המעבדה לעיבוד גיאומטרי של תמונות
Geometric Image Processing Laboratory

GIP
Abiotic Stress Detection In Banana Plants

Itamar Gozlan
Supervisors: Alon Zvirin, Yaron Honen

Table of content

Contents

Abstract

The image processing capabilities are growing fast and bringing new possibilities.
seeking new challenges and applications to the rising technology, we wish to try and perform a task that now can be done only by a specialist.
The banana plants are very important and are a big part of the nutrition of many parts of the world.

More than 100 billion bananas are eaten every year in the world, making them the most popular agricultural product.
4 different qualities of treatment were applied and documented by a picture in a sequence of 17 days.

In the following project, we will try to distinguish between 4 kinds of Banana treatments by observing the pictures only.
We will use augmentations and introduce a novel specific data augmentation that is targeted for our dataset.
We will try to give our classifier an expert's capability since the differences are not notable and might not be observable to the common viewer.

1 Introduction

1.1 Goals and motivation

In this project, we wish to detect and classify banana plants that received different quality treatment.
The differences between the pictured banana plant categories, separated by the quality of treatment, are not easy to spot by a non-expert eye.
we are trying to achieve an expert's capability by training a Convolutional Neural Network (CNN).
To do so, we need to compare several convolutional neural network architectures and data inputs, analyze results and draw insights.
When finally, our main goal is to obtain a prediction tool that gives the possibility to distinguish between different banana plants that got different treatment.

1.2 The Dataset

There are 120 plants that suffer from the same disease.
the plants are getting 4 different treatments (A, B, C, D), there are 30 plants in each category.
Photographed daily, 17 days consecutively, 11-28 September 2018 (except 19/09) The images resolution is (4032 x 3024)
examples:

20/09/2018 - Treat A plant ID 01

21/09/2018 Treat B plant ID 02

15/09/2018 - Treat C plant ID 21

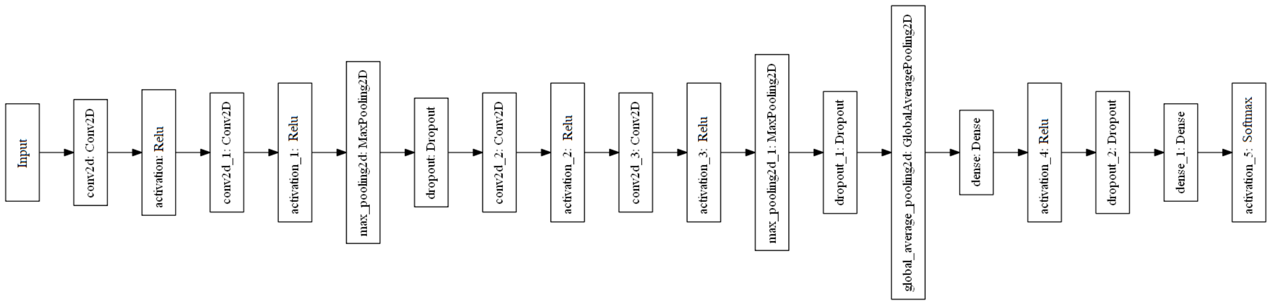
11/09/2018 - Treat D plant ID 04

1.3 CNN Architectures

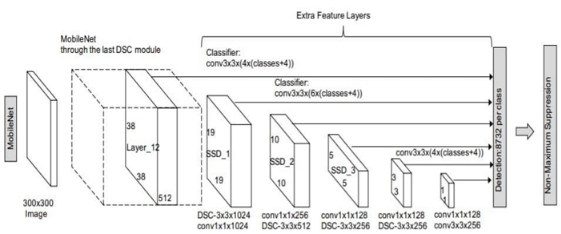
The native CNN we used is a CIFAR-10 based model.
The pre-trained networks we used are MobileNet, VGG16, GoogleNet, and ResNet50.
all are widely used in classification problems.

CNN Architectures plots (top models only)

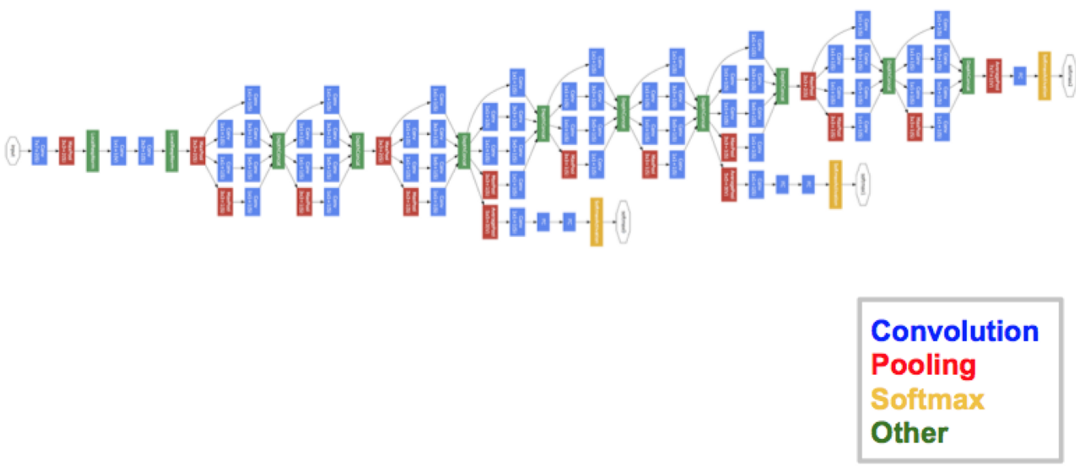
CIFAR-10 Based Model - By Keras visualize tool



MobileNet



GoogleNet



2 Experiment I - Full Pictures (with background)

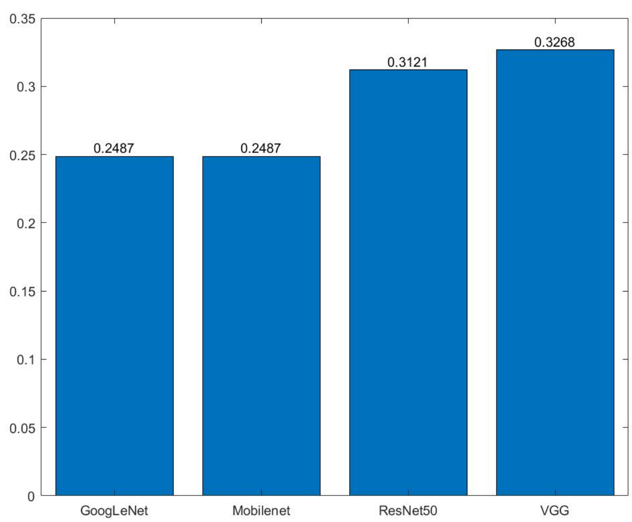
2.1 Experiment introduction

Since we aim to study what will be the best way to exploit our resources, we will divide the experiment into two main parts: the first part will use the Dataset as-is without any augmentations, which will also help us to detect the networks among the pre-trained networks are with the highest tendency to converge on our dataset. we will the networks for a short period of time and will see if improvement has occurred. afterward, we will take the lead networks (the ones who yields the best results) and will train them to a longer period. the second part will exploit the observation of the first part and will run the lead networks on the same dataset only this time we will use data-augmentations methods.

2.2 Experiment method

2.2.1 Choose the best networks

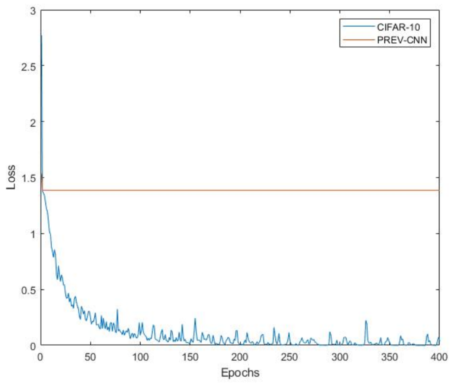
To narrow down the transferred learning networks we tried to detect which ones among the networks will have a better tendency to converge on our dataset. to do so, we ran all of the networks for 10 epochs, when epoch is one pass over the entire dataset a notation that is used to separate training into distinct phases.



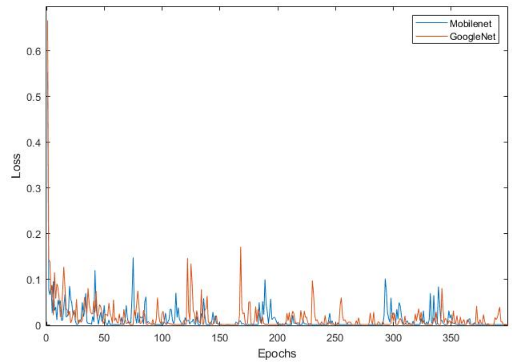
2.2.2 Continue with the best networks, train longer

as the previous part shows, the best pre-trained networks were GoogleNet and MobileNet. is this section we ran both of those networks for long periods of time, as well we used the native network (The CIFAR-10 based)

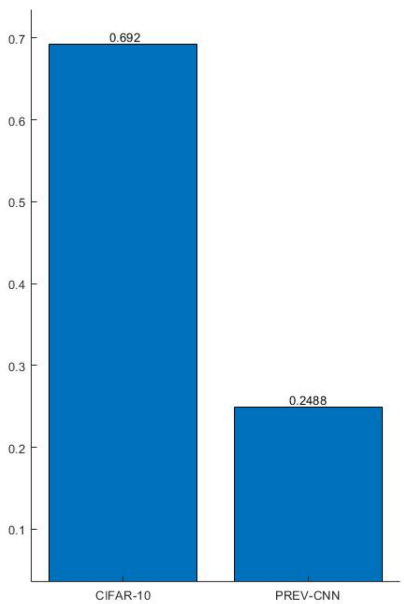
Native CNN

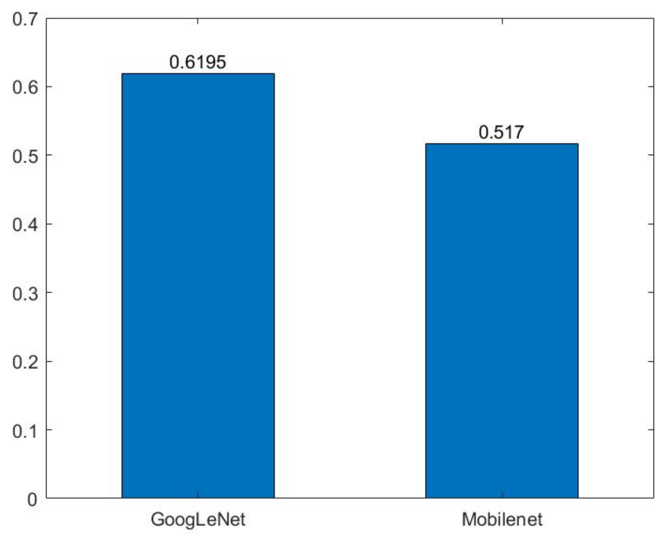


Tarnsfer Learning CNN's



Accuracy result:

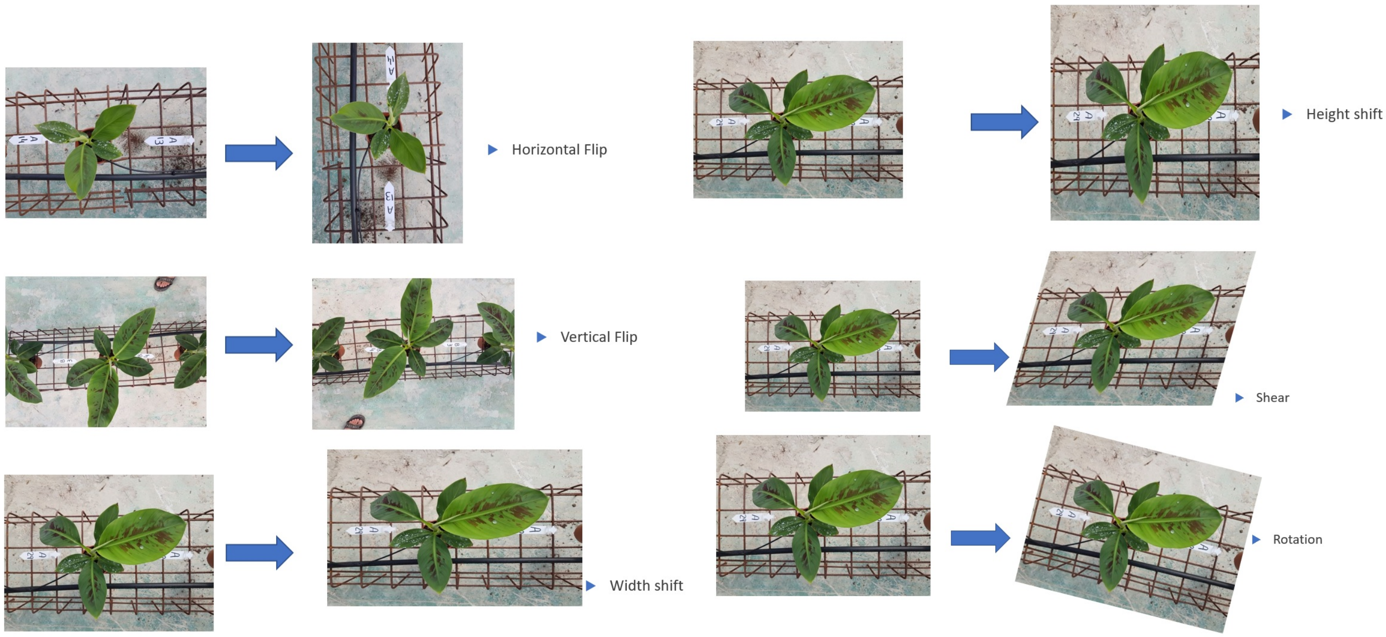




2.2.3 Using data augmentations

Data augmentation is a method that enables to significantly increase the diversity of data available for training models.
That without actually collecting new data.
the augmentations we used included Horizontal Flip, Vertical Flip, Width shift (X-scale), Height shift (Y-scale), Shear and Rotation.
This allowed us to learn from the same picture in more than one way.

Examples:

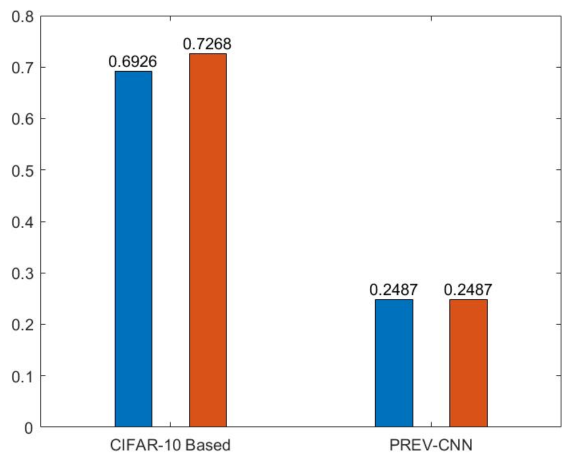


2.3 Results summary

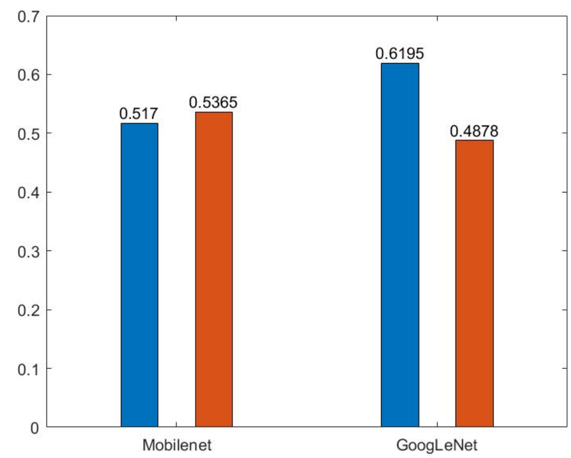
2.3.1 Accuracy result

Augmantation VS non-augmantation
Red - Augmentation
Blue - Non-Augmentation

Native CNN



Transfer Learning CNN



2.3.2 samples of test predictions (CIFAR-10 based)

Without augmentations:

samples of test predictions - True predictions

samples of test predictions - False predictions

With augmentations:

samples of test predictions - True predictions

samples of test predictions - False predictions

2.4 Main observations

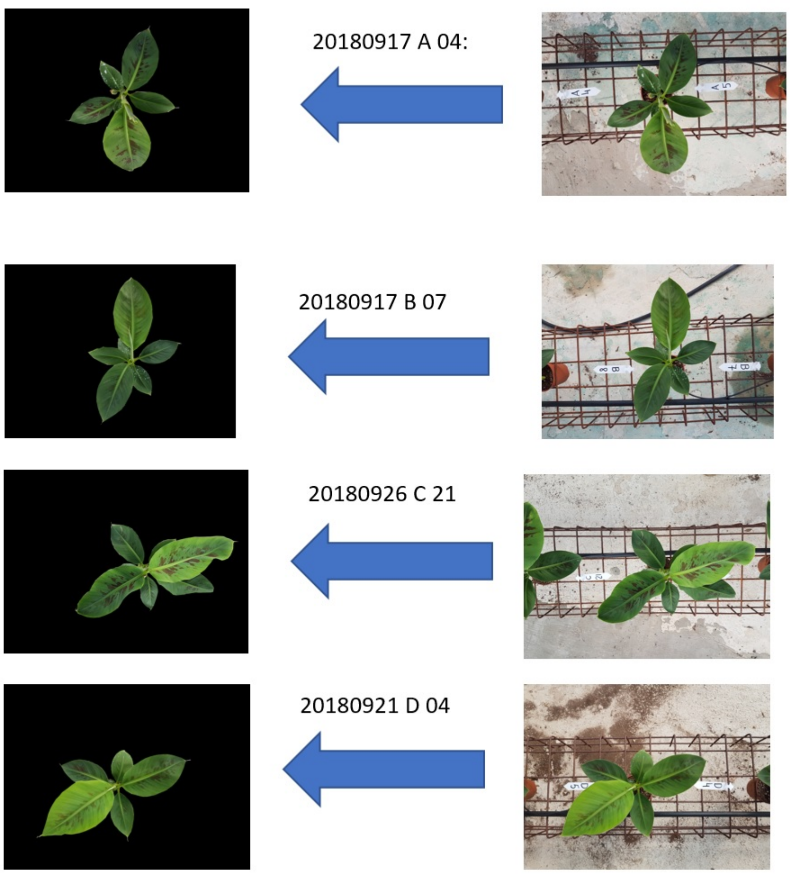
We can learn that there is a big weight in choosing the CNN architecture, while one architecture might be fairly successful another might fail. In the introduction we stated that the common observer might not see any differences between photos from different categories, but as we see the results are a strong indicator that a network can distinguish between the different photos. right now it's only an indicator since as the reader may observe from the data set examples, the tag (A/B/C/D) is present in the photos. In the next part, we will try and study on a different, segmented dataset that does not include the tags. another observation is that sometimes too much information might be confusing. GoogleNet performed better without augmentation and the Transfer learning networks yielded lower accuracy in total while the native CNN reached higher accuracy on our dataset although it was never trained before.

3 Experiment II - Segmented Pictures (No background)

3.1 Experiment introduction

In the previous part, we can approve that there is a notable difference between the plants and the network can distinguish between the different categories although it is no trivial for the common eye. Another thing is that adding relevant information (in the form of augmentations) is helpful to increase accuracy, but it still unclear how much effect the background has.
One of the major concerns was that because some of the labels are presented in the photo (and also showed in Part I dataset examples) it may have a large effect on the results, thus the network not actually learns from the plants themselves. In this part, we aim to answer some of those questions and we also arise some more questions.

Old data in a new representation.
The original pictures were cropped and left without background.
Examples:



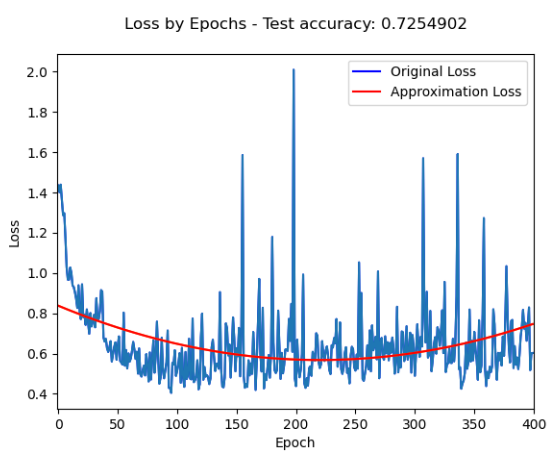
3.2 Experiment method

To examine the amount of effect of the background in this section we used the same methods from part I.
However, now we used only the top leading transfer-learning models.
The networks that were used are the CIFAR-10 based model (which introduced earlier), MobileNet and GoogleNet transferred learning.
Train and test were performed with and without augmentations on the CIFAR-10 based network alone.
The augmentations that were applied:
Horizontal Flip, Vertical Flip, Width shift (X-scale), Height shift (Y-scale), Shear and Rotation.
Another distinction is that now we didn't use a pre-train approach to explore the tendency to converge
on our dataset instead, we ran the network for a longer period, 400 epochs in the CIFAR-10 based network and 300 Epochs in the Transfer-Learning networks (MobileNet and GoogleNet each).

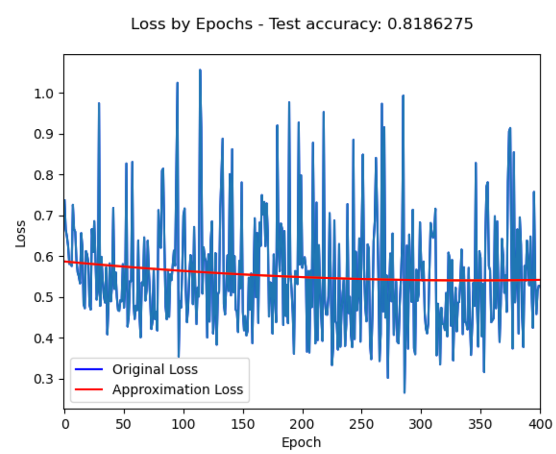
3.3 Results summary

3.3.1 Loss and accuracy

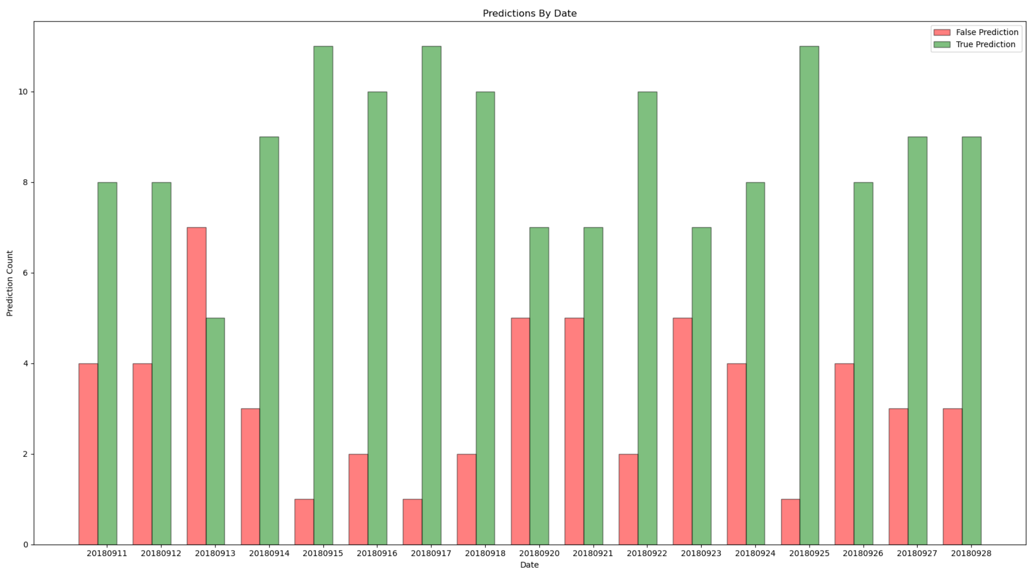
Without augmentations



With augmentations



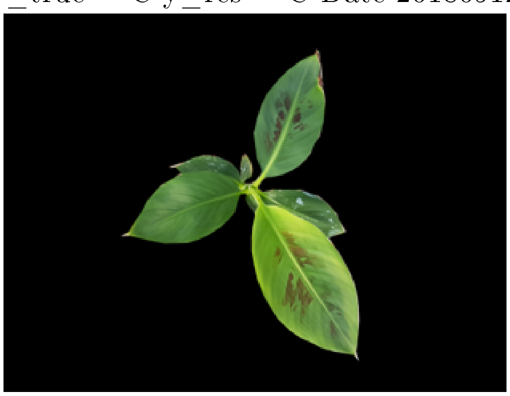
3.3.2 Prediction by day (CIFAR-10 based no augmentations)



3.3.3 samples of test predictions (CIFAR-10 based no augmentations)

True predictions

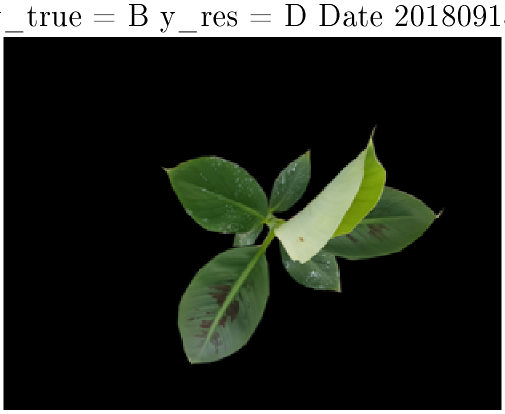
y_true = C y_res = C Date 20180912



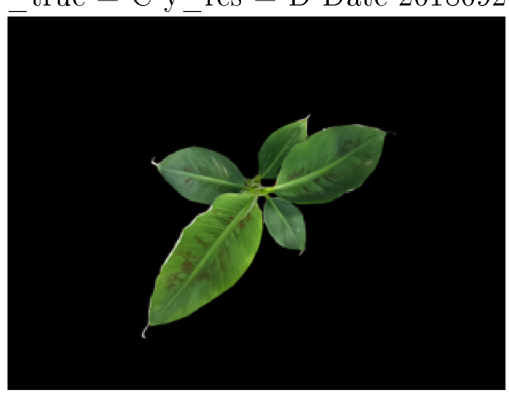
y_true = B y_res = B Date 20180916

False predictions:

y_true = B y_res = D Date 20180913



y_true = C y_res = D Date 20180920



3.4 Main observations

Based on the results, we can say in confidence that there is some difference between the different category treatments.
Another conclusion is that adding irrelevant information will result in low accuracy.
The transferred-learning networks almost fails.
But, additional relevant data, using augmentations on a network that wasn't trained before improves accuracy.
We can also point out the background interrupts, without the background, on a native (pre-trained) network the accuracy is higher.
With the background, the results of the Transferred learning networks are better – that in a way is sanity check because we expect transferred-learning networks to handle a lot of details better.

4 Experiment III - Following a hint

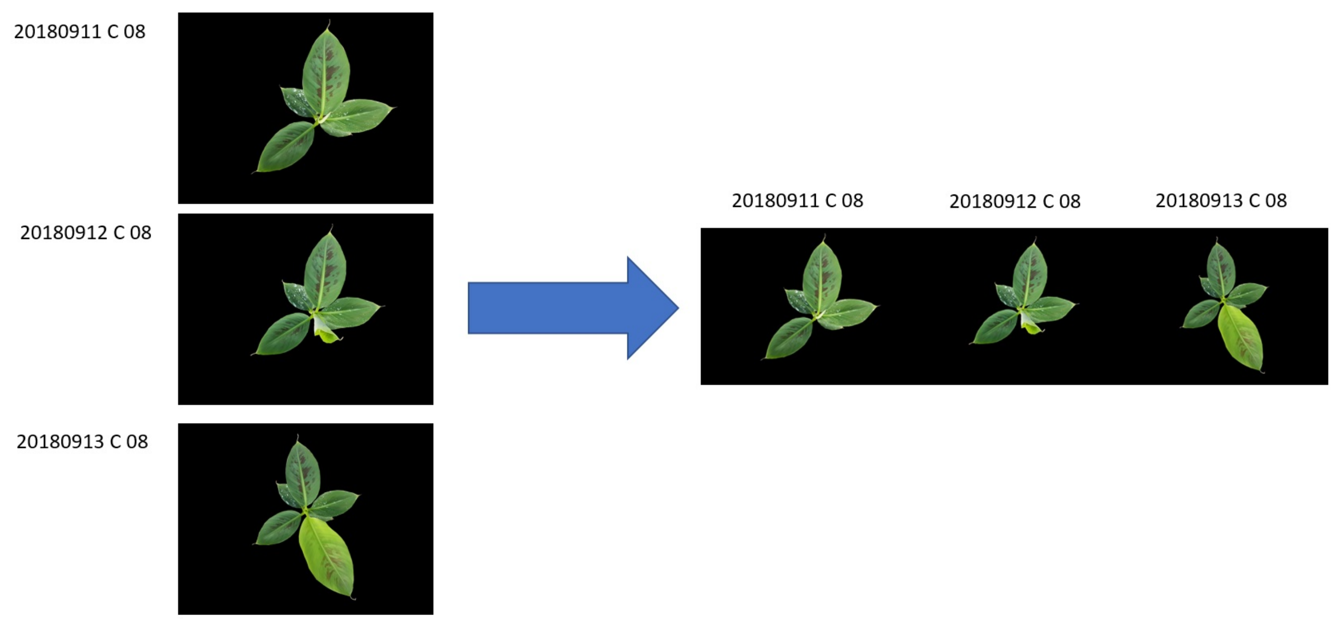
4.1 Experiment introduction

4.1.1 The hint

According to the experts, one of the main differences in the plant reaction to the different treatments
is the growth rate and the number of leaves that are emerging at the same time interval.
To exploit this piece of information, hoping to get better accuracy we decide to
form a "new" augmentation, thus forming new data from the pictures we have.

