



המכללה האקדמית תל-חי
החוג למדעי המחשב
למידה עמוקה בפייתון

תרגיל 4

Face Recognition Using Pytorch

מוגש לידי: ד"ר בוריס יאזמיר

מוגש על ידי:

יובל צל-ציון | 305768871 | yuvalzelzion@gmail.com | 052-8514116

גל רייקין | 308188853 | raykingal@gmail.com | 052-4501581

סמסטר ב'

2021

תיאור האלגוריתם:

בתרגיל זה אנו מקבלים אלגוריתם לזיהוי ואיתור פנים בתמונות וסרטונים. אלגוריתם זה, שנקרא MTCNN, כבר אומן מראש ולמיטב ידיעתנו זהו המימוש המהיר ביותר שקיים כיום. בתרגיל אנו מבצעים שני שלבים. בשלב הראשון מריצים את האלגוריתם על מספר תמונות, והאלגוריתם מזהה פרצופים, ומדפיס לנו באיזו רמת דיוק או וודאות הוא זיהה פרצוף בתמונות אלו. בשלב השני, האלגוריתם מקבל כקלט סרטון, קולט בסרטון זה את כל הפריימים, ובכל פריים מבצע את פעולת ה-detect של האלגוריתם אשר מזהה את כלל הפרצופים בפריים ומצייר ריבוע אדום סביב פרצוף זה. כפלט האלגוריתם מחזיר את אותו הסרטון אך עם הריבועים האדומים מודבקים על הסרטון.

תיאור שלבים במימוש:

דוגמא ראשונה – וודאות זיהוי פרצוף בתמונה:

בשלב הראשון כמובן מייבאים ספריות נחוצות, במקרה שלנו החשובה ביותר היא MTCNN מתוך facenet.

מגדירים את המודל שבו נשתמש, וטוענים את האלגוריתם שאומן מראש.

לאחר מכן טוענים את מסד הנתונים, שקיים ב-repo. מדובר ב-5 תמונות של מפורסמים.

לבסוף מריצים את האלגוריתם על התמונות וניתן לראות כי בכולן האלגוריתם מזהה פרצופים עם וודאות של קרוב ל-100%.

דוגמא שניה – זיהוי פרצופים בסרטון:

השלבים הראשונים דומים לדוגמא הראשונה, מייבאים את הספריות הנחוצות, מגדירים את המודל וטוענים את האלגוריתם שכבר אומן.

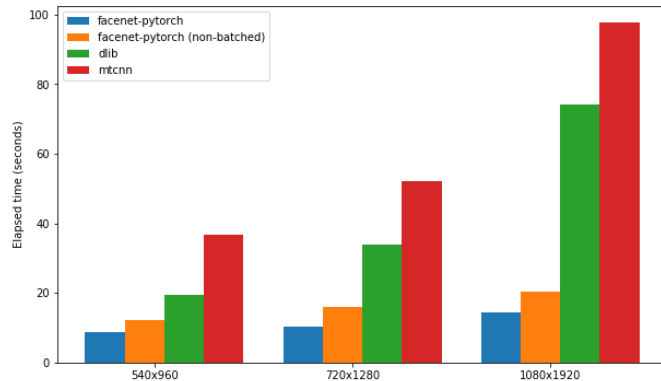
כעת נותנים לאלגוריתם כקלט סרטון, הוא מחלץ ממנו את כל הפריימים ושומר אותם במערך. לאחר מכן מעביר את כל הפריימים האלו באלגוריתם והאלגוריתם מבצע זיהוי פנים בכל אחד מהפריימים ומצייר את הריבוע.

בשלב האחרון הפריימים עם הריבועים על הפנים מתחברים לסרטון ונשמרים בשם ומיקום שנבחר.

במימוש שלנו, הרצנו את האלגוריתם על הסרטון המסופק בתרגיל, ובנוסף הרצנו על סרטון אחר שהעלנו מהדרייב שלנו, סרטון עם מספר רב יותר של פרצופים. ניתן לראות שהאלגוריתם הצליח לעבד גם את הסרטון הזה. התוצאה מצורפת בקישורים מטה.

מסקנות והצעות לשיפור:

כפי שכבר הזכרתי אלגוריתם זה מהיר מאוד, למעשה המהיר ביותר שידוע כיום, כפי שניתן לראות בתרשים:



לכן לא אציע ניסיון שיפור לביצועים.

אם בכל זאת נרצה לחשוב על תוספת, הצעה אפשרית היא הוספת זיהוי הבעה לפרצופים המזוהים. כלומר, כאשר האלגוריתם כבר מזהה פרצוף מסוים, יעבד אותו מעט יותר ויזהה גם האם האדם מחייך או זועף וכו'. עיבוד זה נקרא face alignment וקיים לו כבר מימוש ב-pytorch, לכן הוספה זו לאלגוריתם לדעתי לא אמורה להיות מסובכת מאוד.

תוצאות הרצות:

[קישור](#) ל-GitHub עם הקוד.

[קישור](#) לסרטון של תוצאות הרצה של החלק הראשון.

[קישור](#) לסרטון של תוצאות הרצה של החלק השני על סרטון שהעלנו מהדרייב שלנו.

תמונות לדוגמא :

```
aligned = []
names = []
for x, y in loader:
    x_aligned, prob = mtcnn(x, return_prob=True)
    if x_aligned is not None:
        print('Face detected with probability: {:.8f}'.format(prob))
        aligned.append(x_aligned)
        names.append(dataset.idx_to_class[y])

/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py:477: UserWarning: This DataLoader will create 4 worker processes in
cpuset_checked))
Face detected with probability: 0.999983
Face detected with probability: 0.999934
Face detected with probability: 0.999733
Face detected with probability: 0.999880
Face detected with probability: 0.999992
```

