agriculture

MDPI

Article

Compact Convolutional Transformer (CCT)-Based Approach for Whitefly Attack Detection in Cotton Crops

Aqeel Iftikhar Jajja [1], Assad Abbas [1], Hasan Ali Khattak [2,*], Gniewko Niedbała [3,*], Abbas Khalid [4], Hafiz Tayyab Rauf [5] and Sebastian Kujawa [3]

[1] Department of Computer Science, COMSATS University Islamabad, Islamabad 45500, Pakistan
[2] School of Electrical Engineering & Computer Science (SEECS), National University of Sciences & Technology (NUST), H12, Islamabad 44000, Pakistan
[3] Department of Biosystems Engineering, Faculty of Environmental and Mechanical Engineering, Poznań University of Life Sciences, Wojska Polskiego 50, 60-627 Poznań, Poland
[4] Department of Computer Science and IT, The University of Lahore, Lahore 54590, Pakistan
[5] Centre for Smart Systems, AI and Cybersecurity, Staffordshire University, Stoke-on-Trent ST18 0YB, UK
* Correspondence: hasan.alikhattak@seecs.edu.pk (H.A.K.); gniewko.niedbala@up.poznan.pl (G.N.)

Abstract: Cotton is one of the world's most economically significant agricultural products; however, it is susceptible to numerous pest and virus attacks during the growing season. Pests (whitefly) can significantly affect a cotton crop, but timely disease detection can help pest control. Deep learning models are best suited for plant disease classification. However, data scarcity remains a critical bottleneck for rapidly growing computer vision applications. Several deep learning models have demonstrated remarkable results in disease classification. However, these models have been trained on small datasets that are not reliable due to model generalization issues. In this study, we first developed a dataset on whitefly attacked leaves containing 5135 images that are divided into two main classes, namely, (i) healthy and (ii) unhealthy. Subsequently, we proposed a Compact Convolutional Transformer (CCT)-based approach to classify the image dataset. Experimental results demonstrate the proposed CCT-based approach's effectiveness compared to the state-of-the-art approaches. Our proposed model achieved an accuracy of 97.2%, whereas Mobile Net, ResNet152v2, and VGG-16 achieved accuracies of 95%, 92%, and 90%, respectively.

Keywords: computer vision; CCT; cotton pest attack; whitefly attack; deep learning; precision agriculture

check for updates

Citation: Jajja, A.I.; Abbas, A.; Khattak, H.A.; Niedbała, G.; Khalid, A.; Rauf, H.T.; Kujawa, S. Compact Convolutional Transformer (CCT)-Based Approach for Whitefly Attack Detection in Cotton Crops. *Agriculture* **2022**, *12*, 1529. https://doi.org/10.3390/agriculture12101529

Academic Editor: Wei Ji

Received: 11 August 2022
Accepted: 20 September 2022
Published: 23 September 2022

1. Introduction

Agriculture yield in recent years has declined dramatically, posing a threat to global food security. With the world's population predicted to approach 9.7 billion by 2050, there is a strong exigency to boost productivity through use of new technology. Pakistan is ranked sixth in the world in terms of cotton production [1]. Cotton accounts for around 0.6% of Pakistan's GDP. Its output has steadily fallen in recent years, falling by up to 22% [2], and a further fall at this rate will eventually have a negative impact on productivity. Weather, humidity, insect attacks, numerous viruses, and poor pesticides are all elements which impair cotton production.

The whitefly (Bemisia tabaci) is considered as one of the conventional pests that attack the plant and is a carrier of various viruses, for example, cotton leaf curl disease (CLCUD). The whitefly is one of the world's 100 worst invasive alien species [3]. Whiteflies can infect a cotton field and adjacent crops, restricting plant development by up to 50% [4]. As a consequence, this minuscule insect causes harm to both global food and domestic cash crops.

With technological progress, field inspection is uniformly upgrading to more automated sources, such as state-of-the-art automated disease detection algorithms and spatial drones for detection and prediction. Since beforehand there were no suitable mechanisms to identify plant diseases, one had to manually inspect each plant and prescribe appropriate treatment, which was exceedingly laborious, time-consuming, and required more excellent professional knowledge and skills.

Due to the seriousness and implications of whiteflies for plants, more effective, efficient, and sustainable strategies are needed. Researchers have used machine learning, image processing, and computer vision techniques to develop several solutions to these problems [5,6]. The applicability of image processing to precisely identifying and categorizing disease has arisen in numerous domains.

Recently, several works have employed images of multi-class disease-based private datasets from online dataset repositories. Image acquisition is a critical challenge, as it is hard to acquire photographs from a real-world context. However, Legaspi et al. [7] proposed a framework for detecting and classifying whiteflies using the YOLO-V3 pre-trained model. Initially, 400 images were gathered manually from the fields. Tulshan et al. [8] suggested a multi-class disease classification method based on k-nearest neighbors (KNN) and compared its performance with the support vector machine (SVM). The dataset used for experiments was small, exhibiting various diseases, including mosaic virus, leaf miner, whitefly, mildew, and early blight. Using the SVM methodology, Rajan et al. [9] suggested a strategy for early pest identification. The SVM was trained and evaluated using a sample dataset of 100 images, using slack variable and the threshold value.

The scarcity of adequate cotton datasets is a hindrance to the aggrandizing of deep learning approaches in pest and disease detection. To eradicate this problem, we collected images from a real environment. A total of 5135 image samples were utilized for deep learning model training. The AgriPK dataset (AgriPK Dataset https://doi.org/10.34740/KAGGLE/DSV/2927481) is the largest among the available datasets. Moreover, it is the first dataset for whitefly pest-affected plants to best of our knowledge.

Deep learning models are regarded as pivotal models for classification and detection. Attention-based feature extraction layers applied to models enable them to concentrate on the region of interest (ROI) optimally. We used the latest version of the visual transformer, the Compact Convolutional Transformer (CCT), for the medium-sized dataset. Convolutional layers are utilized in the input layer of the proposed model to build feature maps. The experimental results demonstrate the effectiveness of the developed dataset and the proposed CCT-based approach.

The main contributions of the paper are presented below:

1. A dataset for whitefly attacks in cotton crops comprising 5135 images was developed and published to help future researchers.
2. A multi-class dataset with ground truth annotation was prepared for our model.
3. A Compact Convolutional Transformer (CCT)-based approach is proposed for the classification.
4. The performance of the proposed CCT-based approach is compared with those of various state-of-the-art models, such as MobileNet, ResNet152v2, VGG-16, and SVM. Experimental results demonstrate that the CCT approach outperformed the compared approaches.

This paper is organized as follows: Section 2 presents the related work, whereas Section 3 describes the dataset used in this work. The proposed CCT-based approach is presented in Section 4, whereas experimental results are discussed in Section 5. Section 6 concludes the paper and highlights the directions for future work.

2. Related Work

Precision agriculture highly depends on image datasets when dealing with computer vision applications. The modern applications work effectively for classification and detection, and the state-of-art models have already demonstrated remarkable accuracy with the benchmark datasets. Manual inspection is not only time-consuming and laborious, but also, late inspection can lead to yield losses. Several deep learning-based techniques have been proposed for leaf disease identification in recent years.

Sujatha et al. [10] compared various machine learning algorithms (random forest (RF), support vector machine (SVM), stochastic gradient descent (SGD)) with deep learning algorithms (VGG-19, VGG-16, Inception-v3) in cotton disease classification. Results showed that VGG-16 outdistanced all other models with an accuracy of 89.5%. However, the number of input samples was very low. Azath et al. [11] proposed a mechanism to detect cotton disease and pest control. The authors used a deep learning technique, predominantly a CNN, for segmentation and classification. The model used a dataset of 2400 images divided into four classes of leaf, namely, minor, spider mite, healthy leaf, and bacterial blight. The model achieved an accuracy of 98% on an apportioned imagery dataset.

