

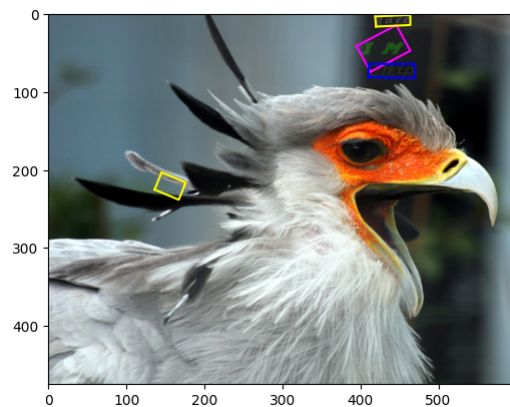
# מבוא לראייה ממוחשבת | דוח פרויקט (22928)

מגיש:דניאל וישנה | ת"ז: 208931089

13 בפברואר 2024

## 0.0.1 התחלת עבודה

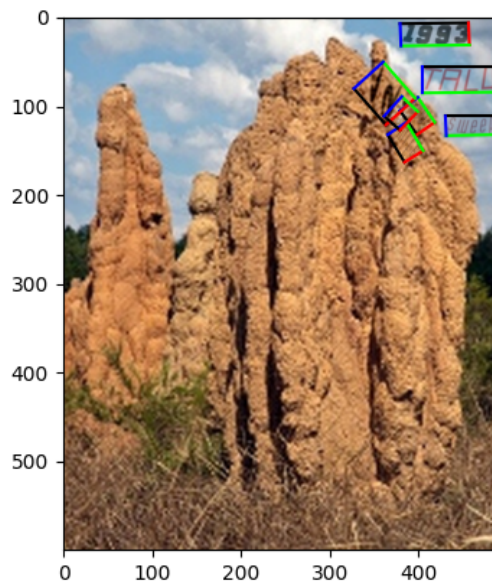
השלב הראשון שלי בקבלת הפרויקט היה לחקור ולשחק עם ה-Data Training ולהבין את מבנה הנתונים. כדי להבין את הנתונים, אחד הדברים הראשונים שעשיתי היה להציג את תמונה אחת עם כל המידע שאנחנו מקבלי עליה. הבסיס לזה נלקח מהקוד שסופק בתור דוגמא באתר של התחרות. לאחר מכן ניסית ליצור תמונות שיכילו את המסגרות של המילים קיבלתי תמונות בסגנון הזה:



איור 1:

תמונה פשוטה מסט האימון

הבחירה בלהציג קודם את המסגרות של המילים נבעה מכך שהמילים הן יותר גדולות מהאותיות והיה לי יותר קל בעין לראות מה יש בתמונה. בכך הצלחתי להבין את המבנה של charBB ו-wordBB. לדוגמא הצלחתי לראות שסדר הצלעות וכיוון הוא הכיון הטבעי שבה המסגרת צריכה להיראות.



איור 2:

כיוון הצלעות הניתנות ב-wordBB

התמקדות במילים ולא אותיות לא היתה נכונה מסתבר וגרמה לי לבעיות בהמשך. הבעיה העיקרית הייתה שנוצר בי הרושם כי על כל האותיות הופעלה רק טרנספורמצית דמיון ולא טרנספורמציה הומוגרפית.

## 0.0.2 בדיקת הצלעות

מתחילת העבודה היה לי ברור כי אני עתיד לפתור את בעיית הסיווג עם רשת. בכדי לבנות מודל הבנתי כי יש להחליט מה איך ליצג תמונה של אות. הבנתי כי נדרש ממני להציג את האותיות כמלבנים, כי כך אנו מייצגים תמונות. לכן נשאר לי להבין מה צריך להיות הגודל של כל אות כתמונה. כמובן לכל אות יכול להיות גודל שונה בין תמונה לתמונה, אך רציתי להבין מה גודל סביר ליצג תמונה של אות. אנו נדרשים לגודל סביר כיוון שצריך לתת לרשת גודל קלט קבוע. כדי להבין מה סביר בדקתי מה הגדלים של הצלעות הקיימות בדאטא ב-charBB על סט האימון. מצאתי כי :

