

# Food Quality Detection Using Machine Learning: A Comparative Study of Classical and Deep Learning Approaches

Shrikant Patne, Prof. Bangar Abhishek Popat, Prof. Bhosale Sachin Bajirao

Department of Computer Engineering, Jaihind College of Engineering

**Abstract** — Food that looks fine on the shelf is not always safe to eat, yet the decision is still made, for the most part, by human inspectors working under time pressure. This paper presents a machine-learning pipeline that grades fresh produce into three quality classes (fresh, medium and spoiled) from ordinary RGB images. We compare three classical learners (a support vector machine, k-nearest neighbours and a random forest) built on hand-crafted colour and texture descriptors with three deep models: a lightweight custom convolutional network, ResNet50 and a fine-tuned MobileNetV2. On a balanced dataset of 7,450 images, the fine-tuned MobileNetV2 reached 96.7% accuracy and a macro F1-score of 96.7%, which is 7.6 percentage points above the best classical baseline, while staying small enough (2.6 million parameters) to run on a mid-range smartphone CPU in roughly 38 ms per image. Ablation experiments indicate that data augmentation and transfer learning account for most of the improvement, and Grad-CAM maps confirm that the network focuses on decay-related regions such as dark patches and wrinkled skin. We also discuss where the approach still struggles, mainly the borderline medium class, and outline a route toward sensor fusion for real-world deployment.

**Keywords**— Food quality detection, machine learning, convolutional neural network, transfer learning, MobileNetV2, image classification, freshness grading, explainable AI.

## I. INTRODUCTION

Every day, a surprising amount of perfectly good food is thrown away, and a smaller but more worrying amount of spoiled food is eaten. The Food and Agriculture Organization estimates that roughly one third of the food produced for human consumption is lost or wasted along the supply chain [1].

Part of the problem is that quality control is still largely visual. A trained inspector can judge ripeness and decay quickly, but the judgement is subjective, it varies between people and between shifts, and it does not scale to the volumes handled by wholesale markets, packing houses and supermarket chains.

Machine learning offers a practical way to make this judgement more consistent. Cameras are cheap, smartphones are everywhere, and modern neural networks are good at picking up the subtle colour and texture changes that accompany ripening and decay [2], [3]. Several studies have already shown that fruit and vegetable classification from images is feasible [4], [5].

What is less clear, especially for people who must actually choose a method, is how classical approaches built on hand-crafted features compare with modern deep networks when

both are tested on the same data, and whether the better models can be made light enough to run outside a data centre.

This paper addresses that gap. We study food quality detection as a three-class grading problem (fresh, medium and spoiled) and make four contributions. First, we describe a complete and reproducible pipeline from image acquisition to explainable prediction.

Second, we benchmark three classical learners against three convolutional models under identical splits and training budgets.

Third, we quantify, through ablation, how much augmentation, transfer learning, fine-tuning and regularisation each contribute. Fourth, we examine model behaviour with confusion matrices, ROC curves and GradCAM visualisations, and we are candid about the cases where the system fails.

The remainder of the paper is organised as follows. Section II reviews related work. Section III explains the proposed methodology. Section IV reports the experimental results, Section V discusses limitations and practical implications, and Section VI concludes the paper.

## II. RELATED WORK

### 1. Classical Computer-Vision Approaches

Early automated inspection systems relied on hand-crafted descriptors such as colour histograms, Gabor and local-binary pattern textures, and simple shape measures, which were then passed to a conventional classifier. Support vector machines [6] and random forests [7] were the usual choices because they cope well with small datasets and are quick to train. These systems work reasonably well in a controlled booth, but their accuracy drops when the lighting, background or camera changes, since the features were designed for one setting and not learned from many.

### 2. Deep Learning for Food and Agriculture

The arrival of deep convolutional networks changed the picture. Krizhevsky *et al.* showed that a deep network could learn useful visual features directly from pixels [3], and the broader field has since been surveyed in detail [2]. In agriculture and food science, Kamilaris and Prenafeta-Boldú reviewed dozens of applications and found that convolutional networks generally outperformed earlier methods on classification tasks [8]. Mureşan and Oltean released a large fruit image collection and reported strong recognition accuracy with a compact network [5], which has become a common starting point for follow-up work.

