DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

IDP Final Report

Machine Learning and Computer Vision for Smart Plant Monitoring

Yoonha Choe

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

IDP Final Report

Machine Learning and Computer Vision for Smart Plant Monitoring

Machine Learning und Bildverarbeitung für intelligentes Pflanzenmonitoring

Author:	Yoonha Choe
Supervisor:	Prof. Senthold Asseng
Advisor:	Malte von Bloh
Submission Date:	14.10.2022

Abstract

When strong hail occurs, severe damage to agricultural crops caused by hail events can be protected by an insurance company that detects the damage rates of corresponding crops. However, the traditional way of detecting crop damage requires expert-level knowledge and high costs. In this study, we propose a computer vision and deep learning-based method of detecting sugar beet damage rates using a convolutional neural network that can be incorporated into a smartphone application. Using a dataset of damaged sugar beet images and corresponding damage rates, we train a convolutional neural network to regress the damage rates from the damaged sugar beet input image. We adopt a transfer learning mechanism by pre-training the model on plant-related images to further improve the performance of our trained model. We visualize the learned features of our trained model to see how our model recognizes the pattern of the damaged sugar beets. Our trained crop damage detection model achieves a Rooted Mean Square Error of 1.71 on the test set, hence demonstrating the feasibility of our deep learning-based approach. The results indicate that our work provides a clear path toward automated crop damage detection method based on a deep learning mechanism.

Contents

1. Introduction

Extreme weather events, such as strong hail events, can cause severe damage and yield losses to agricultural crops. Thus, crop hail insurance is essential to the livelihood of farmers, which provides coverage for damage and destruction of crops caused by hail. The traditional way to detect crop damage is carried out through field investigation and visual analysis done by experts, which generally requires professional knowledge and high costs. In recent years, remarkable advances in computer vision which are made possible by deep learning have paved the way for lots of applications including agricultural applications. This study adopts convolutional neural network (CNN) model, which has been widely used in research fields such as image recognition and machine vision, to detect crop damage rate from the image of a damaged crop. We especially investigate sugar beets that leave strong visual damage such as broken leaves from hail events due to their phenological characteristics.

For the last two years, after hail damage events happened, about 9,700 images including damage rates were taken under the guidance of the Allianz Agrar AG (AAA), and a first proof-of-concept has already been carried out by the Chair of Digital Agriculture at TUM in 2020 (von Bloh 2020). The goal of this study is to further improve the first research result by adopting deeper concepts of machine learning (ML) to make the ML models as robust as possible with a limited amount of data. We use one of the popular CNN models, ResNet (He et al. 2015), which is the winner of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015 in image classification, detection, and localization. We first extract all plant images from ImageNet (Deng et al. 2009) and pre-train the model on them rather than pre-train on ImageNet or train from scratch. Meanwhile, the training images are pre-processed using the object detection method since the dataset from AAA was not standardized. We do several experiments with network architectures, pre-trained models, training datasets, etc. to achieve the best performing model. In the end, crop damage detection can be done via mobile application by taking the image with the smartphone camera and transferring it to the server for prediction. The best performing model from our experiments achieves a Rooted Mean Square Error (RMSE) of 1.71 on the test set, hence indicating our approach of training deep learning model to predict crop damages presents a clear path toward smartphone-assisted crop damage detection.

The remainder of our paper is organized as follows. In the next chapter, we provide a detailed review of previous works related to our method. In Chapter 3, we outline the proposed approach to sugar beet damage detection. Subsequently, our experimental results are presented in Chapter 4. Lastly, Chapter 5 and 6 provide a discussion of the performance of our method and summarize our work respectively.

2. State of knowledge

2.1. Neural networks for crop monitoring

In the past few years, computer vision has made tremendous advances. The PASCAL VOC Challenge (Everingham et al. 2010) and more recently, the Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. 2014) based on the ImageNet (Deng et al. 2009) have been widely used as benchmarks for numerous visualization-related problems in computer vision. While training large neural networks can be time-consuming, the trained models can recognize the object in the images very quickly, which makes them also suitable for consumer applications on smartphones.

Recently, deep neural networks have been successfully applied in many diverse domains including precision agriculture. Neural networks provide a mapping between an image of a diseased plant, for example, to the metric of disease of a crop. The nodes in a neural network are mathematical functions that take numerical inputs from the incoming edges and provide a numerical output as an outgoing edge. Deep neural networks are trained by tuning the network parameters in such a way that the mapping improves during the training process. Mohanty et al. (Mohanty et al. 2016) propose smartphone-assisted disease diagnosis by training a deep CNN to identify 14 crop species and 26 diseases from the image taken smartphone. The trained model achieves an accuracy of 99.35%, demonstrating the feasibility of the deep learning approach. Kim et al. (Kim et al. 2021) also propose a vision-based method of detecting strawberry diseases using a deep neural network (DNN) capable of being incorporated into an automated robot system. Yang et al. (Yang et al. 2019) use CNN model to extract spectral features in the visible near-infrared range to estimate cold damage of corn seedlings. CNN detected the cold damage level of different types of corn seedlings, having a high correlation with the result of the chemical method. Many previous types of research have proven that CNN modeling-based approach can provide a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.

2.2. Transfer learning

Transfer learning is a machine learning technique where a model developed for a task is reused as the starting point for a model on a related task. It is a popular approach in deep learning area where pre-trained models are used as the starting point on computer vision or natural language processing tasks given the vast computing and time resources required to DNN models for these problems. The pre-trained model is a saved network that was previously trained on a large dataset, typically on a large-scale image classification task. We

can use transfer learning to customize this model to an actual given task. The intuition behind transfer learning is that if a model is trained on a large and general enough dataset, this model can effectively serve as a generic model of another similar task. We can take advantage of these learned feature maps without having to start from scratch by training a large model on a large dataset. One method of transfer learning that we use is fine-tuning. It is to unfreeze a few of the top layers of a frozen model base and jointly train both the newly-added classifier layers and the last layers of the base model. This allows us to fine-tune the higher-order feature representations in the base model in order to make them more relevant for the specific task. In our case, we select plant-related images from ImageNet (Deng et al. 2009) and train the model for the plant classification. We use this model as a pre-trained model and fine-tune it for our real task which is sugar beet damage detection as a regression model.

3. Material and Methods

Material and methods that are used in our study are described in this chapter, divided into three sections: pre-training on plant subset, regression task, and feature visualization. To briefly explain, as described in Chapter 2, we use transfer learning from a pre-trained network on a plant subset and fine-tune this model to our real task which is crop damage detection. Lastly, we evaluate what features are learned by doing feature visualization.

3.1. Pre-training on plant subset

3.1.1. Dataset

For pre-training, we select images which belong to plant classes from ImageNet. ImageNet (Deng et al. 2009) is an image database, a large-scale ontology of images built upon the backbone of the WordNet (Fellbaum 1998) structure. Thus, it is organized according to the WordNet hierarchy, where each node of the hierarchy is depicted by hundreds and thousands of images of the corresponding word. There is a class called 'plant life' in ImageNet, and if we search all hyponyms of 'plant life', we can find all classes which are subcategories of the plant. We use Natural Language Toolkit (NLTK) for implementation.

Among many versions of ImageNet, we use ImageNet-21K (Ridnik et al. 2021) which contains around 21,000 classes. We could see that there are 4,170 plant-related classes by finding hyponyms of 'plant life', and the number of images that each class contains differs from 1 to 2,113. To build our own training dataset, first, we set the threshold of the number of images per class as 500, and the number of classes becomes 1,912. However, due to the fixed capacity of our hardware, we choose 1,000 classes among them in order of the number of images they contain. However, with 500 images per class to classify 1,000 classes, the classification performs badly since the number of images is rather small. In the end, we have 1,000 images per class with 500 classes, thus, 500,000 images in total in our plant-related dataset. Our dataset contains flowers (e.g. tulip, azalea), plants (e.g. barley, maize), and tree classes (e.g. cedar tree, fir tree).

3.1.2. Training details

We use ResNet (He et al. 2015) model as the base model for plant classification which has achieved remarkable performance results in the ILSVRC 2015 classification challenge. A residual block is a key part of ResNet architecture which is described in Figure 3.1. In traditional neural networks, each layer feeds directly into the next layer. In contrast, in neural networks with the residual block, each layer feeds into the next layer and also into the layers

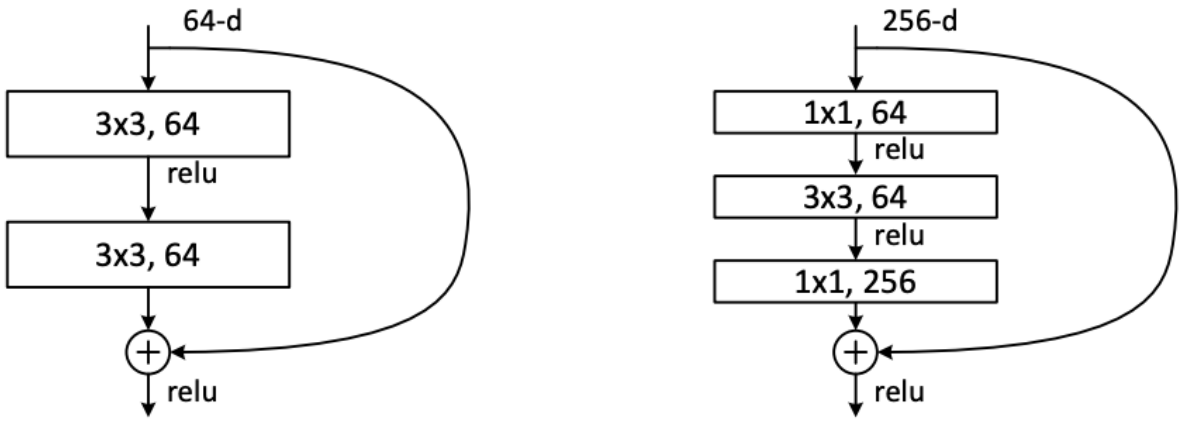


Figure 3.1.: A deeper residual function for ImageNet. Left: a building block (on 56×56 feature maps) for ResNet-34. Right: a building block for ResNet-50/101/152.

about 2 or 3 hops away. The residual blocks allow gradients to flow through the network directly, without passing through non-linear activation functions which cause the gradients to explode or vanish. There are many variants of ResNet architecture, i.e., the same concept but with a different number of layers: ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152, etc.

We train the model to classify 500 plant-related classes, where the loss function is cross-entropy loss. We split the dataset into two sub-datasets with an 80:20 ratio, training set and validation set. We resize the input images to 224x224 and normalize them before feeding them into the network. We use the batch size as 256 and Adam optimizer with learning rate decay as $1 * 10^{-5}$ for optimization. To achieve the best performance with our dataset, we do experiments with the size of the model, i.e., ResNet-18,34 and 50. Further, we do experiments with the learning rate, $5 * 10^{-5}, 1 * 10^{-4}, 3 * 10^{-4}$ ($1 * 10^{-4}$ is commonly used value for Adam optimizer). Lastly, we check if training from scratch or pre-training on ImageNet performs better for the classification of plants. We use PyTorch for implementation. See the results in Chapter 4.1.

3.2. Regression task

3.2.1. Dataset

We have dataset where 259 images are collected by AAA and 9,526 images by TUM, thus 9,785 images in total. The quality of collected images is categorized as low, medium, and high-quality. The low-quality images are not recorded well such as with the wrong recorded angle, the medium-quality images have a quite large gap between the date of the event and recording or are not fully standardized recorded, and high-quality images are in general suitable to use. There are 15 low, 83 medium, and 161 high-quality images from AAA, and all images from TUM are high-quality. Let this dataset be denoted Dataset I.

However, since parts of the dataset were collected by AAA which uses personnel who have little experience with the operation of recording devices, the AAA's data differed significantly in structure from the images from TUM. Thus, the collected images are cropped using an object detection algorithm to make the recorded plant fully fit in the image. Sometimes there are multiple plants captured in one image, in this case, to contain one plant per image, the cropped image can be multiple from one original image. The newly captured images by drones are also included in this dataset. Let this standardized dataset be denoted Dataset II. Dataset II contains 11,957 images in total, 319 images are taken by partner, 11,045 by TUM, and 593 by drone. All drone images are high-quality.

3.2.2. Training details

We use transfer learning from the pre-trained network on a plant subset which is described in Chapter 3.1 and fine-tune the model to the crop damage detection task. We change the fully connected layer of pre-trained model since the crop damage detection model is a regression model. We use the pre-trained weights as an initialization of weights of the damage detection model, which means that we re-train the entire model. We initialize the weights of the hidden layers before fully-connected (FC) layers with the weights of the pre-trained model and the weights of new FC layers are initialized randomly with zero mean. Thus, we don't freeze any layers and we just initialize the weight with the one from the pre-trained model. We set the test set which is 10% of the total dataset while maintaining the image quality ratio (high, medium, and low-quality). The rest 90% of the dataset is again split into training and validation set with an 80:20 ratio. We resize the input images to 224x224 and normalize them before feeding them into the network. We use the batch size as 64 and Adam optimizer with learning rate decay as $1 * 10^{-5}$ for optimization. The learning rate is set to $1 * 10^{-4}$. To achieve the best performance, first, we do experiments with ResNet-34 models which are pre-trained on ImageNet and plant subset respectively to see if pre-training on the plant subset really increases the performance of crop damage detection. Further, we do experiments with datasets, Dataset I and II, to see if the standardization of images helps crop damage detection. Lastly, we do experiments with the size of the model, ResNet-34 and 50. As a baseline model which can be compared with the deep learning approach, we fit our dataset to the polynomial of degree 1 where the parameter is the percentage of green pixels in the image. Intuitively, the crop damage and the number of green pixels should be inversely proportional, which means the crop damage decreases as the number of green pixels increases. With this linear model, we can see how the deep learning approach is contributing to our crop damage detection task. See the results in Chapter 4.2.

3.3. Feature visualization

CNNs learn abstract features and concepts from raw image pixels. The first convolutional layer(s) learn features such as edges and simple textures, and later convolutional layers learn features such as more complex textures and patterns. Visualizing the features learned by

CNN is important to understand how a neural network predicts certain decisions, in specific, how it recognizes specific patterns or objects in the image. We adopt the feature visualization technique called activation maximization which was proposed as a way of producing a saliency map for CNN (Simonyan et al. 2013) to understand how our trained model handles crop damage detection task. It is done by starting with an image consisting of randomly initialized pixels and finding the image that maximizes the activation of the unit which we want to visualize. The unit refers to individual neurons, channels (also called feature maps), or entire layers. We choose the channels as units which is a commonly used choice for feature visualization. In the case of ResNet-34, as shown in Figure 3.2, we visualize the feature maps in the convolutional layers which are conv1, layer1,2,3, and 4.

In mathematical terms, feature visualization is an optimization problem. We assume that the weights of the neural network are fixed, which means that the network is already trained. We optimize a randomly initialized image which maximizes the mean activation of the unit that we want to visualize, here the entire channel, as follows:

$$img^* = argmax_{img} \sum_{x,y} h_{n,x,y,z}(img), \tag{3.1}$$

where the function h is the activation of the neuron, img is the random noise input image of the network, x and y describe the spatial position of the neuron, n specifies the layer, and z is the channel index. This formula computes the mean activation of certain channel z in the layer n, and all neurons in channel z are equally weighted. A naive application of activation maximization on CNNs, however, tends to produce extremely high-frequency images which look nothing like real-world natural images. To tackle this problem, we use three regularization techniques to make the images more meaningful. We first start with a small 28x28 image and slowly upscale it to the desired size. We penalize large pixel values and also large pixel gradients in the image i.e. penalizing any sharp changes in the values of neighboring pixels. The results of feature visualization are shown in Chapter 4.3.

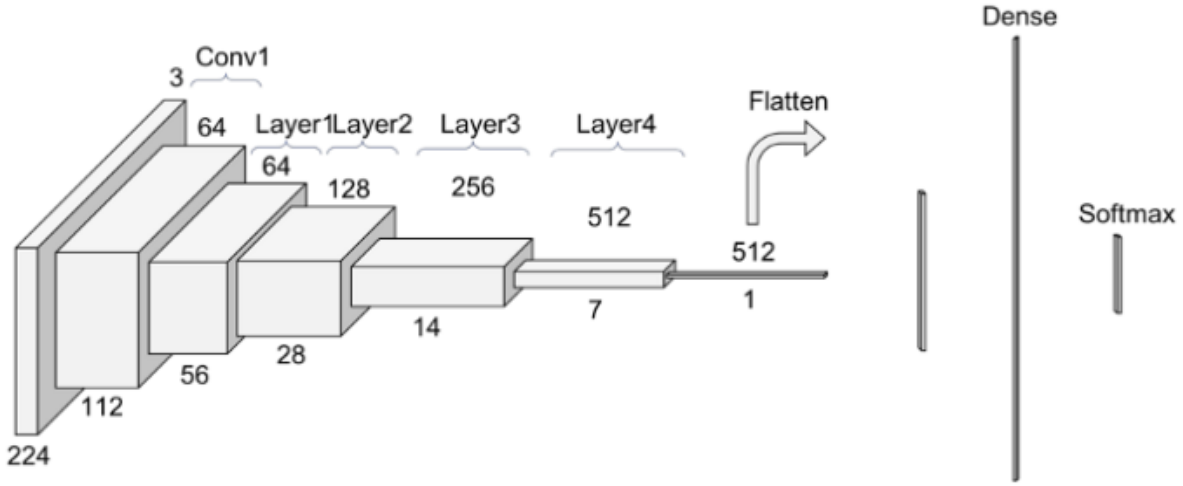


Figure 3.2.: The architecture of the ResNet-34 model. Conv1 layer consists of a convolution, batch normalization, and max pooling operation. Each layer 1,2,3 and 4 consists of convolution, batch normalization, and ReLU activation. After that, a fully connected layer and softmax function come.

4. Results

This chapter shows the experimental results of each section explained in Chapter 3. They contain plots of training and validation loss, the test results of the trained model, and learned feature visualization.

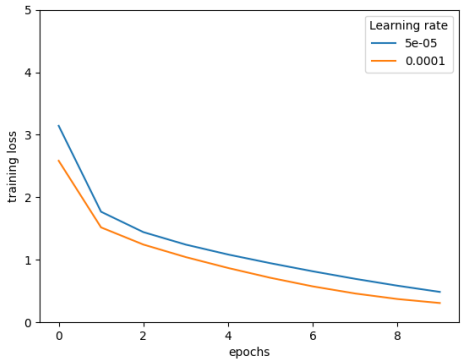
4.1. Pre-training on plant subset

For pre-training, we do experiments with pre-training mechanism, learning rate, and network architecture respectively. We plot not only cross entropy training and validation loss but also training and validation top1/top5 accuracy for deeper understanding. In the case of the top1 accuracy, we check if the top class (the one with the highest probability) is the same as the target label. In the case of the top5 accuracy, we check if the target label is one of top 5 predictions (the 5 ones with the highest probabilities). The cross entropy training and validation loss are depicted in Figure 4.1 and 4.2 respectively. The top1/top5 accuracy are provided in Appendix A.

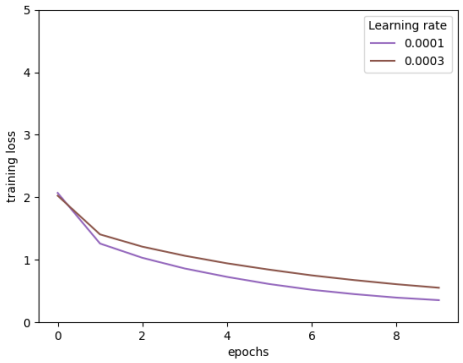
In the first experiment, we train the model with different learning rates. We train ResNet-18 model with learning rate $1 * 10^{-4}$ and $5 * 10^{-5}$ respectively. The validation loss is depicted in Figure 4.2a, and we can see the model with learning rate $1 * 10^{-4}$ approaches the lower validation loss faster. We also train ResNet-50 model with learning rate $1 * 10^{-4}$ and $3 * 10^{-4}$, and as we can see in Figure 4.2b, the model with learning rate $1 * 10^{-4}$ performs better. In the second experiment, we compare the model which is trained from scratch and the model which is pre-trained on ImageNet. Figure 4.2c shows that the model pre-trained on ImageNet performs better. The last experiment is to see the effect of the size of network architecture. We can see the larger model, ResNet-50, performs better as depicted in Figure 4.2d. From those experiments, ResNet-50 model with the learning rate $1 * 10^{-4}$ pre-trained on ImageNet performs the best for plant classification. This best performing model is used as pre-trained model for the regression task of crop damage detection.

4.2. Regression task

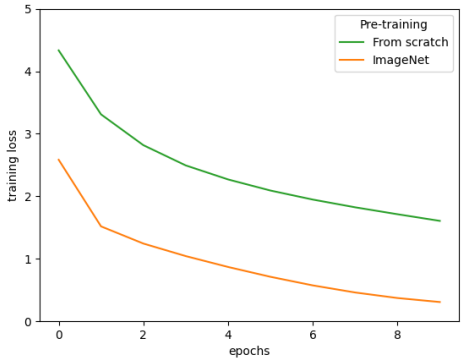
For the regression task of crop damage detection, first, we fit the polynomial of degree 1 which is the baseline and we denote this model as a linear model. The linear model is fitted to Dataset I and the fitted model is depicted in Figure 4.3. For the deep learning-based models, we do experiments with pre-training mechanism, dataset, and network architecture respectively. We change the experimental parameters one by one and keep the rest constant, thus we have four deep learning-based models in total. We plot RMSE training and validation



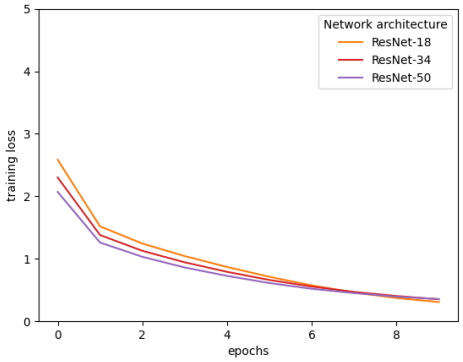
(a) Learning rate 1



(b) Learning rate 2



(c) Training from scratch or pre-training on ImageNet

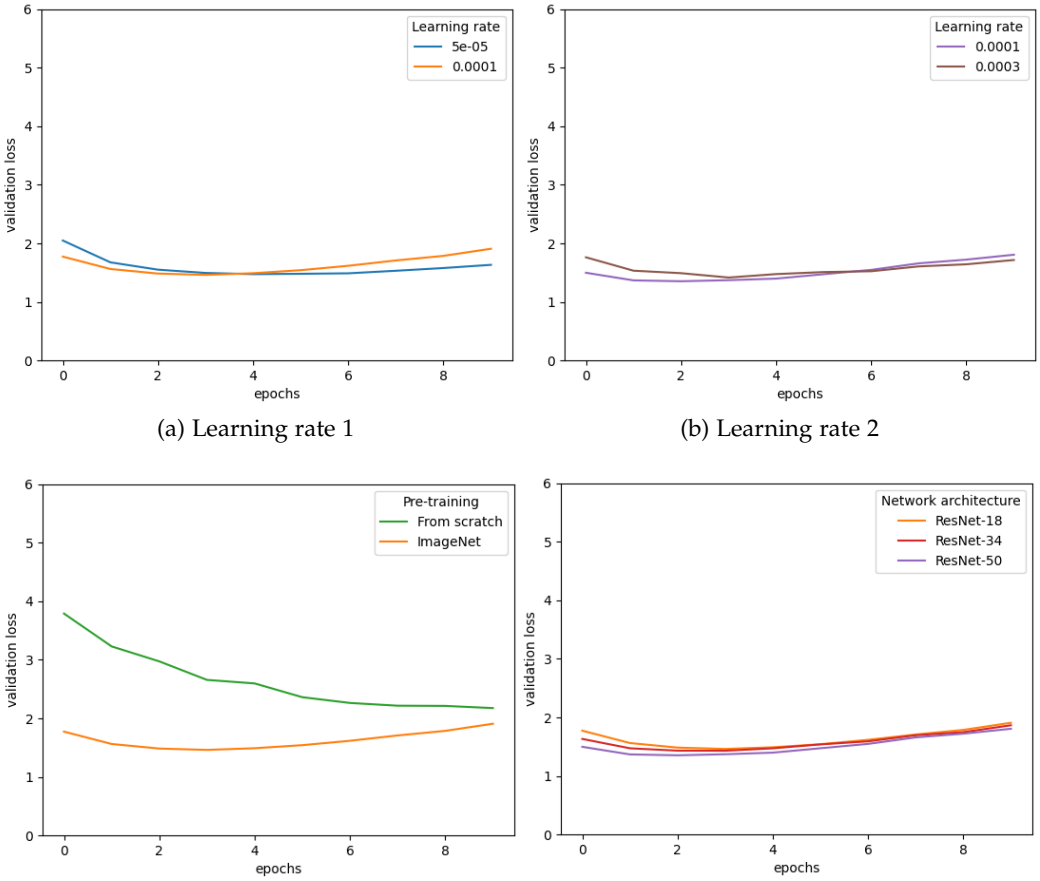


(d) The size of network architecture

Figure 4.1.: Cross entropy training loss through the training period of 10 epochs across comparable experiments with experimental configuration parameters: learning rate, pre-training mechanism, and network architecture.

loss and Mean Absolute Error (MAE) training and validation loss for deeper understanding. RMSE training and validation loss curve are depicted in Figure 4.4 and 4.5. RMSE value is better to see the difference between the performance of experiments than MAE value. MAE training and validation loss curves are provided in Appendix B.

In the first experiment, we use the models as pre-trained model which are trained on ImageNet and plant subset which performs the best on plant classification (see Chapter 4.1) respectively. We compare RMSE loss values to see if pre-training on plant images helps crop damage detection. The model pre-trained on plant subset performs better than ImageNet (see Figure 4.5a). The second experimental parameter is the training dataset. The two versions of dataset (Dataset I and II) show the effect of standardization of AAA data, and the model trained on the standardized one, Dataset II, performs better than the original dataset, Dataset I (see Figure 4.5b). The last experiment is to see if the increased size of the network architecture



(a) Learning rate 1

(b) Learning rate 2

(c) Training from scratch or pre-training on ImageNet

(d) The size of network architecture

Figure 4.2.: Cross entropy validation loss through the training period of 10 epochs across comparable experiments with experimental configuration parameters: learning rate, pre-training mechanism, and network architecture.

improves the accuracy of crop damage detection. The larger network architecture, ResNet-50, performs better than ResNet-34 (see Figure 4.5c). From these experiments, ResNet-50 model trained with Dataset II which is pre-trained on the plant subset performs the best.

We do the test with five models including the linear model and get RMSE test loss values. Table 4.1 shows the RMSE test loss across all our experimental configurations. Across all experiments, the total RMSE test loss we obtain varies from 32.56 (in the case of none, Dataset I, linear) to 1.71 (in the case of plant subset, Dataset II, ResNet-50), hence showing strong promise of deep learning approach for crop damage detection problem. For each TUM, AAA, and drone dataset, the model performed the best in the case of the plant subset, Dataset II, ResNet-50. Figure 4.6 shows the test results of four deep learning-based models. For the test result of the linear model, see Appendix C.

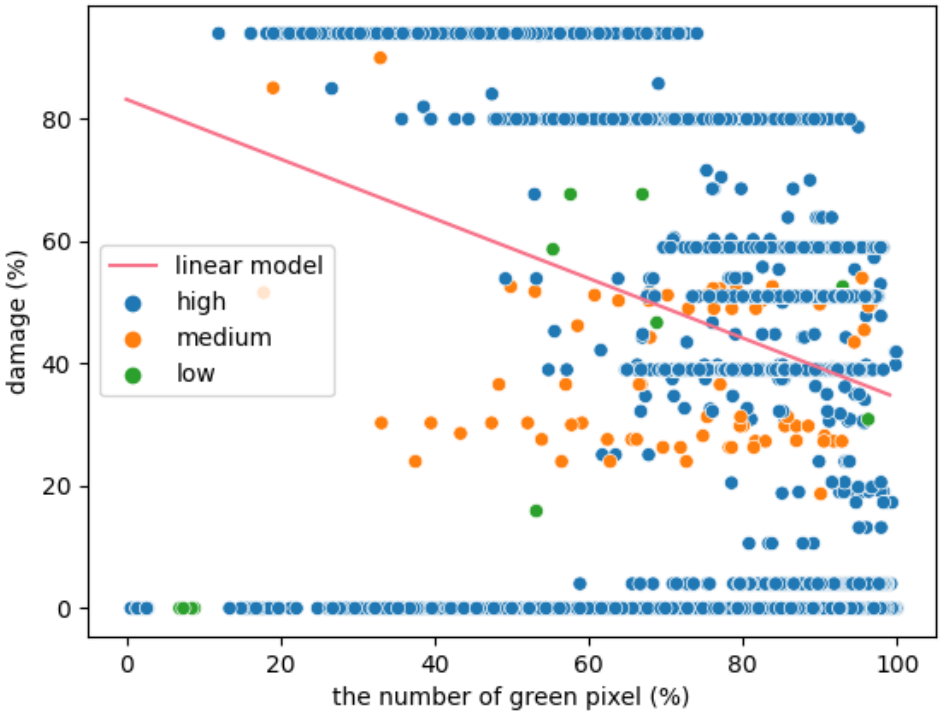
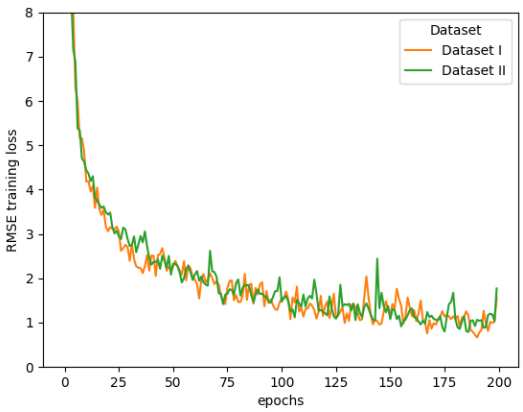


Figure 4.3.: The linear model fitted to Dataset I. The circles depict the relation between green pixel percentage and damage percentage (Blue: high-quality image, orange: medium-quality image, green: low-quality image). The pink-colored line is the fitted linear model.

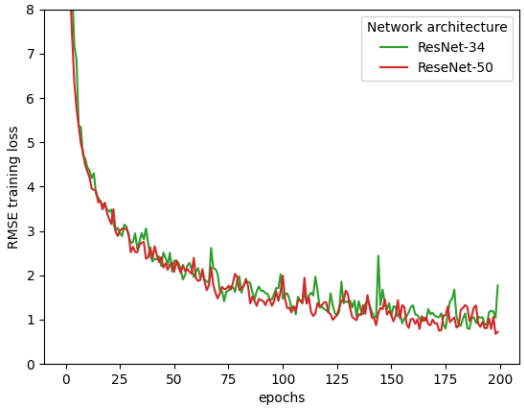
4.3. Feature visualization

As described in Chapter 3.3, we visualize the learned feature of the trained crop damage detection model which performs the best on the test set. We visualize the feature maps in layer 1,2,3, and 4 since they contain rather meaningful and discernable patterns than conv1 or fully connected layers. Among lots of units in each layer, we visualize the top 10 most activated units in each layer to see the most important learned patterns in the network. The visualization of feature maps is described in Figure 4.7. As seen in Figure 4.7, the patterns which the network learns are more complex as the layer is going deeper. Layer 1 learns basic features such as lines or small circles in the image (Figure 4.7a). In layer 2, the patterns get more complex and one of them seems the stem of the sugar beet (Figure 4.7b). In layer 3, the circle patterns in the image gets more discernable (Figure 4.7c), and layer 4 learns quite specific patterns (Figure 4.7d).

(a) Pre-training on ImageNet or plant subset

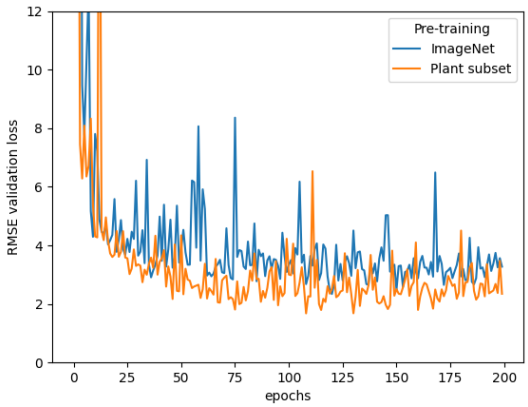


(b) Training dataset

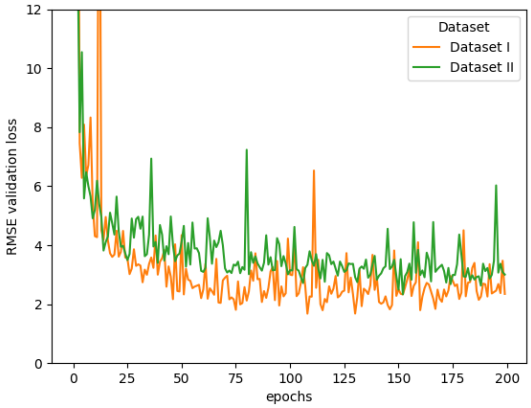


(c) The size of network architecture

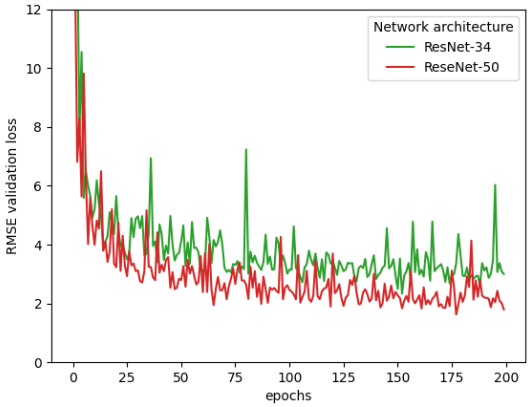
Figure 4.4.: RMSE training loss through the training period of 200 epochs across comparable experiments with experimental configuration parameters: pre-training mechanism, dataset, and network architecture.



(a) Pre-training on ImageNet or plant subset



(b) Training dataset

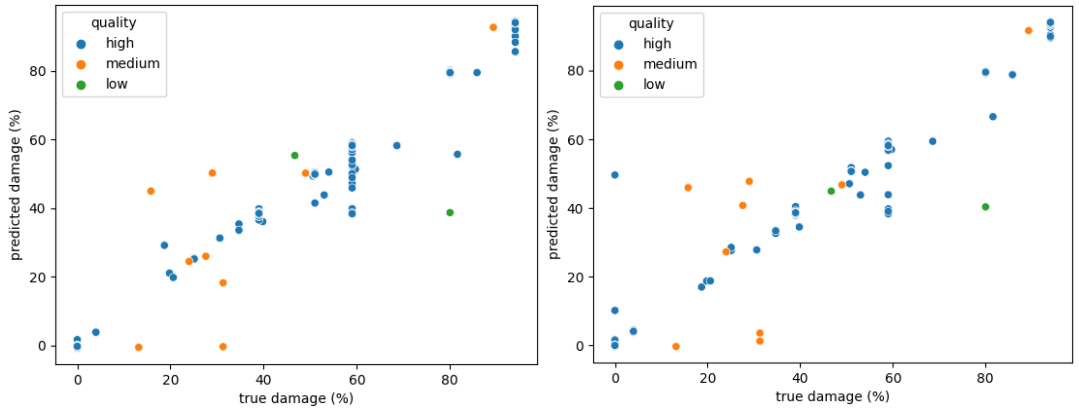


(c) The size of network architecture

Figure 4.5.: RMSE validation loss through the training period of 200 epochs across comparable experiments with experimental configuration parameters: pre-training mechanism, dataset, and network architecture.

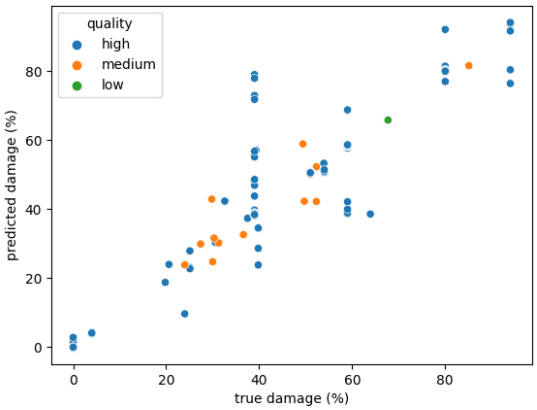
Table 4.1.: RMSE test loss across all trained models. The model with the lower RMSE test loss value performs better. ResNet-50 model trained with Dataset II which is pre-trained on plant subset performs the best with 1.71 RMSE test loss on the total dataset. Dataset I contains 907 TUM and 28 AAA test images, and Dataset II contains 1,105 TUM, 32 AAA, and 59 drone test images.

Model	RMSE test loss			
(pre-training mechanism, dataset, network architecture)	TUM	AAA	Drone	Total
None, Dataset I, Linear	32.77	25.05	-	32.56
ImageNet, Dataset I, ResNet-34	1.86	14.10	-	3.05
Plant subset, Dataset I, ResNet-34	1.16	15.80	-	2.96
Plant subset, Dataset II, ResNet-34	3.31	11.49	0.46	3.69
Plant subset, Dataset II, ResNet-50	**1.29**	**7.15**	**0.20**	**1.71**

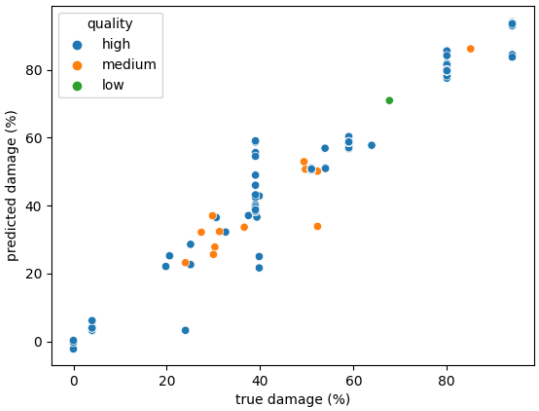


(a) ImageNet, Dataset I, ResNet-34

(b) Plant subset, Dataset I, ResNet-34

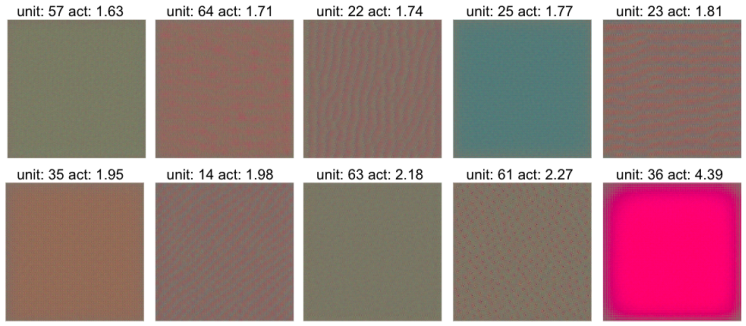


(c) Plant subset, Dataset II, ResNet-34

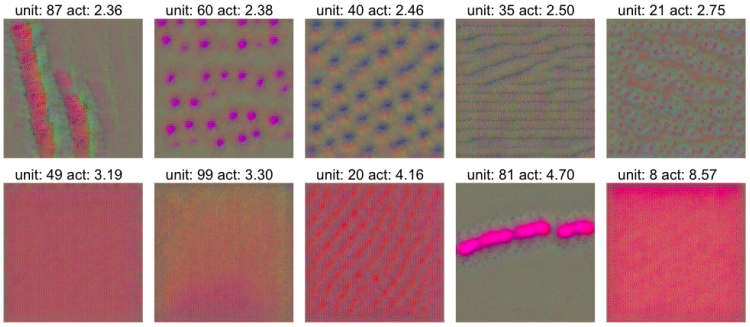


(d) Plant subset, Dataset II, ResNet-50

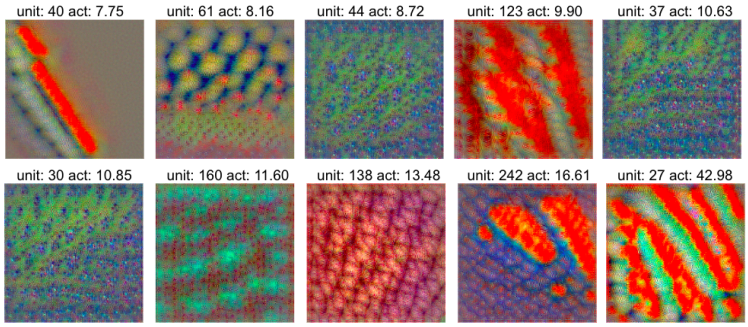
Figure 4.6.: Test results with four different models. The circles depict the relation between true damage and predicted damage (Blue: high-quality image, orange: medium-quality image, green: low-quality image).



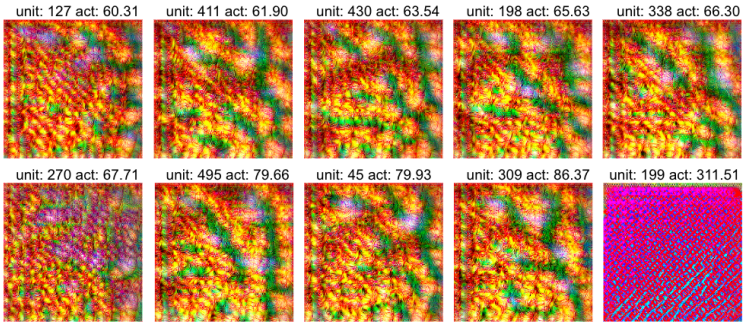
(a) Layer 1



(b) Layer 2



(c) Layer 3



(d) Layer 4

Figure 4.7.: Visualization for the top 10 most activated units in each layer. The learned features in layer 1,2,3 and 4 are visualized. (unit: n-th unit, act: activation score)

5. Discussion

In this chapter, we analyze the results of our trained models, discuss the limitation of our work and suggest feasible future work.

5.1. Discussion on pre-trained model

We showed the results of experiments with learning rate, pre-training mechanism, and network architecture for pre-trained model in Chapter 4.1. From Figure 4.2a and 4.2b, the model with learning rate $1 * 10^{-4}$ achieves a lower cross entropy validation loss than the model with lower learning rate, $5 * 10^{-5}$, or higher learning rate, $3 * 10^{-4}$, thus $1 * 10^{-4}$ is an adequate learning rate for plant classification task. As we can see in Figure 4.2c, the model which is pre-trained on ImageNet achieves lower validation loss than the model which is trained from scratch. This is obviously expected result since the training dataset for plant classification which is plant subset is extracted from ImageNet. We can see the bigger model, ResNet-50, achieves lower validation loss than the smaller model, ResNet-18 or 34 (see Figure 4.2d). We could not train with much bigger model due to the capacity of our resources, however, there is room for improvement in performance regarding the size and the complexity of the model.

The validation top1 accuracy of the best performing model is slightly above 60% and top5 accuracy is about 90% (see Figure A.3 and A.4). However, we have not tested the trained model on held-out test set, thus we could not provide the test accuracy. We suggest comparison the test accuracy with other plant classification models as future work. Furthermore, we extract plant-related images from ImageNet manually, however, the choice of 500 classes for our task among 4,170 classes has no convincing criteria. There are other plant-related datasets in the literature, for example, PlantNet-300K (Garcin et al. 2021) which is better structured than ours and where many species are visually similar, making identification difficult even for the expert eye. Comparison with several different plant-related dataset can also be addressed in future work.

5.2. Discussion on regression model

We showed the results of experiments with pre-training mechanism, training dataset, and network architecture for the regression task in Chapter 4.2. The initialization of weights from the pre-trained model is crucial when we adopt transfer learning mechanism. Since the crops vary in their shapes and colors, it is inappropriate to extract crop features using a pre-trained model based on the ImageNet which does not reflect adequate domain knowledge. The

source and target domains need to be well connected with each other to maximize the effect of transfer learning. Kim et al. (Kim et al. 2021) also prove this statement by showing that the strawberry disease detection model which is pre-trained on PlantCLEF plant dataset achieves higher accuracy than the one pre-trained on ImageNet. Our validation loss curve also shows that the model pre-trained on the plant subset achieves a lower RMSE value (see Figure 4.5a), and for the test result, the performance of the model which is pre-trained on the plant subset has improved for TUM and total dataset compared to the model which is pre-trained on ImageNet (see Table 4.1). This result implies that using pre-trained models with the plant-related dataset for crop damage detection is efficient. We have Dataset II which is generated by using the object detection method to standardize the sugar beet images collected from TUM and AAA. With these standardized images, the validation RMSE loss gets lower (see Figure 4.5b), and the performance of the model has improved for AAA dataset. (see Table 4.1). Especially, test loss of AAA images gets much lower compared to using Dataset I, which implies that standardization using an object detection algorithm definitely improves the quality of AAA dataset which contains many low or medium-quality images and reduces the gap in structure between TUM and AAA images. However, most images in our dataset are collected from TUM, and the model trained on Dataset II performs worse on TUM images than the model trained on Dataset I, hence performing worse on the total dataset. For the size of network architecture, we can see ResNet-50 achieves lower RMSE value than ResNet-34 (see Figure 4.5c), and the performance of ResNet-50 on the test set has improved for TUM, AAA, drone and total dataset (see Table 4.1). As mentioned in Chapter 5.2, with larger and more complex model the performance can be further improved.

For the linear regression model, we fit to the polynomial of degree 1 where the parameter is the percentage of green pixels in the image. Table 4.1 shows the test loss of the linear model performs almost 20 times worse compared to deep learning-based models, hence demonstrating deep learning approach contributes to our crop damage detection task. The problem with the linear regression model is that the green pixel percentage parameter is inappropriate and insufficient to predict crop damage. As depicted in the first row images in Figure 5.1, there is a case of having low green pixel percentage even though the damage rate of the corresponding crop is 0. Thus, it is infeasible to predict properly zero damage crops only with the green pixel percentage.

There are a number of limitations at the current stage that need to be addressed in future work regarding the regression task. First, the damage rates of the collected dataset are not evenly distributed. There are 11,957 crop images in Dataset II, and the number of images of which damage rates belong to [0, 20) is 4,881, [20, 40) is 1,632, [40, 60) is 2,309, [60, 80) is 26, and [80, 100] is 3,109. We can see that the images belonging to [40, 80) are comparably fewer than the images belonging to the rest interval. As seen in Figure 4.6, we can see the test images of which damages are around 40 and 60 are not predicted well. Our current results indicate that new image collection efforts should try to obtain images from many different damage rates, thus we can generate a more evenly distributed dataset and substantially improve the performance of our model.

The second limitation is that we have not tested the robustness of our trained regression

Figure 5.1.: Example of sugar beet images of which damages rates are 0 value. For the top row images, the green pixel percentage is 18.53% (left) and 11.98% (right). For the bottom row images, the green pixel percentage is 80.45% (left) and 88.05% (right).

model. The robustness of the model is crucial in deep learning, which refers to the degree that the model's performance changes when using new data versus training data, thus performance should not deviate significantly. Trust in ML tools depends on reliable performance, and to ensure that a model is performing according to its intended purpose, managing and testing the robustness of the trained model can be addressed in future work.

Crop insurance for extreme natural disasters can play an important role in protecting agricultural production in order to ensure future yield and business as well. Traditionally, insurers send experts to the farm to assess the damage, which involves traveling to remote areas to reach their clients. They must also spend considerable time in manual field scouting for damage quantification, making the process expensive for the crop insurer. Moreover, the damage measurements must be accurate and efficient so that neither the farmer nor the crop insurer suffers losses. Thus, our work can solve the traditional issues that insurers face and provide additional benefits. Crop insurers can process more crop damage claims than before and more accurately. We expect that with deep learning-based detection method they can quickly offer fair compensation to farmers who use insurance as a risk management tool.

5.3. Discussion on feature visualization

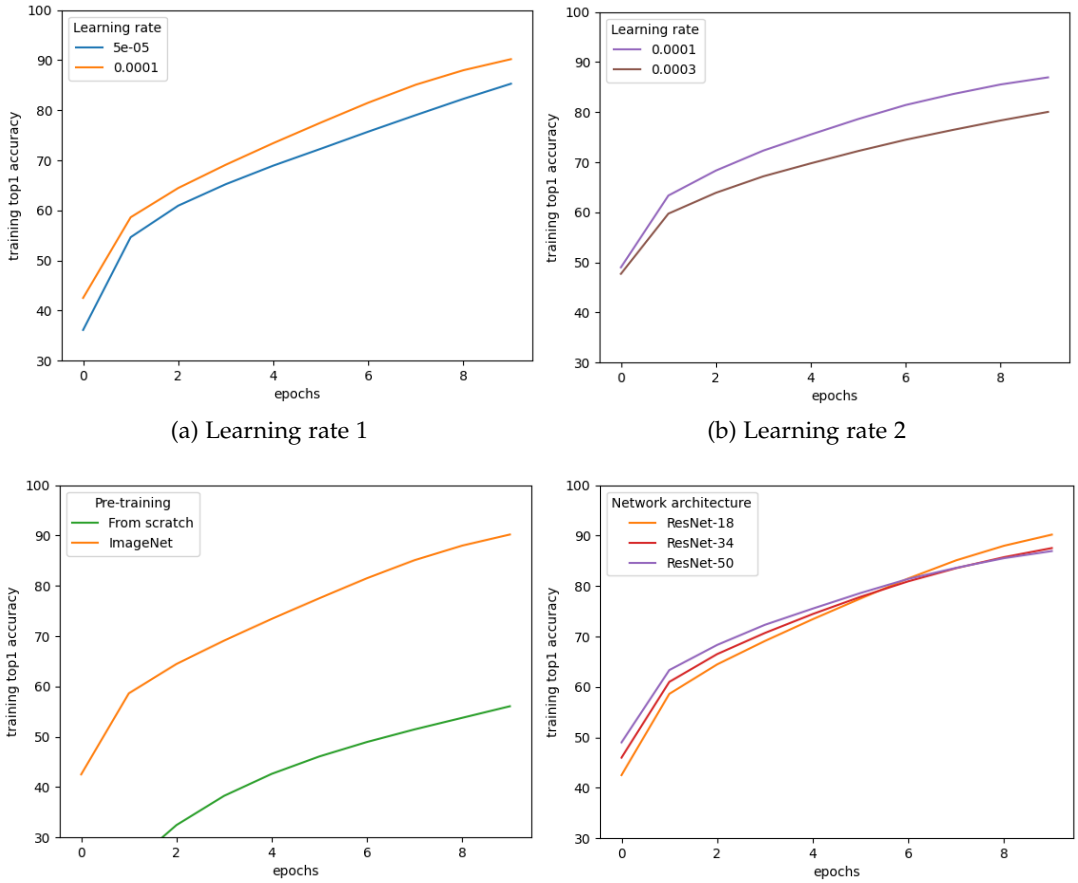
We visualized the learned feature maps of the best performing model in Figure 4.7. Nguyen et al. (Nguyen et al. 2019) review the existing activation maximization techniques in the

literature and show various results of activation maximization techniques. Mostly, the trained model which visualization techniques visualize in the literature is the classification model on ImageNet since in this case, visualization can show distinguished patterns of each object in the image from ImageNet. For example, in the case of an animal, the pattern of eyes or legs of the animal is repeated in the visualization, or in the case of a mountain, the pointed shape of the mountain top appears in the visualization. However, as seen in Figure 4.7, it is hard to interpret that there are characterized patterns in our visualization. We see many circles and a few lines in the visualization, especially in Figure 4.7b and 4.7c, but it is not sufficient to conclude that they are necessary and crucial learned patterns to predict the sugar beet damage. In our task where training images are the same object which is sugar beet and are not prominently different each other, activation maximization technique does not give much information on which patterns in the image trained model sees to predict the damage rates of crops. The study of other feature visualization techniques which can be applied to our task can be addressed in future work.

6. Conclusion

In this work, we have proposed a sugar beet damage detection method based on ResNet CNN model, which achieved RMSE of 1.71 on a held-out test set. By pre-training the model on plant-related images rather than on ImageNet, and with standardized dataset, we could generate the detection model with reliable performance. In addition, we have visualized the learned features of the trained model to understand how it recognizes specific patterns in the sugar beet image to predict damage rates. Detecting crop damages accurately and quickly is necessary for crop insurance to reduce the effort and time cost. Our work can play an important role in providing a clear path toward automated and efficient crop damage detection system which can bring lots of benefits to both farmers and insurers.

A. Additional training and validation loss of plant classification



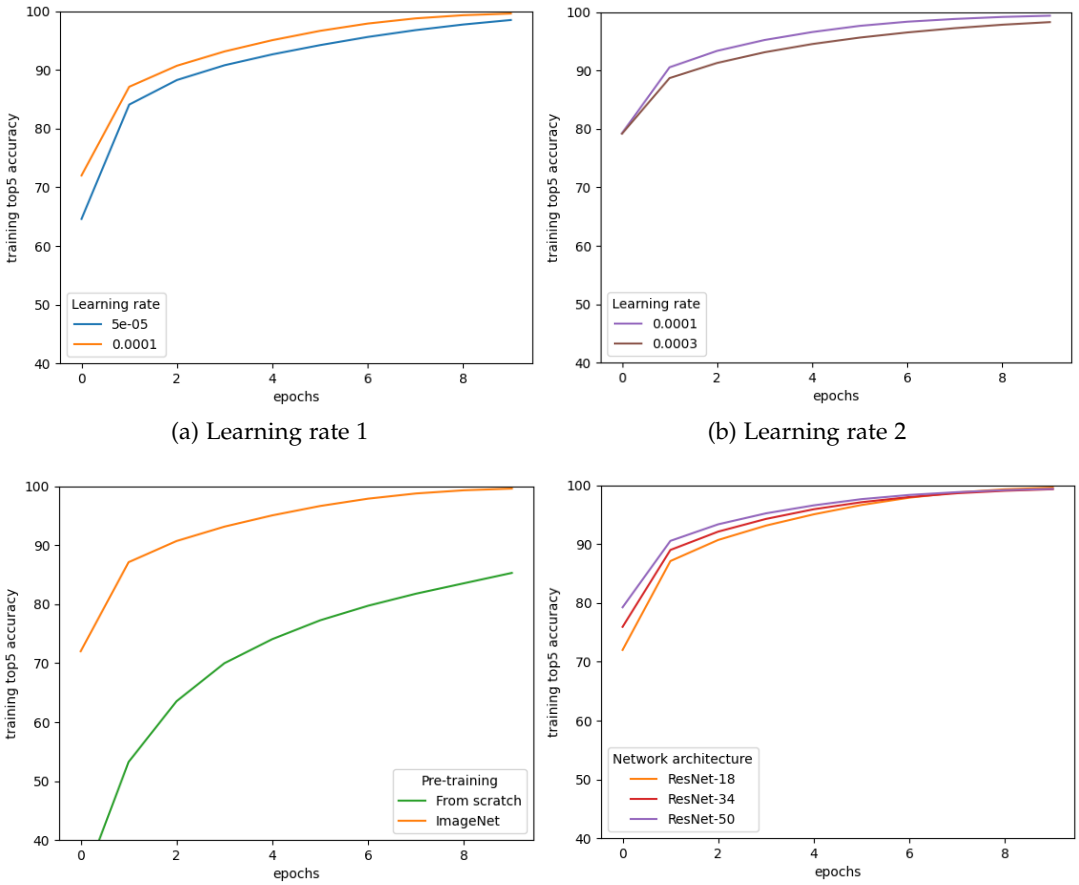
(a) Learning rate 1

(b) Learning rate 2

(c) Training from scratch or pre-training on ImageNet

(d) The size of network architecture

Figure A.1.: Training top1 accuracy through the training period of 10 epochs across comparable experiments with experimental configuration parameters: learning rate, pre-training mechanism, and network architecture.



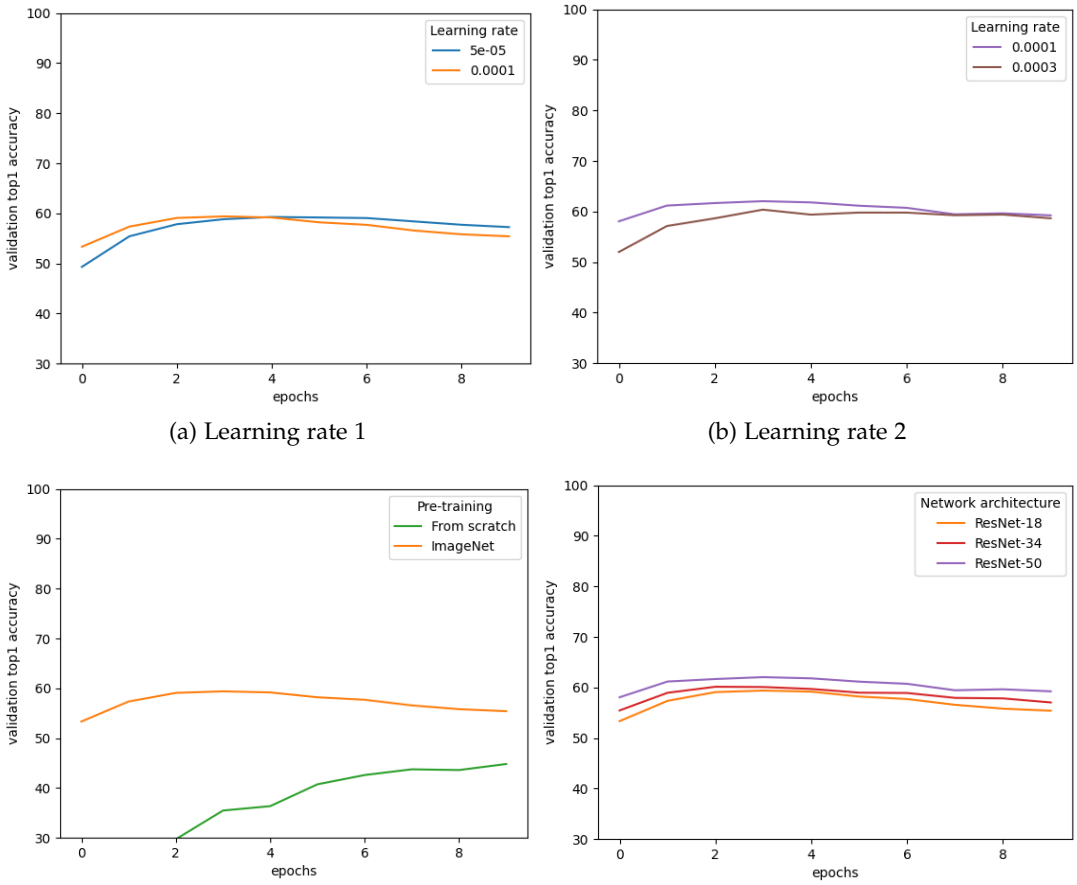
(a) Learning rate 1

(b) Learning rate 2

(c) Training from scratch or pre-training on ImageNet

(d) The size of network architecture

Figure A.2.: Training top5 accuracy through the training period of 10 epochs across comparable experiments with experimental configuration parameters: learning rate, pre-training mechanism, and network architecture.



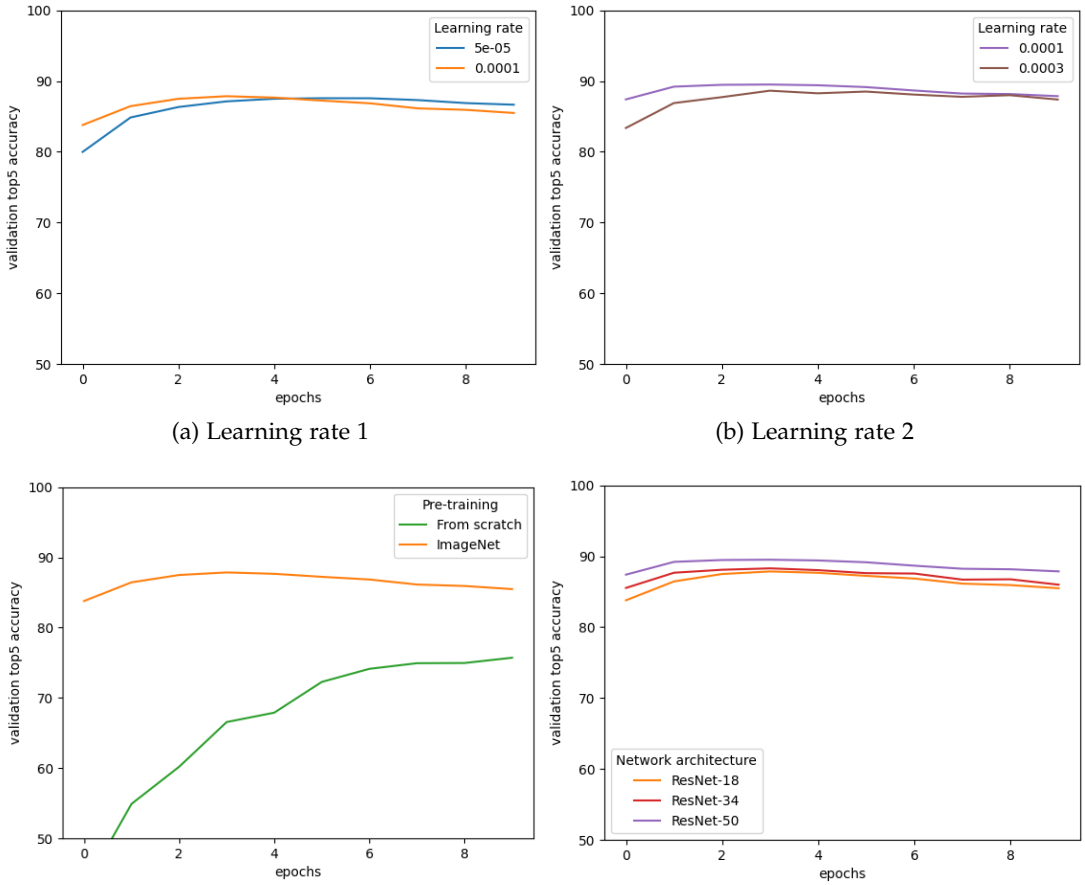
(a) Learning rate 1

(b) Learning rate 2

(c) Training from scratch or pre-training on ImageNet

(d) The size of network architecture

Figure A.3.: Validation top1 accuracy through the training period of 10 epochs across comparable experiments with experimental configuration parameters: learning rate, pre-training mechanism, and network architecture.



(a) Learning rate 1

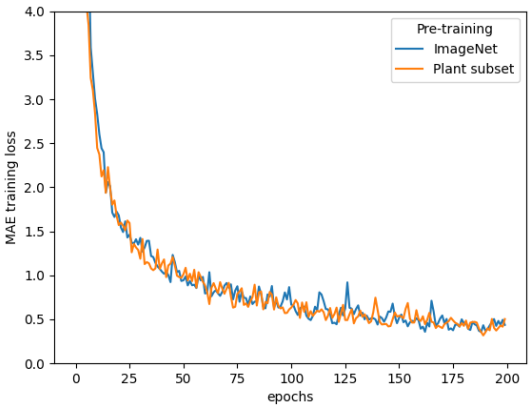
(b) Learning rate 2

(c) Training from scratch or pre-training on ImageNet

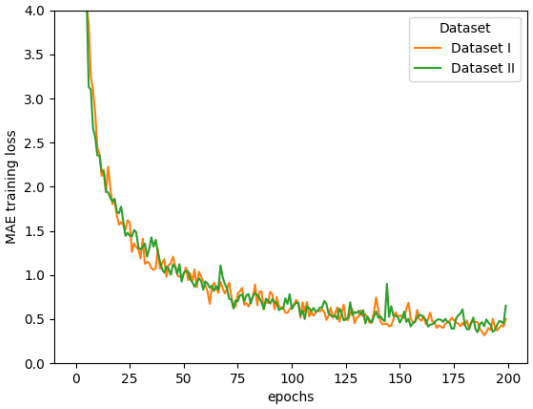
(d) The size of network architecture

Figure A.4.: Validation top5 accuracy through the training period of 10 epochs across comparable experiments with experimental configuration parameters: learning rate, pre-training mechanism, and network architecture.

