Vegetable Classification from Images Based on Deep Learning Model

Raihan Mohammad Kamaluddin
Master's in Computer Science
Texas A & M University – Corpus Christi
Corpus Christi, United States of America
rkamaluddin@islander.tamucc.edu

Rohith Reddy Mandala
Master's in Computer Science
Texas A & M University – Corpus Christi
Corpus Christi, United States of America
rmandala@islander.tamucc.edu

I. INTRODUCTION AND MOTIVATION

In 2016, the US fruit and vegetable market was worth USD 104.7 billion. The top snacking foods in the United States are currently vegetables and fruits. To attract millennials, company marketing in the United States is focused on the image of sustainable, fresh, and locally grown produce. The industry is also leaning toward advanced technology to keep produce fresh for longer periods, as well as cutting-edge manufacturing technology inputs for use in the end-use industry.

However, the majority of vegetable picking, and processing is still done by hand, requiring a significant amount of labor and low productivity, which has a negative impact on the commercialization of vegetable products. Moreover, producing barcode labels requires a lot of background work which can be quite tedious. Cashiers may also sometimes type the serial ID incorrectly or may misplace or confuse labels leading to the wrong price being charged to the customer. Image classification and identification technology can be used to automate vegetable picking and grading, which has a wide range of applications. The proposed approach has a high recognition rate in the supermarket agricultural products database.

In recent years, deep learning has been proposed as a modern machine learning algorithm, and breakthroughs have been made in image recognition, computer vision, and other fields. When compared to the conventional manual design feature approach and the deep learning method, the data features extracted by the deep learning method can better describe the rich internal knowledge of the vast data and can learn representative features automatically without manual extraction. Although deep learning modifies conventional machine learning methods, it also modifies our understanding of human cognition, resulting in a breakthrough in image recognition and other applications. The deep learning approach was applied to the field of image classification by Krizhevsky et al. [2], and the image classification accuracy rate was greatly improved. Ciresan et al. [3] strengthened the classification criteria for widely used images by using a deep neural network to achieve traffic sign classification and recognition. Tan et al. [4] developed a deep learning neural network for recognizing pathological images of vegetables. A parameter learning method for elastic momentum was proposed based on network error propagation analysis, and an identification experiment of fruit pathology images was carried out with an apple as an example, which significantly improved the accuracy of fruit and vegetable disease recognition in learning networks. Stacked AutoEncoder, Restricted Boltzmann Machine, Deep Belief Network, and Convolutional Neural Network (CNN) are the most popular deep learning networks today. In the image classification task, the CNN [5] [6] [7] is the most important among them.

The use of CNN for image recognition is becoming increasingly sophisticated. Gong et al. [8] suggested a CNN-based system for plant leaf classification. The algorithm's accuracy rate is significantly higher than that of the standard blade recognition algorithm, according to the experimental results on the Swedish leaf dataset. After a series of preprocessing steps, Hu et al. [9] proposed a dynamic gesture recognition approach for analyzing static trajectory images using CNN with the normalized image as the input of the CNN model. The experimental results show that the dynamic hand gesture recognition accuracy rate of digital 0-9 has significantly improved after testing the proposed method on a self-built database.

This research explores the technical feasibility of high-resolution classification for vegetables from a variety of categories. The goal of this project is to experiment with applying CNN techniques to this particularly challenging dataset we collected. Specific challenges include these points, first of all, the vegetable is sorted in real time. Secondly, the datasets are difficult to clean, because the quality of the images are quite different, such as light, shooting location, angle and so on. Images were obtained from the Kaggle data set, and the data set was divided into training data set, validation data set and test data set. Since each kind of vegetable has fewer images, the

expected results are not that much good, but we did everything we could with the available dataset.

The rest of the paper is organized as follows. Section 2 introduces related work in online prediction for vegetable supply chain, attributes and multi-labeling. Section 3 describes the collection of datasets, the implementation details. Section 4 experiments and the evaluation results of our proposed algorithms. Section 5 we make discussion and conclusions.

Deep learning techniques can automatically learn features from a large number of image data set. Automatic vegetable image classification is the base of many applications. This report proposed a method for vegetable images classification based on deep learning techniques. The Custom model we developed was used to train the vegetable image data set. The vegetable image data set was obtained from Kaggle and divided into training data set, validation data set and test data set. The output function of the Custom model adopted the Rectified Linear Units (ReLU) instead of the traditional sigmoid function and the tanh function, which can speed up the training of the deep learning network. The normalization technology was used to reduce the complexity of learning rate of the model. With Custom developed model used for training different number of vegetable image data set, the experimental results showed that the classification accuracy decreases as the number of images in data set decreases. The experimental verification indicates that the accuracy rate of the deep learning method in the test data set reached as high as it can depending on data set.

II. RELATED WORK

We briefly review the most recent literature on these methods, including the following, since this work is primarily related to the topics of Online Prediction for Vegetables, Attributes and Multi-labeling

a. *Online Prediction for Vegetables and food supply chain*

Image Processing Techniques for Food Materials: Standard image recognition algorithms (Veggie Vision) for food materials have several flaws. Bolle et al. (1996) [10] created an intelligent food materials recognition method using derived image texture statistics, color, and other characteristics for the first time to achieve the classification of food materials detection and can be any number of free and placed. However, the vegetables stored in the "black box" are resistant to light. Zhang et al. (2012) [11] suggested that shape features be included in the identification feature, as well as the use of support vector machine (SVM) to classify food materials, to mitigate this issue. Tao (2014) [12] proposed a complete local binary pattern texture function extraction algorithm that extracts image color characteristics using hue, saturation, value, color histogram, and external point / interior point histogram, then uses the nearest neighbor classifier to classify food materials. However, the experimental use of vegetables only takes into account differences in light intensity, and an intelligent food materials recognition system (Veggie Vision) was created. However, while the vegetables in the "black box" are susceptible to the effects of light, the experimental use of vegetables only considers variations in light intensity and developed an intelligent vegetable recognition system (Veggie Vision). However, the vegetables on the "black box" are light-sensitive. Paul et al. (2014) suggested a system that uses a three-dimensional (3D) color histogram in conjunction with CNN to classify food products as viewing angle and light transition. However, there is only one experimental image, and the majority of the viewing angles are not readily apparent.

b. *Attributes and Multi-labelling*

A visual property that appears or disappears in an image is referred to as an attribute. Different properties may be used to characterize different aspects of an image, such as colors, textures, and shapes. Some recent research has centered on how to link human-interaction applications through these intermediate attributes, where a clear consistency between computer representations of query attribute phrases and human query expressions can occur. Global vs. Local Attributes: If the global properties in the image are described, e.g., 'soil celery'. In general, global attributes do not involve specific object parts or locations. Localized attributes are used to describe one or more locations of an object, e.g. 'Little red pepper'. Both types are not easy inferred, Because the performance of the under-sampling classifier may be reduced if the classifier is trained only on advanced tags that do not have spatial information such as bounded frames. However, some of the work in (Jayaraman et al., 2014) [13], Zhou et al. (2013) [14] show that sharing visual knowledge can counteract the effects of the lack of training samples. Attributes and Multi-labeling: Learning to assign multiple labels to an image is known as image multi-labeling. If the problem is adapted as is, problems will occur as the number of labels grows, making the possible mixture of output labels impossible to manage (Tsoumakas et al., 2007) [15]. To address this, the problem is broken down into a single set of binary classifiers, resulting in a common conversion approach. Multi-label learning is the process of predicting co-occurring attributes. Sandeep et al. (2014), on the other hand, applied multi-task learning to facilitate a priori sharing or use of any of the mark relational heuristics. Another project is to rate the mark scores using the deep CNN rating tool (Gong et al., 2013) [16].

III. DATA SETS AND METHODS

A. Vegetable image dataset preparing and preprocessing

At present, there is no massive number of datasets for all kind of variety vegetable images but we collected as much as we can. The dataset we used comes from Kaggle, a web-based data-science environment that allows users to find and publish data sets, explore and create models. And we just simply didn't used that dataset, we had to delete manually some images which are not good and merged some images from different sources. Twenty-three vegetable categories were trained in this project. Since the number of images in each of the 23 vegetable categories used in this project was inadequate for learning, the vegetable image data sets were rotated at 90°, 180°, and 270° using the data expansion process. The image data sets were expanded by four times using this approach. Some images from the image dataset are shown in Figure 1. Figure 1 shows the picture of the original, rotated 90°, 180°, and 270°, for each group of vegetables from left to right. In the vegetable image data set, each image size is different; to neatly organize the images, they are processed into the same size as (220*220 pixels).

