פרויקט גמר יחידות גמר

התמחות – תכנון ותכנות מערכות

Deep learning letter identifier

מגיש: ניר עמית

ת.ז: 213409618

בית ספר: מקיף י"א ראשונים ראשון לציון

כיתה: י"ב 4

מורה: דינה קראוס

תאריך בחינה: 18.6.2020





תוכן עניינים:

מבוא..............................................................................................................3 - 4

מדריך למשתמש .............................................................................................5-11

מדריך למפתח...............................................................................................12-14

מסקנות מהרצת המודל......................................................................................15

מסקנות מהרצת המודל......................................................................................16

רפלקציה אישית.................................................................................................17

נספחים: .....................................................................................................18-19

מבוא:

מטרת הפרויקט:

השנה בבית ספרי במסגרת מסלול פיילוט התבקשנו לבצע את פרויקט הגמר שלנו בנושא "למידה עמוקה" (Vision Computer Learning Deep). תחום זה הינו תחום רק גוני העוסק ביכולת לדמות באופן מלאכותי את הליך הלמידה האנושית.

המוח האנושי מורכב מאלפי נוירונים (סוג של תא) כאשר כל תא בפני עצמו אינו מסוגל לבצע פעולה מסובכת כמו קריאה או כתיבה, אך על ידי כל שכל אחד מן הנוירונים עושה פעולה קטנה, ניתן להשיג פעולה גדולה ומורכבת, אשר ערכה גדול מסך חלקיה, כמו לדוגמה קריאה. רשת נוירונים עובדת באופן דומה; היא מורכבת ממאות משקלים אשר כל אחד מהם מבצע פעולה מתמטית פשוטה, אך כאשר בוחנים את סכם פעולתם מסוגלת התוכנה להבין על מה היא מסתכלת בדיוק יחסי. חשוב להבין שב Deep learning איננו כותבים את התוכנה המזהה באופן ידני, אם כי אנו כותבים תוכנה אשר יוצרת תוכנה (הנקראת מודל) אשר מסוגלת לקרוא תמונות

חשוב להבין שאנו משתמשים ב Deep learning מכיוון שלכתוב באופן ידני תוכנה אשר תזהה את התמונה שעליה היא מסתכלת זה בלתי אפשר. יש יותר מדי שונות בין תמונות, גם אם הן תמונות של אותו דבר בשביל שיהיה ניתן לתכנת תוכנה באמצעות לוגיקה מסורתית שתוכל לפענח תמונות. על כן אנו משתמשים ב Deep learning .

בפרויקט זה בחרתי לבנות תוכנה אשר תוכל לזהות אותיות כתב באנגלית. בחרתי בנושא זה מכיוון שאני רואה חשיבות רבה בהעשרת הידע האנושי, והשגת מטרה זו על ידי העלאה לאינטרנט של ספרים רבים ככל האפשר. מן הסתם אם ניתן לבן אדם לכתוב את מה שכתוב בספר ייקח לו המון זמן להקליד זאת למחשב, אך אם נצלם את דפי הספר נוכל להעלות אותם לאינטרנט הרבה יותר מהר, ולשם כך דרושה תוכנה אשר תוכל להבין איזה אותיות היא רואה (אשר לרוב אינן ברורות בתצלום).

במהלך הפרויקט נתקלתי במספר אתגרים, אשר העיקרי ביניהם היה להבין איך להתאים את התוכנה לפרויקט הספיציפי שלי, מכיוון שאפילו שיש הרבה מדריכים באינטרנט על איך לכתוב תוכנה מסוג זה, מעט מאוד מהם מספקים הסברים בנוגע למה שהם עושים

דרישות להרצת התוכנה:

פייתון 3.7

מערכת הפעלה Windows

המודל משתמש בתמונות מהספרייה emnist, אשר איתם באים קבצים בהם כתובה המשמעות של כל תמונה (על מנת שהמחשב ידע על איזה אות הוא מסתכל וכך יוכל למצוא היגיון)

הספריות הבאות:

|  |  |  |
| --- | --- | --- |
| Library name | Installation command | Download link |
| Tensorflow | Pip install tensorflow | <https://pypi.org/project/tensorflow/> |
| Keras | Pip install keras | <https://pypi.org/project/Keras/> |
| Numpy | Pip install numpy | <https://pypi.org/project/numpy/> |
| Emnist | Pip install emnist | <https://pypi.org/project/emnist/> |
| Sklearn | Pip install sklearn-learn | <https://pypi.org/project/scikit-learn/> |
| Opencv | Pip install opencv-python | <https://pypi.org/project/opencv-python/> |
| matplotlib | Pip install matplotlib | <https://pypi.org/project/matplotlib/> |

מדריך למשתמש:

הוראות התקנה:

1 .יש להוריד 7.3 Python או גרסה עדכנית יותר -

https://www.python.org/downloads/

אם יש לך Python על המחשב, בדוק מהי גרסתו

באמצעות הפקודה: python - V

2 .יש להתקין במחשב את סביבת העבודה Anaconda, אך ניתן להשתמש בכל סביבת עבודה אחרת כל עוד משתמשים ב interpreter של אנקונדה.

לינק להורדה:

https://www.anaconda.com/products/individual - Anaconda

3 .יש להוריד מספר ספריות קוד אשר בהן הפרויקט משתמש:

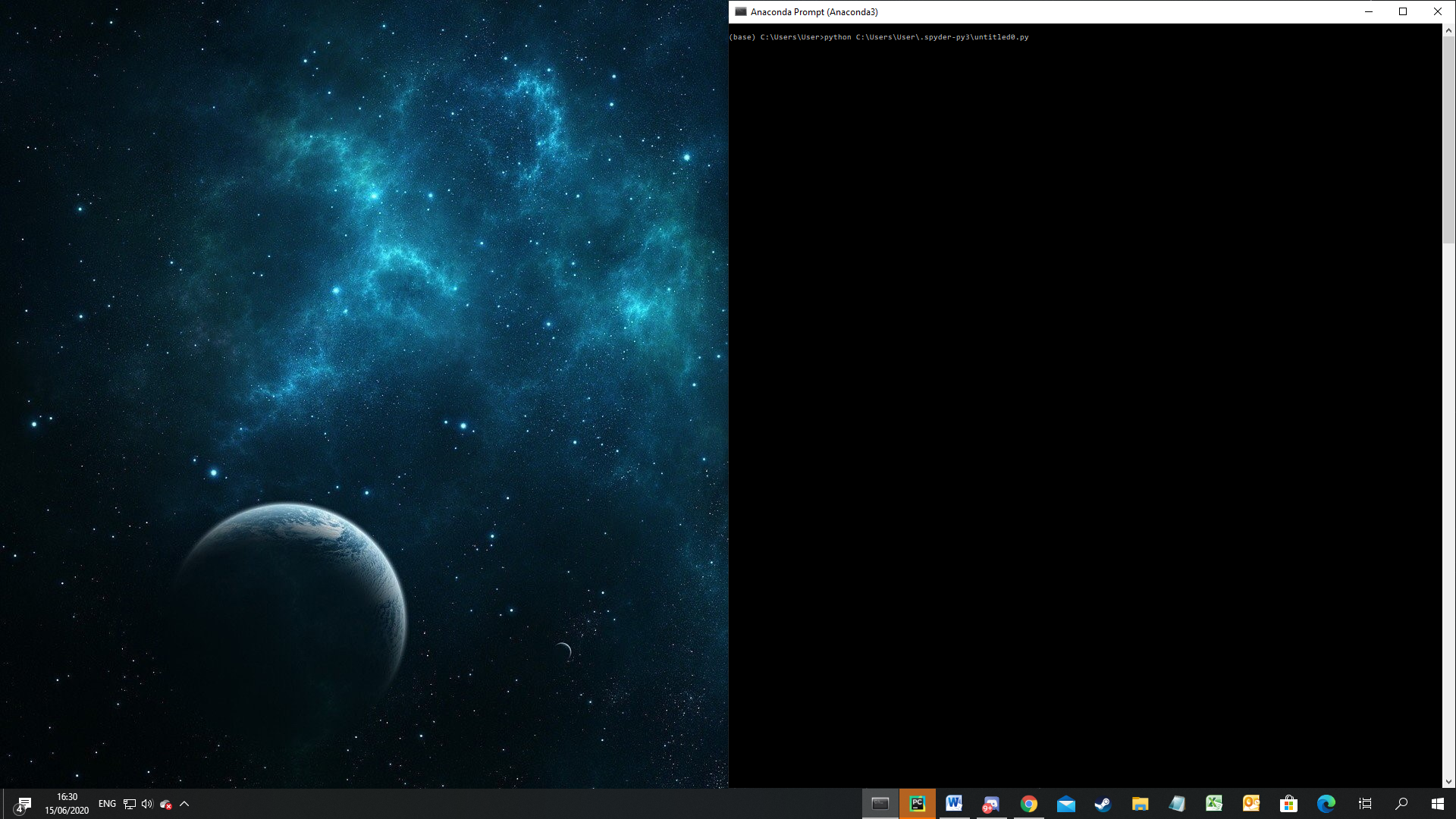
|  |  |  |
| --- | --- | --- |
| Library name | Installation command | Download link |
| Tensorflow | Pip install tensorflow | <https://pypi.org/project/tensorflow/> |
| Keras | Pip install keras | <https://pypi.org/project/Keras/> |
| Numpy | Pip install numpy | <https://pypi.org/project/numpy/> |
| Emnist | Pip install emnist | <https://pypi.org/project/emnist/> |
| Sklearn | Pip install sklearn-learn | <https://pypi.org/project/scikit-learn/> |
| Opencv | Pip install opencv-python | <https://pypi.org/project/opencv-python/> |
| matplotlib | Pip install matplotlib | <https://pypi.org/project/matplotlib/> |

4. יש להוריד מהgithub שלי את קובץ הפייתון ששמתי שם

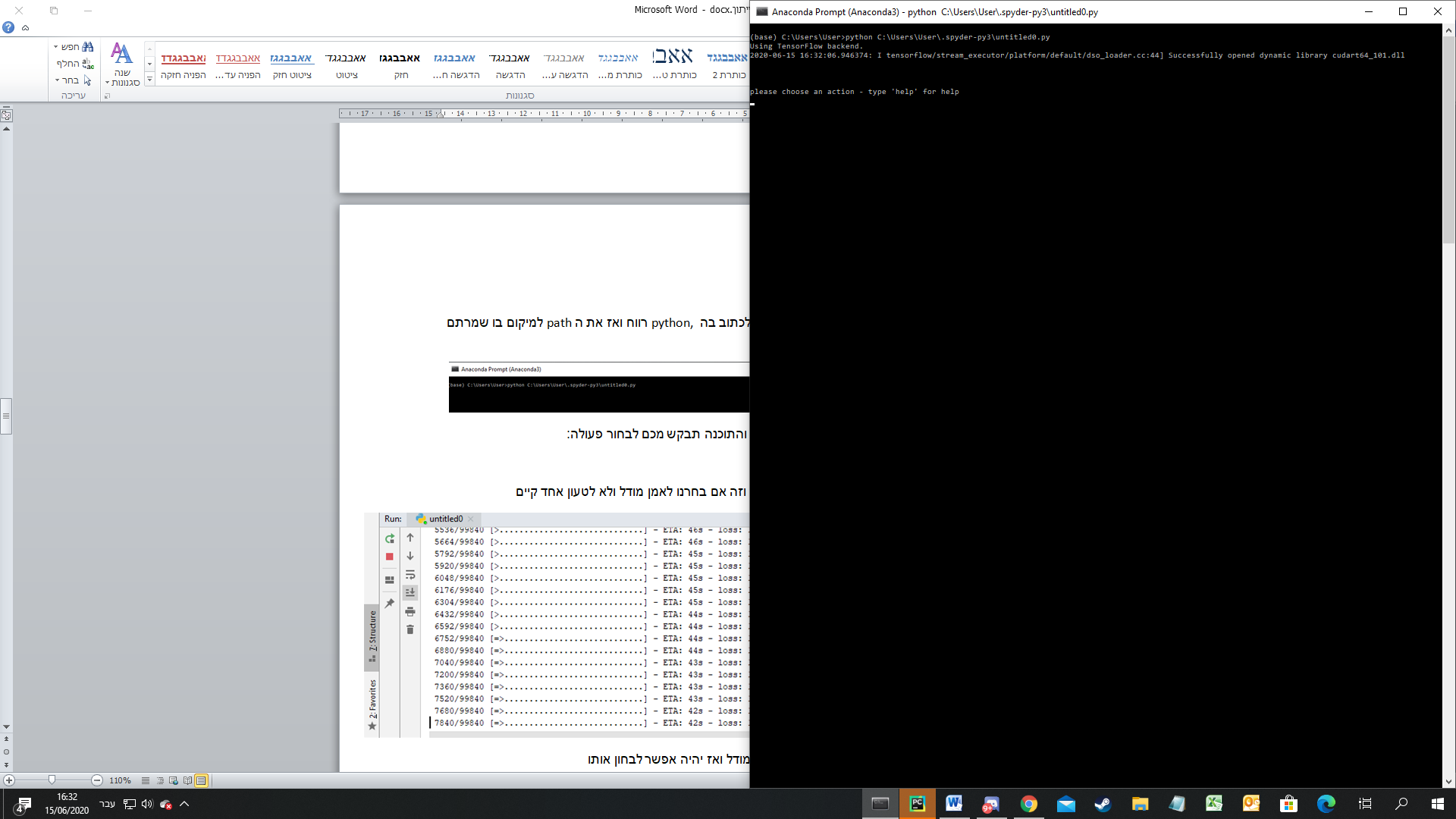
5. צריך חיבור אינטרנט

שלבי התוכנה:

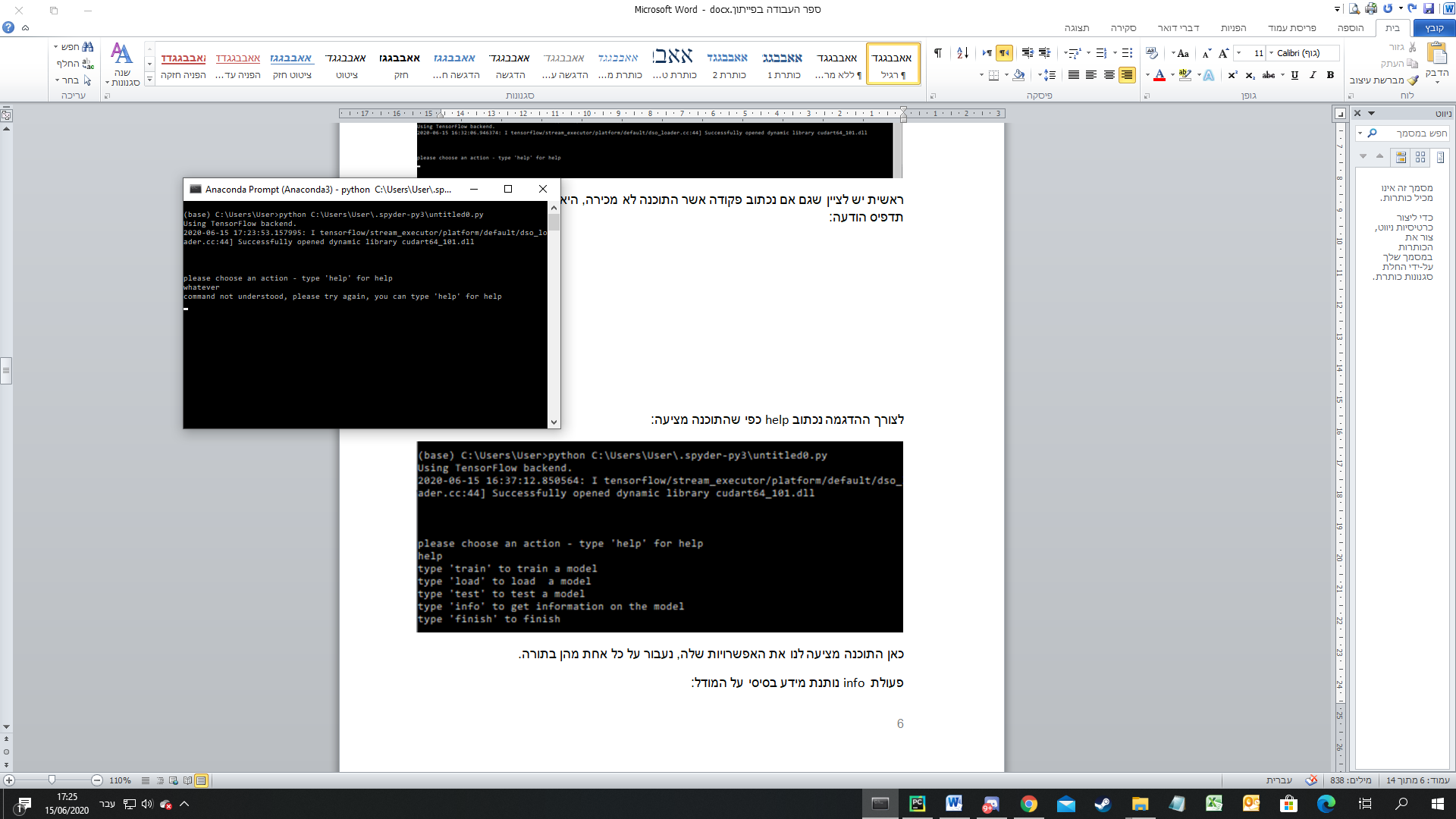
יש לפתוח את שורת הפקודה של אנקונדה ולכתוב בה python, רווח ואז את הpath למיקום בו שמרתם את הקובץ



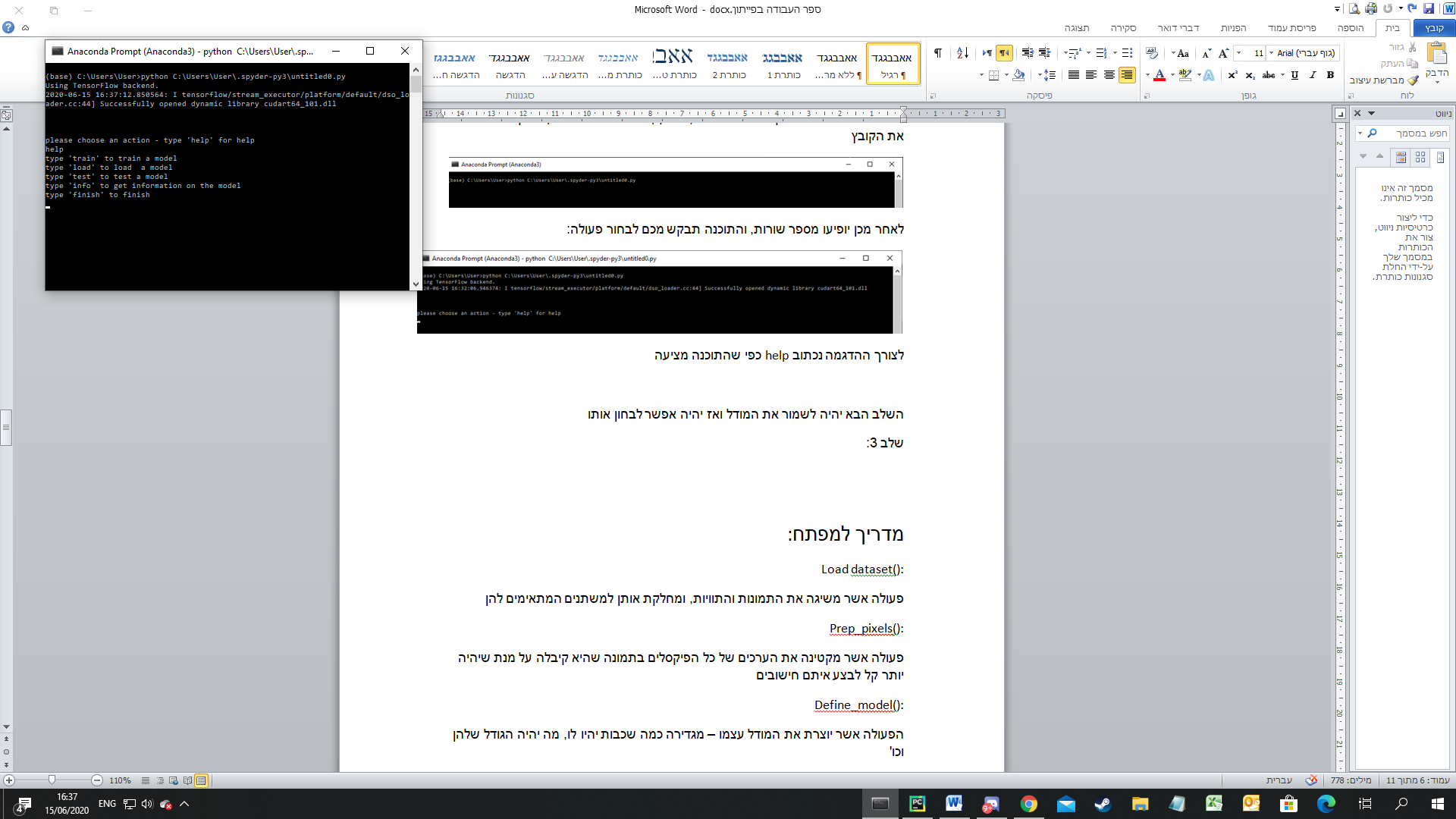
לאחר מכן יופיעו מספר שורות, והתוכנה תבקש מכם לבחור פעולה:



ראשית יש לציין שגם אם נכתוב פקודה אשר התוכנה לא מכירה, היא לא תקרוס, אלא תדפיס הודעה:

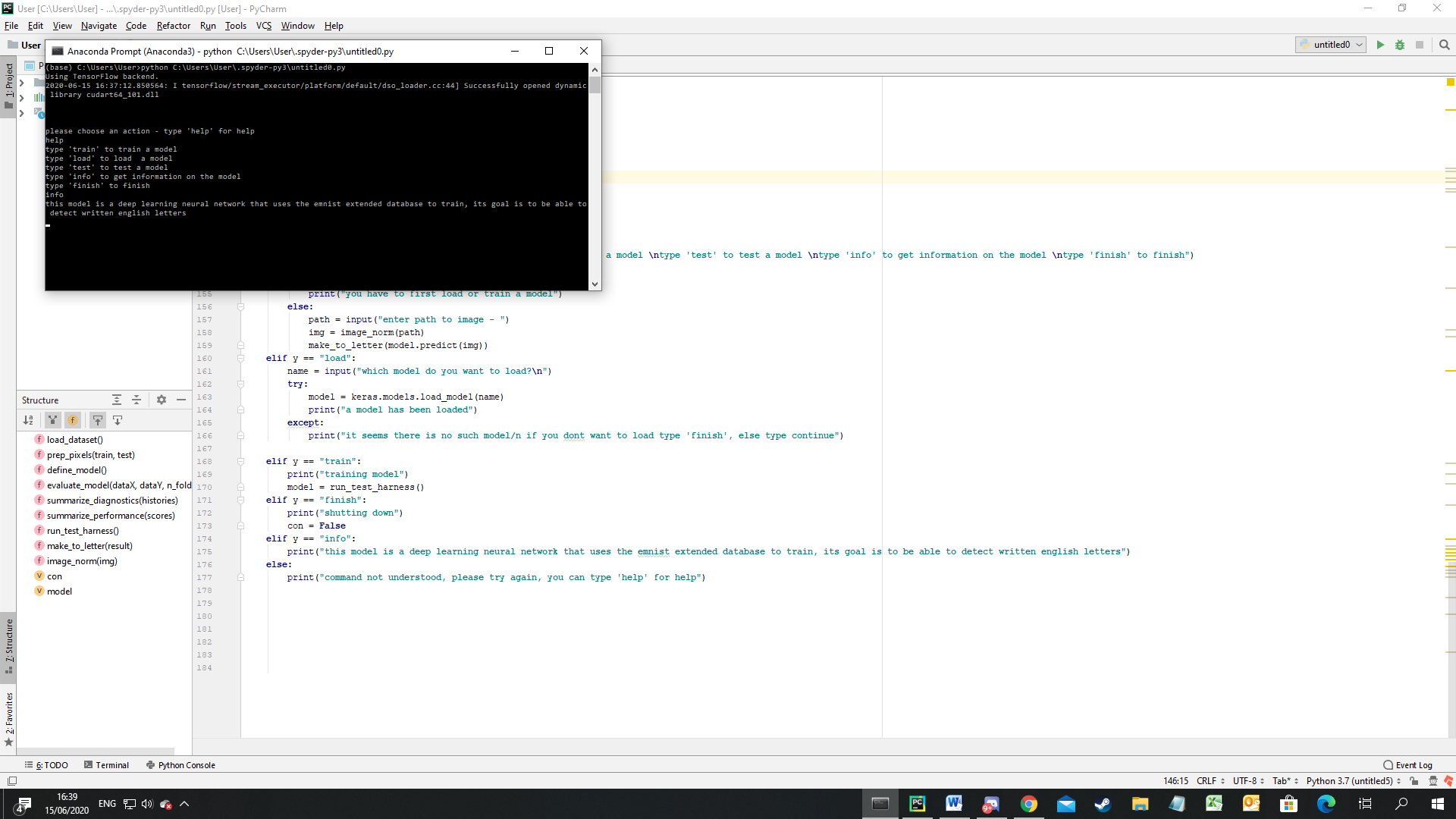


לצורך ההדגמה נכתוב help כפי שהתוכנה מציעה:

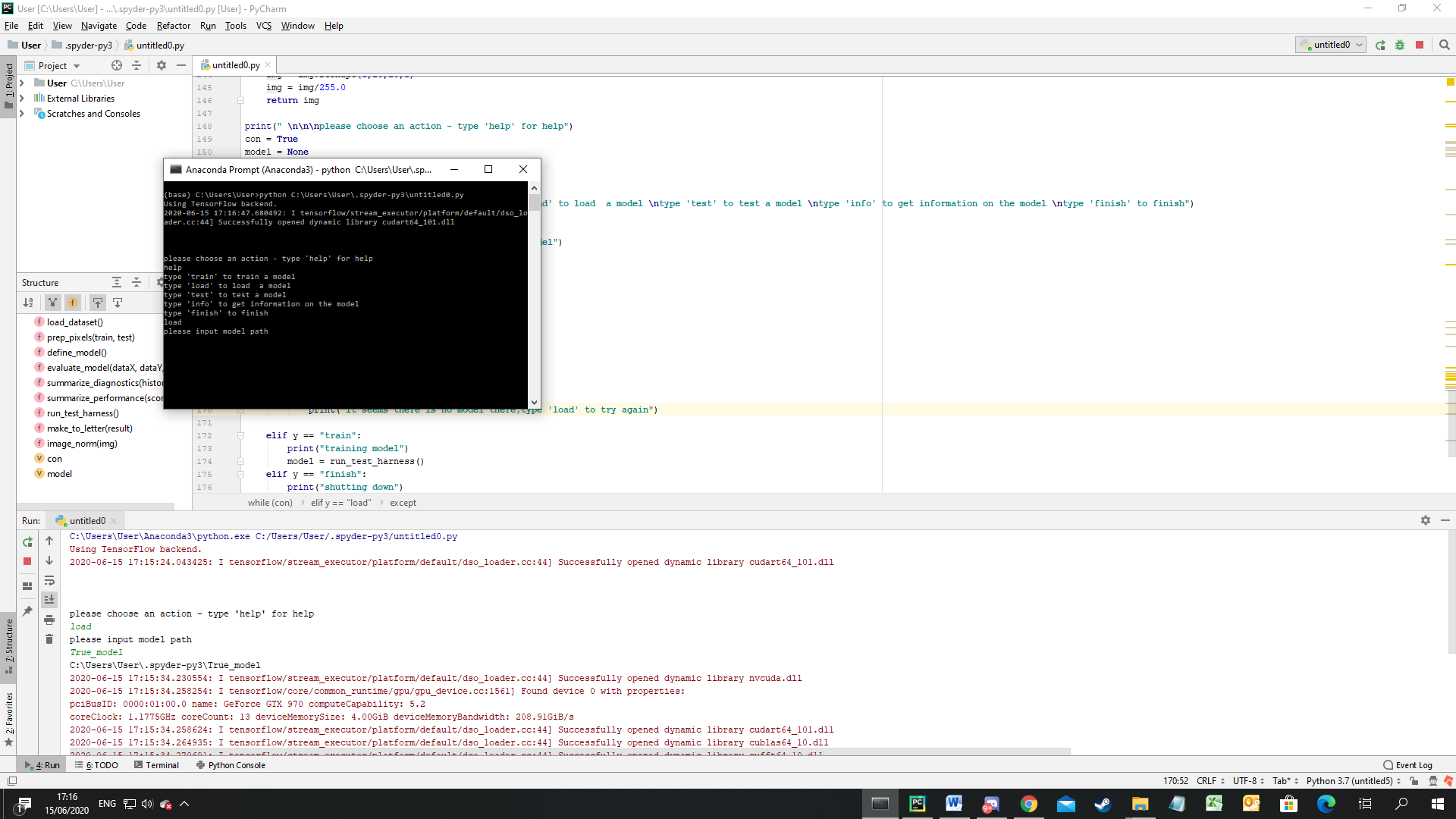


כאן התוכנה מציעה לנו את האפשרויות שלה, נעבור על כל אחת מהן בתורה.