צלע	חציון	ממוצע	מקסימום	מינימום
צלע 1	14	29.81	180	0
צלע 2	25	29.81	271	0
צלע 3	14	18.51	177	0
צלע 4	25	29.8	269	0

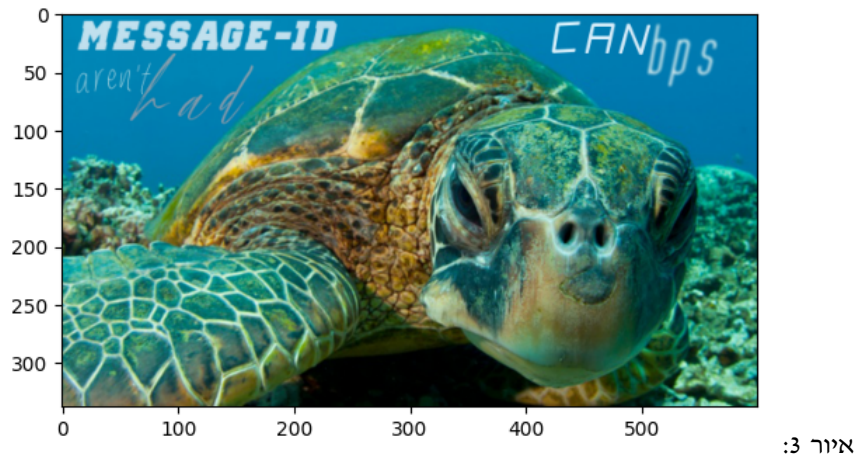
טבלה 1: יחסי הצלעות סט האימון

תוצאת בדיקה זו לא סיפקה לי תובנה מעניינת למטרה שהתחלתי איתה. לא הצלחתי לחשוב מה הגודל שלפיו כדי ליצג את האותיות במודל. כך הייתה לי מחשבה בהתחלה ליצג את האותיות ביחס של 2 : 1 כלומר תבניות שיראו טבעיות לאותיות מהסגנון של g,h,p.

למרות זאת שלב זה התברר כחשוב ביותר כי בו ראיתי שיש אותיות שגודל אחת הצלעות של המסגרת שלהם הינה 0. לנסות ליצג תמונה עם אורך צלע 0 הינו בעייתי, זאת מכיוון שאם ננסה להכניס אותה למודל נקבל תמונה שחורה. תמונה שהיא לגמרי שחורה שווה למידע זבל, אשר יכולה להפריע ללמידה של המודל. בשלב זה עוד למדתי את הדאט ולכן רק רשמתי לי בצד את הנקודה הזו.

### 0.0.3 בניית DataLoader

לאחר מכן עברתי לנסות לכתוב מחלקה שתהיה הבסיס ל-DataLoader. כאשר התחלתי לנסות להתעסק עם זה היה מישהו שהעלה לאתר התחרות קוד בסיס שמטפל בכך, והשתמשתי בקוד זה כבסיס לפתרון שלי. המחלקה של ה-DataLoader מחזירה עבור האימון את התמונה של האות, האות עצמה, הפונט ואינדקסים של התמונה והמילה עליה אנו עובדים. בסופו של דבר לא השתמשתי במידע של איזה אות התמונה מייצגת, אחת הסיבות לקח הינה שראיתי כי אין הלימה בין מה כתוב שהוא הטקסט לבין הטקסט שמוצג בתמונה. החוסר הלימה קיים בכך שהטקסט הנתון בדאטא לפעמים באותיות גדולות ולפעמים קטנות, ובתמונה יש אות גדולה או קטנה בלי שום קשר למה שמופיע בטקסט.



איור 3: הטקסט שהתקבל עבור תמונה זו הינו: 'aren't', 'can', 'bps', 'Message-ID:', 'had'.

קל לראות כי בתמונה כתוב CAN אבל הטקסט שיש בדאטא הוא בכלל can. לכן הגעתי למסקנה שאין טעם לנסות להשתמש במידע הזה כמו שהוא. כדי להשתמש במידע זה יש לתקן את הטקסט וכדי לעשות את זה צריך לבנות מודל שלא קשור למשימה שלנו.