Caldeira et al. [12] stated that cotton leaf lesion is a common disease that can affect plant growth. The authors gathered a dataset of 60,000 images mainly based on leaf lesions, and the dataset was further divided into two classes, such as healthy and lesioned leaves. This paper mostly focused on comparing deep learning models with conventional models to validate and substantiate their performances. The depicted results showed that all models achieved accuracy over 70%, and SVM's was above 80%. Using a radial basis function neural network (RBFN) algorithm, Saleem et al. [13] suggested an IoT-based smart system for whitefly detection. The proposed mechanism was tested in the field using IoT sensors. The devices collected 12,896 images, which were connected to web servers.

Pechuho et al. [14] used the YOLO-V3 pre-trained deep learning model. The author used a multi-class dataset of cotton disease from ImageNet's open repository. The model achieved an accuracy of 90% following multiple experiments.

Rothe et al. [15] used a back propagation model to classify multi-class cotton leaves disease. The imagery dataset was collected from different sources physically. The acquired dataset was implemented on Gaussian filters after removing noise using preprocessing techniques.

The above-mentioned studies provide a succinct overview of cotton plant diseases. However, there are no detailed solutions for infectious plant diseases. The datasets employed for the studies are either private or have small sample sizes. Furthermore, because the images are taken by farmers, the image sample data are susceptible to background noise. The deep learning models implemented in the studies performed well on the small datasets. However, they fared badly on large datasets.

The application of state-of-art machine learning algorithms has drawn the attention of researchers. Mojjada et al. [16] worked on multi-class classification using five different types of wheat diseases from the plant village dataset. The authors employed K-means for dataset segmentation using region-based and threshold value techniques.

Neural networks are the baseline architectures for deep learning models. The CNN is a significant classification model with a higher learning rate and low parameters. It learns the spatial features of input images. Furthermore, several studies utilized self-supervised algorithms such as autoencoders [17] to compress data into low dimensions. The convolutional autoencoder employs N layers to learn those low-dimensional data using convolutional layers. Therefore, Bedi [18] proposed a hybrid model of CAE and CNN for plant disease detection. They used a dataset consisting of 4415 images: 3342 images were used for training and 1115 for testing. The computer-aided engineering (CAE) algorithm was employed to reduce the dimensionality of the images, and CNN was used for image classification. The training and testing accuracy were claimed to be 99.35% and 98.38%, respectively.

Chowdhury et al. [19] implemented different segmentation models for the tomato leaf. The plant village database was used for image acquisition. The unbalanced dataset consisted of 18,161 images divided into binary classes, namely, healthy and unhealthy, which were further divided into ten categories. Of the ten classes, one class was healthy; all other classes were labeled as diseased. Efficient Net-B7 outperformed all other models, gaining accuracy of over 99% in binary classification, and Efficient Net-B4 demonstrated accuracy of 99.89%.

Singh [20] collected images of diseased sunflower leaves manually by using digital cameras and other image capturing tools. Particle swarm optimization (PSO) was used for detection and classification. The model provided accuracy of 98%. Bernardes et al. [21] suggested that pathogens are disruptive in plantations and can affect crops' substantially.

The authors developed a framework based on two distinct datasets that were integrated to form a single dataset. The image samples were transformed into HSV and grayscale. SVM was utilized for RGB image classification. The model achieved an accuracy of 96%.

The indiscernible region of interest (RoI) may affect the performances of deep learning models. Even with excellent accuracy, dealing with noisy data may result in the underfitting and overfitting of the model. Plant diseases indicate distinct patterns and dots in the early stages, which require a more robust model with an attention mechanism to detect the disease. However, in the related studies, no attention mechanism was used for detection. We applied an attention-based mechanism for the classification of whitefly attack classification.

Naeem et al. [22] used multiple classification models on a plant-leaf dataset. The labeled classes were then passed to five different models to dissect the accuracy. Multi-layer perceptron (MLP) gained an accuracy of 95% by utilizing 1200 samples.

Zhang et al. [23] implemented a SIFT algorithm to detect corn ear tests and sequence them according to their appearance. The model achieved a maximum accuracy of 97%. Islam et al. [24] implemented different deep learning models for multi-class papaya-leaf classification. The CNN outperformed all other state-of-the-art algorithms, achieving an accuracy of 98.04%. The validation and training loss was 0.79%, which is very low because any CNN requires a large dataset to train.

Arsenovic et al. [25] articulated that the lack of appropriate datasets is a major obstacle in implementing deep learning and computer vision models in the agriculture field. The author proposed a dataset of fourteen different diseases containing over 70,000 images to overcome this issue. Moreover, a hybrid model called Plant Disease Net (PDN) was proposed for classification, which achieved an accuracy of 93%.

Ngugi et al. [26] developed an application for automatic disease detection on the limited number of datasets gathered manually. The MobileNet-V2 model was implemented along with kijani Net, which achieved an accuracy of 90%. Mobilenet-v2 is a pre-trained model used to address classification and detection tasks. Mobilenet-v2 is widely used in agriculture disease detection, and it has shown exquisite results.

3. Materials

Due to the unavailability of a dataset of reasonable size, in this research, we developed a dataset called the AgriPK dataset containing images of the cotton crops. The details of the dataset are presented in the sub-section below.

3.1. AgriPK Dataset

The sample images of the leaves utilized for the proposed research were collected from Bahawalpur, the Southern city of the Punjab province in Pakistan, between August and October 2021.

The South Punjab region is deemed to be the best cotton-growing region. As a result, the whitefly infestation is intense and prominent. We collected 5135 images from Bahawalpur farms for this study. Our dataset is categorized into five classes, as shown in Table 1. The gleaned dataset is unbiased, not very noisy, and unambiguous. On-field images are vulnerable to natural phenomena such as multiple leaves appearing simultaneously, air resistance, and bright sunlight. We established a controlled environment to mitigate environmental factors to eliminate the challenges above. The different classes are shown in Figure 1.

Table 1. Dataset used for this study.

No. of Classes	Categories	No. of Images	Test/Train	Total No. of Images
Our AgriPK Dataset				
1	Healthy	2213	1600/713	5137
2	Unhealthy	2852	2110/741	
3	Mild	210	152/58	
4	Nutrition Deficiency	235	160/75	
5	Severe	2407	1801/675	
Cotton Diseased Dataset				
1	Diseased Cotton Leaves	288	235/53	1951
2	Diseased Cotton Plant	815	602/203	
3	Fresh Cotton Leaves	427	324/104	
4	Fresh Cotton Plant	421	321/101	

