

Gabor Capsule Network for Plant Disease Detection

Patrick Mensah Kwabena¹, Benjamin Asubam Weyori²

Department of Computer Science and Informatics
University of Energy and Natural Resources
Sunyani, Ghana

Ayidzoe Abra Mighty³

School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu, China

Abstract—Crop diseases contribute significantly to food insecurity, malnutrition, and poverty in Africa where the majority of the population is into Agriculture. Manual plant disease recognition methods are widespread but limited, ineffective, costly, and time-consuming making the need to search for automatic and efficient methods of recognition more crucial. Machine learning and Convolutional Neural Networks have been applied in other jurisdictions in an attempt to solve these problems. They have achieved impressive results in this domain but tend to be ‘data-hungry’, invariant, and vulnerable to attacks that can easily lead to misclassifications. Capsule Networks, on the other hand, avoids the weaknesses of CNNs and has not been widely used in this area. This article, therefore, proposes the use of Gabor and Capsule network to recognize blurred, deformed, and unseen tomato and citrus disease images. Experimental results show that the proposed model can achieve a 98.13% test accuracy which is comparable to the performance of state-of-the-art CNN models in the literature. Also, the proposed model outperformed two state-of-the-art deep learning models (which were implemented as baseline models) in terms of robustness, flexibility, fast converges, and having fewer parameters. This work can be extended to other crops and may well serve as a useful tool for the recognition of unseen plant diseases under bad weather and bad illumination conditions.

Keywords—Convolutional neural networks; capsule network; gabor filters; crop diseases; machine learning

I. INTRODUCTION

Tomatoes and Citrus are major economic crops that are widely cultivated in developing countries where the majority of the farmers have little or no knowledge about the diseases that can affect these crops and how they can be controlled. These crops also form part of the daily nutritional requirements of many people necessary to maintain good health. However, both crops are plagued with several types of diseases that require timely and accurate identification to prevent crop losses. Current identification methods are manual which is laborious, time-consuming, and error-prone especially during the early stages. Newer automatic recognition methods [1] are therefore needed. Convolutional Neural Networks (CNNs) including state-of-the-art deep transfer learning models such as AlexNet [2], ResNet [3], VGG [4] and GoogleNet [5] have been used to identify crop diseases [6][7][8][9]. However, the problem is that they use max-pooling and are deeper (the deeper the CNN, the better the performance [10]). Max-pooling makes the network invariant requiring a lot of data to avoid overfitting. Depth, on the other hand, comes with some drawbacks such as a large

number of parameters, high complexity, high memory requirements, and high computational demands.

Hasan et al. [11] collected a tomato disease dataset and used the pre-trained weights of GoogleNet and InceptionV3 for classification. A 90% and 10% division for training and test respectively resulted in 99% overall classification accuracy. A further division of 80%, 20% for training and testing respectively resulted in 92% accuracy. Fuentes et al. [12] combined Faster Region-based CNNs and Single Short Multibox Detector (SSD) algorithms with deep feature extractor pre-trained models such as VGG and ResNet for tomato disease/pest recognition to obtain 85.98% accuracy. Zhange et al. [13] trained AlexNet, ResNet, and GoogLeNet on the Plant Village (PV) [14] dataset obtaining 97.28% accuracy. This same dataset was used by Iandola et al. [15], Durmus et al. [16], and Krishnaswamy et al. [17], to evaluate the performance of AlexNet and SqueezeNet, AlexNet (95.65%), AlexNet and VGG16 (99.24% for 6 classes) respectively. In [18], a modified LeNet [19] was used to obtain 94.85% accuracy on the PV dataset. Nine out of ten classes of the PV dataset were used by Brahimi et al. [20] to fine-tune GoogleNet and AlexNet resulting in 99.18% accuracy. In [21], VGGnet was trained and evaluated on the PV dataset achieving a classification accuracy of 95.24% while AlexNet and GoogleNet obtained 84.58% test accuracy on the same dataset.

Wang et al. [22] collected sick tomato leaf images from the internet and trained a region-based CNN (R-CNN) to detect disease types and areas of infection. Their networks were so deep that ResNet-101 obtained 23.25 hours of training time.

Other plant disease detection models in the literature [23][1][24][25][26] achieved good results, however, most of them are deep, complex, invariant, not robust, low performing, and lack flexibility. Additionally, they are invariant, cannot encode hue, texture, spatial orientation, and deformation. These weaknesses led to the introduction of Capsule Networks (CapsNets) [27] which are capable of encoding spatial information, texture, hue, and deformation. Capsules perform well on smaller datasets and are well suited for crop disease recognition since texture and orientation play key roles in the recognition of leaf parts that do not conform to the other parts of the leaf. However, capsules have a problem in recognizing real images with complex backgrounds [28].

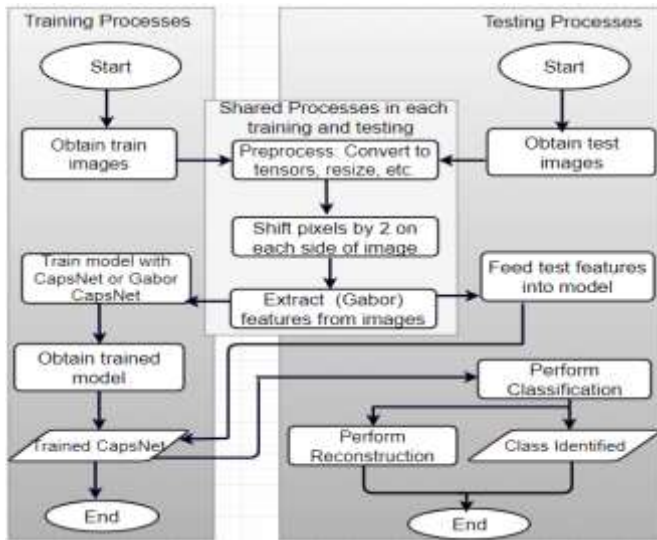


Fig. 1. Workflow Diagram Adopted in this Study.

