Efficient commercial classification of agricultural products using convolutional neural networks

Ali Jebelli[1], Arezoo Mahabadi[2], Rafiq Ahmad[3]
[1,3]Laboratory of Intelligent Manufacturing, Design and Automation, Department of Mechanical Engineering, University of Alberta, Edmonton, Canada
[2]School of Engineering Science, Tehran University, Tehran, Iran

Article Info

Article history:

Received May 21, 2021
Revised Aug 19, 2021
Accepted Nov 6, 2021

Keywords:

Agricultural
Barcode
Classification
Convolutional neural network
Plant images
Visual geometry group

ABSTRACT

Agricultural products, as essential commodities, are among the most sought-for items in superstores. Barcode is usually utilized to classify and regulate the price of products such as ornamental flowers in such stores. However, the use of barcodes on some fragile agricultural products such as ornamental flowers can be damaged and lessen their life length. Moreover, it is time-consuming and costly and may lead to the production of massive waste and damage to the environment and the admittance of chemical materials into food products that can affect human health. Consequently, we aimed to design a classifier robot to recognize ornamental flowers based on the related product image at different times and surrounding conditions. Besides, it can increase the speed and accuracy of distinguishing and classifying the products, lower the pricing time, and increase the lifetime due to the absence of the need for movement and changing the position of the products. According to the datasheets provided by the robot that is stored in its database, we provide the possibility of identifying and introducing the product in different colors and shapes. Also, due to the preparation of a standard and small database tailored to the needs of the robot, the robot will be trained in a short time (less than five minutes) without the need for an Internet connection or a large hard drive for storage the data. On the other hand, by dividing each input photo into ten different sections, the system can, without the need for a detection system, simultaneously in several different images, decorative flowers in different conditions, angles and environments, even with other objects such as vases, detects very fast with a high accuracy of 97%.

Corresponding Author:

Rafiq Ahmad
Laboratory of Intelligent Manufacturing, Design and Automation, Department of Mechanical Engineering
University of Alberta
9211 116 Street NW, Edmonton, Canada
Email: rafiq.ahmad@ualberta.ca

1. INTRODUCTION

Barcoding of products is very common in stores, but for some products, such as agricultural, it is challenging, time-consuming, and it requires providing special instruments for printing and reading barcodes and maintenance of them, which results in high costs [1]. Likewise, barcodes are spoiled or distorted due to insignificant variations such as changes in the color of the background, the width of the barcode, and the formation of added outlines based on environmental circumstances, for example, light, moisture, and temperature, which necessitate the re-encoding of the products [1]-[7]. The barcode's substances are also toxic or plastic, which can damage agricultural or environmental food products [5].

Nowadays, most of the agricultural products available in stores have labels with the brand of their sales company; most of these labels are made of plastic, which causes environmental damage due to their non-biodegradability. Some procedures are currently under review to replace these labels, including edible labels, edible ink, or laser labeling on the fruits [6]. Classification of plant images using neural networks and machine learning techniques is among the least expensive and accurate methods since this method does not need any specific equipment (just a camera) comparable to quick response code (QR code) [2], [8]-[11].

The main advantage of such a system is that sensitivity to various environmental conditions such as light, heat, or humidity is almost eliminated, and there is no need to barcode the products since the visual features of the product will replace the barcode. This research designed a system to categorize different flowers with different colors and types without sensitivity to environmental conditions, including light and background patterns, using the convolutional neural network and visual geometric group 16 layers (VGG16) design trained on ImageNet. We aimed to reduce the classification error related to agricultural products without constructing any waste and increasing product identification speed and its lifetime. The obtained results demonstrated the efficiency of the proposed approach.

2. CONVOLUTIONAL NEURAL NETWORKS

Neural networks have led to great advances in machine learning and are the main reason for the deep learning boom. Neural networks such as convolutional neural networks (CNNs) have successfully worked with image data. Since their use in the ImageNet competition in 2012, they have been at the forefront of research and industry in working with images.

The CNNs are powerful neural networks that use filters to extract features from images. This task is done so that the information of pixel situations is protected. Accurately, they are neural networks in which convolution is used instead of general matrix multiplication. With this approach, the forward function would perform better, and the number of network parameters would be meaningfully reduced. Most common frameworks can support CNNs, such as Tensorflow-Keras and PyTorch. Also, the possibility of custom design of CNNs is available based on the NumPy library. Convolutional neural networks have diverse applications industrially, particularly in computer vision (face recognition, paper documents/optical character recognition (OCR) digitalism, internet of things).

An image received by a computer will be perceived as an array of numbers. The number of arrays rests on the image size (based on pixels). For instance, if a joint photographic group (JPG)-colored image (480*480 px) is inputted to a computer, its replacement array would have 3*480*480 cells (number 3 is pointed to red, green, and blue (RGB)). Each JPG image array element is numbered between 0 and 255 (because the variables are stored as eight units), which indicates the pixel intensity.

As shown in Figure 1, the neural network's input image is divided and processed into smaller arrays, called receptive field arrays, and the filter that dot product by the receptive field is called the kernel; which is an array of the same size as the receptive field that the numbers in the filter are called weight or parameters, respectively. At each glance, the filter sees a portion of the image and then, it moves over it to scan other parts as well (i.e., it convolves). The result of convolving the image with the filter (kernel) is a 2D array called activation map or feature map. If we use p filters instead of one filter, we will have p arrays, and since each filter extract one feature from the image, using a higher number of filters can increase accuracy. Convolution allows the ability to perform different filters such as derivative calculation, edge recognition, and blurring. Each kernel is a small matrix with an odd-dimensional size, and the midpoint of this matrix is called an anchor. An anchor is utilized to indicate the placement of the filter on the image. Moreover, the filter depth is equal to the image depth, i.e., the image depth of y*y*3 is equal to the kernel size of x*x*3.