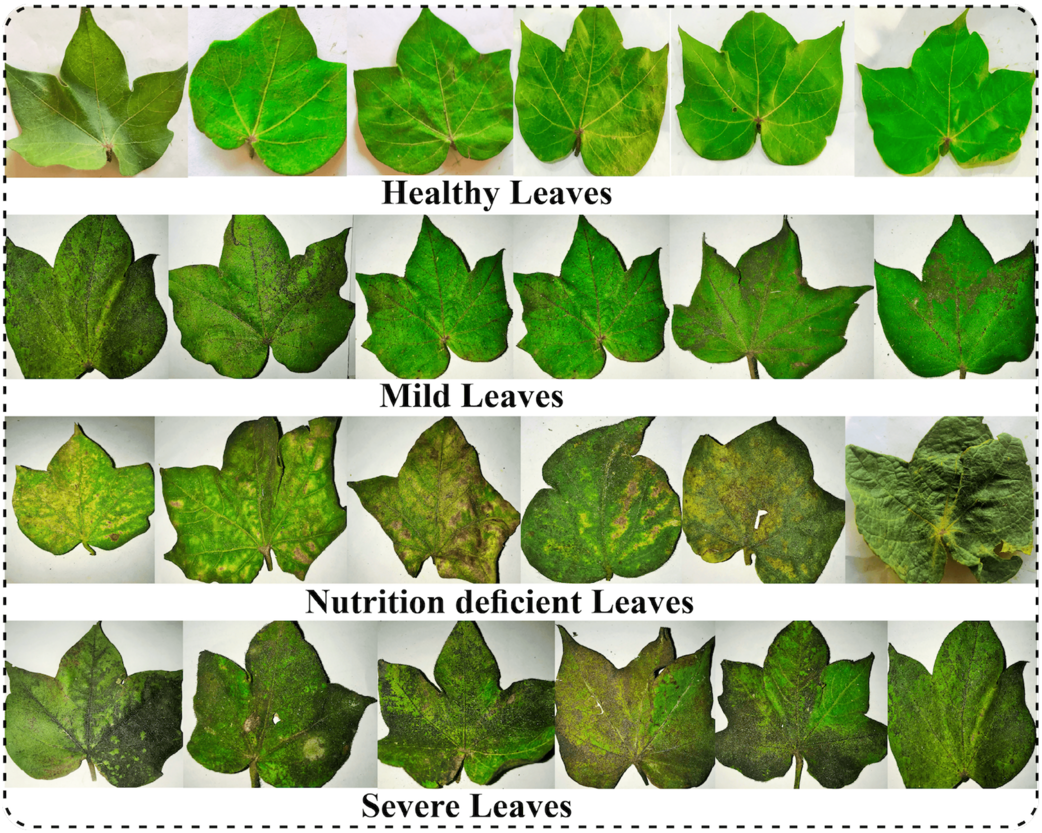


Figure 1. Our proposed AgriPK dataset.

3.1.1. Image Collection

Image capture and labeling is a time-consuming, laborious, and expensive procedure, as it requires significant resources. Multiple devices, including the mobile phones and DSLR cameras, were utilized to acquire the sample images, which ranged in resolution from 5 to 12 megapixels. To ensure the effectiveness of the process, agriculture professionals were involved, who supervised the image gathering and classification processes. After examining numerous cotton fields, around 8000 images were initially captured. To reduce noise, all images were captured in a special environment with artificial light, manually adjusting angles and capturing them at the same distance.

3.1.2. Professional Annotation

After collecting the image data, we annotated them so that we could effectively perform the task of classification. As professional knowledge is required for manual data annotation, the samples were labeled by agriculture professionals from the Islamia University of Bahawalpur, Pakistan. The dataset given to the annotator was first classified into five categories: (a) healthy, (b) unhealthy, (c) nutritional deficit, (d) mild, and (e) severe, as illustrated in Figure 1. Moreover, after experts verification, we placed each image in the corresponding folders. To train the dataset, we distributed the samples into five classes and annotated each sample with a class label. Each image data sample is represented with a label vector of five values.

3.2. Existing Cotton Disease Dataset

There are various public datasets of cotton diseases available. We used the Cotton Disease Dataset [27] to compare our study to the existing dataset. The Cotton Disease Dataset Figure 2 is a free resource found in the Kaggle repository. The four classes in the dataset are: (i) diseased plants, (ii) diseased leaves, (iii) healthy plants, and (iv) healthy leaves. The dataset includes 1918 samples. Real-time cotton photos were captured in the field during the daytime. The Cotton Disease Dataset is one of the few cotton datasets that are currently available online. We contrasted our dataset with a dataset of infected cotton.

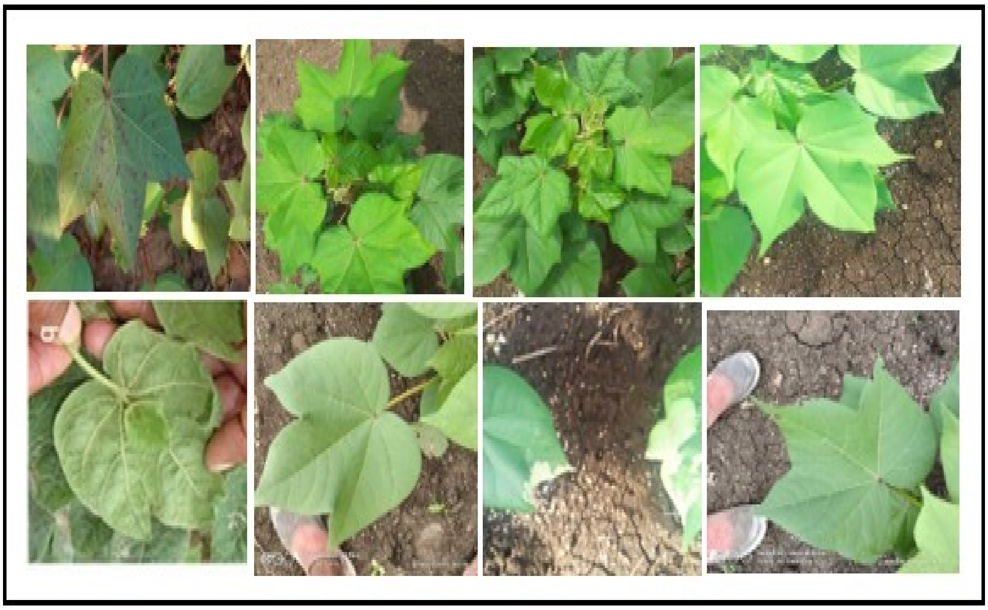


Figure 2. Cotton Disease Dataset [27].

4. Methods

This study presents a classification approach for whitefly attacks on cotton plants. We introduced the AgriPK dataset and implemented a Compact Convolutional Transformer (CCT) for classification. The workflow diagram of the presented research is shown in Figure 3.

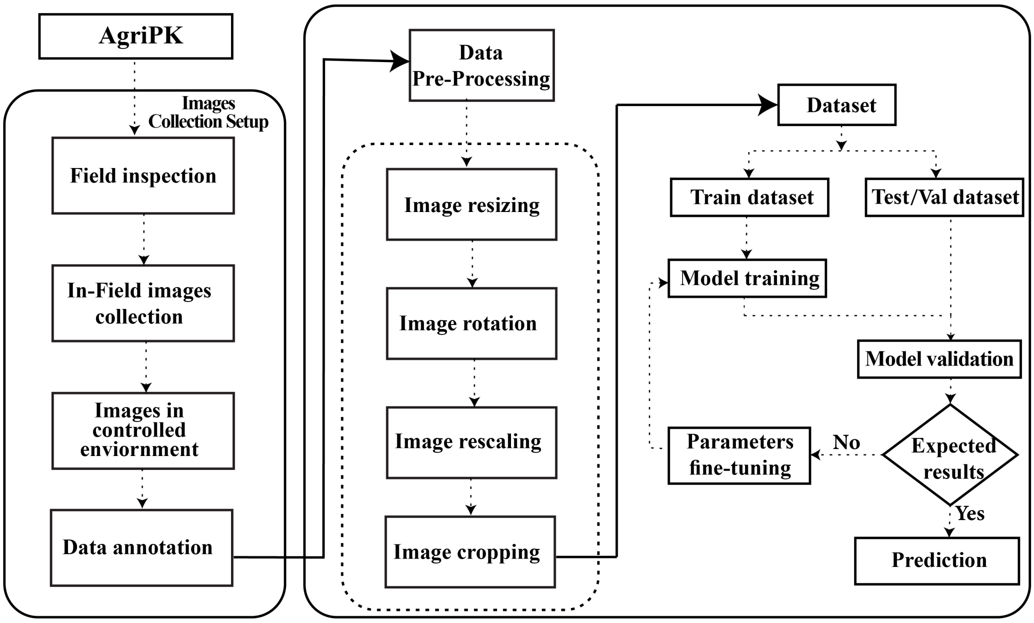


Figure 3. Work flow diagram of proposed scheme.