This paper adopts Capsule’s dynamic routing algorithm by adding a Gabor layer [29] to further enhance its textural and spatial recognition capabilities. The workflow adopted for the proposed work is shown in Fig. 1.

Experimental results on two datasets show that the Gabor CapsNet outperformed both the state-of-the-art CNN baseline models and a CapsNet model on deformed images and unseen images. The proposed model also proved to be more flexible and converges faster than the baseline models. A model’s flexibility and ability to generalize on unseen/deformed data is crucial for the control of plant diseases such as the *early blight* tomato disease which is spread by wind and splashing rain.

The main contributions of this paper are: 1) reusing existing methods to improve the robustness and flexibility of CapsNets on deformed, blurred, and spatially rotated images. The results demonstrate the feasibility of using Gabor Capsules for plant disease recognition under subnormal conditions, 2) the proposed model outperforms existing state-of-the-art CNN models in terms of accuracy and also has fewer parameters compared to deep CNN models in the literature except for GoogleNet, 3) the Gabor-CapsNet architecture has superior texture extraction capabilities capable of identifying sick parts.

This paper is divided into the following sections: Section II presents an introduction to Gabor CapsNets followed by Section III which outlines the Materials and Methods used for this work leading to Sections IV where the experimental setup, the proposed model, and baseline models are presented. Results are presented and discussed in Section V followed by Section VI where the work is concluded and future works provided.

II. GABOR CAPSULE NETWORK

A Capsule [27] is a group of neurons whose activity vector represents the instantiation parameters with the length of the vector representing the likelihood that an entity exists. The first layer of a Capsule network is a CNN layer followed by a Primary Capsule (PC) layer. The Class Capsule (CC) layer

performs the classification while the decoder network performs reconstruction. The CNN layer performs feature extraction to serve as input to the PC layer which in turn produces $\hat{u}_{j|i} = W_{ij}u_i$ as output. A coupling coefficient $c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$ allows a lower-level capsule to choose a higher-level capsule as a cluster centre. The coupling coefficient is the SoftMax of the logits $b_{ij} = b_{ij} + v_j \cdot \hat{u}_{j|i}$. During the routing process, the c_{ij} s are updated based on the agreement $a_{ij} = v_j \cdot \hat{u}_{j|i}$ between the prediction of a lower level capsule and a higher-level capsule. The total input to a higher-level capsule j takes as input the weighted sum of all prediction vectors $\hat{u}_{j|i}$ of a given PC i for a given CC j . This is given by $s_j = \sum_{i=1}^{N_1} c_{ij} \hat{u}_{j|i}$. To constrain the value of the CC’s output between the range [0,1], the squashing function $v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}$ is applied. CapsNets have performed well on a wide range of problems [30].

Gabor Filters [29] on the other hand are linear filters popularly used for texture [31] analysis, edge detection, and feature extraction. They can be used to approximate the characteristics of the visual cortex of some animals. A Gabor filter is composed of real and imaginary parts. The real part is described by equation (1), where λ = sinusoidal factor wavelength, θ = orientation of the normal to the Gabor function parallel stripes, σ = standard deviation of the Gaussian envelope, and γ = spatial aspect ratio specifying the specificity.

$$g\lambda, \theta, \sigma, \gamma(x, y) = \exp\left(-\frac{x'^2 + \gamma y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \phi\right) \quad (1)$$

where $x' = x \cos \theta + y \sin \theta$, $y' = -x \sin \theta + y \cos \theta$.

Practically, λ regulates the width of the Gabor function strips; increasing λ will increase the width and vice versa. θ , on the other hand, governs the orientation of the strips. A $0^\circ \theta$ represents a vertical strip. γ and σ respectfully control the height and overall size of the strips.

Gabor filters recognize orientation and texture. During convolution, global Gabor Filter banks are used to extract the features. Given an input image $I(x, y)$, convolution (*) of the image with a global Gabor filter bank $G(x, y; w, \theta)$, produces $(O_{m,n}(x, y))$ features that can be approximated by equation (2).

$$O_{m,n}(x, y) = I(x, y) * G(x, y; w, \theta) \quad (2)$$

Gabor Capsules [32] (applied to Expectation maximization Capsules) and Gabor CNNs [33][31][34] have performed well on images through texture recognition.

III. MATERIALS AND METHODS

A. Image Acquisition and Preprocessing

Tomato dataset: It is a subset of the Plant Village dataset and consists of 18,159 images; nine categories of infected leaves and one healthy leaves class. Data imbalance, the similarity of images from different classes, and varied image backgrounds make the dataset challenging for classification models.

Citrus dataset [35]: This dataset is made up of sick and healthy leaves and fruits with the following categories: Blackspot, Canker, Scab, Greening, and Melanose. The dataset is made up of 759, 256 x 256x3 images acquired from crop-fields making it complex as well as having the data imbalance problem that plagues most data sets. It was used for the classification of citrus diseases in [36]. Fig. 2 depicts sample raw images from the two datasets.