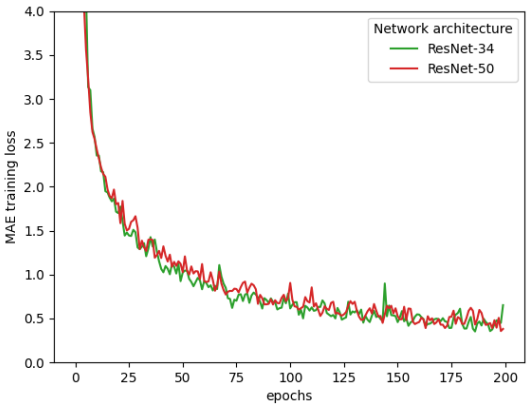
B. Additional training and validation loss of regression model



(a) Pre-training on ImageNet or plant subset

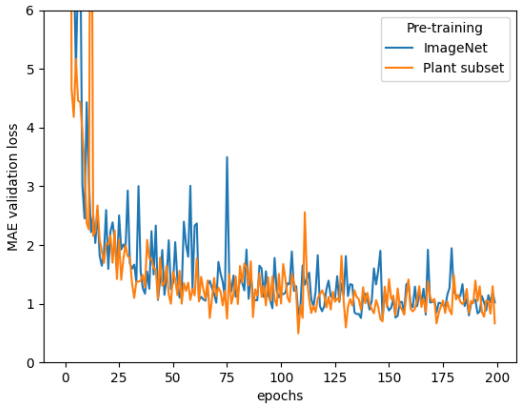


(b) Training dataset

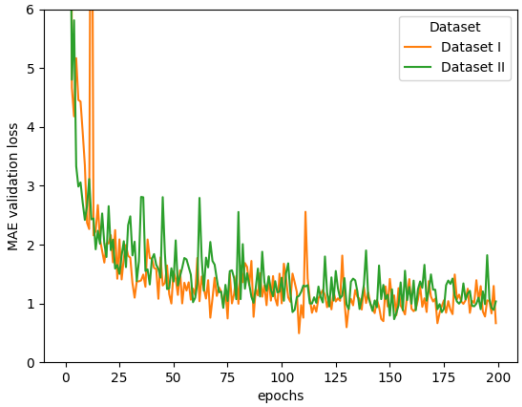


(c) The size of network architecture

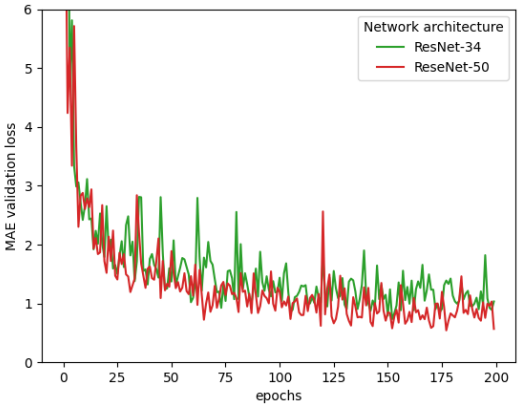
Figure B.1.: MAE training loss through the training period of 200 epochs across comparable experiments with experimental configuration parameters: pre-training mechanism, dataset, and network architecture.



(a) Pre-training on ImageNet or plant subset



(b) Training dataset



(c) The size of network architecture

Figure B.2.: MAE validation loss through the training period of 200 epochs across comparable experiments with experimental configuration parameters: pre-training mechanism, dataset, and network architecture.

C. Additional test result of regression task

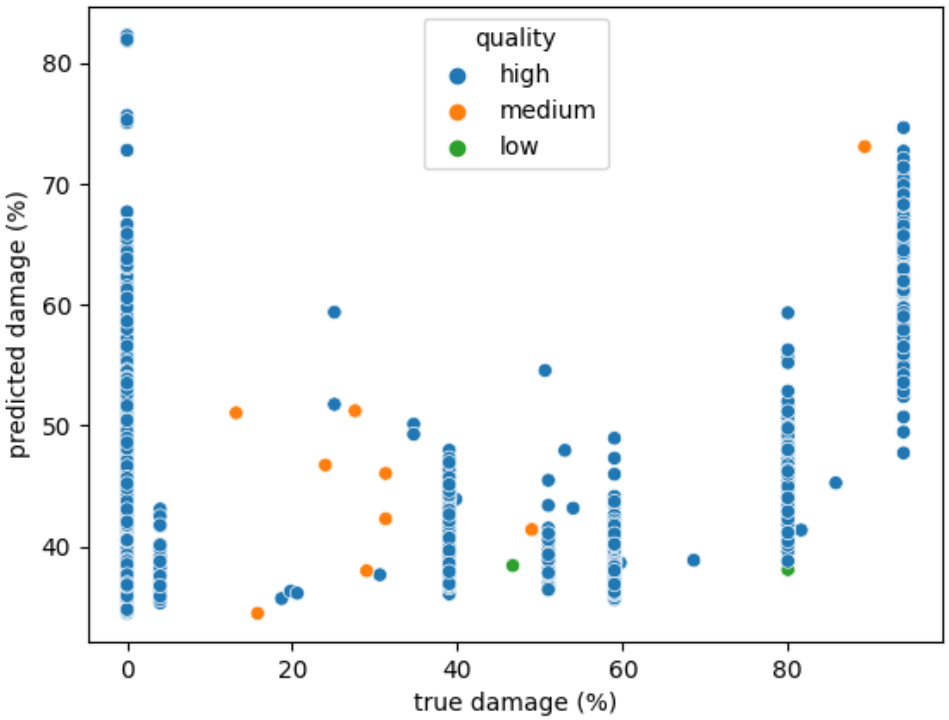


Figure C.1.: Test result with the model (none, Dataset I, linear). The circles depict the relation between true damage and predicted damage (Blue: high-quality image, orange: medium-quality image, green: low-quality image).

List of Figures

List of Tables

Bibliography

von Bloh, M. (2020). *Proof-of-concept for algorithm-based hail damage estimation in sugar beets*. Tech. rep. Chair of Digital Agriculture (TUM School of Life Sciences).

He, K., X. Zhang, S. Ren, and J. Sun (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Everingham, M., L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman (2010). "The Pascal Visual Object Classes (VOC) Challenge." In: *Int. J. Comput. Vis.* 88.2, pp. 303–338. URL: http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWWZ10.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei (2014). "ImageNet Large Scale Visual Recognition Challenge". In: cite arxiv:1409.0575Comment: 43 pages, 16 figures. v3 includes additional comparisons with PASCAL VOC (per-category comparisons in Table 3, distribution of localization difficulty in Fig 16), a list of queries used for obtaining object detection images (Appendix C), and some additional references. URL: http://arxiv.org/abs/1409.0575.

Mohanty, S. P., D. P. Hughes, and M. Salathé (2016). "Using Deep Learning for Image-Based Plant Disease Detection". In: *Frontiers in Plant Science* 7. ISSN: 1664-462X. DOI: 10.3389/fpls.2016.01419. URL: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419.

Kim, B., Y.-K. Han, J.-H. Park, and J. Lee (2021). "Improved Vision-Based Detection of Strawberry Diseases Using a Deep Neural Network". In: *Frontiers in Plant Science* 11. ISSN: 1664-462X. DOI: 10.3389/fpls.2020.559172. URL: https://www.frontiersin.org/articles/10.3389/fpls.2020.559172.

Yang, W., C. Yang, Z. Hao, C. Xie, and M. Li (2019). "Diagnosis of Plant Cold Damage Based on Hyperspectral Imaging and Convolutional Neural Network". In: *IEEE Access* 7, pp. 118239–118248. DOI: 10.1109/ACCESS.2019.2936892.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Ridnik, T., E. Ben-Baruch, A. Noy, and L. Zelnik-Manor (2021). *ImageNet-21K Pretraining for the Masses*. DOI: 10.48550/ARXIV.2104.10972. URL: https://arxiv.org/abs/2104.10972.

Simonyan, K., A. Vedaldi, and A. Zisserman (2013). *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. DOI: 10.48550/ARXIV.1312.6034. URL: https://arxiv.org/abs/1312.6034.

Garcin, C., A. Joly, P. Bonnet, A. Affouard, Lombardo, M. Chouet, M. Servajean, T. Lorieul, and J. Salmon (2021). "Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution". In: *NeurIPS Datasets and Benchmarks 2021*.

Nguyen, A., J. Yosinski, and J. Clune (2019). "Understanding Neural Networks via Feature Visualization: A survey". In: *CoRR* abs/1904.08939. arXiv: 1904.08939. URL: http://arxiv.org/abs/1904.08939.

I confirm that this idp final report is my own work and I have documented all sources and material used.

Munich, 14.10.2022 Yoonha Choe