applied sciences

Article

Feasibility of Using Computer Vision and Artificial Intelligence Techniques in Detection of Some Apple Pests and Diseases

Yousef Abbaspour-Gilandeh [1,*], Abdollah Aghabara [1], Mahdi Davari [2] and Joe Mari Maja [3,*]

[1] Department of Biosystems Engineering, College of Agriculture and Natural Resources, University of Mohaghegh Ardabili, Ardabil 56199-11367, Iran; hemn.aghabara@yahoo.com

[2] Department of Plant Protection, College of Agriculture and Natural Resources, University of Mohaghegh Ardabili, Ardabil 56199-11367, Iran; mdavari@uma.ac.ir

[3] Department of Agricultural Science, Clemson University, 240 McAdams Hall, Clemson, SC 29634, USA

* Correspondence: abbaspour@uma.ac.ir (Y.A.-G.); jmaja@clemson.edu (J.M.M.)

Abstract: There are many methods to detect plant pests and diseases, but they are primarily time-consuming and costly. Computer vision techniques can recognize the pest- and disease-damaged fruits and provide clues to identify and treat the diseases and pests in their early stages. This study aimed to identify common pests, including the apple capsid (*Plesiocoris rugicollis*)/AC, apple codling moth (*Cydia pomonella*)/ACM, Pear lace bug (*Stephanitis pyri*)/PLB, and one physiological disease-apple russeting/AR in two cultivars, Golden Delicious and Red Delicious, using the digital image processing and sparse coding method. The Sparse coding method is used to reduce the storage of the elements of images so that the matrix can be processed faster. There have been numerous studies on the identification of apple fruit diseases and pests. However, most of the previous studies focused only on diagnosing a pest or disease, not on computational volume reduction and rapid detection. This research focused on the comprehensive study on identifying pests and diseases of apple fruit using sparse coding. The sparse coding algorithm in this work was designed using Matlab software. The apple pest and disease detection were performed based on 11 characteristics: R, G, B, L, a, b, H, S, V, Sift, and Harris. The class detection accuracy using the sparse coding method was obtained for 10 classes with three views of apple for *S. pyri* of red apple as 81%, *S. pyri* of golden apple as 88%, golden apple russeting as 85%, *S. pyri* and russeting of red apple as 100%, *S. pyri* and russeting of golden apple as 80%, codling moth of red apple as 86%, codling moth of golden apple as 72%, *S. pyri* of red apple as 83%, *S. pyri* of golden apple as 90%, codling moth and *S. pyri* of red apple as 80%, and codling moth and *S. pyri* of golden apple as 67%. The total processing time for developing the dictionary was 220 s. Once the dictionary was developed, pest and disease detection took only 0.175 s. The results of this study can be useful in developing automatic devices for the early detection of common pests and diseases of apples. Although the study was focused on apple diseases, results for this work have huge potential for other crops.

Keywords: pest; apple capsid; sparse coding; apple russeting; image processing; early detection

check for updates

Citation: Abbaspour-Gilandeh, Y.; Aghabara, A.; Davari, M.; Maja, J.M. Feasibility of Using Computer Vision and Artificial Intelligence Techniques in Detection of Some Apple Pests and Diseases. *Appl. Sci.* **2022**, *12*, 906. https://doi.org/10.3390/app12020906

Academic Editor: João M. F. Rodrigues

Received: 18 December 2021
Accepted: 13 January 2022
Published: 17 January 2022

1. Introduction

The early and smart detection of plant pests and diseases has many advantages in monitoring large fields and gardens. The necessary information about crop health and the early detection of pests and diseases can increase productivity through the appropriate management strategies, such as recommending the appropriate pesticides and fungicides and the quarantine regulations.

At present, it is essential to grade the apples and remove the healthy apples from the infected or diseased apples to boost marketability and compete with the global markets. Pests and diseases cause significant damages to crops, including apples, such as apple capsid, *S. pyri*, codling moth, and russeting. Kubiak et al. [1] examined the pests in an apple tree and reported that the number of pests varies from season to season throughout the year. Maintaining effective storage and identifying the pests and risk levels caused by different species are among the most critical tasks for protecting the apple tree. About 200 pests and diseases attack and destroy the apple [2–4].The traditional methods of apple pest and disease control, such as the use of chemicals, have failed to meet the expectations of growers and have had a devastating effect on the environment. The biological structure and different quantitative and qualitative evaluations for crops have led to the development of non-destructive experiments in recent years, among which computer vision has a particular position [5].With the recent advancement of precision agriculture in spot treatment, this technology's efficacy lies in the accuracy of detection of the area, where it needs intervention. The accuracy can be either on how accurate the sensors are used or the efficient algorithm using commercial off-the-shelf (cOTS) sensors. Therefore, it is beneficial to develop a system that can provide an easy, fast, inexpensive, and accurate method for the detection of plant pests and diseases [6,7].

The automatic early detection of plant pests and diseases can help monitor large farms and gardens. In some developing countries, farmers spend considerable time detecting pests and diseases while working on other essential aspects of farming. Farmers understand that effective pest management only occurs when done at the right time. Failure to address this in a timely manner will have destructive consequences. There are numerous studies on identifying diseases and pests, but only a handful of investigations focused on storage volume reduction and rapid detection have been considered.

Bennedsen et al. [8] evaluated an experimental system for detecting the surface damages of apples based on the rotation of apples in front of the camera, and the camera took several images during the rotation. The camera included optical filters. The dark areas that were in a fixed position relative to the apples during the rotation indicated the peduncles. In contrast, other dark areas whose shape or position was changed from a frame to frame were considered as damages. Although the identification was relatively successful (90%), a false negative occurred where some healthy apples were graded as defective. They stated that the problem of identifying the peduncle has still not been adequately resolved and suggested that additional cameras were needed in conjunction with the peduncle inspection.

Steigerwald et al. [9] found that using high-power light-emitting diodes (LEDs) in the applications related to detecting apple defects showed better results than other light sources.

Throop et al. [10] designed and evaluated a system for exploring the apple surface damages based on computer vision. This system has a conveyor for crop transport and positions the apple under the camera to obtain identical fruit images. The conveyor positioned the apples so that the peduncle and adjacent areas would not come into the camera line of sight.

An expert system can be developed to perform quality control, measurement, and online defect detection using the computational intelligence algorithms, along with the image processing systems [6] in various industries, e.g., the design of aerial and ground moving object tracking systems.

Tian et al. [3] proposed an anthracnose lesion detection method based on deep learning. Plant surface lesion images collected by an optical sensor and Cycle-Consistent Adversarial Network (CycleGAN) deep learning model were used to accomplish data augmentation. The proposed model outperformed three other state-of-the-art models in detection accuracy.

Nakano [11] studied the color and classification of apples based on neural network methods. One of the neural network methods, which examined the pixels and helped to distinguish between the healthy and the infected and diseased apples, divided the pixels into six groups. The accuracy of this method was about 86%, but this method was unable to detect apple defects. However, when the model considered all the variables, the accuracy was sharply reduced.

Kavdir and Guyer [12] graded two Emperor and Golden Delicious apple varieties based on the surface qualitative conditions using the back-propagation neural network.

The values of the pixels with a gray area and the texture characteristics obtained from the initial image were considered as the network input. Two categorization methods were implemented: the categorization method with two defective and healthy groups and the categorization method with five groups, including all defective and healthy cases. The network grading accuracy in the first case ranged from 89.2 to 100%, while in the second case, the grading accuracy in the Emperor variety was 93.8–100%.

Boniecki et al. [13] used neural networks to identify the apple pests. They stated that the usual methods of identifying apple pests are based on the visual observations of the inspector. The criteria for detecting the damage and pest of an apple tree are based on the color, shape, etc. This identification method requires considerable expertise in this area and a good understanding of different species and is very time-consuming.

In recent years, numerous studies have been conducted on diagnosing pests and diseases of fruits and vegetables using advanced algorithms. These advanced algorithms have their advantages and disadvantages. Liu et al. [14] (2018) proposed a new deep convolutional neural network model for the accurate detection of apple leaf diseases. The results were satisfactory, and the proposed model can achieve a detection accuracy of 97.62%. Turkoglu et al. [15] (2019) presented a Multilayer Perceptron (MLP) CNNs model for the detection of plant diseases and pests. They first used different CNN models to extract deep features and then Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) classifiers for feature classification. In this study, LSTM classifier training with a high dimensional feature vector requires a longer processing time. Aside from the processing time, finding the optimal LSTM parameters was another limitation of this work. Liu and Xuewei [16] (2020) proposed an improved You Only Look Once (Yolo) V3 algorithm for the diagnosis of tomato diseases and insect pests. The Yolo V3 network algorithm used multidimensional feature recognition based on the image pyramid, object constraint box clustering, and multidimensional training, which drastically improved the algorithm's performance. Experimental results showed that the detection accuracy of the algorithm was 92.39% and the detection time was only 20.39 ms. Therefore, for diagnosing tomato diseases and pests, the improved Yolo V3 algorithm can identify the location and category of tomato diseases and pests accurately and quickly. An improved convolution neural network model based on VGG16 was proposed by Yang et al. [17] (2020). The VGG16 Classic Network Classifier was modified by adding a batch of normalization layers, a global average integration layer, and a fully connected layer to accelerate convergence and reduce training parameters. The proposed model used 2141 apple leaves to identify apple leaf diseases in the training set. In this study, although the training time was longer than AlexNet and ResNet, the proposed model had fewer parameters and provided higher accuracy. Pardede et al. [18] (2020) investigated the recent advances in machine learning to diagnose plant diseases. In the paper, studies are categorized based on machine learning architecture, in which shallow architectures are used in machine learning. Deep Learning is used to find good features of suitable classifiers for this purpose. With the promising performance of deep learning, it is expected to be the dominant technology in this field. Khan et al. [19] (2021) collected data from healthy and infected apple leaves from various orchards in the Kashmir Valley. They developed a deep learning model to identify and classify apple disease using transitional learning automatically. The results obtained with the proposed approach were promising and reported accuracy of about 97%.

Because image processing algorithms (operations such as feature detection, image tagging, etc.) use pixels' color or gray values, each pixel represents an element of a two-dimensional array. The image processing operation is always associated with working on matrices. In many image tagging methods, only a portion of the image is used, representing the features considered for the image clustering or classification. Other parts of the image (such as the image foreground) are not used, and the tagging operation used is prone to errors. The sparse matrix is a matrix with very high zero elements. Since the storage and use of zero elements are not cost-effective, and it is always used to reduce the storage of these elements to make the matrix operations faster, the images can be expressed by the sparse matrices [20].