Fig. 2. Sample Images from the Two Datasets. Top Row: Tomato Dataset. Bottom Row: Citrus Dataset.

The images were resized from the original 256x256 to 48x48, 68x68, and 224x224 depending on which model was being trained. Standard data augmentation techniques such as vertical and horizontal mirroring, blurring, deformation, and rotation were applied to 50% of the images in each class of the test sets. The deformation was achieved using Moving Least Squares’ affine transformation [37] and the blurring by the use of Gaussian function in what is known as Gaussian Blur [38] with a kernel size of 15x15. These steps were necessary to test the ability of the models to generalize on unseen data and also under bad illumination conditions. Some of the preprocessed images are shown in Fig. 3.

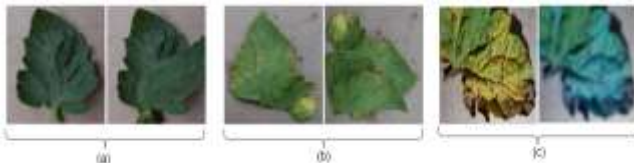


Fig. 3. Sample Preprocessed Images: (a) *Bacterial_Spot* Infected Original Image on the Left and its Deformed Counterpart on the Right, (b) *Bacterial_Spot* Infected Original Image on the Left and its Deformed-Rotated Counterpart on the Right, (c) *Early_Blight* Infected Original Image on the Left and its Gaussian Blurred Counterpart on the Right.

IV. EXPERIMENTS

This work was carried out in Python 3.7. PyTorch 1.3 was used to design all the models with visualizations produced in Visdom server. The computing hardware was a 64bit Windows machine with NVIDIA GeForce GTX 1060 Graphic Processing Unit (GPU) running on CUDA 10.1 with a dedicated memory of 8GB. The CPU is an Intel Core i7, 8th generation.

In this work, a Gabor Capsule network is proposed and trained from scratch. Three baseline models were used to evaluate the performance of the proposed model. The baseline models are 1) Capsule network based on dynamic routing, 2) AlexNet, and 3) GoogleNet. The last two were fine-tuned

based on the implementation in [20]. The models were each trained for 400 epochs with a batch size of 60. Other hyperparameters for implementing the CapsNets include three routing iterations, rectified linear unit (ReLU) for non-linearity, use of the sigmoid function in the last FC layer, SoftMax function, and the Adam optimizer.

A. Proposed Gabor Capsule Model

This paper uses the properties of Gabor filters in Capsule networks to develop a plant disease detection model. Fig. 4 shows the proposed architecture which is made up of one Gabor layer, one CNN layer, a PC layer, and the class capsule (*DiseaseRecognition*) layer. The Gabor layer is implemented as a convolutional layer with its filters constrained to fit a Gabor function [33]. The Gabor layer uses 96, 7x7 kernels to produce 96, 42x42 feature maps for the subsequent convolutional layer at a stride of 1. The first convolutional layer (*Conv1*) uses ReLU non-linear activation and has 96, 9x9 kernels producing 96, 34x34 feature maps. *Conv1* runs at a stride of 1.

The primary capsule layer is a convolutional capsule layer with 12 channels of convolutional 8D capsules. Each component capsule in the primary capsule layer has 13x13 capsules. The PC layer outputs 13*13*12, 8D capsules. The decoder network is a fully connected layer with 512 neurons followed by 1024 and 6912 neurons for the first, second, and third FC layers respectively. It is the responsibility of the decoder network to perform reconstruction of the original images. The frequency f_n and orientation θ_m of the Gabor filters in the Gabor layer is set using the expressions in equations 3 and 4 [33].

$$f_n = \frac{\pi}{2} \sqrt{2}^{-(n-1)} \tag{3}$$

$$\theta_m = \frac{\pi}{8} (m - 1) \tag{4}$$

where $n=1,2, \dots, 5$ and $m=1, 2, \dots, 8$.

B. Baseline Models

In this section, the three baseline models are discussed in detail.

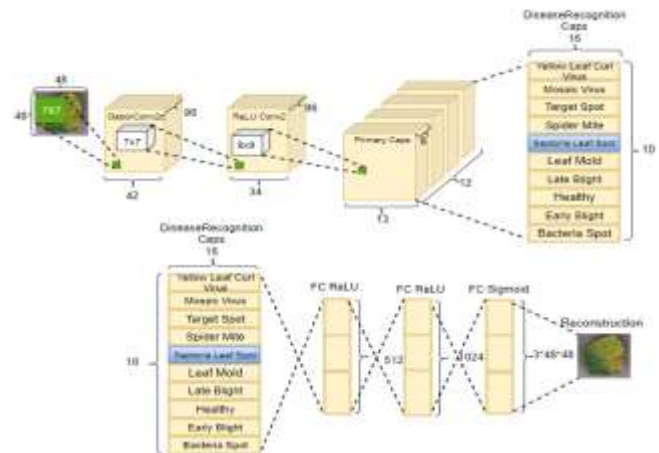


Fig. 4. Gabor CapsNet Architecture for Plant Disease Recognition.

C. CNN Baseline Models

The two CNN models in [20] were implemented in this paper as baseline models to provide a common implementation platform for a fair comparison of results between the proposed and baseline models.

