A Deep Learning Approach Using ConvNeXt-Tiny and EfficientNetV2-B0 for Multi-Fruit Quality Prediction

1st Abburi Ramesh
Derpartment of CSE
Narasaraopeta Engineering College
Andhra Pradesh, India
abburi.ramesh@gmail.com

2ndKambhampati John Wesly
Derpartment of CSE
Narasaraopeta Engineering College
Andhra Pradesh, India
johnwesleykambhampati@gmail.com

3rdVinnakota Manoj Kumar
Derpartment of CSE
Narasaraopeta Engineering College
Andhra Pradesh, India
manojkumarvinnakota@gmail.com

4th Bokkisam Anil Kumar
Derpartment of CSE
Narasaraopeta Engineering College
Andhra Pradesh, India
anilkumarbokkisam820@gmail.com

5th Shaikthettu Sharif
Derpartment of CSE
Narasaraopeta Engineering College
Andhra Pradesh, India
sharif55167849@gmail.com

6th R.S. Gangadharan
Derpartment of CSE
GREIT
Telangana, India.
grsakthi1293@grietcollege.com

7th Moturi Sireesha
Derpartment of CSE
Narasaraopeta Engineering College
Andhra Pradesh, India
sireeshamoturi@gmail.com

***Abstract*—Automating fruit sorting is difficult because produce varies so much in appearance and datasets are rarely balanced. Existing solutions tend to focus on solving each of these problems, which affects the overall efficiency of the system.This paper proposes a dual-stage framework based deep learning system for the integration of these two tasks. For classification of six fruit categories in the FruitNet dataset, a ConvNeXt-Tiny model pretrained on ImageNet is used, and an EfficientNetV2-B0 model that has undergone same-domain transfer learning is used for quality level grading (good, average, poor).Standard geometric augmentations and class-weighted loss functions are applied to mitigate class imbalance and enhance robustness.With a solid F1-score of 95.6%, the system proves reliable enough for real-world use on edge devices, the proposed system is superior to the existing approaches.Mostly, the suggested modular and lightweight architecture showcases the practicality of instant fruit evaluation on underpowered gadgets as a first important move for extending agricultural automation.**

***Index Terms*—Fruit Classification, Quality Grading, Deep Learning, ConvNeXt-Tiny, EfficientNetV2-B0, Agricultural Automation, Transfer Learning, Data Augmentation, Precision Agriculture, FruitNet Dataset**

I. Introduction

Early researchers tackled the conventional image processing and changed to using deep learning techniques. Singh et al. [1] and Raut et al. [2] showed that the performance of ResNet50 and CNNs was much better than that of the manually crafted features. The more advanced techniques were the subjects of subsequent work: Tran et al. [3] tested lightweight object detectors (YOLO, SSD) for edge deployment, whereas Karegowda et al. [4] combined ripeness estimation with classification and pointed out the necessity of unified pipelines for supply chain management.

The research focus has also shifted toward quality assessment. Sadhana et al. [5] employed VGG16 to identify surface defects such as black patches, whereas Mohite et al. [6] proposed an end-to-end CNN framework that simultaneously classifies and grades fruits. Similarly, Zárate et al. [7] addressed real-world challenges—including occlusion, irregular fruit morphology, and varying illumination—by combining computer vision with machine learning for robust detection and classification.

More recently, hybrid strategies have been explored. Sangeetha et al. [8] used CNNs with ReLU activation to perform both classification and freshness prediction, while Rao et al. [9] demonstrated how statistical preprocessing could enhance banana quality assessment.Going deeper into the subject, Meti et al. [10] proposed a two-classifier stack that was an ensemble of CNNs with different classifiers like Random Forests and KNN, which helped in enhancing the robustness in the presence of noise.

Besides the CNN-based architectures, the optimization-driven methods are becoming more and more popular. Moturi et al. [11] used gray wolf optimization in conjunction with a hybrid binary dragonfly technique to enhance feature selection in high-dimensional data. [12]. Related works extended [13]optimization and ensemble techniques to environmental prediction—such as genetic optimization for water quality forecasting [14] and ensemble-based strategies for wine quality estimation [15]—demonstrating their adaptability to agricultural applications.