4.1.2 Dataset specific data augmentation

Previously it was mentioned that the plants were photographed daily, 17 days consecutively.
Forming one picture from every 3 sequential pictures.
This approach aims to benefit from the time connection between the pictures.
Forming triplets from sequential days to create “new” data as a form of augmentation.
Every 3 following pictures from every plant were made to triplet.
For example (presented with original name):
53722_20180911_153147_RGB_Treat_A_04.jpg
53717_20180912_160043_RGB_Treat_A_04.jpg
53721_20180913_151116_RGB_Treat_A_04.jpg
will result:
53722_53717_53721_20180911_20180912_20180913_RGB_Treat_A_04.jpg
The dataset was separated to train\test\validation, plants 05,15,25 were picked as validation group (as triplets).
the other plants were randomly distributed to train and test with the ratio of 0.2 test,0.8 train.

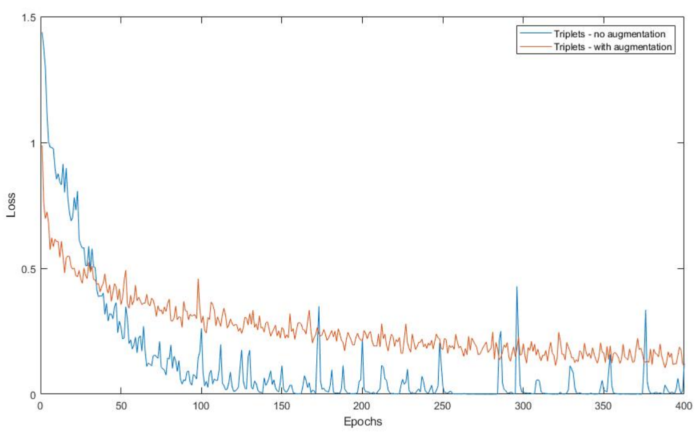


4.2 Experiment method

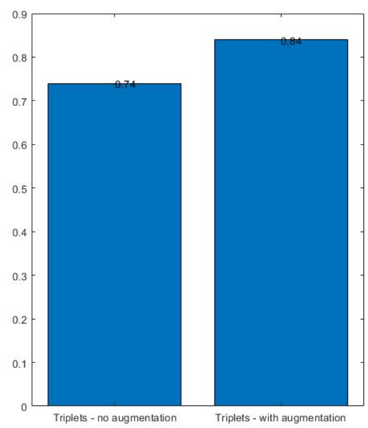
Using plants as triplets, train the same Network from the previous experiment (CIFAR-10 Based) and
the top Transfer Learning networks from before (MobileNet, GoogleNet).
using Keras built-in data augmentation and also try running without any augmentations.
The same augmentation as before, without horizontal flip to maintain the order
Transfer-Learning networks executed 300 epochs while the CIFAR-10 based network was executing 400 epochs.

4.3 Results summary

CIFAR-10 Based loss



CIFAR-10 based accuracy



CIFAR-10 Without augmentation	74%
CIFAR-10 with augmentation	**84%**
MobileNet without augmentation	25%
MobileNet with augmentation	32%
GoogleNet without augmentation	34%
GoogleNet with augmentation	25%

4.4 Main observations

Improvement can be acquired by exploiting the sequential connection, following the expert's hint was worth, again we showed that additional relevant data, using augmentations on a network that wasn't trained before improves accuracy.
And exploiting the connection between the pictures may increase the accuracy significantly.

5 Experiment IV - Compare A with the rest

5.1 Experiment introduction

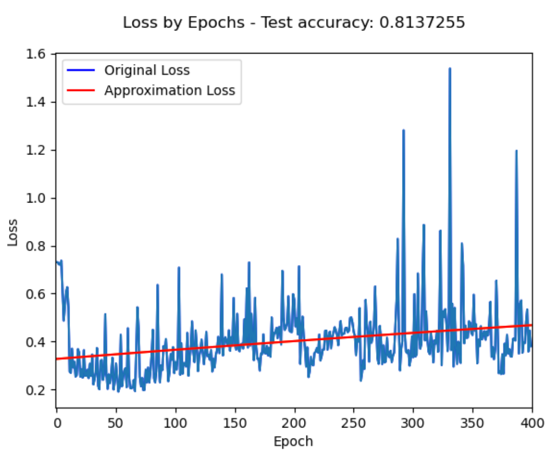
So far we only focused on 4 category classification, in this part, we aim to try a different approach. Since the dataset is on a scale from A as the best quality treatment to B, C, D in decreasing manner we might have insights from this sort of experiment. We ask about the CIFAR-10 flexibility in way of category count and match to our dataset, we will also explore the by-day prediction.

5.2 Experiment method

5.2.1 A vs ALL and A vs B/C/D (each)

First, we will try to train and predict by a given banana plant picture if it got an A treatment or other (B, C, D).
We will divide our dataset to train/test and validation when validation will hold plants with ID 05,15,25.
That is to avoid any bias that might be given as a result of the similarity between the same plant in train\test and validation.
For the "Each" experiment we will choose only A and the single category data.
For the non-A category we picked randomly 0.33 from each B/C/D respectively.
The CIFAR-10 based model will be trained for 400 epochs.