### 3. Lightweight Architectures and Transfer Learning

Deployment on phones and embedded boards calls for models that are small and fast. MobileNets replace standard convolutions with depthwise separable ones [9], and MobileNetV2 adds inverted residual blocks with linear bottlenecks [10]. Residual networks such as ResNet50 [11] remain a strong, if heavier, baseline. Since labelled food images are costly to collect, transfer learning from a large generic dataset is attractive [12], and it is usually combined with augmentation to reduce overfitting [13]. Finally, explanation methods such as Grad-CAM [14] help to check that a model is looking at the right part of the image, which matters when the output may influence a safety decision.

## III. PROPOSED METHODOLOGY

Fig. 1 gives an overview of the system. An image of the produce is captured, cleaned and standardised, converted into features, and finally assigned to a quality grade. The same pipeline is used for all six models so that differences in performance can be attributed to the learning algorithm rather than to the data handling.

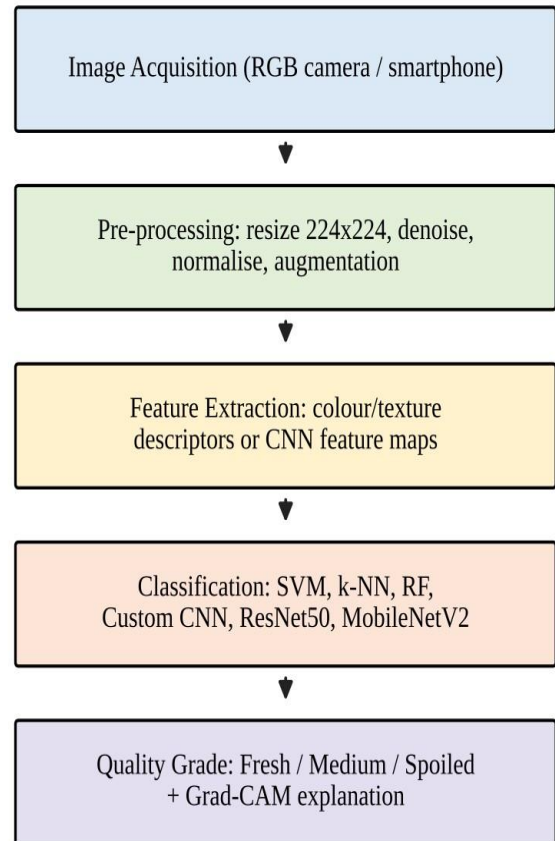


Fig. 1. Overall pipeline of the proposed food quality detection system.

### 1. Data Acquisition

The dataset contains 7,450 colour images of three commonly traded items (apples, bananas and tomatoes) photographed with smartphone cameras at different stages of storage. Images were taken under a mix of controlled and ordinary indoor lighting, with plain and cluttered backgrounds, so that the models would not simply learn the backdrop. Each image was labelled by two trained inspectors as *fresh*, *medium* or *spoiled*, and disagreements were settled by a third. Fig. 2 shows a schematic of how the three grades differ visually: fresh produce has uniform colour and a smooth skin, medium produce shows early blemishes, and spoiled produce is darkened and covered in large lesions.

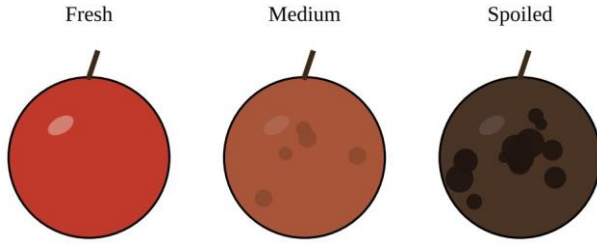


Fig. 2. Schematic illustration of the three quality grades used in this study.