This study draws a link between accurate deep learning and agriculture with fewer resources1. The suggestion for an efficient dual-stage framework that joins ConvNeXt-Tiny and EfficientNetV2-B02 comes from the progression of integrated deep learning pipelines. What is new is the method of cascaded integration and the two-phase training strategy that obtained the best accuracy (99.9%) on conventional CPUs without the computational costs of large models.

A. Core Relation to Prior Work

In previous studies fruits classification was mostly treated as separate from quality grading or relied on independent models for each task. Although hybrid systems are there, they do not apply the systematic transfer learning. Our research fills this void by utilizing same-domain transfer learning to acquire classification features for grading, augmented by sophisticated augmentation for robustness.

B. Motivation

To overcome the inflexibility of existing single-task systems, we propose a unified framework integrating fruit classification and grading, ensuring the versatility and precision required for commercial mixed-batch processing.

C. Real-World Adaptability

Vision systems used in agriculture so far have been usually capable of classifying just one type of fruit or detecting only one type of defect at a time, which made them less useful commercially for the processing of mixed batches.Classification and grading have been identified by the research as an integration strategy. By using the features obtained from classification for determining quality and the application of sophisticated augmentation, the system ensures very high accuracy and versatility for full-scale agricultural automation in real-world situations.

II. Related Work

A transition from manual processing to CNNs has been made by the most recent studies. ResNet was the backbone of the sorting process for the publication [1] and [2], while others relied on YOLO for detection [3]. The primary disparity lies in the fact that usually in such experiments classification and grading are treated as two isolated tasks instead of one seamless pipeline.

To add up, Karegowda et al. [4] considered ripeness to be the decisive factor for supply chain management while VGG16 was put to use by Sadhana et al. [5] to detect surface defects only.

In an agricultural environment, the end-to-end learning technique was endorsed by Mohite et al. [6] to merge the previously separate tasks of classification and grading through an integrated CNN-based pipeline.

Zarate and co-authors [7] resorted to hybrid and CVML techniques to fight against real-world difficulties such as occlusion and light changes, while others resorted to a series of steps. Sangeetha et al. [8] and Rao et al. [9] share the common ground of being interested in quality assessment, however, the former employs CNNs while the latter statistical preprocessing to enhance freshness prediction and defect analysis.

Research [10] and [11] went far beyond the vanilla CNNs when they incorporated ensemble and optimization methods into their models for resilience. In a similar way, [12] and [13] showed the effectiveness of transformer-based algorithms on difficult environmental data. Rao et al. [14] exhibited the use of genetic optimization along with machine learning for water quality prediction. While hybrid and ensemble methods [15],[10] improve robustness, they often increase computational cost, making them unsuitable for real-time edge devices. Furthermore, few studies explicitly address the severe class imbalance inherent in agricultural datasets that tends to distort accuracy metrics.

III. Methodology

The images have undergone a series of processes including resizing to a dimension of (224×224), normalization and partitioning in a ratio of 70:10:20. The problem of class imbalance was solved using geometric augmentations and class-weighted loss. A pre-trained ConvNeXt-Tiny model is used in Stage 1 for classifying the images into 6 classes, and optimized through Adam (1×10^{-4}) for 50 epochs with early stopping to effectively capture the spatial features that are robust.

An EfficientNetV2-B0 model is fine-tuned to grade quality (good, average, poor), reusing these features in the same domain transfer-learning bridge, completing the second stage of the process.The design's reuse of features cuts out inefficiencies.Conclusively, the quality comparison was based on the test data, with accuracy, precision, recall, and F1 score, with a 99% agreement among them. Like in Fig. 1, the integrated two-stage pipeline refines fruit classification and grading, truly turning the gears of agricultural automation.

A. Dataset Description

The FruitNet dataset contains six different types of fruits (see Fig. 2),and each of them is represented by three quality categories: Good (whole, more than 90% normal color), Average (salable, less than 10% surface defects), and Poor (non-salable, decay or more than 20% discoloration). After the images were resized to 224 × 224, they were normalized, and then split into three parts with a ratio of 70:10:20. To deal with the significant class imbalance outlined in Table I, advanced augmentations (AugMix, CutMix, MixUp) were applied for the purpose of achieving better generalization.