Hubel and Wiesel [21] presented the idea of fast image processing using sparse coding. Sparse coding is an interesting technique for computer vision that has not been utilized for detection of pests and diseases of fruits and vegetables.

In the literature, there are several studies on the diagnosis of apple diseases. Although the latest models developed, including the deep learning models, are more accurate than traditional machine learning methods, there are still shortcomings, such as high complexity and long training times, which prevent their practical application in real environments. To this end, the sparse coding technique can save a significant amount of sampling time and sample storage space, and it is favorable and advantageous.

The aim of this study was to detect some pests and diseases of apples using digital image processing and sparse coding based on computational intelligence. In this study, to increase the processing speed, sparse coding was used, in which only a portion of the image is analyzed instead of the whole image. The advantage of this case is that, instead of using the extracted features from the sample images, the extracted features from the sub-images via the sparse method were used to train the neural network. Therefore, instead of analyzing the whole image, examining a part of the fruit, which includes damages caused by pests or diseases, minimizes processing time.

2. Materials and Methods

2.1. Preparation of Apple Samples

In the present study, the infected apples were collected from different plots of Naghadeh, West Azarbaijan province/Iran. In this study, four common pests, including the AC, ACM, PLB, and one physiological disease-AR-in two Golden Delicious and Red Delicious cultivars, were studied. The whole apples were classified into 18 groups; 16 groups of infected apples and 2 groups of healthy apples. A total of 819 photos were taken from three views of 273 apple samples. Table 1 shows the categories of healthy and infected apples.

Table 1. Classification of healthy and diseased apples.

1	Apple capsid (AC) and *Cydia pomonella* of Golden Delicious	10	*Cydia pomonella* (ACM) and apple russeting of Golden Delicious
2	Apple capsid and *Stephannts pyri* (PLB) of Red Delicious	11	*Cydia pomonella* of Red Delicious
3	Apple capsid and *Stephannts pyri* of Golden Delicious	12	*Cydia pomonella* of Golden Delicious
4	Apple capsid and apple russeting (AR) of Red Delicious	13	*Stephannts pyri* and apple russeting of Golden Delicious
5	Apple capsid and apple russeting of Golden Delicious	14	*Stephannts pyri* of Red Delicious
6	Apple capsid of Red Delicious	15	*Stephannts pyri* of Golden Delicious
7	Apple capsid of Golden Delicious	16	Apple russeting of Golden Delicious
8	*Cydia pomonella* and *Stephannts pyri* of Red Delicious	17	Healthy Red Delicious
9	*Cydia pomonella* and *Stephannts pyri* of Golden Delicious	18	Healthy Golden Delicious

The samples were collected in September 2019. After collecting the apples, each sample was placed in a plastic box according to the type of pest and disease. Figure 1 shows the images of healthy and infected apples with pests and diseases.