The filter is of same size as the receptive field, and the result of convolving the image using any filter is a feature map. Each filter must have the same depth as the image (i.e., in this image, the filter and the image both have three RGB color depths). Note that the filter size (height and width) should be an odd number to place the anchor in the center of it.

As shown in Figure 2, the CNN network is composed of three main layers, namely, convolutional layer, pooling layer, and fully-connected layer [12]. In the first level, there is a convolution layer that can extract new features from the image using various kernels. Then, max-pooling is carried out with the aim of noise reduction, overfitting prevention, and decreasing network parameters and dimensions. The combination of "convolution+max-pooling," known as the convolution layer, can be repeated many times to build a deeper network. The max-pooling layer's output is referred to as the fully-connected layer after alteration to the 1D vector. The standard algorithms of neural networks in fully connected layers are employed to output a single vector for which the number of elements is equal to the number of classes defined by the user.

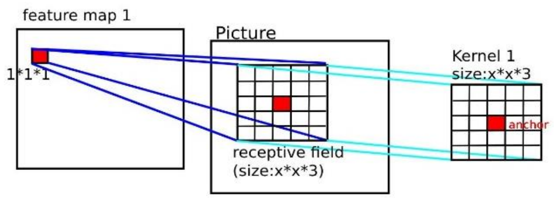


Figure 1. Filter images

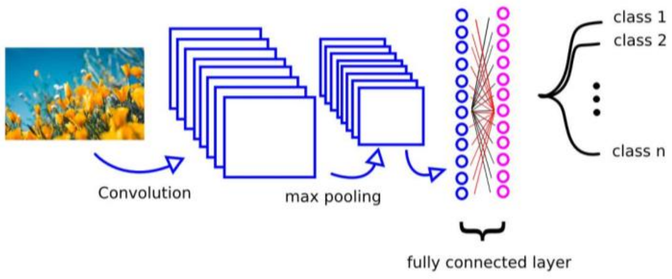


Figure 2. The general illustration of CNN design

3. CNN LAYERS
3.1. Convolution layer

The general mathematical format of convolving two 2D matrixes is presented in (1) [13].

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} * \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix} = \sum_{i=0}^{m} \sum_{j=0}^{n} x_{(m-i)(n-j)} y_{(1+i)(1+j)} \qquad (1)$$

According to that equation, the kernel must initially be horizontally and vertically flipped for convolving the kernel with the image, and every time the filter is located on a specific portion of the image, the element with the same indices is multiplied, and the sum of all results is placed in the anchor element. Figure 3 indicates the steps of the filter convolving with the image.

By using image convolving by the filter, the matrix size would be reduced because the margin elements of the matrix related to the image will never be located in the center of the filter (The anchor is never placed in the margin elements of the image array). As will be detailed in section 3.4, prevention of the matrix's size and loss of image data is done by padding, i.e., adding rows and columns to the image edges, which is a type of zero padding.

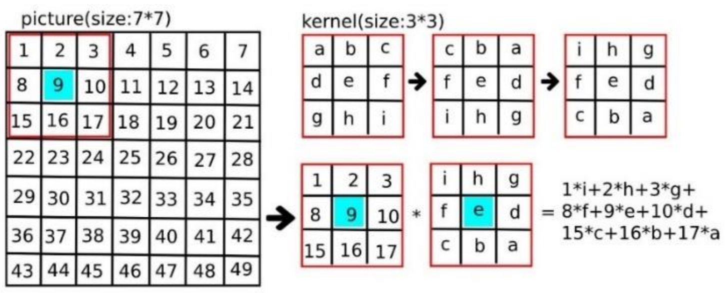


Figure 3. Steps of the filter convolving with the image

Convolving a 3*3 filter with a 7*7 image, at which receptive field is indicated by the red square in the image and anchor is shown by a blue square; the sum of multiplying receptive field elements by the filter is located in the anchor. Feature map will have a size of 5*5.

Similarly, in convolving a filter with an image, a constant is added to the available sum, called bias. According to Figure 3, the bias is then equal to zero. When the function of the linear components is ended with the input, a non-linear function is applied. In this stride, the activation function is implied to the result obtained from the linear components. Based on Figure 4, this function converts the input signals to the output ones. Let us consider n-numbered inputs, X_1 to X_n, applied to corresponding weights (W_1 to W_n). The weights are initially multiplied at their inputs, and added to the bias. The result is u. An activation function, f is then implied to u, and lastly, the final output is reached as y_k from the neuron. There are numerous activation functions from which Sigmoid, ReLU, and softmax are the most common.

3.2. Max pooling layer

The pooling layer reduces the image size since this layer combines the adjacent pixels of a specific portion of the image and converts them to a unit value. Many image-processing applications utilize the pooling technique. To perform the operation in the pooling layer, the pixel selection of images and their expected value should be determined. The adjacent pixels are usually selected from the square matrix and the numbers of selected pixels are varied based on the related problem. As shown in Figure 5, the representative value is usually the mean or maximum of pixels [14].

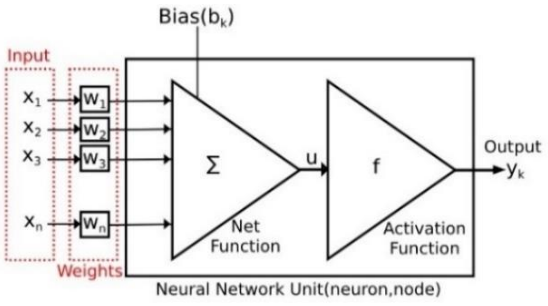


Figure 4. Strides of convolution layer upon a neuron

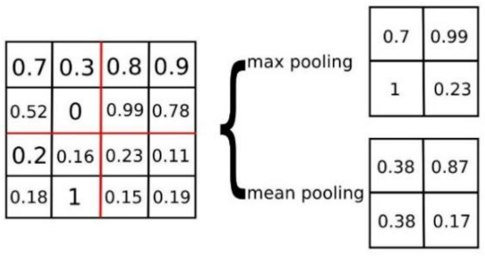


Figure 5. Pooling result using two different methods of mean and maximum

3.3. Fully connected layer

The fully-connected layer transfigures 2D feature maps to a 1D-feature vector to endure the display process of the features, see Figure 6. The fully connected layer permits a presentation of the vector with a given size as the network output, which can be further used to classify the images or carry on the subsequent processes. Using this method, the convolution network transfigures the main image's raw pixels into the classes' scores layer-by-layer at the end of the network. The main issue of these layers is the high number of parameters, leading to the high cost of the process function spent during training time.

So, one commonly used method is to either remove these layers altogether or reduce the number of connections in these layers using some alternatives. Explicitly, convolution and fully-connected layers are the layers with the ability to perform some transformations that function as the activations in the input and the function of the parameters of neuron weight and bias. On the other hand, the pooling and ReLU layers implement a constant function; the available parameters in convolution and fully connected layers are trained by the descending gradient method to meet the classes' scores counted by that convolution network the labels of each image in the train set [15].

3.4. Hyperparameter

Three parameters control the size of the output, i.e., depth, stride, and padding, which is a type of padding that is zero padding. The output is an arbitrarily selected parameter that controls the neuron numbers connected to a region in the input in the convolution layer. Using stride parameter, the deep columns are determined around the spatial dimensions (width and height). Sometimes, it is better to cover the input boundary with zero. In other words, the image frame is filled with zero: a row and column of 0 are added to the opening and end of the image. Hence, the image is placed in the "zero" frame, as shown in Figure 7.

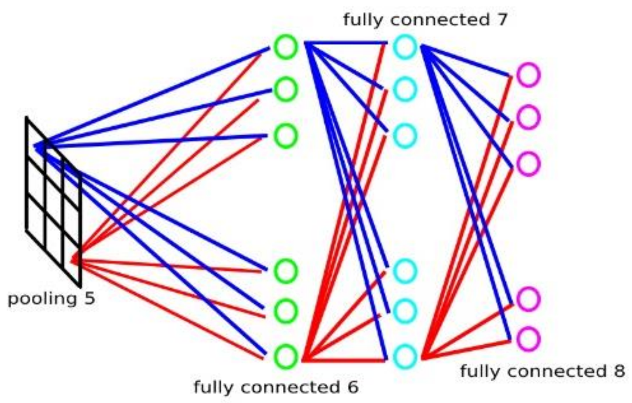


Figure 6. The operations of fully connected layers are located at the last pooling layer

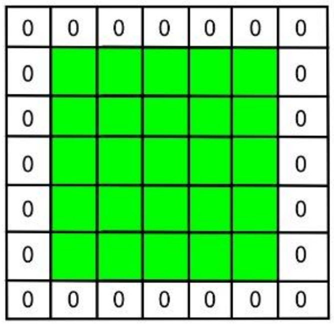


Figure 7. A sample with zero padding with one added layer to the matrix

The sizes of the input (W), the perceptual area of the convolution layer neurons (F), and the stride (S) as well as the amount of zero P padding on the boundary of the input can be used to calculate the size of the output. The following equation can be utilized to estimate the appropriate numbers of the neurons.

$$\frac{W-F+2P}{S}+1 = output\ width \tag{2}$$

Where W stands for the input width, F for the filter width, P for the zero padding, and S for the stride.

If the equation result is a decimal number, the stride's utilized value is not correct, and that neurons have not the ability to be arranged next to each other with this value and cannot be well located across the input flipped. For a better understanding, let us consider the equation of Figure 8. In that figure, zero padding is added to the input width of 5. Formerly, by using a filter of size 3 and stride 1, the image's left side is convoluted and the output size is equal to 5. By using a filter of size 3 and stride 2, the right side of the image is convoluted with an output size of 3. It shows that the output is reduced, and the data set is lost even though the padding is used. Generally, zero padding is set based on (3).

$$P = \frac{F-1}{2} \tag{3}$$

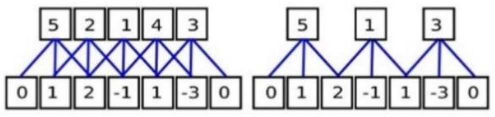


Figure 8. Convolving by a filter of size 3, the input of size 5, the stride of 2 on the right side, and stride of 1 on the left side

4. TRANSFER LEARNING

Transfer learning is a machine learning technique whose pre-trained pattern may be retargeted for the second task [16], [17]. The other important application of transfer learning is the high performance of distinguishing similar images using a pre-trained model in small data sets. Many of the pre-trained models are available such as bidirectional encoder representations from transformers (BERT), universal language model fine-tuning (ULMFiT), and visual geometry group (VGG). Among these models, the proper one for this study was investigated. BERT and ULMFiT are used for language modeling, and VGG16 is used for image classification. VGG16 may have different weights; for example, the trained VGG16 on ImageNet or the trained VGG16 on modified national institute of standards and technology database (MNIST) consists of 1000 classes of 1.2 million images on the data set. Some of these data set classes are animals, cars, stores, dogs, food, tools, etc. On the other hand, MNIST is learned using handwritten numbers, including ten classes from 0 to 9. Considering the case of ornamental flower images, the trained VGG16 on ImageNet was found to be the most suitable.

