

Learning, Representation and Computer Vision

Task 2

Eyal Ginosar & Omer Keidar





תוכן עניינים

3.....	הקדמה	1.
3.....	אחוז השגיאה לפני שיפור המודל	2.
4.....	שיפור המודל	3.
4.....	Data Augmentation : שיפור 1	
4.....	Activation function : שיפור 2	
5.....	שיפור 3 : מספר השכבות לאימון ברשת	
6.....	שיפור 4 : מספר Epochs	
7.....	שיפור 5 : פרמטרי האופטימיזציה	
8.....	מודל סופי	4.
9.....	שגיאות מסוג ראשון ושני	5.
10.....	Recall-Precision curve	6.

1. הקדמה

במטלה זו משימתנו הייתה להשתמש ברשת (ResNet50V2), רשת קיימת שכבר אומנה על סיווג של 1000 מחלקות ולבצע בעזרתה סיווג בינארי למחלקת פרחים. את הסיווג נדרשנו לבצע על-ידי מקסום פרמטרי הרשת והתאמתה למשימה החדשה. בכדי להתאים את הרשת למשימה שלנו, קיבלנו מאגר תמונות של פרחים עם התיוג שלהם (1-פרח, 0-לא פרח). על-פי ההוראות, התמונות חולקו כך שהאימון בוצע על 300 התמונות הראשונות והבחינה בוצעה על התמונות שנותרו (172). בפועל, בשיפור המודל ביצענו Data Augmentation ולכן מספר התמונות אשר עליהן ביצענו את אימון הרשת הייתה גדולה מ-300. בנוסף, פרמטרי הרשת המקורית נבחנו ושונה על מנת שהרשת תלמד בצורה טובה יותר את המשימה החדשה שלה. הדו"ח מסכם את כלל הצעדים והשיפורים שבצענו במטרה לשנות את משימת הרשת ולהתאימה בצורה האופטימלית למשימתנו.

2. אחוז השגיאה לפני שיפור המודל

לפני ביצוע שיפורים למודל, הרצנו את הקוד הראשוני עם הפרמטרים הבאים :

Epochs	Batch Size	Optimizer	Activation	Number of layers for training
1	16	SGD : Learning Rate : 0.01 Decay : 0.001	'sigmoid'	מאמנים את הרשת על השכבה האחרונה בלבד

אחוז השגיאה אשר התקבל בהרצת המודל על סט הבחינה הוא 43%. וערך פונקציית ה loss : 0.7452

כלומר קיבלנו דיוק של 56.98% על סט הבחינה עם הפרמטרים הראשוניים.

על מנת להראות את השיפורים מהמצב ההתחלתי, נציג כאן גם את השגיאה על סט הולידציה :

השגיאה על סט הולידציה היא 54.44%, כלומר רק 45.56% דיוק. בשיפור המודל נראה את השיפור על אחוז הדיוק שהתקבל במצב זה.

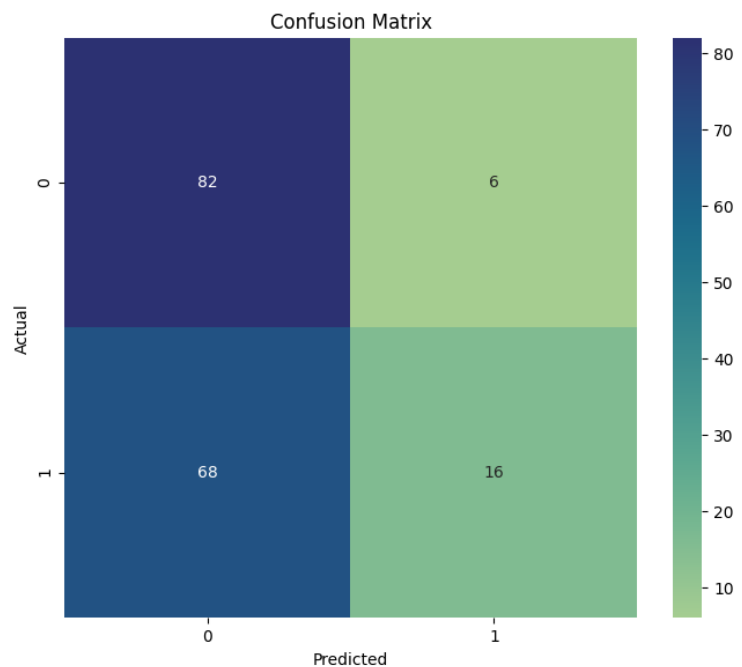
Basic report:

Number of training images: 300

Amount of ``True`` flower images: 164

Precision-recall curve AUC score: 0.625

Test error rate = 0.43023258447647095



3. שיפור המודל

בחלק זה השתמשנו בסט האימון בלבד וחילקנו אותו ל train ו validation (סט האימון חולק ל 70% אימון ו 30% ולידציה). נעזרנו בחלוקה זו על מנת לבצע שיפורים במודל ולמצוא את הפרמטרים המתאימים לו ביותר. בגלל שיקולי זמן ריצה הרצנו את השיפורים אחד אחרי השני, כאשר כל שיפור שכבר בוצע קיבענו ולאחר מכן התבצע השיפור הבא כאשר הוא כבר מתחשב בפרמטרים הקודמים ששופרו.

שיפור 1 : Data Augmentation

מספר התמונות שקיבלנו היה קטן יחסית ולכן מספר התמונות עליהן הרשת התאמנה היה קטן יחסית, לכן בחרנו לבצע תחילה Data augmentation במטרה לאפשר לרשת להתאמן על מספר תמונות גדול יותר, דבר שלהבנתנו יאפשר לרשת לשפר את ביצועיה. את התמונות הנוספות יצרנו על ידי מעבר על כל אחת מהתמונות הקיימות והפעלת פונקציה אשר בוחרת בצורה רנדומלית האם לבצע flip או crop, כך שמכל תמונה נוצרה תמונה נוספת אחת או בכיוון השני (כלומר מראה) או אותה התמונה עם זום. בסופו של דבר קיבלנו סט אימון המורכב מ 600 תמונות במקום 300. בכל השיפורים הבאים שנעשה סט האימון הנ"ל ילקח בחשבון.

לאחר הרצת המודל עם 600 תמונות בסט האימון, אחוז הדיוק שהתקבל על סט הולידציה הוא 67.22%.

```
val_loss: 0.6118 - val_accuracy: 0.6722
```

שיפור 2 : Activation function

פונקציית האקטיבציה הינה פונקציה המופעלת על ערכי הנוירונים בסוף חישובי כל שכבת קונבולוציה. במקרה שלנו, פונקציית האקטיבציה היא פונקציה שתופעל על השכבה האחרונה ברשת והמשמעות היא שה output של הנוירונים בשכבה האחרונה עובר אקטיבציה בסוף התהליך. מטרת האקטיבציה היא לסדר את ה output. לטובת שיפור זה, בחנו 6 פונקציות אקטיבציה שונות מה documentation של keras והן:

```
activations = ['sigmoid', 'relu', 'softmax', 'elu', 'softsign', 'tanh']
```

'sigmoid' – היא פונקציית האקטיבציה כפי שהוגדרה לנו בעבודה : $\frac{1}{1+\exp(-x)}$

'relu' – היא פונקציית האקטיבציה : $\max(0, x)$

'softmax' – היא פונקציית אקטיבציה אשר מבצעת נרמול על ידי הפיכת המספרים להסתברויות על פי הנוסחה :

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K$$

'elu' - היא פונקציית אקטיבציה אשר מחזירה את הערך המקורי במידה והוא גדול מ-0 ומחזירה $e^x - 1$ במידה והערך המקורי של X קטן מ-0.

'softsign' – היא פונקציית אקטיבציה : $\frac{x}{|x|+1}$

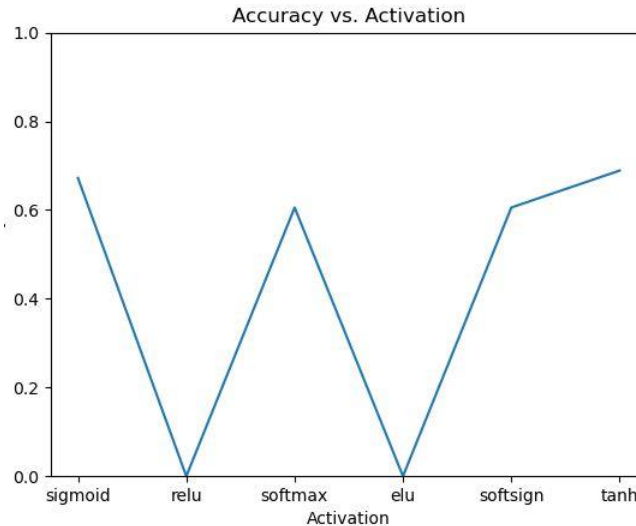
'tanh' – היא פונקציית אקטיבציה היפרבולית מהצורה : $\frac{e^x - e^{-x}}{e^x + e^{-x}}$

אחוז הדיוק שהתקבל על סט הולידציה עבור כל אחד מהערכים (הדיוק של כל ערך לפי הסדר המוצג למעלה) :

```
Tuning activation acc = [0.6722221970558167, 0.0, 0.605555534362793, 0.0, 0.605555534362793, 0.6888889074325562]  
The chosen activation is tanh
```

אחוז הדיוק הגבוה ביותר שהתקבל, התקבל עבור השימוש בפונקציית 'tanh' והוא 68.88% על סט הולידציה. בגלל הקרבה באחוזי הדיוק של 'tanh' ו 'sigmoid', החלטנו לנסות ולבחון את שניהם וגילינו כי המשך שיפורים עם 'sigmoid' מניב מודל טוב בהרבה ולכן בחרנו דווקא להמשיך איתו ולא עם 'tanh'. 'sigmoid' בשלב זה נתן אחוז דיוק של 67.22% על סט הולידציה.

גרף המייצג את השינוי באחוזי הדיוק על סט הולידציה כתלות בשינוי פונקציית האקטיבציה :



לאחר קיבוע שיפור זה נקבל את הרשת הבאה (אותה הרשת שהתחלנו איתה):

Epochs	Batch Size	Optimizer -SGD	Activation	Number of layers for training
1	16	Learning Rate : 0.01 Decay : 0.001	'sigmoid'	מאמנים את הרשת על השכבה האחרונה בלבד

נשתמש ברשת זו להמשך שיפור המודל.

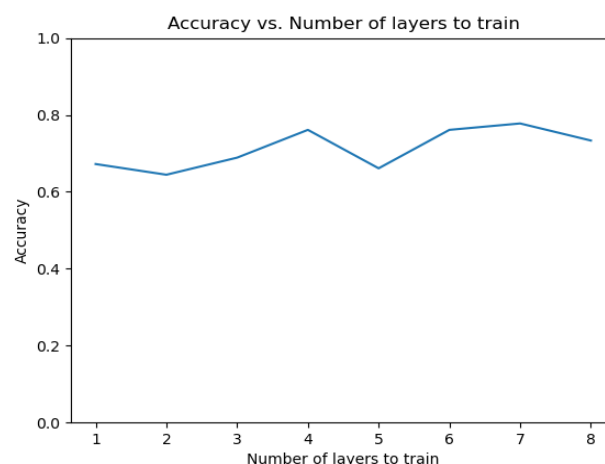
שיפור 3 : מספר השכבות לאימון ברשת

לאחר שקיבענו את פונקציית האקטיבציה, רצינו לבדוק האם אימון של יותר מהשכבה האחרונה ברשת משפר את אחוזי הדיוק. לצורך כך בדקנו את הדיוק כאשר אנו מאמנים רק את השכבה האחרונה וכך ירדנו למטה עד 8 שכבות אחרונות. המוטיבציה שהובילה אותנו לבצע את האימון על מספר שכבות שונות (החל מהאחרונה ואחורה) היא המחשבה שכל שנאמן בעצמנו יותר שכבות ברשת הקיימת, כך נתאים את הרשת טוב יותר למשימה שלנו, מכיוון שכאשר אנו מאמנים שכבות נוספות אשר נמצאות בהיררכיה גבוהה ברשת, אנחנו מאמנים את הפיצורים המורכבים יותר אל מול התמונות החדשות והמשימה החדשה. מהתוצאות ניתן לראות שאכן כך הדבר ובחלק מכמות השכבות, קיבלנו תוצאות טובות יותר על סט הולידציה.

אחוז הדיוק שהתקבל על סט הולידציה עבור כל אחד מהערכים :

Tuning layers acc = [0.6722221970558167, 0.644444465637207, 0.6888889074325562, 0.7611111402511597, 0.661111164093018, 0.7611111402511597, 0.7777777910232544, 0.7333333492279053]
The chosen number of layers is 7

אחוז הדיוק הגבוה ביותר המתקבל הוא 77.77% עבור אימון של 7 שכבות אחרונות של הרשת. להלן גרף המייצג את השינוי באחוזי הדיוק כתלות באימון מספר שכבות שונות של המודל :



נבחר ב 7 שכבות אחרונות לאימון ונקבע את המודל בצורה הבאה :

Epochs	Batch Size	Optimizer -SGD	Activation	Number of layers for training
1	16	Learning Rate : 0.01 Decay : 0.001	'sigmoid'	מאמנים את הרשת על 7 השכבות האחרונות

נשתמש במודל שקיבענו להמשך השיפורים.

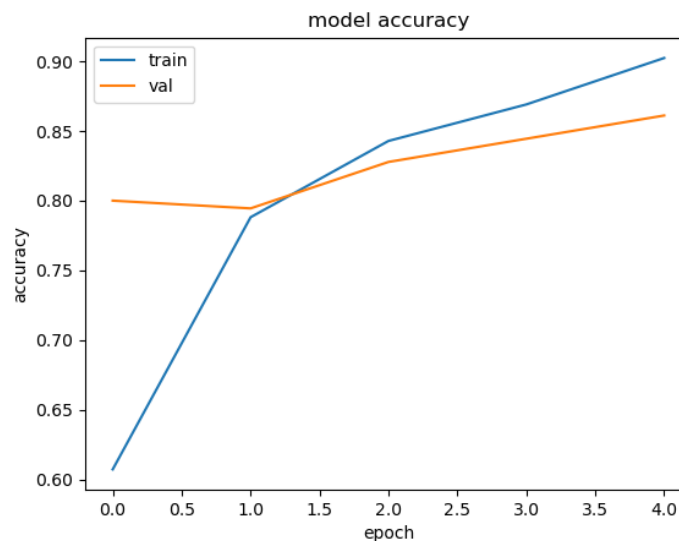
שיפור 4 : מספר Epochs

ה Epochs הוא מדד למספר הפעמים בהן משתמשים בכל סט האימון לאימון הרשת, כלומר, מספר האיטרציות שמתבצעות בשלב האימון של המודל. בכל epochs מתבצע feedforward עבור כל batch בנפרד, כאשר בסוף כל epochs מתבצעת למידה - עדכון משקולות. מספר ה epochs משפיע על מידת ה overfitting ומספר איטרציות גבוה מידי עשוי להוביל ללמידת יתר ולפגוע ביכולת ההכללה של המודל ובכך לפגוע בו. מספרי ה epochs שבחנו היו בין 1-5 והתוצאות שהתקבלו הן :

```
tune epochs acc = [0.7888888716697693, 0.7888888716697693, 0.800000011920929, 0.7833333611488342, 0.8611111044883728]
the chosen epochs is 5
```

ניתן לראות כי הערך הגבוה ביותר התקבל עבור 5 epochs ואחוז הדיוק שהתקבל על סט הולידציה היה 86.11%.

להלן גרף המציג את השינוי באחוזי הדיוק על סט הולידציה והאימון כתלות ב Epochs :



מכאן שהרשת המתקבלת לאחר קיבוע פרמטר זה היא :

Epochs	Batch Size	Optimizer -SGD	Activation	Number of layers for training
5	16	Learning Rate : 0.01 Decay : 0.001	'sigmoid'	מאמנים את הרשת על 7 השכבות האחרונות

נשתמש ברשת הזו להמשך השיפורים.

שיפור 5 : פרמטרי האופטימיזציה

בחרנו במודל שלנו להשתמש בפונקציית האופטימיזציה Stochastic gradient decent –SGD. משמעות פרמטר האופטימיזציה במודל הינה באיזה אופן תתבצע האופטימיזציה של המודל, כלומר בעזרת איזו פונקציה ובעזרת אילו פרמטרים פונקציה זו תעזר. פונקציית SGD הינה פונקציה שמבצעת אופטימיזציה על סמך הגרדיאנט כפי שלמדנו. פונקציה זו מקבלת כפרמטרים learning rate ו decay אשר משפיעים על גודל ואופן הצעדים שהמודל יבצע בזמן הלמידה ולכן בחרנו להתרכז בשני הפרמטרים הללו.

Learning rate - η , פרמטר המשפיע על גודל צעד הלמידה בכך שהוא מאפשר לצעדי הלמידה לקחת בחשבון גם צעדי למידה קודמים שהיו :

$$w := w - \eta \nabla Q(w) = w - \eta \sum_{i=1}^n \nabla Q_i(w)/n,$$

Decay : פרמטר זה מאפשר לשנות את גודלה של η עבור כל epoch כך שכל שמתקדמים בלמידה היא תגדל. באופן כזה, צעדים שנעשו מוקדם יותר בלמידה יקבלו פחות משמעות בצעדים שהתבצעו מאוחר יותר. משתנה הלמידה מתעדכן בצורה

$$lr = lr * \frac{1}{iteration * decay}$$

בעבור הערכים הללו, בחנו 12 קומבינציות וזאת במטרה להבין מהו השילוב הטוב ביותר בין הערכים הללו שיוביל לתוצאות הטובות ביותר. הערכים שנבדקו הם :

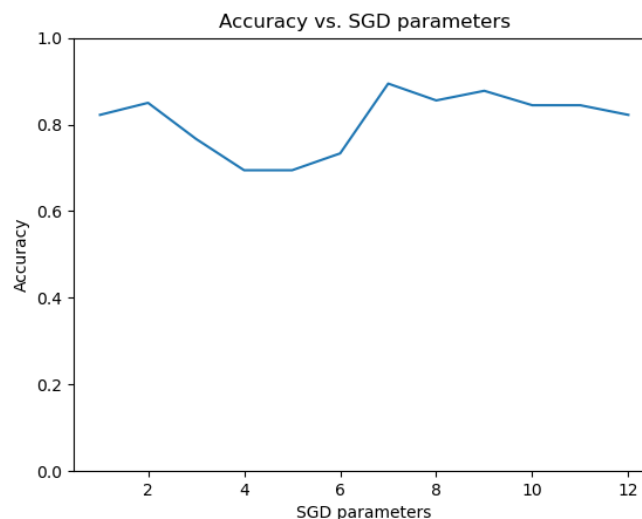
```
learning_rates = [0.01, 0.03, 0.05]
decays = [0.001, 0.005, 0.01, 0.1]
```

ערכי הדיוק שקיבלנו לכל קומבינציה היו :

```
ie lr and decay acc = [0.8222222328186835, 0.8500000238418579, 0.7666666507720947, 0.6944444179534912, 0.6944444179534912, 0.7333333492279053, 0.8944444465637207, 0.855555534362793, 0.877777752604675, 0.8444444537162781, 0.8444444537162781, 0.8222222328186835]
t accuracy index is 6
index = 1, decay index = 2
```

הדיוק הטוב ביותר התקבל עבור קומבינציה 7 (שהיא אינדקס 6 בגלל שמתחילים ב0), הערכים הם Learning Rate = 0.03 עם Decay = 0.01 ואחוז הדיוק על סט הולידציה הוא 89.44%.

להלן גרף המציג את השינוי באחוזי הדיוק על סט הולידציה כתלות בקומבינציות השונות שבחנו. ציר X מבטא את מספר הקומבינציות.



הרשת העדכנית המתקבלת היא :

Epochs	Batch Size	Optimizer -SGD	Activation	Number of layers for training
5	16	Learning Rate : 0.03 Decay : 0.01	'sigmoid'	מאמנים את הרשת על 7 השכבות האחרונות

4. מודל סופי

המודל הסופי שהתקבל הוא המודל שהצגנו לאחר השיפור האחרון :

Epochs	Batch Size	Optimizer -SGD	Activation	Number of layers for training
5	16	Learning Rate : 0.03 Decay : 0.01	'sigmoid'	מאמנים את הרשת על 7 השכבות האחרונות

בעזרת מודל זה נבצע חיזוי על סט הבחינה שלנו ונציג את התוצאות :

Basic report:

Number of original training images: 300

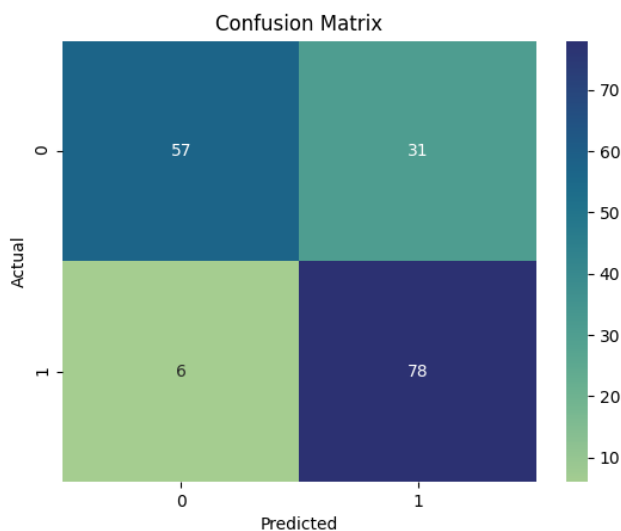
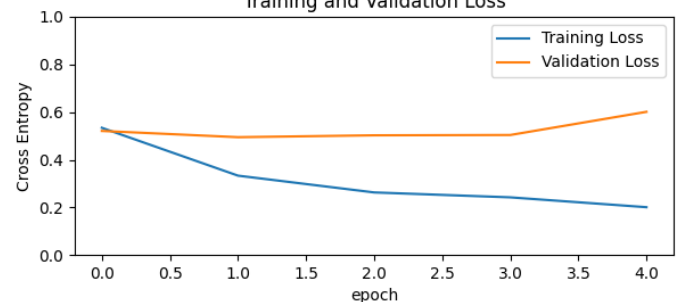
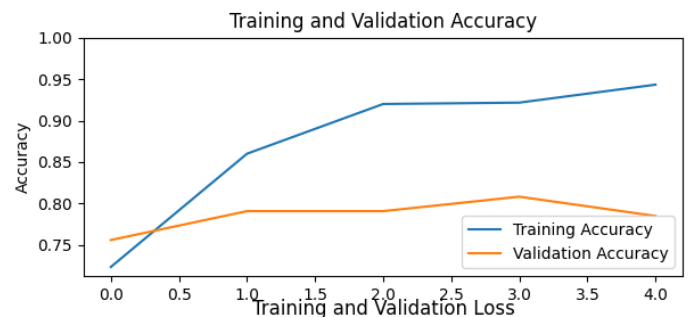
Amount of ``True`` flower images: 164

Number of augmented images: 300

Amount of ``True`` augmented images: 164

Precision-recall curve AUC score: 0.8544764755822054

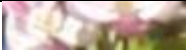







Test error rate = 0.19186043739318848





5. שגיאות מסוג ראשון ושני

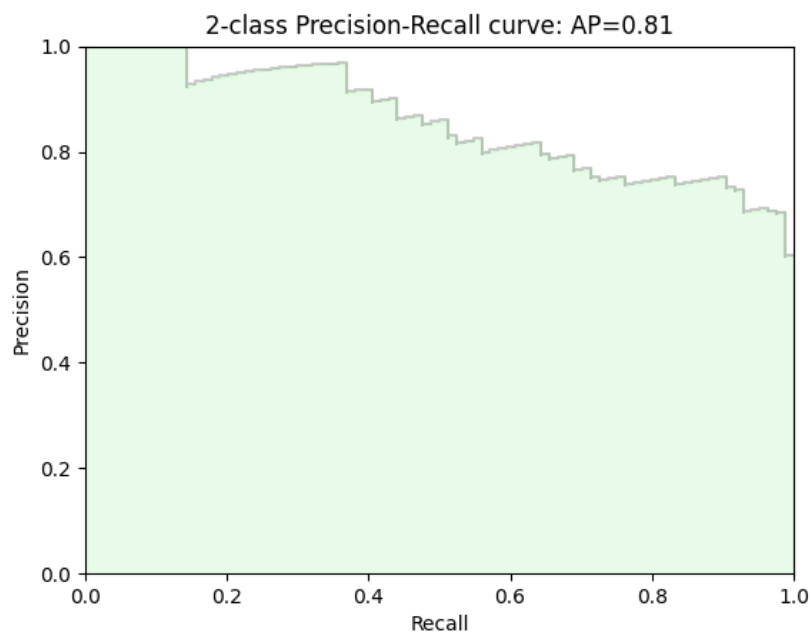
שגיאת מסוג ראשון : המודל שיערך שהתמונה היא פרח אך היא לא.

שגיאה מסוג שני : המודל שיערך שהתמונה היא לא פרח אך היא כן.

index	Type 1 errors		Type 2 errors	
	picture	score	Picture	score
1		0.995649		0.111956686
2		0.98607516		0.3823078
3		0.9850794		0.40054673
4		0.98345625		0.4250559

5		0.97699016		0.43682384
---	---	------------	--	------------

Recall-Precision curve .6



Precision-recall curve AUC score: 0.8544764755822054

הגרף מציג את רמת הדיוק של המודל אל מול רמת "קריאות השב" (סיווג למחלקה פרח) כפונקציה של שינוי ה threshold, אשר מחליט על סיווג המחלקות ובעצם מגדיר מעל איזה אחוז נסווג למחלקת פרח. מהגרף אנו יכולים להבחין בכך שככל שה threshold יורד, מספר הקריאות עולה. כלומר, ככל שהסף יהיה נמוך יותר, ככה יותר תמונות יסווגו כפרחים. בנוסף ניתן לראות כי ככל שה threshold עולה, כמות הקריאות יורדת, וזאת מפני שנדרשת הסבירות גבוהה כדי לסווג תמונה למחלקת פרח. מכאן שעבור סף נמוך, הסיכוי לטעות בסיווג של תמונות נכונות יורד, אך שיעור אזהקות השווא עולה ולהפך כאשר הסף גבוה.