International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

FRUIT RIPENESS DETECTION USING DEEP LEARNING

1st LINGAMGUNTA ASRITHA
Department of Electronics & Communication Engineering
R.V.R. & J.C. COLLEGE OF ENGINEERING
GUNTUR,Andhra Pradesh-522019
lingamguntaasritha@gmail.com

2nd MEDIKONDA HARI CHANDRA PRASAD
Department of Electronics & Communication Engineering
R.V.R & J.C COLLEGE OF ENGINEERING
GUNTUR, Andhra Pradesh-522019
medikonda1811229@gmail.com

3rd KORRAPATI VARUN SAI
Department of Electronics & Communication Engineering
R.V.R & J.C COLLEGE OF ENGINEERING
GUNTUR, Andhra Pradesh-522019
varunsaikorrapati@gmail.com

4th MEKALA JAYARAJ
Department of Electronics & Communication Engineering
R.V.R & J.C COLLEGE OF ENGINEERING
GUNTUR, Andhra Pradesh-522019
jay.jeeva7@gmail.com

***Abstract*—The agricultural industry has been facing challenges in traditional and manual visual grading of fruits due to its laborious nature and inconsistent inspection and classification process. To accurately estimate yield and automate harvesting, it is crucial to classify the fruits based on their ripening stages. However, it can be difficult to differentiate between the ripening stages of the same fruit variety due to high similarity in their images during the ripening cycle. To address these challenges, we plan to develop an accurate, fast, and reliable fruit detection system using deep learning techniques. The modernization of crops offers opportunities for better quality harvests and significant cost savings. Our approach involves adapting the state-of-the-art object detector faster R-CNN, using transfer learning, to detect fruits from images obtained through modalities such as colour (RGB) and Hyper Spectral Imaging System (HSI). Our system's robustness will enable us to differentiate between fruit varieties and determine the ripening stage of a particular fruit with effectiveness and accuracy. The system will also efficiently segment multiple instances of fruits from an image and accurately grade individual objects.**

***Index Terms*—Modernisation, HSI, CNN, RGB**

I. INTRODUCTION

The demand for high-quality fruits and vegetables both in domestic and foreign markets requires strict quality parameters and preservation of fresh produce for an extended period. However, the current methods of fruit evaluation in the field are time-consuming, involving visual inspection techniques and destructive measuring equipment, which takes valuable technician time away from production. To address these challenges, we are exploring the use of deep learning techniques, specifically deep convolutional neural networks (CNN), for supervised learning in fruit evaluation. CNN has shown remarkable performances for visual object recognition, leveraging the spatial structure of input data, such as images.By implementing CNN in fruit evaluation, we aim to overcome the limitations of traditional visual inspection techniques and destructive measuring equipment. This approach can provide a faster and more objective evaluation of fruit quality, leading to better decision making in production and preservation efforts. Classifying fruits based on their ripening stages can be a challenging task due to the high variability in images of the same variety during the ripening cycle, as well as the natural environments in which they grow. Each fruit variety has a different ripening cycle, which typically lasts for a week. Therefore, identifying the specific fruit variety in an image is crucial. However, the similarity between images of contiguous weeks and the large number of classes makes this task difficult even for experienced farmers. To classify the ripeness of fruit using computer vision, we employed a Convolutional Neural Network (CNN) model that utilises transfer learning. The CNN was adapted to the task of fruit detection, using RGB imagery to identify fruit variety and ripening cycle. Our aim was ambitious, but we achieved notable results through the use of deep learning techniques. The CNN architecture is based on three key principles: local receptive fields, shared weights and bias, and pooling layers. Our approach involved an uncontrolled photographic acquisition method, allowing users to take photographs from any device, directly in the field.

