

הקוד שלי מניח שלתיקייה יש השם המקורי

spatial_envelope_256x256_static_8outdoorcategories

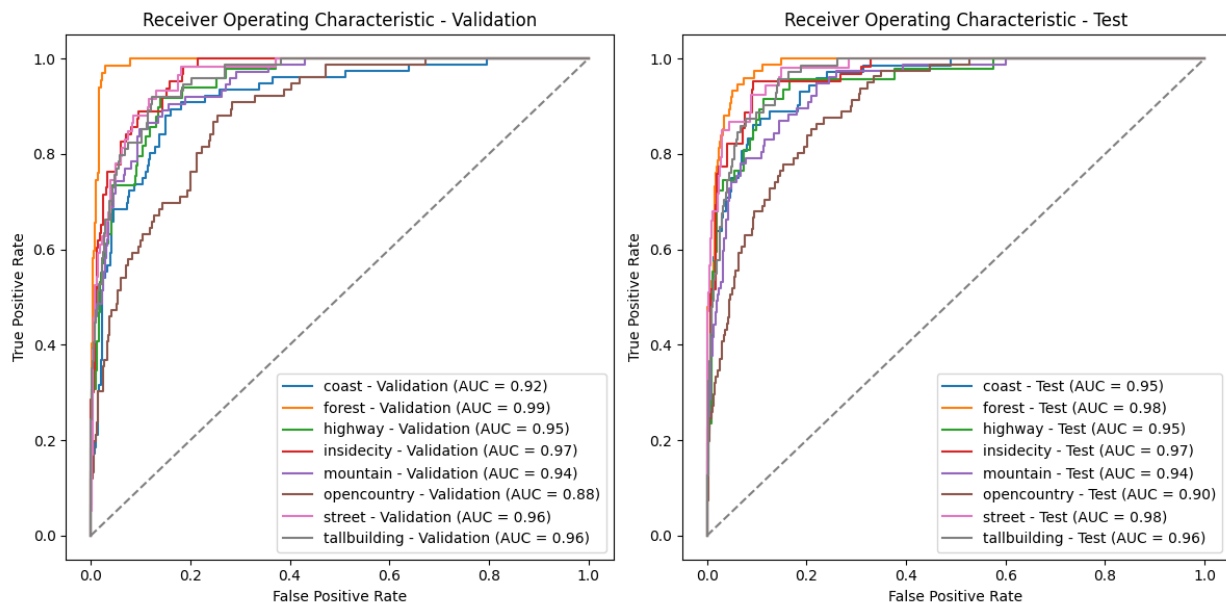
דו"ח ממ 12 מבוא לראייה ממוחשבת גוני דיו הנס 215615519