5. VGG16

In 2014, a standard network design was proposed by VGGNet. It utilized the very small convolution filters of 3*3 and split the number of feature maps after the 2*2 pooling. The network depth was increased from 16 to 19 layers, which led to the flexibility increase for learning non-linear scales by deep learning. Images are required to be in the form (224, 224, 3) in VGG16. The intensity average of all pixels minus each pixel is performed as the only pre-processing. The image passes through a group of convolution layers, with very small filters of 3*3. Filters of 1*1 are utilized in one of the configurations used as a linear conversion of the input channels. The filter stride is the constant amount of a pixel. Padding 3*3 is also used. The pooling layers are 2*2 windows with a stride of 2. Three fully-connected layers follow a set of convolution layers. Each of the first two layers has 4096 channels, while the third layer has 1000 classes. The last layer is called softmax. The configuration of fully-connected layers is similar in all networks. All hidden layers are equipped with a linear rectifier unit. The architecture of this network is indicated in Figure 9. The depth of the configuration increases from left to right because the added layers are increased [18], [19].

The important point in machine learning systems is data overfitting, leading to errors in the test data thus, showing a high error toward the new data. For this purpose, two concepts of dropout and data augmentation are very beneficial.

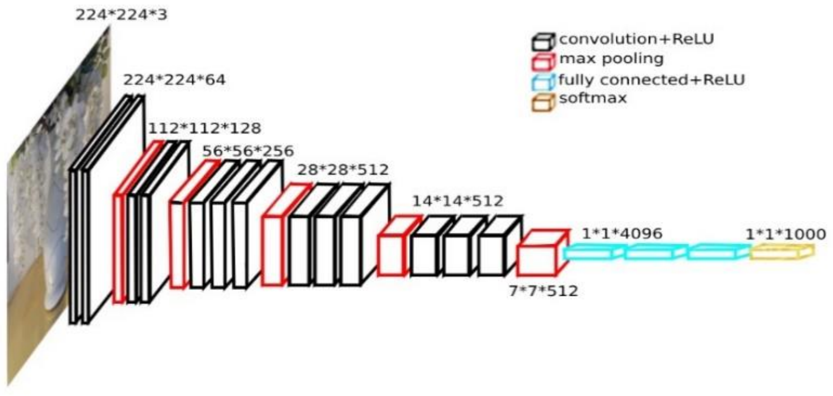


Figure 9. The architecture of the VGG network

5.1. Dropout

Using all same-weighted neurons is among the reasons for overfitting during machine learning of the system. In dropout layer, a fraction of weights was utilized at each learning phase instead of learning all the weights simultaneously. That is to say, a probability (0 to 1) can be utilized for neurons in the learning system. The probability of zero value means using all neurons in the learning phase, while the probability of one value means excluding all neurons from the learning stage [20].

The eliminated neurons no longer participate in the forward or backward pass using this method. Then, every time an input is available, the neural network shows different architectures, but all of these architectures share the weights. This method reduced the complexity of neuron organizing because a neuron cannot depend on any certain neurons' presence. Therefore, it is required to learn more powerful features that would be useful in relationship with different random sets of neurons. In this study, the dropout value was determined as 0.5.

5.2. Data augmentation

The small amount of available data set is the other reason for overfitting. To overcome this issue, one solution is data augmentation, in which the data set is extended by small changes such as spinning and flipping without the need to collect new data. Besides, the defined model can be Invariant using data augmentation. An invariant convolutional neural network can classify things in different directions in a robust way. In particular, CNN can be invariant toward movement, point of view, size, or light intensity (or a combination of them).

6. DESIGNED MODEL

We designed the model in four stages: a) loading the required libraries, b) outlining the model architecture and the weight of kernels (utilizing the VGGNet design), c) pre-processing of data and extracting features from image (herein, we utilized the pre-process system of trained VGG16 on ImageNet), and d) training the model using train data and setting of classification parameters using valid data.

In numerous images in the training set, the classification model has a higher chance for proper function. A total of 944 images were employed to be classified with a proportion of 15 to 85 into two train and valid datasets. The image series of 807 for train and 137 for test data was composed of eight classes: rose flower, orchid, chrysanthemum, matricaria, bird of paradise, marguerite daisy, Alstroemeria, and Lilium. Also, we defined 35 epochs for training the model, and we have assumed 32 for batch size. According to Figure 10, training loss decreases in each stage of training, and after the 30th period (last epoch), it reaches 0.0056 with a training accuracy is 99.98%. The system training was completed in 5 minutes, and the time of image recognition was of 0.26 seconds.