לאחר שלב זה החלטתי לנסות לטפל בכך שהתמונות של האותיות "אין מיושרות". בתחילה עוד חשבתי שהופעל על האותיות רק טרנספורמציות דמיון, לכן ניסיתי למצוא את זווית הסיבוב ואז לחתוך את התמונה לגודל של [64, 128]. לא הספקתי לסיים מימוש זה והגעתי להרצאה על הטנספורמציות שם הוזכר שהופעל על האותיות טרנספורמציות הומוגרפיות. מסיבה זאת הלכתי לשנות את המימוש שלי בהתאם. כדי לישר את המסגרות השתמשתי בפעולות findHomography ו-warpPerspective של cv2. בשביל ליצור גודל אחיד של התמונות החלטתי לעביר את כל התמונות של האותיות לגודל של [128, 128].

### 0.0.4 בניית מודל ראשוני

השלב הבא בעבודה היה לתכנן את הרשת בה אני אשתמש בשביל הסיווג פונט. מאחר שאושר לנו להשתמש במודלים שאנחנו לא בנינו, הגעתי למסקנה שאין טעם שאנסה לממש לבד רשת למשימה. עברתי על הרשתות שקיימות בתוך ספריית pytorch לאחר קריאה בחרתי להשתמש ברשת של resnet. למודל זה בpytorch יש 3 מימושים אפשריים resnet16, resnet34, resnet50. חששתי מכך resnet50 תהיה כבדה מדי resnet16 תהיה קטנה מדי. מסיבה זו החלטתי לבנות רשת ראשונית עם resnet34. הדבר הבא החשוב להגדיר במודל הרשת הוא את פונקציית הloss, המשכתי בקריאה וחיפוש של פונקציות שונות ובחרתי בפונקציית CrossEntropyLoss. בחירה זו נעשתה מכיוון שפונקציה זו היא אחת מהפונקציות שגיא הפופולריות עבור משימות סיווג. לא רציתי להסתבך בלהמציא פונקציות המותאמות בדיוק לבעיה, מחשש שאני אבנה משהו לא נכון שרק יהרוס את התכנסות המודל. בנוסף בממ"ן אחד השתמשתי בפונקציה זו וראיתי שאני מקבל תוצאות לא רעות בסיווג.

דבר נוסף שלקחתי ממ"ן אחד הוא את הגדרת ה-optimizer להיות Adam עם Learning rate של 0.01. הגדרתי גם scheduler=StepLR שיריד את Learning rate כל 5 epochs עם ערך  $\gamma = 0.15$ . הוספתי scheduler מכיוון שאציק הציג בהרצאה שלו

את שיטה זו כאחת שיכולה לעזור לרשת ללמוד. פחדתי שזמן האימון של הרשת יהיה ארוך מדי לכן הגדרתי `batch_size=128` וריצה של 10 epochs. בהרצה הראשונה שלי הצלחתי להגיע עם הרשת לאחוז דיוק של 89%.

## 0.0.5 המשך עבודה

על ההרצה הראשונה התקבלו לי תוצאות מצויינות על סט הבדיקה, לכן הבנתי שאני רק צריך לשפר ולא לשנות יותר מדי דברים. חשבתי מה עוד אפשר להפעיל כדי להעלות את האחוזים. נזכרתי שהפונט זהה עבור כל האותיות באותה מילה. בעקבות זאת החלטתי להגדיר על הטסט כי הפונט יקבע עבור כל מילה ולא עבור כל אות. הדרך שזה יעשה היא בעזרת סיווג כל אות ובסוף ביצוע הצבעה של מה הגופן הכי מתאים למילה. במקרה של תיקו ילקח הפונט שקיבל את הערך הכי גבוה בסכום וקטורי ההסתברויות של כל האותיות של המילה. ניסית לנרמל את וקטורי ההסתברויות כדי שכל אות תתרום את אותו סדר גודל של הסתברות עבור כל פונט, אבל קיבלתי כי האחוז הדיוק רק יורד עם הנורמליזציה, לכן חזרתי לשיטה בלי הנרמול. כן בחרתי באיזה שהוא שלב לנרמל את ערכי הפיקסלים לפי  $(0.225, 0.224, 0.229)$ ,  $(0.406, 0.456, 0.485)$  שזה *Mean* וה-*Standard deviation* של ImageNet צפיתי במעט שיפור במודל עם זה לכן ולכן השארתי את הנירמול.