II. METHODS

Convolutional Neural Network (CNN) is a popular deep learning neural network that is commonly used to identify patterns in images, classify data, and perform regression analysis. The network uses algorithms that work in a similar way to the human brain to accomplish these tasks. Various metrics, such as accuracy, loss rate, elapsed time, and confusion matrix, are used to evaluate the performance of CNN models. The primary characteristic of CNN is that it trains itself using hierarchical representations of large datasets containing images. To adjust the weights with respect to input, a backpropagation algorithm is used during the training process. Different CNN architectures have been developed, including MobileNet, VGG16, and InceptionV3, which were utilized in this study to identify three classes of images: ripen, under ripen, and over ripen. These models extract features by dividing images into small pixels using diminutive processing. CNNs can be classified into seven different classes: channel exploitation, feature mapping, spatial exploitation, depth, width, multipath, and attention-based CNNs. The architecture of CNN comprises alternate layers of convolution and pooling, followed by one or more fully connected layers. In the convolution layer, the image is divided into small blocks, which are then convolved with a specific set of weights corresponding to the receptive field. This helps extract locally correlated pixel values. The convolution layer is classified based on size, type of filter, padding, and direction for extracting image features. The pooling operation is used to extract a combination of features that are invariant to translational shifts, thereby helping to reduce overfitting and generalise the results. Fully connected layers are used to classify the image into different categories. It works globally and generates a nonlinear relation of extracting features by analysing the output of previous layers.

A. Transfer Learning Models

Transfer learning is a machine learning technique that involves reusing a model developed for one task as the starting point for a model on another task. It is a commonly used approach in deep learning, particularly for computer vision and natural language processing tasks, where pre-trained models can be utilised as a starting point to improve model performance. There are numerous pre-trained architectures available for training, which makes comparing them a challenging task. In this study, we experimented with various architectures, including MobileNet, VGGNet, and InceptionNet. After a thorough analysis, we concluded that InceptionNet, VGGNet, and MobileNet architectures possess unique properties that make them valuable for evaluating in this project.

1) MobileNet: MobileNet is a specific type of Convolution Neural Network Architecture that is commonly used for image classification tasks. It is referred to as a ”mobile first” model due to its ability to achieve higher accuracy with lower computation resources. MobileNet architectures are designed to not only reduce the overall model size but also significantly improve prediction speed by up to 10 times while maintaining comparable accuracy. This is achieved by utilising a regular Convolutional layer only once at the beginning of the model, with all other layers using Depth-wise Separable Convolution. This type of convolution is a combination of two different convolution layers - Depthwise Convolution and Pointwise Convolution. The depthwise convolution filter performs a single convolution on each input channel, and the pointwise convolution filter combines the output of the depthwise convolution linearly with 1×1 convolutions.

2) InceptionV3: Transfer learning is a popular machine learning method that involves utilising a pre-trained neural network to accomplish a specific task. One example of a pre-trained model is the Inception-3 image recognition model, which is composed of two main parts: a feature extraction component that utilises a convolutional neural network and a classification component that employs fully-connected and softmax layers. This pre-trained model has been shown to achieve state-of-the-art accuracy for recognizing general objects with classes such as ”Ripen”, ”Under ripen”, and ”Over ripen”. The feature extraction part of the model extracts general features from input images, while the classification part classifies the images based on those extracted features.

3) VGG16: The VGG model is a popular deep learning architecture, as discussed in the deep learning sub-chapter. This study proposes a deep CNN with transfer learning using the VGG16 model, where the upper layer is removed and replaced with an MLP block. The fruit maturity dataset used in this study is labelled, and inductive transfer learning is applied using the ImageNet weighted VGG16 model. The weight held by the VGG16 model from generating and classifying 14 million images is transferred to perform feature extraction on the fruit maturity dataset. The extracted features are then fed into the MLP block for classification. The MLP block consists of a Flatten layer, a Dense layer, and Regularize, where regularisation techniques such as Dropout, Batch Normalisation, and Regularizes Kernel are used to reduce overfitting. The Data Augmentation process is also used to increase the size of the dataset, as the number of images in each category is only 140. The activation function used in the MLP block is ReLu activation, and the output layer uses the softmax activation function with eight classes.

B. Deep CNN model framework

This research work utilised a deep CNN approach to detect the ripening stage of fruits. The model was able to identify various features of fruits such as their colour, texture, freshness, and more. These features were then mapped by the deep learning models to indicate the symptoms of ripening and quantify the freshness level, with the model observing them at low abstraction levels. The proposed methodology involved five stages, including image acquisition, feature extraction, feature selection, feature identification, and classification of ripening stages. All the codes were run using Jupyter Notebook, and the testing dataset was employed for prediction and model evaluation.

