

ראיה ממוחשבת 0510.6251

סמסטר א' תשע"ט

נהלי הפרויקט

1. הפרויקט הינו בזוגות. מועד ההגשה הינו ב - 5.1.19
2. הפרויקט עוסק בגילוי וסיווג אוטובוסים. בעזרת תמונות האימון שאותם ניתן להוריד דרך המודל, תצטרכו לפתח אלגוריתם שמגלה ומסווג אוטובוסים. בסוף הסמסטר, תינתן לכל זוג **רבע שעה** על מנת להסביר בקצרה איך האלגוריתם עובד ולבדוק איך הוא מגלה תמונות של אוטובוסים שהאלגוריתם לא ראה מעולם.
3. הציונים יקבעו על בסיס תחרות ביצועים. הציון הסופי של הפרויקט יכלול ציון על ביצועי הגילוי (פירוט בסעיף 10) במשקל של 75% וציון על ביצועי זמן במשקל של 25%. כאמור, הציון הסופי של הפרויקט יקבע על ידי תחרות בין כל הפרויקטים.
4. את הפרויקט יש לכתוב בשפת Python. (גרסה 3.6)
5. מומלץ להתקין anaconda (גרסה 3.7,5.3.0 Python), ניתן להתקין דרך [האתר הרשמי](#). ניתן ליצור סביבה וירטואלית עם Python 3.6 ע"י הרצת הפקודה הבאה ב - anaconda prompt:

`conda create -n <environment name> python=3.6`
6. מומלץ להכין את הפרויקט בעזרת כלים של למידה עמוקה, אך זה לא חובה. במידה ואכן נעשה שימוש ברשתות נוירונים, מומלץ להשתמש באחת מ-2 החבילות הבאות:
 - TensorFlow-gpu (version 1.9.0)
 - PyTorch (version 0.4.1)) <https://pytorch.org>)

בנוסף ניתן לעשות שימוש ב:

- openCV (version 3.4.3)

מומלץ להשתמש בגרסאות הנ"ל על מנת שתהיה התאמה עם הגרסאות במחשב הבדיקה ושלא תהיינה בעיות בלתי צפויות.

קובץ עם כל החבילות המותקנות במחשב הבדיקה נמצא במודל תחת השם requirements.txt. לאחר התקנת אנקונדה ניתן להוריד ולהתקין את החבילות שבקובץ בצורה אוטומטית ע"י הפקודה:

```
conda create --name <environment name> --file requirements.txt
```

במידה ותרצו להשתמש בחבילות נוספות **אנא ידעו אותי** על מנת שניתן יהיה להתכונן לכך מראש.

7. הבדיקה תתבצע במחשב Windows עם GPU ולכן מומלץ להשתמש ב-GPU (הן בשלב האימון והן בשלב הבדיקה). ללא GPU ביצועי הזמן עלולים להיפגע. קיימות מספר אופציות חניניות לשימוש כגון: AWS (need to request), google Cloud (free 12 months trial), google Collab, ופלטפורמות נוספות.

8. הבעיה: למצוא איפה נמצא כל אוטובוס בתמונה ובנוסף לזהות באיזה אוטובוס מדובר (יש סה"כ 6 אוטובוסים שונים). הגילוי של האוטובוס יעשה על ידי המרובע החוסם המינימאלי שמכיל את כל האוטובוס (דוגמאות בהמשך). המרובע יוגדר על ידי 4 פרמטרים, xmin, ymin, width, height. כאשר xmin, ymin הם הקואורדינטות של הפינה השמאלית העליונה של המרובע, width הוא הרוחב ו-height הוא הגובה של המרובע. בנוסף, מספר בין 1 ל-6 יגדיר את סוג האוטובוס, כאשר {ירוק=1, צהוב=2, לבן=3, אפור=4, כחול=5, אדום=6}. לסיכום, לכל גילוי יש 5 פרמטרים, ארבעת הפרמטרים הראשונים מגדירים את המיקום של המרובע (גילוי) והפרמטר החמישי מגדיר את סוג האוטובוס (סיווג). המוצא של האלגוריתם צריך להיות קובץ טקסט כך שכל שורה מגדירה תמונה אחת כאשר הפרמטרים מופרדים בפסיקים.

```
PIC.JPG:[xmin1, ymin1, width1, height1,color1],...[xminN, yminN, widthN, heightN  
[,colorN
```

לדוגמה:

```
DSCF1013.JPG:[1217,1690,489,201,1],[1774,1619,475,224,2]
```

```
DSCF1015.JPG:[641,1342,1181,892,3]
```

DSCF1016.JPG:[1067,1843,1114,613,4],[1954,1278,1021,561,6]

בדוגמה הנ"ל יש שלוש תמונות, בתמונות DSCF1016 , DSCF1013 זהו 2 אוטובוסים ובתמונה DSCF1015 זהו אוטובוס אחד.

9. קובץ בפורמט הנ"ל בשם annotationsTrain.txt עם תיוג האמת נמצא במודל לשימושכם.