B. Model construction and model training

Our custom model has 13 layers, the first layer is the convolutional layer, the last three layers are the fully connected layers followed with the SoftMax activation function . It can be seen from Figure 2 that the back of the first, fifth, and sixth layers are the convolution layer, the second and seventh layer is the batch normal. The first batch normal layer and second batch normal layer is directly followed by the LeakyReLu layer and MaxPool layer.

Each feature map of the convolutional layer is almost always obtained by combining the multiple feature maps calculated by the upper layer. The main function of the convolutional layer is feature extraction. The calculation process of the convolutional layer is as follows.

$$x_n^l = \Sigma_{i \in M_n} x_i^{l-1} * k_{in}^l + b_n^l$$

Where, x_n^l represents the n^{th} feature map of layer l; M_n represents a set of feature maps selected from the input feature maps; k_{in}^l represents the i^{th} element in the n^{th} convolution kernel of layer l; b_n^l represents the n^{th} offset of layer l; "*" represents the convolution process.

Figure 1 Five categories of vegetable data expansion

Data normalization is the layer after the first layer in our custom model. Global data normalization transforms all the data to have zero-mean and unit variance. However, as the data flows through a deep network, the distribution of input to internal layers will be changed, which will lose the learning capacity and accuracy of the network propose an efficient method called Batch Normalization (BN) to partially alleviate this phenomenon. It accomplishes the so-called covariate shift problem by a normalization step that fixes the means and variances of layer inputs where the estimations of mean and variance are computed after each mini batch rather than the entire training set. Batch normalization has many advantages compared with global data normalization. Firstly, it reduces internal covariant shift. Secondly, BN reduces the dependence of gradients on the scale of the parameters or of their initial values, which gives a beneficial effect on the gradient flow through the network. This enables the use of higher learning rate without the risk of divergence. Furthermore, BN regularizes the model, and thus reduces the need for Dropout. Finally, BN makes it possible to use saturating nonlinear activation functions without getting stuck in the saturated model.

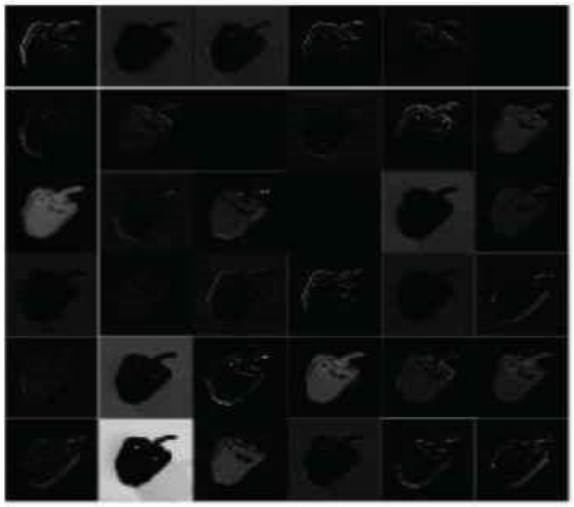
```
[ ] model=ConvNet(num_classes=23).to(device)
    print(model)
```

```
ConvNet(
  (conv1): Conv2d(3, 8, kernel_size=(7, 7), stride=(1, 1), padding=(1, 1))
  (bn1): BatchNorm2d(8, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu1): LeakyReLU(negative_slope=0.01)
  (pool1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(8, 16, kernel_size=(7, 7), stride=(1, 1), padding=(1, 1))
  (conv3): Conv2d(16, 32, kernel_size=(7, 7), stride=(1, 1), padding=(1, 1))
  (bn2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu2): LeakyReLU(negative_slope=0.01)
  (pool2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=80000, out_features=4000, bias=True)
  (fc2): Linear(in_features=4000, out_features=4000, bias=True)
  (fc3): Linear(in_features=4000, out_features=23, bias=True)
  (softmax): Softmax(dim=1)
)
```

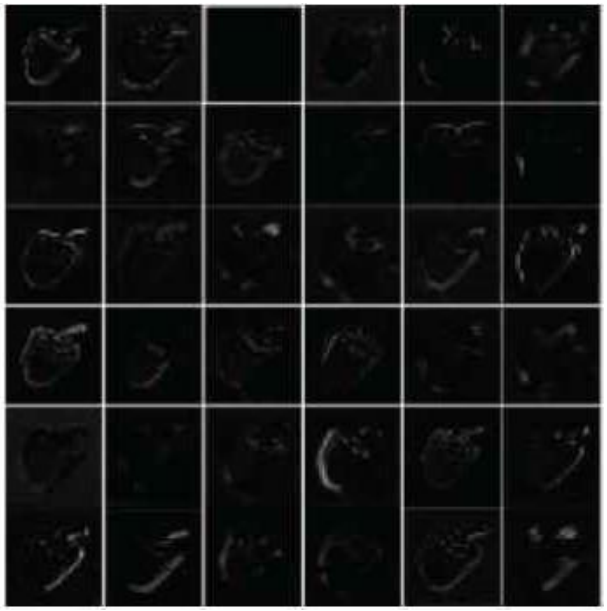
Figure 2 Structure of Model

The Custom model has three convolutional layers, with the performance characteristics of the convolutional layer visualized using green bell pepper as an example. The contour function of the image near the input layer is transparent and similar to the form of the original image, as seen in Figure 3. The features increasingly blur and abstract as the number of layers increases; it is possible to speculate that the layers of the model reflect the characteristics of various degrees of abstraction. The pooling layer, also known as the down sampling layer, generates the input function map's sampling result. The pooling layer alters the scale of the feature rather than the amount of feature maps.

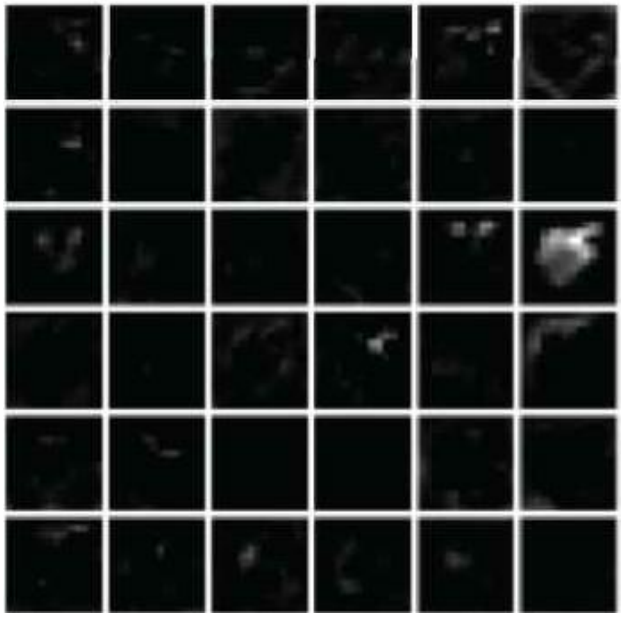
a. The Original Image



b. Conv1- Output Feature



c. Conv2 – Output Feature



d. Conv3 – Output Feature

Figure 3 Output features of partially convolutional layers

The pooling layer is also called the down sampling layer, this layer produces the sampling result of the input feature map. The pooling layer does not change the number of features map but change the size of the feature. The calculation of the pooling layer is as follows.

$$x \quad {}^{l}_{n} = \beta \quad {}^{l}_{n} down(x \quad {}^{l-1}_{n}) + b \quad {}^{l}_{n}$$

where, $\beta \quad {}^{l}_{n}$ represents the $n \quad {}^{th}$multiplication of the layer l; down(.) represents the pooling function.

The pooling layer in this report uses the max pooling process. The procedure is performed after the first, second, and fifth convolutional layers (conv5). Finding the maximum value in each field and then extracting the key features from the initial features map is what max pooling is all about. The maximum pooling is calculated as follows.

$$P \quad {}_{n} = max \quad {}_{i \epsilon R} \quad {}_{n} C \quad {}_{i}$$

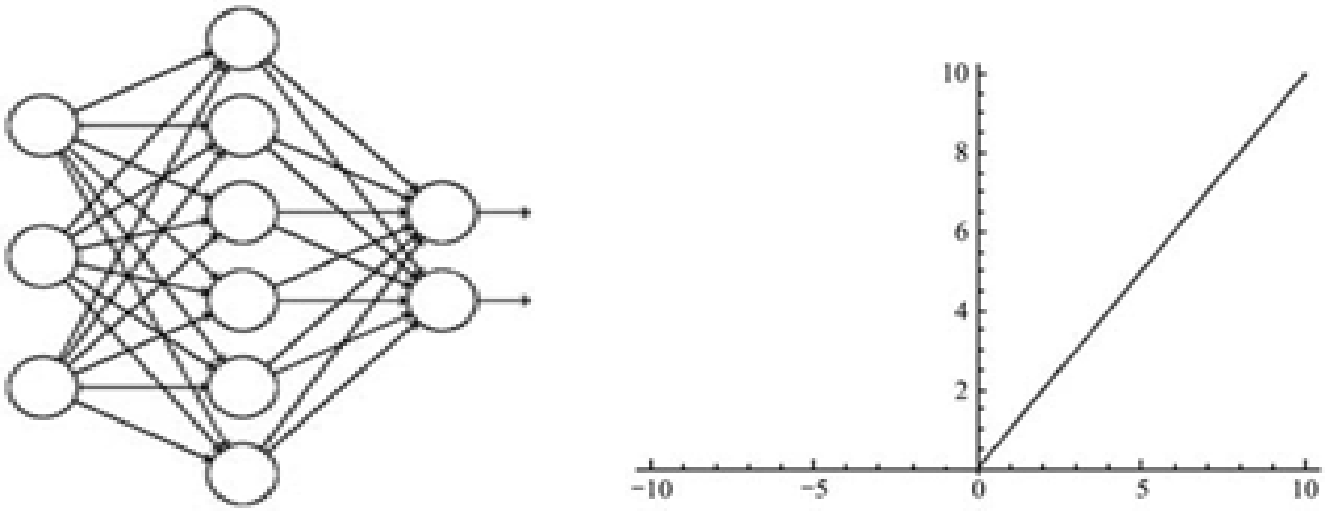
Where, $R \quad {}_{n}$ represents the $n \quad {}^{th}$pooling area in the feature map; $C \quad {}_{i}$represents the $i \quad {}^{th}$pixel value of $R \quad {}_{n}$.

The feature vectors of the convolutional layer output are minimized by max pooling operations, and the feature propagation invariance is increased. It will also help to avoid overfitting.

The pooling layer's input is extracted from the previous convolutional layer, and its primary function is to have high robustness while reducing the number of parameters. Since the pooling layer usually has no parameters, only the input parameters are derived during back propagation, and the weights are not modified. The output function followed by each conv layer and the full connected layer is the ReLU.
The output function of the AlexNet network uses the ReLU instead of the traditional sigmoid function and the tanh function, the calculation formula for ReLU is as follows.

$$f(x) = \max(0, x)$$

ReLU is a nonlinear equation that is unsaturated. If the input is greater than 0, the output of the ReLU function is equal to the input; otherwise, the output is 0. Furthermore, by using the ReLU function, the output does not saturate as the input steadily rises.



a. Fully Connected

b. ReLu Function Curve

Figure 4 Fully Connected layer and ReLu Function Curve

In order to overcome the problem of overfitting, in training our custom network model, first, the learning rate of 0.01 is used to make the network converge as soon as possible. Then reduce the learning rate continuously, and make the network reach the optimal solution (with learning rate of 0.001). The last of the full connected layer is a SoftMax layer with 4000 outputs, according to the number of packets of the vegetable image data set, the number of output layer classification is changed from the original 4000 to 23. The traditional gradient descent method is used to train the parameters of the network. The whole learning process in the network gradually reduces the learning rate, to improve the learning speed of the initial network while ensuring that the model can make the loss function reliable convergence.

IV. EXPERIMENTS AND RESULTS

Initially, we tried transfer learning to the AlexNet model used to perform the image classification. This model had different parameters and layers. AlexNet has eight layers, with the first five being convolutional layers and the last three being fully connected layers. The pooling layer is at the back of the first, second, and fifth convolutional layers, while the SoftMax layer is at the last (output layer). The Response - normalization layer, which is norm1, norm2 layer, is immediately preceded by the first convolutional layer (conv1) and the second convolutional layer (conv2).

To verify the relationship between the accuracy rate of vegetable image classification and the number of image data set, an approximate of 500 images were randomly selected and trained on the AlexNet network from 1840 vegetable image data set. The issue faced with this model was that accuracy rate of vegetable image classification was much less than the current Custom model we developed.

We tried some techniques in data pre-processing like Data Augmentation which includes flipping, rotation and cropping.
As a result, the model became more perplexed and produced poor performance.
So, the above are the experiments we attempted while working on this project, and both of them failed to yield satisfactory results, so we moved on to our custom model, as explained in section 2.