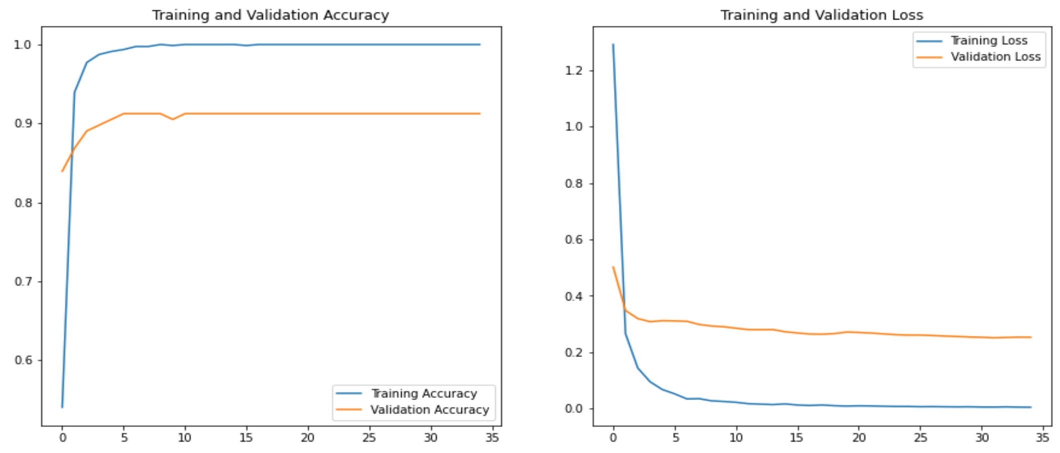


Figure 10. Loss and accuracy diagrams in model training using train and valid data

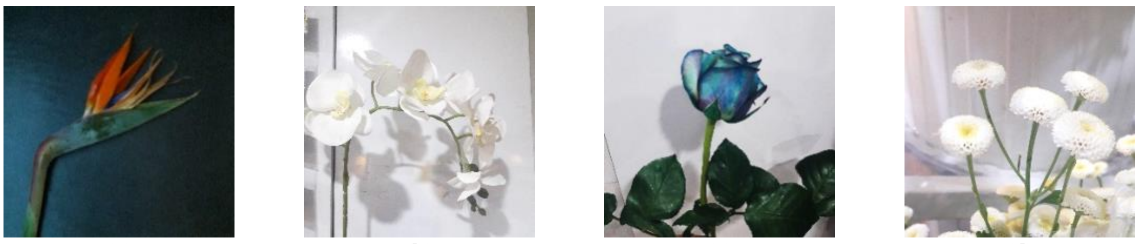
7. PRELIMINARY TESTS

To determine the class associated with each image taken by a HERO 4 Black measure 4000 by 3000 pixels, the image was initially sized to 224*224*3 using ImageDataGenerator, and its features extracted from 512*7*7 VGG network. Then, the extracted features were inputted through the designed model, and for each image, a single output vector was extracted who's each element shows the extent of the image's similarity/difference to each flower class. The maximum element is chosen as the flower class, and the image of the flower is presented in the monitor along with the similarity percentages and the recognized class.

Due to VGG network use, all image sizes of the training and test phase are square. In this work, we retained eight flowers, i.e., rose flower, orchid, chrysanthemum, matricaria, bird of paradise, marguerite daisy, alstroemeria and lilium. Different weather conditions were considered, such as rainy and sunny weather, and different places were selected such as a home, garden, store. Furthermore, different hours of the day and night were considered in the test and training data. For real-time testing of the designed network, we used 68 images split into three categories: single flower (14 images), bunches (27 images), and pods (27 images).

7.1. Test 1-single flowers

The Figure 11 shows using the designed model, 14 images of flower branches were classified in this test. To test the network at different conditions, images were taken at various times of the day and night, with stem and leaves. Based on the obtained results, 12 images were detected correctly with a model accuracy of 85%. Note this can be improved by increasing the number of train photos from different angles of the flower branches. It should also be noted that the flower background is also involved in the network training. Therefore, the best way to train the network was to consider the same amount of data for the different classes and the same angles of each flower so that the designed model does not have priority in choosing the flower class. As can be observed, the flower conditions are very different. Matricaria flower was not recognized correctly because the train images of this flower are mostly in the form of big bunches.



(a) (b) (c) (d)

Figure 11. Four samples of the tested images: (a) matricaria, (b) rose, (c) orchid, and (d) bird of paradise

7.2. Test 2-bouquet

In the second test as shown in Figure 12, 27 flowers were inputted as bunches. The flower bunches' challenge is the density and intertwined of the flowers and the change in the shape of the edges, making it difficult for a trained system to distinguish a flower from the rest of them and imply more errors than the mode of a single flower. Twenty-one images were recognized correctly, i.e., a reached trained network's accuracy of 77.78% due to the compression and intertwined of the flowers and non-detectable flower edges. However, the resolution and blur of the image were very effective.

Orchids were incorrectly classified due to their intertwining and chrysanthemums were incorrectly classified due to the paper around the bouquet. Also, liliums were incorrectly classified because of the presence of hands in the photo.

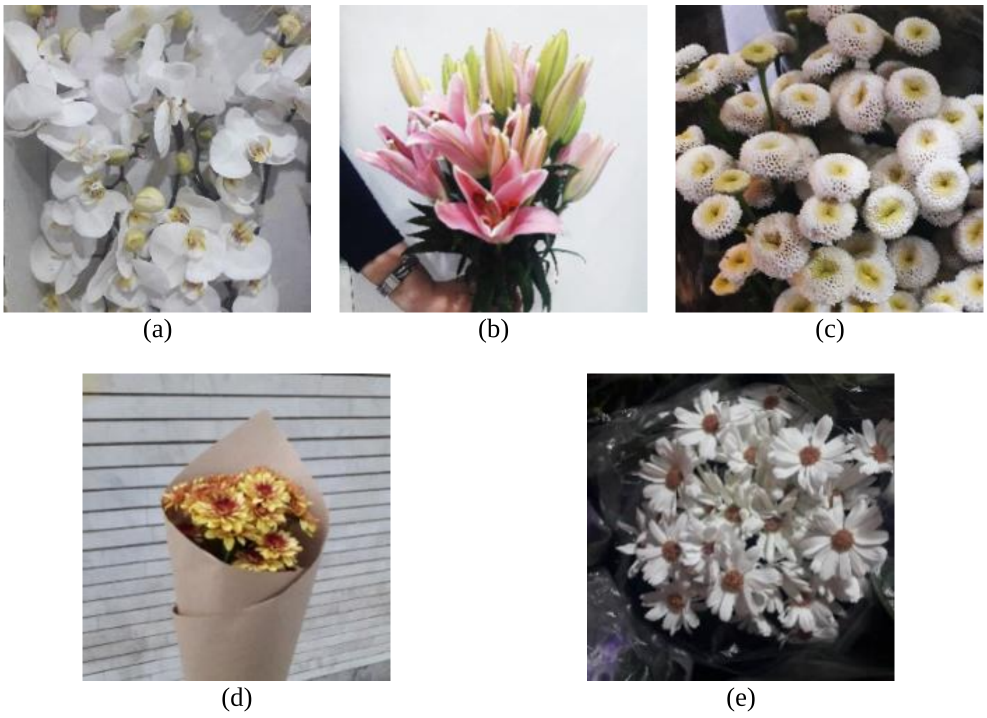


Figure 12. A sample from the tested images: (a) chrysanthemum, (b) matricaria, (c) lilium, (d) orchid, and (e) marguerite daisy

7.3. Test 3-flowerpots

The other challenge in the classification of flowers is the presence of other objects in the image. In this section, in addition to any special shape of the flowers and their intertwining, the shape and color of a vase or a pot, as well as the fact that a specific flower is not allocated in the entire image frame, can affect the system response. In fact, owing to numerous flowers and the allocation of a small and low-resolution segment of an image to the desired flower, the potential error increases in pots with high numbered of small flowers or large pots with similar-colored flowers. In this test, 20 flowers were correctly recognized out of 27 ones, and the system accuracy was reduced to 74.07%, see Figure 13. Chrysanthemum pots are incorrectly recognized due to the large number of flowers and Alstroemeria vases due to the high density of the photo.

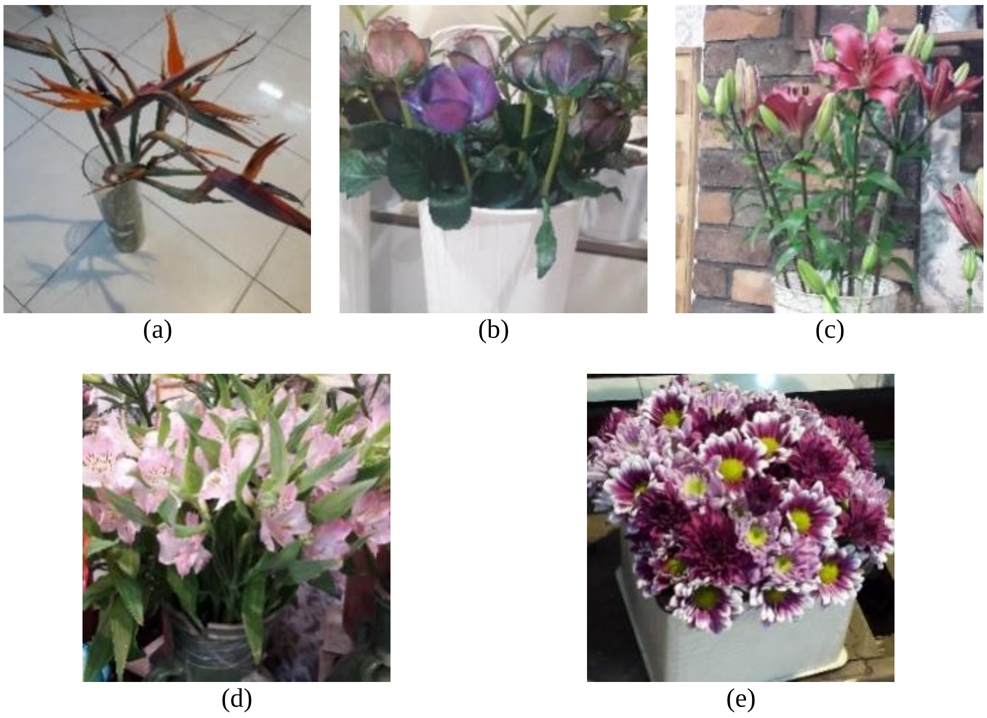


Figure 13. A sample of the tested images: (a) chrysanthemum, (b) Alstroemeria, (c) lilium, (d) rose, and (e) bird of paradise

8. RESULTS AND DISCUSSION

Based on the results above obtained from preliminary tests, all tested images were appropriately categorized according to different environmental conditions such as the time of the day and the flower states (with or without branches, bunch, and/or pods). The following conditions are operative on system error within the train/test data:

a. Imaging from different angles and various spaces.
b. Balance in the number and condition of imaging in each class such as packing of flowers in the image, imaging angles, and imaging environment.
c. Quality of the camera and image opaqueness.
d. Similarity degree of the leaves.
e. Effect of the branches.

