# Fruit Quality Assessment with Densely Connected Convolutional Neural Network

Md. Samin Morshed*, Sabbir Ahmed†, Tasnim Ahmed‡, Muhammad Usama Islam§, A.B.M. Ashikur Rahman††

*†‡Department of Computer Science and Engineering, Islamic University of Technology, Gazipur 1704, Bangladesh

§ School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA

†† Department of ICS, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia

{*saminmorshed, †sabbirahmed, ‡tasnimahmed}@iut-dhaka.edu, §usamaislam1@louisiana.edu, ††g202204800@kfupm.edu.sa

*Abstract*—Accurate recognition of food items along with quality assessment is of paramount importance in the agricultural industry. Such automated systems can speed up the wheel of the food processing sector and save tons of manual labor. In this connection, the recent advancement of Deep learning-based architectures has introduced a wide variety of solutions offering remarkable performance in several classification tasks. In this work, we have exploited the concept of Densely Connected Convolutional Neural Networks (DenseNets) for fruit quality assessment. The feature propagation towards the deeper layers has enabled the network to tackle the vanishing gradient problems and ensured the reuse of features to learn meaningful insights. Evaluating on a dataset of 19,526 images containing six fruits having three quality grades for each, the proposed pipeline achieved a remarkable accuracy of 99.67%. The robustness of the model was further tested for fruit classification and quality assessment tasks where the model produced a similar performance, which makes it suitable for real-life applications.

*Index Terms*—Fruit classification, CNN, Transfer learning, Deep learning, Neural network, Explainable AI

## I. INTRODUCTION

Fruit classification has emerged as an important aspect in the domain of agriculture, machine learning, and image classification in recent years. Fast and accurate fruit classification is a major challenge to consider to increase efficiency in the farming sector [1], [2]. Deep learning models, which are basically based on artificial neural networks have shown formidable performances in fruit detection and classification tasks [3]. The potential and prospect of Deep Learning (DL) were further explored using Convolutional Neural Network (CNN) based architectures [4]. Albeit having substantial prospects, the main challenges that fruit identification research faces involve challenges pertaining to the irregularity of form, size, and variability in color [3].

Classification of fresh and damaged fruits was explored through the means of CNN by Kumar *et al*. [5]. While the work carried out by Kazi and Panda [4] also explored the freshness of fruits, their approach differed by employing the power of transfer learning instead of a vanilla CNN. Kumar's work [5] showed an accuracy of 97.14% while Kazi *et al*. [4] reported having 99% accuracy, which implies DL models can perform better through transfer learning and fine-tuning.

Siddiqi *et al*. [6] affirmed the findings reported by Valdez *et al*. [7] and reiterated finding better results with SSD The authors of [8] investigated several networks and proposed a CNN-YOLO induced regression network for fruit quality detection on six types of fruits that aligned with the work carried out using YOLO in [6], [7]. Hussain *et al*. [9] provided a dataset worth 10,000 fruits images and proposed a Deep CNN achieving an accuracy of 96%. Meshram *et al*. [10] curated a dataset containing more than 19,526 images of highly popular fruits in India with three quality labels, namely good, bad, and mixed quality. A framework named MNet was proposed in [11] for reducing fruit misclassification, where the authors curated a dataset having 12,000 images for binary classification and experimented with different state-of-the-art CNN architectures.

## II. METHODOLOGY

### A. Dataset

The experiments were conducted on the 'FruitNet' dataset proposed by Meshram *et al*. [10], which contains images of fruits along with quality levels. There are six different fruits (apple, banana, guava, lime, orange, and pomegranate) and for each fruit, three types of quality information are available (good, bad, and mixed). The dataset contains 19,526 images belonging to a total of 18 classes. Detailed information about the class distribution is available in Table I.