III. Model Process

In this study, Deep CNN models were utilised to identify the ripening stage of fruits by utilising visual cues and colour-based features during the training process. The models were trained using the same hyper-parameters, such as learning rate, epochs, and batch size, to ensure a fair comparison between the models. Jupyter software with a deep learning toolbox was employed to design, implement, visualise, and monitor the training process of MobileNet, VGG16 and Inception V3

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

models. The image classification process using Deep CNN models involved several stages, including dataset preparation, CNN architecture selection, model training, cross-validation, and model testing.

A. Preparation of Datasets

A total of 170 images were obtained from online sources, representing different stages of fruit ripening under mid-ripen, under-ripe, and over-ripe conditions, with each image having 960*540 pixels. The images were downloaded and resized to fit the input size specifications of different models, with MobileNet, VGG16, and Inception V3 models requiring images of size 224*224. Pre-processing of the cropped images was performed to remove noise and undesirable illumination effects, while adjusting the intensity values to fill the entire intensity range of [0, 255] without affecting the RGB colour values. The dataset was stored in three separate folders representing the three ripening stages: over-ripe, under-ripe, and mid-ripe, and was further divided into three groups for training, validation, and testing. The training process utilised 80each, using a ratio of 80:10:10. A Python code was utilised for random selection of images for training, validation, and testing.

TABLE I
DATASETS

class	Count
underripen	49
midripen	88
overripen	33

B. Methodology

The study evaluated the accuracy of three predefined models, MobileNet, VGG16 and Inception V3, for ripening identification using different parameters such as minimum and maximum batch sizes, number of classes, epochs, and validation frequency. Each model was tested for various epochs and batch sizes, with images randomly selected for each epoch. The models' architectures were already defined, including the number of convolutional layers, activation function, and hidden units for each layer, and remained fixed throughout the experimentation process. Convolutional Neural Networks (CNNs) are a type of artificial neural network that utilises convolutional layers to filter input data and extract useful information. In CNNs, convolution involves combining the input data, also known as a feature map, with a convolution kernel or filter to produce a transformed feature map. These filters are modified based on learned parameters to extract the most relevant features for a specific task. The CNN architecture automatically adjusts the filters to find the optimal features for the task at hand. For example, when confronted with a general object recognition task, a CNN would filter information about the shape of an object. However, when faced with a bird recognition task, the CNN would extract the bird's colour, since different types of birds are more likely to differ in colour than in shape. This demonstrates the CNN's ability

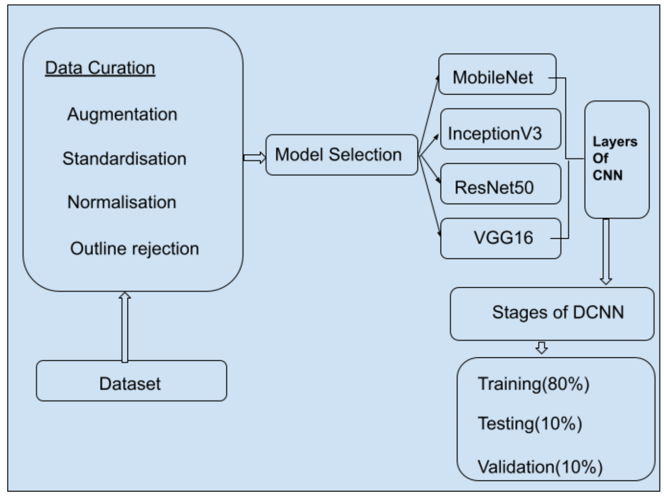
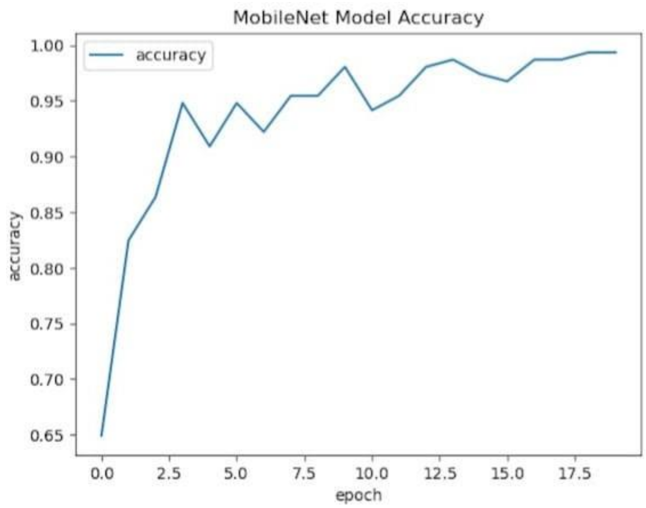