4.1. Data Pre-Processing Module

Data pre-processing is the preliminary procedure for improving and adjusting input image quality, size, and flipping. For data cleaning, many pre-processing techniques, such as rotation, scaling, cropping, and resizing, were used which were imported from the Python framework. Many re-scaling techniques are mostly used, but data normalization, also known as Min-Max scaling, is a preferred technique. Noise removal is an important aspect of deep learning models where model accuracy is highly dependable on the image background. While dealing with more acute and sensitive diseases and finding the region of interest (RoI), background noise can affect model performance.

Image rotation is required to fix the input images to the same length and angle to enable the model to smoothly classify the required output. We used data augmentation techniques, such as image translation, scaling, and image rotation, to balance the dataset during the experiment. Image scaling was used to resize an image, whereas image rotation was used to rotate an image to a specific degree from its central axis. Several other techniques, such as image cropping, horizontal rotation, adjusting angles, removing noise, and changing the image dimension and input size, were also applied as shown in Figure 4.

4.2. Classification Model

Compact Convolutional Transformer

In this study, a Compact Convolutional Transformer (CCT) was used for whitefly classification. The CCT [28] is the most recent version of the Vision Transformer (Vit), which is a compact model for image processing problems. The conventional transformer is considered data hungry for image processing tasks. Many authors proposed different techniques to address this issue, such as DeiT [29], ConViT [30], CvT [31], and T2T-ViT [32]. However, the CCT model outperformed all existing state-of-art techniques. The model was trained and evaluated on three types of datasets: small-scale and low-resolution images (FashionMNIST, MNIST, and CIFAR-10/100), medium-sized (Image Net), and small-scale high-resolution images (Flowers-102). It utilized few parameters and thus minimized the time complexity of the model. The model architecture as shown in Figure 4 includes various novel frameworks that distinguish it from prior approaches. The model architecture is described in the following paragraphs.

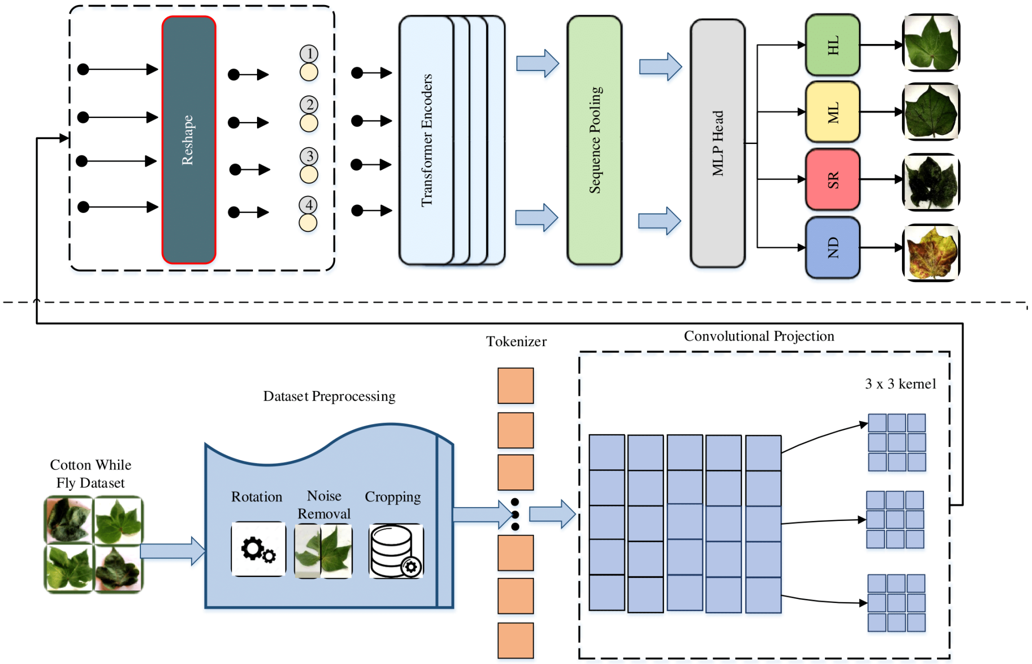


Figure 4. CCT system model proposed in this work.

First, the augmented RGB images are fed to an input layer with dimensions of 150 × 150 × 3. A dual tokenizer with a kernel size of 3 × 3 convolutional layers and a stride size of 1 is applied to preserve the boundary-level information. The resultant feature map attained from convolutional layer helps to alleviate self-attention computation complexity. The rectified linear activation (Relu) is used as an activation function in conjunction with the "he normal" kernel-initializer with padding size 1. The mathematical representation of convolutional layers employed in the CCT tokenizer is given as follows:

$$Xo = MaxPool(ReLU(conv2d(x)) \tag{1}$$

where MaxPool in Equation (5) [28] represents the convolutional layer. The feature map

$$x\epsilon R^{H\times W\times C}, \tag{2}$$

as shown in Equation (2), represents the extraction of local features. C is the number of channels, and H represents the resolution of images. Due to convolutional blocks, the model is not dependent on image resolution, as it creates a feature map and preserves locally partial information. Then, an attention-based encoder–decoder mechanism is applied on a high-dimensional feature map area to extract the spatial and intensity-based pertinent features in the multi-class cotton dataset. All the features are flattened into a 1D array and fed to the sequence embedding layer. It locates information within the sequence regarding the relative locations of image patches. After passing through the self-attention module, this class embedding predicts each class of an input image.

$$MultiHead(Q, K, V) = [Head_1..., Head_\mathrm{h}]W_\mathrm{o} \tag{3}$$

where queries, keys, and values are represented by Q, K, and V, respectively, as shown in Equation (6) [29]. After positional embedding, we applied multi-layer perceptron (MLP) stacking on cotton image patches with their ground truths for whitefly attack classification. The deep learning models correspond to various hyperparameter tuning techniques during pre-processing and model training. We kept an input size of 224, as shown in Table 2; the input size directly affects the model performance. The input image is the sample image on which the model will predict. Therefore, the algorithm required a fixed size to learn the temporal and spatial features. Similarly, the batch size is also a hyperparameter where a total number of sample images is processed prior to model updates. The batch size can be increased or decreased depending on the dataset. For our experiments, we used a batch size of 64 with 100 epochs. Usually, training a neural network would require a series of epochs where training data will pass through every single epoch.

Table 2. Parameters used during training.

Parameter	Value
Learning rate	0.06
Batch size	64
Input size	224
No. of epochs	100
Weight decay	0.006

The accuracy, precision, recall, and F1-score were among the evaluation measures used to assess the suggested method's performance. Precision is the ratio of correct positive results to all positive instances predicted by the classifier. The ratio of true-positive predictions over the sum of true-positive and false-negative predictions is used to calculate recall. The following are the mathematical representations of these measures.

$$Accuracy = \frac{No. of\ correct\ instances}{total\ no. of\ input\ samples} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1\text{-}score = \frac{2(PrecisionRecall)}{Precision + Recall} \tag{7}$$

Positive and negative class samples are represented by P and N, respectively. The rightly identified positive class is referred to as true positive (TP). Similarly, true negative (TN) depicts the identification of an abnormal class. FN and FP, on the other hand, stand for the misclassified normal and abnormal classes.

