Proceedings of the 9th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'23)
Brunel University, London, United Kingdom - August 03-05, 2023
Paper No. MVML 112
DOI: 10.11159/mvml23.112

Deep Learning Models for Wheat Diseases Detection and Recognition

Chekir Amira[1], Goodhope-Kudakwashe Dhliwayo[2], Fanyana-Jushua Dube[2]
[1,2,3] Universite des sciences et de la technologie Houari Boumediene
Algiers, Algeria
achekir@usthb.dz

Abstract - Wheat is a grass widely cultivated for its seed, a cereal that is a staple food around the world. However, cereal wheat is subject to many wheat diseases, including bacterial, viral and fungal diseases, as well as parasitic infestations. The need to use Deep Learning methods to identify automatically wheat diseases has become a challenge. In this paper, we proposed and compared two models based on Convolutional Neural Network (CNN) for wheat diseases detection and recognition. The convolutional layers of a CNN can be considered as matching filters derived directly from data images (images of healthy and unhealthy wheat). CNNs thus produce a hierarchy of visual representations optimized for our task. As a result of CNN training, a model is obtained - a set of weights and biases - which then responds to the specific task for which it was designed. One of the main strengths of CNNs is their ability to generalize, that is, the ability to process data never seen before. This allows a certain robustness to the heterogeneity of the background, to the image acquisition conditions and to the intra-class variability. A large image dataset of various wheat diseases, including healthy wheat, was used for training our models to learn, recognize and detect diseases and/or abnormalities in wheat.

Keywords**:** Deep learning, CNN, wheat diseases, classification

1. Introduction

Cereal wheat is subject to many wheat diseases. The main diseases in temperate environments are as follows, in rough order of their impact from colder to warmer climates: eyespot, stagonospora nodorum leaf spot (also known as glume leaf spot), rust yellow or streaked, powdery mildew, septoria spot tritici (sometimes known as leaf spot), brown or leaf rust, fusarium head blight, brown spot, and black rust. In the tropics, spotted leaf spot (also known as helminthosporium leaf blight) is also prominent. These diseases according to the Food and Agriculture Organization of the United Nations (FAO) cause a loss from 20 % to 40 % of world food production, constituting a major threat to food security. On the other hand, the impact of climate change has resulted in increased changes in humidity, temperature and wind direction thus affecting wheat yield. Sometimes these changes, like the wind, can make it easier to spread diseases like leaf rust, stem rust, and yellow rust [1].

On the other hand, deep neural networks (DNNs) are applied in many practical problems. Thanks to their ability to learn, based on searching for similarities between objects and their generalization, they are able to deal with problems where a very accurate classification is required. Image recognition is one of the tasks in which DNN excel.

In the recent years, researches used deep learning methods for automatic wheat diseases identification. We established in table1 the summary review of previous works [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. In our review study, we considered the following research questions: (a) the problem they answered, (b) the approach used, (c) the data sources used and (d) the obtained accuracy. We also recorded: (e) whether the authors compared their CNN-based approach with other techniques, and (f) what was the difference in performance.

This paper presents and compares two models for wheat diseases automatic detection and classification. We designed our two modesl, which we named Model_1 based on CNN, and another Model_2 based on the architectures of EfficientNetB7 [13]. Our models used transfer-learning of CNNs. Training and validation of our models were performed on a large wheat database built in this study.

Table. 1: Summary of the literature review

	Agriculture field	Description of the problem	Data used	Precision	CNN model used	Comparison with other techniques	Ref
1	Detection of plant diseases	Identify 14 crop species and 26 diseases	PlantVillage's public dataset of 54,306 images of diseased and healthy plant leaves	0.994 (F1)	AlexNet, GoogleNet	Substantial margin in standard benchmarks with approaches using hand-designed features	[2]
2		Classify banana leaf diseases	Dataset of 3,700 banana disease images obtained from PlantVillage Dataset	More than 96 % (CA), 0,968 (F1)	LeNet	Methods using artisanal functionalities are not well generalized	[3]
3		Deep learning models for the detection and diagnosis of plant diseases	Dataset of 87,848 images containing 25 different plants in a set of 58 distinct classes.	99.53% accuracy	Defined by author	Several CNN models have been trained, with the best performance reaching a success rate of 99.53%.	[4]
4		Identifying Rice Diseases Using Deep CNN	Dataset of 500 natural images of diseased and healthy rice leaves and stems, for 10 common rice diseases	95,48% accuracy	Author-defined CNN-based model	N/A	[5]
5		On the use of transfer learning for the detection of plant diseases	PlantVillage dataset which contains 38 classes of 54,305 images of sick and healthy diseases for 14 species	0.94 (F1)	ResNet50	Better F1 scores than Inception-v3 (0.93), MobileNet (0.93), DenseNet169 (0.93)	[6]