10. הגדרה של גילוי נכון:

גילוי אמת יוגדר על ידי יחס חפיפה IOU (intersection over union) בין המרובע החוסם של הגילוי לבין המרובע החוסם של תיוג האמת. יחס החפיפה מוגדר על ידי:

A – detection bounding box

B – ground truth box

$$IOU = \frac{A \cap B}{A \cup B}$$

גילוי ייחשב כגילוי נכון בתנאי ש-2 התנאים הבאים מתקיימים:

- $IOU > 0.7$

- הסוג של האוטובוס הוא הסוג הנכון (הצבע).

לבסוף מחושב מדד [F1Score](#), זהו המדד היחיד הממשקל את כל תוצאות הגילוי מכל התמונות, זהו המדד שבעזרתו יבחן טיב הפרויקט ביחס לפרויקטים אחרים. (יש גם משקל לזמן הריצה).

11. סקריפט הגילוי שנדרש להכין:

עליכם ליצר סקריפט בשם runMe.py, הסקריפט צריך לכלול פונקציה בשם run שתקבל כקלט את שם קובץ התיוגים שהסקריפט ייצר ואת התיקיה שבה נמצאים כל התמונות שאותן צריך לתייג.

כלומר ההגדרה של הפונקציה צריכה להיות מן הצורה:

```
def run(estimatedAnnFileName, busDir):
```

חשוב להשתמש בשמות הנ"ל על מנת שסקריפט הבדיקה (מוסבר בהמשך) יעבוד ללא תקלות.

12. במודל, בתוך הקובץ, python_scripts.rar נמצאים 3 קבצי פייתון, Main.py, runMe.py, busProjectTest.py.

runMe – זהו קובץ שמייצר קובץ תיוגים על ידי הרעשה של הקובץ המקורי. זהו הקובץ שתצטרכו להחליף בקובץ שלכם שיכין את תיוגי האמת. ניתן להריץ אותו על מנת לבדוק שהפייתון עובד כראוי, על מנת להריץ אותו תדרשו לשמור את קובץ תיוגי האמת בתיקיה שממנו קוראים לסקריפט.

busProjectTest – סקריפט שמשווה בין תיוגי האמת לתיוגים המשוערכים.

Main – הסקריפט המרכזי שרק אותו נדרש להריץ. זהו הסקריפט שבעזרתו יבדק הפרויקט על תמונות המבחן, לכן חשוב לבדוק את אלגוריתם הזיהוי בעזרת הסקריפט הנ"ל.

על מנת לבדוק שאכן סקריפט הבדיקה עובד כראוי ניתן להוריד את שלושת הקבצים לאותה תיקיה ולהריץ את קובץ ה-Main, קובץ זה קורא ל-2 הקבצים האחרים, עבור כל תמונה מחשב את מספר תיוגי האמת (TP) מספר הפספוסים (Missed) ומספר תיוגי השקר (FP), לאחר מכן מחשב את הציון הכללי ושומר הכל.

הרצה של ה-Main:

Main מקבל כקלט את קובץ תיוגי האמת (מה שמורידים מהמודל), את שם הקובץ של התיוגים המשוערכים (מה ש-runMe ייצר), את התיקיה שבה נמצאים כל התמונות ואת התיקיה שבה הסקריפט ישמור את כל התוצאות. לפני כל קלט נדרש לכתוב באיזה קלט מדובר כאשר:

-myAnns <estimated annotations file>

-anns <real annotations file>

-buses <directory of all the images>

-saveDir <directory for the output of the script>

את הסקריפט ניתן להריץ דרך ה-command line, דוגמה להרצה של הסקריפט:

*python Main.py -myAnns newAnns.txt -anns realAnns.txt -buses busesDir -saveDir
dirToSave*

בנוסף על מנת לקבל הסבר על כל קלט ניתן לקרוא לסקריפט עם `-h` – כלומר:

python Main.py -h

13. הגשת הפרויקט:

כל זוג צריך להגיש במועד שנקבע להגשת הפרויקט:

- דו"ח בהיקף של 2-3 עמודים המסביר את הפתרון שמומש.
- את כל הקבצים הנדרשים על מנת להריץ את אלגוריתם הגילוי (עדיף כקובץ zip).
- את כל הנ"ל יש לשלוח למייל mayamayberg@mail.tau.ac.il עם הכותרת

`cv_proj_19_ID1_ID2`

במקום ID1, ID2 יש לכתוב את ת.ז של המגיש.

14. מהלך הבחינה:

- הסבר קצר של הסטודנטים על האלגוריתם שמומש.
- הרצת האלגוריתם על תמונות המבחן וקבלת F1 Score.
- זמן הריצה מוגבל ברבע שעה! מומלץ להשתמש ב-GPU.
- מומלץ להגיע גם עם "דיסק און קי" המכיל את כל הקבצים הדרושים להרצה של האלגוריתם למקרה שיהיו בעיות כלשהן עם הקבצים שנשלחו במייל.

דוגמאות לפלט של סקריפט הבדיקה (Main.py):



