Nigerian Research Journal of Engineering and Environmental Sciences 7(1) 2022 pp. 231-237
p ISSN: 2635-3342; e ISSN: 2635-3350

Nigerian Research Journal of Engineering and Environmental Sciences
Journal homepage: www.rjees.com

Original Research Article

Simplified Image Classification for Nigeria's Agricultural Produce through Deep Neural Network Techniques

***Oyebode, K.O. and Osah, U.J.**

Department of Design and New Media, School of Media and Communication, Pan-Atlantic University, PMB 73688, Lagos, Lagos State, Nigeria.
*koyebode@pau.edu.ng; josah@pau.edu.ng

http://doi.org/10.5281/zenodo.6725124

ARTICLE INFORMATION

Article history:
Received 01 Jan, 2022
Revised 27 Mar, 2022
Accepted 31 Mar, 2022
Available online 30 Jun, 2022

Keywords:
Deep neural network
Image processing
Convolution
Computer vision
Quality checks
Agriculture

ABSTRACT

Image processing techniques can support quality checks for agricultural produce. Noting the relevance to the Nigerian context, this paper details the development of a computer vision system for automatic screening of produce. The development included four phases, namely: model development, training, graphical user interface (GUI) creation, and module testing. Model development involved a convolution process with capacity to extract useful features from the image of an agricultural produce. Training enabled the created model to learn crucial image structures based on fine-tuned mask parameters that support the classification of similar images efficiently. The GUI enables non-technical users to train new models, as well as carry out single and multiple classifications. The testing phase enables the evaluation of the system. Images are tested using a pre-trained model as well as without a pre-trained model. The model backed by the pre-trained model performed better than that not associated with a pre-trained model. Importantly, whereas the application of deep neural networks has been explored, a useful contribution here is the introduction of a user friendly GUI to underpin its employment.

1. INTRODUCTION

The agricultural small and medium enterprise (SME) sector in Nigeria possesses unrealized potential (Sertoğlu et al., 2017). One way to contribute to the realization of this potential, is to have standard organizations within the sector commit to establishing structures, to ensure thorough quality checks of produce outputs (Vibhute and Bodhe, 2012).

Recently, concern has been expressed by the international community about the quality of Nigeria's exported agricultural produce, which as it is believed is as a result of indifference to quality checks (Famurewa, 2016). Such quality checks aid in eliminating the possible export of bad product. Granted, the temptation may have existed in the past to evade intense quality checks for produce outputs for a number of convincing reasons (Raji and Alamutu, 2005), these reasons however may currently be considered baseless due to the advent of more technologically driven solutions to agricultural produce quality checks (Vaysse et al., 2005; Kodagali and Balaji, 2012; Adhitya et al., 2020).

One option in particular, which is the focus of this study is image processing of agricultural output, enabled by artificial intelligence (AI) (Adhitya et al., 2020). More recently, the availability of A.I has enabled the development of systems capable of mimicking the human visual sensory organ, to more consistently and efficiently perform agricultural produce grading functions (Kodagali and Balaji, 2012). Many advanced countries have embraced this technological progression (Gunasekaran, 1996). This is however not the case in Nigeria (Raji and Alamutu, 2005; Wossen et al., 2019). Specifically, within the Nigerian context, the proposition of AI-enabled image processing of agricultural produce quality checks is limited to research. For instance, Peter and Abdulkadir (2018), proposed an image processing and neural networks application for determining the readiness of maize. Whereas such studies are considered welcome developments, it is still bothersome that these innovative solutions are not being adopted at scale by institutions and bodies that require them. Retrospectively, there is a need for more studies aimed at illustrating how AI-enabled image processing techniques can be adopted by non-technically savvy users in the agricultural sector in Nigeria. The non-existence of such studies is a noted research gap that this study hopes to address. Noting this background, the objective of this study is to develop a graphical user interface (GUI) image classification system for automatic screening of agricultural produce. Pointedly, a four stage process of the development of a simplified computer vision system for automatic screening of agricultural produce is described. With this system, it is intended that it will allow non-technical users to carry out training and image classification of agricultural yields enabled through deep neural network.