5. Results and Discussion

To demonstrate the effectiveness of CCT-based approach on the AgriPK dataset, extensive experimentation was performed. During the experiments, the images were collected in the natural environment and labeled under the supervision of experts. Google Colab and Kaggle were used to train the deep learning and CCT models using Python 3.0. All the deep learning models were implemented using Keras, TensorFlow, and scikit learn libraries, which are open-source libraries. We compared the proposed CCT-based approach with several state-of-the-art deep learning models, namely, MobileNet, VGG16, and Resnet 152 V2. The brief overview of the compared models is given below.

5.1. MOBILE NET

MobileNet [33] is reliable and efficient when applied to real-world applications. It comes up with many frameworks. The standard convolutional layer is replaced by depth-wise separable convolution to make it a lighter model. MobileNet-v2 is based on 53 hidden layers with a ReLU activation function. These are already pre-trained on millions of images from various imagery repositories.

5.2. VGG-16

VGG-16 [34] is a vision model, which is being utilized in multiple disease classification and detection scenarios. The model architecture is based on a complex 16-layer structure. The activation layer is present after the block of the input convolutional layer. The model randomly updates weights after each training layer to minimize error.

5.3. RESNET 152-V2

Resnet152-V2 [35] is the state-of-the-art model used for image classification. It consists of multiple hidden layers. The input data pass through the reshaped layer and then to the flattened layer. Moreover, the dense layer in the model consists of 128 neurons. To avoid model overfitting, a drop-out layer is added. Finally, a SoftMax function for image classification is used for prediction.

5.4. Yolo V5

Yolo V5 [36] is based on the conventional architecture of the Yolo series, which was put forward by ultralytics. It is mainly based on three steps known as the model backbone, neck, and head for one-step object detection. To mitigate the time consumption problems and to attribute the duplicate features, a CSP Net (cross-stage partial networks) is employed. The input image fed to the model is processed by the model backbone, where the important feature is extracted using CSP Nets. PANet (path aggregation network) [37] is utilized for image scaling and to formulate feature pyramids in the model neck module. However, the model head of Yolo v5 is similar to those of Yolo v3 and Yolo v4. The model head primarily focuses on anchoring boxes and predicting final outputs with bounding boxes. The model follows regression approaches for detection inducing fewer parameters.

5.5. Performance on AgriPK

To demonstrate the efficacy of our proposed AgriPK dataset we experimented several state-of-the-art deep learning models and the latest version of the Vision Transformer, namely, the CCT. We conducted several experiments on the small and medium-sized datasets to evaluate their performances. The results show the effectiveness of proposed model on small and medium-sized datasets as shown in Table 3. A total of 5135 images were used for training and testing, as shown in Figure 5. We used parameter tuning mechanisms to test the models' performances, as shown in Figure 6. The efficacy of our proposed dataset was evaluated based on the true positive and false negative rates as shown in Figure 7.

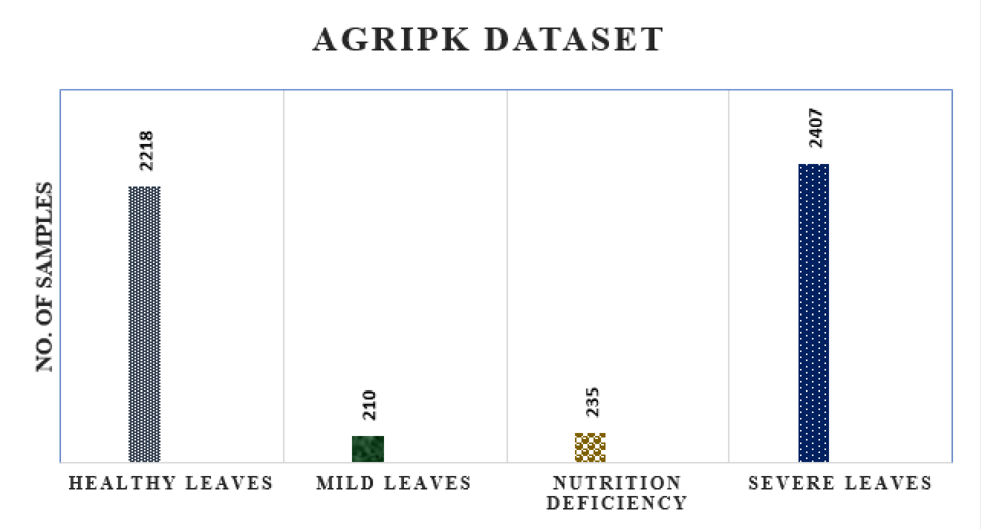
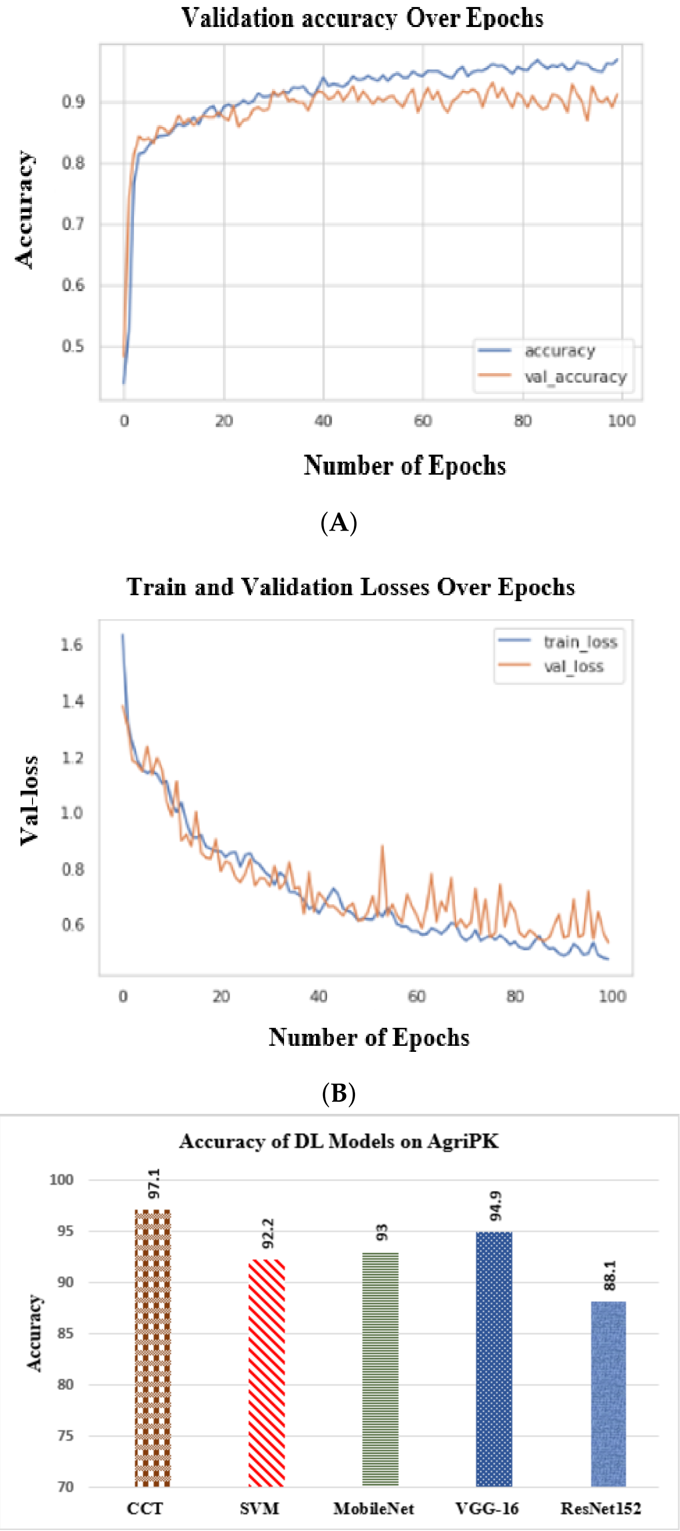


Figure 5. AgriPK sample distribution.



(**A**)

(**B**)

(**C**)

Figure 6. (**A**) Validation accuracy over epochs. (**B**) Training and validation loss epochs. (**C**) Accuracy of all DL models.

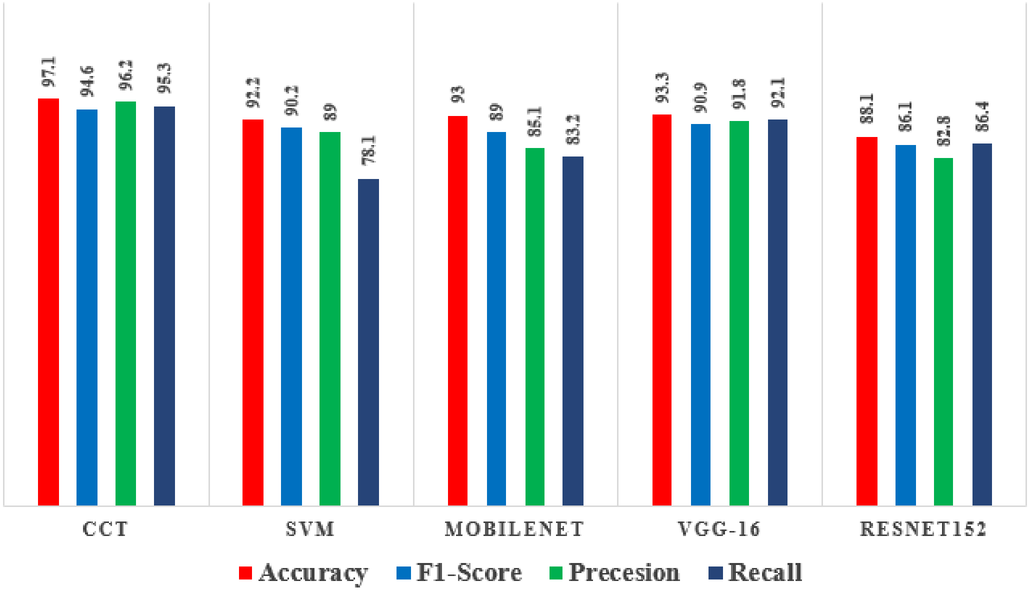


Figure 7. Benchmark models' performances on AgriPK as shown in Table 3.

Table 3. Benchmark models' performances on AgriPK.

Parameters	Mobile Net	VGG-16	ResNet-152	YoloV5	SVM	CCT
Accuracy (%)	93	93.3	88.1	95.1	92.2	97.1
F1-Score	89	90.9	86.1	93.2	90.2	94.6
Precision	85.1	91.8	82.8	91.1	89	96.2
Recall	83.2	92.1	86.4	85.7	78.1	95.3

We utilized five classes in the first experiments, namely, (i) healthy, (ii) unhealthy, (iii) severe, (iv) mild, and (v) nutrition deficit. The final experiment used a 120 × 120 × 3 input image size for 100 epochs. A learning rate of 0.001 and a batch size of 64 were utilized for the experiment. To improve the model's accuracy, the image size was lowered to 120, leading to an accuracy of 97%, as shown in Figure 7. We initiated the trials with an image size of 550 × 550 × 3. In the first stage, the model was explicit 88 percent of the time. Moreover, the batch size was lowered to 32 to comprehend the varied CCT parameters better as given in Figure 8.

The CCT demonstrated the highest accuracy over 100 epochs, with the least validation and training losses. When compared to MobileNet and VGG-16, the CCT took more time. During our trails, we noticed that CCT's time consumption was slightly higher than those of Mobilenet and VGG-16. This is because CCT is based on complex transformer architectures, as shown in Table 4.

Table 4. Comparison of computational time and number of parameters.

Model	No. of Params.	Training Time
CCT	897,413	7200 ms
VGG-16	750,567	6600 ms
MobileNet	699,156	6300 ms
Resnet 152	950,567	8900 ms

The Resnet model achieved the lowest accuracy of 88% among all the models. The SVM model achieved 92% accuracy, as shown in Table 5. Resnet has a complex structures; therefore, it required more training time as compared to other models. On the other hand, VGG-16 achieved around 95% accuracy. However, the validation loss as compared to the CCT model was greater than 0.9% for VGG-16. We learned that varying the training epochs also effects the performance of classification modelas shown in Figure 8. Mobilenet and SVM showed accuracies of 93% and 92.2%, respectively.A real-time detection model, Yolov5, was employed on the proposed AgriPK dataset to evaluate its efficacy. The model showed an accuracy of 95.1%. It can be noted that the CCT resulted in balanced recall and precision, whereas the Yolo v5 lagged when applied to similar domains. Reasons for the tangible decreases in recall and precision were the indistinguishable circles and constant colors in the input images. Although all state-of-art model depicted good accuracy, for whitefly diseased leaf classification, CCT showed the best classification accuracy.

Table 5. Performance evaluation of AgriPK and Cotton Disease Dataset.

Model	Cotton Disease Dataset	AgriPK Dataset	F1-Score Increment
CCT	91.8	94.6	2.8%
SVM	80.2	90.2	9.6%
Mobile Net	80.4	89	8.6%
VGG-16	89.8	90.6	0.8%
ResNet152-v2	80.7	86.1	5.7%
Yolo V5	92.6	93.2	0.5%

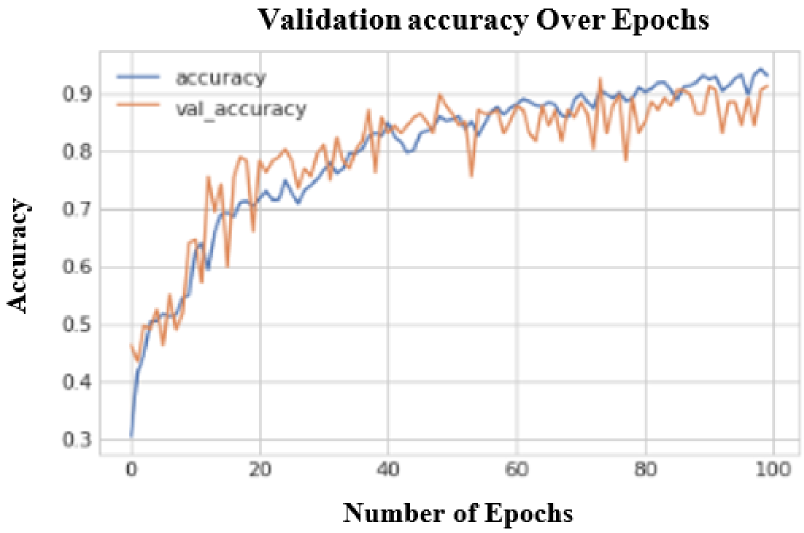


Figure 8. Training loss.

5.6. Performance on Cotton Disease Dataset

We also evaluated the accuracy of our proposed CCT model on the existing Cotton Disease Dataset. The CCT model showed an accuracy of 95% on the Cotton Disease Dataset. As shown in Figure 8, the simulations were conducted over 100 epochs to test the model. However, we kept the corresponding parameters consistent during the comparison session. Initially, we conducted experiments over 20 epochs for CCT, and the accuracy was 85%. Subsequently, after adjusting the epochs to 50, we attained the best results.

The highest accuracies attained by using the CCT were 97.1% and 95.4% on the AgriPK dataset and Cotton Disease Dataset, respectively, as demonstrated in Figure 9. The reason CCT performed well on both datasets is that it is highly adaptive and has an effective learning rate, ascribed to the presence of MLP layers and convolutional blocks in the model architecture. We evaluated the performances of CCT and deep learning models using various evaluation metrics. We noticed that sufficient trainable data are required for complex models for classification and detection.

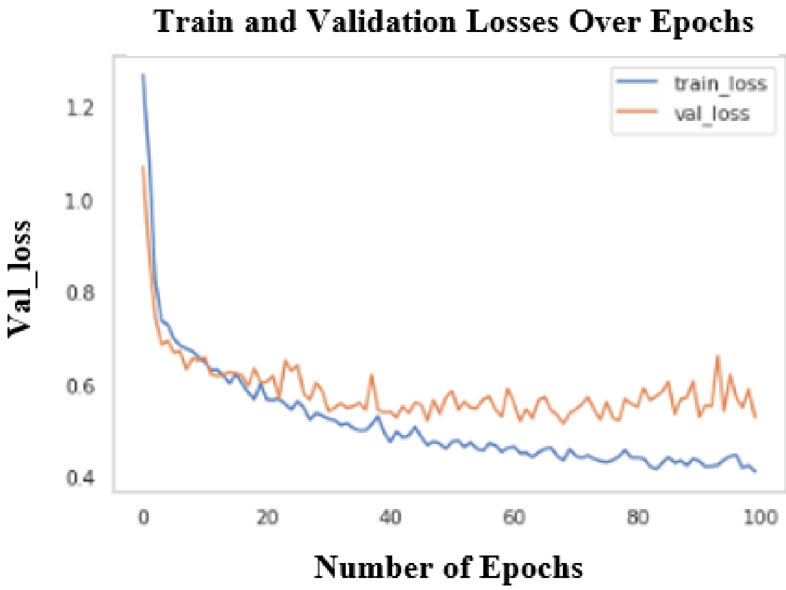


Figure 9. Validation accuracy.

We utilized F1-score, precision, recall, and accuracy as key metrics for model evaluation, as stated earlier in Table 3. To train and test the models, the data were split into 25% and 75%, respectively (X and Y). Throughout the simulations, we kept the training and testing settings consistent. The SVM and Mobile Net performed poorly on both parts of the Cotton Disease Dataset, whereas the F1-score on our AgriPK dataset was substantially higher as shown in Figure 10. Yolov5 showed a slight increase in the F1-score in both datasets. The model utilizes a real-time detection mechanism. However, the baseline datasets are prone to having similar features; therefore, model performance deteriorated when new data were given to the model. The top F1-score of 94.6% of CCT on both datasets demonstrated its effectiveness. We employed transfer learning and pre-trained layers, with customized layers replacing the last 15% of training data as shown in Figure 11.

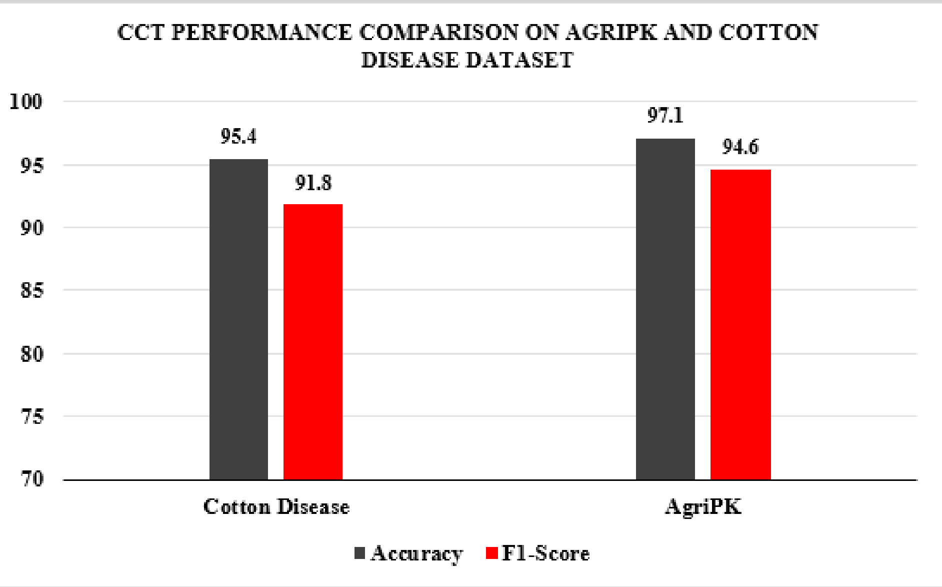


Figure 10. Accuracy and F1-score.

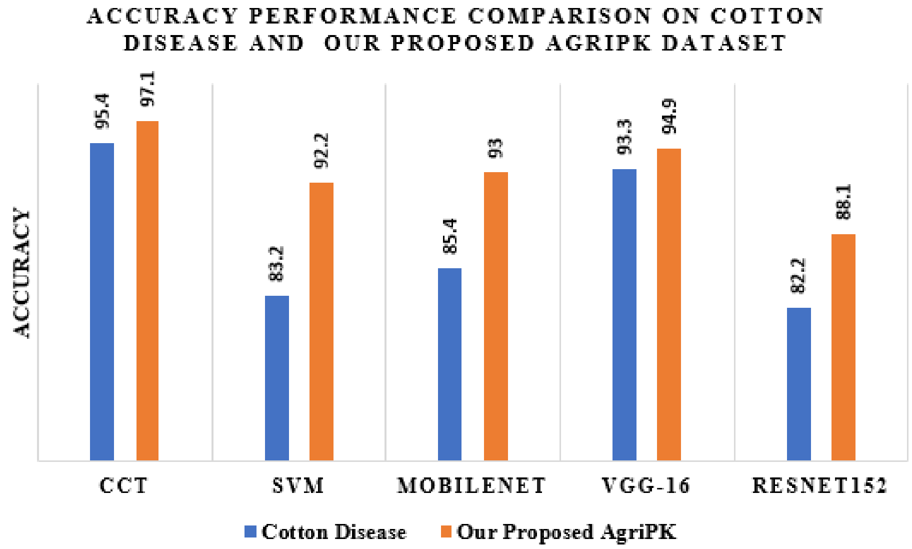


Figure 11. Performance comparison.

6. Conclusions and Future Work

The agriculture domain lacks enough datasets, and there are not enough imagery samples present online to help future research. As a result, there is a need to generate new datasets for the future of AI in agriculture. To that end, we presented AgriPK, a dataset based on cotton leaves damaged by whiteflies. The dataset was created in supervised controlled conditions to eliminate noise and irrelevant components from the image. The dataset contains 5137 images and is publicly available. Furthermore, we used a Compact Convolutional Transformer on the developed dataset to ascertain its generalizability. Despite the model's intricacy, it showed strong performance when compared to other deep learning models. However, other state-of-the-art models also demonstrated substantially better accuracy.

In future research, we will focus on enhancing our ApriPK dataset's samples. Furthermore, the proposed model can be utilized on other large public datasets for performance evaluation. In the future, we will work on different cotton diseases and will work to develop a hybrid model for effective pest classification and detection.

Author Contributions: Conceptualization, A.I.J., A.A., H.A.K. and A.K.; methodology, A.I.J., A.A., H.A.K., G.N., A.K., H.T.R. and S.K.; software, A.I.J., A.A. and H.A.K.; visualization, A.I.J., A.A. and H.A.K.; writing—original draft, A.I.J., A.A., H.A.K., G.N., A.K., H.T.R. and S.K.; data curation, A.I.J., A.A., H.A.K., G.N., A.K., H.T.R. and S.K.; supervision, A.A., H.A.K. and A.K.; writing—review and editing, A.I.J., A.A., H.A.K., G.N., A.K., H.T.R. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The dataset is available at Kaggle https://doi.org/10.34740/KAGGLE/DSV/2927481 (accessed on 10 August 2022), and the code is available upon request.

Acknowledgments: The AgriPK dataset was collected and labeled with the help of Naveed Iftikhar and M. Irfan Akram , Entomology Department Islamia University Bahawalpur, Pakistan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shuli, F.; Jarwar, A.H.; Wang, X.; Wang, L.; Ma, Q. Overview of the cotton in Pakistan and its future prospects. *Pak. J. Agric. Res.* **2018**, *31*, 291–418. [CrossRef]
2. Ali, A.; Ahmed, Z. Revival of Cotton Pest Management Strategies in Pakistan. *Outlooks Pest Manag.* **2021**, *32*, 144–148. [CrossRef]

3. Poorter, M.d.; Browne, M. The Global Invasive Species Database (GISD) and international information exchange: Using global expertise to help in the fight against invasive alien species. In *Plant Protection and Plant Health in Europe: Introduction and Spread of Invasive Species, Held at Humboldt University, Berlin, Germany, 9–11 June 2005*; British Crop Protection Council: Alton, UK, 2005; pp. 49–54.
4. Zia, K.; Hafeez, F.; Bashir, M.H.; Khan, B.S.; Khan, R.R.; Khan, H.A.A. Severity of cotton whitefly (*Bemisia tabaci* Genn.) population with special reference to abiotic factors. *Pak. J. Agric. Sci.* **2013**, *50*, 217–222.
5. Hara, P.; Piekutowska, M.; Niedbała, G. Selection of independent variables for crop yield prediction using artificial neural network models with remote sensing data. *Land* **2021**, *10*, 609. [CrossRef]
6. Sedri, M.H.; Niedbała, G.; Roohi, E.; Niazian, M.; Szulc, P.; Rahmani, H.A.; Feiziasl, V. Comparative Analysis of Plant Growth-Promoting Rhizobacteria (PGPR) and Chemical Fertilizers on Quantitative and Qualitative Characteristics of Rainfed Wheat. *Agronomy* **2022**, *12*, 1524. [CrossRef]
7. Legaspi, K.R.B.; Sison, N.W.S.; Villaverde, J.F. Detection and Classification of Whiteflies and Fruit Flies Using YOLO. In Proceedings of the 2021 13th International Conference on Computer and Automation Engineering (ICCAE), Melbourne, Australia, 20–22 March 2021; pp. 1–4.
8. Tulshan, A.S.; Raul, N. Plant leaf disease detection using machine learning. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
9. Nesarajan, D.; Kunalan, L.; Logeswaran, M.; Kasthuriarachchi, S.; Lungalage, D. Coconut disease prediction system using image processing and deep learning techniques. In Proceedings of the 2020 IEEE 4th International Conference on Image Processing, Applications and Systems (IPAS), Genova, Italy, 9–11 December 2020; pp. 212–217.
10. Sujatha, R.; Chatterjee, J.M.; Jhanjhi, N.; Brohi, S.N. Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocess. Microsyst.* **2021**, *80*, 103615. [CrossRef]
11. Azath, M.; Zekiwos, M.; Bruck, A. Deep learning-based image processing for cotton leaf disease and pest diagnosis. *J. Electr. Comput. Eng.* **2021**, *2021*.
12. Caldeira, R.F.; Santiago, W.E.; Teruel, B. Identification of cotton leaf lesions using deep learning techniques. *Sensors* **2021**, *21*, 3169. [CrossRef] [PubMed]
13. Saleem, R.M.; Kazmi, R.; Bajwa, I.S.; Ashraf, A.; Ramzan, S.; Anwar, W. IOT-Based Cotton Whitefly Prediction Using Deep Learning. *Sci. Program.* **2021**, *2021*, 8824601. [CrossRef]
14. Pechuho, N.; Khan, Q.; Kalwar, S. Cotton Crop Disease Detection using Machine Learning via Tensorflow. *Pak. J. Eng. Technol.* **2020**, *3*, 126–130.
15. Rothe, P.; Kshirsagar, R. Cotton leaf disease identification using pattern recognition techniques. In Proceedings of the 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015; pp. 1–6.
16. Mojjada, R.K.; Kumar, K.K.; Yadav, A.; Prasad, B.S.V. Detection of plant leaf disease using digital image processing. *Mater. Today Proc.* **2020**. [CrossRef]
17. Bisong, E. Autoencoders. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Springer: Cham, Switzerland, 2019; pp. 475–482.
18. Bedi, P.; Gole, P. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artif. Intell. Agric.* **2021**, *5*, 90–101. [CrossRef]
19. Chowdhury, M.E.; Rahman, T.; Khandakar, A.; Ayari, M.A.; Khan, A.U.; Khan, M.S.; Al-Emadi, N.; Reaz, M.B.I.; Islam, M.T.; Ali, S.H.M. Automatic and reliable leaf disease detection using deep learning techniques. *AgriEngineering* **2021**, *3*, 294–312. [CrossRef]
20. Singh, V. Sunflower leaf diseases detection using image segmentation based on particle swarm optimization. *Artif. Intell. Agric.* **2019**, *3*, 62–68. [CrossRef]
21. Bernardes, A.A.; Rogeri, J.G.; Oliveira, R.B.; Marranghello, N.; Pereira, A.S.; Araujo, A.F.; Tavares, J.M.R. Identification of foliar diseases in cotton crop. In *Topics in Medical Image Processing and Computational Vision*; Springer: Cham, Switzerland, 2013; pp. 67–85.
22. Naeem, S.; Ali, A.; Chesneau, C.; Tahir, M.H.; Jamal, F.; Sherwani, R.A.K.; Ul Hassan, M. The classification of medicinal plant leaves based on multispectral and texture feature using machine learning approach. *Agronomy* **2021**, *11*, 263. [CrossRef]
23. Zhang, X.; Liu, J.; Song, H. Corn ear test using SIFT-based panoramic photography and machine vision technology. *Artif. Intell. Agric.* **2020**, *4*, 162–171. [CrossRef]
24. Islam, M.A.; Islam, M.S.; Hossen, M.S.; Emon, M.U.; Keya, M.S.; Habib, A. Machine learning based image classification of papaya disease recognition. In Proceedings of the 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 5–7 November 2020; pp. 1353–1360.
25. Arsenovic, M.; Karanovic, M.; Sladojevic, S.; Anderla, A.; Stefanovic, D. Solving current limitations of deep learning based approaches for plant disease detection. *Symmetry* **2019**, *11*, 939. [CrossRef]
26. Ngugi, L.C.; Abelwahab, M.; Abo-Zahhad, M. Recent advances in image processing techniques for automated leaf pest and disease recognition—A review. *Inf. Process. Agric.* **2021**, *8*, 27–51. [CrossRef]
27. D3v. Cotton Disease Dataset, Version 1. 2020. Available online: https://www.kaggle.com/datasets/janmejaybhoi/cotton-disease-dataset (accessed on 6 January 2022).

28. Hassani, A.; Walton, S.; Shah, N.; Abuduweili, A.; Li, J.; Shi, H. Escaping the big data paradigm with compact transformers. *arXiv* **2021**, arXiv:2104.05704.
29. d'Ascoli, S.; Touvron, H.; Leavitt, M.L.; Morcos, A.S.; Biroli, G.; Sagun, L. Convit: Improving vision transformers with soft convolutional inductive biases. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 2286–2296.
30. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 10347–10357.
31. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. Cvt: Introducing convolutions to vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 22–31.
32. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.H.; Tay, F.E.; Feng, J.; Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 558–567.
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. ultralytics/yolov5: v6.2. YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai Integrations. Available online: https://github.com/ultralytics/yolov5/releases (accessed on 10 August 2022).
37. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.