1) *GoogLeNet*: In 2014, GoogLeNet [5] achieved an impressive top-5 error rate of 6.67% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The network was built on LeNet [19] with an Inception module and global average pooling. Its architecture is a 22-layer deep CNN. 1x1 convolutions were used for reduction in dimensionality and computations. Its pre-trained weights have since been reused and transferred to solve other image recognition tasks. In this work, GoogLeNet was implemented as one of the baseline models occupying 25.91MB disk space with approximately 6.55 Million parameters. The following changes were made to the pre-trained model to solve the problem at hand: 1) the output layer was changed from 1000 to 10 since there are ten classes in the tomato dataset, 2) the top three layers were fine-tuned since the initial layers usually extract generic features (e.g. edge detectors or color blob detectors) while the upper layers are dataset-specific.

2) *AlexNet*: AlexNet [2] won the 2012 ILSVR Challenge with a top-5 error of 26% to 15.3%. The network comprises stacked convolutional layers of 11x11, 5x5, 3x3 convolutions. It utilized dropout, stochastic gradient descent (SGD), does max pooling, and uses a rectified linear unit (ReLU) for non-linearity. The architecture of AlexNet is such that it occupied 242.03 MB disk space with approximately 61 million parameters. The pre-trained model was loaded and the last three layers fine-tuned to adapt to the new classification problems with 10 and 6 classes respectively for the tomato and citrus datasets as well as reducing the image sizes from 227x227, 3 to 224x224, 3 channels.

D. Capsule Baseline Model

Fig. 5 is the architecture of the baseline Capsule network model. The input images are resized from (256x256x3) to (48x48x3). They are then fed into the first convolutional layer (Conv1) with 7x7, 96 kernels with ReLU non-linear activation. The convolutions in the Conv1 layer are performed with a stride of 1. Conv1 then produces 256, 42x42 feature maps which are fed into the second convolutional layer (Conv2) also with ReLU non-linear activation. Conv2 is made up of 96, 9x9 kernels performing convolution over the image at a stride of 1. Conv2 then produces 96, 34x34 feature maps as input to the primary capsule (PC) layer. The primary capsule layer is a convolutional capsule layer with a kernel size of 9x9 and a stride of 2. In the PC layer, the output of the standard convolution layer comes in the form of 96 channels of scalars in 13x13 arrays. These are seen as 12 channels of 8-dimensional vectors organized in 13x13 arrays. The resulting value for the PC output is a 13*13*12, 8-dimensional vectors (also called the routing nodes) which are changed into a 16-dimensional vector in the DiseaseRecognitionCaps layer. These dimensions may hold features such as size, texture, deformation, orientation, hue, and position. A tensor product

between u and the weights (W) produces \hat{u}_{ij} which is made up of 2028, 16-dimensional vectors for each DiseaseRecognitionCaps output. Since there are 10 classes in all, the total number of outputs for the DiseaseRecognitionCaps is 2028*10, 16D vectors. These are fed into a Fully connected (FC) layer consisting of three layers. This part is usually referred to as the decoder and is made up of 512 neurons in the first FC layer followed by 1024 neurons. The last layer of the decoder network is made up of 6912 neurons necessary for reconstructing the input image.

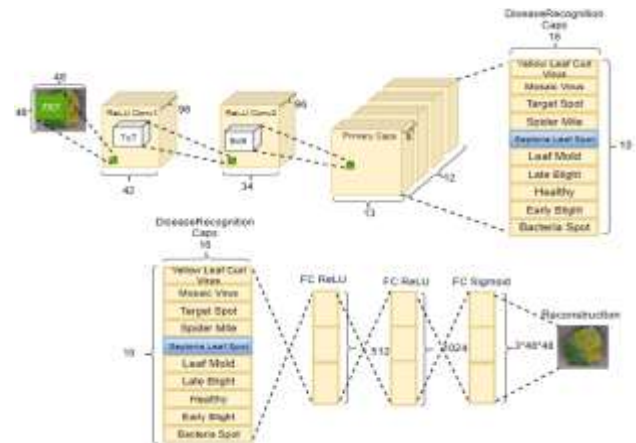


Fig. 5. The Baseline CapsNet Architecture for Plant Disease Recognition.

V. RESULTS AND DISCUSSION

The datasets were divided with a ratio of 8:2 for training and testing respectively for all the models. The loss function used to train the model is made up of the margin and reconstruction losses as depicted in Fig. 6(a) and (c) and Appendix A (Fig. 11). The default values for $m+$, $m-$, and λ of the loss function in [27] were maintained in this implementation. Three routing iterations were used during training for the Capsule models. The proposed model obtained 98.13% and 93.33% accuracies for the tomato and citrus datasets respectively. The proposed model outperformed all the other models on both datasets. Fig. 7 and 8 depict the confusion matrices obtained by training the proposed model with the datasets. It can be seen from Table I that GoogLeNet achieved 97.60% accuracy outperforming the CapsNet (95.29%) and AlexNet (94.40%) baseline models on the tomato dataset.

A. Model Flexibility and Robustness

Random changes to parameter values and/or intermediate layers in all the models were carried out to determine how sensitive each model is to these changes. The effect of these changes adversely affected the performance of the CNN models as compared to the proposed Capsule models.

Varying the momentum, batch size, learning rate, dropout, and learning rate decay did not significantly affect the performance of the CapsNets models as observed in [30]. The single most important hyperparameter that significantly affected the performance of the CapsNet models was the number of routing iterations with three producing the best performance values. To illustrate the flexibility and robustness of the CapsNets, the input images were resized from 256x256

to 48x48, 68x68, and 224x224, and the models trained. The results in Table I show that the performance of the baseline CNN models was affected by simple resizing. The default settings in AlexNet could not train with the 48x48 images.