6	Leaf disease detection	13 different types of healthy plant and leaf diseases	Database created by the authors containing 4483 images	96.30%(CA)	CaffeNet	Better results than SVM	[7]
7		Recognition of corn diseases based on EfficientNet by classification of leaf images	9279 images, with 6496 images for training data and 2783 for validation	98.52% accuracy and 1.48% loss	EfficientNet-b0 transfer learning	Data increase due to small data size. Better results than ResNet50 (96.76%), Inception-v3 (93.9%), VGG16 (96.35%)	[8]
8	Fruit counting	Predict the number of tomatoes in pictures	24,000 synthetic images produced by the authors	91% (RFC) 1.16 (RMSE) on real images, 93% (RFC) 2.52 (RMSE) on synthetic images	Inception, ResNet	Area Based Teachnic (ABT) (66,16% accuracy), (RMSE = 13.56)	[9]
9		Detection of peppers and melons	122 images obtained from two modalities: color (RGB) and Near-Infred (NIR)	0.838(F1)	Fast Regional CNN with VGG16 model	Conditional random field to model visual color and texture characteristics (F1 = 0.807)	[10]
10	Weed identification	Identify the thistle in the images of winter wheat and spring barley	4,500 images at 10, 20, 30 and 50 m altitude captured by a camera Canon PowerShot G15	97,00 % (CA)	DenseNet	(Based on the color characteristics) Thistle-Tool (95%)	[11]
11		Pixel-close classification of weeds and crops in images using a fully CNN	301 soil images and 8,430 segmented plant images, Plants cover 23 different weed and corn	94% accuracy, 100% detection rate for corn and weeds	Modified version of the VGG-16 and transfer learning	N/A	[12]

2. Materials and Methods

2.1. Wheat Disease Datase

Our dataset consists of a total of 7,540 images, divided into 8 classes - 7 wheat diseases classes and one class for healthy wheat images. The dataset englobes main diseases are as follows: stagonospora nodorum leaf spot (also known as glume leaf spot), rust yellow or streaked, powdery mildew, septoria spot tritici (sometimes known as leaf spot), brown or leaf rust, fusarium head blight, and loose smut. This dataset has been compiled from existing datasets namely PDDP - images datasets, Kaggle, Google Drive - Large Wheat Diseases Classification and Dataset. Our dataset was divided as follows: 80\% for training models, 30\% for validation during training and we chose randomly 36 images for each class, to test the model's performance on new images data after training. This is summarized in Table 2.

Table 2: Description of the wheat disease dataset

	Healthy	Fusarium Head Blight	Leaf Rust	Loose Smut	Powdery Mildew	Spetoria Leaf Blotch	Stem Rust	Yellow Stripe Rust	Total
Training	894	251	1214	314	410	652	646	976	5252
Validation	416	124	505	134	158	295	258	389	2252
Test	4	5	8	5	7	4	5	4	36
Total	1324	380	1727	453	575	951	909	1360	7540

2.2. Image Preprocessing

To input the image dataset into the CNN model, the images had to be the same size (height and width) and the same format as the input image size required for each of our networks. To fit our images into the deep learning models, we converted each image to a three-dimensional tensor, that is, a length of 224, a width of 224, and a color channel of RGB. We used the Python Imaging Library (PIL) to do these conversions. On the other hand, EfficientNet, which uses noisy-student [14] type weights, does not require any preprocessing of the input images. Noisy Student is a semi-supervised learning approach that extends the idea of self-training and knowledge distillation.

2.3. Proposed Method

We proposed two methods to learn wheat disease from the training database. Each method established a model, and used it to classify wheat images primarily into two main classes - healthy and unhealthy wheat. The unhealthy wheat images was then classified according to their respective disease type.

2.3.1 Design of a transfer-learning model based on pre-trained models

We have chosen a number of different pre-trained CNN models to use as the basis for our transfer learning models. We explored models such as InceptionV3 [16], VGG16 [17], VGG19 [17], EfficientNet (B0 - B7) [13], and MobileNet [18]. We performed fine-tuning, added new layers, removing some layers and freezing layers on the pre-trained base models while doing hyperparameter searches to determine the number of layers to remove or freeze. We will only present the models that gave the best results based on the pre-trained EfficientNetB7 models.

2.3.2 Model_1

We created a sequential CNN model as a starting point from the simple model provided on the TensorFlow and added a few more layers to deepen the network. The summary of the model is given in Table 3. All parameters of all layers are trainable. The number of outputs (8) in the last dense layer (Dense_3) corresponds to the number of classes in our dataset. We also used the sigmoid activation function for the last dense layer. Our model can have several correct answers and the sigmoid gives the probabilities for an image to belong to each of the classes, that is why we chose this one, that is to say that an image of a wheat plant may have one or more diseases present.

2.3.3 Model_2

Noisy-student specific pre-trained weights exist for the Efficient Nets from B0 to B7, so we used the noisy-student-b7 weights to train our EfficientNetB7 model. We used the pre-trained weights from noisy-student-b7 and set all the layers so they could be trained. We have added a pooling layer of the overall average. The last layer is a dense layer with 8 as the number of outputs and a sigmoid activation function. The summary of the simplified model is presented in Table 4.

Table 3: Model_1 Summary

Layer	Type	Kernels	Kernel size	stride	Activation	Parameters
0	Conv2D (Input)	16	3*3	'same'	ReLU	448
1	MaxPooling2D	-	2*2	'default'	-	0
2	Conv2D	32	3*3	'same'	ReLU	4640
3	MaxPooling2D	-	2*2	'default'	-	0
4	Flatten	-	-	-	-	0
5	Dense_1	512	-	-	ReLU	51380736
5	Dropout	512	-	-	-	0
6	Dense_2	256	-	-	ReLU	131328
7	Dense_3	8	-	-	sigmoid	2056
Total params: 51,519,208 Trainable params: 51,519,208 Non-trainable params: 0						

Table 4: Model_2 Summary

Layer	Type	Kernels	Activation	Parameters
0	ResNet50(base model)	7*7*2560	-	64097687
1	Global Average Pooling	2560	-	0
3	Dense	8	sigmoid	20488
Total params: 64,118,175 Trainable params: 63,807,448 Non-trainable params: 310,727				

3. Experimental Results

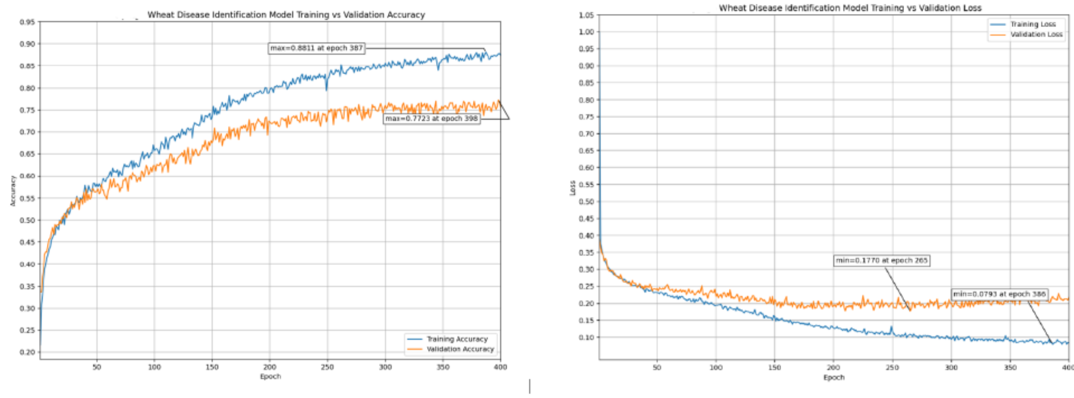
3.1 Training Results

3.1.1 Model_1

The TPU training took about 45 minutes to complete the 400 epochs and achieve convergence. As shown in Figure 1-a and Figure 1-b, training resulted in unstable incremental improvement. The precision and loss of validation ceased to have significant improvement after 300 and 175 epochs respectively. The accuracy and training loss seem to keep improving, but we stopped the model anyway because we are interested in the validation values as they can indicate whether an overfit or underfit has occurred. The precision and the validation loss obtained are respectively 0.7723 and 0.0793.