2. METHODOLOGY

The development of a deep neural network for automatic screening of agricultural produce is categorized into four sections. These include the model development phase, training, graphical user interface creation, and testing module phase. In respective order, these phases are discussed thus:

2.1. Model Development

This section discusses the two deep neural networks (DNN) used for the classification of agricultural produce. A neural network is classified deep if it has more than one hidden layer. If it has one hidden layer, then it is classified as an artificial neural network (ANN). The first DNN is seen in Figure 1 with 13 hidden layers while that of Figure 2 has 3 layers. Each layer has its function. At the core of the model is the convolution layer (Conv) (Smelyakov et al., 2019). This layer convolves a 3 by 3 mask with an input image with the key objective of extracting features from images. This process enables the DNN to extract useful features that enable it to carry out classification tasks optimally. Figure 1 details the first developed model. This model has four crucial blocks. Each block on the left-hand side has the convolution layer, activation layer, and pooling layer. First, colored images of size 128 by 128 are fed into the model as input. The images then pass through a series of convolution, the activation and the pooling then follow. The activation layer enables the model to adapt to the non-linearity of input images while the pooling layer shrinks the size of the image, at the same time retaining critical image features (Kumar and Kumar, 2018). The dropout layer helps to overcome over fitting (Sanjar et al, 2020). The final output is the classification layer. This layer classifies the input image as either bad or good. One may also make use of a pre-trained model. Pre-trained models are popular in DNN, as a way of transferring the knowledge gained from another model that had undergone training on a very large dataset. The pretrained model VGG19 (Keras, 2021) has been utilized in Figure 2 to form another variant of the DNN model.

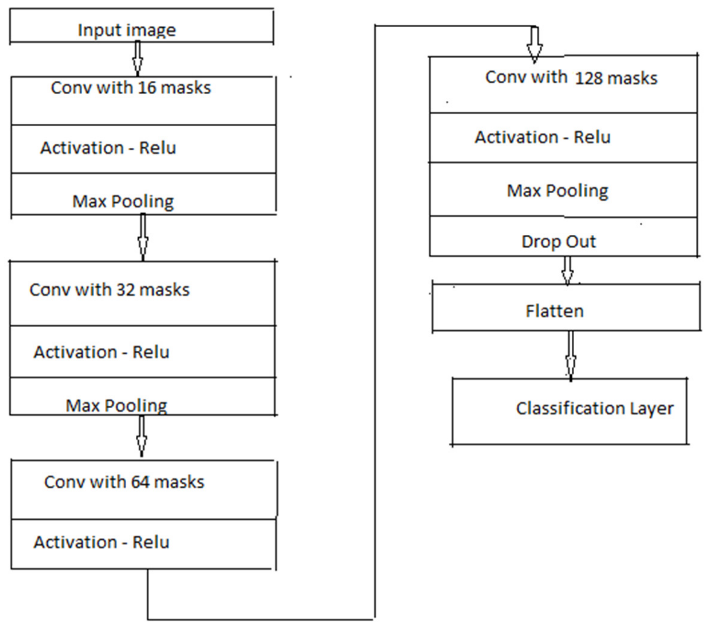


Figure 1: The deep neural network for image classification

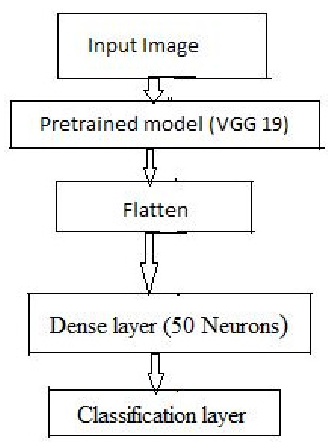


Figure 2: The deep neural network using a pre-trained model for image classification

2.2. Training Phase

As with any supervised machine learning model, the training process is crucial. This is so because it helps the model learn important image structures such as edges, curves, and contours. During the training process, the mask parameters (see Figure 2) were adjusted such that they were capable of classifying similar images efficiently. The sigmoid activation function (Xing and Wu, 2020) was deployed at the output layer for classification purposes. During the forward pass (when images are fed into the model's input), the output generates classifications that are evaluated using binary cross-entropy (Ho and Wookey, 2020). The binary cross-entropy then evaluates the model's performance on selected images and calculates the loss between the expected outcome and the predicted. The Adams optimizer (Zhang, 2018) was then used to minimize this loss via a backward propagation strategy (from the output layer back to the input layer). This process was continued until no further performance improvement. At this time, the training is concluded and the trained model is saved as a "*.h5 file*". This .h5 model can then be called to classify similar images never seen before. This implies that for given agricultural produce samples, one would have to identify good and bad sample images and then pass them to the model. An example is depicted in Figure 3 using carrots. 100 images were used. 50 for bad images and 50 for good images. In addition, a total of 150 images of tomatoes were used, 75 for each class.

(a) (b)

Figure 3: Sample training images (a) good carrot image (b) bad carrot image

Here the collection of good and bad sample images are gathered and passed onto a developed DNN model. This research makes use of two DNN models – one that is built on acquired training images and another built with a pre-trained model alongside acquired training images. Developing a model using a pre-trained model can be described as taking advantage of a transfer learning approach. This means one has transferred the trained parameters of hundreds of masks onto an existing model. Pre-trained models are useful particularly when training with fewer images. This training process goes through series of loops, and for every loop, the output error is evaluated and mask weights are adjusted to minimize this error during the next loop (training process).

2.3. Development of the Graphical User Interface

The proposed software has three GUI requirements. They are the ability to train models, the ability to do single image classification and multiple image classification. The Tkinter framework, a plugin for Python programming was used for interface development. The GUI and its backend development went through three phases. In the first phase, the buttons, text boxes, text areas were created. The development of backend codes using Python happened in the second phase. The backend code implements user actions, such as clicking a button to start a training process, image selection button, and classifications buttons. This stage also establishes the connection between the GUI actions and the code behind them. Extensive testing was carried out in the third phase. The phase ensures that user actions are properly wired to their respective Python implementations. This is to ensure it meets the stated requirements.

2.4. Graphical User Interface

The GUI provides the opportunity for non-technical users to train a new model, and to also carry out single and multiple classifications. The GUI hides the technicalities away from users. The GUI has three sections (Figure 4). The training phase, single image classification, and multiple image classification. In the first section of the GUI, the user points to the folders housing bad and good sample images (75 images for each class is used in the case of tomato). After that, the name of the yet-to-be-trained model is specified. Here the name "tomato20-09-21" is entered. Next, the user clicks on the "Start Training" button. Once the training is completed, the model "tomato20-09-21.h5" is ready for the classification of tomatoes. For the single image classification section, the user selects the model, an image to classify is also selected and then the "Classify Image" button is clicked. Examples of the resultant outputs are shown in Figure 5. The proposed system can be found in (Agric produce classification, 2021).

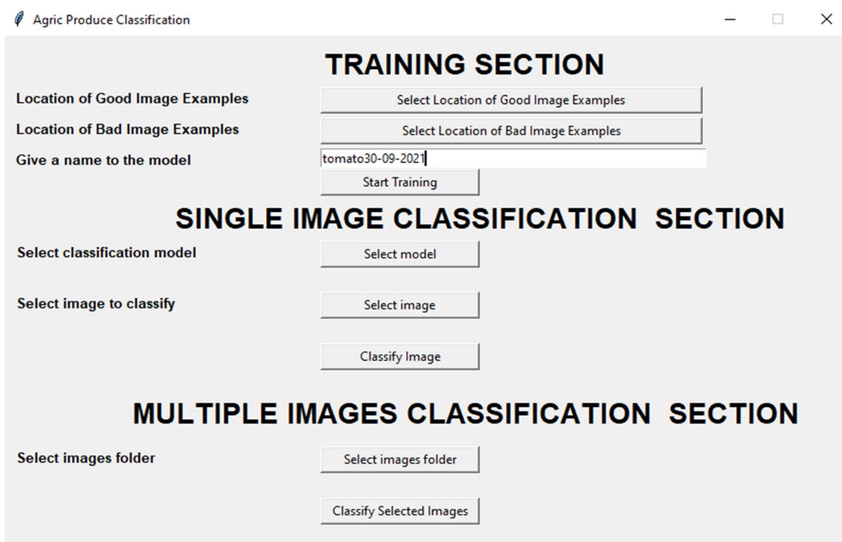
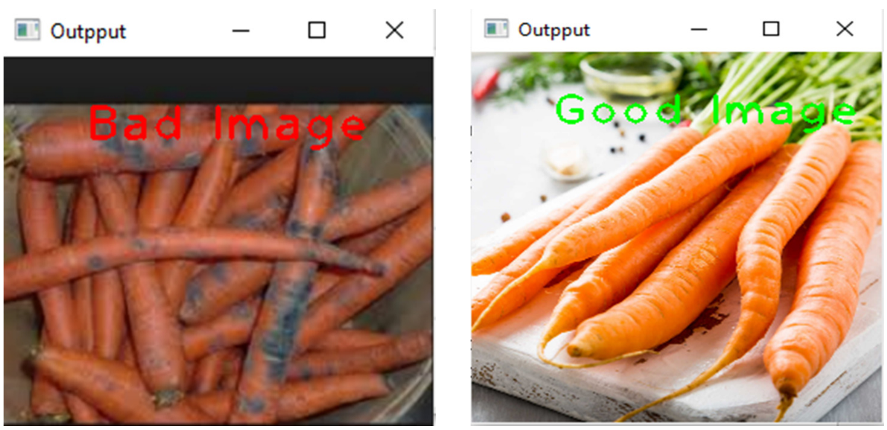