בשלב מסויים קראתי קצת על *ReduceLROnPlateau* שזה scheduler שמפסיק להוריד את ה-*Learning rate* אחרי שה-*loss validation* לרד  $x$  אפוקים רצופים. לא ראיתי שיפור של ה-*ReduceLROnPlateau* על פני *StepLR* ונראה על פני הוליצדיה כי *StepLR* מביא תוצאות יותר טובות. אני חושב שscheduler של *ReduceLROnPlateau* יכול לעבוד איפה שהו שנדרש הרבה אפוקים כדי שהמודל יתכנס. מה שכן השימוש הראשוני ב-*ReduceLROnPlateau* הכריח אותי להכניס כבר בשלב האימון את הריצות של המודל על ה-*set validation*, דבר זה עזר לי מאוד לראות במהלך האימונים האם המודל אכן עושה עבודה טובה בכללי.

## 0.0.6 יחס סט אימון סט ולידציה:

בחרתי בכל האימונים שלי שהיחס בין סט האימון לסט הולידציה יהיה 90% אימון 10% ולידציה. הגעתי למספרים אלו לאחר שקראתי שזה בדרך כלל היחסים שנותנים. בהתחלה חלקתי בין סט אימון וולידציה בצורה רנדומלית. בהמשך הבנתי עלו לי בעיות אי יציבות בין הרצות שונות של המודל. הסיבה העיקרית שבגללה הפסקתי לחלק את הסטים בצורה רנדומלית היא שהבנתי שבצורה רנדומלית אני לא יכול העריך את סט הולידציה כמו שאני מעריך את סט המבחן. אם יש רנדומליות אין אפשרות לעשות הצבעת הרוב על מה הפונט של המילה. לפני שקיבלתי את זה שאני אחלק את הסט אימון וולידציה לפי מיקום מוסכם של סוף הסט ותחילת הסט בדקתי שאני אכן מקבל את אותו יחס של פונטים בשני המקומות.

פונט	אימון	ולידציה
always forever	22%	24%
VertigoFLF	20%	18%
Mono Ubuntu	20%	20%
Sweet Puppy	7%	7%
Skylark	9%	10%
Wanted M54	15%	14%
Flower Rose Brush	7%	7%

טבלה 2: יחסי סט אימון סט ולידציה

כמו שניתן לראות בטבלה גם בסט האימון וגם בסט הולידציה נשמר אותו יחס בין הפונטים השונים. זה מצויין מבחינתינו מכיוון שאנו לא באמת יכולים לדעת מה ההתפלגות של הפונטים על סט המבחן. חלוקה די דומה בין סט הולידציה והאימון זה הדבר הכי טוב שאנו יכולים לבקש. לאחר שראיתי את התוצאות האלה זנחתי לגמרי את המחשבה לחלק את המידע בין הסטים בצורה רנדומלית.

## 0.0.7 עבודה על מידע לא תקין

במהלך העבודה שלי על הדאט ראיתי שהקודקודים של ה-*charBB* הם לא פעם מספרים לא שלמים. בתחילה דאגתי מזה כי אין לנו באמת חצי פיקסל. לדבר זה מצאתי פתרון ע"י זה שהחלטתי לשלל את ה-*charBB* צריך לעבור אחרי הטרינספורמציה

לנקודות  $(0, 128)$ ,  $(128, 128)$ ,  $(128, 0)$ ,  $(0, 0)$  ובכך נפתרתי מהצורך לטפל בחצי פיקסלים. הבעיה שהזכרתי לפני כן של אותיות שהתמונה שלהן יוצאת עם אורך או רוחב אפס נפתרה על ע"י זה שבאימון בדקתי שאם יש לי מסגרת שאחת הצלעות שלה היא באורך אפס נגריל אות אחרת במקומה. מתוך סט האימון שהוא בגודל 19,956 היה לנו 23 אותיות שהן עם צלע באורך 0 (כלומר פחות מאחוז אחד  $\sim 0.11\%$ ). היה חשוב לי להוריד את האותיות האלה מהאימון כי אני בעצם סתם מעניש את המודל שלי שהוא לא יודע "לנחש".

כדי להתמודד עם אותה בעיה גם בשלב הטסט הגדרתי פשוט שנדלג על החיזוי של המודל על אותה אות, ונעשה את כל הקביעות של הפונט למילה עם פחות אות. כל מה שהצגתי עד עכשיו הצליחו להביא אותי להצלחה של 94% על הטסט שהיה באתר.

דבר נוסף שעשיתי היה לסדר את הטווחים של הקודקודים של ה-charBB. זאת עקב כך שיש קודקודים שהיו בעלי שיעור שלילי או שיעור הגדול מגודל התמונה. דבר זה נראה לי כטעות לך פשוט עשיתי *clipping* לקודקודים שהיו בטווח הנכון.

## 0.0.8 שיפור המודל

לאחר שבניתי רשת ראשונית ועשיתי מניפולציות על הדאט ניסיתי לחשוב מה עוד אני יכול לעשות. חשבתי שבגלל שהמודל שלי מגיע לאחוז סיווג נכון גבוה על סט המבחן אין לי צורך לסבך יותר את המודל או לבנות מודל מקביל או נוסף. לכן נשאר לי לשחק קצת עם ההיפר פרמטרים שיש לי במודל. פרמטרים אלו כוללים את:

1. מספר *epochs*

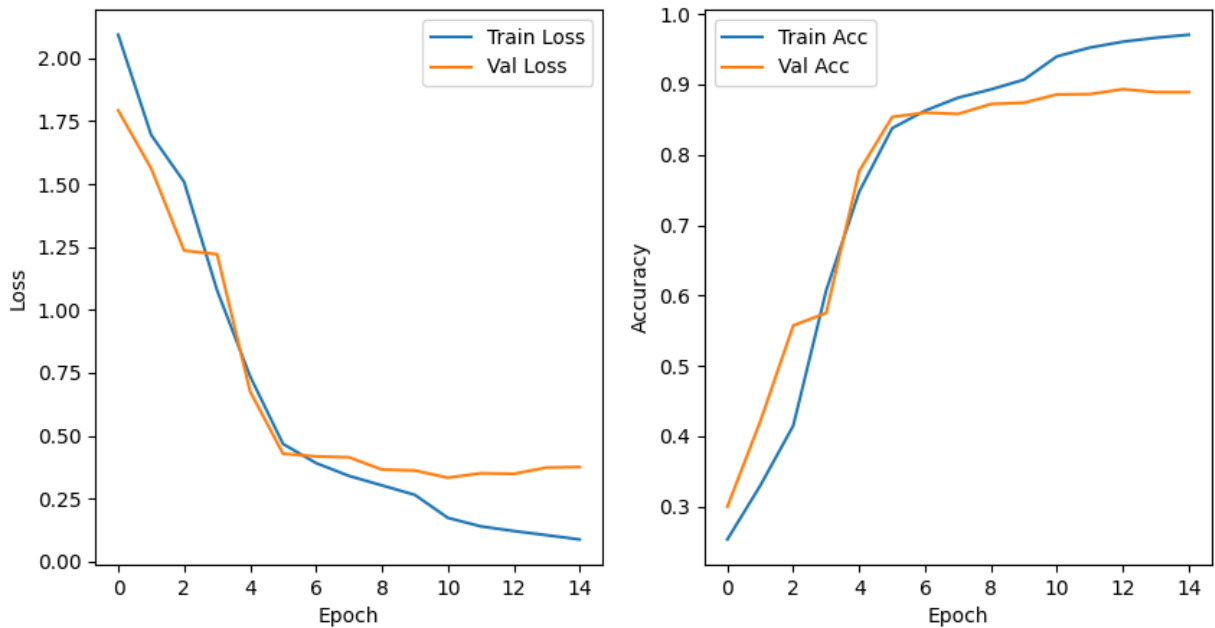
2. גודל ה-batch

3. ערך ה-rate Learning

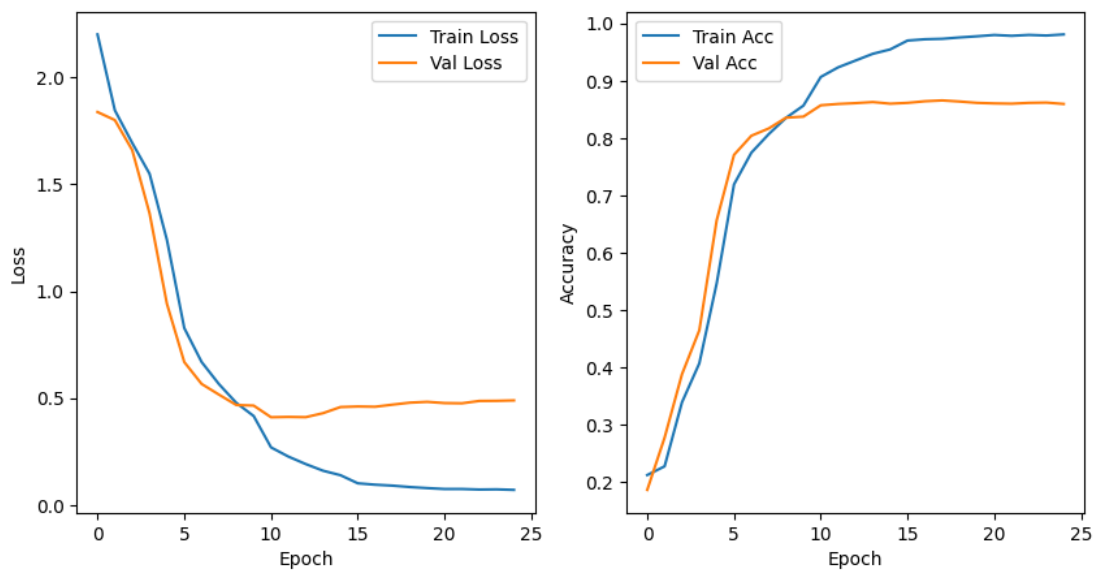
4. גודל הצעד וגודל הגמא של ה-*scheduler* (מסוג lr\_scheduler.StepLR)

בהתחלה הגדרתי את מספר ה-epochs להיות 10 אבל אז ראיתי שהמודל עוד לא סיים להתכנס, לאחר מספר הרצות שונות שמתי לב שלאחר בערך 12 ריצות המודל מגיע לדיוק הרב ביותר על סט הוולידציה. לכן הגדרתי את  $epochs = 15$  ושמירה של המודל מהאפוק עם אחוז הדיוק הגבוה ביותר על סט הולידציה.

ראיתי גם שהמודל לא רץ לאט כמו שחשבתי ולכן החלטתי להחליף את הרשת של המודל מ-resnet34 ל-resnet50. בנוסף החלטתי להוריד את גודל ה-batch מ-128 ל-32 דבר זה גם סייע לי לשפר את תוצאות המודל. עם ערך ה-rate Learning וערכי ה-*scheduler* הייתי צריך לשחק לא מעט כדי להצליח להתכנס בסוף אבל לא להתכנס יותר מדי מהר, לא רציתי להתכנס מהר מדי מכיוון שאני רוצה שהמודל שלי ילמד את המרחב באמת ולא ילמד מהדגימות בלבד. לבסוף החלטתי להגדיר את ה-rate Learning להיות 0.025 ואת ה- $step\_size=5$ ,  $gamma=0.15$ . כל הבחירות האלו נעשו ע"י משחק ולראות מה עובד יותר ומה פחות.



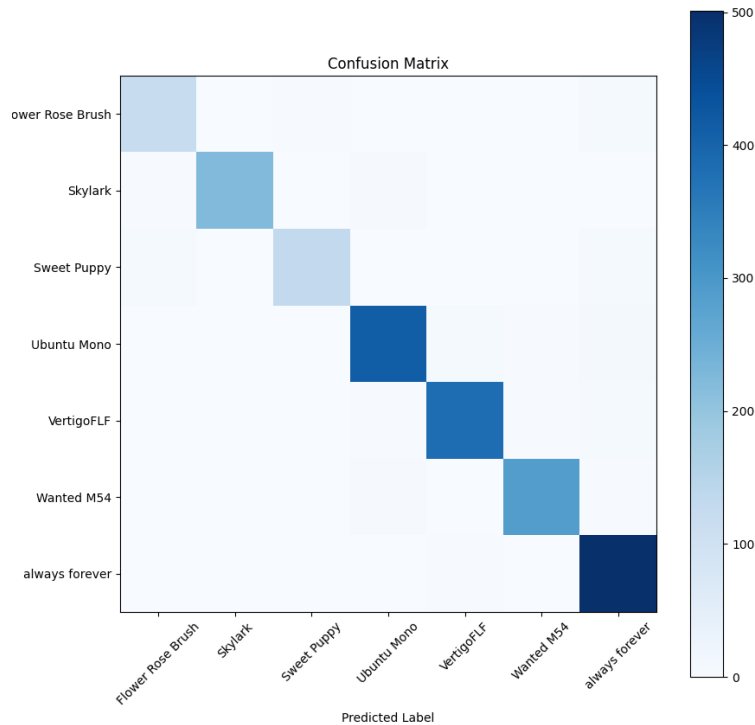
מהגרף ניתן לראות כי המודל ממשיך ללמוד לאורך כל שלבי האימון על סט האימון, אבל סט הוולידציה מפסיק להשתפר החל מהאפוק ה-12. לכן היה לי חשוב לקחת את המודל לא בסוף האימון אלה מתי שהגענו לתוצאה הכי טובה על סט הוולידציה, כל זה נעשה על מנת למנוע מצב של overfitting של המודל. ביצעתי גם בדיקה עם הרצה של 25 אפוקים וניתן לראות שאני מקבל את אותם תוצאות פחות או יותר.



איור 5:

### 0.0.9 מטריצת ה-Confusion

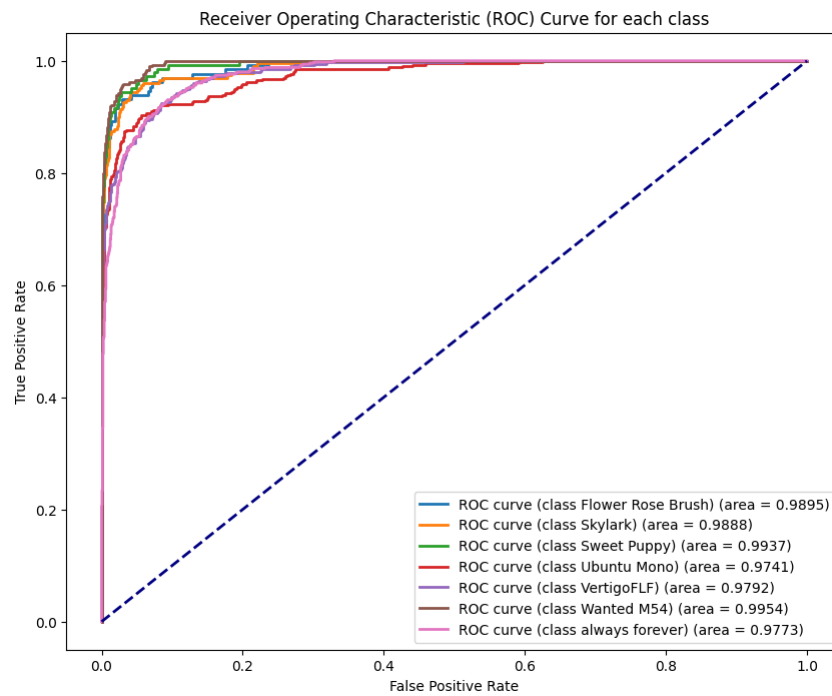
כדי לראות אם אין בעיות נקודתיות של סיווג מחלקה מסויימת במודל החלטתי לבדוק מה ה-Matrix Confusion של המודל על הולידשאין סט. כמו שניתן לראות אין משהו נקודתי שהמודל לא מצליח לזהות ונראה שקיבלנו ממש את מטריצת היחידה.