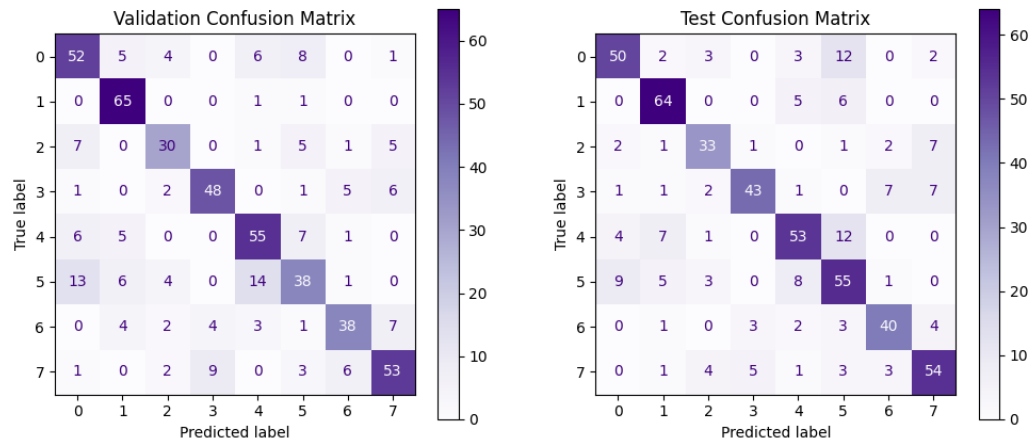
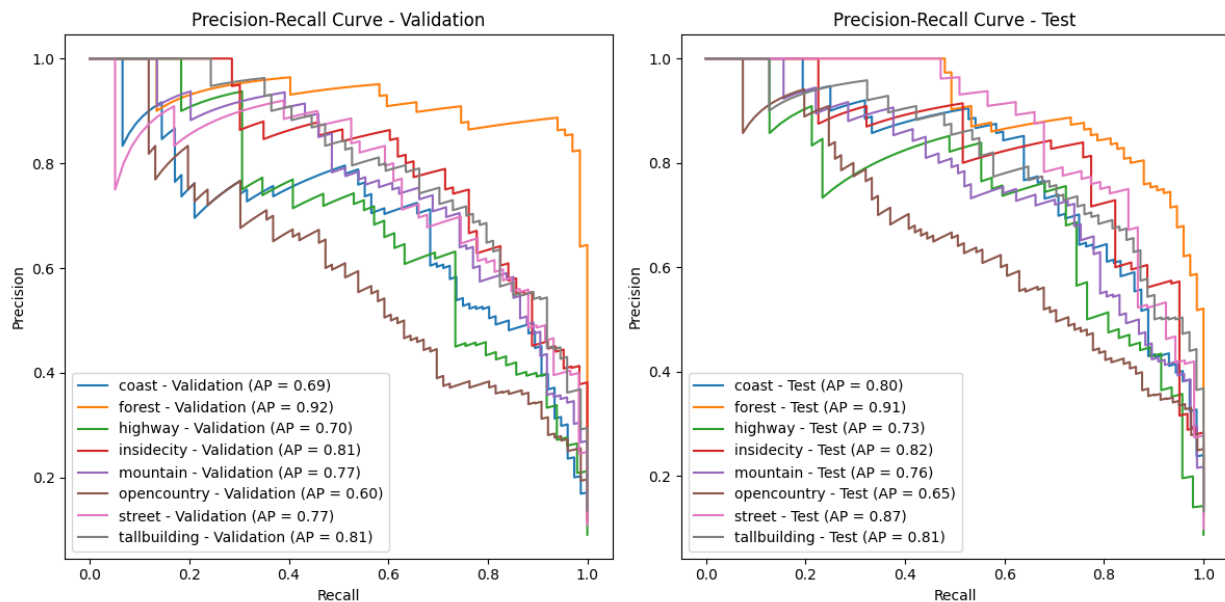
לאחר לילות לבנים רבים של עבודה אני גאה להציג את הממן!

שאלה 1

בקשר למציאת מאפיינים עם סיפט: לאחר ניסוי וטעייה מצאתי נכון להגדיר שישמרו 500 המאפיינים הכי משמעותיים וכן הגדרתי contrast threshold על מנת שנקודות עם קונטרסט נמוך ממנו לא יבחרו (קונטרסט נמוך מצביע על רעש) וכן הגדרתי edge threshold על מנת לשמור על יותר edges מאוזנים כגון פינות. שכן נקודות עניין של edge ישרים וארוכים אינן כה אינפורמטיביות ומעניינות שכן הן דומות כולן. מצאתי נכון להשתמש בpca על מנת להוריד כל גודל של descriptor בחצי מ128 ל64 וכן מצאתי נכון להשתמש ב200 קלאסטרים עבור kmeans. בתהליך בחירת הk התחשבתי בכך שא גבוה מעלה את הדיוק אך k גבוה מדי מוריד דיוק ולידציה כלומר יש התאמת יתר. לכן צריך לשים לב ל trade off הזה ולמצוא את האיזון ובכך את הk הנכון לאחר ניסוי וטעייה. כמו כן ביצעתי נורמליזציה ל2 על ההיסטוגרמות בשביל שיהיה magnitude עקבי להן שכן כל ההיסטוגרמה בעצם מומרת לוקטור יחידה. כך גם מוודאים שההשוואה של המודל בין ההיסטוגרמות תהא מבוססת על ההתפלגות היחסית של המאפיינים. על מנת למצוא את הפרמטרים האופטימליים למסווג svm עשיתי חיפוש גרידי על טווחי ערכים שונים עבור c (פרמטר עוצמת הרגולרציה), gamma (כמות ההשפעה של כל ההיסטוגרמה), והשתמשתי בגרעין rbf. להלן הפרמטרים האופטימליים של המסווג וביצועים (הצגתי בנוסף לבדיקה גם על הולידציה סתם בתור בונוס, יודע שלא הכרחי):

```
Best SVM Parameters: {'C': 1, 'gamma': 1, 'kernel': 'rbf'}
Validation AUC (One-vs-Rest): 0.95
Test AUC (One-vs-Rest): 0.95
```