5.3 A vs ALL - Results summary



True prediction

Date 23/09/2018 Treatment C plant ID 05

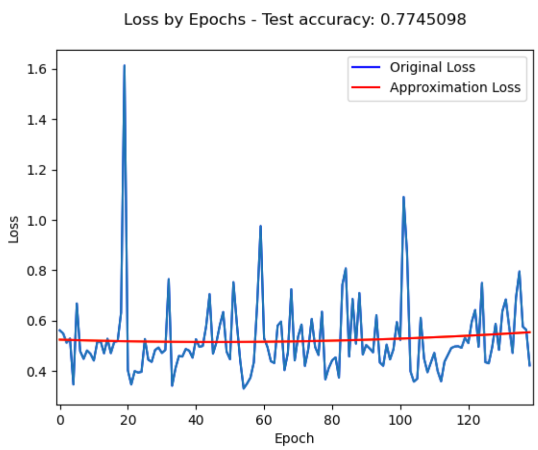
False prediction

Date 21/09/2018nTreatment B plant ID 15

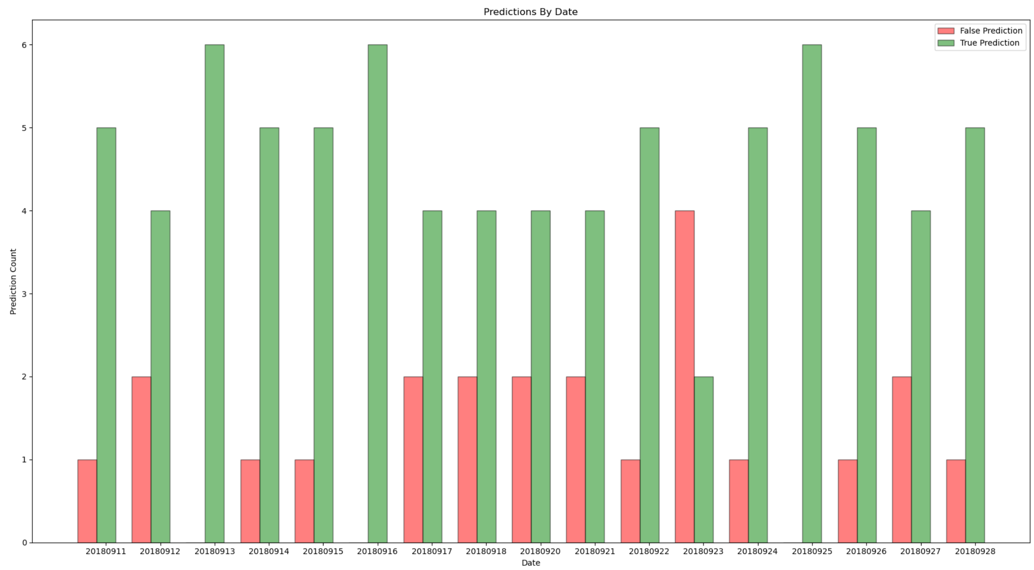
5.4 A vs Each (B/C/D) - Results summary

5.4.1 A vs B

Loss and accuracy:



Predictions by day:



True prediction

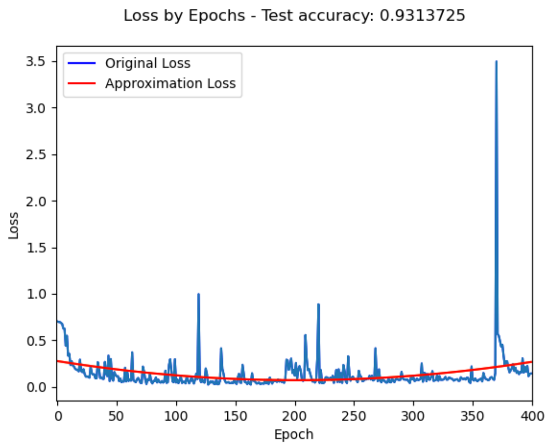
Date: 13/09/2018 Treatment B plant ID 05

False prediction

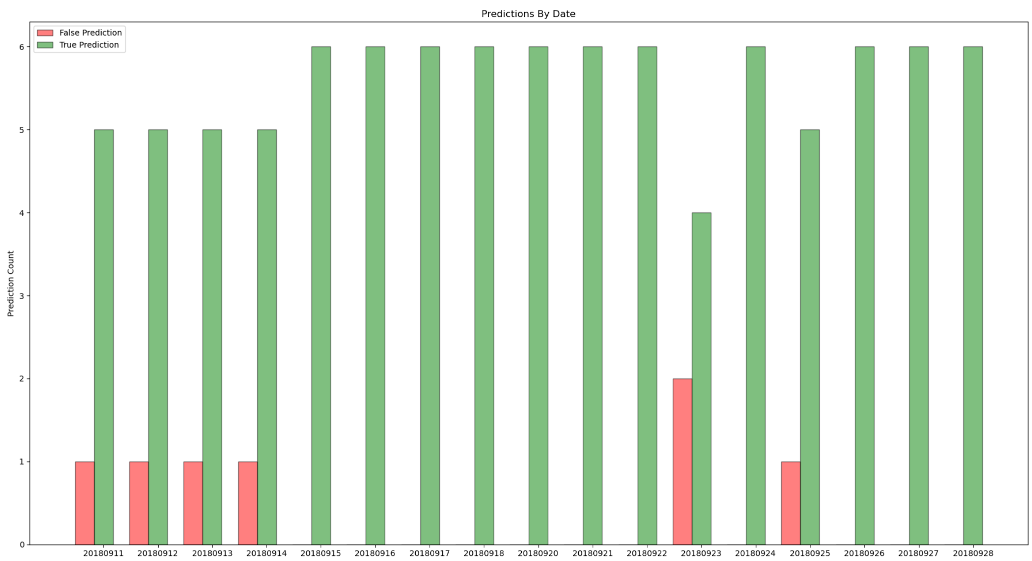
Date: 17/09/2018 Treatment A plant ID 05

5.4.2 A vs C

Loss and accuracy:



Predictions by day:



True prediction

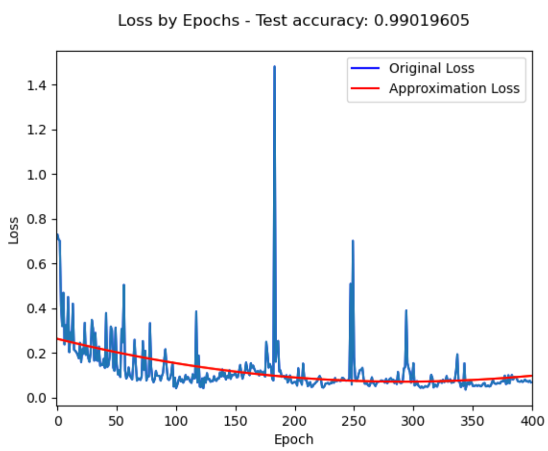
Date 28/09/2018 Treatment A plant ID 05

False prediction

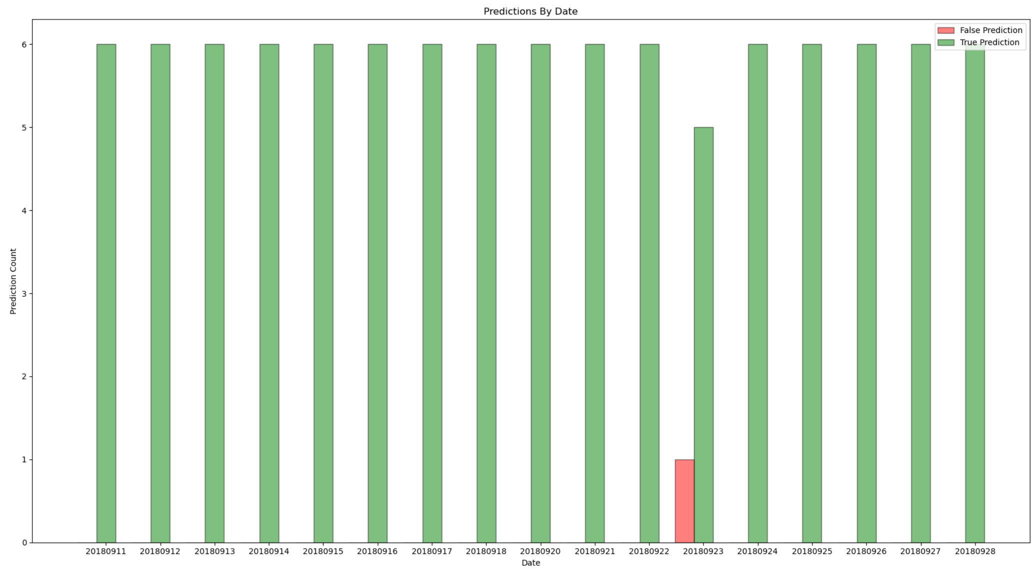
Date: 12/09/2018 Treatment A plant ID 25

5.4.3 A vs D

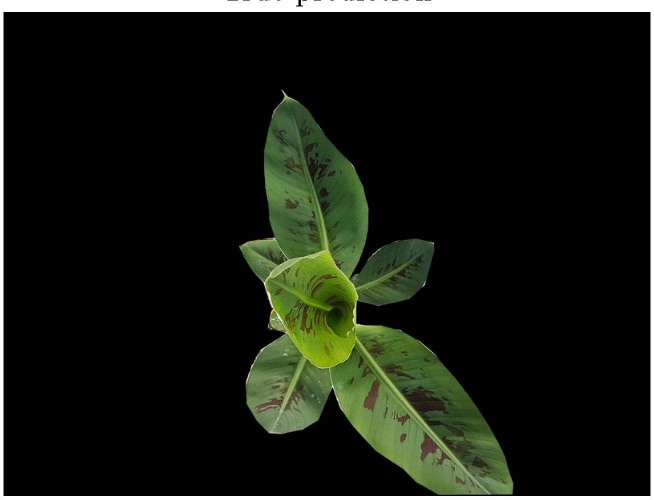
Loss and accuracy:



Predictions by day:



True prediction



Date: 26/09/2018 Treatment A plant ID 25

False prediction

Date: 23/09/2018 Treatment A plant ID 25

5.5 Result table

A vs. ALL	A vs. B	A vs. C	A vs. D
81.3%	77.45%	93.13%	99%

5.6 Main observations

The result indicated strong differences among the categories although it does not obvious to the non-expert eye.
The treatment scale is well expressed in the different network results and they are like we expected.
We can also conclude that the CIFAR-10 based architecture is well suitable for this dataset, also in 2 category detection.
Also, there is a higher accuracy in the prediction around 14/09/2019 and 24/04/2018 but not in a notable way.

6 Summary

6.1 Project summary

by only looking at the pictures, two plants that got different treatments might not express highly notable features differences.
But, the network is capable to distinguish and determine by the picture alone what treatment each plant got in a fairly hight accuracy that is far from random.
In the first part, we focused on the original photos that included the background and we yielded 72% accuracy with CIFAR-10 based architecture and augmentations.
The second part was about examing the specific effect of the background, we repeated the experiment with segmented pictures that only contain the plant themselves without background. All the following experiments focused only on this dataset.
we conclude that the background only interrupts and effect badly on the accuracy. we yield 81% accuracy with the CIFAR-10 based CNN, with augmentations.
In the third part, we introduce a dataset-specific augmentation, exploiting the sequential connection between the pictures and forming "triplets" from sequential pictures of the following days. we yield 84% accuracy with the CIFAR-10 based CNN.
In the last part, we try a different approach and try to do a 2 category classification. only with the CIFAR-10 based architecture.
First, we try A vs the rest (A/B/C/D) and yield 81.3% accuracy.
Later, we experiment A vs Each of the categories and yield 77.45% in A vs B, 93.13 in A vs. C and finally in A vs D we yield 99% accuracy.

6.2 Main conclusions

There are three main conclusions from this project.
First, although the differences are not notable by a non-expert viewer, the Network can detect and identify in a rather good each of the treatment quality that the plant got, by picture only. One conclusion is that CNN can develop a specific task specialty of an expert.
Another conclusion is specific about the CIFAR-10 based architecture, In all of the experiments and all of the networks we tried in the first part of the project the CIFAR-10 based network performed the best. that might indicate that it might also succeed in detection experiments on a dataset that has any close relation to our dataset.
The last conclusion is about the dataset specific augmentation. Identifying the connection and following the expert's hint led to the best results. From this, we can expect that this method might serve in all the experiments with datasets that have any sequential connection between them.

7 References

7.1 Libraries and technologies

Tensorflow - Library for dataflow and differentiable programming
Keras - neural-network library written in Python, running on top of TensorFlow
Matplotlib - plotting library for the Python programming language
Matlab - plotting of functions and data

7.2 Convolutional neural networks

CIFAR-10 model
MobileNet
VGG16
GoogleNet
ResNet50