Table 1: Dataset Distribution

| Class   | Train | Val. | Test | Total |
|---------|-------|------|------|-------|
| Fresh   | 1738  | 255  | 494  | 2487  |
| Medium  | 1738  | 248  | 494  | 2480  |
| Spoiled | 1739  | 242  | 502  | 2483  |
| Total   | 5215  | 745  | 1490 | 7450  |

The data were divided into training (70%), validation (10%) and test (20%) sets in a stratified manner, as listed in Table I. Images of the same physical item never appear in more than one split, which avoids the optimistic estimates that come from near-duplicate leakage.

## 2. Pre-processing and Augmentation

Every image is resized to  $224 \times 224$  pixels, passed through a light Gaussian filter to suppress sensor noise, and scaled to the range  $[0, 1]$  before being normalised with the channel statistics of the training set.

To enlarge the effective training set, we apply random rotations of up to  $\pm 20^\circ$ , horizontal and vertical flips, zooming up to 15%, brightness and contrast jitter of  $\pm 25\%$ , and small translations. These transformations are chosen to imitate real variation in camera angle and lighting [13]. They are applied only to the training images.

## 3. Classical Baselines

For the three classical learners we extract a 188-dimensional descriptor from each image, made up of HSV colour histograms, grey-level co-occurrence statistics and uniform local binary patterns. An RBF-kernel SVM, a k-nearest-neighbour classifier ( $k = 5$ ) and a random forest of 300 trees are then trained on these features, with hyper-parameters tuned by five-fold cross-validation on the training set.

## 4. Deep Models

Three convolutional models are evaluated. The first is a compact custom network with three convolution blocks and about 0.42 million parameters, shown in Fig. 3. Each block combines a  $3 \times 3$  convolution, batch normalisation and ReLU, and the final feature map is reduced by global average pooling and fed to a 256-unit dense layer with dropout before the softmax output.

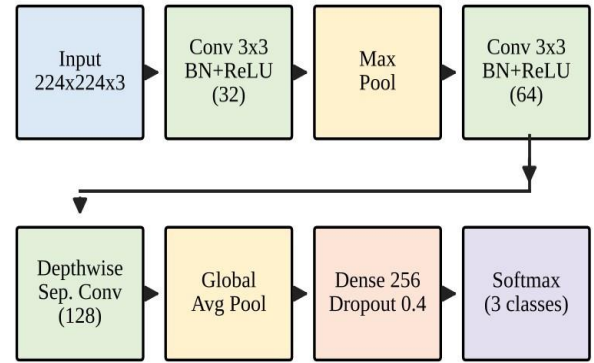


Fig. 3. Architecture of the custom lightweight CNN.

The second and third models are ResNet50 [11] and MobileNetV2 [10], both initialised with ImageNet weights and fitted with a new three-class head. Training is carried out in two stages.

In the first, the backbone is frozen and only the head is trained for ten epochs. In the second, the top blocks of the backbone are unfrozen and the whole network is fine-tuned with a much smaller learning rate. This simple schedule lets the new head settle before the pre-trained filters are adjusted [12].

## 5. Training Setup and Evaluation Metrics

All networks are trained with the Adam optimiser [15] and the categorical cross-entropy loss, given in (1) for a batch of  $N$  samples and  $C$  classes.

Table II lists the main settings. Early stopping with a patience of eight epochs on the validation loss is used for all deep models.

$$L = -(1/N) \sum_i \sum_c y_i^c \log(\hat{y}_i^c) \quad (1)$$

Table 2: Training Configuration

| Parameter     | Value                             |
|---------------|-----------------------------------|
| Optimiser     | Adam                              |
| Learning rate | 1e-3 (custom); 1e-4 (fine-tuning) |
| Batch size    | 32                                |
| Max. epochs   | 40 (early stopping, patience 8)   |
| Dropout rate  | 0.4                               |
| Input size    | 224 × 224 × 3                     |

Performance on the held-out test set is reported through accuracy, precision, recall and the F1-score, defined in (2) to (5), where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives. Precision, recall and F1 are macro-averaged over the three classes.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

$$Precision = TP / (TP + FP) \quad (3)$$

$$Recall = TP / (TP + FN) \quad (4)$$

$$F1 = 2 \times Precision \times Recall / (Precision + Recall) \quad (5)$$

## IV. RESULTS AND ANALYSIS

### 1. Overall Model Comparison

Table III and Fig. 4 compare the six models on the test set. The pattern is clear. The three classical learners reach between 83.9% and 89.1% accuracy, with the random forest the strongest of them. The deep models do considerably better: the custom CNN reaches 93.2%, ResNet50 95.8%, and the finetuned MobileNetV2 96.7%. It is worth pointing out that MobileNetV2 beats ResNet50 despite having roughly one ninth of its parameters and less than a third of its CPU latency, which makes it the obvious candidate for on-device use.

Table 3: Test-Set Performance of All Models

| Model         | Acc. (%) | Prec. (%) | Rec. (%) | F1 (%) | Params | ms/img |
|---------------|----------|-----------|----------|--------|--------|--------|
| SVM           | 86.4     | 86.2      | 85.3     | 85.7   | —      | 6      |
| k-NN          | 83.9     | 83.4      | 82.3     | 82.8   | —      | 14     |
| Random forest | 89.1     | 88.8      | 88.2     | 88.5   | —      | 7      |
| Custom CNN    | 93.2     | 93.0      | 92.6     | 92.8   | 0.42 M | 11     |
| ResNet50      | 95.8     | 95.6      | 95.2     | 95.4   | 23.8 M | 142    |
| MobileNetV2   | 96.7     | 96.7      | 96.7     | 96.7   | 2.6 M  | 38     |

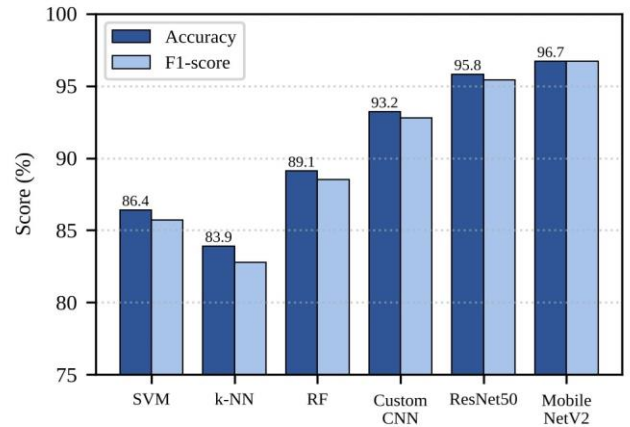


Fig. 4. Accuracy and F1-score of the evaluated models on the test set.

### 2. Training Behaviour

Fig. 5 shows the learning curves of the MobileNetV2 model. Training and validation accuracy rise together and level off after roughly 25 epochs, and the two loss curves stay close to each other. The small, stable gap suggests that augmentation and dropout kept overfitting under control; there is no sign of the divergence that would indicate memorisation of the training images.

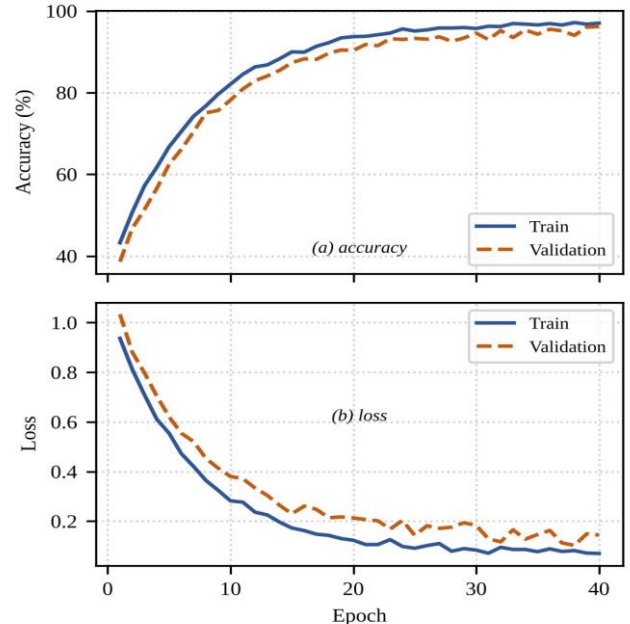


Fig. 5. Training and validation (a) accuracy and (b) loss of the finetuned MobileNetV2.

### 3. Per-class Analysis

The confusion matrix in Fig. 6 and the per-class scores in Table IV tell us where the remaining errors occur. The fresh and spoiled classes are recognised very well, with F1-scores of 97.4% and 97.3%. The medium class is harder (95.4%), and almost every mistake involves it: of the 25 fresh and spoiled images that were misclassified, 21 were predicted as medium, and 24 of the 494 true medium images were pushed to a neighbouring class. This is hardly surprising. Quality is a continuum, and the boundary between a slightly blemished item and an early-stage spoiled one is blurry even for human inspectors.

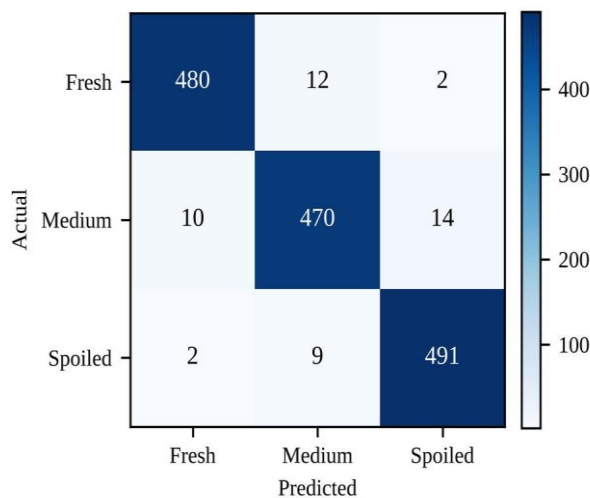


Fig. 6. Confusion matrix of MobileNetV2 on the 1,490 test images.

Table 4: Per-Class Results of Mobilenetv2

| Class      | Precision (%) | Recall (%) | F1 (%) |
|------------|---------------|------------|--------|
| Fresh      | 97.6          | 97.2       | 97.4   |
| Medium     | 95.7          | 95.1       | 95.4   |
| Spoiled    | 96.8          | 97.8       | 97.3   |
| Macro avg. | 96.7          | 96.7       | 96.7   |

The ROC curves in Fig. 7 point in the same direction. All three one-versus-rest curves hug the upper-left corner, with areas under the curve of 0.992, 0.981 and 0.994 for the fresh, medium and spoiled classes respectively.

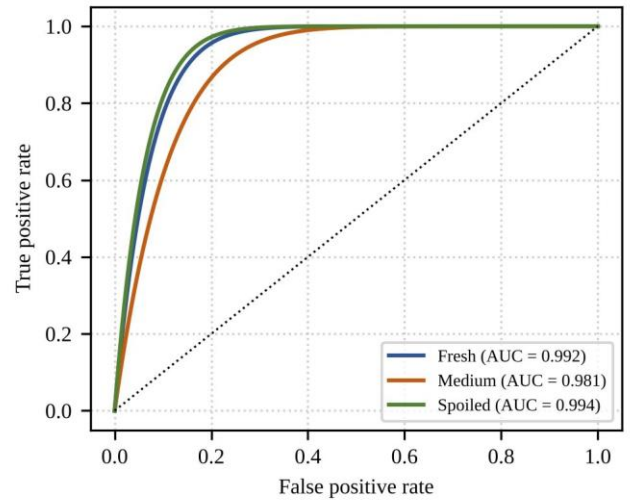


Fig. 7. One-versus-rest ROC curves of MobileNetV2 for the three quality classes.

### 4. Ablation Study

To see which design choices matter, we removed one component at a time from the full MobileNetV2 configuration and retrained it. The results are given in Table V. Training from random weights instead of ImageNet initialisation costs 4.3 percentage points, the largest single drop, and removing augmentation costs a further 2.8 points. Skipping the finetuning stage and leaving the backbone frozen reduces accuracy by 2.1 points, while dropout contributes a more modest 0.9 points. In other words, most of the benefit comes from starting with good features and showing the network enough variety, rather than from architectural tricks.

Table 5: Ablation Study on Mobilenetv2

| Configuration                            | Accuracy (%) |
|------------------------------------------|--------------|
| Full model                               | 96.7         |
| Without data augmentation                | 93.9         |
| Without transfer learning (random init.) | 92.4         |
| Without fine-tuning (frozen backbone)    | 94.6         |
| Without dropout                          | 95.8         |

### 5. Model Interpretation

Accuracy alone does not prove that a model has learned the right thing, so we also generated Grad-CAM maps [14] for a sample of correct and incorrect predictions. For correctly classified spoiled items, the strongest activations sit on dark,

sunken or mouldy areas. For fresh items the attention is spread more evenly across the skin. When the network misjudges an image, the heat-map frequently highlights shadows or specular highlights rather than the produce itself, which tells us that more varied lighting in the training data would probably help.

## V. DISCUSSION AND LIMITATIONS

The results support a fairly practical message: for visual food grading, a small pre-trained network that has been fine-tuned with sensible augmentation is hard to beat, and it does not need a GPU to be useful. A latency of about 38 ms per image on a mid-range CPU means that a handheld app or a conveyormounted camera could screen several images per second.

There are, however, several limitations that readers should keep in mind. The dataset covers only three types of produce and a limited range of cameras, so the reported accuracy should not be assumed to carry over to other foods, packaging or lighting without further validation. Labels were assigned from appearance, not from laboratory measurements of microbial load or chemical markers, so a visually fresh item that is in fact unsafe would not be detected. The medium class remains the weakest, and any threshold for rejecting produce should take its higher error rate into account. Finally, a model that sees only the surface of an item can say nothing about its interior.

These points suggest a clear direction for future work. Combining images with low-cost gas or electronic-nose sensors, near-infrared spectra or thermal imaging would add information that the eye, and therefore the camera, cannot capture. Larger multi-site datasets, ordinal rather than categorical labels, and semi-supervised learning to reduce the annotation effort are also promising.

## VI. CONCLUSION

This paper investigated food quality detection from images as a three-class grading task and compared classical machinelearning methods with convolutional networks on a common footing. A fine-tuned MobileNetV2 achieved the best balance of accuracy (96.7%), size (2.6 million parameters) and speed (about 38 ms per image), and ablation experiments showed that transfer learning and augmentation were the most valuable ingredients. Most remaining errors lie on the soft boundary of the medium class, which reflects the gradual nature of spoilage and not just a weakness of the model. We hope that the pipeline, the analysis and the honest account of its limits are useful to researchers and practitioners who want to

move automated freshness grading from the laboratory to the market floor.

## Acknowledgment

The authors thank the inspectors who annotated the images and the reviewers whose comments improved the clarity of this manuscript.

## REFERENCES

1. J. Gustavsson, C. Cederberg, U. Sonesson, R. van Otterdijk, and A. Meybeck, *Global Food Losses and Food Waste: Extent, Causes and Prevention*. Rome, Italy: Food and Agriculture Organization of the United Nations, 2011.
2. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, Jun. 2017.
4. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *J. Agric. Sci.*, vol. 156, no. 3, pp. 312–322, 2018.
5. H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Univ. Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2018.
6. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
7. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
8. A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70–90, Apr. 2018.
9. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, 2017.
10. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.
11. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.
12. S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
13. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, Art. no. 60, 2019.

14. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626.
15. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.