Deep Learning Models to Determine Nutrient Concentration in Hydroponically Grown Lettuce Cultivars (Lactuca sativa L.)

Mostofa Ahsan 1, Sulaymon Eshkabilov 2, Bilal Cemek 3, Erdem Küçüktopcu 3, Chiwon W. Lee 4 and Halis Simsek 5,*

1 Department of Computer Sciences, North Dakota State University, Fargo, ND 58108, USA; mostofa.ahsan@ndsu.edu
2 Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, ND 58108, USA; sulaymon.eshkabilov@ndsu.edu
3 Department of Agricultural Structures and Irrigation, Ondokuz Mayis University, Samsun 55139, Turkey; bcemek@omu.edu.tr (B.C.); erdem.kucuktopcu@omu.edu.tr (E.K.)
4 Department of Plant Sciences, North Dakota State University, Fargo, ND 58108, USA; chiwon.lee@ndsu.edu
5 Department of Agricultural and Biological Engineering, Purdue University, West Lafayette, IN 47907, USA

* Correspondence: simsek@purdue.edu

Abstract: Deep learning (DL) and computer vision applications in precision agriculture have great potential to identify and classify plant and vegetation species. This study presents the applicability of DL modeling with computer vision techniques to analyze the nutrient levels of hydroponically grown four lettuce cultivars (Lactuca sativa L.), namely Black Seed, Flandria, Rex, and Tacitus. Four different nutrient concentrations (0, 50, 200, 300 ppm nitrogen solutions) were prepared and utilized to grow these lettuce cultivars in the greenhouse. RGB images of lettuce leaves were captured. The results showed that the developed DL’s visual geometry group 16 (VGG16) and VGG19 architectures identified the nutrient levels of lettuces with 87.5 to 100% accuracy for four lettuce cultivars, respectively. Convolution neural network models were also implemented to identify the nutrient levels of the studied lettuces for comparison purposes. The developed modeling techniques can be applied not only to collect real-time nutrient data from other lettuce type cultivars grown in greenhouses but also in fields. Moreover, these modeling approaches can be applied for remote sensing purposes to various lettuce crops. To the best knowledge of the authors, this is a novel study applying the DL technique to determine the nutrient concentrations in lettuce cultivars.

Keywords: image processing; nutrient level; lettuce; deep learning; RGB

1. Introduction

Lettuce (Lactuca sativa L.) is grown under a wide range of climatic and environmental conditions, and it is unlikely that any one variety would be ideally suited to all locations [1]. The high value of vegetable production encourages growers to apply high nitrogen (N) rates and frequent irrigation to ensure high yields. N is an essential macronutrient required for the productive leaf growth of lettuce. An optimal amount of N is critical to maintaining healthy green lettuce leaves, while high N concentration could be detrimental to leaf and root development. Similarly, excess N application increases both environmental concerns and the cost of lettuce production. Moreover, the recent regulation requires all lettuce growers to keep track of the amount of fertilizers and irrigation water that are used in the field. Therefore, an appropriate nutrient management plan with the prediction of optimal N requirement of lettuce results in higher crop yield [2,3].

Generally, two standard procedures, including destructive and non-destructive methods, have been used for assessing crop N status. One conventional method is laboratory-based content measurement using an oven-drier and a scale to measure N-concentration on sampled lettuce leaves, which is a destructive method. This type of approach includes
leaf tissue N analysis, petiole sap nitrate analysis, monitoring soil N status, and so forth. For example, an integrated diagnosis and recommendation system was used to calculate leaf concentration norms [4]. This method was also found to be accurate in determining nutrient concentrations. These techniques are generally labor-intensive, time-consuming, and require potentially expensive equipment. Moreover, they may affect other measurements or experiments because of the detachment of leaves from the plants.

In contrast, non-destructive methods are simple, rapid, cheaper, and save labor compared to destructive methods, and they can determine N concentration without damaging the plant. For instance, a morphological growth profile measurement technique can be used to determine lettuce growth profiles and nutrient levels. This morphological method uses periodic measurements of lettuce leaf area changes using triangular and ellipse area-based flap patterns on specific parts of a selected leaf, and then after completing the morphological data collection, leaf stem growth and overall nutrient contents of the selected parts of the leaf and the whole lettuce are calculated [5]. This morphological measurement method is precise and well correlated with conventional dried content measurements. The method is also slow and requires a large number of accurate measurements. Among the non-destructive methods, the digital image processing technique has been employed effectively for predicting the N status of crops. For instance, using a hyperspectral imaging technique of freshly cut lettuce leaves was found to be not only highly accurate in nutrient level determination but also in predicting nutrient changes with respect to the amount of applied fertilizers, evaluating contamination, and determining shelf-life [6–8].

As with many bio-systems, observing nutrient levels or identifying plant growth levels is highly complex and eventually linked to dynamic environment variables. Two basic modeling approaches are proven effective, which are “knowledge-driven” and “data-driven”. The knowledge-driven approach relies on existing domain knowledge, whereas data-driven modeling can formulate solutions from historical data without using domain knowledge. Data-driven models such as machine learning techniques, artificial neural networks, and support vector machines have been very efficient for the last decade because of their versatile applications in different fields [9,10]. Artificial intelligence applications have been successfully implemented in other agricultural domains such as the Normalized Difference Vegetation Index (NDVI), soil pH level measurements, yield prediction, etc. [11,12]. These solutions are formulated with both tabular and visual data types. Recent research indicates that scientists rely on image analysis for quick answers to questions about precision agriculture [13]. Since we are trying to solve an issue that can be determined with visual detection, image analysis was deemed a promising concept to classify nutrient levels in various lettuce breeds. With the advancement of computer technology, the ability to handle large data sets, including image data, has great potential. Moreover, novel computation algorithms and software applications have been developed by applying machine learning (ML) and deep learning (DL) techniques to process large sets of images [14]. For instance, DL techniques with pre-trained model approaches employ Visual Geometry Group (VGG) models, such as the VGG16 and VGG19 models, which were proven to be effective in image recognition problems like leaf disease detection, beef cut detection, and soil health detection using fewer input images to produce better classification accuracy.

DL techniques applied to hyperspectral imaging data can be used to extract plant characteristics and trace plant dynamics or environmental effects. Recently, the ML and DL techniques have been progressively used to analyze and predict a variety of complex scientific and engineering problems [15–19], and they are therefore becoming more and more popular. One of the recent studies employing DL techniques applied the VGG16 and multiclass support vector machine modeling (MSVM) approaches to identify and classify eggplant diseases [20]. The study results demonstrated that applications of the VGG16 and MSVM model approaches resulted in 99.4% accuracy in classifying diseases in eggplants.

To the authors’ best knowledge, there is no published study to date has applied DL to evaluate the concentrations of nutrients in various lettuce cultivars. The above-discussed
destructive approaches have a few significant shortcomings, and the other non-destructive measurement methods require special tools, technical qualifications, and long processing time to estimate crop nutrient levels. Therefore, there is a need to develop a simple, rapid, economical, and accurate method to estimate the concentration of nutrients in lettuce cultivars grown in the greenhouse, which was chosen to be this study’s core objective.

2. Materials and Methods

2.1. Plant Material, Cultivation Condition, and Image Acquisition

In this study, four different lettuce cultivars, namely Rex, Tacitus, Flandria, and Black Seeded Simpson, were grown hydroponically in four different concentrations of nutrient fertilizers, 0, 50, 200, and 300 ppm, to investigate the influence of various nitrogen concentrations on the performance of lettuce grown [21]. Reverse osmosis water was used for the 0 (zero) ppm N solution as a control. The necessary parameters, including nitrate (NO₃⁻), calcium (Ca²⁺), potassium (K⁺), tissue soluble solid content, chlorophyll concentration, and SPAD values, were measured in the laboratory or greenhouse conditions and presented in our previous study [6].

Additionally, the composition of elemental N, P, and K used in the different nutrient solutions during the hydroponic experiments was presented in our previous study [6]. At the beginning of the experiments, the lettuce cultivars were planted in Rockwool cube slabs, and two weeks old seedlings were transferred in 10-L plastic tubs containing different levels of N solutions with 20-20-20 commercial analysis (N-P₂O₅-K₂O). The nutrient solutions were aerated continuously using compressed air. The nutrient solutions were replenished weekly. The lettuce cultivars in plastic tubs were grown for 3 weeks and harvested accordingly.

Before the harvesting, all the lettuce images were captured in the greenhouse using a digital camera (CANON EOS Rebel T7). About 50 to 65 pictures from every lettuce leaf from random angles were captured during the daytime under daylight conditions. All the collected images were saved in .jpeg format. The resolution of the collected images was within the range of 1200 × 1600 and 2592 × 1944 pixels. During image collection, the camera was kept within 0.5 to 1.0 m from the lettuces. The collected image data were sorted and stored as shown in Table 1. About 60, 20, and 20% of the collected data were used for training, testing, and validation purposes, respectively.

Table 1. Multiclass accuracy comparison of models.

| Class          | Training Sample | Validation Sample | Test Sample | VGG16 Accuracy, % | VGG19 Accuracy, % | CNN Accuracy, % |
|----------------|-----------------|-------------------|-------------|-------------------|-------------------|-----------------|
| Black Seed 0   | 28              | 8                 | 8           | 100               | 100               | 100             |
| Black Seed 50  | 28              | 8                 | 8           | 100               | 100               | 0               |
| Black Seed 200 | 31              | 9                 | 9           | 100               | 100               | 22.2            |
| Black Seed 300 | 30              | 9                 | 8           | 100               | 100               | 100             |
| Flandria 0     | 28              | 8                 | 8           | 100               | 100               | 87.5            |
| Flandria 50    | 29              | 9                 | 8           | 100               | 100               | 0               |
| Flandria 200   | 35              | 10                | 9           | 100               | 100               | 0               |
| Flandria 300   | 34              | 10                | 9           | 88.9              | 88.9              | 77.8            |
| Rex 0          | 31              | 9                 | 9           | 100               | 100               | 77.8            |
| Rex 50         | 29              | 9                 | 7           | 100               | 100               | 28.6            |
| Rex 200        | 31              | 9                 | 8           | 87.5              | 87.5              | 12.5            |
| Rex 300        | 35              | 10                | 9           | 100               | 88.9              | 100             |
| Tacitus 0      | 29              | 9                 | 8           | 100               | 100               | 87.5            |
| Tacitus 50     | 29              | 9                 | 8           | 100               | 100               | 87.5            |
| Tacitus 200    | 40              | 12                | 10          | 90                | 100               | 90              |
| Tacitus 300    | 31              | 9                 | 8           | 100               | 100               | 100             |
2.2. Modeling

The input data in this study were images of various lettuce cultivars with different N levels. CNN models were built to classify all the images according to the 16 target variables associated with different lettuce cultivars and N levels. Training an efficient CNN model might require a lot of input images. The training images per target variable were not sufficient. Hence, the transfer learning approach was attempted. In the present work, the VGG-16 convolution neural network (CNN) model was adopted for RGB image processing and recognition based on a deep learning technique. A pre-trained version of the network trained on more than a million ImageNet databases was used to find a fit VGG-16 model. The input images were rescaled to 224 by 224 over three dimensions, such as RGB. VGG is a CNN model which was first proposed elsewhere in the literature [24]. VGG16 model architecture with a kernel size of 3 × 3 was used to analyze the images up to the granular level was developed. The kernel size 3 × 3 of VGG16 was found to have the best pixel depth and it helped to build a good classifier [25]. In addition to the VGG19 model architecture, VGG 16 was employed to compare the performance of species detection. The DL model architectures used in this study contained 16 layers in-depth for the VGG16 pre-trained model, including the input, hidden, and output layers (Figure 1). All the computations had several layers of neurons in their network structures, and each neuron received input data. The input and output vectors in the system represented the inputs and the outputs of the VGG16 model.

Figure 1. The architecture of pre-trained VGG16.

Some significant key performance indicators (KPIs) were primarily calculated using a confusion matrix and its parameters to evaluate its performance in classification problems [26]. If the target labels were predicted correctly, then the actual class label was “Yes” and the value of the predicted class was also “Yes,” and they were denoted as true positive (TP). Similarly, the labels which were predicted negative were called true negative (TN). If the calculated label was “No” and the actual label was “Yes,” then it was defined as false negative (FN). A false-positive (FP) was recorded if the actual class was “No” where the predicted class was “Yes.” Performance measurements such as accuracy, precision, F1 score, and recall were calculated using these parameters (TP, TN, FN, and FP), according
to well-defined Equations (1) through (4). All these measurements denoted the classifiers’ dependence in predicting unlabeled data [26,27].

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{2}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{3}
\]

\[
\text{F1 Score} = 2 \times \frac{(\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})} \tag{4}
\]

For DL problems, accuracy is the most widely used performance measurement to assess a model. Pretrained neural network models, such as VGG16, consist of multiple layers and different activation functions in between those layers. The employed VGG16 model architecture uses Rectifier Linear Unit (ReLU), as described in Equation (5), and incorporates multiple convolutional and fully connected layers [27]. A softmax function, a modified form of a sigmoid function expressed in Equation (6), was used to calculate the probability of the distribution of the events over different events and to add to the last stage of the VGG16 before the loss function was calculated. Moreover, a categorical cross-entropy equation (Equation (7)), a loss function that is well recognized in many multiclass classification problems, was employed. This formulation was used to distinguish two different discrete probability distributions from each other, as recommended in the literature [28].

\[
R(z) = z, \{z, z \geq 0\} \text{ or } R(z) = 0, \{0, z < 0\} \tag{5}
\]

\[
\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{6}
\]

\[
\text{Loss} = \sum_{i=1}^{\text{output size}} z_i \ast \log \log \hat{z}_i + (1 - z_i) \ast \log(1 - \hat{z}_i) \tag{7}
\]

where \(\hat{z}_i\) is the \(i\)th scalar value expressed as the model output, \(z_i\) is the corresponding actual target value, and output size is the number or scalar value in the model output.

2.3. Data Augmentation Implementation

Data augmentation (DA) plays a vital role in increasing the number of training images, which aids in improving the classification performance of deep learning techniques for computer vision problems [29]. Training image classification models often fail to produce robust classifiers due to the insufficient availability of training data. To alleviate the relative scarcity of data compared to the free parameters of a classifier, DA was found to be an appropriate solution [30]. An image DA includes a rotation in various angles, zoom in and out, cropping the image, shearing the image to different angles, flipping, changing brightness and contrast, adding and removing noise, scaling, and many segmentation and transformation techniques [29]. DA is not only used to increase the size of the dataset and find patterns that are otherwise obscured in the original dataset, but also used to reduce extensive overfitting in the model [31]. Different DA techniques are available in Tensor-flow that can be performed using the TFLearn DA method [32]. DA was proven to be effective in various agricultural experiments like plant leaf disease image detection, crop yield prediction, and pest control.

The present study employed Keras, an inbuilt augmentation technique proposed by Sokolova and Lapalme [33]. Due to size and processing power limitations, a randomly selected batch size of 16 images from the training dataset was used. Rescaling of both training and testing datasets was the first step applied. Most input images were already aligned sufficiently well, and therefore, image correction rotations of relatively small angles of 0 to 5 degrees was performed. A crop probability was set at 0.5 to remove different parts of images in order to classify a wide variety of test inputs successfully. Horizontal flip, vertical flip, width shift range, and height shift range were used to detect different positions and sizes with the same input image. The zoom-in and out parameter was set at
0.2 since the input images already had different levels of elevation capture. A shear range was set at 0.2 and rotation occurred in the counterclockwise direction.

The results obtained were linearly mapped to change the geometry of the image based on the camera position relative to the original image [34]. Linear mapping transformations were used to correct the dimensions of the images, which allowed the detection of any possible irregularities. The quality of the images was sufficient for the research objective of detecting lettuce species types and applying different N levels based on the color compositions of the used images. A set of augmented images obtained after the transformations is shown in Figure 2. The outputs from the augmented images were fed as an input to the VGG16 and VGG19 models. Using the same parameters, the CNN model was built without changing any original labels. Previously Kuznichov and Cap used a similar approach to increase the input variables for leaf and disease detection for deep learning methods [22,35].

![Augmented output data](image)

**Figure 2.** Augmented output data (left to right): (a) original, (b) width shift, (c) height shift, (d) shear, (e) horizontal flip, (f) vertical flip, and (g) zoomed in.

### 2.4. Implementation of Algorithms

One source of the utility of CNNs is that they are configurable in such a way as to adjust image quality. A CNN with a grid search technique is highly efficient, but computationally expensive [36]. Transfer learning has reduced the heavy computational load of CNNs by reusing weights from previous, effective models. Pretrained models like VGG16 and VGG19 can produce the best results with less configuration. Many studies have been conducted for the comparison of CNNs with other transfer learning methods to find efficient methods to detect plants, leaves, disease, etc. [37–39]. In this study, to classify the lettuce breeds and their N levels, a configurable CNN was employed, along with VGG16 and VGG19, to compare their accuracy with the augmented dataset. The flowcharts of the algorithms are shown in Figure 3.
2.4.1. CNN Implementation

Different types of convolution processes were employed as shown in Figure 3, and filters were applied. Subsequently, feature maps were created to obtain the desired features from the Rectifier Linear Unit (ReLU) layer [40]. The output was used as the input of the ReLU layer, which works as an activation function to convert all the negative values to zero.
After the convolution and ReLU were performed, the pooling layer reduced the spatial volume of the output. In the present study, the architecture of the CNN, as described in the studies [41], was implemented, and the linear activation function was used to achieve a higher accuracy. The augmented dataset was used as an input to the CNN with dimensions of $224 \times 224 \times 3$. The first max-pooling layer had an input of $224 \times 224 \times 64$, and the output using the ReLU layer was $111 \times 111 \times 32$. Three max-pooling layers with DenseNet at the last end were used before the softmax. To mitigate overfitting, a 40% dropout was introduced before feeding the output of the pooling to DenseNet. Grid search was employed to find out the best probability of dropout for the dataset. The input array of the grid search was 30, 40, 50, 60, and 70%. A 40% dropout from the last max-pooling output dimension ($27 \times 27 \times 64$) proved to achieve the best classification accuracy. Then, the output was flattened. The DenseNet had an output of 16 classes. The learning rate was set to 0.1 to expedite the training process.

2.4.2. VGG16 Implementation

In the present work, VGG16 was used for classification and detection of a depth of 16 layers, as explained in Figure 3b. A pre-trained version of the network trained on more than a million ImageNet [23] databases was used to find a best fit VGG16 model. The input images were rescaled by $224 \times 224$ in size over three dimensions, such as RGB [42,43]. Using the Keras library with TensorFlow 2.0 backend, the model was developed to build a classifier to detect four different lettuce species and their four nutrient levels, resulting in 16 classes to detect lettuce breeds and their different N levels. In this study, the last three fully connected layers were followed by a softmax function that was a modified sigmoid function to predict multiclass labels. Each convolutional layer in the VGG16 had a ReLU layer. A ReLU layer was chosen over a sigmoid function to train the model at a faster pace. No normalization was applied to the layers of VGG16 as it did not significantly impact accuracy, even though it often increased the processing time.

The input images began at $224 \times 224$ in size with three layers of RGB images. These images were the output of the data augmentation process, and they then underwent convolution in two hidden layers of 64 weights. For this study, the max-pooling reduced the sample size from 256 to 112 samples. This process was followed by the other two convolution layers with weights increasing from 128 to 512. Five max-pooling layers followed these five convolution layers. At the last end of the model, the total number of parameters obtained was 14,714,688. No normalization was applied, and all the parameters were used to train the model to detect lettuce N levels efficiently.

2.4.3. VGG19 Implementation

The depth of the VGG models varied from 16 to 19. VGG19 had a depth of 19 layers, as explained in Figure 3c, as compared to 16 layers for VGG16. VGG19 added extra three convolutional layers of 512 channels of the $3 \times 3$ kernel but used the same padding as previous layers. Then, one more max-pool layer was added to the structure. Three extra Conv2D layers were placed before the last three max-pool layers. The input stride to the output stride was the same as in VGG16. The last max-pool layer had dimensions of $7 \times 7 \times 512$ which was then flattened and fed to a Dense layer. No normalization was applied in any layer. ReLU was used for a fast-paced training process. A sigmoid function followed the last three fully connected layers as in VGG16. The literature shows that breed classification needs to transfer learning of the deep convolutional neural network comparison for the correct model to be selected [44–46]. We included two VGG models in our experiment.

2.5. Optimization and Validation

The results generated from VGG16 were fitted to a separate convolution layer obtained from Conv-1d, in Keras. Initially, the batch size was set to 64 to adjust the computing power. For multiclass classification problems, the literature suggested [47] to use a large batch size
and a standard process to set the steps per epoch as the number of classes divided by the number of batch sizes. However, the 64-batch size was the first fed to the augmentation technique, and it was then fitted into the VGG16 model, and the whole process was run on the fly. The number of steps per epoch was increased to process more data in every cycle. The steps per epoch were initially set to 32, which increased the training time but helped to decrease loss. A separated test data set was available to use besides the validation data. Five validation steps per epoch were taken, which affected validation data and made the classifier more robust.

The softmax activation function was applied at the final layer because it converted the score into probabilities considering other scores. A multiclass was the subject for prediction, and thus, Categorical Cross-Entropy with the softmax function, called a softmax loss, was used for loss measurement. After defining the loss, the gradient of the Categorical Cross-Entropy was computed with respect to the outputs of the neurons of the VGG16 model to back-propagate it through the net and optimize the defined loss function in order to tune the network parameter. The adaptive moment estimation (Adam) optimizer was used to update the network weight training and reduce overfitting [42,48].

3. Results and Discussion

3.1. Results Interpretation

The DA technique with the VGG16 model achieved a very high accuracy of 97.9% over 134 test images (Table 1 and Figure 4) despite a low number of input images. On the other hand, 498 images achieved ~99.59% accuracy with 147 validation images during the training process. The model reached 98.19% of accuracy on its third epoch. The training process was performed for 15 epochs, and the Adam optimizer efficiently optimized the loss factor from the first epoch. Due to 32 incremental steps per epoch, the training process helped the optimizer reach global minima with fewer epochs. Based on the decrement of categorical cross-entropy, the predicted probability from the softmax function was aligned with an actual class label. Figure 5b shows the loss and accuracy history graph of the VGG16 model, which indicates an optimum loss of 0.013 during the training of epoch 9 and a loss of 0.02 during validation, after epoch 15. Although accuracy is the most intuitive performance measure to observe the prediction ratio, the precision of the VGG16 pre-trained model was measured [49]. To evaluate the robustness of the model to predict unknown samples, the precision of every model associated with this experiment was calculated. Figure 4 shows an accuracy of 100% using the VGG16 model.

Figure 4. Accuracy matrices of VGG16 on the test dataset.
High precision indicates a low false-positive rate. Figure 4 shows the Recall (sensitivity) to be higher than the standard value of 0.5. The F1 score displayed in Figure 4 also suggests that the model performance was above 90% when using test data, which is a good indication
for reproducing consistent output with unknown data samples. Some false-positive results were found with two lettuce species: Rex treated with 50 ppm of N (Rex 50), and Black Seed treated with 200 ppm of N (Black Seed 200). The overall prediction accuracy of the model was 97.9%. These experiments were conducted using a local machine (HP OMEN 15t Laptop) with 32 GB of RAM, a Core i9-9880H processor, and a GeForce RTX-2080 GPU consisting of 2944 CUDA cores. The results from 15 epochs were documented, where the steps per epoch was 32, and the batch size was 64. As a result, the training period took an average of 85 s per epoch. Figure 5 demonstrates that both training and validation accuracy were stable and consistently over 92% after three epochs. The training and validation loss were also consistently lower during the training process. This result demonstrates that the model is efficient in detecting lettuce types and N levels in unknown data samples.

3.2. Model Performance Comparison

To evaluate the performance of the existing models for object classification, the VGG16 model proposed in this work (Figure 3b) was compared with the VGG19 and CNN models (Figure 3a,c). The VGG19 model was pre-trained with extra three Conv2D layers and accepted an input size of 224 × 224. Figure 3a,b shows that each parameter of the VGG16 and VGG19 models was tuned in the same way. The resulting accuracy of VGG19 with the data augmentation technique was 97.89% (Figure 6a), which was less accurate (about 1% less) than VGG16 (Figure 6b) for the tested dataset. The CNN model was constructed with three Conv2D models followed by three dense layers, and it accepted the input size of 224 × 224, as shown in Figure 3c. Figure 3b shows the loss and accuracy history graph of the VGG16 model, which indicates optimum losses of 0.013 and 0.02 during the training process while on epoch 9 and after epoch 15, respectively, during the validation cycle. Figure 6a,b shows that the VGG19 model demonstrated similar loss and accuracy as VGG16, but the CNN model with data augmentation failed to produce an efficient classifier. The highest precision obtained using the CNN model after 15 epochs of evolutions was 80.59%, with an average of 62.19% accuracy per test dataset (Figure 5c). When the number epochs were increased to 50, the highest accuracy the CNN achieved for validation was 97.59%, with an average of 64.17% per test dataset. The overall accuracy of the CNN was essentially sufficient; however, it failed, as shown in Figure 6c, to classify three classes, namely Black Seed with 50 ppm of nutrients and Flandria with 50 and 100 ppm of nutrients, in the tested datasets (Table 1) indicating that the classifier could not differentiate between all of the classes with smaller sample sizes [33,50]. A detailed comparison of all model performances was generated with 15 epochs of evolution (Figures 5 and 6). Figure 6a,b demonstrates that the accuracy of VGG16 and VGG19 models in determining the classifiers of all the studied lettuce species and the applied nutrients (N levels) was above 87%. The accuracy of the CNN model was significantly lower than the other two models. Figure 6c identifies the classes of the studied lettuce cultivars based on their applied nutrient levels, e.g., Black Seed with 50 and 200 ppm of N, Flandria with 0, 50, 200, and 300 ppm of nutrients, Rex with 0, 50, 200, and Tacitus with 0, 50, 200 ppm of nutrients. Primary investigation of the CNN by adding different convolution and pooling layers proved that the lack of sufficient training images over multiple target variables creates a weak learning rate. Figure 5c exhibits that the CNN model did not converge properly. This indicates that there were not enough data in the literature to train it [51]. In addition, this graph shows an underfitting issue. The highest outputs from the several configurations studied, as shown in Figure 3c, were tabulated in Table 1. In this study, we attempted to follow a well-established model comparison to detect lettuce breeds and their nutrient levels using deep learning methods, which has already been proven effective for various agricultural image classifications [37,52,53]. The primary observation of this study indicates that our model achieved better results than those studies due to the usage of DA, which helped us overcome the insufficient number of training images of lettuce.
The present research has great potential to integrate the proposed models into agricultural robotic systems for precision management. This study used RGB images, and the image processing techniques associated with deep learning models performed in real-time. Therefore, the results of this study would fit well with the real-time detection system requirements in the field. The present study not only monitored different lettuce cultivars but also classified their different nitrogen levels, which could have great potential for the disease/growth condition monitoring tool. Our experiment establishes that deep learning models are efficient to detect different lettuce breeds and nitrogen levels with a smaller amount of input data. The evaluation metrics show evidence of the reusability of these predictive models for further application. These experimental models are the primary building block for the development of image detection applications to identify different object types. At this point, it can be concluded that the experiment’s outcomes in this study could be applicable to various problem sets such as vegetable leaf classification, disease identification of plants, and growth measurements of vegetables. Researchers have successfully developed numerous applications using deep learning techniques to diagnose plant disease using smartphones [23, 54–56]. The recent research interest [13, 38, 39, 57–59] has increased to establish better classifiers by addressing problem statements like the detection of plant stress levels, medicinal plant detection, and overall breed identification. Overall, the current study will impact this domain significantly and could eventually be applied to a better predictive model for use in smartphone applications.

3.3. Accuracy Evaluation Metrics

The ratios, recall, and precision of the F1 scores shown in Figure 4, summarize the trade-off between the false-positive rate and the true-positive predictive value for our VGG16 model using different propensity thresholds. The F1 score considers the number of false positives and false negatives, while precision represents the true positive predictive value of our model. The precision ratio denotes the performance of our model at predicting the positive class which is mentioned in Equation (2). The recall is described as the ratio of the number of true positives divided by the sum of the true positives and the false negatives, which is denoted in Equation (3). Recall is a measure of how many true-positives are identified correctly, and, as shown in Figure 4, most precision vs. recall values tilt towards 100 percent or their ratio is 1, which means that our VGG16 model achieves high accuracy and minimizes the number of false negatives.

Table 1 shows the classification accuracy and prediction time across the four lettuce breeds and four nitrogen levels of each species, summarized as 16 classes. The VGG16 model achieved an overall average classification accuracy of 97.9%. This is evidence that the predictive model can classify any of the trained lettuce breeds and their nitrogen levels near perfectly. From the Table 1 data, we can see that 13 out of 16 classes have 100% accuracy, using VGG16 model. This evidence establishes that our model is robust and can operate in real-time inference in the agricultural field efficiently.

3.4. Model Limitations and Strength

The primary goal of the study was to increase the detection accuracy of the classifier rather than fine-tuning it for a particular dataset, which would increase the reproducibility
of the developed models. The CNN model does not converge to its loss graph due to several reasons. The primary reason is the ratio of class to the distribution of training data. To reduce this issue, we introduced data augmentation. Since the data augmentation was not helpful in this situation, we changed the weight distribution to random. Figures 5c and 6c show the results of using a cross-entropy loss function, which yielded the best results from the testing images. Table 1 shows that some of the classes had 0 (zero) prediction accuracy, indicating bias and an insufficient number of training samples. There is a possibility to increase the amount of data with more data augmentation techniques and filters, but we tried to skip those experiments because the wrong augmentation technique could have led the model to less predictive accuracy [36].

The present research has great potential to integrate the proposed models into agricultural robotic systems for precision management of lettuce production. This study used RGB images, and the image processing techniques associated with deep learning models performed in real-time. Therefore, the results of this study would fit well with the real-time detection system requirements in the field. The present study not only monitored different lettuce cultivars but also classified their different nitrogen levels, which could have great potential for the disease/growth condition monitoring tool. Our experiment establishes that deep learning models are efficient to detect different lettuce breeds and nitrogen levels with a smaller amount of input data. The evaluation metrics show evidence of the reusability of these predictive models for further application. These experimental models are the primary building block for the development of image detection applications to identify different object types. At this point, it can be concluded that the experiment’s outcomes in this study could be applicable to various problem sets such as vegetable leaf classification, disease identification of plants, and growth measurements of vegetables. Researchers have successfully developed numerous applications using deep learning techniques to diagnose plant disease using smartphones [23, 34–36]. The recent research interest [13, 38, 39, 57–59] has increased establish better classifiers by addressing problem statements like the detection of plant stress levels, medicinal plant detection, and overall breed identification. Overall, the current study will impact this domain significantly and could eventually be applied to a better predictive model for use in smartphone applications.

4. Conclusions

In this study, lettuce breed images of four different types with four nutrient levels were taken to investigate the growth performance and nutrient concentrations in the leaves of the lettuces using image data using ML algorithms. The proposed deep learning model, VGG16, was found to be highly accurate in classifying the species of the four different lettuce cultivars studied (Black Seed, Rex, Flandria, Tacitus), not only by species type but also by the amount of the applied nutrient levels (0, 50, 200, and 300 ppm of N). The accuracy of the VGG16 and VGG19 models in identifying the nutrient levels of four studied lettuce cultivars based on RGB images mainly was 88 to 100%. The VGG16 and VGG19 models significantly outperformed CNN models, which performed poorly in identifying nutrient levels of Blackseed, Flandria, and Rex. The study results revealed that computer vision combined with deep learning and robotic systems has a great potential for lettuce growth and nutrient level monitoring in real-time with high accuracy and speed.

Author Contributions: Conceptualization, S.E., C.W.L., and H.S.; methodology, M.A., H.S., S.E., and E.K.; software, M.A., S.E., and E.K.; validation, S.E., and B.C.; investigation, M.A., S.E., and H.S.; resources, H.S., C.W.L.; data curation, S.E., C.W.L., H.S.; writing—original draft preparation, M.A., S.E., B.C., and E.K.; writing—review and editing, B.C. and H.S.; supervision, S.E., B.C. and H.S.; funding acquisition, B.C. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

1. Prasad, S.; Singh, P.P. Medicinal plant leaf information extraction using deep features. In Proceedings of the TENCON 2017—2017 IEEE Region 10 Conference, Penang, Malaysia, 5–8 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2722–2726.
2. Hartz, T.K.; Johnstone, P.; Williams, E.; Smith, R. Establishing lettuce leaf nutrient optimum ranges through DRIS analysis. HortScience 2007, 42, 143–146. [CrossRef]
3. Tang, Y.T.F. Learn: TensorFlow’s high-level module for distributed machine learning. arXiv 2016, arXiv:1612.04251.
4. Mikołajczyk, A.; Grochowski, M. Data augmentation for improving deep learning in image classification problem. In Proceedings of the 2018 International Interdisciplinary PhD Workshop (IIPhDW), S winouście, Poland, 9–12 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 117–122.
5. Shijie, J.; Ping, W.; Pei, J.; Siping, H. Research on data augmentation for image classification based on convolution neural networks. In Proceedings of the 2017 Chinese automation congress (CAC), Jinan, China, 20–22 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 4165–4170.
6. Pawara, P.; Okafor, E.; Schomaker, L.; Wiering, M. Data augmentation for plant classification. In International Conference on Advanced Concepts for Intelligent Vision Systems; Springer: Berlin/Heidelberg, Germany, 2017; pp. 615–626.
7. Pink, D.; KEANE, E.M. Lettuce: Lactuca sativa L. In Genetic Improvement of Vegetable Crops; Elsevier: Amsterdam, The Netherlands, 1993; pp. 543–571.
8. Eshkabilov, S.; Lee, A.; Sun, X.; Lee, C.W.; Simsek, H. Hyperspectral imaging techniques for rapid detection of nutrient content of hydroponically grown lettuce cultivars. Comput. Electron. Agric. 2021, 181, 105968. [CrossRef]
9. Minervini, M.; Giuffrida, M.V.; Perata, P.; Tsaftaris, S.A. Phenotiki: An open software and hardware platform for affordable and easy image-based phenotyping of rosette-shaped plants. Plant J. 2017, 90, 204–216. [CrossRef]
10. Kaya, A.; Keceli, A.S.; Catal, C.; Yalic, H.Y.; Temucin, H.; Tekinerdogan, B. Analysis of transfer learning for deep neural network based plant classification models. Comput. Electron. Agric. 2019, 158, 20–29. [CrossRef]
11. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. Proc. ICML 2013, 30, 3.
12. Fawzi, A.; Samulowitz, H.; Terzis, P.; Frossard, P. Adaptive data augmentation for image classification. In Proceedings of the 2016 IEEE International Conference On Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3688–3692.
13. Agostinelli, F.; Hoffmann, M.; Sadowski, P.; Baldi, P. Learning activation functions to improve deep neural networks. arXiv 2014, arXiv:1412.6830.
14. Ososkov, G.; Goncharov, P. Shallow and deep learning for image classification. Opt. Mem. Neural Netw. 2017, 26, 221–248. [CrossRef]
15. Ahsan, M.; Nygard, K. Convolutional Neural Networks with LSTM for Intrusion Detection. CATA 2020, 69, 69–79.
16. Koutsoukas, A.; Monaghan, K.J.; Li, X.; Huan, J. Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J. Cheminform. 2017, 9, 1–13. [CrossRef] [PubMed]
17. Verma, S.; Chu, A.; Singh, A.P.; Sharma, P.; Rajvanshi, P. Deep learning-based mobile application for plant disease diagnosis: A proof of concept with a case study on tomato plant. In Applications of Image Processing and Soft Computing Systems in Agriculture; IGI Global: Hershey, PA, USA, 2019; pp. 242–271.
18. Koutsoukas, A.; Monaghan, K.J.; Li, X.; Huan, J. Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. J. Cheminform. 2017, 9, 1–13. [CrossRef] [PubMed]
19. Odabas, M.S.; Leelaruban, N.; Simsek, H.; Padmanabhan, G. Quantifying impact of droughts on barley yield in North Dakota, USA using multiple linear regression and artificial neural network. Neural Netw. World 2014, 24, 343. [CrossRef]
20. Alippi, C.; Disabato, S.; Roveri, M. Moving convolutional neural networks to embedded systems: The alexnet and VGG-16 case. Proc. ICML 2011, 30, 3.
21. Arslan, H.; Taşan, M.; Yıldırım, D.; Köksal, E.S.; Cemek, B. Predicting field capacity, wilting point, and the other physical properties of soils using hyperspectral reflectance spectroscopy: Two different statistical approaches. Environ. Monit. Assess. 2014, 186, 5077–5088. [CrossRef]
22. Francis, M.; Deisy, C. Disease detection and classification in agricultural plants using convolutional neural networks—A visual understanding. In Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 7–8 March 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1063–1068.
23. Ahsan, M.; Gomes, R.; Chowdhury, M.; Nygard, K.E. Enhancing Machine Learning Prediction in Cybersecurity Using Dynamic Feature Selector. J. Cybersecur. Priv. 2021, 7, 199–218. [CrossRef]
24. Gao, Z.; Luo, Z.; Zhang, W.; Lv, Z.; Xu, Y. Deep learning application in plant stress imaging: A review. AgriEngineering 2020, 2, 430–446. [CrossRef]
25. Rangarajan, A.K.; Purushothaman, R.; Ramesh, A. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput. Sci. 2018, 133, 1040–1047. [CrossRef]
28. Zhang, Y.; Gao, J.; Zhou, H. Breeds classification with deep convolutional neural network. In Proceedings of the 2020 12th International Conference on Machine Learning and Computing, Shenzhen, China, 15–17 February 2020; pp. 145–151.

29. Samiet, S.; Rasti, P.; Yu, J.L.; Buitink, J.; Rousseau, D. Deep learning-based detection of seedling development. *Plant Methods* **2020**, *16*, 1–11. [CrossRef] [PubMed]

30. Arya, S.; Singh, R. A Comparative Study of CNN and AlexNet for Detection of Disease in Potato and Mango leaf. In Proceedings of the 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, 27–28 September 2019; IEEE: Piscataway, NJ, USA, 2019; Volume 1, pp. 1–6.