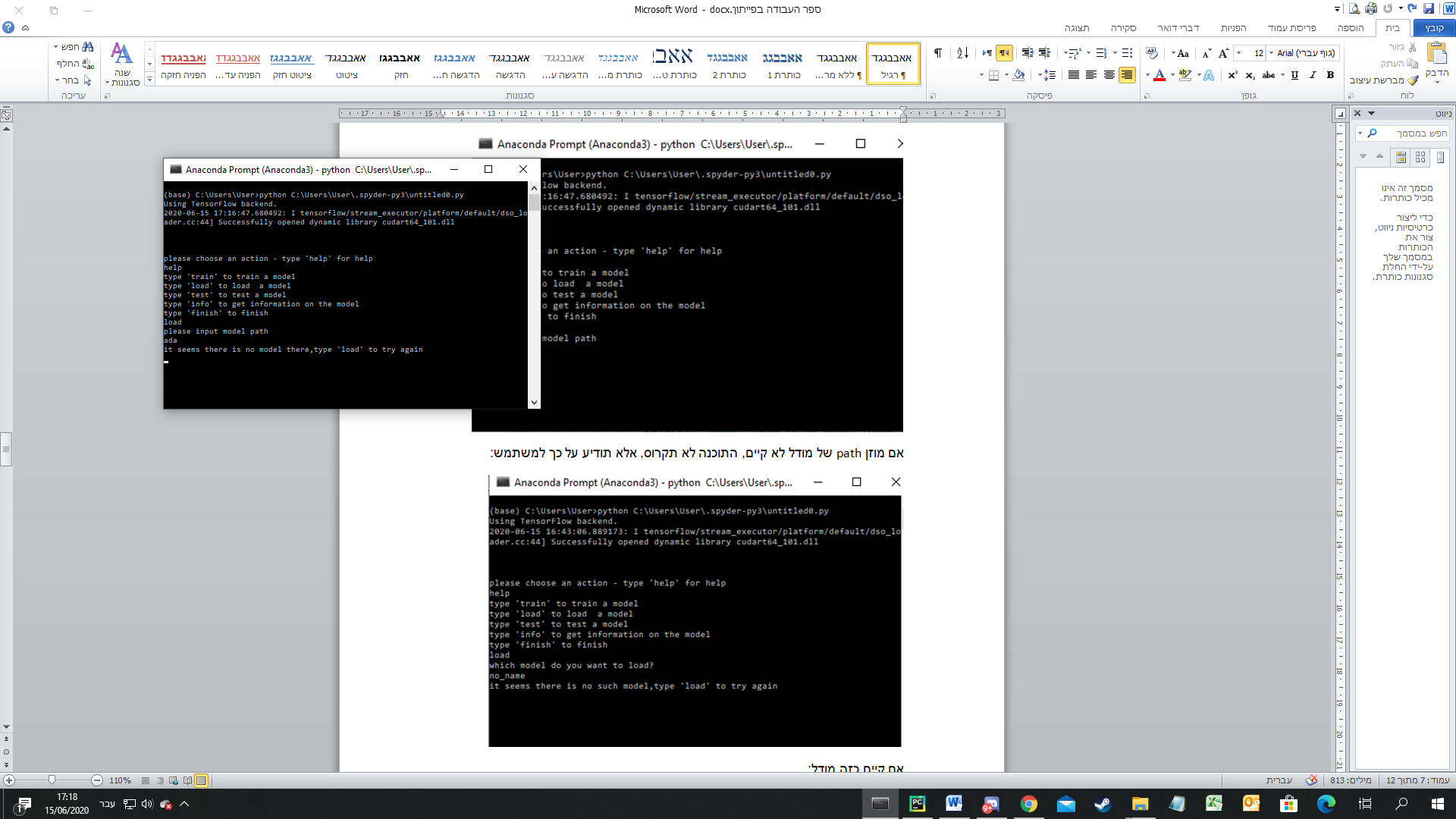
פעולת info נותנת מידע בסיסי על המודל:



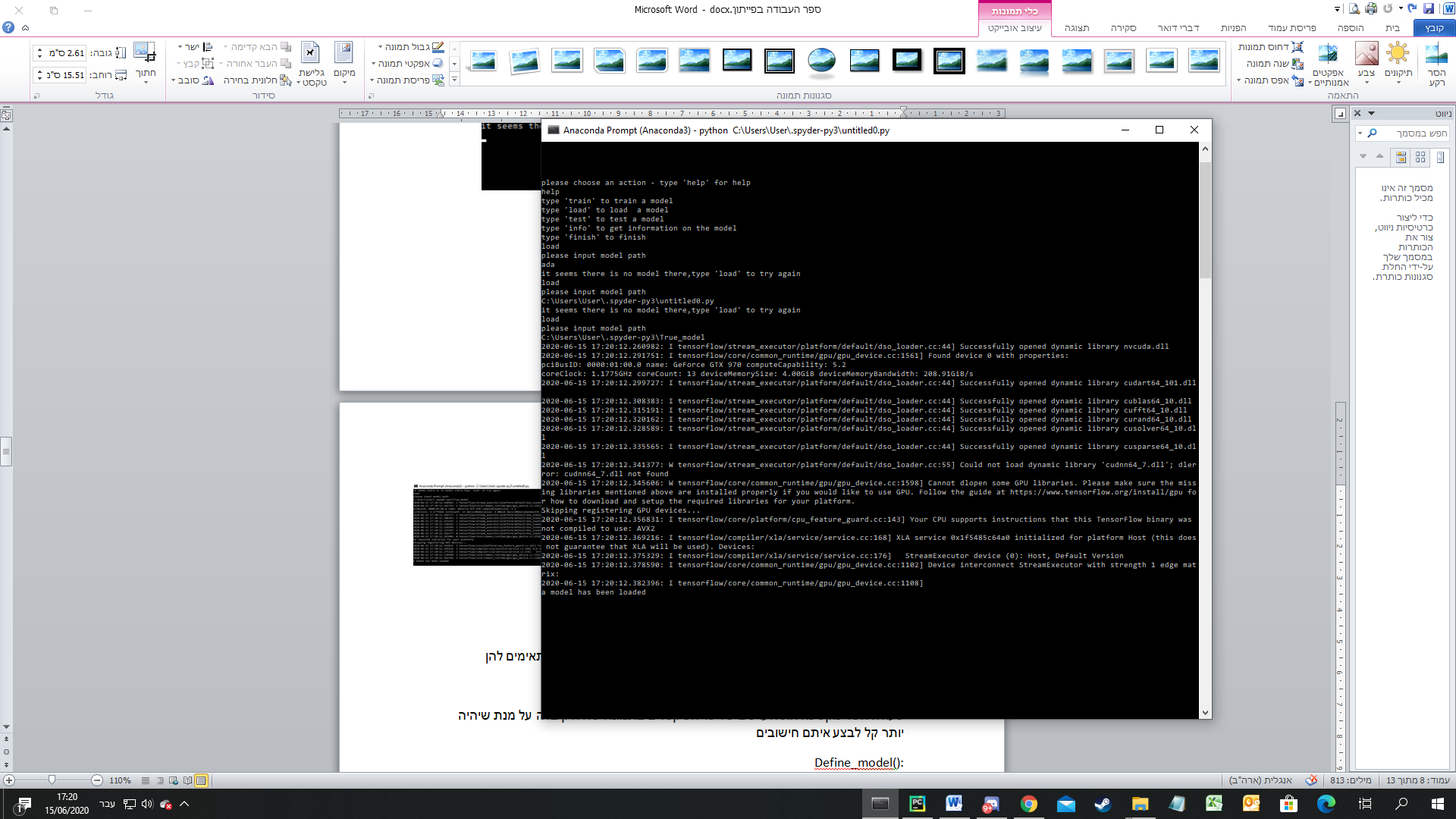
פעולת load מאפשרת להעלות מודל קיים לשימוש:



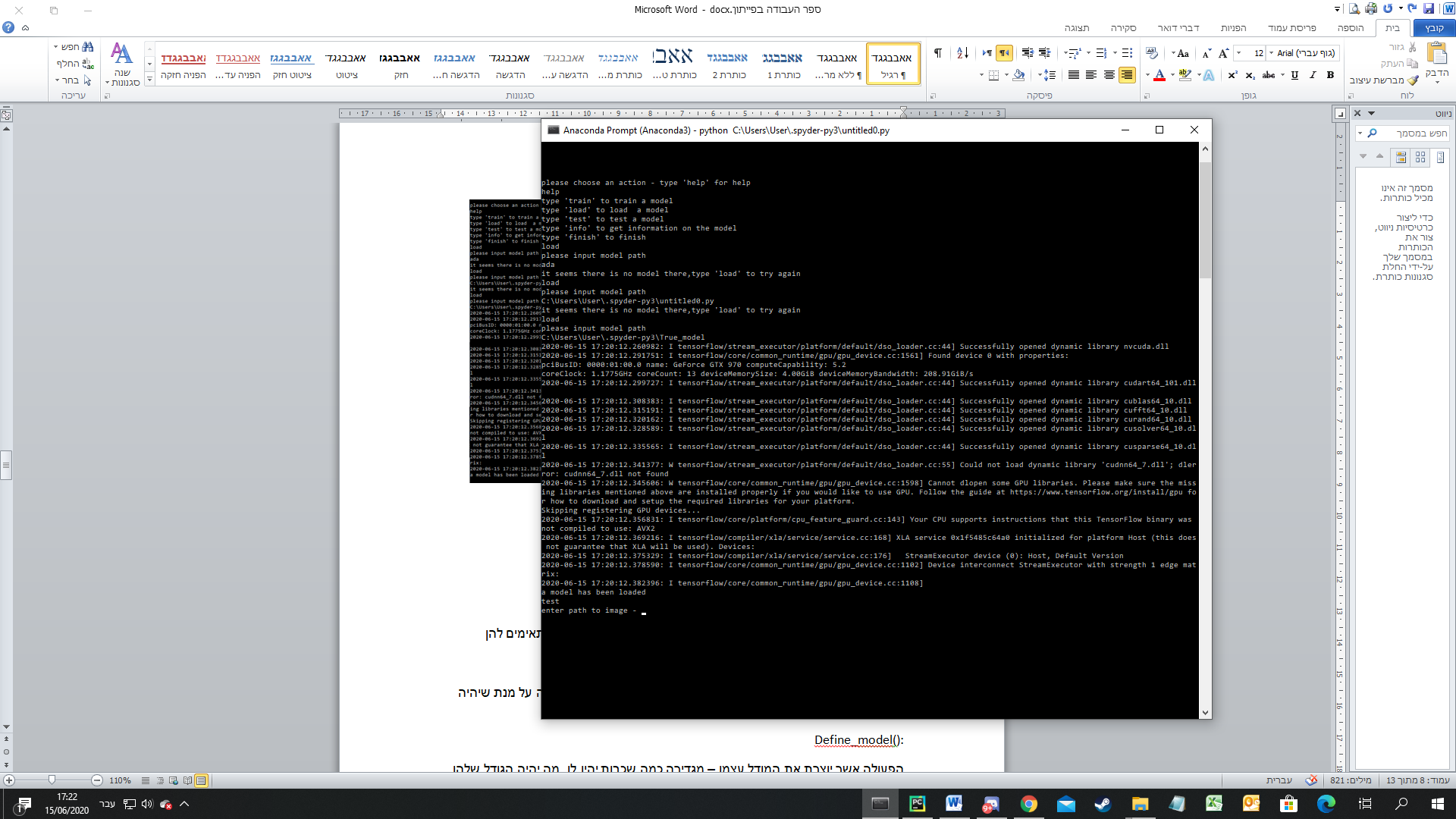
אם מוזן path של מודל לא קיים, התוכנה לא תקרוס, אלא תודיע על כך למשתמש:



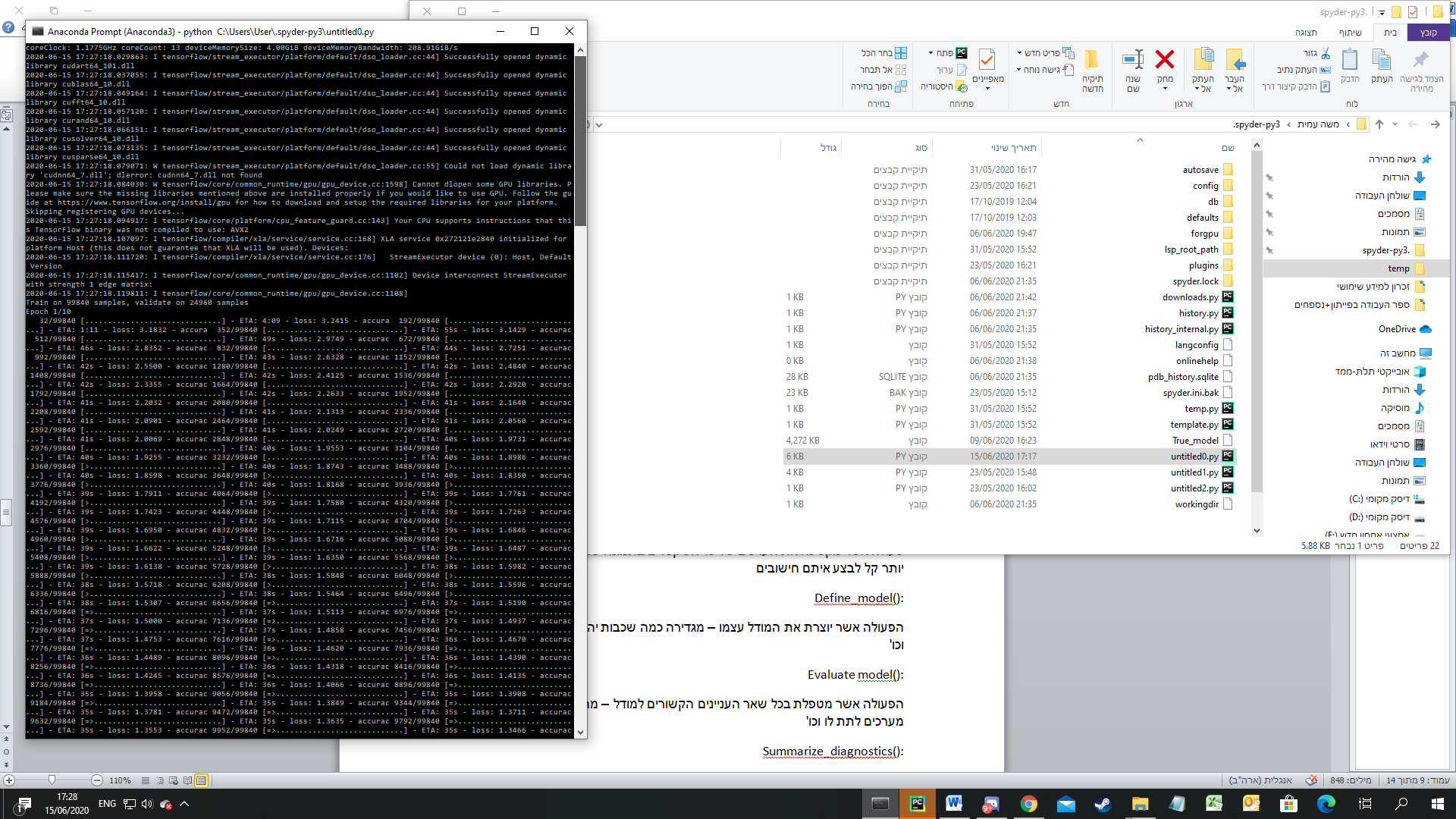
אם קיים כזה מודל:



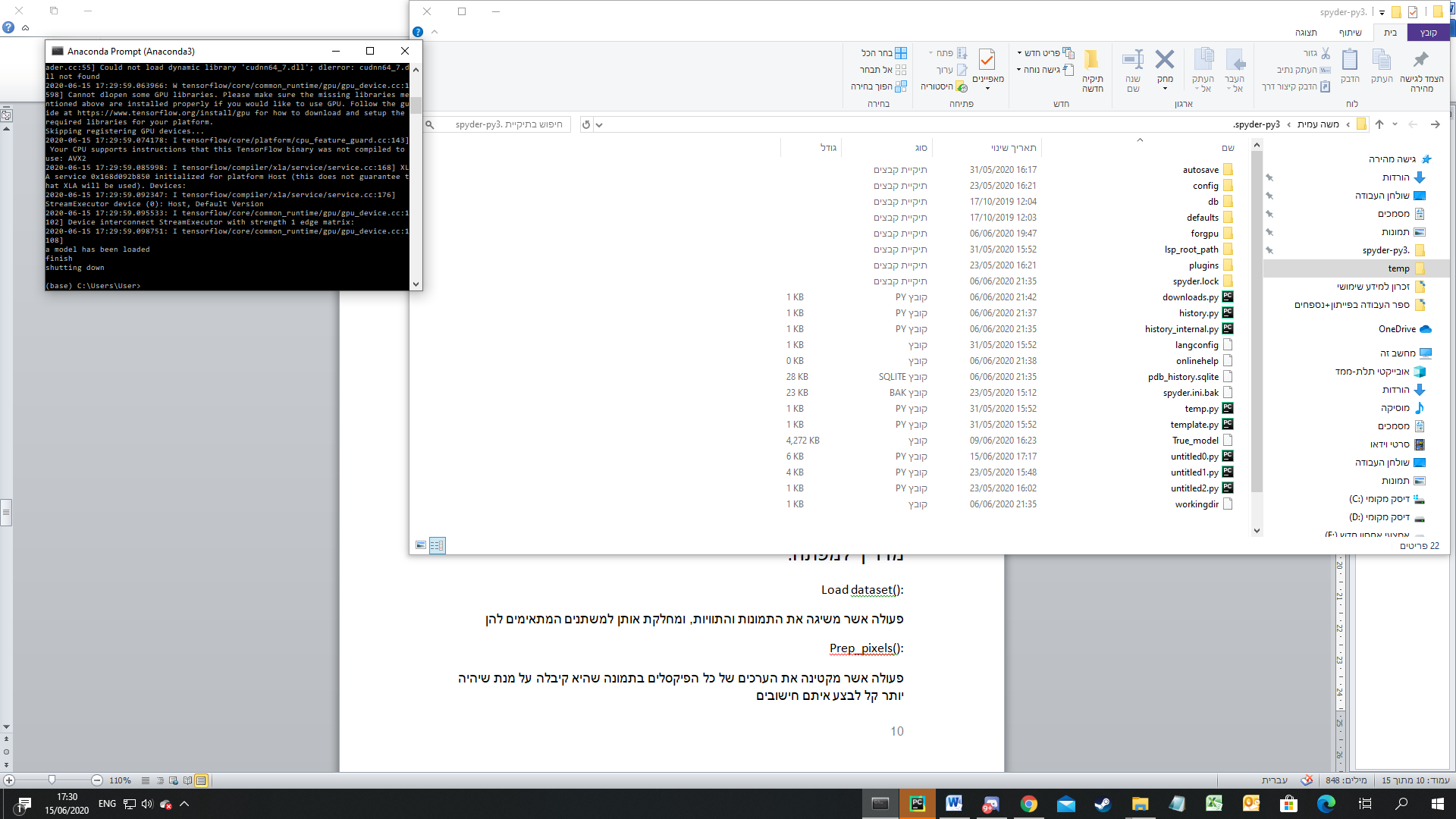
ניתן כעת לבחון את המודל באמצעות הפעולה test:



ניתן גם כמובן להתחיל ללמד מודל מחדש באמצעות פעולת train:



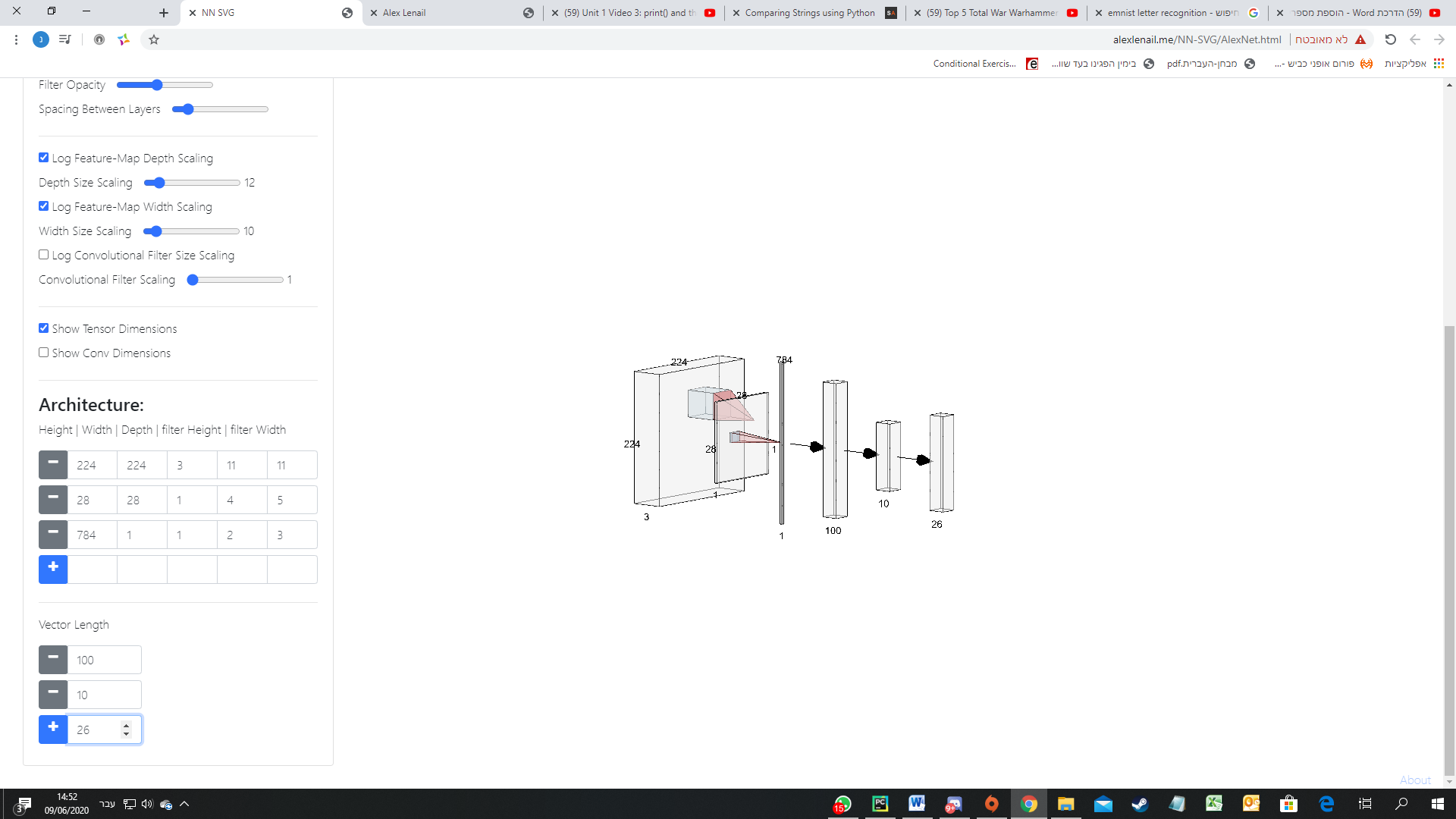
וכמובן שניתן לסיים על ידי פעולת finish:



מדריך למפתח:

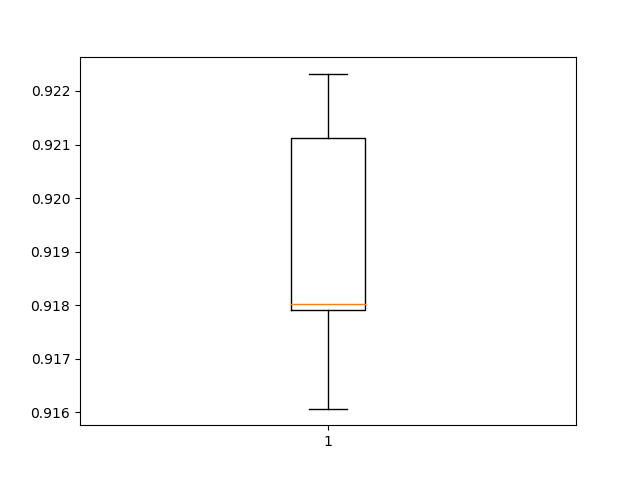
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מזמנת את: | מזומנת על ידי | איך היא עושה | מה היא עושה | שם הפעולה |
|  | Run test harness() | משתמשת במודל ובdataset emnist על מנת להוריד את התמונות והתוויות מהאינטרנט | מחזירה ארבעה מערכים אשר משני סוגים: תוויות ותמונות. שניים לבדיקה ושניים לאימון | Load dataset() |
|  | Run test harness() | מחלקת את המערכים שהתקבלו מ load dataset() ב 255.0 | פעולה אשר מחזירה מערך של מספרים המייצג את התמונות שהמודל אמור לעבוד איתן, לאחר שהן עברו הפשטה(מלשון פשטות) | Prep pixels() |
| Define model()  Summarize diagnostics()  Summarize performance() | Run test harness() | באמצעות define model(), summarize performance, summarize diagnostics וכן פקודות של keras ו sklearn | הפעולה אשר יוצרת את המודל, מאמנת אותו ומחזירה את הסטטיסטיקה שלו | Evaluate model() |
|  | Evaluate model() | באמצעות פקודות של keras | הפעולה אשר מגדירה את המודל עצמו | Define model() |
|  | Run test harness() | פקודות של matplot lib | פעולה אשר מציגה את הנתונים הסופיים של המודל | Summarize diagnostics() |
|  | Run test harness() | פקודות של matplot lib | פעולה אשר מציגה את הערכים הסטטיסטיים של המודל (סטיית תקן, ממוצע וכו') | Summarize performence() |
| Load dataset()  Prep pixels()  Evaluate model()  Summarize diagnostics()  Summarize performance() |  |  | הפעולה אשר מאחדת את כל הפעולות הנ"ל על מנת לאמן מודל | Run test harness() |
|  |  | קוראת את התמונה בשחור לבן, משנה את המימדים שלה למימדים הרצויים, ומחלקת את כל הערכים של הפיקסלים של התמונה ב 255.0 | מקבלת תמונה כלשהי, מחזירה מערך שהמודל יכול לעבוד איתו | Image\_norm() |
|  |  | מוצאת את המקום עם המספר הכי גדול, מחזירה את המקום הזה במערך אותיות נתון | מקבלת מערך אשר מהווה את התשובה של המודל, מדפיסה את האות שהמודל חושב שנתנו לו לבחון | Make to letter() |

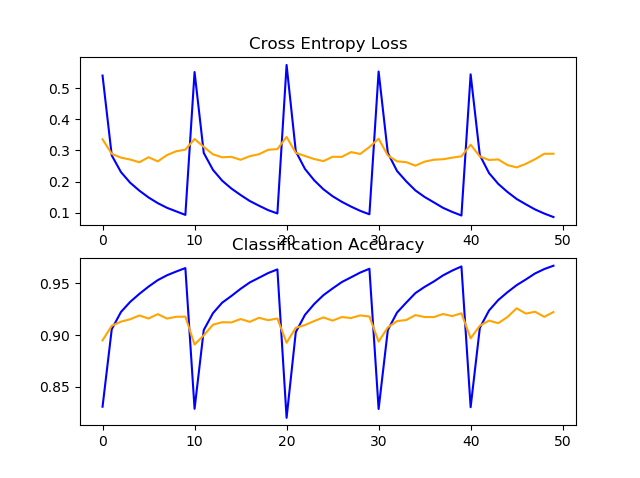
תרשים הזרימה של כל התהליך נראה בערך ככה:



פרק 4: מסקנות מהרצת המודל

המודל פועל ביעילות של 91 אחוזים, ובעל סטיית תקן 0.5 אחוזים של כפי שניתן לראות בגרף:





5: רפלקציה אישית

הפרויקט היה לא פשוט, נאלצתי לחפור בעשרות תשובות של stack overflow במטרה למצוא תשובה לשאלה אחת, והיו לי הרבה שאלות. בסופו של דבר הצלחתי להגיע למודל ואני גאה בתוצאה אליה הגעתי.

6: ביבליוגרפיה

Emnist dataset

נספחים:

