

תיאור הפרויקט – Transfer Learning

בפרויקט בחרתי לאמן רשת על בסיס RESNET50V2 שתסווג תמונות לפרחים או לאץ

ניסיתי לשפר את ביצועי המסווג על ידי הדברים הבאים:

1. שימוש באוגמנטציה של סט האימון.
 2. הוספה ואימון של שתי שכבות נוספות ברשת RESNET50V2.
 3. שימוש ב-DROPOUT של 2 השכבות האחרונות על מנת להימנע מ-over-fitting.
 4. שימוש ב-maxPooling במקום AvgPooling בשכבה אחת לפני אחרונה.
 5. שימוש ב-Flatten של השכבה האחת לפני אחרונה- להפוך את השכבה האחרונה ממימד 3 לוקטור.
 6. קומבינציות של הדברים לעיל (פרט ל-4 ו-5 יחד כי זה לא הגיוני).
- * כל הקונפיגורציות הללו קיימות בקוד וניתנות להפעלה בפונקציית CreateModel.
- * בכדי להימנע מ-over-fitting ומזמן ריצה ארוך, בעת אימון המודלים השתמשתי ב-EarlyStopping- הפרשתי מסט הנתונים 10% שישמשו כסט ולידציה ועצרתי את אימון הרשת כאשר ביצועי הרשת לא השתפרו על סט הולידציה (פרמטר ה-"patience" היה 4 epochs).

סט הנתונים

סט תמונות של פרחים מהמכון הוולקני. הסט כולל 473 תמונות חתוכות של פרחים או לא ועם התוויות שלהם.

טענו את התמונות, העברנו אותם לסקאלה של RGB ושינינו את גודלם ל-3X224X224 כמתבקש בשביל הרשת.

300 התמונות הראשונות שימשו לסט האימון ושאר התמונות לסט הבחינה.

במהלך העבודה השתמשתי ב-Cross Validation על מנת לכוון את הפרמטרים הנדרשים. השתמשתי ב-K-Folds, כאשר K=5. בחלק מהמקרים לא התבצע Cross-Validation אלא בסט ולידציה קבוע של 25%, מצוין זאת בדו"ח.

הרשת הבסיסית

על מנת להצליח לסווג פרחים או לא פרחים היה צורך לבצע טרנספורמציה לשכבה האחרונה של הרשת, לכן בחרתי להשתמש בפונקציית Sigmoid:

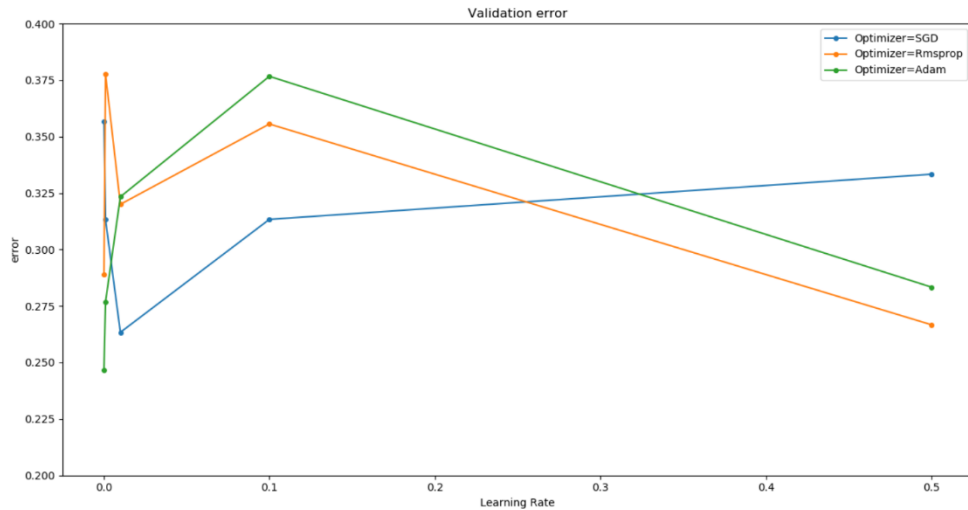
$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

והחלטנו לסווג לפרחים במידה והפונקציה העריכה מעל 0.5.

על מנת למצוא את הoptimizer הטוב ביותר ואת קצב הלמידה התבצע כוונן פרמטרים של הרשת עבור batch=30 ו-epoch=10 ובדקתי עבור איזה מצב אנחנו מקבלים את אחוז השגיאה הנמוך ביותר (ערך שגיאה ממוצע עבור חיזוי של כל אחד מ-5 folds). מרחב החיפוש עבור שני הפרמטרים הללו:

קצב למידה: 0.0001, 0.001, 0.01, 0.1, 0.5.

אופטימיזר: Adam, SGD, RMSprop.

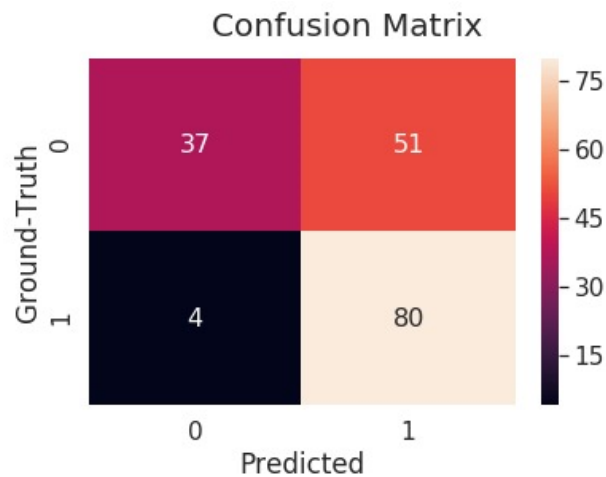


כפי שניתן לראות באיור לעיל, אחוז השגיאה הממוצע הנמוך ביותר שקיבלנו היה עבור:

.learning rate=0.0001 -I optimizer=Adam

אחוזי הדיוק על סט הבחינה הם: 68.02%

מטריצת המבוכה:

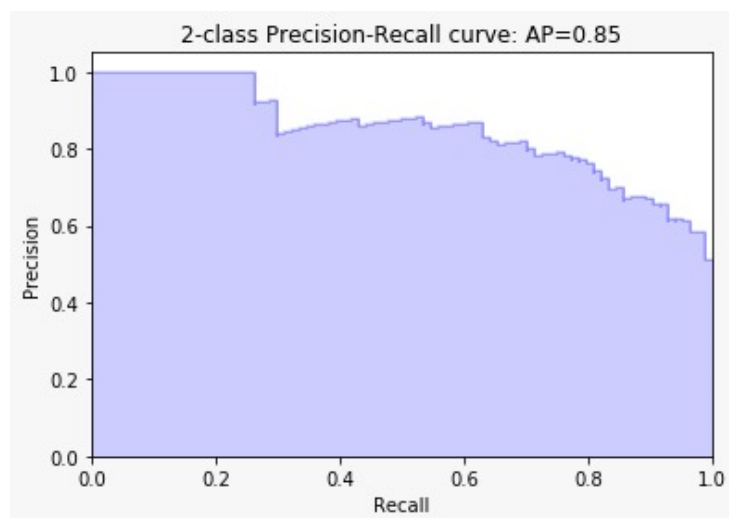


ניתוח תוצאות מטריצת המבוכה:

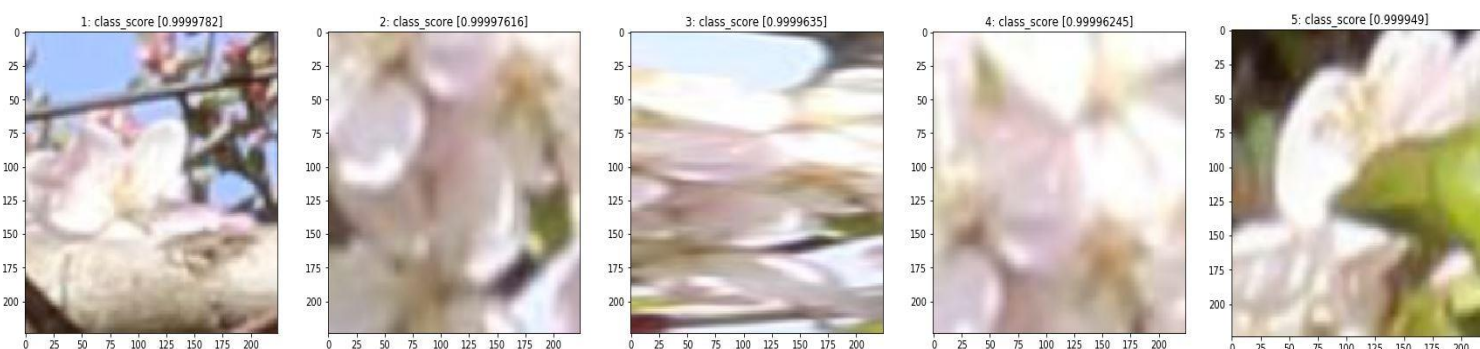
נשים לב כי הרשת שאימנו חוזה ברובה שהתמונות הן פרחים (131 מתוך 172). עקב כך כמעט ואין לנו טעויות מסוג שני לעומת המון טעויות מסוג ראשון. עקב כך נשקול להגדיל את הסף שממנו אנחנו מחליטים לסווג כפרח במודל הסופי (כרגע עומד על 0.5).

Precision-Recall Curve:

העקומה מיישרת קו עם מטריצת המבוכה, precision שנשאר גבוה גם שה-Recall עולה עד 0.6.



5 התמונות (המקוריות, לא לאחר preprocess_input) עם הטעות מסוג ראשון החמורות ביותר מסודרות משמאל לימין. כלומר התמונות שמסווגות כלא פרחים, אך הרשת סיווגה כפרחים. ה-class score זה הערך שהרשת נתנה, ככל שקרוב יותר ל-1 הטעות גדולה יותר:



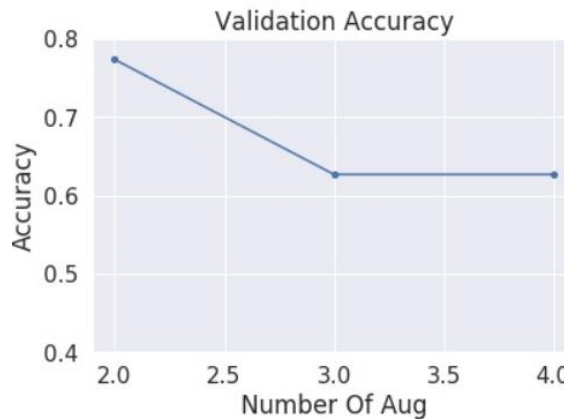
4 התמונות (המקוריות, לא לאחר preprocess_input) עם הטעות מסוג שני החמורות ביותר מסודרות משמאל לימין. כלומר התמונות שמסווגות כפרחים, אך הרשת סיווגה כלא פרחים. ה-class score זה הערך שהרשת נתנה, ככל שקרוב יותר ל-0 הטעות גדולה יותר:



ניסיון שיפור ראשון-Data Augmentation

התבצע היפוכים אופקיים ואנכיים, סיבוב, זום ושינוי תמונות עבור סט האימון. בחנתי את הנתונים ושמתי לב שרוב התמונות צולמו מלמעלה. לכן הוחלט להשתמש בהיפוך אנכי בנוסף להיפוך האופקי. בנוסף לאחר ניסיון שמתי לב שהגדלות וחיתוכים של התמונה מביאים לביצועים פחות טובים, ככל הנראה בגלל תיוג התמונות ההתחלתי ולכן עבדתי רק עם היפוכים וסיבובים. בנוסף ביצענו PCA COLOR AUGMENTATION לאחר מחקר קטן וראיתי שבפועל התוצאות טובות יותר בגלל שהוא מייצר תמונות עם צבעים יותר שכיחים ובכך הופך את אימון המודל לכללי יותר ורובוסטי (קשיח) יותר. מצורף קישור <https://machinelearning.wtf/terms/pca-color-augmentation/>

כדי לדעת כמה היפוכים/סיבובים הרשת מתאמנת בצורה המיטבית ביותר בוצע כיוון פרמטרים ע"י חלוקה לסט אימון וסט ולדציה. הפעם עבדתי עם סט ולדציה 1 של 25% מסך סט האימון שנבחר בהתחלה באופן רנדומי (לא התבצע cross-validation בגלל שמספר התמונות הוכפל פי 3/4/5 וזמן הריצה היה ארוך מדי). להלן גרף שמראה את אחוזי הדיוק על סט הולידציה:

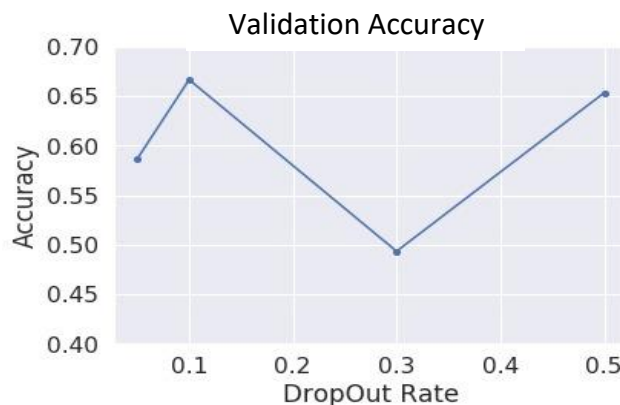


אחוז הדיוק שהתקבל על סט הבחינה היה: 60.19%

ניסיון שיפור שני – DropOut

השתמשתי ב-Drop out ב-2 השכבות האחרונות של הרשת. כלומר בחרתי את אחוז הנוירונים אשר יהיו לא אקטיביים ב-2 השכבות האחרונות. מטרת השימוש בקונפיגורציה זו הייתה להימנע מ-overfitting.

התבצע כיוון פרמטרים לאחוז ה-drop-out (השתמשתי בפרמטרים שמצאנו ברשת הבסיסית כלומר learning-rate=0.0001, optimizer=adam, שיעור ה-DROPOUT הטוב ביותר היה 0.1. הגרף מוצג להלן:



אחוז הדיוק שהתקבל על סט הבחינה היה: 69.17%

ניסיון שיפור שלישי- הוספה ואימון של 2 שכבות נוספות

בחרתי להוסיף 2 שכבות נוספות לרשת- fully connected, אחת עם 512 ניורונים ואחת עם 256 ניורונים ולאפשר לרשת ללמוד את המשקולות של השכבות הללו בנוסף ללמידה של שכבת היציאה.

אחוז הדיוק שהתקבל על סט הבחינה היה: 70.33%

ניסיון שיפור רביעי- שינוי הקונפיגורציה של השכבה האחרונה

בשכבה האחרונה של RESNET50V2 מתבצע average pooling עבור כל מפה. ניסיתי לשפר את ביצועי הרשת באמצעות שינוי של דרך החישוב עבור כל מפה. 2 השינויים הם:

- ביצוע מקסימיזציה עבור כל מפה: max pooling.
- שיטוח המפה לוקטור (Flatten). שימוש בכל המפה ע"י הפיכתה לוקטור אחד ארוך.

שיטה	Max-Pooling	Flatten
אחוזי דיוק	65.33%	71.18%

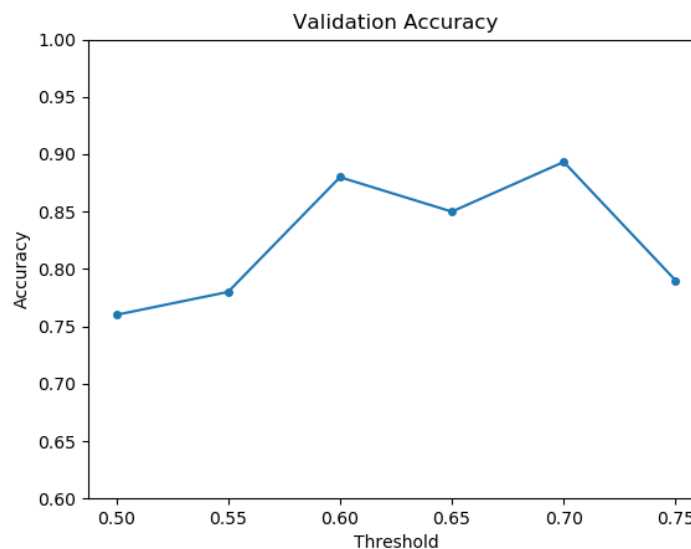
המודל הנבחר

ניסיתי את כל הקומבינציות האפשריות מבין השיפורים שהצענו לעיל (למשל הגדלת סט האימון ע"י אוגמנטציה בנוסף להוספת 2 שכבות נוספות שעליהן ביצענו DROPOUT)

בסופו של דבר המודל שהפיק את אחוזי הדיוק הטובים ביותר הוא:

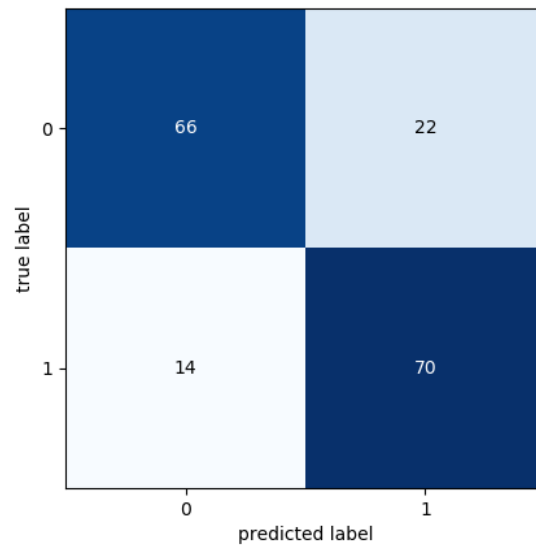
שיטוח של השכבה אחת לפני אחרונה והוספת 2 שכבות Fully Connected נוספות.

לבסוף, בהתאם לעובדה שראינו כי הרשת נוטה לסווג יותר פרחים הוחלט לכוון את הפרמטר של הסף ממנו הרשת מסווגת פרחים. להלן גרף שמציג את אחוזי הדיוק על סט הולידציה (25% אחוז מסט האימון), הערך הנבחר הוא 0.7:



אחוזי הדיוק על סט הבחינה במודל הסופי הם: 79.07%

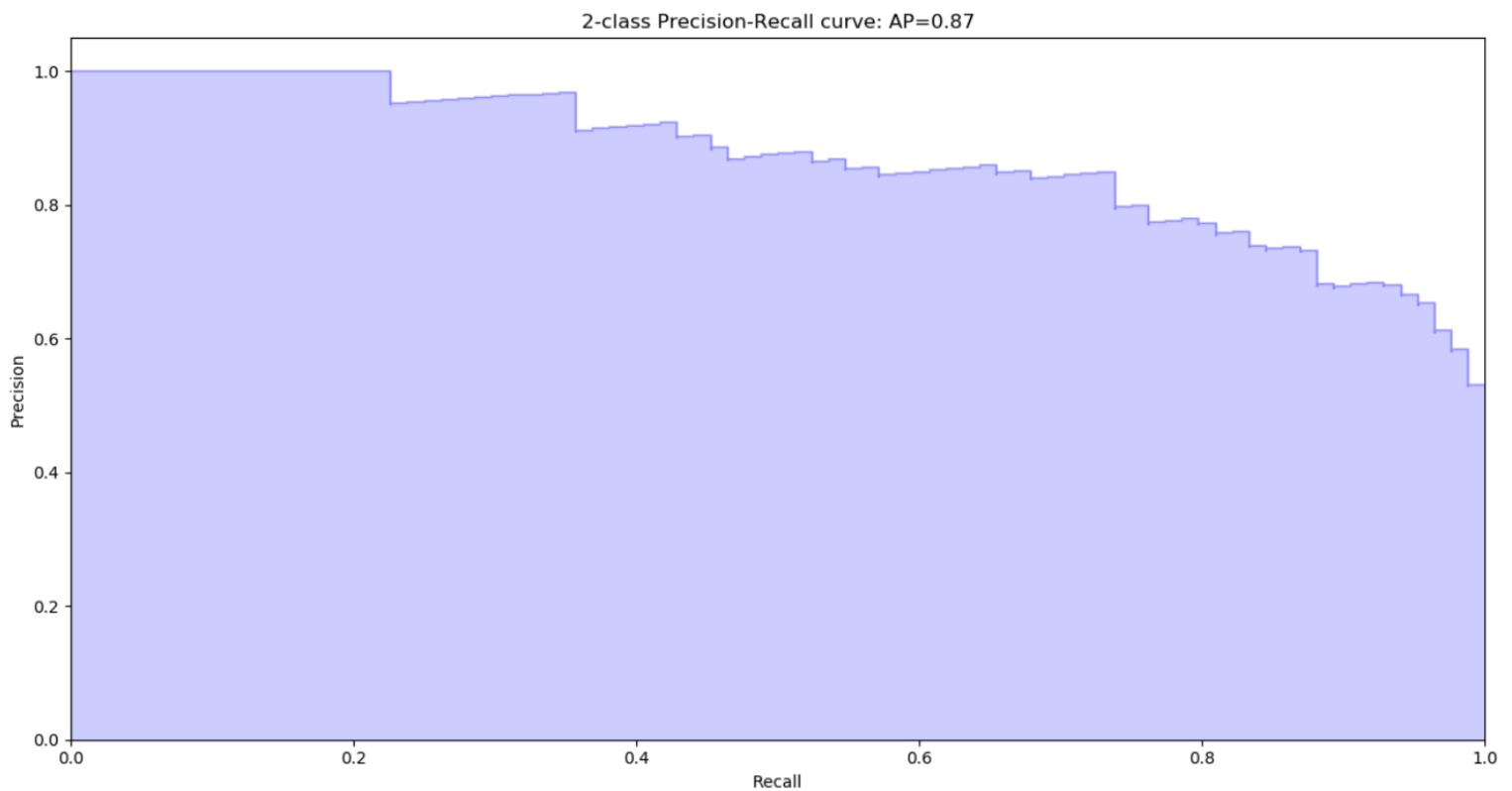
מטריצת המבוכה:



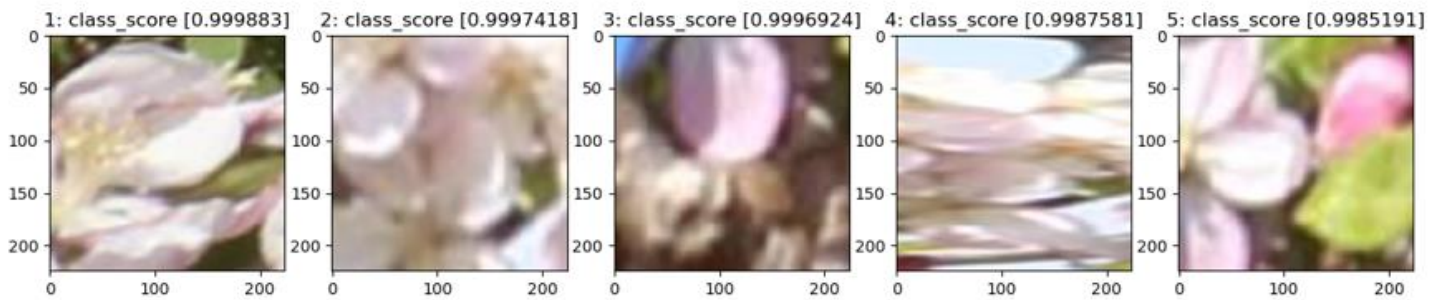
ניתוח תוצאות מטריצת המבוכה:

נשים לב כי המודל הנוכחי מבצע את כמות דומה של טעויות מסוג ראשון וסוג שני, כפי שצפינו ע"י העלאת נקודת הסף ל-0.7. בנוסף המודל חוזה בצורה יחסית מאוזנת כמות פרחים ולא פרחים בהתאם לחלוקה האמיתית של סט האימון.

Precision-Recall Curve:



5 התמונות (המקוריות, לא לאחר preprocess_input) עם הטעות מסוג ראשון החמורות ביותר מסודרות משמאל לימין. כלומר התמונות שמסווגות כלא פרחים, אך הרשת סיווגה כפרחים. ה-class score זה הערך שהרשת נתנה, ככל שקרוב יותר ל-1 הטעות גדולה יותר:



5 התמונות (המקוריות, לא לאחר preprocess_input) עם הטעות מסוג שני החמורות ביותר מסודרות משמאל לימין. כלומר התמונות שמסווגות כפרחים, אך הרשת סיווגה כלא פרחים. ה-class score זה הערך שהרשת נתנה, ככל שקרוב יותר ל-0 הטעות גדולה יותר:

