Article

An Explainable and Lightweight Deep Convolutional Neural Network for Quality Detection of Green Coffee Beans

Chih-Hsien Hsia [1,2,*], Yi-Hsuan Lee [2] and Chin-Feng Lai [2,*]

[1] Department of Engineering Science, National Cheng Kung University, Tainan city, Taiwan; chhsia625@gmail.com; cinfon@ieee.org
[2] Department of Computer Science and Information Engineering, National llan University, Yilan county, Taiwan; shawn80324@gmail.com
* Correspondence: chhsia625@gmail.com; cinfon@ieee.org

Abstract: In recent years, the demand for coffee has increased tremendously. During the production process, green coffee beans are traditionally screened manually for defective beans before they are packed into coffee bean packages; however, this method is not only time-consuming but also increases the rate of human error due to fatigue. Therefore, this paper proposed a lightweight deep convolutional neural network (LDCNN) for the quality detection system of green coffee beans, which combined depthwise separable convolution (DSC), squeeze-and-excite block (SE block), skip block, and other frameworks. To avoid the influence of low parameters of the lightweight model caused by the model training process, rectified Adam (RA), lookahead (LA), and gradient centralization (GC) were included to improve efficiency; the model was also put into the embedded system. Finally, the local interpretable model-agnostic explanations (LIME) model was employed to explain the predictions of the model. The experimental results indicated that the accuracy rate of the model could reach up to 98.38% and the F1 score could be as high as 98.24% when detecting the quality of green coffee beans. Hence, it can obtain higher accuracy, lower computing time, and lower parameters. Moreover, the interpretable model verified that the lightweight model in this work is reliable, providing the basis for screening personnel to understand the judgment through its interpretability, thereby improving the classification and prediction of the model.

Keywords: green coffee bean; lightweight framework; deep convolutional neural network; explainable model; random optimization

1. Introduction

Coffee is a brewed beverage obtained by extracting in water the soluble components of the roasted pits of green coffee beans, which is not only flavorful but contains various levels of antioxidants and nutrients. Inoue *et al*. [1] conducted a prospective study with 90,452 subjects (including 43,109 men and 47,343 women) and found that hepatitis B and C virus-positive patients who consume 1 to 2 cups of coffee per day had a lower cirrhosis risk (relative risk = 50%) compared with those who almost never consumed coffee. Also, hepatitis B and C virus-positive patients who consume 4 cups of coffee per day had lower cirrhosis and hepatitis risk (relative risk = 25%) compared with those who almost never consumed coffee. According to Loftfield *et al*. [2], drinking coffee is beneficial for heart health as caffeine can improve the cell processes in the blood vessels, especially the proteins in the cells of the elderly. When consumed in moderation, coffee can help prevent diseases like liver cancer, heart disease, dementia, or stroke. Hence, coffee has been one of the most widely consumed beverages in the world.

After collection, coffee beans should be handled quickly; otherwise, they will smell bad. Mature fruits will sink to the bottom of the tank after sun exposure and washing, while immature or broken beans tend to float on top. The green coffee beans are then picked out of the coffee fruit. Regardless of previous processing and selection, to remove defective beans, green coffee beans need to be manually selected; the selection process is a difficult one to the brewing, mildew, brokerage, or worm damage during the peeling of beans. If defective beans are present before baking, chemical reactions may occur due to uneven heating, resulting in chemical toxins which can be harmful to health [3]. Therefore, to reduce labor costs, improve the quality of coffee, and increase profits, artificial intelligence (AI) has been introduced to pick out defective beans. The accurate and fast selection of defective beans using AI is an important automatic detection technology.

Many different algorithms have been applied to detect the quality of green coffee beans. For example, Santos *et al*. [20] employed spectroscopy to analyze the correlation between the quality of green coffee beans and near-infrared rays with partial least square regression (PLS) model to predict defective coffee beans. However, the cost of detection instruments was too expensive; thus, it would be difficult to mass-produce. Oliveira *et al*. [21] put green coffee beans into the dark box without external light, captured the image through high-pixel RGB cameras, and converted RGB space colors to CIELAB. Later, they used a Bayesian classifier to improve the coffee quality prediction. However, the test required a special environment, and the instrument cost was expensive. Arboleda *et al*. [22] extracted coffee bean features such as area of the bean, perimeter, equivalent diameter, and percentage of roundness, and employed an artificial neural network (ANN) and K nearest neighbor (KNN) to automatically categorize the coffee beans. Using ANN, the classification scores achieved 96.66%, while the classification scores using KNN were at 84.12% [23]. Image processing techniques were used to control the coffee bean quality by extracting RGB color components based on 105 images of green coffee beans and 75 images of black beans with high accuracy. However, only 180 green coffee beans were tested [22-23]; this may cause poor stability during mass production due to the limited amount during testing. Hence, scholars began to improve stability through deep learning (DL). Pinto *et al*. [24] developed a convolutional neural network (CNN) that classifies 13,000 green coffee beans images into six defect types. They sorted defective beans from 72.5% (broken bean) to 98.7% (black bean) accuracies, and the difference in the accuracy rate of detection in defective beans led to the poor generalization of the model. Wang *et al*. [18] used the lightweight model with knowledge distillation (KD) and improved the accuracy of the training method to 91%, with the model parameters at only 256, 779. Huang *et al*. [25] extracted 1,000 pleasant coffee beans and 1,000 defective coffee beans, and used image processing and data augmentation to deal with the data. Next, they applied YoloV3 to divide good and bad beans which had a recognition rate of 94.63%. Yang *et al*. [19] employed the CNN model based on KD, spatial-wise attention module (SAM), and SpinalNet [26] to achieve an accuracy rate of 96.54% on the F1 score. The recent progress in AI technology enables the labeling of data and the use of neural network design to allow the machine to automatically learn the data, and the neural network to predict and make decisions based on the characteristics of the learned data. Although the deep convolutional neural network (DCNN) can accurately classify the images, it cannot be easily applied in embedded systems because of its large Giga floating-point operations per second (GFLOPS).

Generally, the accuracy or evaluation indicators are very important for model efficiency when using DCNN for prediction and decision-making. Furthermore, explainable AI (XAI) will be lacking when DL technology is utilized. Hence, when using DL in models with high accuracy, the complexity may be high, while the correlation and hidden information cannot be explained through accuracy. Thus, scholars usually have difficulty understanding the correlation between input and output and how the model achieves its purpose. As a result, the uncertainty remains if people blindly believe in the model's prediction. To solve this problem, XAI should be introduced to determine whether the prediction and evaluation of the decision-making of the model based on certain characteristics are reasonable. Only in this way can the reliability of the model and quality detection be ensured in the future.

To deal with the above issues, this paper proposed a lightweight deep convolutional neural network (LDCNN) to detect the quality of green coffee beans. First, the features of defective coffee beans through RGB images were extracted and the model was employed to classify the beans. Next, the rectified Adam (RA), lookahead (LA), and gradient centralization (GC) were utilized to train the optimization methods to improve the accuracy of the model, enabling the model to operate in embedded systems.

2. Fundamental Knowledge

2.1. ResNet

To address the issue of gradient descent in DCNN, ResNet [4] introduced a residual learning framework, which was named a building block and is shown in Figure 1. By doing so, the training of deep networks was much easier while the accuracy and rate of the model remained high.

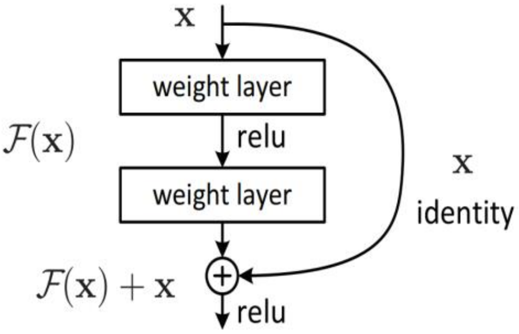

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 1. Structure of building block [4].

2.2. MobileNetV3

MobileNetV3 [5] was put forward by Google in 2019 to deal with lightweight, whose bottleneck block can make the model lightweight and reduce GFLOPS using depthwise separable convolution (DSC), inverted residual block (IRB), and squeeze-and-excite block (SE block), as illustrated in Fig. 2.

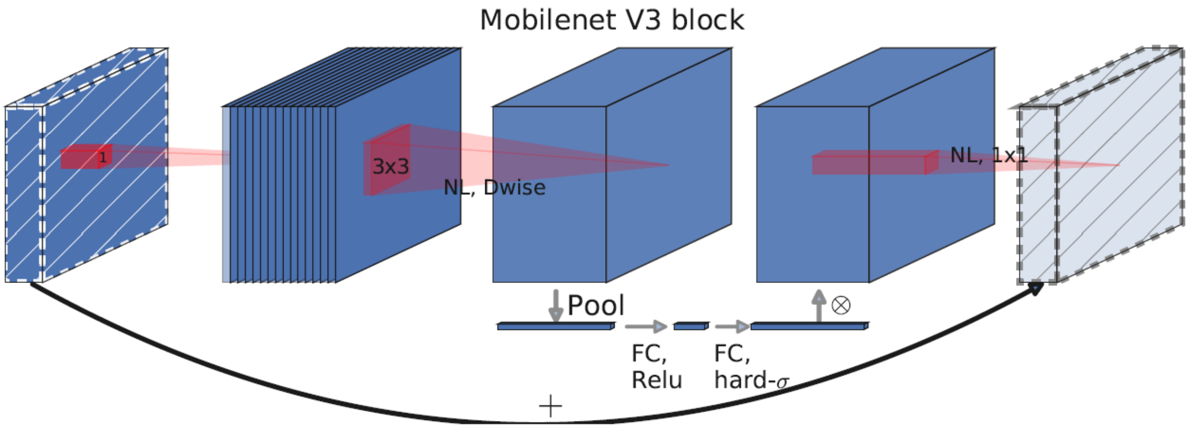
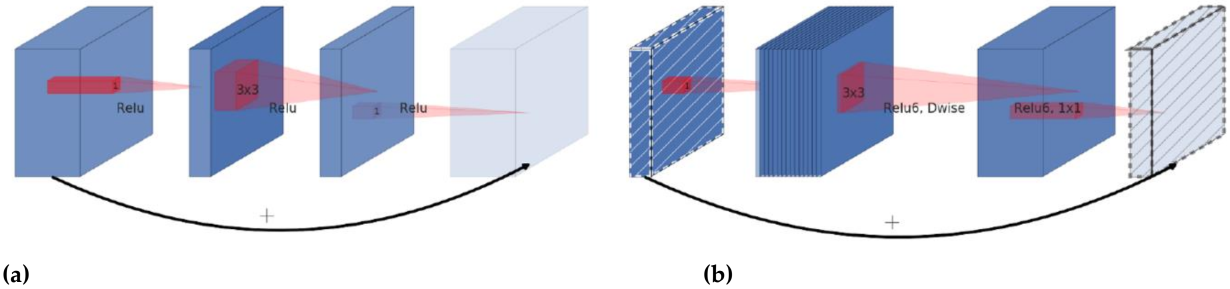


Figure 2. Structure of MobilenetV3 [5].

DSC is comprised of depthwise convolution (DC) and pointwise convolution (PC): the former aims to compress the channel of images, while the latter improves or reduces the dimensions of images of 1×1 pixel. Compared with the ordinary convolutional layer (CL), DSC can largely reduce GFLOPS without influencing performance; thus, it is now a widely used framework in lightweight models. The residual block of ResNet employs the general convolution layer to enhance dimensions and extract characteristics, and connects two layers of large dimensions with longer GFLOPS. However, IRB connects two layers of small dimensions and extracts features of images through PC and DC, which can deliver a high accuracy while reducing GFLOPS (see Fig. 3).



(a) (b)

Figure 3. Comparison of two frameworks: (a) Residual block, (b) IRB.

As a lightweight attention module, the SE block, as shown in Fig. 4, involves the relations among each channel and makes the model learn characteristics through the loss function. Thus, the weights of effective features increase while the weights of unimportant features decrease, allowing the model to learn varied importance levels of channel features to improve the accuracy.

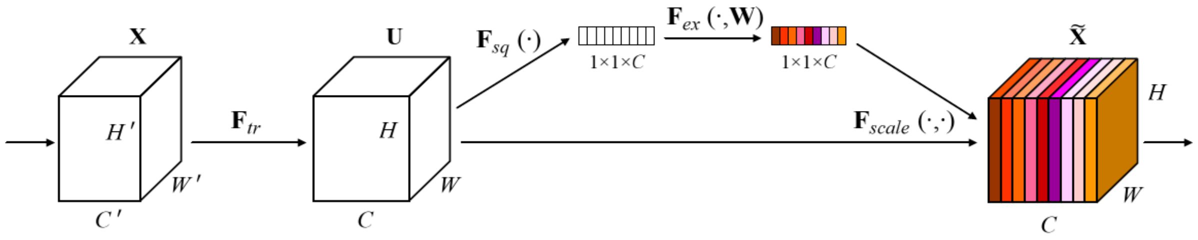


Figure 4. Structure of SE block [7].

2.3. Rectified Adam (RA)

RA [8] employs both Adam [9] optimizer and warm-up [10] to optimize the model. Unlike the general experimental algorithm that uses fixed intervals to reduce the learning rate, warm-up initially uses a rather low learning rate training model. Later, during the mechanism of warm-up, the learning rate increases gradually. When warm-up ends, the general process continues (see Fig. 5). As previously suggested [9], since the initial value of model training is generated randomly and the model has no comprehension of the data, the data loss will be large in the first epoch. Moreover, a large gradient causes large weight changes every time. Hence, it is easy to correct the feature distribution of data at the beginning of model training, while the overfitting may occur frequently, or correction may be achieved only after multiple training. However, the warm-up can address all those issues.

During the training of the model, RA can automatically correct the learning rate based on the degree of variation of gradient and the number of samples. Therefore, with the quick convergence of Adam, the convergence can avoid the local minimum and reach the same results as the stochastic gradient descent (SGD) to make the training stable.

doi:10.20944/preprints202209.0190.v1

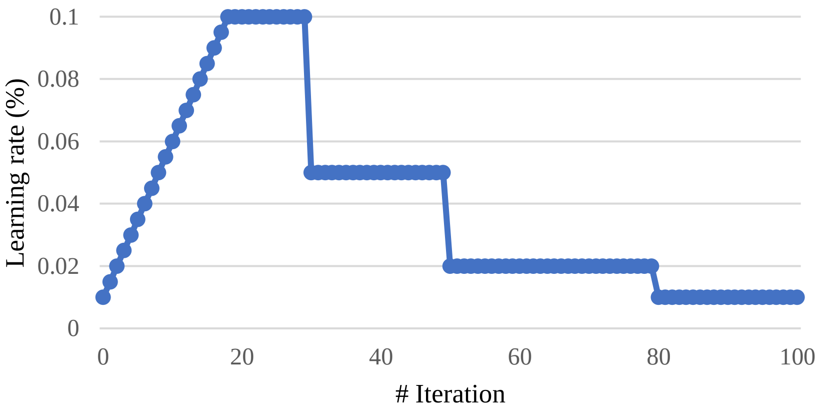


Figure 5. Learning rate changes in warm up.

2.4. Lookahead (LA)

LA [11] initially generates fast and slow weights for the model separately and then updates fast weights during the training. The slow weights are updated towards the last fast weights after k batches. After each slow weight is updated, the fast weights are reset to the current slow weights value. Hence, the model's weight is not easy to converge at the local minimum. Figure 6 shows the contour map of the gradients. The LA optimization method can help the model continue to converge to the local minimum.

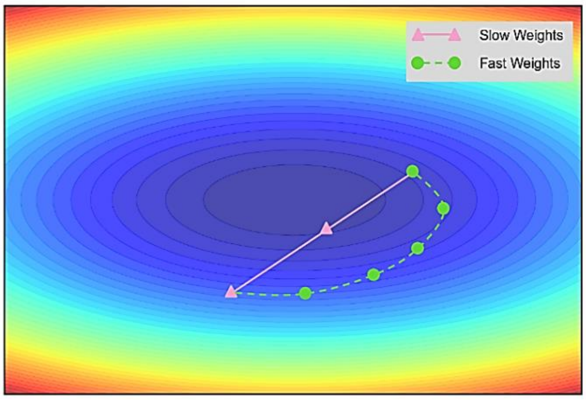


Figure 6. Principle of LA optimization.

2.5. Gradient Centralization (GC)

Gradient descent is of great importance to effectively and efficiently train a DCNN, which can directly affect the model's convergence speed and prediction accuracy. When the model becomes larger, gradient descent becomes more difficult, causing issues like gradient vanishing, gradient exploding, or non-convergence to affect the accuracy. To deal with this, GC [12] was presented to make the gradient descent smoother and to normalize the gradient before backward propagation, as described in (1). GC operates directly on gradients by centralizing each column of convolutional layers and fully connected layers to have zero mean; the gradient is then subtracted from its mean value and sent back to continue the training of the model, as illustrated in Fig. 7. Hence, the abnormal data will not affect the training process.

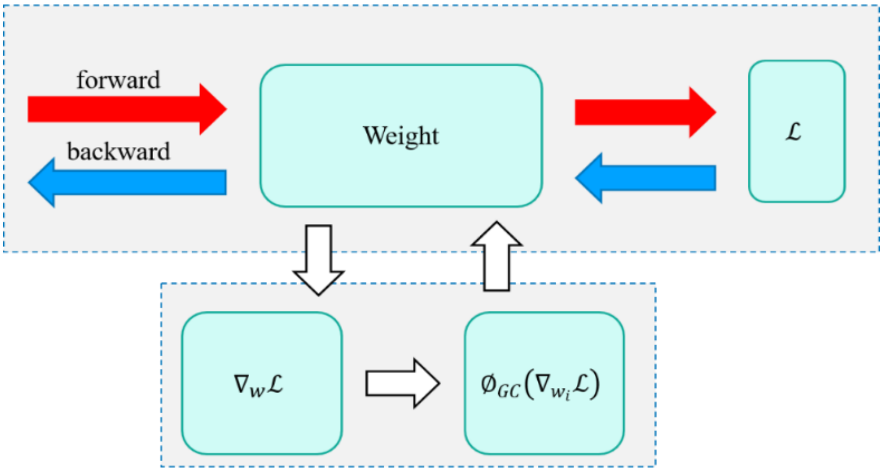


Figure 7. Sketch map for GC.

$$\emptyset_{GC}(\nabla_{w_i}\mathcal{L}) = \nabla_{w_i}\mathcal{L} - \mu\nabla_{w_i}\mathcal{L} \quad (1)$$

where *W*

where W_i denotes the weight matrix of CL or FCL. The gradient $\nabla_{w_i}\mathcal{L}$ is obtained through backward propagation of $\mathcal{L}$. After the calculation of mean value μ, average $\mu\nabla_{w_i}\mathcal{L}$ can be realized and $\emptyset_{GC}(\nabla_{w_i}\mathcal{L})$ is gained through GC.

2.6. *Local Interpretable Model-agnostic Explanations (LIME)*

LIME 0, a local interpretable regression model, can provide the local area of the sample to find a simple and interpretable model when faced with complicated models. First, the images are separated into multiple sub-blocks; next, sub-blocks are randomly disturbed and their predictions of complicated models are observed; later, after obtaining many samples of features, the local similarity is defined through regression and the XAI is trained to explain and visualize complicated models as suggested in (2).

$$explanatoion(x) = \frac{argmin\{\mathcal{L}_{(f,g,\pi_x)}+\Omega(g)\}}{g\in G} \quad (2)$$

Initially, the sample π_x is divided into several sub-blocks; then, sub-blocks are randomly disturbed to observe the predictions of the model f with high reviews. Next, a simple linear regression model g is trained to explain the predictions; and $\mathcal{L}(f,g,\pi_x)$ is used to measure the differences between the complex model and the interpretation model. The $\Omega(g)$ refers to a measure of complexity to analyze and explain the judgment basis of the complex model for the image.

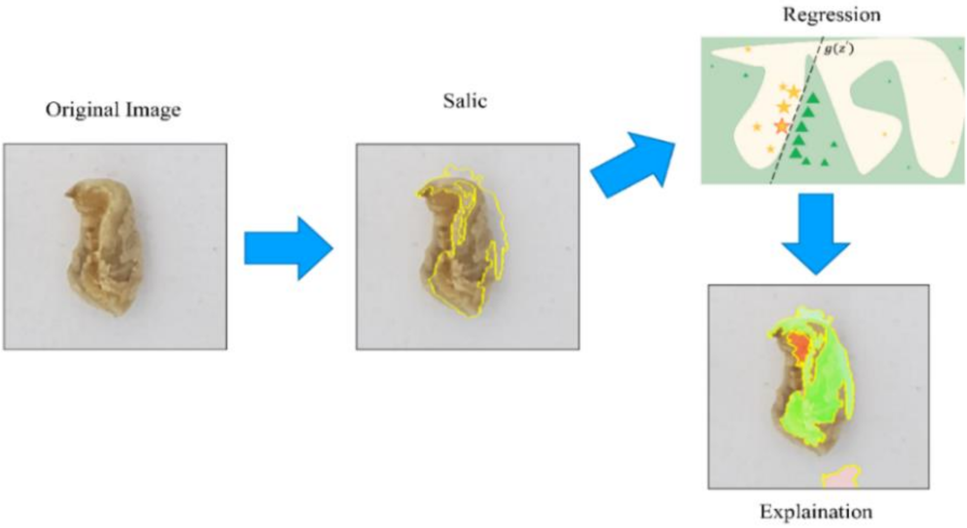


Figure 8. Process of LIME.

3. Proposed Methodology

The proposed methodology and experiment process is shown in Fig. 9. The data set and DA are described in next section. In this section, the image pre-processing and the training process of the proposed model are explained.

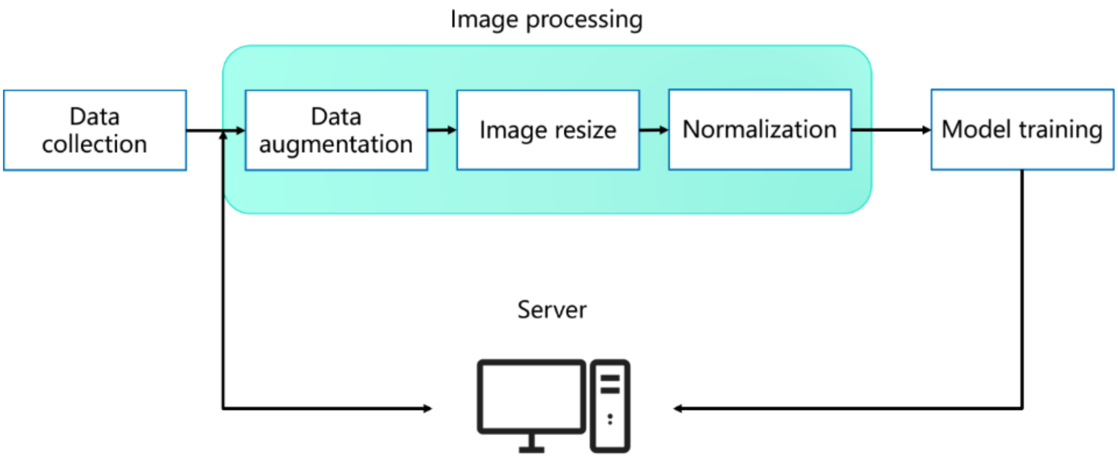


Figure 9. Framework of system.

3.1. Image Preprocessing

The model should be lightweight to extract features of images and reduce noises and the size of input images. Hence, the image pre-processing was carried out before inputting it into DCNN. To ensure that each input image is of the same size during the training and validation when minimizing the image size can reduce the parameters of the model and increase the GFLOPS, the image size was reduced from 400×400×3 to 224×224×3. Using bilinear interpolation, the image of the data set was resized. In this paper, the RGB average value and standard deviation of the parameters set for image normalization (Figure 10) were the average value and standard deviations obtained from millions of images using ImageNet [15]. This is described in (3) and (4).

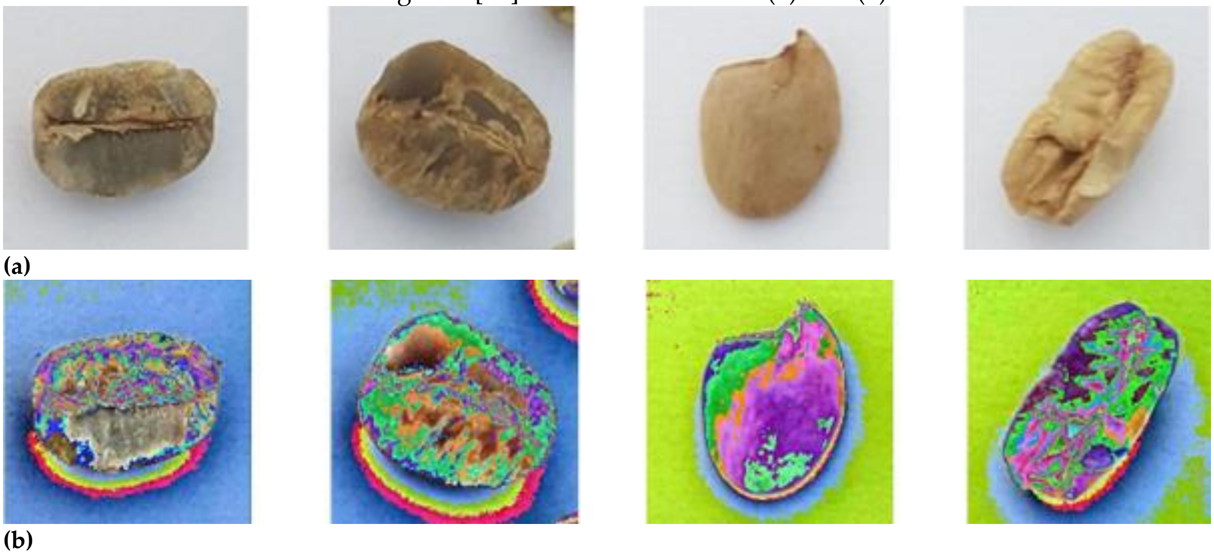


Figure 10. Images of green coffee beans: (a) Before normalization, (b) After normalization.

$$Z = \frac{(x-\mu)}{\sigma} \quad (3)$$

$$\mu_{RGB} = [0.485, 0.456, 0.406] ; \sigma_{RGB} = [0.229, 0.224, 0.225] \quad (4)$$

3.2. Lightweight Deep Convolutional Neural Network (LDCNN)

The LDCNN proposed in this paper took SE block as the backbone and used DSC to extract features of input images and adjust output dimensions. Figure 11 demonstrates the structure of the SE block. The image features went through the PC first to improve the dimensions, and pointwise features were extracted in the space. Next, image features between each pixel channel were extracted by DC and compressed using global average pooling (GAP). Then, the features were multiplied by input images through the ReLU activation function to expand. Therefore, high weights were enhanced while other weights were diminished, and the dimensions of output images were finally changed.

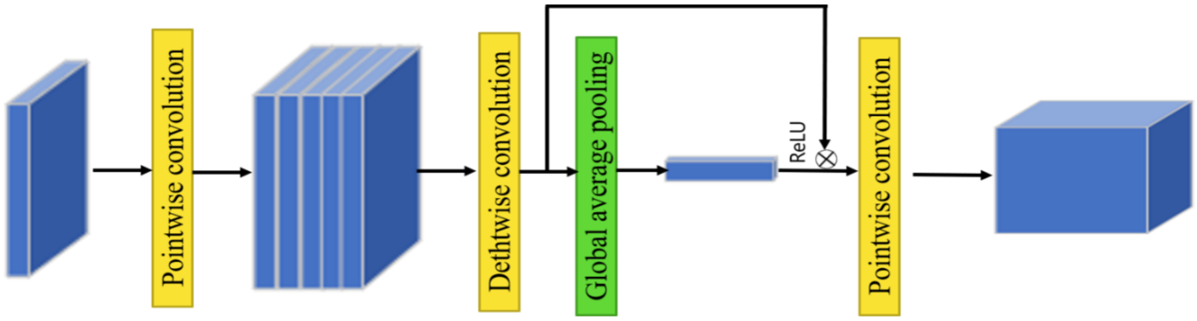


Figure 11. Structure of SE block.

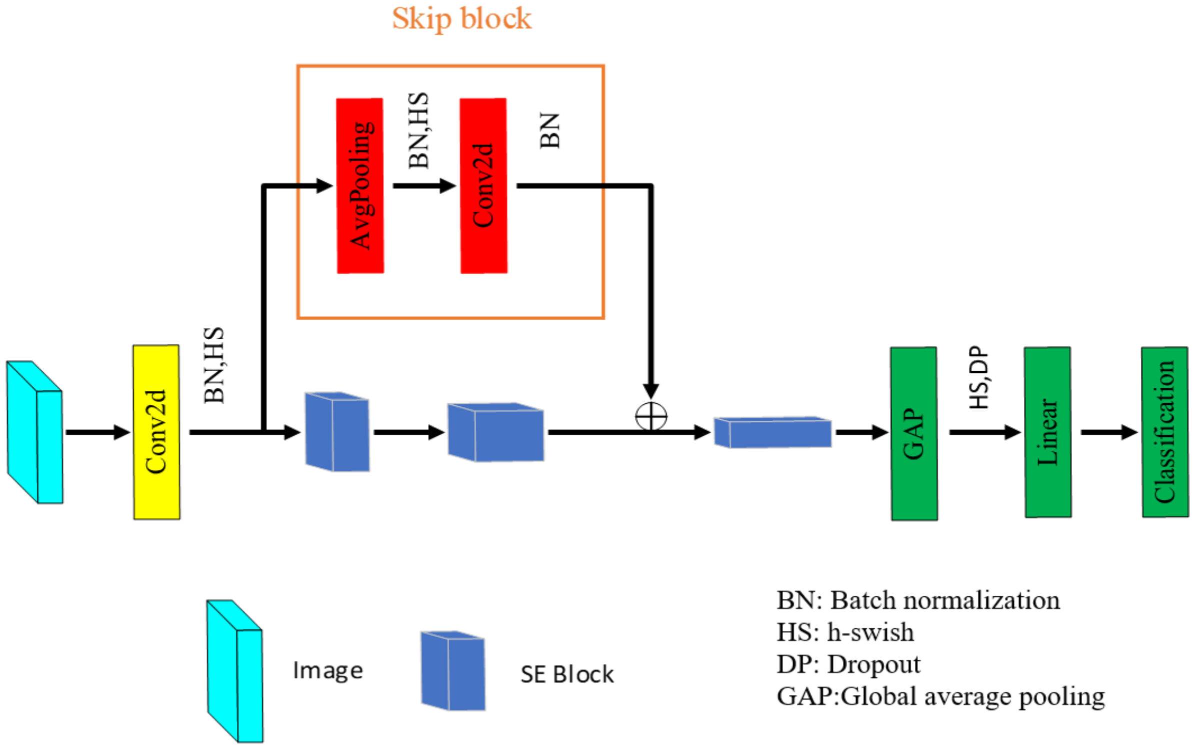


Figure 12. Proposed LDCNN Framework in the study.

The LDCNN model in this study used the SE block as the backbone. Initially, the input image improved the dimensions of images through a single CL. Later, after undergoing SE block twice, tiny features were extracted. Then, a skip block was employed to extract features through a skip connection. For instance, residual learning of ResNet can help the model avoid the difficulty of training deep networks due to weight degradation. Therefore, a skip block was added in this paper to improve the model's accuracy. SE block can extract channel-wise features of image features. However, after feature extraction, the features mainly focus on the correlation between image channels and channels, while feature extraction of global images is lacking. Hence, the model in this study additionally extracted features of the average pooling layer (APL) and CL as the skip block of the model. Two different features were added for classification, and GAP was used to connect CL and FCL on the tail structure of the model. Feature extraction was performed on global images, and the results were then classified.

Table 1 shows the model structure. As this study targets a lightweight model, the depth of the model was reduced, and a three-layer bottleneck block and ReLU activation function were adopted to minimize the GFLOPS and the parameters of the model. Furthermore, to maintain the model's accuracy, the SE layer was used, and the dimensions of the images were increased. H-swish (HS) activation function was adopted at both ends of the model structure to reduce GFLOPS and keep the accuracy of the model as opposed to ReLU. A 28×28 APL was used at the end to reduce the GFLOPS. By combining those models, this paper believes that high accuracy and a lightweight model could be realized.

Table 1. Proposed model structure in the study.

Input Size	Operator	Exp size	Out	Stride	HS
224×224×3	Conv2d, 3×3	—	16	2	✓
112×112×16	Bneck, 3×3	64	32	2	—
56×56×32	Bneck, 3×3	128	48	2	—
28×28×48	Bneck, 3×3	128	48	1	—
28×28×48	Pool, 28×28	—	48	1	—
1×1×48	Linear	—	1024	—	✓
1×1×1024	Dropout, 0.2	—	1024	—	—
1×1×1024	Linear	—	2	—	—

To evaluate the efficiency of the model, five-fold cross-validation was adopted in the experiment to train and evaluate the model. Dataset was randomly divided into five groups, four of which contained 3,701 images, and the remaining one included 3,700 images. During the training process, one dataset was chosen while the others were used for training. After performing the experiment five times, the average evaluation results were calculated. Due to the limited parameters of the lightweight model and small GFLOPS, the model failed to extract complete features, which may have resulted in overfitting. To solve those issues, RA, LA, and GC were introduced in this lightweight model to optimize the learning rate strategy, weight, optimizer, and gradient. Finally, to evaluate the efficiency of LDCNN and predict reliability, except for calculating evaluation indicators, model size, parameters, GFLOPS, evaluation time, and comparing state of the art, this model was put into embedded systems. Moreover, this work adopted LIME to explain the predictions of LDCNN in model reliability.

4. Experimental Result

The green coffee bean dataset provided by the small optical sorter [14] included 4,626 images that are 400×400 pixels. It consisted of 2,149 good images and 2,477 bad images; sample images, are shown in Fig. 13. The author collected the images of green coffee beans through high-speed cameras and conveyor belts and reduced the brightness of the shooting environment to minimize the shadows and centralize each image. Figures 13 (a) and (b) are good and bad images respectively, while Fig. 13 (c) and (d) are images with adjusted brightness.

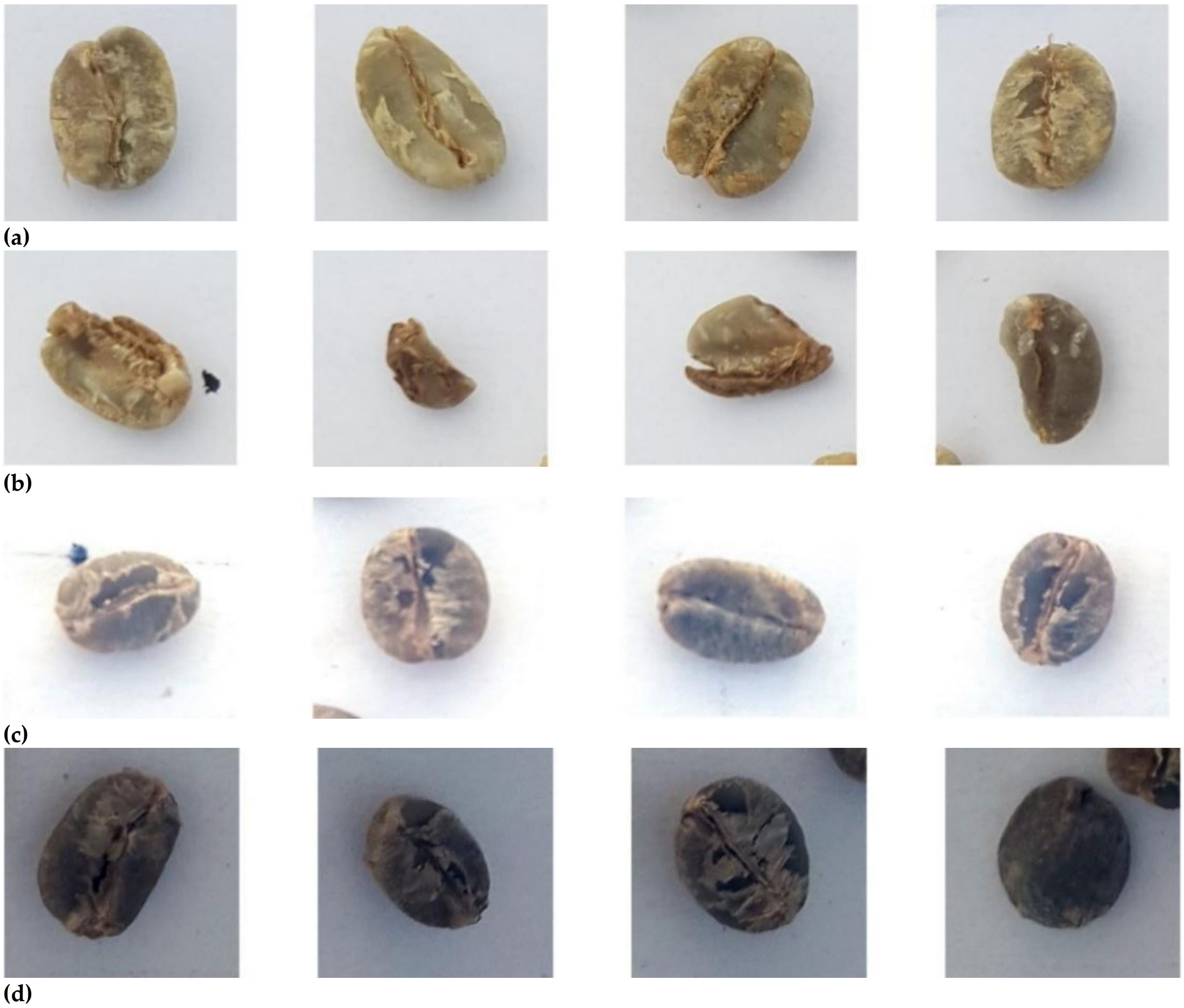


Figure 13. Green coffee bean dataset: small optical sorter: (a) Good, (b) Bad, (c) Increased brightness, (d) Reduced brightness.

To achieve DA (including horizontal and vertical turning and 180° rotation) without any changes in shape, color, and background of images, the number of images was expanded from 4,626 to 18,504 as the training dataset of this study. The image input model after DA helped improve the accuracy and generalization of the model.

4.1. The Influence of Image Normalization on Model

Image normalization means scaling the values of original images within an interval to extract the features and avoid the influence of abnormal values on the training results of the model. In this paper, Adam optimizer was used, learning rate was set at 0.001, batch size was 16, and 100 epochs were trained. Using the same validation method, we figured out that image normalization had significantly improved the accuracy of LDCNN as shown in Table 2.

Table 2. The influence of normalization on LDCNN.

