CBS
COPENHAGEN
BUSINESS SCHOOL
HANDELSHØJSKOLEN

Multi-label Classification on Aerial Farmland Images

Image Classification based on the AlexNet CNN Architecture

Ramon Daniel Habtezghi (S149855)
Marten Jakob Stubenrauch (S149896)
Marin Elisabeth Henrica Mes (S149899)
Oskar Munck af Rosenschöld (S149873)

Department of Digitalisation
MSc Business Administration and Data Science
Data Mining, Machine Learning, and Deep Learning - KAN-CDSCO1004U

N.o. pages: 12
N.o. characters: 32 938 (15 standard pages)
Date of submission: 24th May 2022

Abstract

The COVID-19 pandemic and the ongoing conflict in Ukraine has caused significant supply chain disruption and resulting food supply shortages around the world. The importance of anticipating and ensuring vital supplies in an integrated and complex supply chain ecosystem is on top of the agenda for companies and world leaders. In this paper, we test how the prominent AlexNet architecture can be used to classify damages to agriculture fields. Due to need of agriculture operations to stay financially efficient we compared the standard AlexNet architecture with a modified version utilizing focal loss to counter class imbalance as such avoiding using a large dataset containing augmented pictures. We trained our model with over 76,000 pictures and deployed pre-processing steps such as using masks, boundaries, normalization, and scaling. We found that the modified AlexNet outperformed the standard model by being more time efficient and more capable based on the mean Intersection over Union metric. Additionally, the analysis shows that the modified version of AlexNet shows more capabilities to be improved even further. Thus, we can recommend for agriculture corporations to utilize AlexNet for their crop management to increase efficiency regarding damage detection.

Keywords: CNN, AlexNet, Focal Loss, Deep Learning, Image Classification

1. Introduction

The outbreak of the conflict between Ukraine and the Russian Federation has left a major footprint on the world. Beyond the considerable emotional impact, the conflict has also created disruptions to supply chains across the agricultural industry. Ukraine and Russia have been two of the primary suppliers of agricultural goods in the world, and ever since the start of the conflict, food supply shortages and subsequent food price increases have emerged (Baker, 2022). In the light of these developments, other countries are forced to step up by quantitively investing in their own food production, or by improving their current efficiency.

One way to achieve the latter is to improve farmers' ability to detect field damages and flaws. Detecting these as early as possible, enables them to recognize and fix those damages before it develops into something more severe (Fan et al., 2012). This task seems to be a good fit for machine learning algorithms, but it has shown to be inherently difficult for computers to accomplish (Kamilaris Prenafeta-Boldu, 2018). This paper tests the robustness of both neural and non-neural networks in identifying different damages to a field via flight and sentinel imagery. Since there are rising numbers of satellites and planes producing numerous pictures every minute, this technology could help farmers to improve production efficiency and thus be a strong support in battling global food supply shortages (Mang et al., 2020). We describe the training and testing of three algorithms using more than 76,000 images. We deploy two prominent Convolutional Neural Networks with distinct architectures which we benchmark against a Random Forest Classifier.

2. Related Work

The dataset used in this paper is the Agricultural-Vision Challenge dataset which comprises of more than 90,000 aerial images of farmland with important field patterns. This dataset has been used by numerous others to achieve state-of-the-art image classification by performing semantic segmentation, a method where each pixel is assigned a label, and thus allows for one picture to have multiple labels.

Park et al. (2020) use a Residual DenseNet with Squeeze-and-Excitation blocks (RD-SE) as the base model for semantic segmentation, a model based on the U-net encoder/decoder architecture. To compensate for the spatial loss arising during feature extraction in RD-SE they utilize skip connections and residual dense blocks, each with five convolution layers with kernel size 3x3 and batch normalization (Park et al., 2020). For the less frequent class objects they used expert networks to segment, a technique based on RD-SE but with a lighter architecture which speed up the training process. Their final prediction was based on a combination of the prediction results from the RD-SE and the expert networks.

Liu et al. (2020) on the other hand utilize a self-constructing graph module (SCG) in combination with a graph convolutional network (GNN) to perform semantic segmentation on the images. They utilize the pretrained ResNet50 as the backbone of their model. They rotate the images to extract the features at multiple views and they overcome class imbalance by designing an adaptive class reweighing loss function.

Huynh et al. (2020) tackle class imbalance by using focal loss as the objective function and they also perform data augmentation by random flipping and/or rotating each image. They use however a deep convolutional encoder/decoder architecture where the encoder is based on a pre-trained CNN called MobileNetV2 which uses an attention block to assign the contribution of each spectral challenge. The decoder module utilizes ASPP blocks and Squeeze-and-Excitation blocks to upsample the feature map to the original input size.

Barbosa Trevisan (2020) decide to tackle class imbalance by using focal loss and the Lovász-Softmax function. They use the ESP Net V2 with dropout layers as their base model and encoder/decoder architecture, and used the Adam optimizer to find the best value for the weights of their model. Baid et al. (2020) also use the Adam optimizer, but in combination with the Jaccard loss function and encoder-decoder architecture using EfficientNet.

Instead of a multi-class classification problem, Zhao et al. (2020) view the problem as independent binary segmentation tasks for each label type. They utilize the IBN-Net architecture with incorporated switchable normalization modules to reduce the feature divergence between the images which may occur while performing semantic segmentation.

Finally, Chiu et al. (2020) compare the performance of the DeepLabV3, DeepLabV3+ and a specialized FPN-based model, based on pretrained ImageNet. DeepLabV3 and DeepLabV3+ have proved to perform well across several semantic segmentation datasets, but since NRGB images are the input, Chiu et al. (2020) alter the DeepLabV3(+) models by duplicating the weights corresponding to the Red channel of the pretrained convolution layer, resulting in a convolution layer with four input channels. For the FPN-based model, similar to Park et al. (2020), Chiu et al. (2020) use ResNet as encoder, but they change the last residual block into a dilated residual block.

3. Methodology

3.1. Dataset description

The Agricultural-Vision Challenge dataset is a publicly available dataset consisting of 94,986 aerial images from 3,432 different farmlands across the United States. All images have a size of 512x512 pixels and have a resolution of 10cm per pixel. The images come in two forms: Red-Green-Blue (RGB) and Near-Infra-Red (NIR) and can be labeled as one or more of the following nine categories: double plant, dry down, endrow, nutrient deficiency, planter skip, storm damage, water, waterway and weed cluster. See Appendix A for an example of each category.

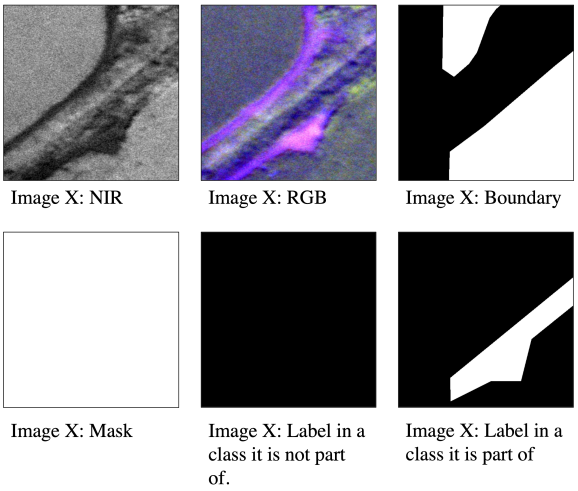


Figure 1: Example set of full complement of pictures

Besides the images, the dataset also consists of masks and boundaries for each picture. Masks are pictures containing black and white pixels where the black pixels indicate invalid pixels, and the white pixels indicate valid pixels. Boundaries are also pictures containing black and white pixels, but here the white pixels indicate what part of the picture the model should focus on, and the black pixels indicate what part of picture the model can ignore. So, for each picture there is an RGB version, a NIR version, a mask, and a boundary (figure 1).

The dataset has been split by Agriculture Vision in a train set (59.95 percent of all images), a test set (20.75 percent of all images) and a validation set (19.30 percent of all images). The train and validation set come with labels which work as follows: each image is represented in each category but only belongs to that category if there is a difference in pixels in the label image. To illustrate, if the image belongs to a certain class, the label image will consist of both black and white pixels for that specific class. If the image does not belong to a certain class, the label image will only consist of black pixels in that specific class. See figure 1 for an example. To make use of the labels, as a preprocessing step, we decided to convert the labels from black and white images to a binary form

only indicating the presence of a field condition.

Since the test set does not include labels, we decided to disregard it and create a new test and validation set out of the old validation set. This decreased the total amount of images from 94,986 to 76,652. See Appendix B for a visual representation of the train test validation split.

3.2. Training strategy

The models trained in this paper are an unmodified AlexNet, a modified AlexNet and a Random Forest Classifier, all trained with the same training set. To prepare the images for the models, we completed multiple pre-processing steps: data filtering, data normalization, data scaling and Principal Component Analysis (Random Forest Classifier only). We filtered out distorted images, applied the masks and the boundaries, and normalized and scaled the data to make it uniform along the entire pipeline. We performed Principal Component Analysis (PCA) for the Random Forest Classifier to decrease the complexity of our data, while retaining the patterns in the data. See Figure 2 for the full data pipeline.

3.3. Data preprocessing

3.3.1. Data filtering

All images have gone through an RGB check where the values of each color in the 512x512x3 array have been averaged to determine if any of the pictures have distorted values. Pictures where one of the three layers prompt an anomaly in the average value have been removed from the data set. Anomalies are defined as a picture with an average R, G, or B value of below 5 or above 250. In total, 1097 pictures were removed in the train set and 361 pictures were removed in the validation and test set.