Figure 4: Developed GUI



(a) (b)

Figure 5: (a) ‘Bad’ image classification (b) ‘Good’ image classification

2.5. Testing Module Phase

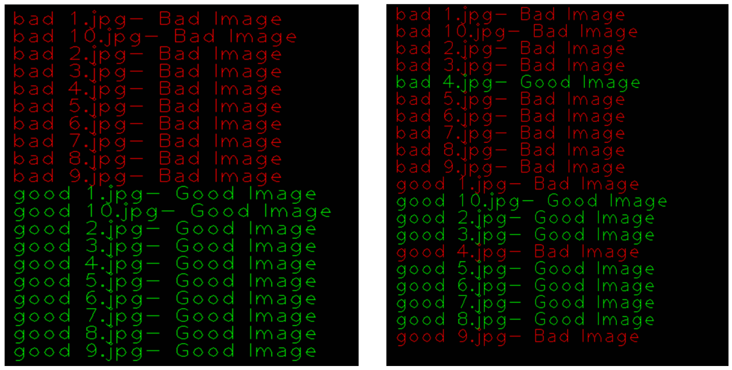
The proposed system is evaluated on two datasets- tomato and carrot. These datasets were acquired from Google, where each had 20 images (Agric produce classification, 2021). Two classification models were developed for each dataset. One model was derived from the pre-trained model (Figure 2) and another without the pre-trained (Figure 1). The classification accuracy (A) metric used for the evaluation of the models is shown in Equation 1.

$$\text{Classification accuracy} = \frac{\text{Total number of correctly classified images}}{\text{Total number of images}} * 100 \qquad (1)$$

Specifically, this phase describes the test carried out on the two datasets (tomato and the carrot). Succinctly put, the user would upload the .h5 model (two .h5 models for tomato and carrot without pre-trained model and two .h5 models for tomato and carrot with pre-trained model) and then select the path to the images (last section of the GUI). After that, the classification pops up. For the multiple classifications, one would point the system to the folder housing images to be tested.

3. RESULTS AND DISCUSSION

The final classification results pop up eventually as depicted in Figure 6. Table 1 shows the outputs of two kinds of DNN models for image classification. As may be noticed, the model backed with a pre-trained model performs better. Importantly, the knowledge transferred from a pre-trained model has improved its performance. The system with a pre-trained model would be ideal in the agricultural space where only few training images can be generated for training purposes.



(a) (b)

Figure 6: Carrot dataset (a) Output of developed software with a pre-trained model, all images are classified correctly (b) Output of developed software without pre-trained model, three images are misclassified

Table 1: Classification outputs of proposed software

Dataset	Classification accuracy (%)	
	Software with a pre-trained model	Software without a pre-trained model
Carrot	100	85
Tomato	85	85

4. CONCLUSION

This research develops a GUI-based image classification system for agricultural produce. It enables non-technical users to also carry out training single and multiple image classification effortlessly. With the illustration employed here two kinds of deep neural networks (DNN) models are applied. One with a pre-trained model and the other without. The results demonstrate that when there are fewer training images, the model supported by a pre-trained model significantly outperforms the one that is not. This shows that when fewer images are available for training, the systems backed with pre-trained models may perform better. It may be suggested therefore, that when fewer images from agricultural produce are collected, the proposed system would still perform optimally. This proposed system would also strengthen the quality control process in ensuring products meet an acceptable quality level before consumption or being exported.

5. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Adhitya, Y., Prakosa, S. W., Köppen, M. and Leu, J. (2020). Feature Extraction for Cocoa Bean Digital Image Classification Prediction for Smart Farming Application. *Agronomy*, *10*(11), pp. 1–16.

Agric produce classification. (2021). *Agric-produce-classifier*. [Online]. Available: https://github.com/kazeemkz/agric-produce-classifier. [Accessed 2 10 2021].

Famurewa, W. (2016). Addressing quality control issues in Nigeria's agricultural exports. *The Guardian*, 1. [Online]. Available: https://guardian.ng/business-services/business-roundup/addressing-quality-control-issues-in-nigerias-agricultural-exports/. [Accessed 2 7 2021]

Gunasekaran, S. (1996). Computer Vision Technology for Food Quality Assurance. *Trends in Food Science and Technology*, *7*(8), pp. 245–256.

Ho, Y. and Wookey, S. (2020). The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. *IEEE Access*, vol. 8, pp. 4806-4813.

Keras, K. (2021). *Keras*. Retrieved September 2, 2021, from https://keras.io/api/applications/vgg/

Kodagali, J. A. and Balaji, S. (2012). Computer Vision and Image Analysis based Techniques for Automatic Characterization of Fruits – a Review. *International Journal of Computer Applications*, *50*(6), pp. 6–12.

Kumar, S. and Kumar, M. (2018). *A Study on the Image Detection Using Convolution Neural Networks and TenserFlow*. International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, pp. 1080-1083.

Peter, A. and Abdulkadir, S. (2018). Application of image processing and neural networks in determining the readiness of maize. *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing, Phu Quoc Island, Phu Quoc city, Vietnam,, February 2018*, pp. 104–108.

Raji, A.O. and Alamutu, A. (2005). Prospects of Computer Vision Automated Sorting Systems in Agricultural Process Operations in Nigeria. *The CIGR Journal of Scientific Research and Development*, *VII*, pp. 1–12.

Sanjar, K., Rehman, A., Paul, A. and JeongHong, K. (2020). *Weight Dropout for Preventing Neural Networks from Overfitting*. 8th International Conference on Orange Technology (ICOT), 2020, pp. 1-4.

Sertoğlu, K., Ugural, S. and Bekun, F. V. (2017). The Contribution of Agricultural Sector on Economic Growth of Nigeria. *International Journal of Economics and Financial Issues*, *7*(1), pp. 547–552.

Smelyakov, K., Shupyliuk, M., Martovytskyi, V., Tovchyrechko, D. and Ponomarenko, O. (2019). *Efficiency of image convolution*. In: 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers, Sozopol, Bulgaria, pp. 578-583.

Vaysse, P., Grenier, G., Lavialle, O., Henry, G., Khay-Ibbat, S., Germain, C. and Da Costa, J. P. (2005). Image Processing as a tool for quality assessment of fruits in bulk shipping bins. *FRUTIC 05*, pp. 381–388.

Vibhute, A. and Bodhe, S. K. (2012). Application of Image Processing in Agriculture: A Survey. *International Journal of Computer Application*, *52*(2), pp. 34–40.

Wossen, T., Alene, A., Abdoulaye, T., Feleke, S., Rabbi, I. Y. and Manyong, V. (2019). Poverty Reduction Effects of Agricultural Technology Adoption: The Case of Improved Cassava Varieties in Nigeria. *Journal of Agricultural Economics*, *70*(2), pp. 392–407.

Xing, S. and Wu, C. (2020). *Implementation of A Neuron Using Sigmoid Activation Function with CMOS*. IEEE 5th International Conference on Integrated Circuits and Microsystems (ICICM), 2020, pp. 201-204.

Zhang, Z. (2018). *Improved Adam Optimizer for Deep Neural Networks. IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1-2.