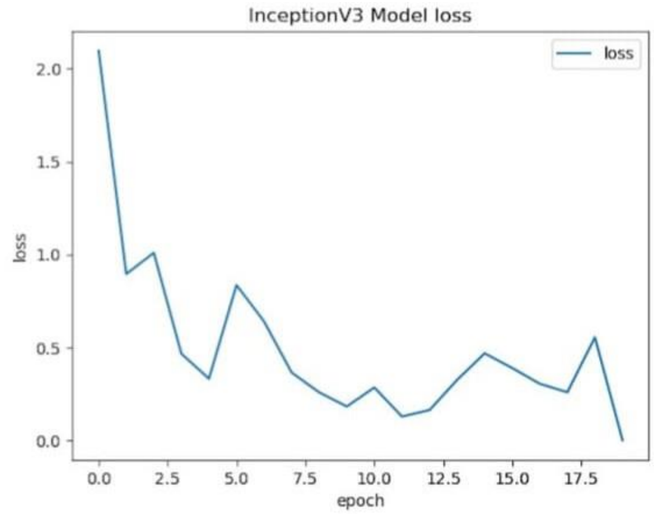
Fig. 1. Schematic Diagram

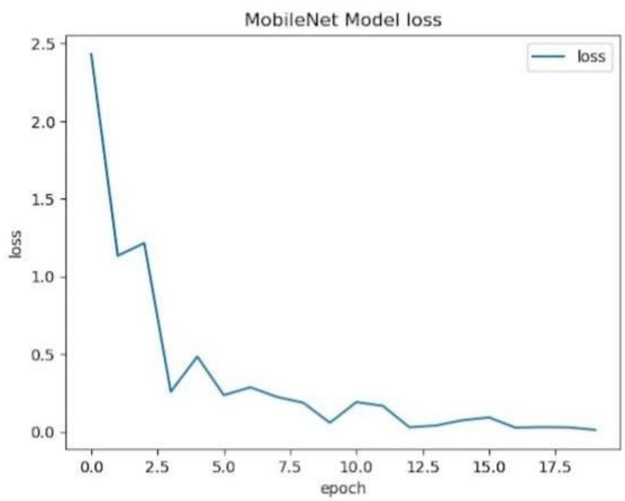
to understand that different classes of objects have different characteristic features, and that the relevant features may vary depending on the task. Convolutional Neural Networks (CNNs) use pooling layers to enhance their object detection capabilities, even for objects placed in unusual positions. These pooling layers allow for limited translation and rotation invariance, which helps to reduce memory consumption and allows for the use of more convolutional layers. Normalisation layers are also employed in CNNs to normalise over local input regions, which moves all inputs in a layer towards a mean of zero and a variance of one. This helps to ensure that the network is robust to variations in the input data. In addition to normalisation layers, other regularisation techniques can also be used to improve the performance of CNNs. For example, batch normalisation involves normalising across the activations for the entire batch, while dropout involves ignoring randomly chosen neurons during the training process. These techniques help to prevent overfitting and improve the generalisation performance of the network. The models were trained using the training images and their corresponding true labels. The aim was to reduce the losses between the actual and estimated results and minimise errors. The input and output vectors of the model were initially fitted on a training dataset, and the result was compared with the target for each input vector in the dataset. The model parameters were adjusted based on the comparison results and the learning algorithm used.