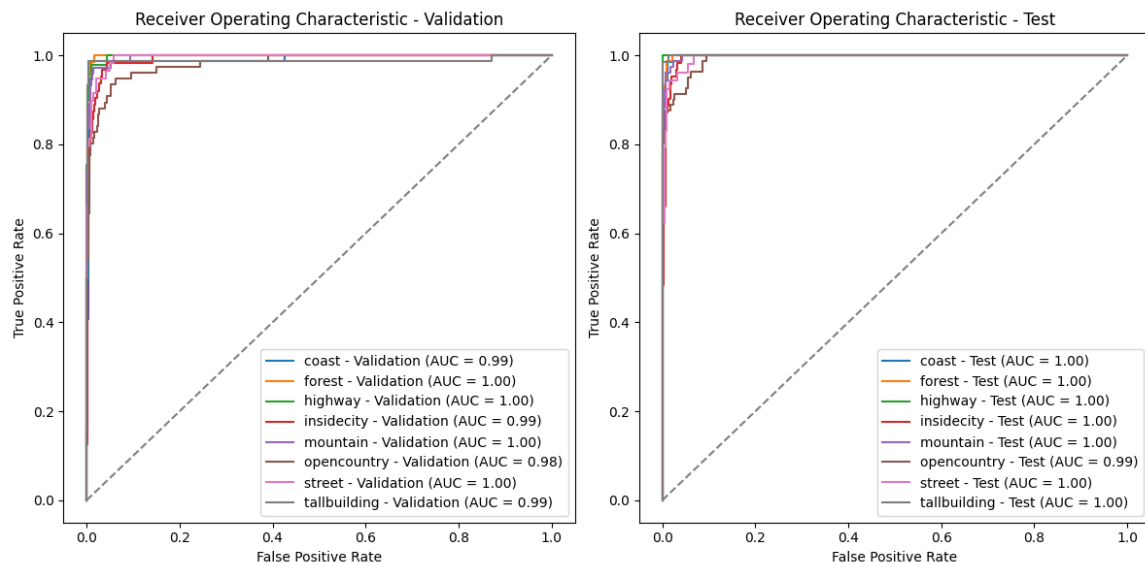


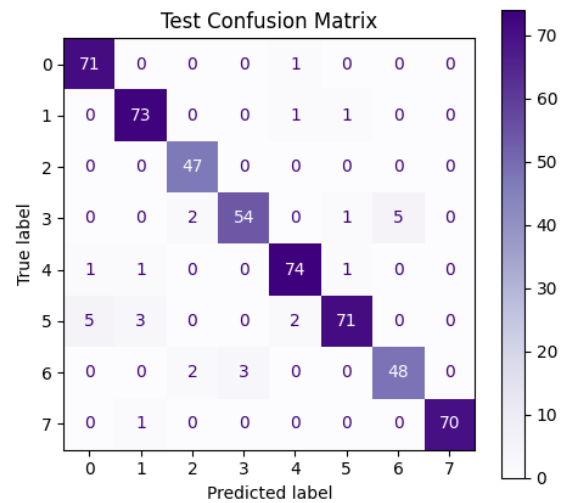
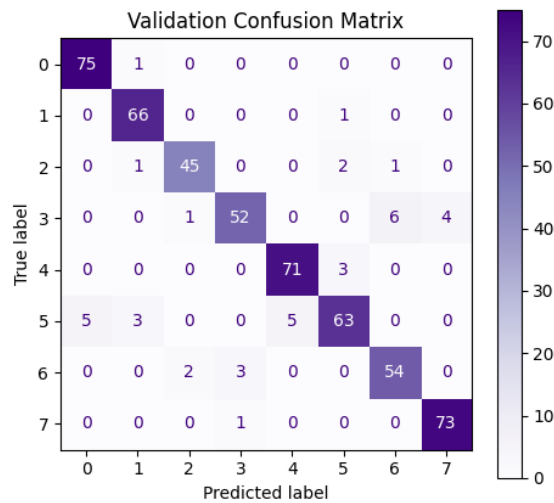
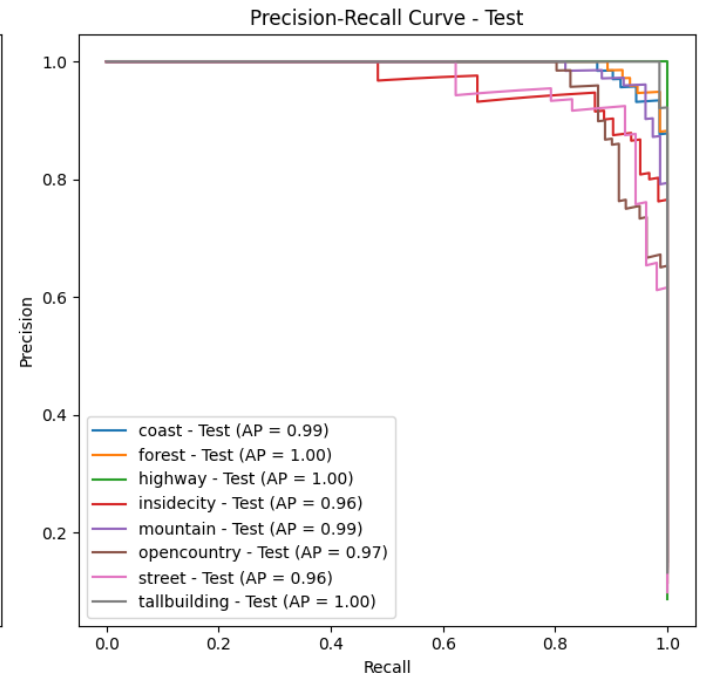
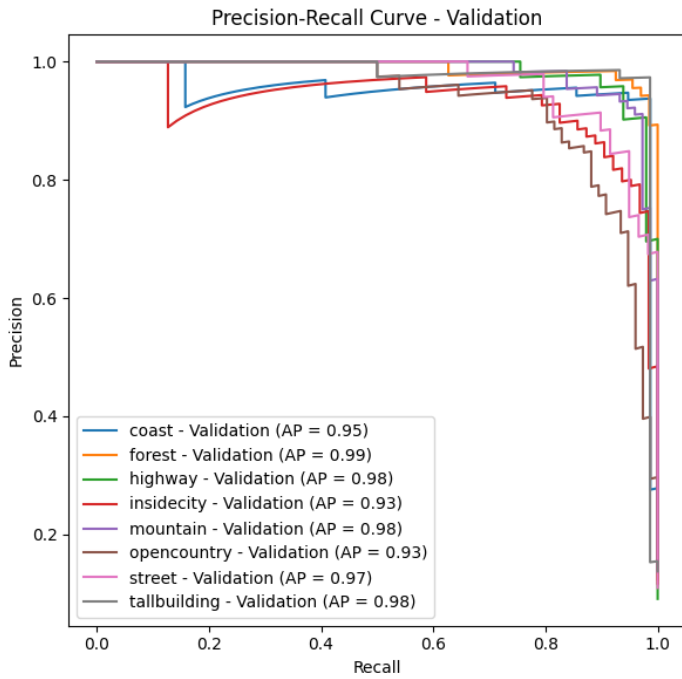


שאלה 2

כמו שאלה 1 רק שהשתמשנו בvgg imagenet כדי לחלץ מאפיינים במקום sift. נשים לב שפה דווקא נבחר פרמטר רגולציה שונה וגאמא שונה. ובטסט קיבלנו 100 עבור ה-AUC!

```
Best SVM Parameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
Validation AUC (One-vs-Rest): 0.99
Test AUC (One-vs-Rest): 1.00
```