The samples were captured with a high-resolution smartphone camera under varying lighting and background conditions. Samples of the same fruit belonging to different quality levels contain a high degree of visually similar features which is also an additional challenge for the model to overcome. The samples of different fruits contain similarities in terms of color, shape, and texture pattern. Moreover, samples within the same class contain a high degree of variability due to the diversity of breeds. These challenges make this dataset a suitable one to train intelligent systems that can be applicable to real-life problems. In this work, we utilized this dataset for three different tasks, which are, 'Fruit Classification' (6 classes), 'Quality Assessment' (3 classes), and 'Fruit Classification with Quality Levels' (18 classes). Figure 1 contains one sample from each of the classes of the dataset.
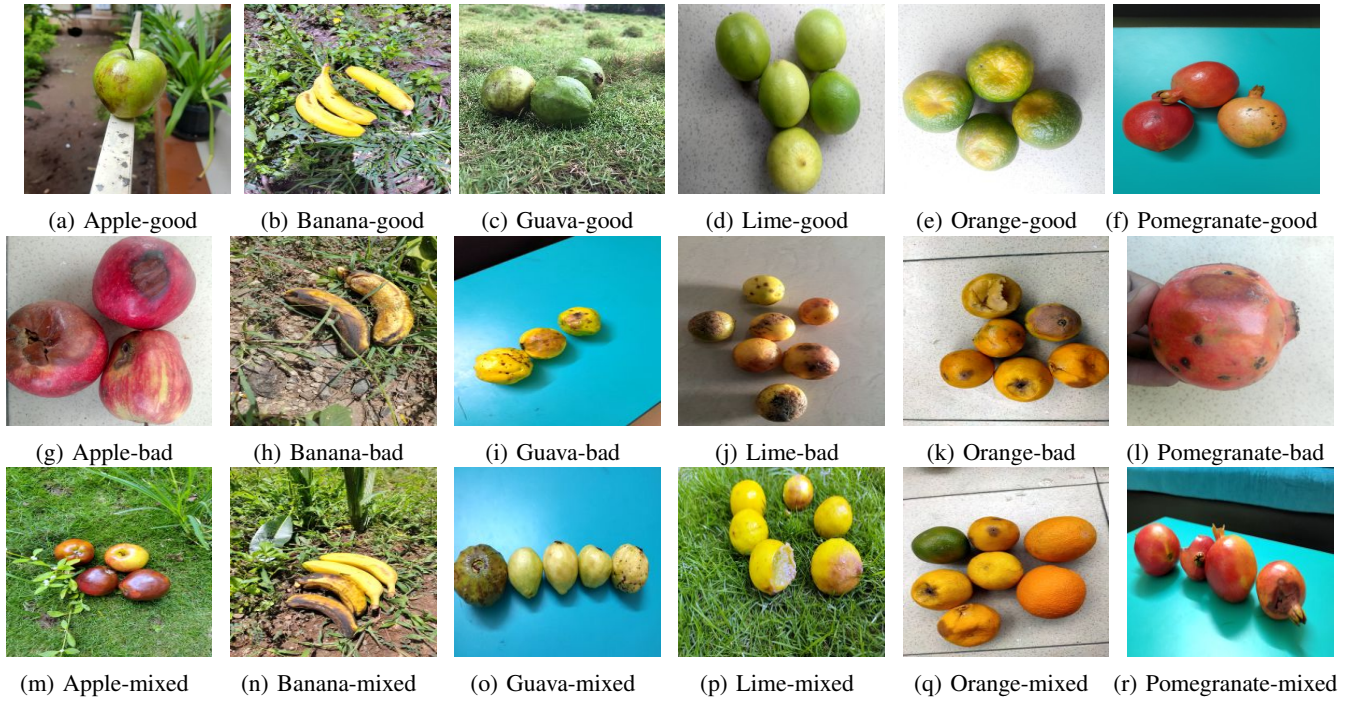
|  |  |  |  |  |  |
|---|---|---|---|---|---|
| (a) Apple-good | (b) Banana-good | (c) Guava-good | (d) Lime-good | (e) Orange-good | (f) Pomegranate-good |
| (g) Apple-bad | (h) Banana-bad | (i) Guava-bad | (j) Lime-bad | (k) Orange-bad | (l) Pomegranate-bad |
| (m) Apple-mixed | (n) Banana-mixed | (o) Guava-mixed | (p) Lime-mixed | (q) Orange-mixed | (r) Pomegranate-mixed |

Fig. 1: Sample images of the FruitNet dataset

## B. Model Description

Convolutional Neural Networks (CNNs) are being considered the gold standard to deal with image classification tasks in recent times [12], [13]. Since the introduction of Alexnet, CNNs outperformed all existing algorithms by a significant margin and there has been a great deal of interest in investigating various neural network topologies and experimenting with various strategies such as expanding the width or depth of neural networks [14]. Deeper networks are thought to perform better in general due to their increased learning capacity. However, with the increasing depth of the architecture, the problem of vanishing gradient arises.

To address these difficulties, alternative architectures with residual connections were developed. Densely Connected Deep Neural Networks (DenseNets) handle the same issue by having the output from each layer concatenated to all subsequent layers [15]. In DenseNets, every subsequent layer receives information from all the preceding layers of the chain. These values are concatenated to ensure that no information is lost. This concatenated feature map is run through a composite function that includes Batch Normalization, Relu, and $3 \times 3$ Convolution. The result is then passed on to the next layer and the process is repeated.