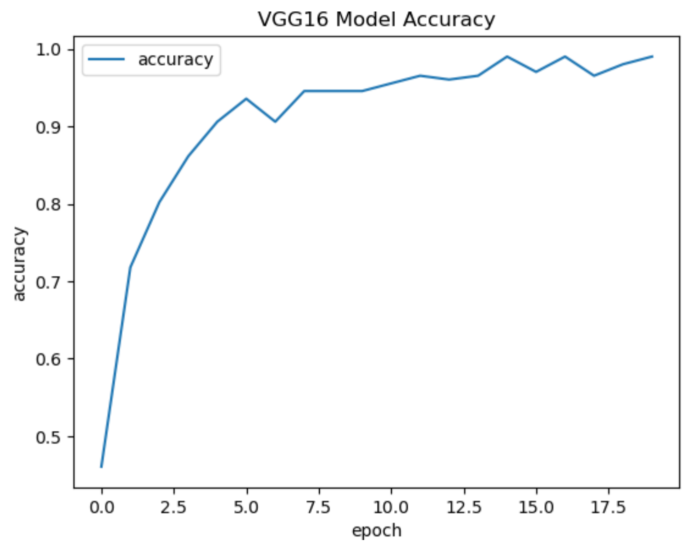
TABLE II
EXPERIMENTAL RESULTS

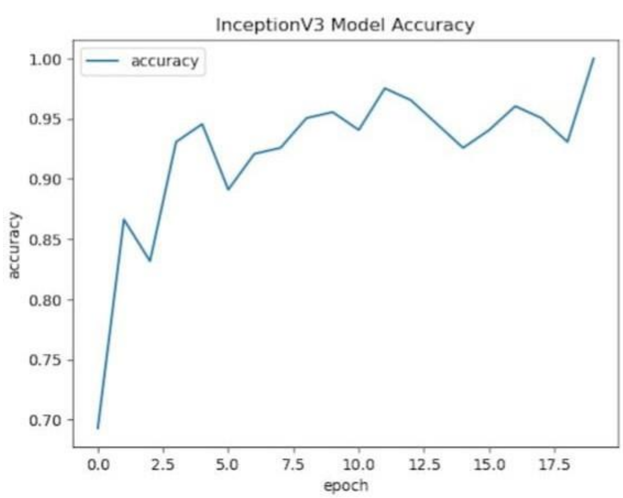
Model	Epochs	Batchsize	Accuracy
MobileNet	20	16	99.35
InceptionV3	20	16	97.5
VGG16	20	16	99

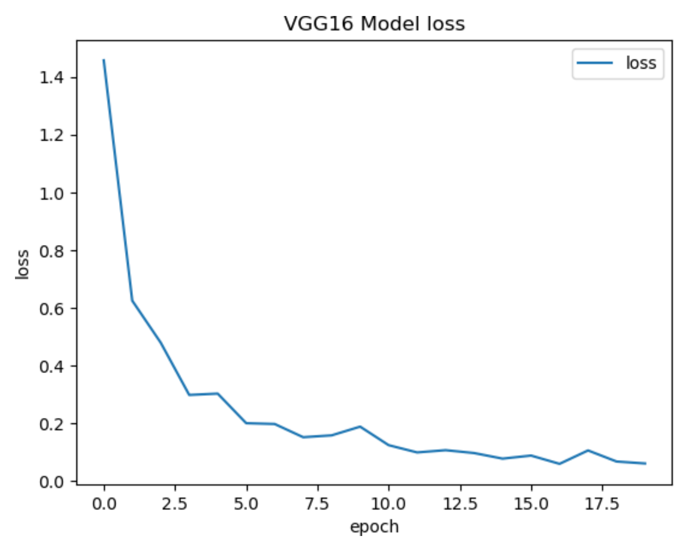

MobileNet Model Accuracy
accuracy
accuracy
epoch
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5


InceptionV3 Model loss
loss
loss
epoch
2.0
1.5
1.0
0.5
0.0
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5


MobileNet Model loss
loss
loss
epoch
2.5
2.0
1.5
1.0
0.5
0.0
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5


VGG16 Model Accuracy
accuracy
accuracy
epoch
1.0
0.9
0.8
0.7
0.6
0.5
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5


InceptionV3 Model Accuracy
accuracy
accuracy
epoch
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5


VGG16 Model loss
loss
loss
epoch
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 04 | April - 2023 Impact Factor: 8.176 ISSN: 2582-3930