On the other hand, as the image size was increased, the CapsNet models produced almost consistent results. However, increasing the image size to 224x224 required more computational resources and training time and could not be implemented for this study.

It is noted here that Pytorch can accumulate gradients over multiple smaller batches as long as enough memory exists for a batch, however, 400 epochs were excessive, slow to train, and was taking too long a time.

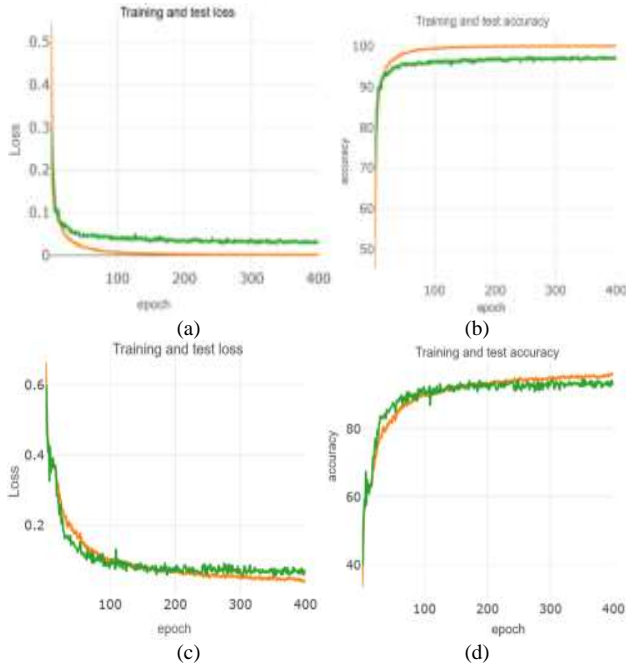


Fig. 6. Loss and Accuracy of the Proposed Gabor CapsNet Model Trained on (a) and (b) Tomato Dataset, (c), and (d) Citrus Dataset. The Training Curves are shown in Orange with the Test Result Curves shown in Green.

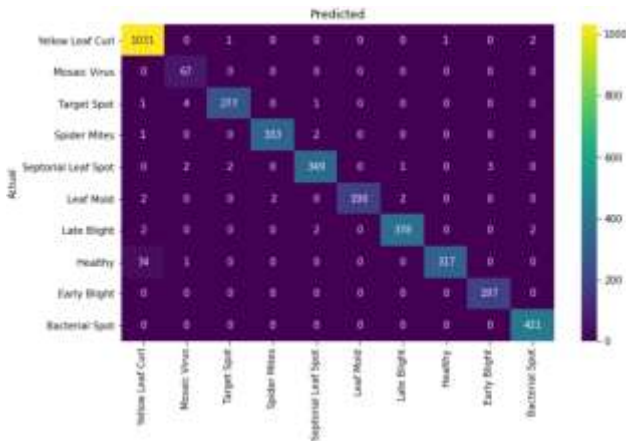


Fig. 7. Confusion Matrix for Tomato Dataset.

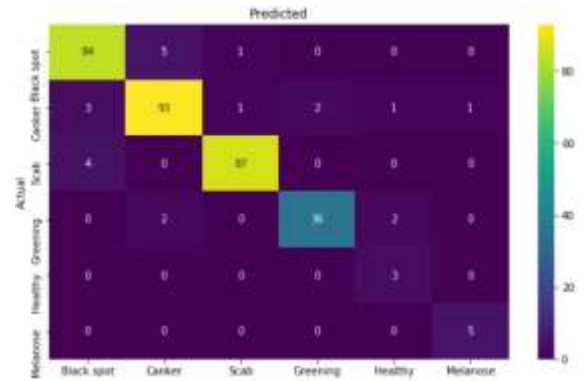


Fig. 8. Confusion Matrix for Citrus Dataset.

B. Model Convergence

The resulting plots in Fig. 9 and 10 show that the proposed Gabor CapsNet learns and converges faster than the other models. For instance, between epochs 0 to 100, the Gabor CapsNet attains accuracies higher than all the other models. The final accuracies are approximately equal to the accuracies they assume in the initial stages. As a result, the final accuracy of the proposed Capsule network can be approximated during the first few epochs.

On the contrary, the baseline models rise gradually through each epoch up to the last epoch. The final accuracy of the baseline models can therefore not be approximated at the initial stages. One has to wait for the entire duration of the training before a determination of the final accuracy can be made. This convergence is attributed to the ability of the Gabor filters to encode the texture of the diseased parts of leaves. The ensuing capsule layer after the Gabor layer also can encode texture, pose, and deformation. Fast learning and convergence are the results of the working together of these layers. These are particularly useful during a preliminary investigation into crop diseases and for prototyping.