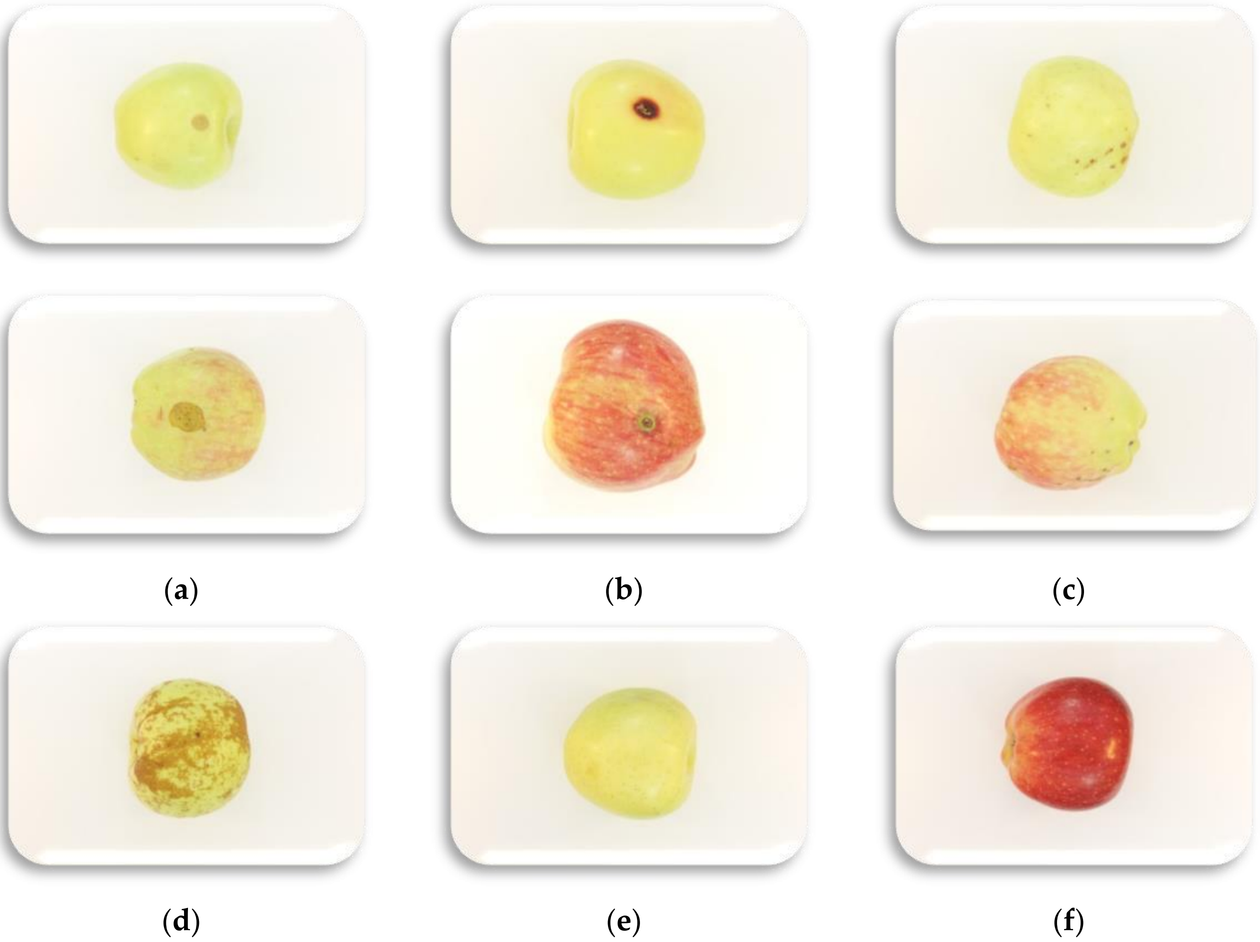

Figure 1. Images of healthy and infected apples with pests and diseases. (**a**) Apple capsid, (**b**) *Cydia pomonella*, (**c**) *Stephannts pyri*, (**d**) Apple rusting, (**e**) Healthy Golden Delicious, and (**f**) Healthy Red Delicious.

2.2. Preparation of Apple Images

The photos were taken with a dome-shaped white illumination and imaging chamber. The chamber was designed to have no external light sources. Inside the chamber, there were four rows of LED lights and four fluorescent lamps with the same spacing. The light sources had a 45° angle to the object. A constant voltage of 12 volts fed the illumination system. At the top of the chamber was a cavity that contained the camera lens, and the surrounding was completely enclosed so that no light source other than the light within the chamber could penetrate. The camera used in this study was a SONY (α200, Sony Corporation, Tokyo, Japan) CCD camera with a resolution of 10.1 MP and a 40 mm lens with a shutter speed of 5.6 f. The background on which the apples were positioned was a white Steinbach paper. The camera's lens distance to the sample surface was set to 29 cm. All images were taken in fully stabilized conditions, and the auto-adjustment modes were disabled. The software used for the image processing and neural network classification was the MATLAB software package (R2013a, MathWorks, Natick, MA, USA). All the algorithms were programmed and implemented by an Acer computer with the 64-bit quad-core Intel CPU and processing speed of 2.30 GHz.

The dimension of apple images used to create the dictionary and training stage was 3872 × 2592 pixels. The images formats used were Portable Network Graphics (PNG) and Joint Photographic Group (JPG).

2.3. Image Pre-Processing

For the image pre-processing, the extension of all database images was converted from JPG to PNG; therefore, only uncompressed images were processed. On the other hand, the PNG format allowed a transparent background around an irregular-shaped object and avoided a white (or other colored) box outlining the image. Several color filters and a quasi-Wiener filter were then applied to all images as preparation for the training operation.

2.4. Proposed Algorithm

The description of this algorithm from the first to the last, including the main image processing step and the method of using the software, is presented in Figure 2. M-files of sparse approximation, apple classifier, displaying network, and updating network in MATLAB have been attached as Supplementary Materials to the manuscript.

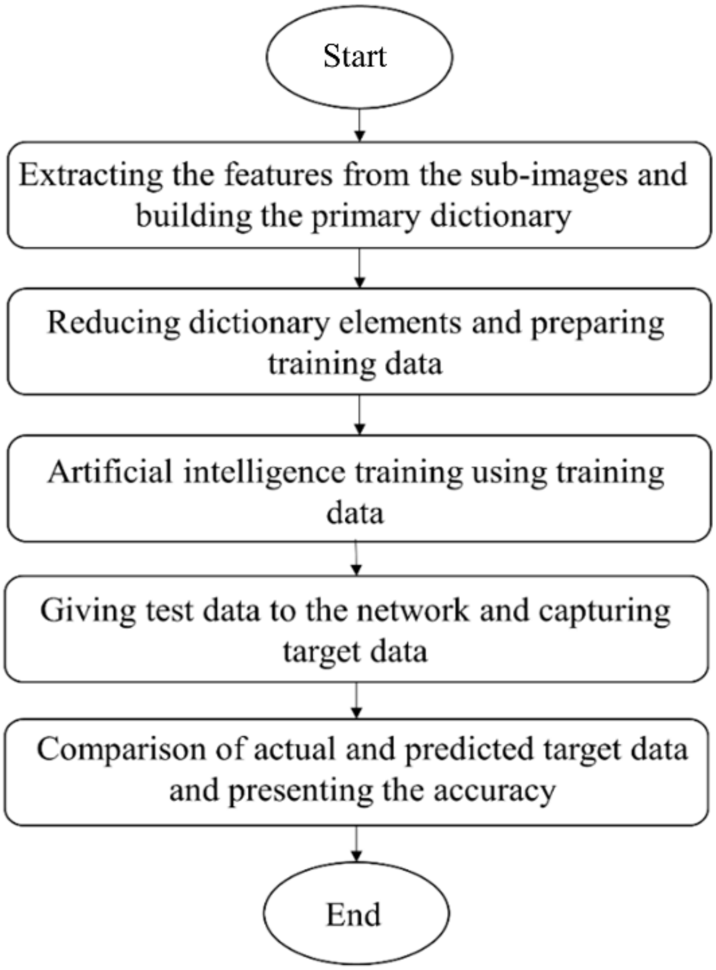
Start
Extracting the features from the sub-images and building the primary dictionary
Reducing dictionary elements and preparing training data
Artificial intelligence training using training data
Giving test data to the network and capturing target data
Comparison of actual and predicted target data and presenting the accuracy
End

Figure 2. Proposed algorithm.

2.5. Main Image Processing

In the proposed method, instead of working with the whole image, each image is subdivided into sub-images of 64 × 64 pixels. Each of the color and morphological features (Table 2), including R, G, B, L, a, b, H, S, V, SIFT, and Harris, is extracted for the sub-images. Since they may not be a square image or do not have a dimension multiple of 64, the squares are randomly overlaid within the selected image. Table 2 shows the 11 components extracted from the apples. The number on the right indicates the number of components.

Table 2. Features Extracting.

RGB	R: red, G: green, and B: blue (3)
Lab	L: lightness, a: Amount of red or green, and b: Amount of yellow or blue (3)
HSV	H: hue, S: saturation, and V: value (3)
SIFT	The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images. (1)
Harris	Corner detection operator (1)

The input test image is subdivided into a series of sub-images, and the features of the sub-images are searched in the dictionary to find the sub-images similar to the test image in the input data (Figure 3). It can then be concluded that the test image tag is the same as the tag selected in the dictionary or the parameters that the sub-images of that image want to tag are close and similar to the sub-images in the dictionary. In sparse coding, the set of training sub-images is called the dictionary. Since the number of training datasets (which are the sub-images here) may well exceed the 10,000,000 samples, matching the features of the test image sub-images to the features of the training sub-images can prolong the tagging operation with a computer for hours. Eliminating similar sub-images in the training of the dataset can minimize the time for the tagging operation and does not affect the processing accuracy.

Figure 3. Bases obtained when running the algorithm for iterations and choosing 64 bases Sparse with 64 × 64 pixels.

The most important step in this approach is to reduce the dictionary dimensions so that the duplicated data is removed. The proposed method uses the regularized graph [22]. The dictionary is well prepared when the error value equal to R = X-D*W is minimized in the shortest possible time. In this formula, X represents the desired image, W is the weights (or coefficients), and D is the dictionary atoms.

Figure 2 shows an example of bases computed with sparse coding in this study. The bases were obtained from the training images. In this study, we were first trying to create a dictionary that contained a set of bases that are a combination of healthy and damaged apples. The dictionary was constructed using training data or basic images. The purpose of this study was approximating test data (new image) using a small number (Sparse) of training images. We first approximated the new image with a set of training images by minimizing the squares of the reconstruction error, and then it was categorized in one of the 18 groups. The computing details of the bases were discussed by Gehring and Lemay [23].

2.6. Artificial Neural Network Architecture

The ANNs designed in this study were multilayer perceptron networks (Figure 4). The dictionary elements were considered as training data in the neural network training phase rather than the main samples being trained to the neural network after the dictionary was created by the sparse coding. The advantage of this case is that instead of the extracted features from the sample images (images of healthy and unhealthy apples) to be used in the network training, the extracted features from the sub-images used by the sparse method were used to train the neural network. These sub-images were from healthy and unhealthy apples (each of the sub-images according to Figure 3 are 64 × 64 pixels). It is possible to deal with local minimums and overlaid results if sub-images without the sparse coding were used for the neural network training. However, using the sparse coding, only the sub-images remain whose features can be used to determine whether the samples are healthy or unhealthy. The total number of data were 2592, of which about 80% was allocated for training and the remaining of 20% was used for the test. Figure 3 illustrates the structure of a multilayer neural network. It had an input layer of 11 neurons (components), an intermediate layer (hidden layer) of 10 neurons, and an output layer of 1 neuron.

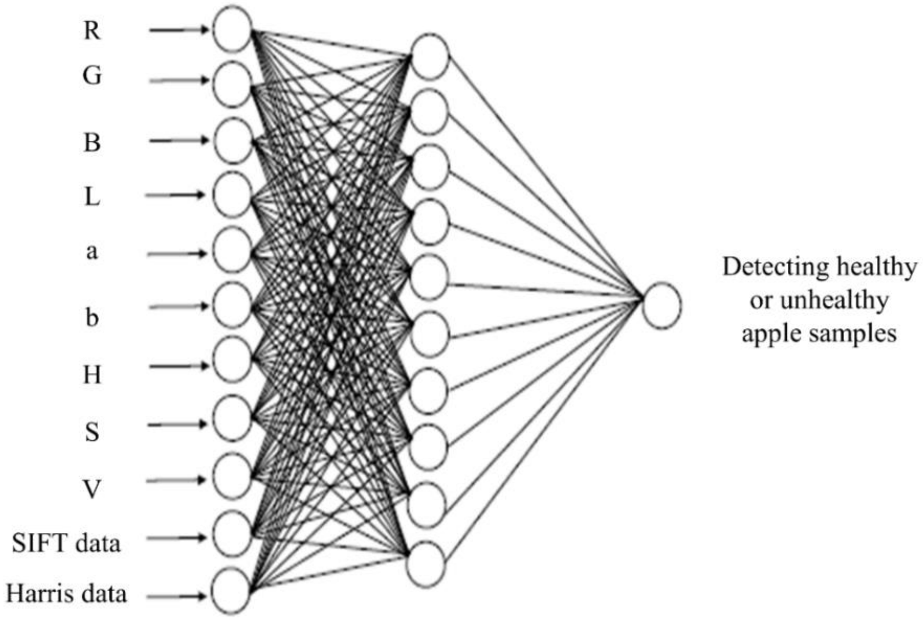

Figure 4. Architecture of used artificial neural network.

In a neural network, the purpose of learning is to optimally determine the weight coefficients of neurons so that the expected output (actual output) of the network is as close to the current output (output from the input applied to the network and obtained from the network) as possible, in other words, the error between the actual output and the network output is minimized. The aim is to train a neural network using the training data so that the order in the detection of apple pests can be identified and classified successfully.

2.7. Data Structure

Before starting the classification, the input data should be divided into two groups.

Training data: This data was used from the input data to train the network. After the randomization of input data, 80% of the data was selected as the training data. Once the network was trained, the weights final value were generated based on the least error for the training data.

Test data: After the network was trained by the training data until reaching the minimum error, the remaining 20% of the data were used as input to the network, and the network response was compared with the optimal response of tags. In this way, the efficiency of the trained network was tested.

The total number of data was 2592, where 80% of which was allocated training and the remaining 20% as validation. Table 3 provides the information on the data.

Table 3. Information of the amount of data used in the artificial neural network classifier.

Number of features	11
Number of classes	18
Number of data used for training	2073
Number of data used for validation	519

The data was first normalized before using it in the network, and the values were in the range of (0,1). After normalizing the data, the network output was encoded in the form of the classification categories. Usually, if the number of classification categories is not large, outputs are considered as the number of categories in the network, and they are coded binary.

3. Results and Discussion

3.1. Accuracy and Processing Time of Proposed Method

In order to calculate the accuracy of the proposed method, the clustering accuracy was first obtained using the perceptron artificial neural network with 10 neurons and a learning rate of 0.2 for each view of the apples. In some apples that had a particular feature and were only identified in one view, such as worminess, the accuracy of one view was very low, but in the second phase, two views were selected from each apple to be used in the training of the neural network. Finally, all three views were used to learn the pattern, which significantly improved the accuracy of this technique, relative to one view and two views.

The total processing time for building the dictionary was about 220 s. After creating the dictionary, pest and disease detection was about 0.175 s.

3.2. Clustering Accuracy Results for Each Group

The confusion matrix is an illustration of important classification results. The resulting confusion matrix with three views of Red Delicious and Golden Delicious apples are shown in Tables 4 and 5, respectively.

In this study, there were 18 groups of apples (2 groups for healthy apples and 16 groups for unhealthy apples), and the accuracy of this method was shown to be about 90%. Each time the program is run, the accuracy varies by about 1~2%, from 88 to 92%, due to the random selection of sub-images within the images. After creating the dictionary by sparse coding, it was used to train the multilayer perceptron neural network. The perceptron neural network variables can be the number of neurons in the hidden layer, the number of hidden layers, and the learning rate. The following tables show the average clustering accuracy for all classes in a percentage for the change in network parameters. It should be noted that, since this is a clustering problem, the error rate (MSE) and coefficient of determination (R^2) parameters will not be considered, and the performance evaluation parameters only include the clustering accuracy in terms of percentage and processing time.

Since the processing times for the networks with different architectures were very close and not significantly different, Tables 6 and 7 present the average clustering accuracy for all the classes.

Table 4. Confusion matrix and data classification using sparse coding with three views of Red Delicious apples.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**Number of Samples**	**Accuracy (%)**
1 Healthy Red Delicious	30	1	0	0	0	0	0	31	97
2 Apple capsid of Red Delicious	3	22	0	0	0	1	1	27	81
3 *Cydia pomonella* of Red Delicious	0	0	24	2	2	0	0	28	86
4 *Stephannts pyri* of Red Delicious	1	0	1	24	3	0	0	29	83
5 *Cydia pomonella* and *Stephannts pyri* of Red Delicious	0	0	0	1	4	0	0	5	80
6 Apple capsid and *Stephannts pyri* of Red Delicious	0	0	0	0	0	1	0	1	100
7 Apple capsid and apple russeting of Red Delicious	0	0	0	0	0	0	1	1	100

Table 5. Confusion matrix and data classification using sparse coding with three views of Golden Delicious apples.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**Number of Samples**	**Accuracy (%)**
1 Healthy Golden Delicious	29	1	0	0	0	0	0	0	0	0	0	30	97
2 Apple capsid of Golden Delicious	2	21	0	0	0	0	0	0	1	0	0	24	88
3 *Cydia pomonella* of Golden Delicious	2	0	23	0	0	2	1	0	1	0	0	29	72
4 *Stephannts pyri* of Golden Delicious	0	0	0	18	0	0	0	1	0	1	0	20	90
5 Apple russeting of Golden Delicious	1	0	0	0	23	0	1	1	0	0	1	27	85
6 *Cydia pomonella* and *Stephannts pyri* of Golden Delicious	0	0	0	0	0	2	1	0	0	0	0	3	67
7 *Cydia pomonella* and apple russeting of Golden Delicious	0	0	0	0	0	0	1	0	0	0	0	1	100
8 *Stephannts pyri* and apple russeting of Golden Delicious	0	0	0	1	0	0	0	8	0	1	0	10	80
9 Apple capsid and *Cydia pomonella* of Golden Delicious	0	0	0	0	0	0	0	0	1	0	0	1	100
10 Apple capsid and *Stephannts pyri* of Golden Delicious	0	0	0	0	0	0	0	0	0	1	0	1	100
11 Apple capsid and apple russeting of Golden Delicious	0	0	0	0	0	0	0	0	1	0	4	5	80

Tables 6 and 7 show that the neural network performance of a layer with 10 neurons in the hidden layer is more appropriate than other neural network architectures. The reason for this may be the decrease in the learning efficiency due to the weight chosen for the neurons and the order of learning in the networks with more than one hidden layer. In a single-layer neural network, decreasing or increasing the number of neurons in the layer does not directly correlate to the tagging accuracy of the test data. The following table shows the clustering accuracy for a single-layer neural network with 10 neurons and a learning rate from 0.1 to 0.9.

Table 8 shows that the best learning rate was 0.2, and for this rate, the average clustering accuracy was 91% for all the classes, yielding the best performance.

Table 6. Average clustering accuracy for all classes with a hidden layer.

Number of Neurons in the Hidden Layer	Average Clustering Accuracy for All Classes (%)
2	79
4	85
6	86
8	89
10	91
12	90
14	87
16	86
18	85
20	81
22	80
24	77
26	75
28	71
...	...
70	30

Table 7. Average clustering accuracy for all classes with two hidden layers.

Number of Neurons in the First Hidden Layer	Number of Neurons in the Second Hidden Layer	Average Clustering Accuracy for All Classes (%)
...	...	...
2	4	75
4	2	79
...	...	...
8	6	89
8	8	90
8	10	90
10	8	91
10	10	91
10	12	91
12	10	90
12	12	88
12	14	87
14	12	87
14	14	83
...	...	...
20	20	80
...	...	...
30	30	69

Table 8. Average clustering accuracy for all classes with a hidden layer, 10 neurons, and different learning rates.

Learning Rate	Average Clustering Accuracy for All Classes (%)
0.1	89
0.2	91
0.3	90
0.4	89
0.5	88
0.6	83
0.7	84
0.8	80
0.9	79

In this study, the sparse coding, which was used to reduce the size of database in the neural network training phase, had several benefits, such as increasing the learning speed of training data in the problems with very large database or increasing the performance of clustering problems in the image processing, where the feature extraction methods had a poor performance. The hidden pattern of feature relationships in the image processing problems can also be extracted using sparse coding. However, alongside these benefits and despite being often used for image processing problems, its application to other pattern recognition issues was limited. Additionally, implementing this method required the dimensions of the images in the database to be large.

4. Conclusions

In the present study, a computer vision hardware system was presented for the detection of apple pests and diseases. This system was used to classify the apple pests and diseases by means of the image processing techniques, sparse methods, artificial neural networks, and MATLAB software under the controlled conditions (exposure system, camera distance to target, camera angle, and light source). The advantage of this method over the conventional methods is the detection of some common pests and diseases, with low cost and readily available parts. The huge benefit to the farmer for this system is having an advanced tool at their disposal with results comparable to an expert consultant. With the development of the present system, it can also be used in the areas of quarantine, export, and import of agricultural products. In general, the results of this study can be summarized as follows.

The results showed that using a combination of computer vision techniques and artificial neural networks and a CCD camera enables the detection of apple pests and diseases with great accuracy without costly, destructive, and time-consuming tests.

The detection accuracy of the system using the sparse coding method was 86% for the codling moth of red apple, 72% for the codling moth of golden apple, 83% for *S. pyri* of red apple, 90% for *S. pyri* of golden apple, 80% for codling moth and *S. pyri* of red apple, 67% for codling moth and *S. pyri* of golden apple, 100% for codling moth and russeting of golden apple, 80% for *S. pyri* and russeting of golden apple, 100% for capsid and codling moth of golden apple, 100% for capsid and *S. pyri* of red apple, and 100% for capsid and *S. pyri* of golden apple with three views of apples. The higher accuracy was due to an increased number of views.

For the capside of red apple, the detection accuracy of the group using the sparse coding method was 81%, 88% for capsid of golden apple, 85% for russeting of golden apple, 100% for capsid and russeting of red apple, and 80% for capsid and russeting of golden apple. Similar to the previous result, the increased accuracy was due to the increased number of views.

The accuracy was obtained from three views of apples for 16 groups of infected apples. The detection accuracy was obtained by a sparse coding method for 2 groups of healthy apples: 97% for a healthy golden apple and 97% for a healthy red apple.

In general, this study investigated apple pests and diseases using the proposed algorithm. The algorithm accurately detected 90% of the pests and diseases, which is more accurate and faster than the earlier studies.

Although the study was focused on apple diseases, results for this work have huge potential for other crops.

In future works, we plan to collect many high-quality images of different types of apple diseases and pests and use the convolution layers of the deep learning models for feature extraction. In addition, other classifiers will be investigated.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app12020906/s1, M-files of sparse approximation, apple classifier, displaying network, and updating network in MATLAB have been attached as Supplementary Materials to the manuscript. The source code is also available at https://github.com/jmaja/ApplePestDetector.git.