Data scientists use a method called cross-validation to see if their model is overfitting. They divide their data into two parts: the training set and the validation set. The validation set is only used to test the model's results, while the training set is used to train it. Metrics on the training set show us how the model is doing in terms of training, but metrics on the validation set give us a measure of the model's quality, or how good it will make new predictions based on data it hasn't seen before.

At the moment our custom model has an accuracy of 0.050729 on the training data set and 0.053658 on the validation data set. The accuracy and losses of both validation and training data set are analyzed in figure 5 and 6.

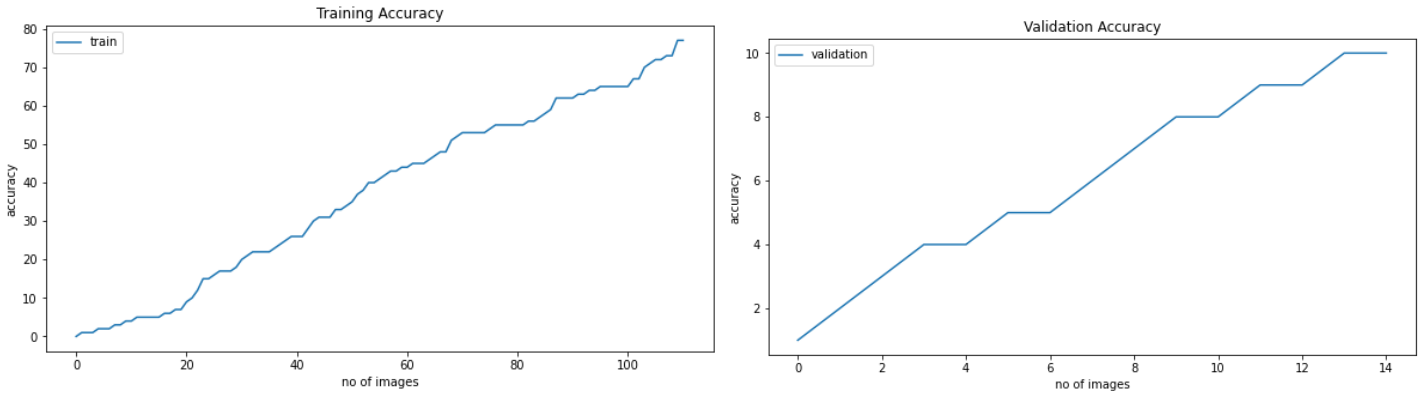


Figure 5 Training Accuracy and Validation Accuracy

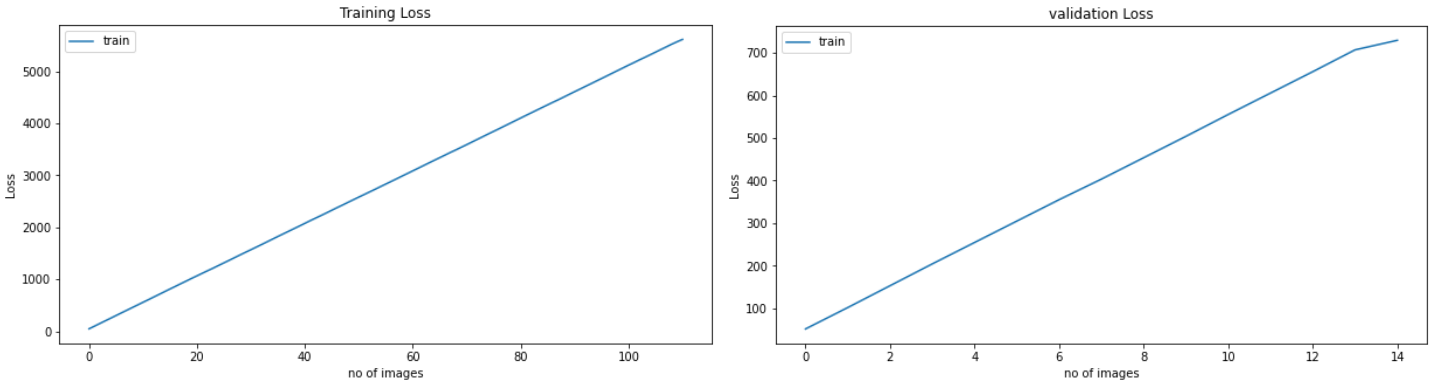


Figure 6 Training Loss and Validation Loss

V. DISCUSSION AND CONCLUSION

The application of image recognition and computer vision technologies in the food industry and agriculture are evolving. Size, color, form, texture, and defect are the most significant quality characteristics of agricultural products. A computer vision system is used to replace manual vegetable identification, providing authentic, fair, and non-destructive ratings. Acquisition, segmentation, feature extraction, and classification are the four major stages of computer vision-based quality inspection. The aim of this study is to investigate and compare the different methods/algorithms. The detailed experiments conducted in this report conclude that, although a number of researchers have suggested different methods for image recognition of vegetables, a comprehensive computer vision-based device with improved performance is still required.

