שולח שוב גם מה-gmail שלי לוודא שאם לא הועבר כראוי למישהו באמצעות המייל האלביטאי שלי.

בברכה

שלום רב,

בהמשך למפגש הראשון שנערך בתאריך 5/7/2022 אני מביא כאן מידע משלים הנחוץ להשלמת התכנים עליהם שוחחנו היום.

**חשוב לי להתחיל עם:**

**1.**       **לגבי משוב:**

o        לי הייתה חוויה ראשונית סופר חיובית וזה בזכות כולנו יחד.   
האווירה, ההקשבה, השיחות, השאלות, בעיני היה כפי שתכננתי וקיוויתי.

o        תודה על המשוב שהעברתם בסוף המפגש

o        תודה על משובים שנשלחו אליי בפרטי

o        אנא המשיכו עם זה בבקשה

o        אני לוקח כל עצה והארה במלוא תשומת הלב ובשיא הרצינות כדי לשמר מה שטוב ולשפר מה שצריך

**o**        **לגבי נקודות משוב עיקריות שהועלו:**

§         תחילה אזכיר שאני מגיע מהשטח עם ניסיון בתחום DL בפרט תשתיות NVIDIA עליהם מתבסס חלק זה של הקורס של למעלה מ-4 שנים.

§         אני מזכיר זאת כי חשוב לי שתדעו שהמטרה שלי היא לחשוף אתכם לעשייה הטיפוסית המאפיינת את שלב ה-Deployment ולא להתמקד ובמיוחד לא לעצור ולהציג רק את ה-Best case כי אז תישארו עם פער עצום בין הרצוי למצוי.

§         באוטופיה שלי אני מזכיר/ מלמד / מדגים ביחד אתכם את כל האתגרים המכשולים ודרכי התמודדות

§         לכן אם ניתן כדוגמא את הסוגייה האם לקבל מדריך מראש מלא לשלב ה-Setup לעומת לעשות זאת עצמאית, כבר אתמול ראיתם שכמה חברים דרכו בעצמם על תקלות טיפוסיות וזו בדיוק הייתה המטרה שלי ובעצם שלב זה היה חלק מתרגול מעשי שתכננתי שתבצעו.

§         ברור לחלוטין שכרגע אתם בשלב שלא מאפשר לכם לראות את כל התמונה אבל כפי שהתחייבתי במהלך המפגש, בדיעבד אתם תראו שכל מה שתבצעו עצמאית מהנחיות שלי מגיע ממוטיבציה מציאותית

§         לגבי הרחבה על נושא כזה או אחר, חשוב לזכור שהתפלגות האנשים מייצגת ממצב של ללא ניסיון כלל עד למצב של אנשים שכבר עובדים עם CPU וטכנולוגיות DL.  
מה שעומד מול עיני זה תמיד למצוא את האיזון.

§         לגבי תשתיות, נעשה מאמץ רב מצידי ומצד מנהלי הארגון לתת מענה שעלה מטעם סטודנטים ממחזורים קודמים, הכל עניין של תקציבים וגם כאן התחייבתי במפגש שסדר גודל של למעלה מ-90% מהעשייה הפרקטית תיתמך על ידי  
GPUs cloud ותהיה מPython Based.

§         אשתדל יותר למנוע התבדרויות בהעברת הידע שלי ומה שנקרא "לפתוח סוגריים", פשוט, בכנות, כאשר אני מעבר מסר כלשהו לעיתים אני נסכר בתת מסר סופר חשוב, וחשוב לי להעביר לכם אותו גם, אני תמיד משתדל "לקפל" את הסוגריים חזרה בצורה הכרונולוגית הרציפה כדי לשמור על שלמות המידע וביצוע ההקשרים בין המסרים.

§         אני מבטיח ברמה האישית שלי שהקשר המקצועי ביננו (עם מי שירצה בכך כמובן) לא מסתיים ב-4 המפגשים במסגרת הקורס ואשמח להיות פה לתמיכה עבור מי שירצה בכך גם בעתיד.

**2.**      **שיעורי בית:**

a.       השלמת התקנות NVIDIA SDKs

b.       הרצה של ה-TensorRT Sample שנקרא onnx\_resnet50.py.

                                                   i.      את הקובץ קוד עצמו ניתן למצוא תחת:

<TRT extracted zip directory>\samples\python\introductory\_parser\_samples\

                                                 ii.      Data שקוד דוגמא זו מחפש נמצא תחת:

<TRT extracted zip directory>\samples\python\introductory\_parser\_samples\data\resnet50\

                                               iii.      Definition Of Done (DOD) של תרגיל זה הוא הדפסה של ההודעה:

"Correctly recognized…."

c.       בונוס: yolov3\_onnx

                                                   i.      הרצה של דוגמא זו בהצלחה

1.       תחילה מריצים את הקובץ yolov3\_to\_onnx.py

2.       לאחר מכן את הקובץ onnx\_to\_tensorrt.py

3.       יתרון - למי שיעשה זאת step by step על מנת להתחיל לחוש את האובייקטים של TRT ושל ה-Onnx ומה הם ה-APIs בהם משתמשים להשלמת כל המסלול.