In a deep network, this concatenation may result in a very large number of parameters. As a result, the design is divided into many dense blocks, each separated by a transition block. Downsampling occurs in the transition block to prevent blowing up. The transition layer consists of batch normalization layers, a $1 \times 1$ convolution layer, and a $2 \times 2$ average pooling layer. Based on the number of layers, several variants of the basic DenseNet architectures are available, where DenseNet201 has achieved the highest accuracy in the ImageNet dataset. Hence, it has been chosen as the backbone.

## C. Data Augmentation

As described in Table I, the dataset contains class imbalance since the number of samples for mixed quality is significantly lower compared to the good or bad quality classes. As Deep CNNs require a huge amount of data to automatically find the hidden patterns, data augmentation can be extremely useful in this regard to ensure that the model has the opportunity to learn to its fullest capacity. However, traditional augmentation techniques follow policies like undersampling each class to the number of samples in the least populated class or oversampling each class to expand to a threshold. However, undersampling has the risk of losing critical samples, and oversampling can extend the samples only to some extent. Hence, instead of increasing/decreasing the samples before the training phase, we adopted runtime augmentation, which augments the samples with random strength in each epoch, ensuring providing new challenges to the model [16].

TABLE I: Distribution of Samples in the Dataset

| Class Label | Good | Bad | Mixed | Total |
|---|---|---|---|---|
| Apple | 1149 | 1141 | 113 | 2403 |
| Banana | 1113 | 1087 | 285 | 2485 |
| Guava | 1152 | 1129 | 148 | 2429 |
| Lime | 1094 | 1085 | 278 | 2457 |
| Orange | 1216 | 1159 | 125 | 2500 |
| Pomegranate | 5940 | 1187 | 125 | 7252 |
| Total | 11664 | 6788 | 1074 | 19526 |

2

To reflect the real-world scenarios, we have used augmentations like random rotation, horizontal & vertical flipping, height & width shifting, and shearing with a random strength. Multiple of these augmentations are applied to the samples allowing a new version to be presented in front of the model each time it is loaded during training.

## III. RESULTS AND DISCUSSIONS

### A. Experimental Setup

The proposed pipeline was trained in a Google Colaboratory environment on a NVIDIA Tesla T4 GPU having a virtual memory of 15GB. The input images were resized to a dimension of $256 \times 256 \times 3$. The split ratio of 60:20:20 was followed corresponding to the training, validation, and testing set. The model was trained with a batch size of 32 to satisfy the available GPU memory. Training was allowed up to 1000 epochs with early stopping to restrict if no significant improvement was found within 5 epochs. For all of the tasks, the Adam optimizer was used and the Learning rate was set to 0.0001 along with a decay value of 0.1 after 5 patient epochs.

### B. Performance of Different Baseline Architectures

To choose the best-suited model, a number of state-of-the-art CNN models were evaluated, which were pretrained for the large-scale image classification task and fine-tuned using the FuritNet dataset. With the utilization of transfer learning, the models were already capable of learning complex patterns, ensuring faster convergence. With the goal of picking the best-suited model, we only modified the softmax layer. Due to the high learning capacity of these models, all combinations achieved an accuracy above 97%. Surprisingly, MobileNetV2 produced an accuracy of 98.62% which is higher than deeper architectures such as ResNet152 and Inception-ResNetV2. However, only DenseNet201 architecture achieved an accuracy of more than 99%. Furthermore, the number of training parameters for DenseNet201 does not impose high computational resources, making it the most suitable model for fruit quality assessment.

We further enhanced the capability of the best model with Data augmentation techniques. Our idea of runtime augmentation improved the accuracy of the DenseNet201 architecture up to 99.67% for fine-grain fruit quality assessment. To ensure robustness, two new tasks were introduced; namely, fruit classification (6 class) and quality classification (3 class). In the first task, the model had to predict the class of fruit, and in the second one, it had to predict whether the fruit is of good, bad, or mixed quality. The model, respectively, achieved 99.7% and 99.67% accuracy for the two additional tasks.

### C. Class-wise Performance & Error Analysis

Table III illustrates the class-wise performance along with the number of samples of individual classes in the test set. Although accuracy is a very well-known performance metric, it is also known for its sensitivity toward class imbalance. For a more robust analysis, we have considered precision, recall, and F-1 score for each class. Furthermore, to get an insight into

TABLE II: Performance Comparison of Baseline Architectures

| Architecture | Accuracy (%) | Trainable Parameters (millions) |
|---|---|---|
| ResNet152 | 97.86 | 58.26 |
| InceptionResNetV2 | 97.91 | 54.3 |
| EfficientNetV2B0 | 97.95 | 5.88 |
| VGG16 | 98.6 | 14.72 |
| MobileNetV2 | 98.62 | 2.25 |
| InceptionV3 | 98.8 | 21.81 |
| Xception | 98.98 | 20.84 |
| DenseNet201 | **99.26** | 18.13 |

TABLE III: Performance of Individual Class

| Fruit | Quality | Precision | Recall | F1-Score | AUC-ROC | Support |
|---|---|---|---|---|---|---|
| Apple | Bad | 1.0000 | 0.9956 | 0.9978 | 1.0000 | 229 |
| | Good | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 231 |
| | Mixed | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 24 |
| Banana | Bad | 0.9909 | 0.9954 | 0.9931 | 0.9973 | 218 |
| | Good | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 224 |
| | Mixed | 0.9821 | 0.9649 | 0.9735 | 0.9998 | 57 |
| Guava | Bad | 0.9913 | 1.0000 | 0.9956 | 0.9999 | 227 |
| | Good | 0.9957 | 1.0000 | 0.9978 | 1.0000 | 231 |
| | Mixed | 1.0000 | 0.9355 | 0.9667 | 0.9999 | 31 |
| Lime | Bad | 0.9954 | 1.0000 | 0.9977 | 0.9999 | 217 |
| | Good | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 220 |
| | Mixed | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 57 |
| Orange | Bad | 0.9830 | 0.9914 | 0.9872 | 0.9999 | 233 |
| | Good | 0.9917 | 0.9836 | 0.9877 | 0.9999 | 244 |
| | Mixed | 1.0000 | 0.9600 | 0.9796 | 1.0000 | 25 |
| Pomegranate | Bad | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 238 |
| | Good | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1188 |
| | Mixed | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 25 |

the class-separability of our proposed model, the area under the receiver operating characteristics curve (AUC-ROC) is also investigated. The AUC-ROC score can be used to evaluate a classifier's performance by describing the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) using several probability thresholds. Our model achieved an AUC-ROC score of 1 for the majority of the classes, as indicated in Table III. Scores for the remaining classes are fairly high, showing that class separability is satisfactory. Figure 2 depicts a few of the misclassified samples, and it can be observed that the majority of them can also perplex human decisions regarding the predicted class.
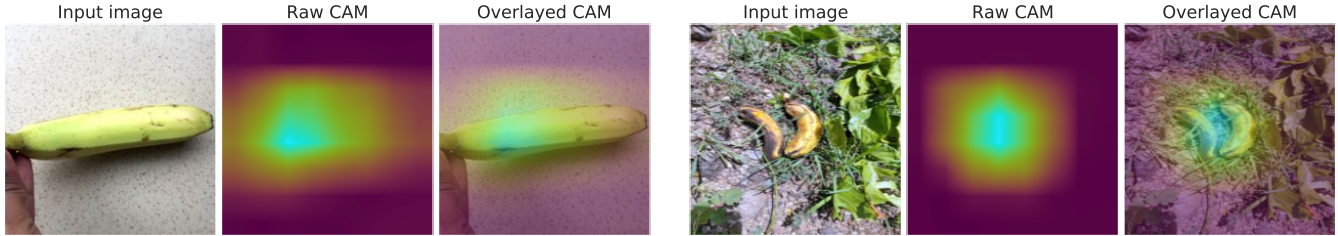
### D. Explainability

An explainability study was conducted to gain insights into the decision-making strategy of the proposed framework. Using Grad-CAM [17], we examined the Class Activation Map (CAM) during a forward pass provided by the last convolutional layer. The weight for each of the input feature maps is determined by the gradient of the loss with respect to the final convolutional layer. The weighted sum of the activation maps is then up-sampled to the input resolution to build a heatmap that resembles the region that contributes the most to the predicted class. Figure 3 depicts the class activation map of a correctly classified and a misclassified sample, and it is clear that our model provides the highest activation score for the fruit region while disregarding the background.