Although the recognition and classification of vegetable image data set has achieved some success, the accuracy has increased by just a small fraction as compared to traditional AlexNet on the dataset we gathered, there are still flaws that can be improved from the following aspects:

The data set for vegetable images is being improved. The dataset is difficult to clean, because the quality of the images are quite different, such as light, shooting location and so on. While there are fewer types of vegetables included in this report, the variety of vegetables and the number of vegetable images for each type should be increased on this basis to include more of the vegetables we see on a daily basis. Utilizing ensemble learning techniques. On the basis of this article, ensemble learning approaches can be used to increase the precision of vegetable image recognition and classification by integrating the characteristics and advantages of various models.

This report proposed a robust approach for vegetable image classification based on a deep learning algorithm. The following is the key work: This article suggests an image classification system based on convolutional neural network to meet the need for automated recognition of vegetables to target classification. From the Kaggle, 23 different types of vegetable images are collected. Using a deep learning framework to train our own data set, and validation data set to tune the hyperparameters and then using the test data set to test the network's classification effect. The experimental findings demonstrate that the accuracy rate declines as the number of data sets decreases by using the network model to train various numbers of vegetable image data sets.

References

[1] Lee S H, Chan C S, Mayo S J and Remagnino P, "How deep learning extracts, and learn leaf features for plant classification," in *Pattern Recognition*, 2017, pp. 1-13.

[2] Krizhevsky A, Sutskever L and Hinton G E, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012.

[3] Dan C, Meier U and Schmidhuber J., in *Multi-column deep neural networks for image classification*, 2012, p. 3642–3649.

[4] Tan W X, Zhao C J, Wu H R and Gao R., "A deep learning network for recognizing fruit pathologic images based on flexible momentum," in *Transactions of the CSAE*, Translated version, 2015, pp. 20-25.

[5] Zeng X and Jie L I, "Time-frequency image recognition based on convolutional neural network," in *Machinery & Electronics*, 2016.

[6] Zhou T, "An image recognition model based on improved convolutional neural network," in *Journal of Computational & Theoretical Nanoscience*, 2016, p. 4223–4229.

[7] A. M, "Improved gait recognition based on specialized deep convolutional neural networks," in *Computer Vision and Image Understanding*, 2017, p. 103–110.

[8] Gong D X and Cao C R, "Plant leaf classification based on CNN," in *Computer and Modernization*, 2014, pp. 12-15.

[9] Hu J T, Fan C X and Ming Y, "Trajectory image based dynamic gesture recognition with convolutional neural networks," in *International Conference on Control, Automation and Systems, IEEE*, 2015, p. 1885–1889.

[10] Bolle, R. M. , Connell, J. H., Haas, N., Mohan, R. and Taubin, G., "U.S. Patent No. 5,546,475. Washington, DC: U.S. Patent and Trademark Office," 1996.

[11] Zhang, Y. and Wu, L, "Classification of fruits using computer vision and a multiclass support vector machine," in *Sensors*, pp. 12489-12505.

[12] Tao, H., Zhao, L., Xi, J., Yu, L. and Wang, T., "Fruits and vegetables recognition based on color and texture features," in *Transactions of the Chinese Society of Agricultural Engineering*, 2014, pp. 305-311.

[13] Jayaraman, D., Sha, F. and Grauman, K., "Decorrelating semantic visual attributes by resisting the urge to share," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* , pp. 1629-1636, 2014.

[14] Zhou, Q., Wang, G., Jia, K. and Zhao, Q., "Learning to share latent tasks for action recognition," *In Proceedings of the IEEE International Conference on Computer Vision*, pp. 2264-2271, 2013.

[15] Tsoumakas, G. and Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining,*, p. 3(3), 2007.

[16] Gong, Y., Jia, Y., Leung, T., Toshev, A. and Ioffe, S., *Deep convolutional ranking for multilabel image annotation*, 2013.