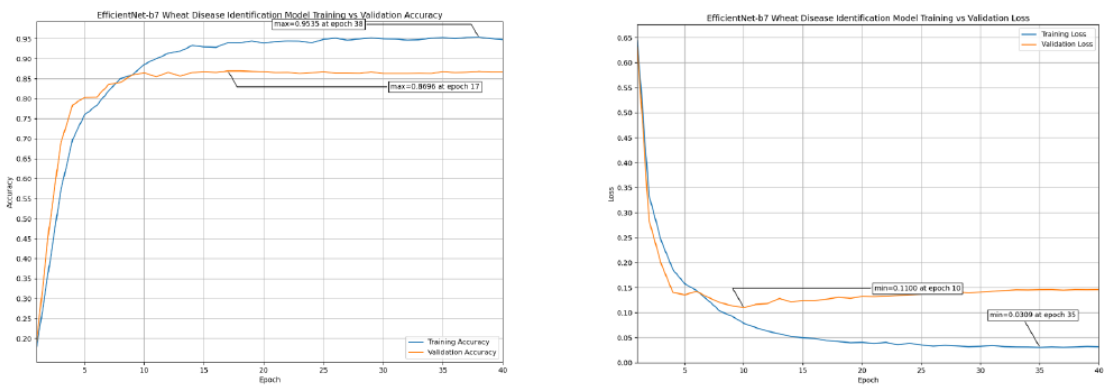
3.1.2 Model_2

The model took 40 epochs to converge. The TPU training took about 7 minutes. As the precision and loss graphs below show, training had an unstable incremental improvement. The precision and loss of validation stopped improving significantly after 15 and 10 epochs, respectively. The accuracy and validation loss achieved were 0.8696 and 0.1100 respectively. Figure 2 shows the training analysis of model_2.



(a) Model_1 Accuracy (b) Model_1 Loss

Fig. 1: Training analysis of Model_1.



(a) Model_2 Accuracy (b) Model_2 Loss

Fig. 2: Training analysis of Model_2.

3.2 Classification Results

3.2.1 Model_1

This model achieved a validation precision (CA) of 0.7723 during training. The results obtained in reported in Table 5, observations and interpretations are summarized in Table 6.

3.2.2 Model_2

This model achieved a validation precision (CA) of 0.8696 during training. The results obtained in reported in Table 7, observations and interpretations are summarized in Table.

Table 5: Model_1 Classification Results

	Precision	Recall	F1-score	Validation Data	Train Data
Fusarium Head Blight (a)	0.6693	0.6855	0.6773	124	251
Healthy Wheat (b)	0.8470	0.9183	0.8812	416	894
Leaf Rust (c)	0.7316	0.7663	0.7485	505	1214
Loose Smut (d)	0.8560	0.7985	0.8263	134	314
Powdery Mildew (e)	0.7045	0.3924	0.5041	158	410
Septoria Leaf Blotch (f)	0.5796	0.6542	0.6146	295	652
Stem Rust (g)	0.6699	0.8101	0.7333	258	646
Yellow Stripe Rust (h)	0.6503	0.8843	0.7495	389	976
Micro avg	0.7093	0.7762	0.7413	2279	5252
Macro avg	0.7135	0.7387	0.7168	2279	5252
Weighted avg	0.7142	0.7762	0.7376	2279	5252
Samples avg	0.7576	0.7811	0.7628	2279	5252
CA = 0.7723					

Table 6: Observations and interpretations for Model_1

Observations	intepretations
Micro avg > Macro avg for recall and F1	- Bias towards the most populous classes.
Micro, Macro avg recall > Macro, Micro avg precision	- The model tends to find as many positive instances as possible rather than returning the predicted positive instances.
Class (a) has the worst precision, recall and F1 scores.	- It is the least populated class - Lack of more data images prevented the model from fully learning this class during training
CA > Micro, Macro , Weighted avg for précision	- The model has more accuracy than precision
The precision is significantly lower than the recall for classes (e), (f) and (g).	- An indication that the model has less predictive power for these classes but that it is more efficient in predicting positives which are currently positive (than predicted positives).
AUC-ROC significantly lower than AUC-ROCs of Macro avg and Micro avg.	- The probability that the model ranks a randomly chosen positive instance higher than a randomly chosen negative instance is low for these classes.
The ROC curves for classes (c), (e), (f) are well below the ROC Micro avg and Macro avg curves.	- These classes will have significantly lower predictive accuracies at different **FPR** or fasle alarm thresholds.

Table 7: Model_2 Classification Results

	Precision	Recall	F1-score	Validation Data	Train Data
Fusarium Head Blight (a)	0.7171	0.8790	0.7899	124	251
Healthy Wheat (b)	0.9010	0.8966	0.8988	416	894
Leaf Rust (c)	0.6731	0.8970	0.7691	505	1214
Loose Smut (d)	0.9912	0.8433	0.9113	134	314
Powdery Mildew (e)	0.8769	0.7215	0.7917	158	410
Septoria Leaf Blotch (f)	0.8491	0.6678	0.7476	295	652
Stem Rust (g)	0.8188	0.8760	0.8464	258	646
Yellow Stripe Rust (h)	0.8464	0.8072	0.8263	389	976
Micro avg	0.8040	0.8333	0.8184	2279	5252
Macro avg	0.8342	0.8236	0.8226	2279	5252
Weighted avg	0.8188	0.8333	0.8196	2279	5252
Samples avg	0.8232	0.8366	0.8262	2279	5252
CA = 0.8696					

