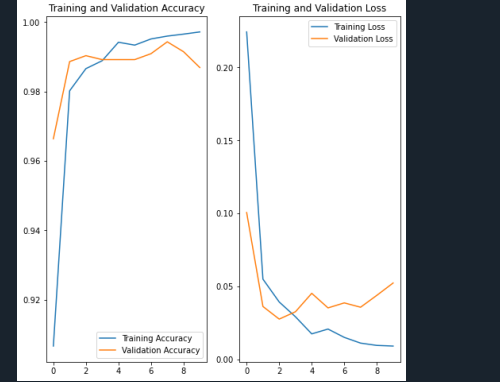
המודל דרכו נאמן את המודל בעזרת הדאטהסט, המודל יופעל בעזרת הכפתור train the model

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

תמונה שמכילה טקסט, צג, צילום מסך, מסך

התיאור נוצר באופן אוטומטי

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 רנדומוליות ותבחן את התוצאות עליהן

ולבסוף תראה את התמונה ואת אחוזי ההצלחה בעזרת המודל המאומן

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

# 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 יותק 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 שהדאטהסט 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 כפיליות של תמונות רבות בקבצי התמונות שלי ולאחר שהצלחתי לאתר את כולן ולמחוק אותן התוצאות שלי היו טובות ומדוייקות.**

**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 הדאטהסט 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 הפרוייקט 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**](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-**](https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html)[**little-data.html**](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/**](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-**](https://medium.com/%40manasnarkar/image-classification-with-convolution-neural-networks-cnn-with-keras-dbd71c05ed2a)[**neural-networks-cnn-with-keras-dbd71c05ed2a**](https://medium.com/%40manasnarkar/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-**](https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deep-learning/)[**and-deep-learning/**](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-**](https://towardsdatascience.com/image-classification-python-keras-tutorial-kaggle-challenge-45a6332a58b8)[**kaggle-challenge-45a6332a58b8**](https://towardsdatascience.com/image-classification-python-keras-tutorial-kaggle-challenge-45a6332a58b8)

**Stack Overflow.** [**https://stackoverflow.com/**](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**](https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7)