4.       הערה – בדוגמא זו נדרש להוריד חומר נוסף מהרשת מעבר למה שמגיע כחלק מה-TRT SDK כפי שהיה בדוגמא onnx\_resnet50.py, כל הפרטים מופיעים בקובץ download.yml שנמצא באותה תיקייה של הדוגמא

d.       אשמח שתשתפו טרם המפגש הבא עם תוצאות, תובנות, קשיים, אתגרים, דרכי התמודדות, נקודות פתוחות בכל דרך שתמצאו לנכון לעשות זאת.

e.       כמובן שנעבור על הכל יחד בתחילת המפגש הבא בשאיפה לסגור לכולם את הנקודות הפתוחות.

f.        מאוד אשמח שהמעבר יכלול דוגמא של מישהו שישלים מה שיתאפשר על תשית GPUS cloud ולאו דווקא רק על תשתית לוקאלית

**3.**      **השלמת מידע:  
אם חסר משהו אשמח שתסבו את תשומת ליבי**

g.       התקנות:

                                                   i.      CUDA Tlk:

1.       <https://developer.nvidia.com/cuda-downloads>

2.       יש לעקוב אחר ההנחיות להורדה, להתקדם צעד אחר צעד ולבחור את מה שמתאים לפלטפורמה שלכם.  
זה תלוי במערכת ההפעלה, בגרסה שלה ואם אתם מעדיפים הורדה של כל ההתקנה למחשב לוקאלית ואז להריץ את ההתקנה או לעשות התקנה Online.  
  
אני ממליץ על הורדה של ההתקנה במלואה אליכם לוקאלית ולהריץ אותה.  
זה מאפשר שחזור של הפעולה באופן קל וזמין יותר.  
דוגמא לקובץ: cuda\_11.7.0\_516.01\_windows.exe.

3.       לבחור בהתקנה תמיד את האופציה שמאפשרת לכם לשלוט באילו מרכיבים של החבילה אתם תתקינו.  
למשל:

                                                 ii.      CUDNN:

1.       <https://developer.nvidia.com/rdp/cudnn-download>

                                               iii.      TensorRT (TRT):

1.       <https://developer.nvidia.com/nvidia-tensorrt-download>

2.       Samples guidance:

<https://docs.nvidia.com/deeplearning/tensorrt/sample-support-guide/index.html>

                                               iv.      חשוב מאוד לזכור שאתם בוחרים צירוף שמתאים בין כל המרכיבים הבאים:

1.       GPU device

2.       Display driver:

a.       ניתן להתקנה עצמאית מכאן: <https://www.nvidia.com/download/index.aspx>

b.       או כחלק מה-CUDA Tlk

3.       CUDA Tlk

4.       CuDNN

5.       TRT

     לינק שניתן להיעזר בו להבין את התלויות ניתן למצוא כאן:

<https://docs.nvidia.com/deeplearning/tensorrt/support-matrix/index.html>

h.      PyTorch \*.whl להתקנות:

<https://download.pytorch.org/whl/torch/>

i.         Tensorflow GPU support matrix:

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

j.          Netron:

<https://github.com/lutzroeder/netron>

k.       Remote Development - Visual Studios C++ for Linux Development:

                                                   i.      הצגתי לכם את הטכנולוגיה באמצעותה אנו בתחום מפתחים לסביבת Linux

                                                 ii.      השיטה אומרת לחבר מחשב Host עליו מותקן Windows ובאמצעות VS + Cross platform Linux Extension להתחבר למחשב ה-Remote Linux

                                               iii.      כך בעצם חווית הפיתוח למשתמש נשארת זהה כאילו הוא מפתח לוקאלית למכונת ה-Windows ונהנה מכל היתרונות שסביבת VS מספקת משלב הקידוד, בנייה, הרצה ו-Debug והכל בעצם קורה על ה-Linux עצמו.

                                               iv.      מצורף מדריך Remote Development - Visual Studios C++ for Linux Development להשמשת היכולת ב-VS.

l.         OpenCV:

                                  i.    Free [Getting Started with OpenCV](https://click.convertkit-mail.com/27uzrvol3lcohpez93u3/08hwhgum7n9264sl/aHR0cHM6Ly95b3V0dS5iZS9QNFo4X3FlMkN1MA==) video tutorial for Python users.

                                 ii.    **Blog series** that covers the material for beginners in both **C++ and Python**[OpenCV For Beginners](https://click.convertkit-mail.com/27uzrvol3lcohpez93u3/8ghqh3ug5w2o64ik/aHR0cHM6Ly9sZWFybm9wZW5jdi5jb20vZ2V0dGluZy1zdGFydGVkLXdpdGgtb3BlbmN2Lw==)

                                iii.    NewsLetter - [LearnOpenCV](https://learnopencv.com/" \t "_blank)

                                                iv.      התקנה ל-Windows – ראו קובץ מצורף How to compile your own version.zip

                                                 v.      התקנה ל-Linux – ראו קובץ מצורף OpenCVX.Y.Z

m.     Boost C++:

                                                   i.      ספרייה המשלימה יכולות ל-C++ שהן עדיין לא חלק מה-std

                                                 ii.      [Boost C++ Libraries](https://www.boost.org/)

                                               iii.      [Boost C++ Libraries - Browse /boost-binaries at SourceForge.net](https://sourceforge.net/projects/boost/files/boost-binaries/) ל- Windows.

                                               iv.      Linux – ראו מדריך מצורף installBoost.txt

בברכה