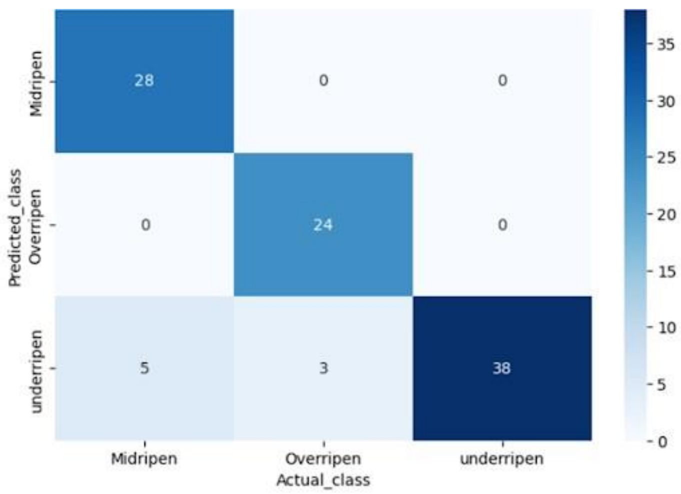


Fig. 2. Confusion matrix for MobileNet

C. Cross Validating the Models

To ensure the model's generalizability, a cross-validation dataset was employed to evaluate the model's performance while tuning hyperparameters. The incorporation of cross-validation data in the model configuration ensured a more skill-based evaluation. Cross-validation was crucial in preventing overfitting when adjusting classification parameters. An error function was used to assess the performance of different networks on a new set of data that was independent of the training data. The confusion matrix is a valuable tool for evaluating a classification system with regards to different categories.

It compares the output class and target class, where the former is the prediction result of the deep learning model, and the latter is the image's true or actual class. The classification model's performance was evaluated on a testing dataset, and the confusion matrix was used to interpret the results. The confusion matrix is a two-dimensional matrix with two rows and two columns, where the rows represent the predicted class's outcome, and the columns represent the actual class. A confusion matrix was developed using a cross-validation dataset to describe the classification model's performance. The accuracy (A), error or misclassification (E), precision (P), and sensitivity (S) were derived from the confusion matrix using the following expressions:

Accuracy=(TP+TN)/(TP+TN+FP+FN)
Error=(FN+FP)/(TP+TN+FP+FN)
Precision=TP/(TP+FP)
Recall=TP/(TP+FN)

TABLE III
RESULTS

class	Error	Precision	Accuracy	Recall
underripen	8	83	92	90
midripen	5	84	94	92
overripen	3	88	97	94

D. Testing the model

The final step in the development process was to test the models on 60 individual banana images to evaluate their efficacy in accurately classifying the images. Testing the models ensured their performance and ability to generalise to new, unseen data. The complete process of deep learning model development is illustrated in Figure 1.

CONCLUSION

THE BEST METHOD FOR FRUIT RIPENESS DETECTION

The present study has explored and compared the performance of three deep learning models, namely MobileNet, VGG16, and Inception V3, for the classification of ripeness stages of bananas. The results indicate that the MobileNet model outperforms the other two models in terms of accuracy and consistency. This finding highlights the potential of the MobileNet deep CNN model for accurately identifying the ripeness stages of bananas. Such a model can be employed in real-time image-based systems to help farmers identify the ripening stage of bananas and make informed decisions about harvesting.

REFERENCES

[1] M. Pak and S. Kim, "A review of deep learning in image recognition," in Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), pp. 1–3, IEEE, Kuta Bali, Indonesia, Aug. 2017.

[2] H. Zhai, "Research on image recognition based on deep learning technology," in Proceedings of the 2016 4th International Conference on Advanced Materials and Information Technology Processing (AMITP 2016), Guilin, China, September 2016.

[3] L. Jiang, Y. Fan, Q. Sheng, X. Feng, and W. Wang, "Research on path guidance of logistics transport vehicles based on image recognition and image processing in port area," EURASIP Journal on Image and Video Processing, vol. 2018, no. 1, 2018.

[4] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, and M. Woźniak, "Automatic classification of fruit defects based on Co-occurrence matrix and neural networks," Annals of Computer Science and Information Systems, vol. 5, pp.

[5] F. Liu, L. Snetkov, and D. Lima, "Summary on fruit identification methods: a literature review," in Proceedings of the 2017 3rd International Conference on Economics, Social Science, Arts, Education and Management Engineering (ESSAEME 2017), Atlantic press, AV Amsterdam, Netherlands, July 2017.

[6] J. Feng, L. Zeng, and L. He, "Apple fruit recognition algorithm based on multi-spectral dynamic image analysis," Sensors, vol. 19, no. 4, p. 949, 2019.

[7] Horea Mures¸an and Mihai Oltean, "Fruit recognition from images using deep learning," Dec 2, 2017.