Table 8: Observations and interpretations for Model_2

Observations	Interpretations
Macro, Micro, Weighted avg for recall and precision almost equal to that of F1 scores	- This indicates a harmonic balance between recall and precision, i.e. there is a balance between classes and the model shows less bias towards a particular class.
CA is a little higher than Micro avg, Macro avg, Weighted avg for accuracy.	- Indication that the model is slightly more exact than precise
Class (d), (e), (f) has much more precision than recall.	- The model, for class (a), is good at selecting positive instances from its positive predictions, but fails to do the same from all instances that are currently positive.
class (a), (c), (g) have significantly higher recall than precision	- For these classes, the model is very good at predicting the greatest number of positive instances from instances that are currently positive, but it is bad at predicting positive instances from its predictions.
AUC-ROC for class (c) significantly lower than all classes and AUC-ROC averages	- Although the probability that the model is able to rank a randomly chosen positive instance higher than a randomly chosen negative instance is high, its performance for class (c) is low.
The ROC curves for classes (d), (f), are significantly lower than the ROC curves for Micro avg and Macro avg.	- These classes will have significantly lower predictive accuracies at different **FPR** or fasle alarm thresholds.

4. Conclusion

In this work, we have proposed two models two based on the CNN and EfficientNetB7, respectively. These models was was able to recognize the wheat disease automatically and effectively. We used the transfer learning and fine-tuning approach for both models.

Our models classify the wheat disease images into eight classes (seven for wheat diseases and one for healthy wheat). We detailed the dataset images used in this study and how we processed and prepared the images for training. Our models can correctly detect diseases in wheat, accurately and over time.

References

[1] M. Prank, C. Kenaley, G. Bergstrom, M. Acevedo, N. Mahowald, Climate change impacts the spread potential of wheat stem rust, a significant crop disease, Environmental Research Letters 14(12) (2019) 124–053.

[2] S.-P. Mohanty, D.-P. Hughes, M. Salathe, Using deep learning for imagebased plant disease detection, Frontiers in plant science 7 (2016) 14–19.

[3] J. Amara, B. Bouaziz, A. Algergawy, A deep learning-based approach for banana leaf diseases classification, in: Datenbanksysteme f¨ur Business, Technologie und Web (BTW 2017)- Workshopband, 2017, pp. 14–19.

[4] K.-P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Computers and Electronics in Agriculture 145 (2018) 311–318.

[5] Y. Lu, S. Yi, N. Zeng, Y. Liu, Y. Zhang, Identification of rice diseases using deep convolutional neural networks, Neurocomputing 267 (2017) 311–318.

[6] A. Sagar, J. Dheeba, On Using Transfer Learning For Plant Disease Detection, bioRxiv 6 (2020).

[7] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovi´c, Deep neural networks based recognition of plant diseases by leaf image classification, Computational Intelligence and Neuroscience 2016 (2016).

[8] J. Liu, M. Wang, L. L. Bao, X.Li, Efficientnet based recognition of maize diseases by leaf image classification, in: Journal of Physics: Conference Series, 2020, p. 1693(1).

[9] M. Rahnemoonfar, C. Sheppard, Deep count: fruit counting based on deep simulated learning, Sensors 17(4) (2017) 905.

[10] I. Sa, Z. Ge, F. Dayoub, B. U. T. Perez, C.McCool, Deepfruits: A fruit detection system using deep neural networks, Sensors 16(8) (2016) 12–22.

[11] R. Sorensen, J.Rasmussen, J. Nielsen, R. Jørgensen, Thistle detection using convolutional neural networks, in: EFITA WCCA 2017 Conference, Montpellier Supagro, Montpellier, France, 2017.

[12] M. Dyrmann, K.-M. Anders, S.-M. H. Skov, N.-J. Rasmus, Pixel-wise classification of weeds and crops in images by using a fully convolutional neural network, in: Proceedings of the International Conference on Agricultural Engineering, Aarhus, Denmark, 2016.

[13] T. Mingxing, Q.-V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, CoRR 1905 (2019).

[14] Q. Xie, M.-T. Luong, E. Hovy, Q.-V. Le, Self-training with Noisy Student improves ImageNet classification, CoRR 1911 (2019).

[15] H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian, Deep Residual Learning for Image Recognition, CoRR 1512 (2015).

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, CoRR 1512 (2015).

[17] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large- Scale Image Recognition, arXiv (2015).

[18] A.-G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, A. Hartwing, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, CoRR 1704 (2017).