Also, the conditions that affect the tests, including:

a. Compression degree of flowers and leaves.
b. Possible presence of other items rather than flowers, such as covering paper, hands, pods.
c. Changing of image size (drag rate).
d. Central or marginal position of flowers in the image.
e. Number and size of flowers in the image.

Considering the obtained results from the previous section, we changed the data set and the designed network into a data set of flowers compactly put next to each other from different angles, and in the images, leaves and flower petals cover most of the image space with completely recognizable edges. Likewise, each image was processed ten times using the following algorithm: the images are outputted into a 10*8 matrix of elements x_{ij}, which stands for the ith image and the jth category of the flowers. As shown in Figure 14, nine cuts are created with the ½ dimensions of the image. For instance, if the maximum matrix element is 0.99 with i=1 and j=4, it is thus located in the upper left corner of the image with a similarity of 99% to the marguerite daisy.

The proposed algorithm was then included to the network. From that, only two images out of 68 images were detected incorrectly, and we received an increment of system accuracy up to 97% as shown in Figure 15. The proposed algorithm was performed over the zoomed test images, preventing overcrowding and image error. The paper around the chrysanthemum was eliminated, the marguerite daisy and matricaria were exaggerated, and more details were entered into the network. However, due to the very similarity of papillon to the lilium, the Figure (e) has not been correctly identified.

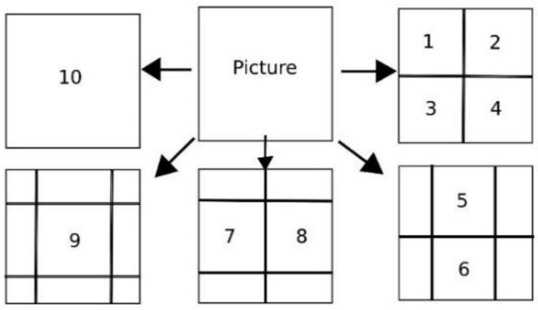


Figure 14. Nine new images have been created and entered along with the origin image into the network from cutting each image

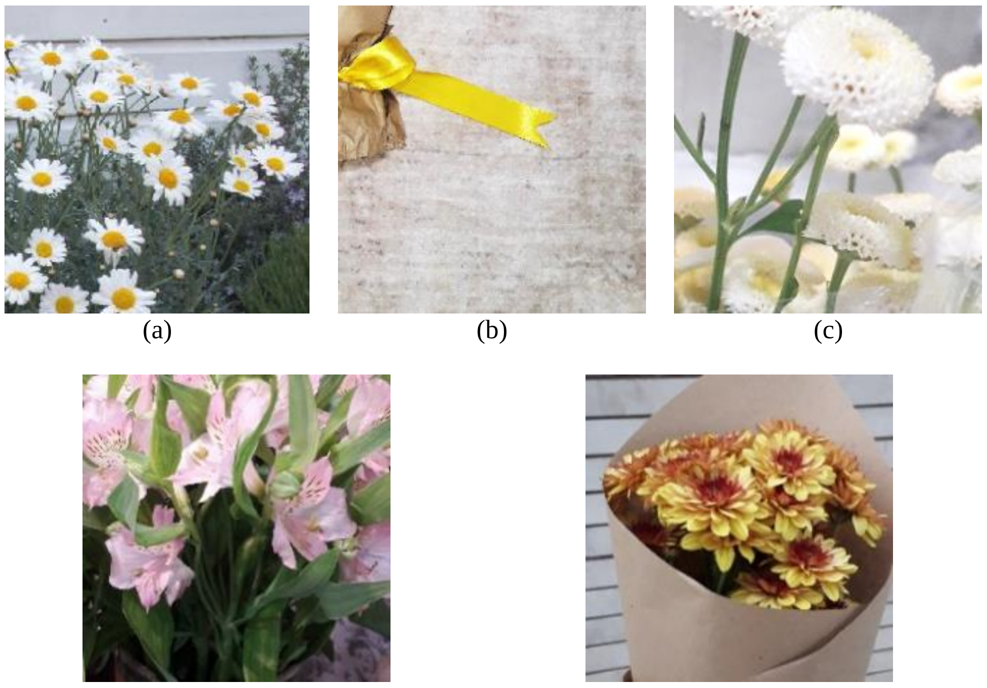


Figure 15. Zoomed test images: (a) marguerite daisy, (b) chrysanthemum, (c) Alstroemeria, (d) matricaria, and (e) lilium

9. CONCLUSION

Currently, barcoding of products is performed to recognize the type and price of products in stores. However, the barcoding of agricultural products is harmful to both the product and the environment and reduces the lifetime of the product due to the need for one-by-one barcoding and the presence of toxic materials in barcodes, and costly and time-consuming procedures. The application of machine learning and the image process allows for identifying products with high accuracy and swiftness using a simple camera without producing any waste-free of barcoding. In this research, we investigated the classification of eight different kinds of ornamental flowers in different environmental conditions by designing a convolution network of VGG16 to increase the rapidity and accurateness of their detection in stores, and increase the lifetime of agricultural products, and prevention of damages toward products and environment as a result. The obtained results demonstrated that the designed network categorizes the input images at high speed with 97% accuracy, with the ability to categorize the inputted images simultaneously.