Author Contributions: Conceptualization, Y.A.-G., A.A. and M.D.; methodology, Y.A.-G., M.D., A.A. and J.M.M.; software, A.A. and Y.A.-G.; validation, Y.A.-G. and J.M.M.; formal analysis, A.A., Y.A.-G. and J.M.M.; investigation, Y.A.-G., A.A., M.D. and J.M.M.; resources, Y.A.-G. and A.A.; data curation, A.A.; writing—original draft preparation, A.A. and Y.A.-G.; writing—review and editing, J.M.M.; visualization, Y.A.-G. and J.M.M.; supervision, Y.A.-G.; project administration, Y.A.-G.; funding acquisition, Y.A.-G. and J.M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by University of Mohaghegh Ardabili and is based on work supported by NIFA/USDA under project numbers SC-1700611.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Kubiak, K.; Kogut, W.; Krajewski, A.; Lindner, E.; Maciejczyk-Kempińska, U.; Mirkowska, Z.; Strojewska, I. *Arboriculture in Poland*; COBRO: Warsaw, Poland, 2000.
2. Chen, F.; Tan, T.; Chen, Z.; Chen, A. Studies on the functional response and searching efficiency of *Aphelinus mali* (Haldeman). *Southwest China J. Agric. Sci.* **2006**, *19*, 1082–1085.
3. Wang, S.; Zheng, G.; Li, X.; Xu, B.; Xing, X.; Li, Y. Investigations on main insect pests and natural enemies in ecological apple orchards in Xiuyan District, Liaoning Province. *Liaoning Agric. Sci.* **2009**, *2*, 36–38.
4. Zhou, H.; Yu, Y.; Tan, X.; Chen, A.; Feng, J. Biological control of insect pests in apple orchards in China. *Biol. Control.* **2014**, *68*, 47–56. [CrossRef]
5. Du, C.-J.; Sun, D.-W. Learning techniques used in computer vision for food quality evaluation: A review. *J. Food Eng.* **2006**, *72*, 39–55. [CrossRef]
6. Jamdar, A.V.; Patil, A.P. Detection and Classification of Apple Fruit Diseases using K-means clustering and Learning Vector Quantization Neural Network. *Int. J. Sci. Dev. Res.* **2017**, *2*, 423–429.
7. Xing, S.; Lee, M.; Lee, K.-K. Citrus Pests and Diseases Recognition Model Using Weakly Dense Connected Convolution Network. *Sensors* **2019**, *19*, 3195. [CrossRef] [PubMed]
8. Bennedsen, B.; Peterson, D. Performance of a system for apple surface defect identification in near-infrared images. *Biosyst. Eng.* **2005**, *90*, 419–431. [CrossRef]
9. Steigerwald, D.A.; Bhat, J.C.; Collins, D.; Fletcher, R.M.; Holcomb, M.O.; Ludowise, M.J.; Martin, P.S.; Rudaz, S.L. Illumination with solid state lighting technology. *IEEE J. Sel. Top. Quantum Electron.* **2002**, *8*, 310–320. [CrossRef]
10. Throop, J.; Aneshansley, D.; Anger, W.; Peterson, D. Quality evaluation of apples based on surface defects: Development of an automated inspection system. *Postharvest Biol. Technol.* **2005**, *36*, 281–290. [CrossRef]
11. Nakano, K. Application of neural networks to the color grading of apples. *Comput. Electron. Agric.* **1997**, *18*, 105–116. [CrossRef]
12. Kavdir, I.; Guyer, D. Apple sorting using artificial neural networks and spectral imaging. *Trans. ASAE* **2002**, *45*, 1995. [CrossRef]
13. Boniecki, P.; Dach, J.; Pilarski, K.; Piekarska-Boniecka, H. Artificial neural networks for modeling ammonia emissions released from sewage sludge composting. *Atmos. Environ.* **2012**, *57*, 49–54. [CrossRef]
14. Liu, B.; Zhang, Y.; He, D.; Li, Y. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry* **2018**, *10*, 11. [CrossRef]
15. Turkoglu, M.; Hanbay, D.; Sengur, A. Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *2019*, 1–11. [CrossRef]
16. Liu, J.; Xuewei, W. Tomato diseases and pests detection based on improved Yolo V3 convolutional neural network. *Front. Plant Sci.* **2020**, *11*, 898. [CrossRef]
17. Yan, Q.; Yang, B.; Wang, W.; Wang, B.; Chen, P.; Zhang, J. Apple leaf diseases recognition based on an improved convolutional neural network. *Sensors* **2020**, *20*, 3535. [CrossRef]
18. Pardede, H.F.; Suryawati, E.; Krisnandi, D.; Yuwana, R.S.; Zilvan, V. Machine learning based plant diseases detection: A Review. In Proceedings of the 2020 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Tangerang, Indonesia, 18–20 November 2020; pp. 212–217.
19. Khan, A.I.; Quadri, S.M.K.; Banday, S. Deep learning for apple diseases: Classification and identification. *Int. J. Comput. Intell. Stud.* **2021**, *10*, 1–12. [CrossRef]
20. Huang, K.; Aviyente, S. Sparse representation for signal classification. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 4 December 2006; pp. 609–616.
21. Hubel, D.H.; Wiesel, T.N. Brain mechanisms of vision. *Sci. Am.* **1979**, *241*, 150–163. [CrossRef] [PubMed]

22. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1548–1560. [PubMed]
23. Gehring, C.; Lemay, S. Sparse coding Applied to Digit Recognition. *sibi* **2012**, *1*, 1–12.