Actual: Apple_Bad
Predicted: Lime_Bad

Actual: Banana_Bad
Predicted: Banana_Mixed

Actual: Guava_Mixed
Predicted: Guava_Bad

Actual: Orange_Good
Predicted: Orange_Bad

Actual: Orange_Bad
Predicted: Orange_Good

Actual: Orange_Mixed
Predicted: Orange_Good

Fig. 2: Misclassified samples

Input image

Raw CAM

Overlayed CAM

Input image

Raw CAM

Overlayed CAM

(a) Grad-CAM output for correctly classified 'Banana' sample

(b) Grad-CAM output for misclassified 'Banana' sample

Fig. 3: Grad-CAM analysis of the test samples

## IV. CONCLUSION

Automated quality assessment systems can play an important role in the processing of perishable raw materials by minimizing manual labor. In this work, we proposed a pipeline utilizing the densely connected convolutional neural networks for assessing fruit quality from images. The internal dense connections alleviated vanishing-gradient problems, strengthened the propagation, and reused features in deeper layers. The capability of the model was enhanced even more by employing run-time augmentation. The robustness of the pipeline was thoroughly investigated for three different tasks- fruit classification, quality assessment, and fine-grained fruit-quality assessment; and it provided a remarkable accuracy of around 99.67% for all tasks. Furthermore, we provided a detailed class-wise analysis along with an explanation regarding the parts of the weights that excite the decisions. In the future, this work can be extended by applying image segmentation and object detection models to accurately detect the regions of bad-quality fruits. Assessing the degree of decomposition can be another interesting task to consider.

## REFERENCES

[1] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 3, pp. 243–257, 2021.

[2] M. M. Hasan, M. U. Islam, and M. J. Sadeq, "Towards the technological adaptation of advanced farming through artificial intelligence, the internet of things, and robotics: A comprehensive overview," *Artificial Intelligence and Smart Agriculture Technology*, pp. 21–42, 2022.

[3] C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, Z. Almaspoor, and H. N. Monday, "Recent advancements in fruit detection and classification using deep learning techniques," *Mathematical Problems in Engineering*, vol. 2022, 2022.

[4] A. Kazi and S. P. Panda, "Determining the freshness of fruits in the food industry by image classification using transfer learning," *Multimedia Tools and Applications*, vol. 81, no. 6, pp. 7611–7624, 2022.

[5] T. B. Kumar, D. Prashar, G. Vaidya, V. Kumar, S. Kumar, and F. Sammy, "A novel model to detect and classify fresh and damaged fruits to reduce food waste using a deep learning technique," *Journal of Food Quality*, vol. 2022, 2022.

[6] R. Siddiqi, "Automated apple defect detection using state-of-the-art object detection techniques," *SN Applied Sciences*, vol. 1, no. 11, pp. 1–12, 2019.

[7] P. Valdez, "Apple defect detection using deep learning based object detection for better post harvest handling," *arXiv preprint arXiv:2005.06089*, 2020.

[8] Y. Fu, "Fruit freshness grading using deep learning," Ph.D. dissertation, Auckland University of Technology, 2020.

[9] D. Hussain, I. Hussain, M. Ismail, A. Alabrah, S. S. Ullah, and H. M. Alaghbari, "A simple and efficient deep learning-based framework for automatic fruit recognition," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[10] V. Meshram and K. Patil, "Fruitnet: Indian fruits image dataset with quality for machine learning applications," *Data in Brief*, vol. 40, p. 107686, 2022.

[11] V. A. Meshram, K. Patil, and S. D. Ramteke, "Mnet: A framework to reduce fruit image misclassification." *Ingénierie des Systèmes d Inf.*, vol. 26, no. 2, pp. 159–170, 2021.

[12] T. Ahmed, M. N. Raihan, R. Kushol, and M. S. Salekin, "A complete bangla optical character recognition system: An effective approach," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1–7.

[13] A. Yasmeen, F. I. Rahman, S. Ahmed, and M. H. Kabir, "Csvc-net: Code-switched voice command classification using deep cnn-lstm network," in *2021 Joint 10th International Conference on Informatics, Electronics Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, 2021, pp. 1–8.

[14] A. B. M. Ashikur Rahman, M. B. Hasan, S. Ahmed, T. Ahmed, M. H. Ashmafee, M. R. Kabir, and M. H. Kabir, "Two decades of bengali handwritten digit recognition: A survey," *IEEE Access*, vol. 10, pp. 92 597–92 632, 2022.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of CVPR*, July 2017.

[16] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony, and M. H. Kabir, "Less is more: Lighter and faster deep neural architecture for tomato leaf disease classification," *IEEE Access*, vol. 10, pp. 68 868–68 884, 2022.

[17] 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 *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.