Normalization	Accuracy	Precision	Recall	F1-Score
Without	89.79%	84.86%	94.00%	87.78%
With	**96.84%**	**97.06%**	**96.21%**	**96.50%**

4.2. Ablation Study of Training and Optimization in the Model

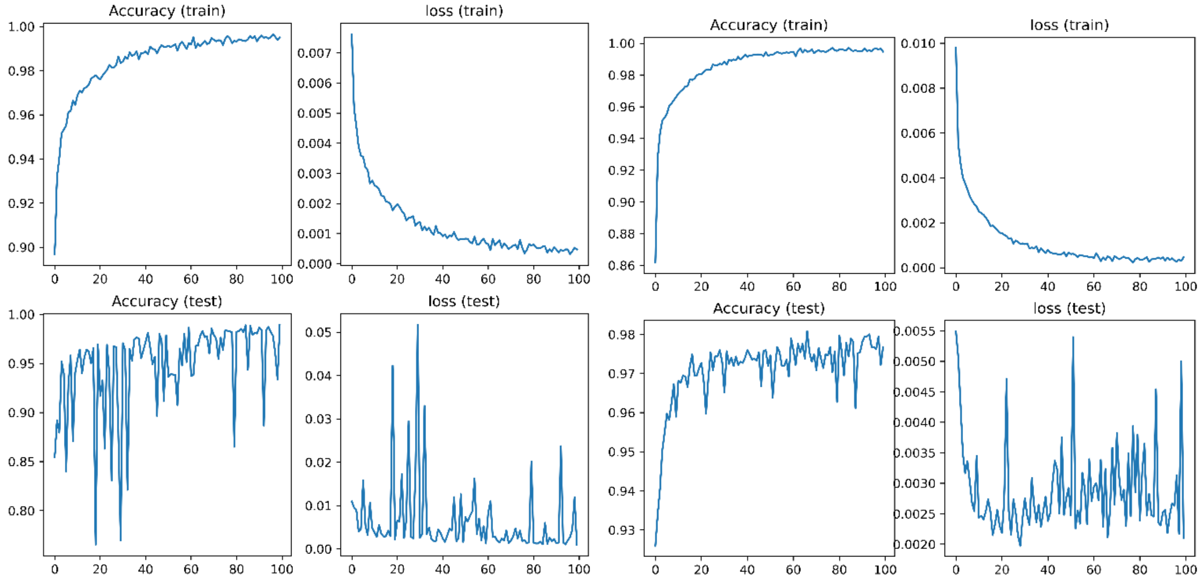
Cross-validation was employed to obtain evaluation indicators using the training dataset. As indicated in Table 3, the accuracy rate of the model was 98.38%, the precision rate was 98.60%, the recall rate was 97.89%, and the F1 score was 98.24%. The parameters were 149,842, the model size was 0.57 MB, the GFLOPS was 0.05, and the computing time was 10.08 ms (see Table 4). The computing time is the average time of image pre-processing and model prediction. As shown, the various evaluation indicators of the model are of satisfactory accuracy while the model is kept lightweight. After, the training and optimization methods were employed to carry out the ablation study. Adam, cross-entropy loss function, and learning rate of 0.001 were used to evaluate and analyze the RA, LA, and GC methods.

Table 3. Using the training and optimization to begin ablation study on LDCNN.

RA	LA	GC	Accuracy	Precision	Recall	F1-Score
			96.84%	97.06%	96.21%	96.50%
✓			96.98%	**99.09%**	94.41%	96.65%
	✓		94.24%	93.78%	94.22%	93.62%
		✓	98.13%	97.53%	**98.47%**	98.00%
✓	✓		98.00%	97.79%	97.92%	97.85%
	✓	✓	97.74%	97.35%	97.82%	97.57%
✓		✓	97.81%	98.22%	97.07%	97.83%
✓	✓	✓	**98.38%**	98.60%	97.89%	**98.24%**

Based on the experiment results, compared to when the Adam optimizer was used, the accuracy rate improved by 0.14%, the precision rate increased to 99.09%, and the recall rate was reduced by 1.80% when RA was employed. Therefore, the generalization ability of the RA model is low. When using GC, evaluation indicators of the model improved compared with those without optimization, indicating that GC effectively improved the detection rate of the model. On the contrary, when using LA, all the evaluation indicators were lower than those without optimization. Moreover, the evaluation indicators of LA-GC were lower than those of GC. While using the three methods simultaneously, the accuracy rate reached 98.38% and F1 score achieved 98.24%. According to the results in Table 3, Adam optimizer is not suitable for training with LA. Only when both RA and LA were used could the model's accuracy be improved. When GC was the only one used, the recall index was the highest. Therefore, the three optimization methods to train the model simultaneously could obtain excellent stability and generalization.

Figures 14 (a) and (b) show the training process of LDCNN and the training process using three optimization and training methods, respectively. No obvious underfitting or overfitting occurred during both training processes. However, the accuracy sometimes dropped suddenly during the training, as the model generated random predictions due to rapid convergence and insufficient parameters. After adding the optimization method, the training stability significantly improved. Hence, the convergence should be stable to improve the accuracy, showing that the model can significantly improve the stability and generalization after combining the training method.



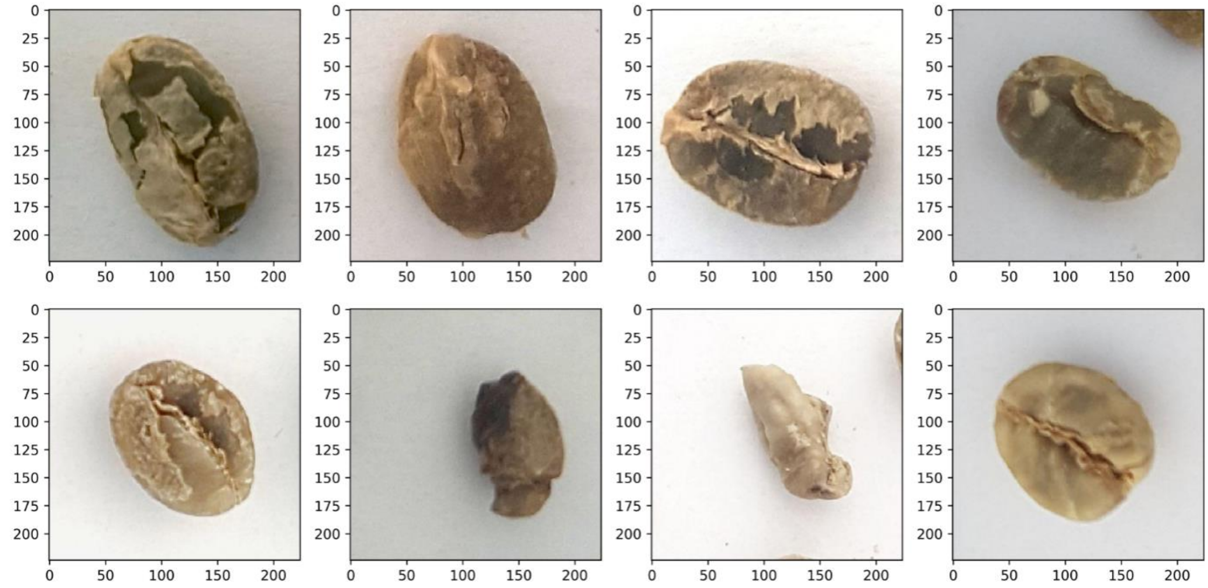
(a) (b)

Figure 14. Training process of lightweight model: (a) Model of this work, (b) Optimization of model of this work.

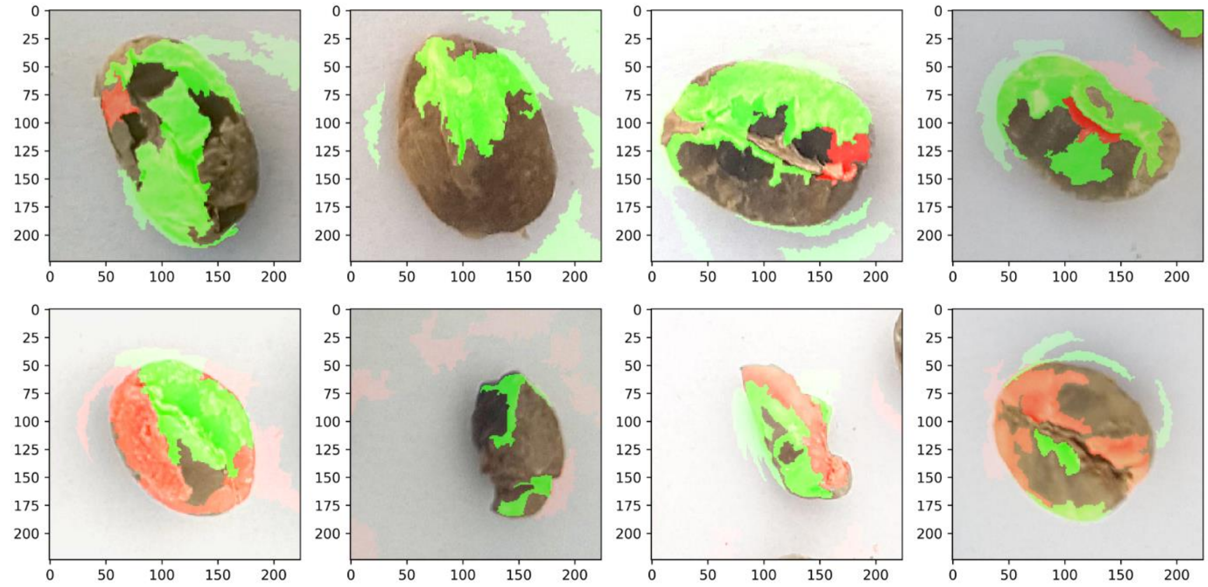
4.3. Evaluation Results of Interpretable Model

Figure 15 (a) represents the original image of green coffee beans, and the upper and lower parts are good and bad coffee beans, respectively. Figures 15 (b) and (c) were the visualization of the interpretation model prediction when the LDCNN optimization model went through LIME. The green block of coffee beans is the area favorable to the prediction results, and the red block is the unfavorable area. Based on Fig. 15 (b), after XAI, the distribution of green areas may also exist in the surroundings of the coffee beans. It shows that the model took the background of coffee beans as the basis for judgment when predicting the quality of beans, and there was no obvious area that could be considered as the reference for judgment. Figure 15 (c), demonstrates that the favorable and unfavorable areas of beans could be revealed by predicting the area of the coffee beans.

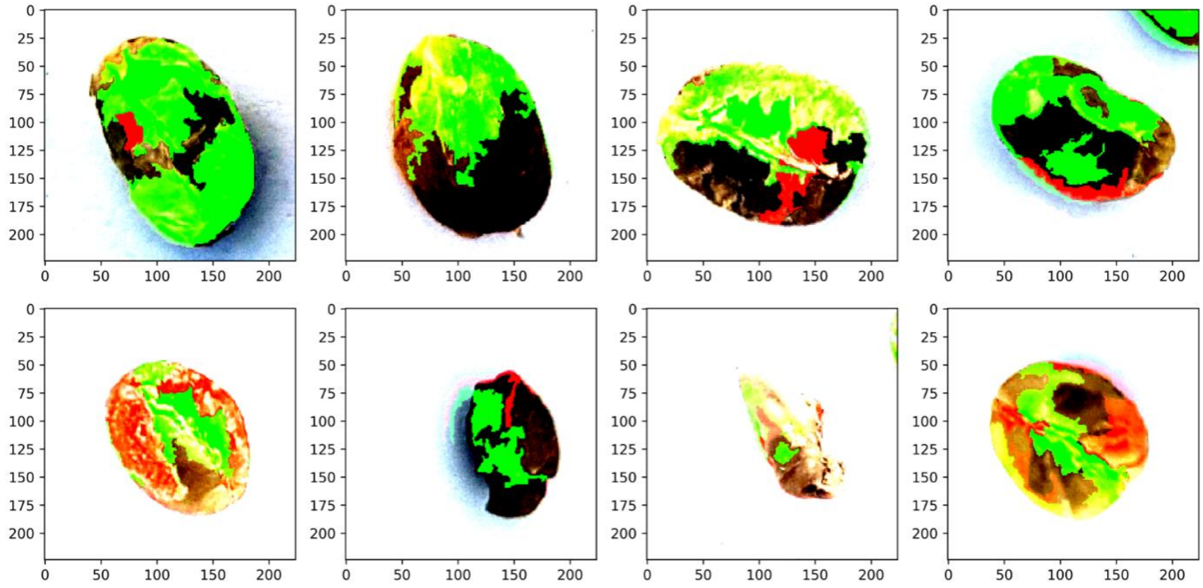
To conclude, the prediction of the LDCNN optimization model is reliable, and the impact of image normalization on model training can also be understood after the image was visualized through LIME. Hence, the training can be optimized, or the abnormal data can be screened out from the data set so the model can get better accuracy during the training process.


0 25 50 75 100 125 150 175 200
0 50 100 150 200

(a)


0 25 50 75 100 125 150 175 200
0 50 100 150 200

(b)



(c)

Figure 15. Results of LIME: (a) Original green coffee beans, (b) Without image normalization, (c) With image normalization.

4.4. Comparison of Model Efficiency & Embedded System

To compare the models and training methods proposed in this paper, this experiment chose famous models, including ResNet [4], MobileNetV3 [5], EfficientNetV2 [16], and ShuffleNetV2 [17], to evaluate and compare with LDCNN. In this experiment, the same data set was used to train each model. Adam optimizer was employed, learning rate was set to 0.001, batch size was 16, and 100 epochs were trained. As suggested in Table 4, in the quality detection of green coffee beans, the accuracy rate was better when LDCNN was used than when other models were utilized. However, the accuracy rate was quite low with the ResNet model compared with the other models; the ResNet18 had an accuracy of only 89.66%, and ResNet 34 and ResNet 50 had decreased accuracy due to overfitting under increased CL.

Table 4. Efficiency comparison of each model.

Models	Accuracy	Precision	Recall	F1-Score	Parameter	Model Size	GFLOPS	Evaluate time
ResNet18[4]	89.66%	89.17%	96.07%	91.26%	11689512	44.59 MB	1.82	19.45 ms
ResNet34[4]	85.93%	98.12%	80.45%	87.56%	21797672	83.15 MB	3.68	31.00 ms
ResNet50[4]	87.85%	87.60%	88.95%	86.28%	25557032	97.49 MB	4.12	57.10 ms
MobileNetV3small[5]	95.97%	95.26%	96.08%	95.65%	2542856	9.7 MB	0.06	10.56 ms
MobileNetV3large[5]	96.69%	97.75%	95.53%	96.49%	5483032	20.92 MB	0.23	20.92 ms
EfficientNetV2-S[16]	95.91%	95.24%	96.01%	95.57%	2278604	8.69 MB	0.149	13.72 ms
ShuffleNetV2[17]	96.04%	96.10%	96.04%	95.23%	22103832	85.1 MB	8.8	73.23 ms
LDCNN	**98.38%**	**98.60%**	**97.89%**	**98.24%**	**149842**	**0.57 MB**	**0.05**	**10.08 ms**

Based on relevant research of public data sets [14], a lightweight model was proposed by Wang *et al*. [18] in which the ResNet18 model was trained as a teacher model through knowledge distillation (KD) to train the lightweight model. The accuracy rate of the lightweight model reached up to 91% with parameters of 256,779. As illustrated in Table 5, Yang *et al*. [19] put forward DSC, SAM, SpinalNet, and KD methods to train the model when the F1 score achieved 96.54%. Compared with LDCNN, the previous model [18] had higher accuracy, and lower parameters since the latter took ResNet18 as the teacher model for training. Nevertheless, the ResNet18 in this experiment was not the optimal model, resulting in a low accuracy rate of the lightweight model. In contrast to Yang *et al*. [19], the precision of LDCNN increased by 2.12%, Recall was raised by 0.36%, and F1-score gained by 1.74%. Finally, the LDCNN was placed on Raspberry Pi 4B to execute the green coffee bean quality detection system (see Table 6). The evaluation time included the model building, image pre-processing, and image estimation time, which showed that LDCNN can achieve the task of real-time detection on the embedded system.

Table 5. Comparison of using public dataset.

Models	Accuracy	Precision	Recall	F1-Score	Parameter
ResNet50 [4]	N/A	84.00%	84.89%	84.21%	N/A
Wang *et al*. [18]	91%	N/A	N/A	N/A	256779
Yang *et al*. [19]	N/A	96.48%	97.53%	96.54%	N/A
MobileNet [27]	N/A	76.98%	81.31%	77.03%	N/A
DenseNet121 [28]	N/A	88.28%	88.28%	88.28%	N/A
Xception [29]	N/A	89.70%	89.56%	89.60%	N/A
Vgg16 [30]	N/A	93.55%	93.52%	93.09%	N/A
Chen *et al*. [31]	N/A	97.38%	97.16%	97.21%	N/A
LDCNN	**98.38%**	**98.60%**	**97.89%**	**98.24%**	**156370**

Table 6. Performance efficiency of LDCNN model on Raspberry Pi 4B.

# Image	Evaluate Time	Average Frames per Second (FPS)
3701	1226.57 s	3.0174 s

5. Conclusions

In this study, a new quality detection of green coffee beans model, LDCNN was proposed, which combined DSC, SE block, skip block, and other frameworks, as well as HS and ReLU activation functions, to make the model lightweight and efficient. To improve the performance and training stability, RA, LA, and GC models were combined to avoid random prediction caused by the lightweight model. Based on the experimental results, compared with other state-of-the-art models, our model could achieve a higher accuracy rate of 98.38% and an F1 score of 98.24% in the quality detection of green coffee beans indicating excellent detection performance. When the model was placed in the embedded system, the average speed reached up to 3.02 FPS. Finally, the LIME interpretable model was used to verify that the model in this work is reliable, indicating that the impact of image pre-processing on the model after the image is interpretable and can be understood to optimize the training of the model or screen the abnormal data in the data set. Hence, the accuracy and generalization of the model can be improved during the training.

Author Contributions: C.-H. H. and C.-F. L. carried out the studies and drafted the manuscript. C.-H. H., Y.-H. L and C.-F. L. participated in its design and helped to draft the manuscript. C.-H. H. and Y.-H. L. conducted the experiments and performed the statistical analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C. under Grant No. MOST 111-2221-E-197-020-MY3.

Institutional Review Board Statement: Not applicable. This study did not involve humans or animals.

Informed Consent Statement: Not applicable. This study did not involve humans.

Data Availability Statement: This study did not report any data.

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

References

[1] M. Inoue, I. Yoshimi, T. Sobue, and S. Tsugane, “Influence of coffee drinking on subsequent risk of hepatocellular carcinoma: a prospective study in Japan,” *Journal of the National Cancer Institute*, vol. 97, no. 4, pp. 293-300, 2005.
[2] E. Loftfield, N. D Freedman, B. I Graubard, K. A Guertin, A. Black, W.-Y. Huang, F. M Shebl, S. T Mayne, and R. Sinha, “Association of coffee consumption with overall and cause-specific mortality in a large US prospective cohort study,” *American Journal of Epidemiology*, vol. 182, no. 12, pp. 1010-1022, 2015.
[3] P. Mazzafera, “Chemical composition of defective coffee beans,” *Food Chemistry*, vol. 64, no. 4, pp. 547-554, 1999.
[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.
[5] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, “Searching for MobileNetV3,” *IEEE/CVF International Conference on Computer Vision*, pp. 1314-1324, 2019.
[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: inverted residuals and linear bottlenecks,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510-4520, 2018.
[7] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132-7141, 2018.
[8] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” *International Conference on Learning Representations*, pp. 1-13, 2020.
[9] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” *International Conference for Learning Representations*, pp. 1-15, 2015.
[10] P. Goyal, P. Doll'ar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch SGD: training ImageNet in 1 hour,” *arXiv preprint* arXiv:1706.02677, 2017.
[11] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, “Lookahead optimizer: k steps forward, 1 step back,” *Conference on Neural Information Processing Systems*, pp. 1-12, 2019.
[12] H. Yong, J. Huang, X. Hua, and L. Zhang, “Gradient centralization: a new optimization technique for deep neural networks,” *European Conference on Computer Vision*, pp. 635-652, 2020.
[13] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you ?,” *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135-1144, 2016.
[14] Coffee bean dataset: small optical sorter. [Online]. Available: https://github.com/tanius/smallopticalsorter.
[15] ImageNet. [Online]. Available: https://www.image-net.org/
[16] M. Tan and Q. V. Le, “EfficientNetV2: smaller models and faster training,” *International Conference on Machine Learning*, pp. 10096-10106, 2021.
[17] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “ShuffleNet V2: practical guidelines for efficient CNN architecture design,” *European Conference on Computer Vision*, pp. 122-138, 2018.
[18] P. Wang, H.-W. Tseng, T.-C. Chen, and C.-H. Hsia,” Deep convolutional neural network for coffee bean inspection,” *Sensors and Materials*, vol. 33, no. 7, pp. 2299-2310, 2021.
[19] P.-Y. Yang, S.-Y. Jhong, and C.-H. Hsia, “Green coffee beans classification using attention-based features and knowledge transfer,” *IEEE International Conference on Consumer Electronics-Taiwan*, pp. 1-2, 2021.
[20] J. R. Santos, M. C. Sarraguça, A. O. S. S. Rangel, and J. A. Lopes, “Evaluation of green coffee beans quality using near infrared spectroscopy: a quantitative approach,” *Food Chemistry*, vol. 135, no. 3, pp. 1828-1835, 2012.
[21] E. M. de Oliveira, D. S. Leme, B. H. G. Barbosa, M. P. Rodarte, and R. G. F. A. Pereira, “A computer vision system for coffee beans classification based on computational intelligence techniques,” *Journal of Food Engineering*, vol. 171, pp. 22-27, 2016.
[22] E. R. Arboleda, A. C. Fajardo, and R. P. Medina, “Classification of coffee bean species using image processing, artificial neural network and K nearest neighbors,” *IEEE International Conference on Innovative Research and Development*, pp. 1-5, 2018
[23] E. R. Arboleda, A. C. Fajardo, and R. P. Medina, “An image processing technique for coffee black beans identification,” *IEEE International Conference on Innovative Research and Development*, pp. 1-5, 2018
[24] C. Pinto, J. Furukawa, H. Fukai, and S. Tamura, “Classification of green coffee bean images based on defect types using convolutional neural network (CNN),” *IEEE International Conference of Advanced Informatics*, pp. 1-5, 2017.
[25] N.-F. Huang, D.-L. Chou, C.-A. Lee, F.-P. Wu, A.-C. Chuang, Y.-H. Chen, and Y.-C. Tsai, “Smart agriculture: real-time classification of green coffee beans by using a convolutional neural network,” *IET Smart Cities*, vol. 2, no. 4, pp. 167–172, 2020.
[26] H. M. D. Kabir, M. Abdar, S. M. J. Jalali, A. Khosravi, A. Atiya, S. Nahavandi, and D. Srinivasan, “SpinalNet: deep neural network with gradual input,” *arXiv*:2007.03347, 2020.
[27] A. G. Howard M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyang, M. Andreetto, and H. Adam, “MobileNets: efficient convolutional neural networks for mobile vision applications, "*arXiv*:1704.04861, 2017.
[28] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700-4708, 2017.
[29] F. Chollet, “Xception: deep learning with depthwise separable convolutions,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800-1807, 2017.
[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *International Conference for Learning Representations*, pp. 1-14, 2015.

[31] P.-H. Chen, S.-Y. Jhong, and C.-H. Hsia, “Semi-supervised learning with attention-based CNN for classification of coffee beans defect,” *IEEE International Conference on Consumer Electronics-Taiwan*, pp. 1-2, 2022.