שאלה 3

a

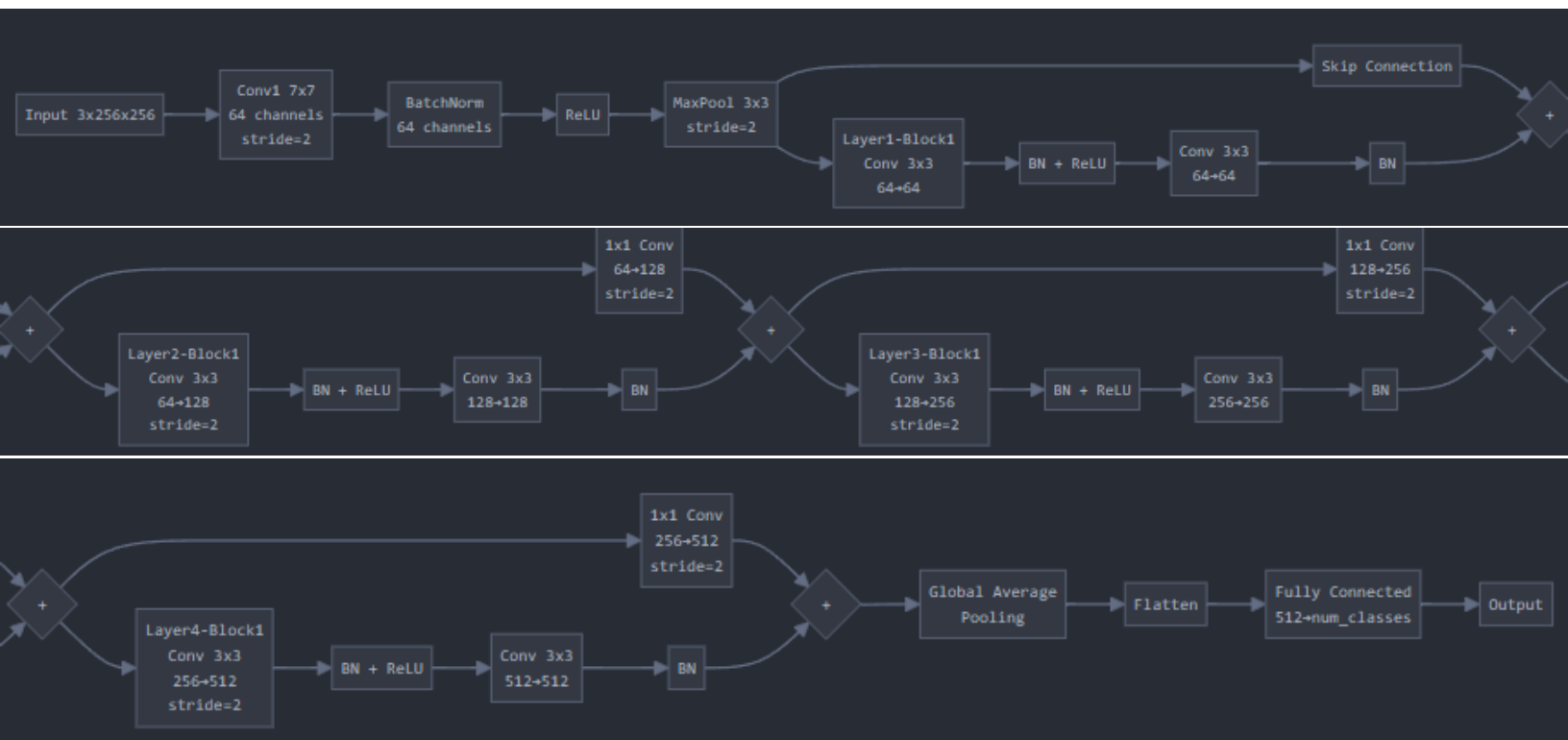
יש קוד בתוכנית שיצר את שלהלן: (התמונות נראות כך כי הן כמובן לאחר טרנספורמציה)



b

הצגת מבנה הרשת: (יש להביט מלמעלה למטה)

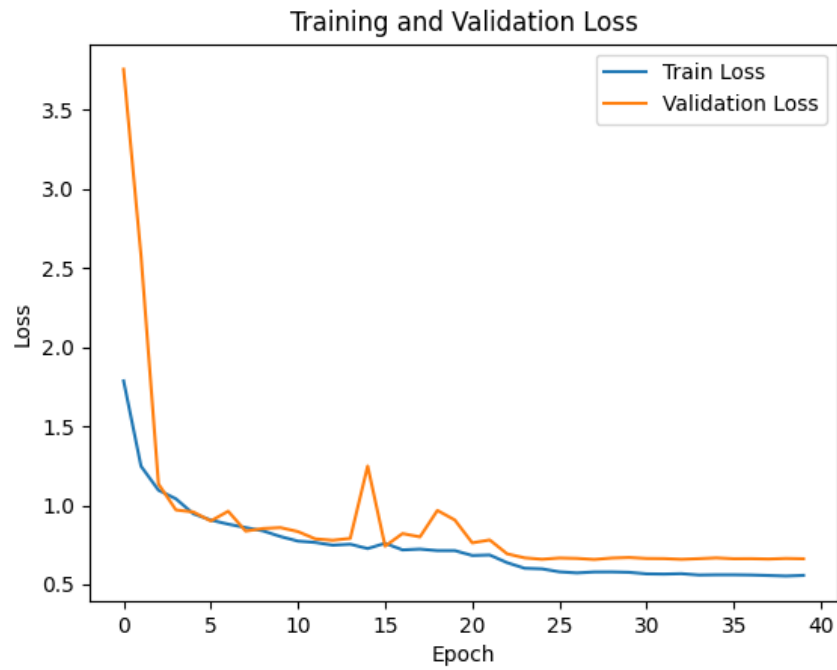
כמו בשאלות 1,2 חילקתי אימון ולידציה וטסט ביחס 20 20 60. השתמשתי ברשת עמוקה עם בלוקים שיוריים. החוקרים שחשבו על רעיון זה שמו לב שיש רשתות עמוקות מדי שלא לצורך, כלומר קיימות רשתות פחות עמוקות עם ביצועים טובים לא פחות ואף יותר (התאמת יתר למשל יכולה לקרות). נניח שקלט של השכבה הוא y_1 אזי הקלט של הבלוק השיורי יהיה $y = y_1 + x$ כאשר x הקלט, כלומר הוספנו את הקלט לפלט הסטנדרטי וזהו הפלט הסופי. כך אנחנו מאפשרים לרשת ללמוד את פונקציית הזהות במידת הצורך ובכך לייחס פחות חשיבות לשכבות עודפות ע"י לקיחת המסלול העוקף של קירוב לפונקציית הזהות. בדיעבד, יצא שהבלוקים הללו פותרים גם את בעיית הגרדיאנט המתאפס. שכן הגרדיאנט של פונקציית המחיר מורכב לפי כלל השרשרת מגרדיאנטים רבים, מספיק כמובן שאחד מהם יתאפס בשביל לגרום גם לו להתאפס (זה יכול לקרות למשל ע"י הפעלת relu). אך עם בלוק שיורי בגלל הוספת פונקציית הזהות הוא לא יתאפס. וכן, מכיוון שהמבנה הדומה של בלוק שיורי לשכבה לינארית, $y_1 = y_1(x) + x$, בלוק שיורי בעצם חייב להיות בגודל לפחות 2 באנאלוגיה לכך שחייבת להיות אקטיבציה לינארית בין 2 שכבות לינאריות (אחרת הן שקולות לשכבה יחידה). כמו כן, השתמשנו ב batch normalization ואתחול פרמטרים רנדומלים לשם אופטימיזציה (נרמול עוזר למקרים בהם בסט הנתונים יש סקאלה בסדר גודל שונה לאחד ממשתני הקלט לרשת, ואתחול שובר סימטריה ומאפשר יותר גמישות לכושר ההבעה של המודל). וכן השתמשנו בסוגים שונים של פולינג לשם downsampling ולשם קבלת מאפיינים אשר אינם רגישים לתזוזות קטנות של הקלט. שכן בסופו של דבר מעניין אותנו קיום מאפיין מסוים ולא המיקום המדויק שלו.



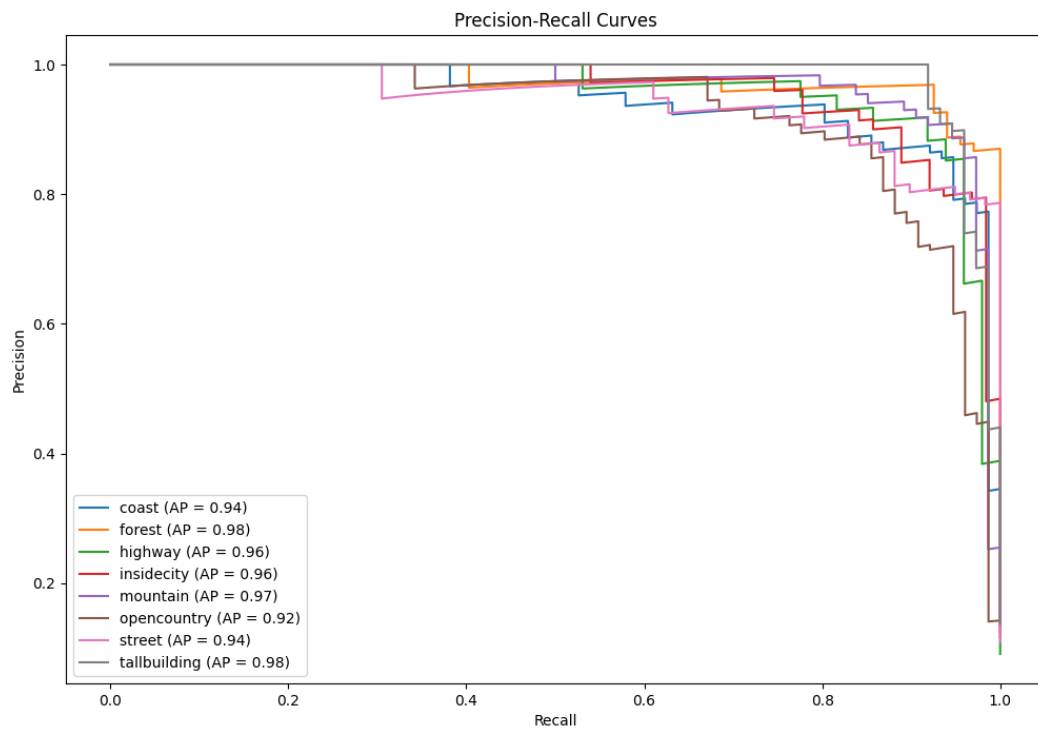
c

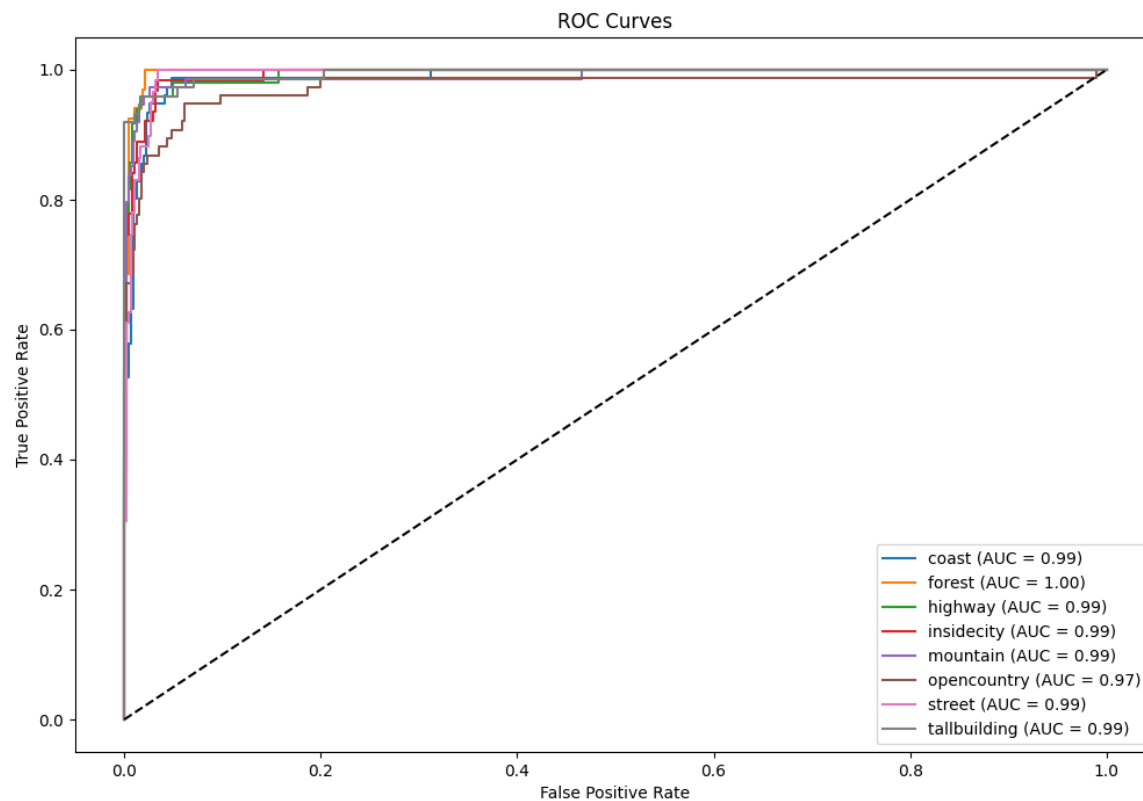
בקובץ הקוד המצורף ק3.py

d



e





Test Precision (Macro Average): 0.9075
 Test Accuracy: 0.9052

בריצה נוספת עם 40 אפוקים שלא שמרתי את הגרפים שלה הגעתי גם לכמעט 92 אחוז:

Test Precision (Macro Average): 0.9176
 Test Accuracy: 0.9126

f

לאחר ניסוי וטעייה הגעתי לגודל 128 עבור הבאטץ ומצאתי אותו כאופטימלי. כמו כן, השתמשתי באופטימיזר של אדם עם רגוליזציה של 12 למניעת אוברפיטינג ומניעת העדפת משקלים גדולים, וכן גם scheduler שמשנה את קצב הלימוד. אדם מכיל שני שיפורים לאלגוריתם מורד הגרדיאנט:

1. קצב למידה אדפטיבי-פרמטרים בעלי השפעה גדולה על הפונקציה יעודכנו בקצב איטי על מנת לא ליצור מצב בו הצעד גדול מדי ואנו מדלגים הלוך חזור מעל העמק של נק המינימום ולא מתכנסים. בעוד פרמטרים בעלי השפעה נמוכה יעודכנו מהר יותר על מנת להאיץ התכנסות. את זאת מבצעים על ידי חישוב קירוב לשונות הגרדיאנט (שונות גבוהה גוררת הקטנת קצב הלמידה).

2. מומנטום – אדם מתבסס על sgd אשר דוגם מיניבאטצים רנדומלית. אי לכך עלולות להיווצר בעיות התכנסות וכאן בא המומנטום אשר מתחשב גם בגרדיאנט הנוכחי של האיטרציה וגם בממוצע של כל הגרדיאנטים הקודמים, כך ממתן את הרנדומליות. וגם המומנטום מתחשב בממוצע של הגרדיאנטים הקודמים כמעין מהירות וכך גורם לצעדים גודל וכיוון שונים במשטח פונק המחר ובכך מסייע בהאצת ההתכנסות. הנה הנוסחאות למהירות ולקירוב השונות ועדכון הפרמטרים של אדם:

$$v_t = \beta_1 v_{t-1} + (1 - \beta_1) \tilde{\nabla} C_t$$

$$s_t = \beta_2 s_{t-1} + (1 - \beta_2) (\tilde{\nabla} C_t)^2$$

$$Parameters = Parameters - \alpha \frac{v_t}{\sqrt{s_t}}$$

לאחר כל השיפורים הגעתי לנכונות ודיוק של כמעט 92 אחוז, כמו שהראיתי בסעיף הקודם