3.3.2. Masks boundaries

To be able to use the images, masks, and boundaries as input, we convert them to arrays. This conversion yields sparse arrays containing ones for areas of interest and zeroes for areas to be disregarded. Each aerial image is multiplied with the array of its respective mask and boundary image and added into a new set of images to be further preprocessed. The sparse arrays ensure no additional image altering is conducted as the multiplication with the areas of interest gives the same pixel value as the original image. Similarly, the multiplication with the areas of non-importance gives pixel values of zero and are blacked out in the resulting image.

Using the masks and boundaries on the training set helps to improve and focus the learning of the model. We chose, however, to run the model on our test set without boundaries to stress the model's generalizability. We realized that it might be better to exclude boundaries in the test set since new input would not be easily annotated. The masks on the other hand we did apply to the test set because we considered it important to remove any invalid pixels.

3.3.3. Normalisation scaling

All images are scaled and normalized to bring uniformity to the pixel values. The pixel values are divided by 255 to scale them between 0 and 1 and reshaped into a 277 x 277 shape to fit the input of the AlexNet architecture and to decrease computational complexity. We use these same scaled and normalized images as input for the Random Forest Classifier for valid comparisons.

3.3.4. Principal component analysis (PCA)

Further preprocessing steps are implemented for the input to the Random Forest Classifier. Principal Component Analysis is used for dimensionality reduction and is suitable for tasks involving images. The dimensionality reduction works as a compression of the image and minimizes the size of the image while maintaining almost all variability from the original image. Initially the dataset had 230,187 features. By decreasing this to 4000 principal components, we preserve around 96.7 percent of the variance while decreasing the running time. This step will decrease the need for computational and memory resources.

3.3.5. Class imbalance

The dataset has substantial class imbalance as the categories "dry down" and "nutrient deficiency" are considerably more frequent than "storm damage" or "water" (see figure 3). One of the goals of this paper is to find the model that can handle the imbalance of the data the best.

The unmodified AlexNet model will not receive any changed input data to provide a benchmark for the modified AlexNet model. The latter model will use the Focal Cross Entropy Loss function which has proved to be suitable for multilabel classification of highly imbalanced datasets (Lin et al., 2018). We do not adjust the dataset for the random forest classifier as to build a benchmark and a base for comparison. The formula for the focal cross entropy loss is:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t)$$

In the case of our paper, we will use an alpha having the different weights of each class as such receiving an alpha weighted focal loss, and we

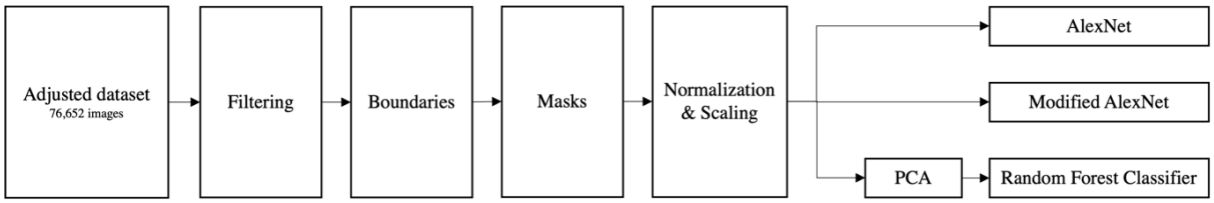


Figure 2: Data pipeline

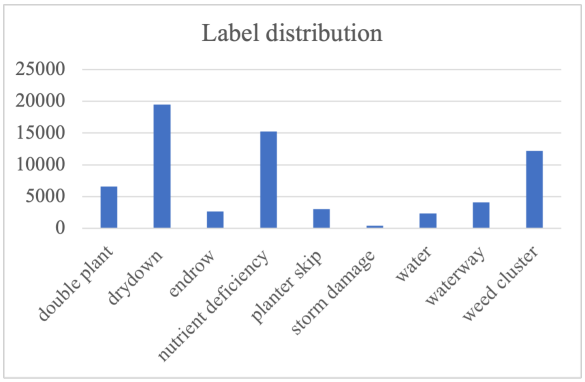


Figure 3: Label distribution in the complete dataset

will use a gamma equal to 2. Gamma controls the shape and how high the loss is for well-represented examples. As such, with a high gamma we can turn the attention of the model to hard-to-classify examples (Lin et. al., 2018).

3.4. Models and architecture

3.4.1. Convolutional Neural Network based on the AlexNet Architecture

The basic components of any convolutional neural network (CNN) include convolutional layers, pooling layers, and fully connected layers. The convolutional layers are the main reason as to why CNNs are suitable as networks for input consisting of large matrices, like images. The convolutional layers work through passing a mathematical function across the matrix and assigning the output of that function to a new, smaller matrix called a feature map (Géron, 2019). The feature map is then passed as input to the next layer. The pooling layers down sample the feature maps and give the model the ability to handle spatial invariance, i.e., giving the model the capability to classify images that are different from the ones seen in training (Géron, 2019). The AlexNet architecture employs the commonly used max pooling function in its pooling layers which preserves the detected features in a good way (Géron, 2019). Finally, the fully connected layers operate on a flattened input and are included in the purpose of optimizing objectives like for example assigning class scores to finally be presented in the output layer (Géron, 2019).

