**Assignment 3**

* שיר כהן, 315805168
* עמית שקרצ'י, 313278889

# Data

*ה-data קיים* [*בקישור*](https://www.robots.ox.ac.uk/~vgg/data/flowers/102/)*.*

*ה-data מכיל 8189 קבצי תמונות של פרחים הנפוצים בבריטניה. כאשר יש 102 סיווגים שונים של סוגי פרחים.*

## SPLIT TRAIN, VALIDATION TEST

*ביצענו חלוקה באמצעות הפונקציה* [*train\_test\_split של sklearn.*](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html) *על מנת לבצע שתי חלוקות שונות הגדרנו seed שונה לביצוע החלוקה.*

|  |  |  |
| --- | --- | --- |
| ***Percentage of data*** | ***Number instance*** |  |
| *50%* | *4094* | ***Train*** |
| *25%* | *2047* | ***Validation*** |
| *25%* | *2048* | ***Test*** |

## Preprocessing

*תהליך ה-preprocessing לתמונות הוגדר בצורה הבאה –*

1. [***center crop (אופציונאלי)***](https://github.com/antonio-f/Inception-V3/blob/master/TF2_InceptionV3/InceptionV3_fine_tuning.ipynb) *– המטרה של הפונקציה היא למרכז את התמונה על מנת ל"הסיר" את הרעש מסביב.*

** ***לפני אחרי***

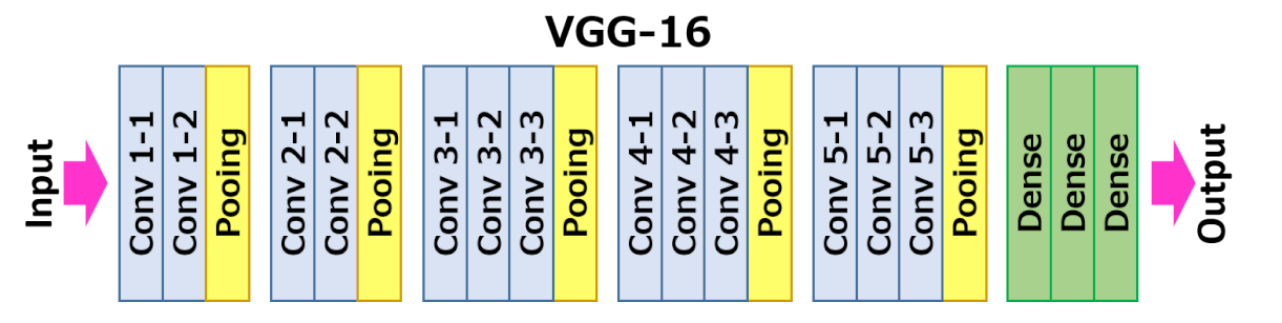
*כפי שניתן לראות התהליך מאפשר למקד את מרכז התמונה לפרח. הרעיון מאחורי ביצוע ה-preprocessing הזה שהפרח לרוב נמצא במרכז התמונה ולכן נרצה "לחדד" את החלק הזה בתמונה.*

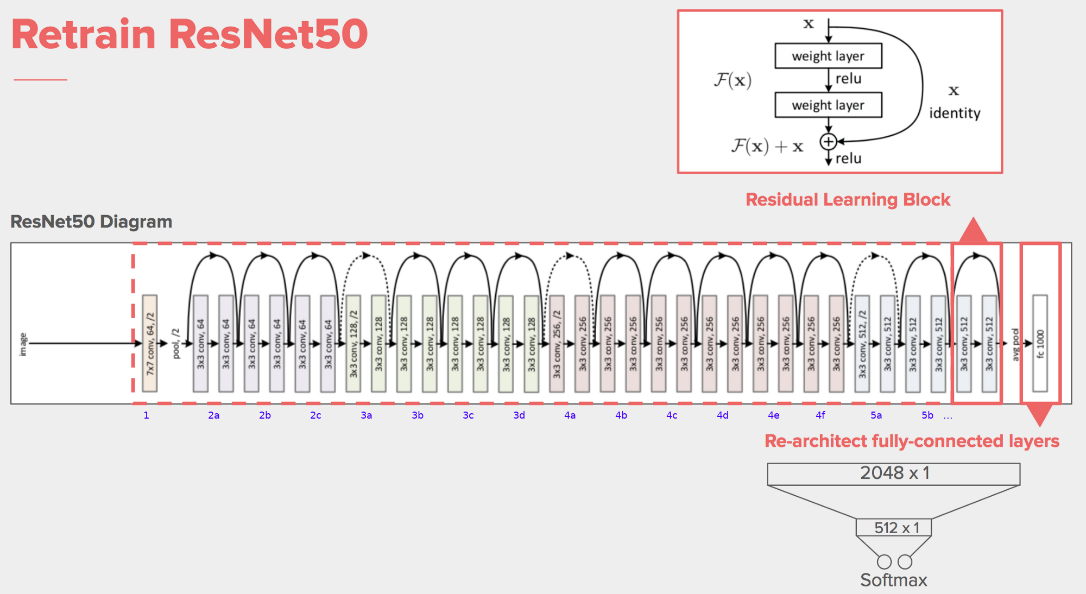
1. ***Resize*** *– שינוי גודל התמונה לגודל קבוע על מנת להתאים את הגודל לארכיטקטורה. הגודל הוגדר כ- (224, 224).*
2. ***נרמול (אופציונאלי)-*** *ביצענו נרמול על ידי חלוקה ב- 255 על מנת לייצר תמונה המכילה טווח ערכים של [0,1]. מטרת התהליך הנ"ל הוא לאפשר לרשת להתכנס בצורה מהירה יותר עם ערכים מנורמלים.*