31. GC, S.; Saidul Md, B.; Zhang, Y.; Reed, D.; Ahsan, M.; Berg, E.P.; SUN, X. Using Deep Learning Neural Network in Artificial Intelligence Technology to Classify Beef Cuts. *Front. Sens.* **2020**, *2*, 5. [CrossRef]

32. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Processing Manag.* **2009**, *45*, 427–437. [CrossRef]

33. Ahmed, A.; Reddy, G. A mobile-based system for detecting plant leaf diseases using deep learning. *AgriEngineering* **2021**, *3*, 478–493. [CrossRef]

34. Hoagland, D.R.; Arnon, D.I. Growing plants without soil by the water-culture method. In *Growing Plants without Soil by the Water-Culture Method*; University of California: Berkeley, CA, USA, 1938.

35. Joshi, A.J.; Portik, F.; Papankolopoulos, N. Multiclass batch-mode active learning for image classification. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1873–1878.

36. Kuznichov, D.; Zvirin, A.; Honen, Y.; Kimmel, R. Data augmentation for leaf segmentation and counting tasks in rosette plants. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.

37. Asadi, K.; Littman, M.L. An alternative softmax operator for reinforcement learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 243–252.

38. Oppenheim, D.; Shani, G. Potato disease classification using convolution neural networks. *Adv. Anim. Biosci.* **2017**, *8*, 244–249. [CrossRef]

39. Fu, Y.; Huang, X.; Li, Y. Horse Breed Classification Based on Transfer Learning. In Proceedings of the 2020 4th International Conference on Advances in Image Processing, Chengdu, China, 13–15 November 2020; pp. 42–47.

40. Denton, A.M.; Mostofa, A.; David, F.; John, N. Multi-scale analysis of geospatial agricultural data for sustainability. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2139–2146.

41. Cap, Q.H.; Uga, H.; Kagiwada, S.; Iyatomi, H. Leafgan: An effective data augmentation method for practical plant disease diagnosis. *IEEE Trans. Autom. Sci. Eng.* **2020**, *1*, 10– [CrossRef]

42. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* 2014, arXiv:1412.6980.

43. Agrawal, D.; Minocha, S.; Namasudra, S.; Kumar, S. Ensemble Algorithm using Transfer Learning for Sheep Breed Classification. In Proceedings of the 2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACl), Timisoara, Romania, 19–21 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 199–204.

44. Cometti, N.N.; Martins, M.Q.; Bremenkamp, C.A.; Nunes, J.A. Nitrate concentration in lettuce leaves depending on photosynthetic photon flux and nitrate concentration in the nutrient solution. *Hortic. Bras.* **2011**, *29*, 548–553. [CrossRef]

45. Ahsan, M.; Gomes, R.; Denton, A. Smote implementation on phishing data to enhance cybersecurity. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 531–536.

46. Gao, F.; Fu, L.; Zhang, X.; Majeed, Y.; Li, R.; Karkee, M.; Zhang, Q. Multiclass fruit-on-plant detection for apple in SNAP system using Faster R-CNN. *Comput. Electron. Agric.* **2020**, *176*, 105634. [CrossRef]

47. Sunoj, S.; Hammel, A.; Igathinathane, C.; Eshkabilov, S.; Simsek, H. Identification, quantification, and growth profiling of eight different microalgae species using image analysis. *Algal Res.* **2021**, *60*, 102487. [CrossRef]

48. Cemek, B.; Ünlükara, A.; Kurunç, A.; Küçüktopcu, E. Leaf area modeling of bell pepper (Capsicum annuum L.) grown under different stress conditions by soft computing approaches. *Comput. Electron. Agric.* **2020**, *174*, 105514. [CrossRef]

49. Luque, A.; Carrasco, A.; Martin, A.; de las Heras, A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognit.* **2019**, *91*, 216–231. [CrossRef]

50. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors* **2017**, *17*, 2022. [CrossRef] [PubMed]

51. Güler, M.; Arslan, H.; Cemek, B.; Ergaşin, S. Long-term changes in spatial variation of soil electrical conductivity and exchangeable sodium percentage in irrigated mesic ustifluvents. *Agric. Water Manag.* **2014**, *135*, 1–8. [CrossRef]

52. Liu, C.-W.; Sung, Y.; Chen, B.-C.; Lai, H.-Y. Effects of nitrogen fertilizers on the growth and nitrate content of lettuce (Lactuca sativa L.). *Int. J. Environ. Res. Public Health* **2014**, *11*, 4427–4440. [CrossRef] [PubMed]

53. Dureja, A.; Palwa, P. Analysis of non-linear activation functions for classification tasks using convolutional neural networks. *Recent Pat. Comput. Sci.* **2019**, *12*, 156–161. [CrossRef]

54. Le, V.N.T.; Abderom, S.; Alameh, K. Performances of the lbp based algorithm over cnn models for detecting crops and weeds with similar morphologies. *Sensors* **2020**, *20*, 2193. [CrossRef] [PubMed]
55. Xu, C.; Lu, C.; Liang, X.; Gao, J.; Zheng, W.; Wang, T.; Yan, S. Multi-loss regularized deep neural network. *IEEE Trans. Circuits Syst. Video Technol.* 2015, 26, 2273–2283. [CrossRef]

56. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* 2014, arXiv:1409.1556.

57. Rangarajan, A.K.; Purushothaman, R. Disease classification in eggplant using pre-trained VGG16 and MSVM. *Sci. Rep.* 2020, 10, 2322. [CrossRef]

58. Gomes, R.; Ahsan, M.; Denton, A. Random forest classifier in SDN framework for user-based indoor localization. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 537–542.

59. Simsek, H. Mathematical modeling of wastewater-derived biodegradable dissolved organic nitrogen. *Environ. Technol.* 2016, 37, 2879–2889. [CrossRef] [PubMed]