Healthy vs. Unhealthy Plant Classification

AI4ALL Group 3

Table of Contents

01

Introduction

Our Group

Group Mentor: Chuer Pan

Group Members: Ariana Devito, Kenan Erol, Maigan Lafontant, and Luis Pabon

What Problem did we Address?

According to a UN report, the World must sustainably produce 70% more food by the middle of the century

- In an effort to mitigate this problem, our group decided to make an AI that distinguishes between healthy and unhealthy plants.

- This early detection of plant diseases can prevent plant loss; therefore, contributing to the optimization of agricultural procedures.

02
Data Augmentation

What happens when we do not have enough Training Data?

- Inaccurate Results
- Biases
- Model is less prepared for Test Data

Data Augmentation: the solution to your problems!

- Data Augmentation grows data
- Creates new images based on the present data set
- Various techniques to make unique images

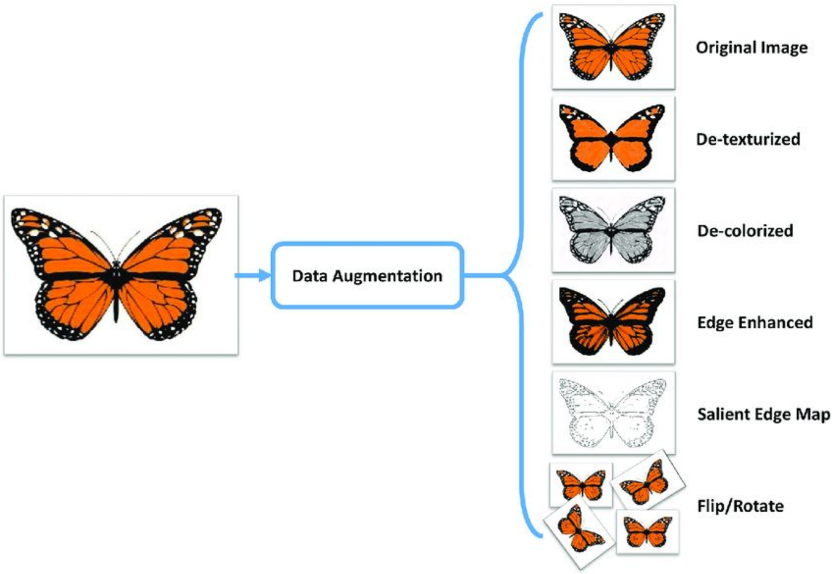
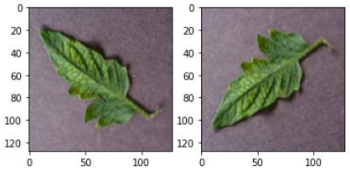
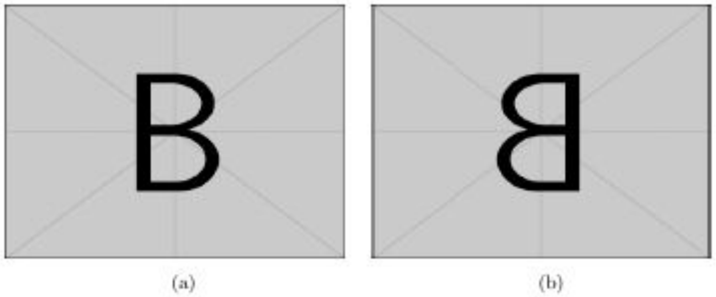


Image Flip

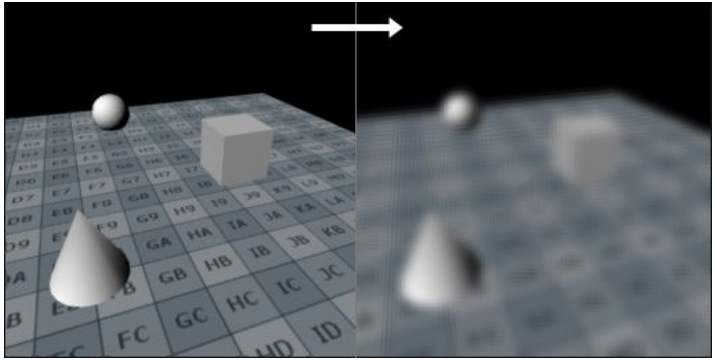


Vertical Flip

Horizontal Flip

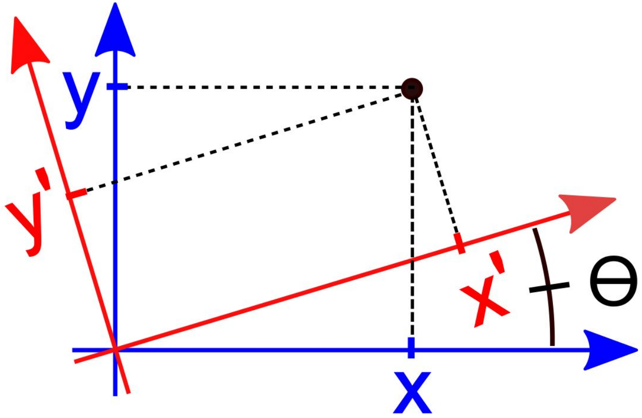


More Data Augmentation Techniques



Blur

Rotation



Google Colaboratory: Limitations

- Our use of Data Augmentation for more training data
- Randomized options
- Adding new images and matching labels
- RAM and Google Colab Limitations

```
options = [0, 1, 2, 3, 4]
version = np.random.choice(options,2)
final_images.append(image)
final_labels.append(label)
if 1 in version:
    #flip image vertically
    flipped_image = np.flip(image, axis=0)
    final_images.append(flipped_image)
    final_labels.append(label)
if 2 in version:
    #flip image horizontally
    flipped_image = np.flip(image, axis=1)
    final_images.append(flipped_image)
    final_labels.append(label)
if 3 in version:
    #blur image
    blurred_image = gaussian_filter(image, sigma=0.5)
    final_images.append(blurred_image)
    final_labels.append(label)
if 4 in version:
    #rotate image
    angle = np.random.rand() * 30
    rotated_image = rotate(image, angle, mode='nearest')
    rotated_image_resized = cv2.resize(np.array(rotated_image), image.shape[:2])
    final_images.append(rotated_image_resized)
    final_labels.append(label)
```

Runtime disconnected

The connection to the runtime has timed out.

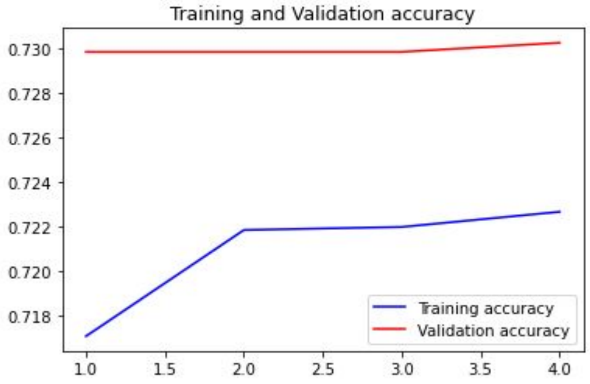
CLOSE RECONNECT

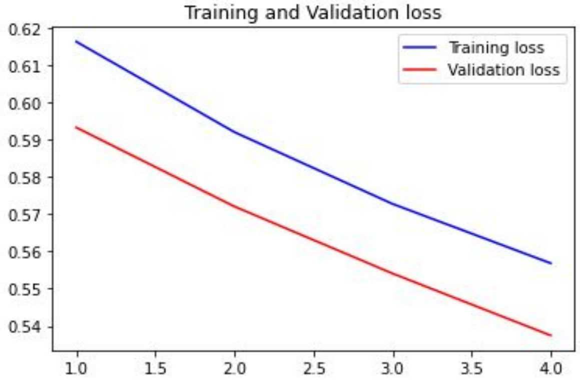
03
Hyperparameter Tuning

Baseline Values

Number of Epochs: 4
Batch Size (BS): 16
Dimensions: 128 x 128
Depth: 3
Learning Rate: 1e-6

Test Accuracy:
73.02497029304504

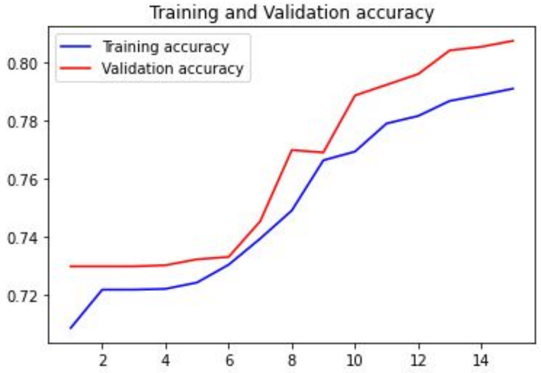


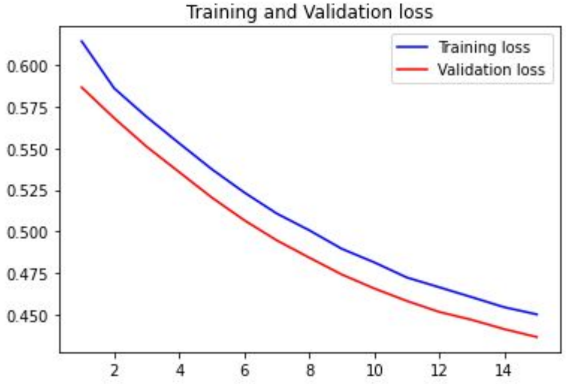


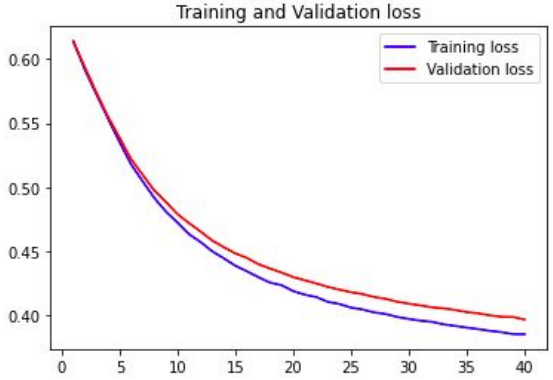
Increasing Number of Epochs

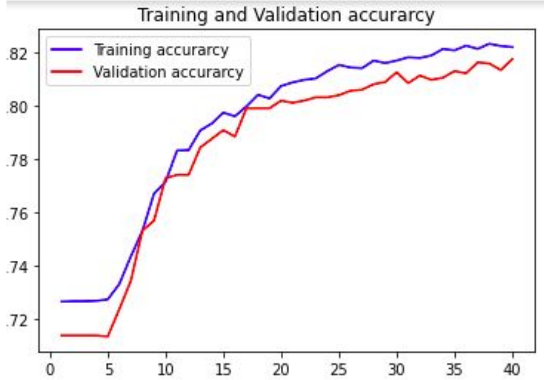
Number of Epochs: 4 → 15

Test Accuracy: 80.7613







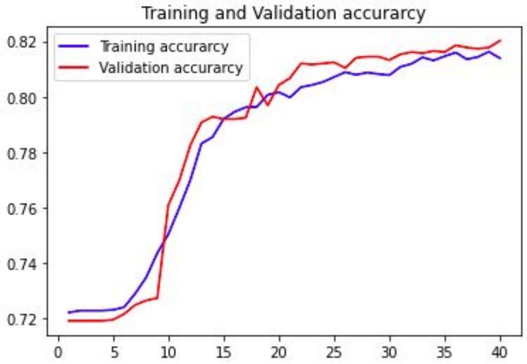


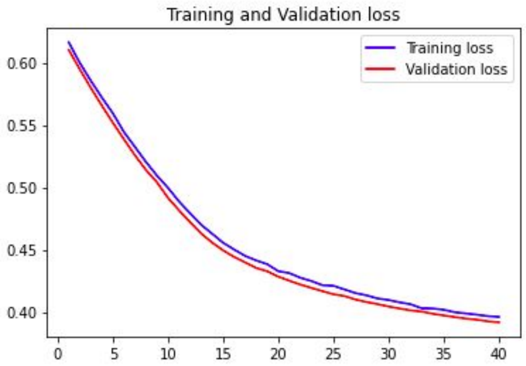
Number of Epochs: 15 → 30

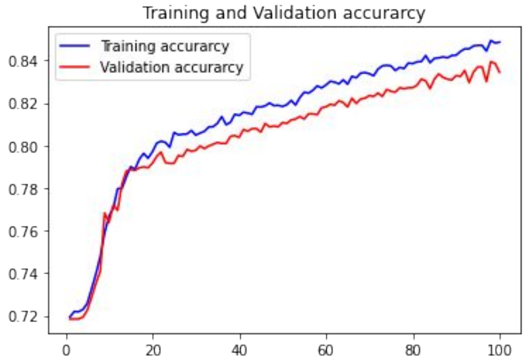
Test Accuracy: 80.6636

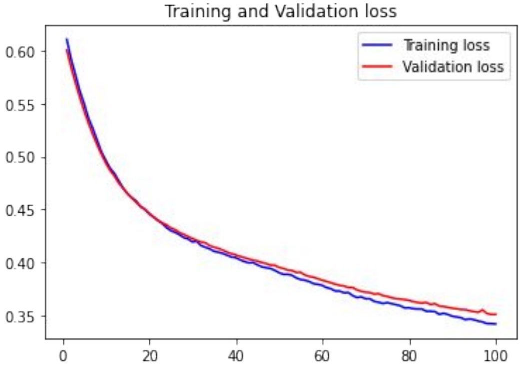
Number of Epochs: 30 → 40

Test Accuracy: 82.0187









Overfitting

Number of Epochs: 40 → 100

Test Accuracy: 83.44911

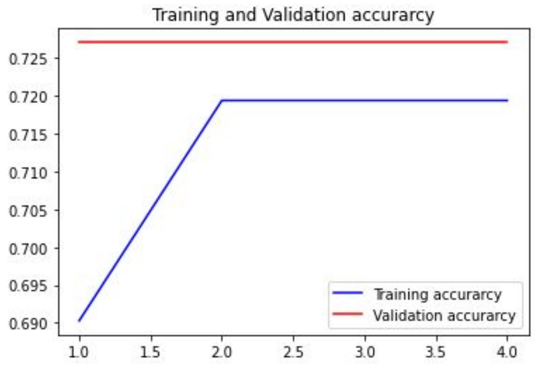
Epoch Summary

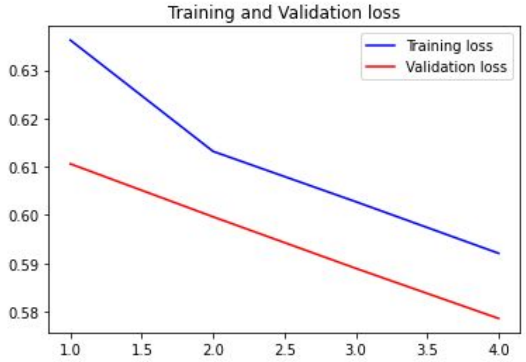
Epochs	Test Accuracy
4	73.02%
15	80.76%
30	80.66%
40	82.01%
100	83.44%

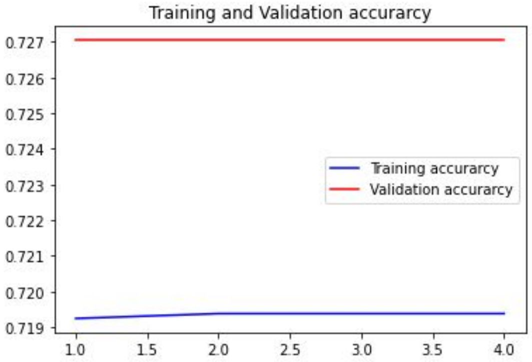
Increasing Batch Size

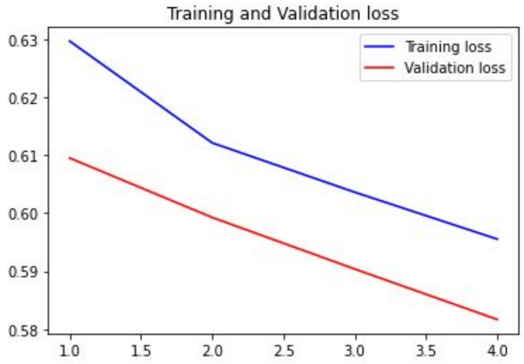
Number of Epochs: 4
Batch Size (BS): 16 → 32
Dimensions: 128 x 128
Depth: 3
Learning Rate: 1e-6

Test Accuracy: 72.7049









Number of Epochs: 4
Batch Size (BS): 32 → 64
Dimensions: 128 x 128
Depth: 3
Learning Rate: 1e-6

Test Accuracy: 72.7049

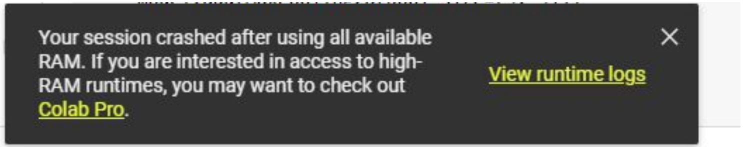
Increasing Batch Size (cont.)

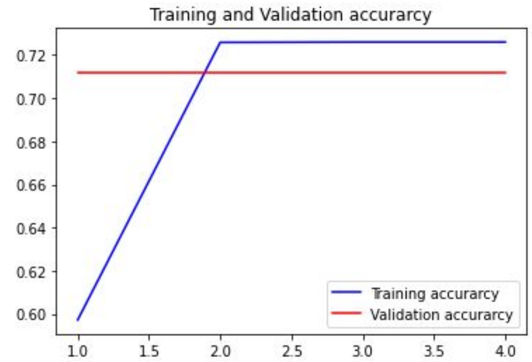
Number of Epochs: 4
Batch Size (BS): 64 → 128
Dimensions: 128 x 128
Depth: 3
Learning Rate: 1e-6

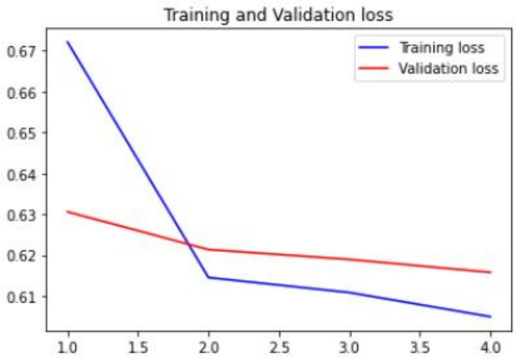
Test Accuracy: 71.1961

```
EPOCHS = 4
BS = 256
width=128
height=128
depth=3
INIT_LR = 1e-6
```

```
EPOCHS = 4
BS = 512
width=128
height=128
depth=3
INIT_LR = 1e-6
```







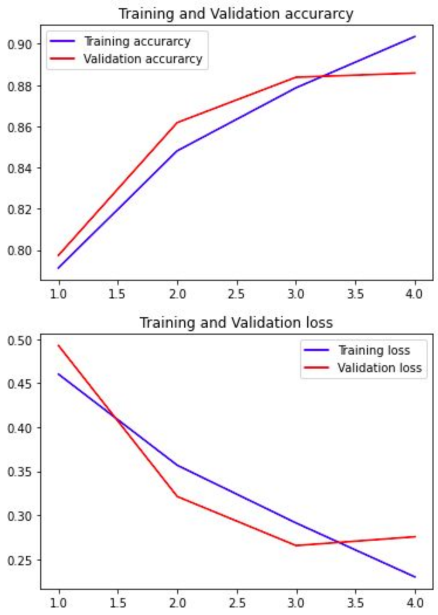
Batch Size Summary

Batch Size	Test Accuracy
16	73.02%
32	72.70%
64	72.70%
128	71.20%
256	ERROR
512	ERROR

Adjusting Learning Rate

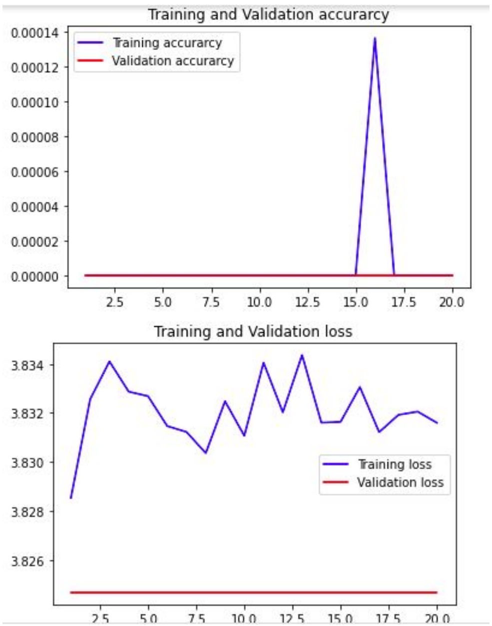
Learning Rate: 1e-3

Test Accuracy: 88.5854



Learning Rate:1e-15

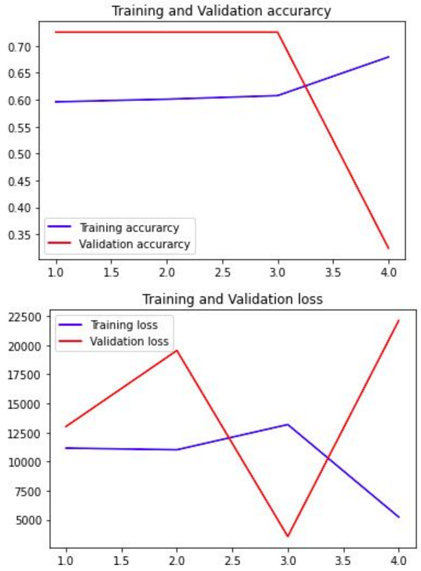
Test Accuracy: 0.0



Adjusting Learning Rate (cont.)

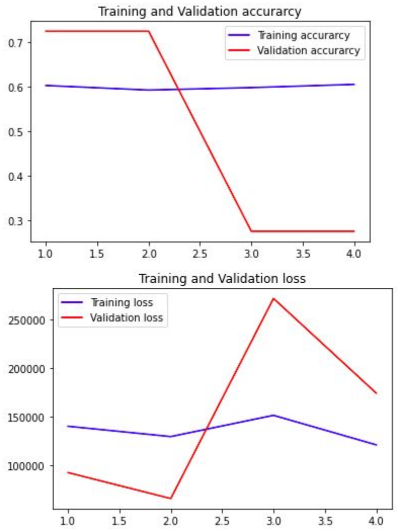
Learning Rate: 1

Test Accuracy: 32.4093



Learning Rate: 10

Test Accuracy: 27.4358



Learning Rate Summary

Learning Rate	Test Accuracy
10	27.44%
1	32.41%
1e-3	88.59%
1e-6	73.02%
1e-15	0.0%

Observations

Number of Epochs

Increasing the number of epochs increases the accuracy because we are training the model for longer. Yet, too many epochs would lead to a divergence in optimization and overfitting.

Batch Size

Increasing the batch size increases the test accuracy and speeds up training. However, if the batch size is too large, it can have the opposite effect.

Learning Rate

Increasing the learning rate increases the accuracy and decreases the training time. Again, when the value is either too small or too large, that accuracy will be compromised.

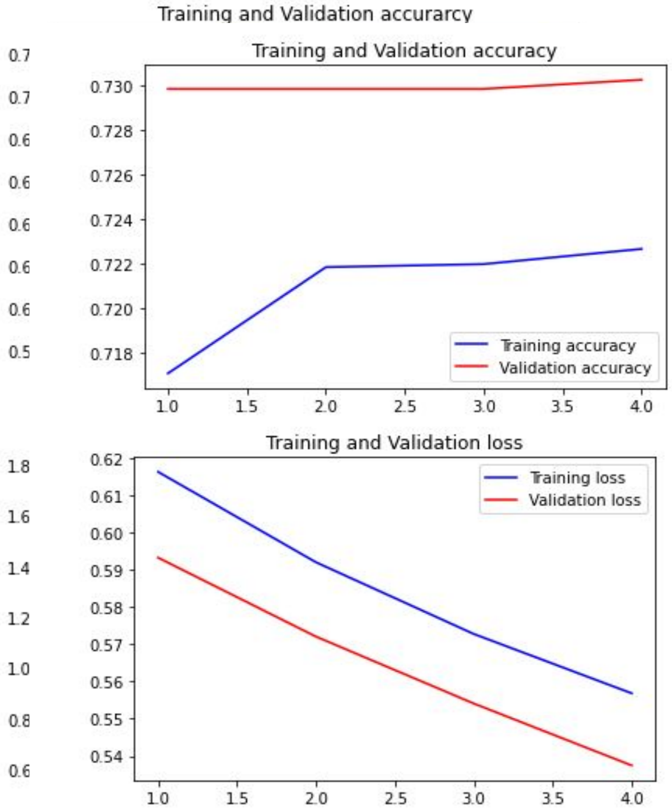
04

Activation Functions

Adjusting the Activation Functions

```
EPOCHS = 4
BS = 16
width=128
height=128
depth=3
INIT_LR = 1e-6


model = Sequential()
inputShape = (height, width, depth)
chanDim = -1

model.add(Conv2D(64, (3, 3), padding="same", input_shape=inputShape))
model.add(Activation("relu "))

model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(n_classes))
model.add(Activation("softmax"))
```

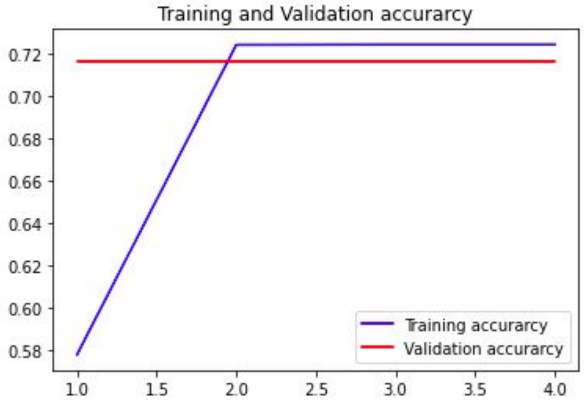
Test Accuracy: 73.02497029304504

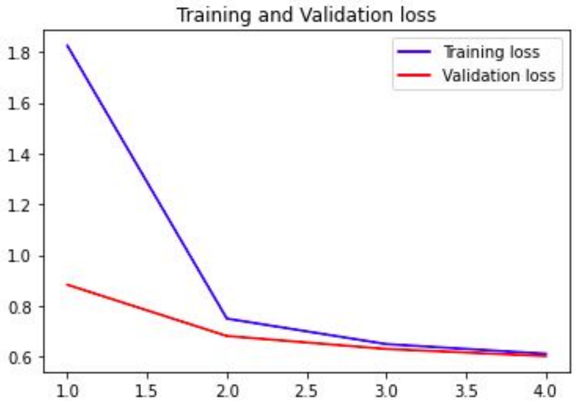


Adjusting the Activation Functions

```
EPOCHS = 4
BS = 16
width=128
height=128
depth=3
INIT_LR = 1e-6


model = Sequential()
inputShape = (height, width, depth)
chanDim = -1

model.add(Conv2D(64, (3, 3), padding="same", input_shape=inputShape))
model.add(Activation("sigmoid"))

model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(n_classes))
model.add(Activation("softmax"))
```

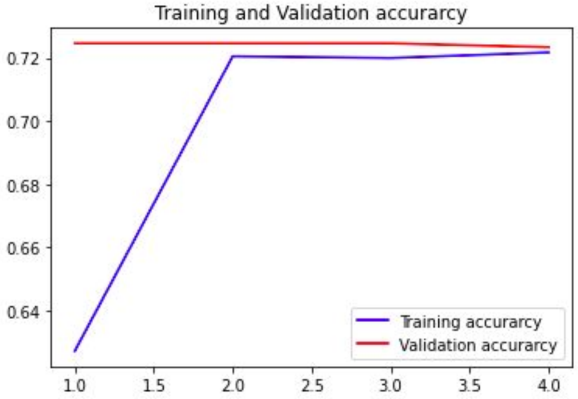
Test Accuracy: 71.63206934928894

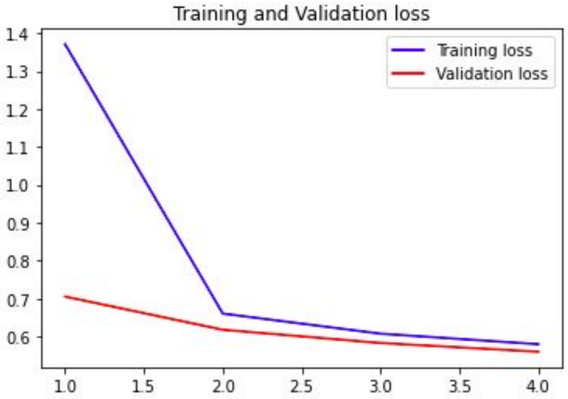




Adjusting the Activation Functions

```
EPOCHS = 4
BS = 16
width=128
height=128
depth=3
INIT_LR = 1e-6


model = Sequential()
inputShape = (height, width, depth)
chanDim = -1

model.add(Conv2D(64, (3, 3), padding="same",
input_shape=inputShape))
model.add(Activation("tanh"))

model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Dropout(0.25))

model.add(Flatten())

model.add(Dense(n_classes))
model.add(Activation("softmax"))
```





Test Accuracy: `72.33782410621643`

05

Transfer Learning

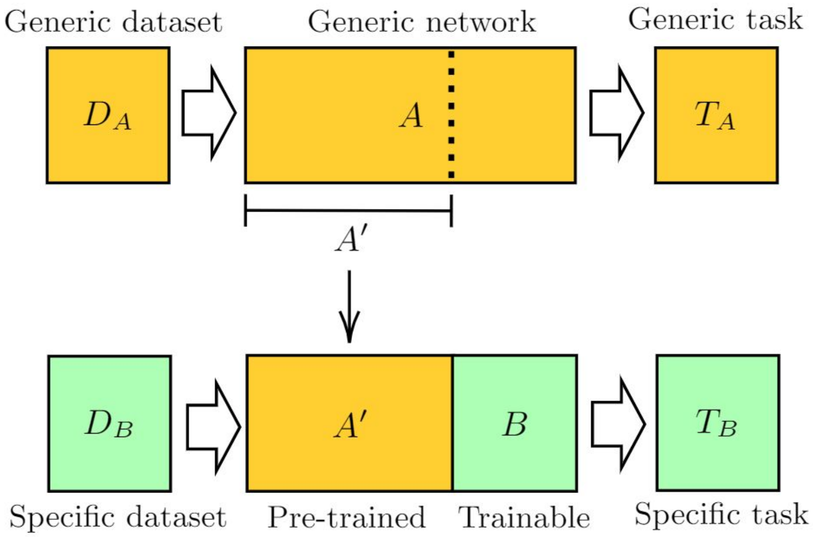
What is Transfer Learning?

Transfer Learning is a method that allows us to reuse developed models.

- This increases our accuracy
- Better performance
- We can utilize big pre-existing data pools

How did we do it?

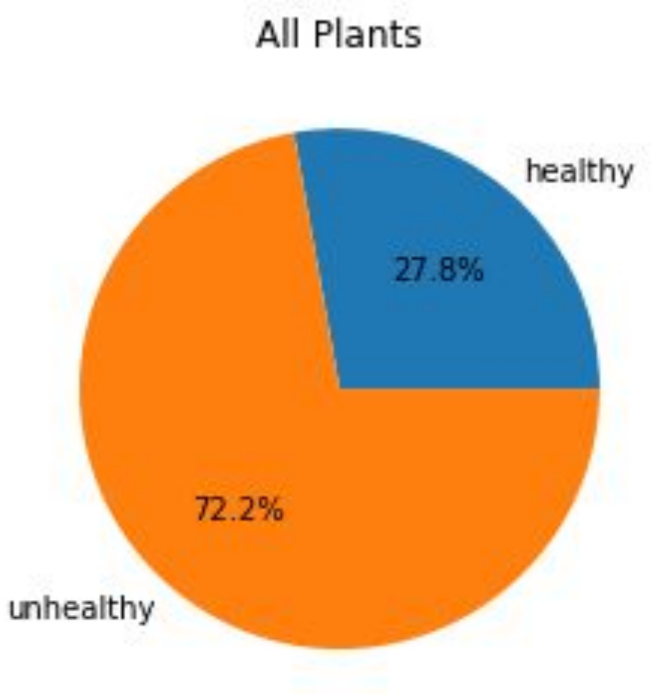
- We loaded ResNet50 Pre-Trained Model pretrained on imagenet
- Added layers and retrained data



06

Displaying Results

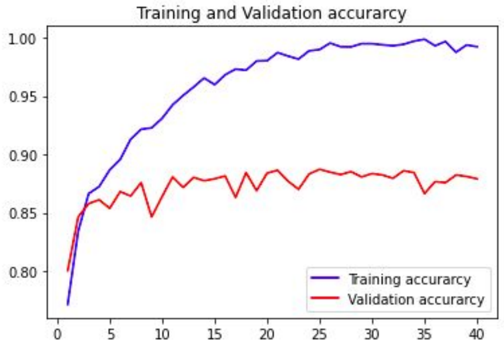
Healthy vs Unhealthy Plants: Pie Chart

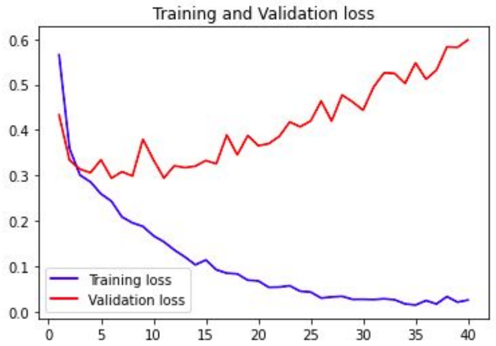


Results with Best Tuned Parameters:

BASELINE MODEL

- Combining all of the best parameters led to overfitting
- Decreasing the number of epochs solved this problem





Best Tuned Parameters (Baseline Model):

```
EPOCHS = 20
BS = 64
width=128
height=128
depth=3
INIT_LR = 1e-2
```