*על החלקים בהם כתוב אופציונאלי ביצענו ניסויים על מנת לקבוע האם ביצוע ה-preprocessing הנ"ל עוזר לתהליך הלמידה.*

# MODELS

*ביצענו השוואה בין שלוש ארכיטקטורות שונות –*

1. ***VGG16*** *– המודל הוצע במאמר "* *Very Deep Convolutional Networks for Large-Scale Image Recognition". המודל אומן על ImageNet ומגיעה לאחוזי דיוק של 92.7%.*
2. ***MobileNet V2*** *– המימוש נלקח מהקישור* [*הבא*](https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4)*. המימוש מתבסס על המאמר "* "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation","*.*
3. ***ResNet50*** *– למודל יש 48 שכבות קונבולציה ו-maxPool אחרי כל שכבת קונבולוציה.*



*עבור כל ארכיטקטורה הסרנו את שכבת ה-output והוספנו שלוש שכבות נוספות –*

1. *Dense – עם כמות נוירונים משתנה לפי ניסויים.*
2. *Dropout – עם rate משנה לפי ניסויים.*
3. *Dense – output layer – עם כמות נוירונים של 102 (מספר ה-classes) ואקטיבציה של 'softmax'.*

הוספנו early stopping על מנת למנוע overfitting. ה-loss function '*categorical\_crossentropy'.*

# EXPERIMENTS

השתמשנו ב- [tensorflow\_hub](https://www.tensorflow.org/hub) על מנת להגדיר ארכיטקטורות רשת שונות. הערכים –

1. **Dense** – כמות נוירונים של 128, 256 ו- 1024.
2. **Dropout** – rate של 0 או 0.3. שכבת ה-Dropout תאפשר לנו שהרשת לא תסתמך על נוירונים ספציפיים בסיווג. כך נאפשר לרשת יותר robust עבור מצב שה-data מכיל רעשים או שגיאות.
3. **Optimizer** – 'adam' או 'sgd.

קיבלנו סך הכל 12 ארכיטקטורות שונות. וביצענו ניסויים עם שני seed שונים על מנת לאפשר חלוקה שונה של ה-dataset.

## Split 1

קבענו seed של 42 לחלוקה רנדומלית בצורה שונה של ה-train, validation ו- test.

**להלן התוצאות –**

הצגנו בטבלה את תוצאות הארכיטקטורה שהשיגה את ה-accuracy המקסימלי על ה-test set.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **# run** | **seed** | **model** | **normalization** | **crop** | **Parameter for architecture** | **Max Accuracy Test set** |
| 1 | 42 | feature\_vector | T | T |  |  |
| 2 | 42 | VGG16 | T | T |  |  |
| 3 | 42 | resnet | T | T |  |  |
| 4 | 42 | feature\_vector | F | T |  |  |
| 5 | 42 | VGG16 | F | T |  |  |
| 6 | 42 | resnet | F | T |  |  |
| 7 | 42 | feature\_vector | T | F |  |  |
| 8 | 42 | VGG16 | T | F |  |  |
| 9 | 42 | resnet | T | F |  |  |
| 10 | 42 | feature\_vector | F | F |  |  |
| 11 | 42 | VGG16 | F | F |  |  |

להלן הגרפים מסודרים משמאל לימין לפי מספר הריצה –

## Split 2

קבענו seed של 12 לחלוקה רנדומלית בצורה שונה של ה-train, validation ו- test.

**להלן התוצאות –**

הצגנו בטבלה את תוצאות הארכיטקטורה שהשיגה את ה-accuracy המקסימלי על ה-test set.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **# run** | **seed** | **model** | **normalization** | **crop** | **Parameter for architecture** | **Max Accuracy Test set** |
| 1 | 12 | feature\_vector | T | T |  |  |
| 2 | 12 | VGG16 | T | T |  |  |
| 3 | 12 | resnet | T | T |  |  |
| 4 | 12 | feature\_vector | F | T |  |  |
| 5 | 12 | VGG16 | F | T |  |  |
| 6 | 12 | resnet | F | T |  |  |
| 7 | 12 | feature\_vector | T | F |  |  |
| 8 | 12 | VGG16 | T | F |  |  |
| 9 | 12 | resnet | T | F |  |  |
| 10 | 12 | feature\_vector | F | F |  |  |
| 11 | 12 | VGG16 | F | F |  |  |

להלן הגרפים מסודרים משמאל לימין לפי מספר הריצה –

# Conclusion

**להלן המסקנות שלנו –**

1. השוואה בין המודלים –
2. השוואה בין החלוקה הרנדומלית בצורה שונה –
3. השפעת preprocessing –
   1. השפעת crop –
   2. השפעת normalization –

# SUMMARY

בעבודה זאת השתמשנו ברשתות קונבולוציה מאומנות וביצענו להן transfer learning על מנת להתאים אותם ל-dataset אחר. מצאנו את ההשפעה של תהליך preprocessing על התמונות וביצענו השוואה בין ארכיטקטורות שונות. כל אלה, הראו את ההשפעה הגדולה של פרמטרים שונים על תהליך הלמידה.