TABLE I. ACCURACY DECREASES AS IMAGE SIZE IS REDUCED FOR THE BASELINE CNN MODELS. CAPSNET MODELS DID NOT SHOW ANY SIGNIFICANT CHANGE IN ACCURACY

Model (dataset)	Accuracy		
	48x48 images	68x68 images	224x224 images
Gabor CapsNet (tomato)	98.12%	98.93%	-----
CapsNet (tomato)	95.29%	95.11%	-----
AlexNet (tomato)	-----	81.85%	94.40%
GoogleNet (tomato)	90.20%	95.98%	97.60%
Gabor CapsNet (Citrus)	93.33%	92.85%	-----
CapsNet (Citrus)	90.69%	90.26%	-----
AlexNet (Citrus)	-----	81.85%	90.90%
GoogleNet (Citrus)	88.81%	90.17%	91.04%

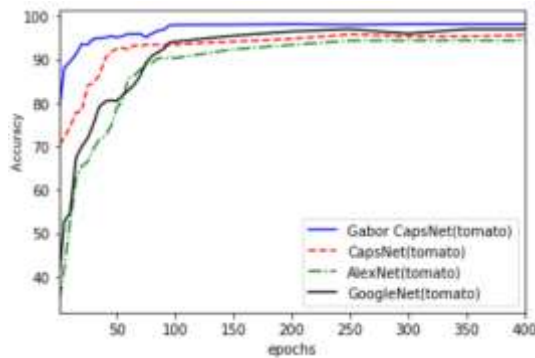


Fig. 9. Model Convergence on Tomato Dataset.

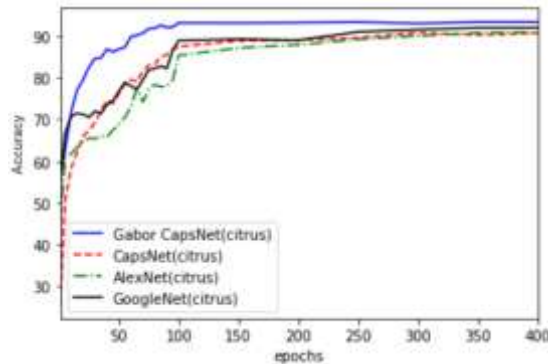


Fig. 10. Model Convergence on the Citrus Dataset.

C. Reduced Parameters

The models were evaluated on the number of trainable parameters using the 68x68 images and the results shown in Table II. The complexity of a model can be inferred from the number of trainable parameters it generates. As shown in Table II, the proposed model had fewer parameters than the AlexNet baseline model. This is a contribution to the state-of-the-art since fewer parameters are needed to reduce model complexity and its ability to over-fit smaller datasets.

D. Comparison to Related Works in Literature

The tomato dataset has been used in the literature to fit several CNN and deep learning models. In Table III, a comparison between these models and the proposed model based on average test accuracies is provided for the tomato dataset. For a fair comparison, other implementations in the literature using custom tomato datasets [12][11][22] were not adopted for this exercise. It can be seen from Table III that the proposed model produced results that are comparable to the state-of-the-art models irrespective of the complexity of the input images.

TABLE II. COMPARISON OF PARAMETERS BETWEEN THE PROPOSED MODEL AND THE BASELINE MODELS

Model	Number of Parameters
Gabor CapsNet	12,397,216 ~ 12 Million
CapsNet	9,583,936 ~ 10 Million
Baseline AlexNet	61 Million
Baseline GoogleNet	6.55 Million

TABLE III. COMPARISON OF RESULTS OF THE PROPOSED MODEL TO THOSE IN THE LITERATURE. NOTICE THAT SOME OF THE MODELS USED 6 OR 9 CLASSES OUT OF THE TOTAL 10 CLASSES

Reference	Model Algorithm	Number of Classes	Accuracy
[16]	AlexNet (pre-trained weights)	10	95.65%
[16]	SqueezeNet (pre-trained weights)	10	94.30%
[18]	Variation of LeNet [19]	10	94.85%
[21]	VGG	10	95.24%
[20]	AlexNet	9	98.66%
[20]	GoogleNet	9	99.18%
[13]	ResNet [3]	10	97.28%
[17]	VGG16	6	99.24%
Proposed	Gabor CapsNet	10	98.12%
Baseline	CapsNet	10	95.29%

VI. CONCLUSION

In this work, Gabor Capsule Network for the recognition of tomato and citrus diseases has been proposed. Two state-of-the-art CNN and one capsule baseline models were also implemented for comparison. To determine the robustness of the proposed models, extensive preprocessing such as rotation, deformation, and Gaussian blur was applied to a proportion of the test set and used to test each of the models. The Gabor CapsNet outperformed the other models on the two datasets in terms of accuracy, convergence, robustness, complexity, and flexibility. The results suggest that Capsule Networks can outperform other deep learning methods on complex real-world datasets. Furthermore, they can recognize unhealthy plants even in challenging weather and illumination conditions as well as from diverse angles. The results in this paper show that Capsules have a huge potential to improve agriculture especially as the algorithm is being improved by researchers to enable it mature for practical adoption.

In the future, a further reduction in the number of parameters for possible implementation on mobile devices like smartphones will be pursued since a high percentage of farmers have mobile phones. The possibility of using a custom routing algorithm will also be considered.

REFERENCES

- [1] M. Dong, S. Mu, T. Su, and W. Sun, "Image Recognition of Peanut Leaf Diseases Based on Capsule Networks," in ICAI 2019, 2019, pp. 43–52, doi: 10.1007/978-981-32-9298-7_4.
- [2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, vol. 2016-December, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in ICLR 2015 Conference proceedings, arXiv:1409.1556v6 [cs.CV], 2015, pp. 1–14.
- [5] C. Szegedy et al., "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '15), 2015, pp. 95–111, doi: 10.1109/CVPR.2015.7298594.

- [6] A. F. Fuentes, S. Yoon, J. Lee, and D. S. Park, "High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank," *Front. Plant Sci.*, vol. 9, no. August, pp. 1–15, 2018, doi: 10.3389/fpls.2018.01162.
- [7] J. G. M. Esgario, R. A. Krohling, and J. A. Ventura, "Deep learning for classification and severity estimation of coffee leaf biotic stress," *Comput. Electron. Agric.*, vol. 169, no. July 2019, 2020, doi: 10.1016/j.compag.2019.105162.
- [8] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cogn. Syst. Res.*, pp. 1–11, 2018, doi: 10.1016/j.cogsys.2018.04.006.
- [9] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, "Deep interpretable architecture for plant diseases classification," *Signal Process. - Algorithms, Archit. Arrange. Appl. Conf. Proceedings, SPA*, vol. 2019-Sept, pp. 111–116, 2019, doi: 10.23919/SPA.2019.8936759.
- [10] R. Cadène, "Deep Learning for Visual Recognition," 2016.
- [11] M. Hasan, B. Tanawala, and K. J. Patel, "Deep Learning Precision Farming: Tomato Leaf Disease Detection by Transfer Learning," in *2nd International Conference on Advanced Computing and Software Engineering (ICACSE-2019)*, 2019, pp. 1–5, doi: 10.2139/ssrn.3349597.
- [12] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors (Switzerland)*, vol. 17, no. 9, 2017, doi: 10.3390/s17092022.
- [13] K. Zhang, Q. Wu, A. Liu, and X. Meng, "Can deep learning identify tomato leaf disease?," *Adv. Multimed.*, vol. 2018, 2018, doi: 10.1155/2018/6710865.
- [14] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," <https://arxiv.org/abs/1511.08060>, 2015.
- [15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-Level Accuracy with 50 X Fewer Parameters and < 0.5mb Model Size," *arXiv1602.07360v4 [cs.CV]*, pp. 1–13, 2017.
- [16] H. Durmus, E. O. Gunes, and M. Kirci, "Disease detection on the leaves of the tomato plants by using deep learning," in *2017 6th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2017*, 2017, doi: 10.1109/Agro-Geoinformatics.2017.8047016.
- [17] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Comput. Sci.*, vol. 133, pp. 1040–1047, 2018, doi: 10.1016/j.procs.2018.07.070.
- [18] P. T. A. Pranathi, K. S. Ashritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks," in *2018 11th International Conference on Contemporary Computing, IC3 2018*, 2018, pp. 2–4, doi: 10.1109/IC3.2018.8530532.
- [19] Y. LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," *Sci. Signal.*, vol. 7, no. 329, pp. 541–551, 2014, doi: 10.1126/scisignal.2005580.
- [20] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 299–315, 2017, doi: 10.1080/08839514.2017.1315516.
- [21] E. Suryawati, R. Sustika, R. S. Yuwana, A. Subekti, and H. F. Pardede, "Deep structured convolutional neural network for tomato diseases detection," in *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*, 2019, no. October 2019, pp. 385–390, doi: 10.1109/ICACSIS.2018.8618169.
- [22] Q. Wang, F. Qi, M. Sun, J. Qu, and J. Xue, "Identification of Tomato Disease Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Techniques," *Comput. Intell. Neurosci.*, vol. 2019, 2019, doi: 10.1155/2019/9142753.
- [23] R. V. Kurup, M. A. Anupama, R. Vinayakumar, V. Sowmya, and S. K. P., "Capsule Network for Plant Disease and Plant Species Classification," in *ICCVBIC -Advances in Intelligent Systems and Computing*, 2019, vol. 186 AISC, pp. 413–421, doi: 10.1007/978-3-030-37218-7.
- [24] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Comput. Intell. Neurosci.*, vol. 2016, 2016, doi: 10.1155/2016/3289801.
- [25] Y. Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease," *Plant Phenomics*, vol. 2019, pp. 1–14, 2019, doi: 10.34133/2019/9237136.
- [26] S. Verma, A. Chug, and A. P. Singh, "Exploring capsule networks for disease classification in plants," *J. Stat. Manag. Syst.*, vol. 23, no. 2, pp. 307–315, 2020, doi: 10.1080/09720510.2020.1724628.
- [27] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, vol. 2017-Decem, no. NIPS 2017, pp. 3857–3867.
- [28] R. Mukhometzianov and J. Carrillo, "CapsNet comparative performance evaluation for image classification," <https://arxiv.org/abs/1805.11195>, pp. 1–14, 2018.
- [29] D. Gabor, "Theory of communication. Part 1: The analysis of information," *J. Inst. Electr. Eng. - Part III Radio Commun. Eng.*, vol. 93, no. 26, pp. 429–441, 1946, doi: 10.1049/ji-3-2.1946.0074.
- [30] M. Kwabena Patrick, A. Felix Adekoya, A. Abra Mighty, and B. Y. Edward, "Capsule Networks – A survey," *J. King Saud Univ. - Comput. Inf. Sci.*, 2019, doi: 10.1016/j.jksuci.2019.09.014.
- [31] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biol. Cybern.*, vol. 61, no. 2, pp. 103–113, 1989, doi: 10.1007/BF00204594.
- [32] S. Hosseini and N. I. Cho, "GF-CapsNet: Using Gabor Jet and Capsule Networks for Facial Age, Gender, and Expression Recognition," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, 2019, pp. 1–8, doi: 10.1109/fg.2019.8756552.
- [33] A. Alekseev, "GaborNet: Gabor filters with learnable parameters in deep convolutional neural networks Gabor function," *AXRIV Arch.*, 2019.
- [34] S. Luan, C. Chen, B. Zhang, J. Han, and J. Liu, "Gabor Convolutional Networks," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4357–4366, 2018, doi: 10.1109/TIP.2018.2835143.
- [35] H. T. Rauf, B. A. Saleem, M. I. U. Lali, M. A. Khan, M. Sharif, and S. A. C. Bukhari, "A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning," *Data Br.*, vol. 26, pp. 1–7, 2019, doi: 10.1016/j.dib.2019.104340.
- [36] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Comput. Electron. Agric.*, vol. 150, no. April, pp. 220–234, 2018, doi: 10.1016/j.compag.2018.04.023.
- [37] S. Schaefer, T. Mcphail, and J. Warren, "Image Deformation Using Moving Least Squares," *ACM SIGGRAPH 2006*, vol. 1, no. 212, pp. 533–540, 2006.
- [38] J. Flusser, S. Member, S. Farokhi, C. H. Iv, and M. Pedone, "Recognition of Images Degraded by Gaussian Blur," *IEEE Trans. IMAGE Process.*, vol. 25, no. 2, pp. 790–806, 2016.