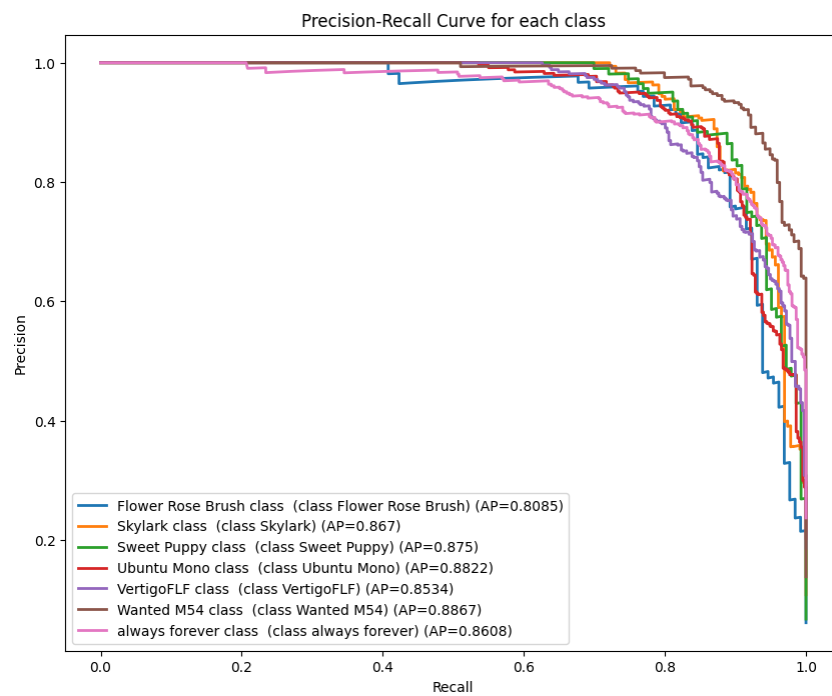
איור 6:

### 0.0.10 תוצאות הריצה על שלב האימון:

בסופו של דבר המודל שלי הצליח להגיע לאחוז דיוק של 97% על סט המבחן הגלוי. כדי לבדוק שביצועי המודל שלי אכן מוצלחים החלטתי להוציא את הגרפים כמו ממ"ן 1. בסך הכל יצאו לי הגרפים תומכים בכך שהמודל מצליח לסווג את הגופנים.

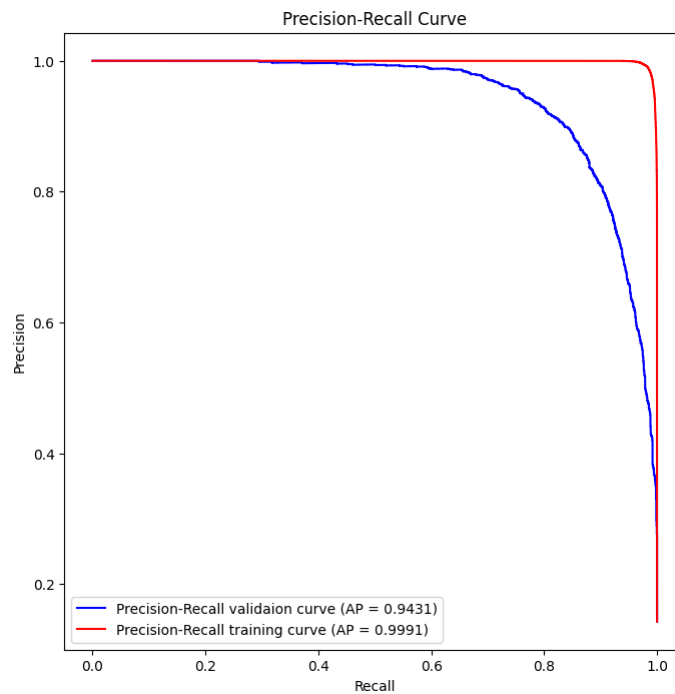


איור 7:



איור 8:





איור 9:

#### 0.0.11 הוראות להפעלת המודל:

כדי להריץ את המודל על סט המבחן יש לבצע את השלבים הבאים:

1. נדרש להוריד את הקבצים המצורפים בתקיה עם הדוח.
2. בשביל הנוחות אפשר ליצור סביבה וירטואלית עם החבילות הנדרשות המופיעות בקובץ requirements.txt.
3. להוריד את המודל מהתיקיים דרייב מהקישור :
4. <https://drive.google.com/file/d/1W1u7kIQsdTf960ty2r0dK8rX3v7iZLbm/view?usp=sharing>
5. כדי ליצור את הקובץ csv כמו שנדרשנו ליצור יש בכללי להריץ רק את הקובץ eval\_model.py.
6. אם הקובץ סט המבחן לא נמצא בתקיה של הפרויקט נדרש לשנות את המשתנה test\_file\_loc בקובץ eval\_model.py-במיקום ושם הקובץ.
7. כל שאר הקבצים מצורפים בשביל לראות את כל תהליך העבודה.