ACKNOWLEDGEMENTS

We express our gratitude to the Minister of Economic Development, Trade, and Tourism for funding this project through Major Innovation Funds. The authors also would like to acknowledge the NSERC (Grant Nos. NSERC RGPIN-2017-04516 and NSERC ALLRP 545537-19) for further funding this project.

REFERENCES

[1] L. Catarinucci, S. Tedjini, R. Colella, F. P. Chietera, K. Zannas and D. Kaddour, "3D-Printed Barcodes as RFID Tags," *2020 International Workshop on Antenna Technology (iWAT*), 2020, pp. 1-4, doi: 10.1109/iWAT48004.2020.1570613640.

[2] V. Uzun and S. Bilgin, "Evaluation and implementation of QR code identity tag system for healthcare in Turkey," *SpringerPlus*, vol. 5, pp. 1454, 2016, doi: 10.1186/s40064-016-3020-9.

[3] J. S. Park, Y. O. Lee, C. Y. Park, Y. S. Kim, K. B. Nahm, and J. D. Kim, "Barcode recognition for multistrip lateral flow assay reader," *Sensors and Materials*, vol. 32, no. 8 (2), pp. 3669-3678, 2020. [Online] Available: SM2367.pdf (myukk.org)

[4] R. Ahmad, S. Tichadou, J. Y. Hascoet, "New computer vision based snakes and ladders algorithm for the safe trajectory of two axis CNC machines," *Computer-Aided Design*, vol. 44, no. 5, pp. 355-366, 2012, doi: 10.1016/j.cad.2011.12.008.

[5] A. F. Isnawati, M. A. Afandi and J. Hendry, "Performance Analysis of FBMC-OQAM System for Barcode and QR Code Image Transmission," in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 2020, pp. 162-166, doi: 10.1109/IAICT50021.2020.9172034.

[6] R. Ahmad, S. Tichadou, J. Y. Hascoet, "3D Safe and intelligent trajectory generation for multi-axis machine tools using machine vision," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 4, pp. 365-385, 2013, doi: 10.1080/0951192X.2012.717720.

[7] A. Jebelli and M. C. E. Yagoub, "Efficient robot vision system for underwater object tracking," in *2016 2nd International Conference on Control Science and Systems Engineering (ICCSSE)*, Singapore, 2016, pp. 242-247, doi: 10.1109/CCSSE.2016.7784390.

[8] M. L. Gao *et al*., " Microporous hexanuclear Ln(III) cluster-based metal–organic frameworks: Color tunability for barcode application and selective removal of methylene blue," *Inorganic Chemistry*, vol. 56, no. 1, pp. 511-517, 2017, doi: 10.1021/acs.inorgchem.6b02413.

[9] J. S. Ariadna, "Carbon dioxide laser fruit labelling," M.S. thesis, Escola d'Enginyeria Agroalimentària i de Biosistemes de Barcelona, Dept. de Teoria del Senyal i Comunicacions, Universitat Politècnica de Catalunya, Spain, 2020. [Online] Available: https://upcommons.upc.edu/handle/2117/327088.

[10] A. Picon, M. Seitz, A. A. Gila, P. Mohnke, A. O. Barredo, J. Echazarra, "Crop conditional Convolutional Neural Networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions," *Computers and Electronics in Agriculture*, vol. 167, 2019, doi: 10.1016/j.compag.2019.105093.

[11] Z. Lin et al., "A Unified Matrix-Based Convolutional Neural Network for Fine-Grained Image Classification of Wheat Leaf Diseases," in *IEEE Access*, vol. 7, pp. 11570-11590, 2019, doi: 10.1109/ACCESS.2019.2891739.

[12] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[13] P. Wang and D. Wang, "Filter-and-convolve: A Cnn based multichannel complex concatenation acoustic model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 5564-5568, doi: 10.1109/icassp.2018.8462395.

[14] B. B. Traore, B. Kamsu-Foguem, and F. Tangara. "Deep convolution neural network for image recognition." *Ecological Informatics*, vol. 48, pp. 257-268, 2018, doi: 10.1016/j.ecoinf.2018.10.002.

[15] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi and J. A. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690-6709, Sept. 2019, doi: 10.1109/TGRS.2019.2907932.

[16] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[17] T. Chuanqi, F. Sun, T. Kong, W. Zhang, Ch. Yang, and Ch, Liu. "A survey on deep transfer learning." in *Artificial Neural Networks and Machine Learning – ICANN 2018*, Springer, Cham , 2018, pp. 270-279, doi: 10.1007/978-3-030-01424-7_27.

[18] K. Tu *et al*., "A non-destructive and highly efficient model for detecting the genuineness of maize variety, 'JINGKE 968′ using machine vision combined with deep learning," *Computers and Electronics in Agriculture*, vol. 182, pp. 106002, 2021, doi: 10.1016/j.compag.2021.106002.

[19] J. J. Sanchez-Castro *et al*., "A Lean Convolutional Neural Network for Vehicle Classification," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, 2020, pp. 1365-1369, doi: 10.1109/ISIE45063.2020.9152274.

[20] S. Rohrmanstorfer, M. Komarov, and F. Mödritscher, "Image classification for the automatic feature extraction in human worn fashion data," *Mathematics*, vol. 9, no. 6, pp. 624, 2021, doi: 10.3390/math9060624.