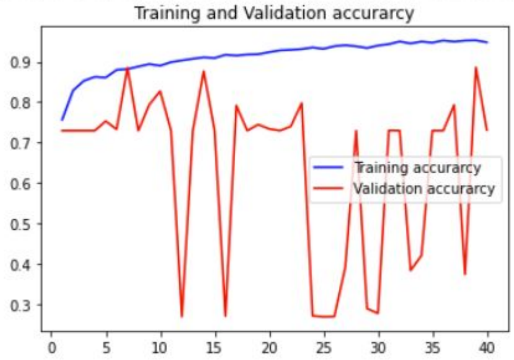
Transfer Learning Tuning

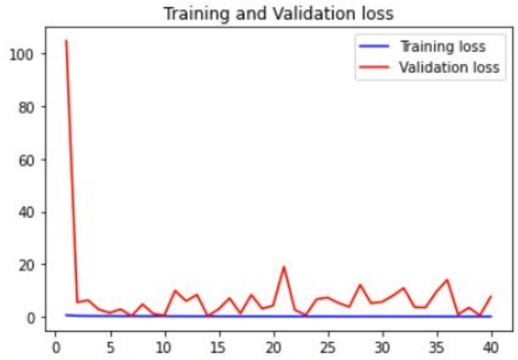
TRANSFER LEARNING MODEL

- Using a large learning rate with the transfer learning model led to a lower accuracy (73.11)
- High number of epochs led to overfitting
- To solve this, we used a smaller learning rate and a lower number of epochs

Best Tuned Parameters (Transfer Learning):

```
EPOCHS = 20
BS = 64
width=128
height=128
depth=3
INIT_LR = 1e-6
```

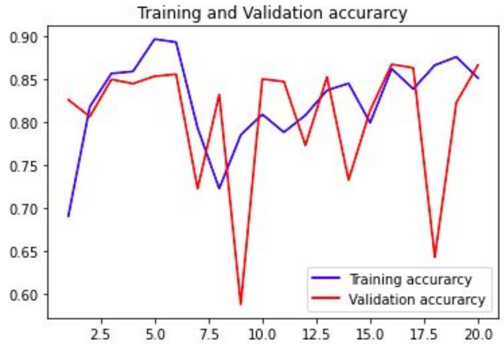


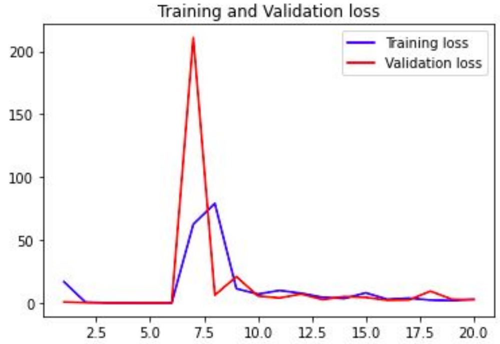



```
[INFO] Calculating model accuracy
104/104 [==============================] - 5s 44ms/
Test Accuracy: 73.11534881591797
```

Results with Best Tuned Parameters:

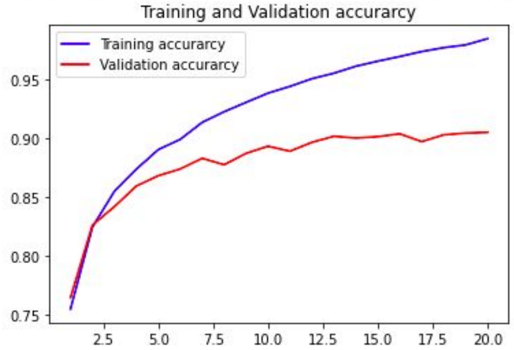
Baseline Model

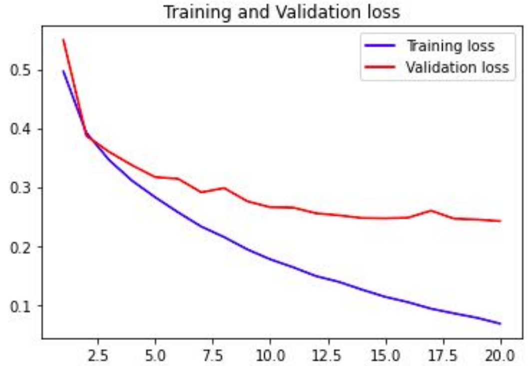




`Test Accuracy: 86.6775631904602`

Transfer Learning Model





`Test Accuracy: 90.0272011756897`

Thanks!

Do you have any questions?

Works Cited

Brownlee, J. (2019, September 16). *A Gentle Introduction to Transfer Learning for Deep Learning*. Machine Learning Mastery. https://machinelearningmastery.com/transfer-learning-for-deep-learning

Brownlee, J. (2020, August 18). Transfer Learning in Keras with Computer Vision Models. Machine Learning Mastery. https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

Jordan, J. (2020, August 29). Setting the learning rate of your neural network. Jeremy Jordan. https://www.jeremyjordan.me/nn-learning-rate/

United Nations. (n.d.). *Feeding the World Sustainably*. https://www.un.org/en/chronicle/article/feeding-world-sustainably