The AlexNet architecture uses a strategy of stacked convolutional layers where after the data has passed through two convolutional and Max-Pool layers, it enters three stacked convolutional layers without any pooling layers in between them (Krizhevsky et. al., 2012). The three stacked convolutional layers use smaller filters and lower strides than the previous convolutional layers to adapt to the smaller input sizes. To combat the vanishing gradient problem, batch normalization layers are added in five places as well as making use of the Rectified Linear Unit as activation function in the hidden layers. In our implementation of the AlexNet we use Batch Normalization instead of the originally proposed Local Response Normalization (Krizhevsky et. a., 2012) as the latter has later been found to have small to no contribution while adding complexity (Gao Wi, 2022). Stochastic gradient decent is used as an optimizer and binary cross entropy loss as a loss function. The AlexNet models used in this article have a total of 71 955 209 parameters. The full architecture is visualized in appendix C.

3.4.2. Modified AlexNet CNN architecture

The AlexNet CNN architecture is inherently slow to train and shows a possible weakness to multi-label classification because of the substantial class imbalance in the dataset. Given these conditions we modified the AlexNet architecture based on previous literature. Firstly, we used the Focal Cross Entropy loss function since it has proved to handle high imbalanced datasets, however, was not yet tested in conjunction with the AlexNet CNN architecture. Focal Cross Entropy Loss penalizes mislabeled well-represented classes and provides the model with the ability to also label pictures with a rarer damage (Tsung-Yi et al., 2017). The second modification is the introduction of momentum to the stochastic gradient descent optimizer. As the AlexNet has many layers, we aimed to increase the training speed by introducing Nesterov accelerated learning. This momentum update to the stochastic gradient descent provides an ex-post check of the descent to check if the direction is advantageous.

As a third modification we introduced a scheduled learning rate and added a decay rate to optimize the learning further.

3.4.3. Random Forest Classifier

To benchmark our convolutional neural network architectures using a non-neural approach we used a Random Forest Classifier (RFC). The RFC can perform multi-label classification and is thus a better alternative over other classifiers like Support Vector Machines for this dataset. The idea of the model is to create a collection of decision trees and to use bootstrapping to collectively decide and predict the label of the picture. To get the most suitable parameters in the Random Forest model, we performed a random grid-search as it is faster than the other grid-searches with a 5-fold cross-validation. Based on the grid-search result, we choose the entropy criterion with the suggested hyperparameters. Before putting the input into the RFC, we must transform the images from having three dimensions to two dimensions, since this is the required input for the RFC. We converted the RGB dimensionality by stacking the RGB dimensions to avoid losing spatial information.

3.5. Evaluation metrics

The scope of multi-label classification presents certain limitations on popular evaluation metrics such as precision, accuracy, and recall. For example, a prediction containing a subset of correct classes while misclassifying some classes should be interpreted as a better result than not predicting any, or less instances of, correct classes. Depending on the nature of the dataset there are different approaches to mitigating these limitations, two of them being micro-averaging and macro-averaging.

Micro-averaging entails taking the sum of the individual true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) for each class and calculating the average. Due to the heavy imbalance observed in between classes it would give greater emphasis on the best represented classes which, depending on model performance for said classes, can skew the results. A better method of getting overall model performance in terms of F1-score is macro-averaging. Then the straight average of the metric is calculated, and class imbalance is mitigated as it disregards the individual classes' support values. To have a balanced view of the model performance and to mitigate the inherent trade-off between precision and recall, the macro-averaged F1-score could be used as one evaluation metric (Géron, 2019). However, as F1-scores become low if either precision or recall are low, the interpretability of the aggregated score for the whole model would also diminish with a low value for either one of these. Therefore, we have chosen not to use the F1-score as a metric in our evaluation.

Instead, the mean Intersection-Over-Union metric (mIoU) was chosen as evaluation metric. It gives a good indication of overall model performance and rewards predictions that heavily overlaps with the actual values (Rezatofighi et al., 2019).

$$mIoU = \frac{1}{n}\sum_{i=1}^{n}\frac{TP}{FN + FP + TP}$$

In the above equation, n equal the number of classes. The mIoU is preferred over the Exact Match ratio in this paper as it does not penalize incorrect labels as harshly. The Exact Match ratio gives the number of instances with an exact label match over the total number of instances. So, a model prediction that correctly identifies three of four field conditions in an image is not deemed correct and the ratio decreases. It's a very strict evaluation metric and not suited for the number of labels and complex dataset in this paper.

4. Results

4.1. General

The test results show that the modified AlexNet model is outperforming the other models with a mIoU of 0.4410. The second-best model is Random Forest Classifier with a mIoU of 0.4408 and the worst performing model is standard AlexNet with an mIoU of 0.4407 (table 1).