APPENDIX A

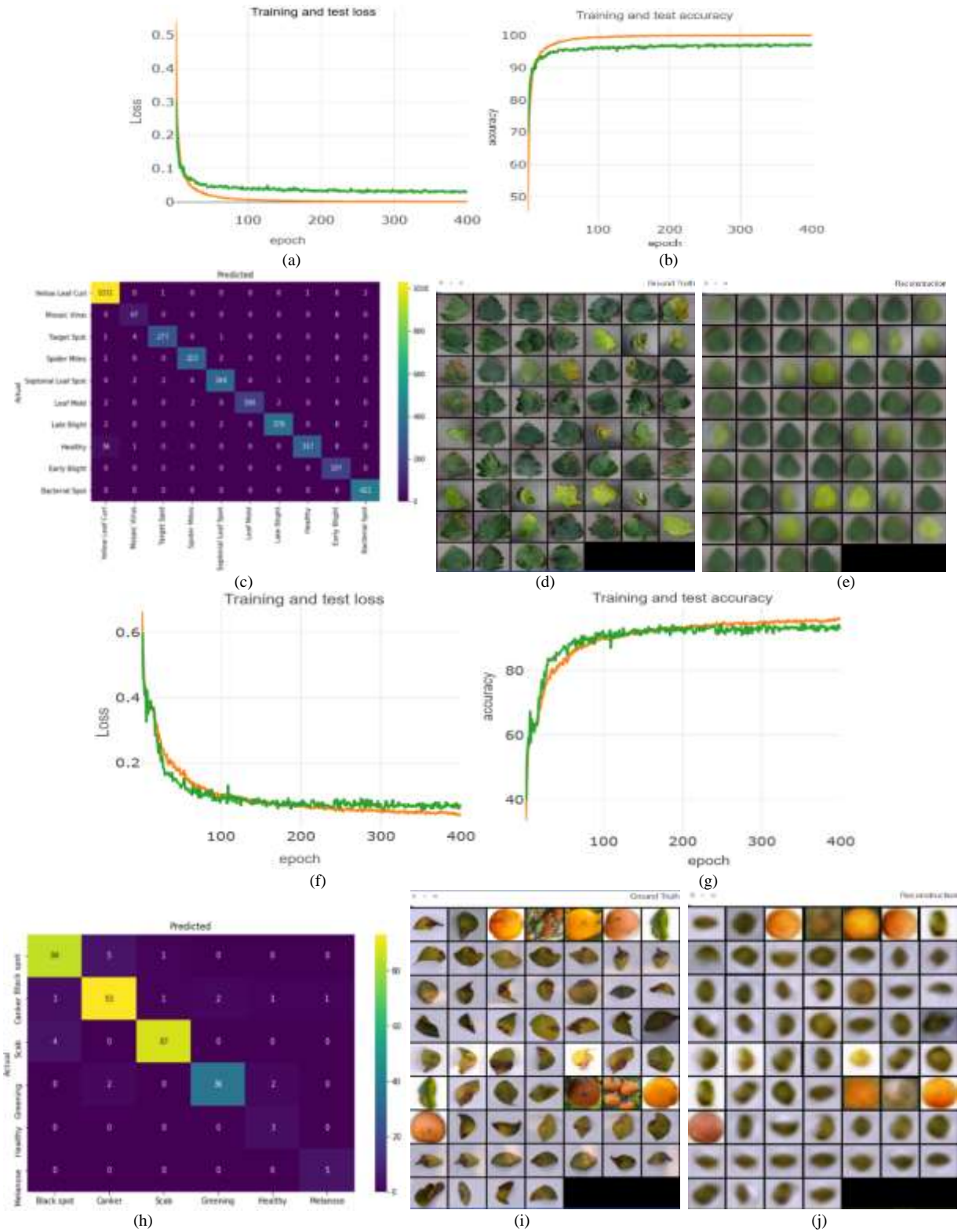


Fig. 11. Sample Results from the Proposed Gabor Capsule Network: (a)-(e) Tomato Dataset, (f)-(j) Citrus Dataset. (a), (b), (f), and (g) are Loss and Accuracy for Training and Test. (c) and (h) are the Confusion Matrices. (d), (e), (i), and (j) are the Ground Truth and Reconstructed Images.