Model	mIoU
mod AlexNet	0,4407
AlexNet	0,4410
RCF	0,4408

Table 1: mean Intersection-over-union score of the three models

Figure 4 shows that the class imbalance is also present in the test set. The fact that the modified AlexNet is outperforming the other models is therefore expected, since it is the only model that deals with class imbalance.

The training and validation loss graphs show that the standard AlexNet model is slightly overfitting at the end of our 50 epochs, as the validation loss is increasing slightly while the training loss is still decreasing (figure 5). This behavior indicates that AlexNet might face difficulties of generalizing on new data. The training and validation loss graphs for the modified AlexNet show that both validation loss and training loss are still steadily

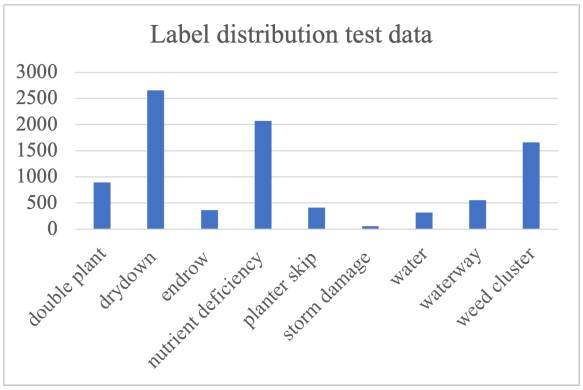


Figure 4: Label distribution over the test data

decreasing and not yet stabilized after 50 epochs (figure 6). This indicates that we could increase epochs to reap better results.

4.2. Complexity and run time analysis

When assessing our models, we also considered the training time of the models because sometimes a less accurate model might be beneficial if time can be saved in the training process. All three models have been executed on a machine with 64 vCPU and RAM of 376 GB on the Ucloud platform. The running time of the different models is visualized in Figure 7.

This figure shows that AlexNet has the longest runtime compared to the other models. Our proposed modifications to the AlexNet cut down the runtime by one hour compared to the standard AlexNet. Given that the modified AlexNet has the highest m-IoU, it is the more effective from the CNNs. The running time for the Random Forest Classifier is however much shorter than the two neural networks, and at the same time only differs slightly from the two AlexNet models in mIoU. This suggests its suitability as the most effective model. However, it must not be forgotten that, as suggested earlier, the modified AlexNet might become more accurate when it is given more training time.

5. Discussion

The results suggest that the modified AlexNet is the strongest model as it handles the class imbalance well. It is however not outperforming the other models as much as expected. Guided by Figure 7, we believe that if we had invested more time in optimizing the AlexNet further, the model might have been improved and might have showed stronger outperformance. A possible extra modification would be to include a different activation function such as the sigmoid, since this has proven

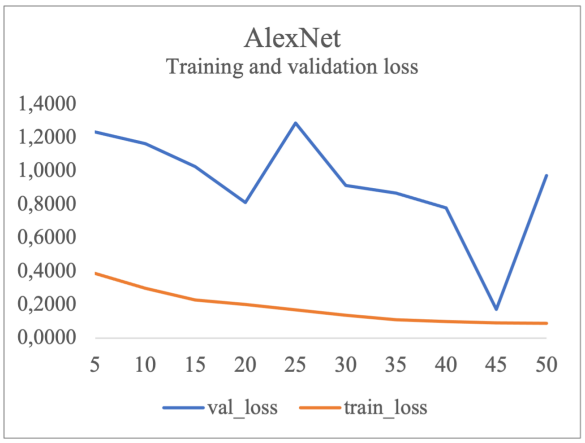


Figure 5: AlexNet Training and validation loss over epochs

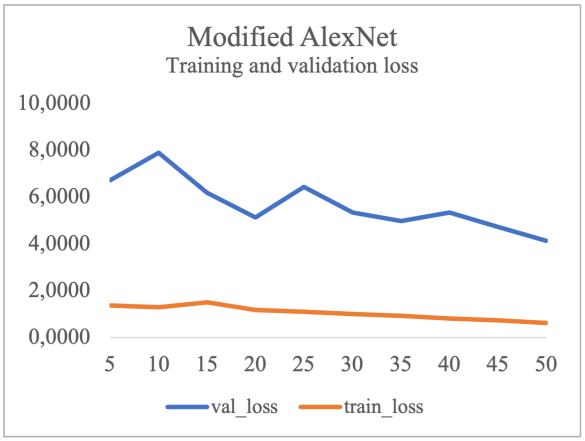


Figure 6: Modified AlexNet training and validation loss over epochs

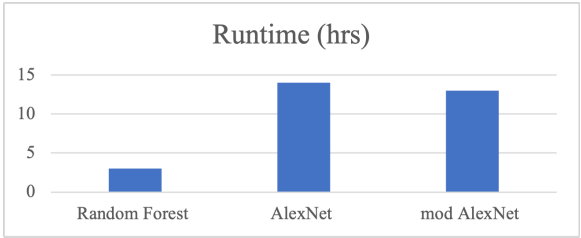


Figure 7: Runtime of the three models

to be useful for multi-label labeling. A second adjustment could be to increase the number of epochs since the validation mIoU within the training was up at 0.5207, suggesting that with more training epochs the model might improve further.

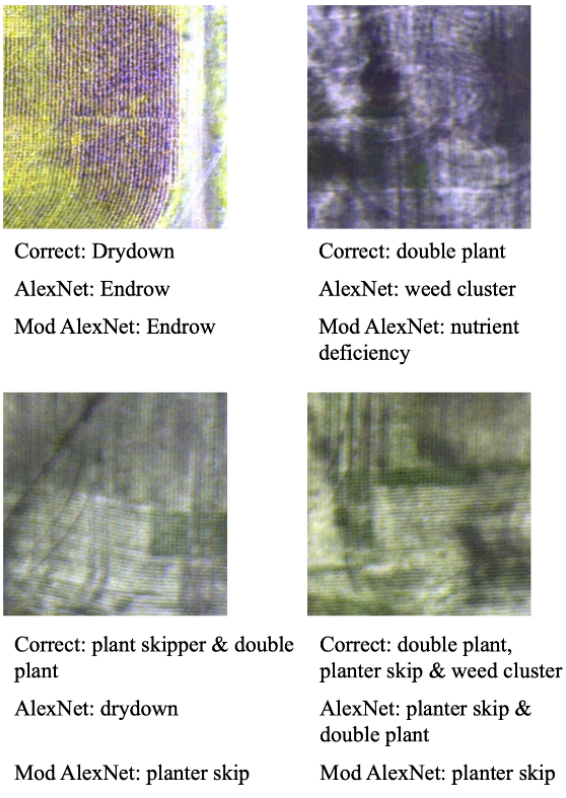
Our results contribute academically by showing that a simple Random Forest Classifier would be a computational efficient algorithm to spot damages to agricultural fields. Moreover, the fact that the AlexNet architecture displays promising results when using Focal Loss to counter class imbalance within computational power restrictions, is a notable academic contribution. The strength of the modified AlexNet in this case lies in its errors, as it always correctly identified one damage to the field, even for the less common categories. Given that many agriculture operations do not have strong computers at their disposal, use of the focal loss function together with the AlexNet architecture might show benefits and help optimize crop management.

For future work, it would be interesting to see how far data augmentation would help in building a more efficient model. Due to the strong level of imbalance, only small data augmentation through flipping or graining badly represented pictures might already help the algorithms to become more capable and show stronger performance. Considering all the above, we suggest in the scenario of less computational power to use the Random Forest, and in case of more computational resources to continue training the modified AlexNet until the validation loss stops decreasing.

5.1. Error analysis of CNN models

Out of 8,987 pictures, the CNN model based on the unmodified AlexNet architecture had the least wrong predictions, followed by the CNN model based on the modified AlexNet and then the Random Forest. The errors include completely wrong predictions where no label is guessed correctly, but also partly wrong predictions where for instance only two out of three categories were classified correctly. In this section we will investigate some of the wrong predictions and explain why each failed. We focus on the errors of the two neural networks in this section as these show the biggest room for improvement.

Figure 8 shows an overview of four wrongly classified images by the standard AlexNet model and the modified AlexNet model. We can see that many of the wrong predictions are due to bad image quality. Weed clusters and double plant are quite similar in their appearance and are already hard for humans to correctly identify. This, in combination with bad image quality, clearly challenges the algorithm. While going through the results, it was found that



ht

Figure 8: Errors of CNN models

These two categories miss classified categories as correct classification of weed clusters is often only the case if the field is empty. Thus, the weights might be biased for searching for dark areas surrounded by light areas indicating an empty field with weed cluster. What was surprising is that pictures with a shadow casted by a cloud or other flying objects did not pose a significant challenge to the algorithm and pictures when classified correctly.

5.2. Limitations

5.2.1. Scope

The scope of this paper should be viewed as an explorative study in how certain CNN architectures perform multi-label classification on agricultural fields. The chosen preprocessing structure to convert the labels from black and white images indicating the location of the field damage, to a binary form only indicating the presence of a field condition, limits the scope of this study to a multi-label classification issue. Related work has exclusively attempted to perform semantic segmentation to fulfil the task both identifying and localizing field conditions in an image. For this specific research however, using a different approach to ensure pixel position integrity would have increased the needed computational resources, and were therefore defined to be out-of-scope. We realize this limits the scope of the paper and that the comparability of the results to that of related work is diminished. The study however set a strong foundation for future iterations of this paper with larger scopes.

5.2.2. Image usage

The second limitation of this paper concerns the choice of only utilizing the RGB version and to disregard the NIR version of the images for the input. Including the NIR images would have added an additional layer to the input (in addition to the RGB, masks and boundaries) and therefore could have improved the quality of the input, and respectively, the performance of the models. The inclusion of the supplementary images would have, however, put further strains on the computational and memory resources available for this paper and has therefore been excluded. Despite this, it is something to consider in future work.

5.2.3. Data augmentation

Other limitations of this study concern the lack of data augmentation. Even though we used a vast dataset, the varying nature of field conditions in agriculture and the high degree of heterogeneity in field damage remains a general challenge. Image augmentation such as randomly flipping, rotating, and altering color shades would have been a realistic means to remove some of this uncertainty. The choice to not perform these data augmentation steps stems from the previously mentioned constraints on computational resources in this study. It could perhaps have been mitigated by performing stratified sampling to construct a smaller training set, allowing for the execution of the proposed augmentation tasks. This technique is to be considered in future iterations of this study in order to deal with the inordinate heterogeneity of the subject matter.

6. Conclusion

In this paper we explore the robustness of the convolutional neural network AlexNet architecture in performing multi-label classification for identifying field conditions in agriculture. We train and test a rather standard AlexNet based model and a modified AlexNet based model and compare this to a non-neural Random Forest Classifier. We find that modified AlexNet model outperforms the other two models, but that the differences in performance are minimal.

The modified AlexNet shows signals for applications within the domain as the modifications to the loss function and the addition of momentum to SGD enables training on imbalanced datasets and decreases the computational resources needed. Despite its similar performance results to the other two models, a longer training time suggests possibilities for future improvements and operationalization. Provided greater computational resources, further data preprocessing steps, and/or increased training time, an improved model performance and a stronger ability to compare the impact of different preprocessing pipelines could be achieved. Besides overcoming this papers limitations, future work can utilize the, in this paper, seen robustness of the AlexNet architecture in agricultural image classification to further augment preprocessing steps, increase the scale of training, or derive new use cases.

7. References

Baker, A. (2022, April 29). The Ukraine Food Price Crisis is Just a Preview of What Could Happen as Climate Change Worsens. Time. Retrieved on May 17, 2022, from https://time.com/6172270 /ukraine-food-price-crisis-climate-change/

Chiu, M. T., Xu, X., Wang, K., Hobbs, J., Hovakimyan, N., Huang, T. S., Shi, H. (2020). The 1st agriculture-vision challenge: Methods and results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 48–49).

Chiu, M. T., Xu, X., Wei, Y., Huang, Z., Schwing, A. G., Brunner, R., ... Shi, H. (2020). Agriculture-vision: A large aerial image database for agricultural pattern analysis. In Proceedings of the IEEE /CVF Conference on Computer Vision and Pattern Recognition (pp. 2828–2838).

Fan, M., Shen, J., Yuan, L., Jiang, R., Chen, X., Davies, W. J., Zhang, F. (2012). Improving crop productivity and resource use efficiency to ensure food security and environmental quality in China. Journal of experimental botany, 63(1), 13–24.

Gao, R. Wu, J. (2022). Convolutional Neural Networks (CNNs / ConvNets) [Unpublished manuscript] CS231n Convolutional Neural Networks for Image Recognition, Stanford University, Retrieved on May 22, 2022, from https://cs231n.github.io/convolutional-networks/

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd edition, O'Rielly: Sebastopol

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, Silvio Savarese; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 658-666

Kamilaris, A., Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and electronics in agriculture, 147, 70–90.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

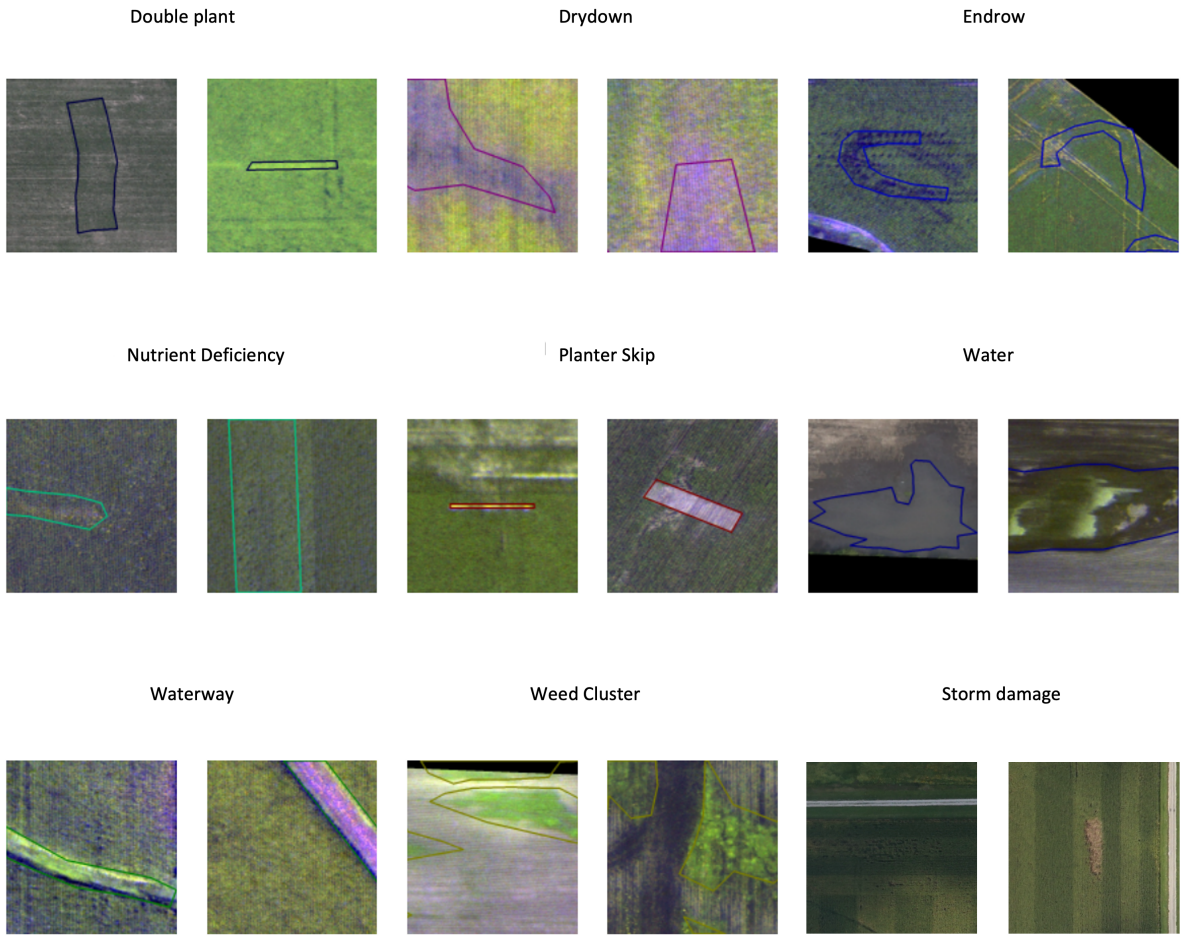
Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P. (2018) Focal Loss for Dense Object Detection, Facebook AI Research (FAIR), Retreived on May 23 2022 from https://arxiv.org/pdf/1708.02002.pdf

Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

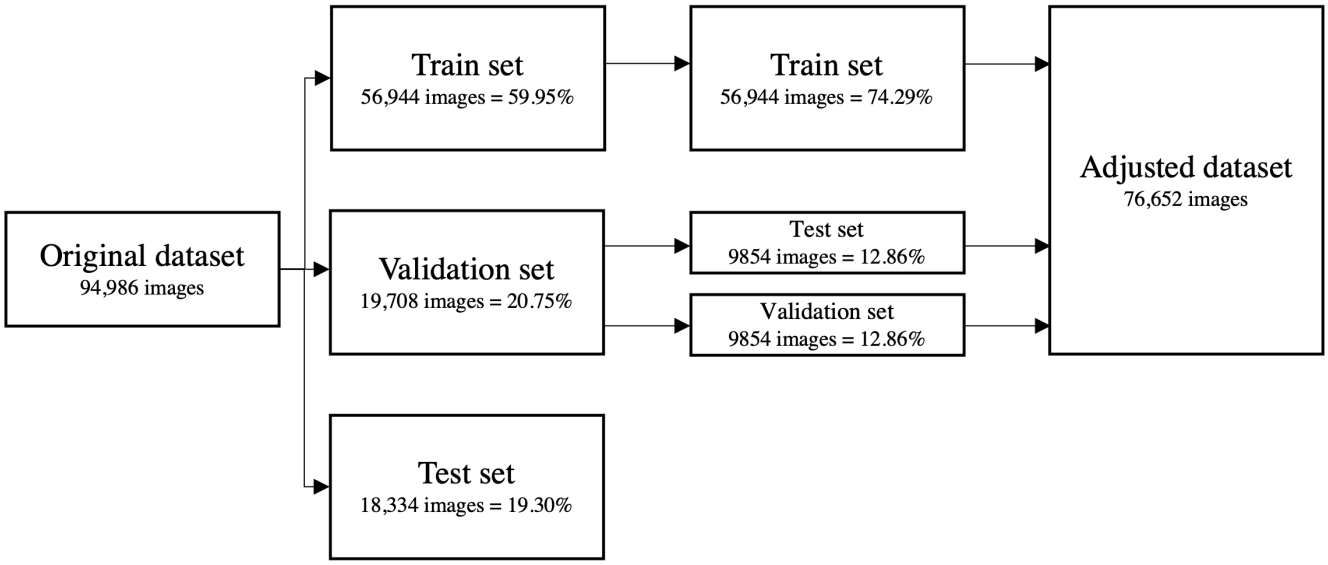
Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. IEEE transactions on pattern analysis and machine intelligence.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ´ Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017. 3, 6

Appendix A. Example images from dataset



Appendix B. Train Test Validation Split



Train Test Validation Split from original Agriculture-Vision dataset to the one used in this paper

Appendix C. Mod AlexNet and AlexNet Architecture