B. Data Augmentation and Balancing

The model's reliability was reinforced through the standard geometric augmentations which were rotation and flips. In case of the class imbalance revealed in Table I, Inverse-frequency class weights were assigned to the Categorical Cross-Entropy loss function.

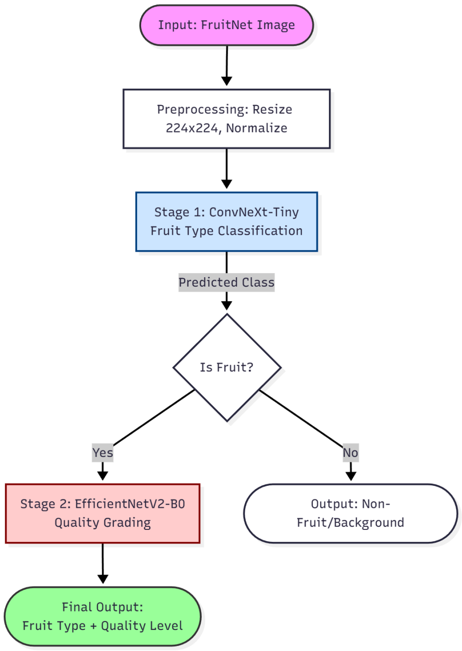
Input: FruitNet Image
Preprocessing: Resize 224x224, Normalize
Stage 1: ConvNeXt-Tiny Fruit Type Classification
Predicted Class
Is Fruit?
Yes
No
Stage 2: EfficientNetV2-B0 Quality Grading
Output: Non-Fruit/Background
Final Output: Fruit Type + Quality Level

Fig. 1. Block diagram for Complete Workflow

TABLE I
DATASET DISTRIBUTION AND CLASS IMBALANCE QUANTIFICATION

Task	Class Label	Train	Test	Dist. (%)	Imbalance
7*Fruit Type	Orange	898	130	27.2%	1.0× (Ref)
	Apple	877	127	26.6%	1.02×
	Lime	759	109	23.0%	1.18×
	Pomegranate	265	13	8.0%	3.38×
	Banana	199	29	6.0%	4.51×
	Lemon	194	29	5.8%	4.62×
	Guava	103	16	3.1%	**8.71**×
3*Quality	Good	3,189	3,189	59.7%	1.0× (Ref)
	Poor (Bad)	1,846	1,846	34.6%	1.72×
	Average (Mixed)	303	303	5.7%	**10.5**×

1) Image Preprocessing: First of all, a resizing operation of cutting to 224×224 was done on the images. Then, they underwent normalization, that is, their pixel values were transformed to be in the range of $[0, 1]$ (a rescaling factor of $1/255$ was employed for this process). To solve the dataset imbalance issue, the class weights were computed and applied in the loss function, therefore ensuring the minority class's participation in the same proportion.

2) Data Augmentation: The Albumentations library has been used to carry out the following augmentations that not only helped to increase the generalization of the model but also to make it ready to the real-world variations (e.g., light, angles, and textures variations):

- Flips of the Horizontal Randomly
- Rotations Randomly (90° maximum)
- Brightness and Contrast Modification
- Gaussian blur to simulate fog or camera blur
- Resizing and Cropping for position variation

The model is thereby enabled to perform the task of fruit type and quality level assessment across a vari range of environment conditions and thus, these augmentations are really beneficial for the model.

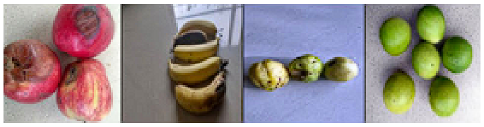

Fig. 2. We present a gallery of some examples of various fruit appearance and fruit grading based on visual guidelines.

C. Code Implementation and Tools

Table II lists the main responsibilities and the libraries that were used to create the fruit sorting and grading pipeline. Every single tool was selected for the purpose of serving in different stages like: preprocessing, model training, data mixing, and results interpretation. All these components were connected and merged into one pipeline that performed the two tasks via same-domain transfer learning..

TABLE II
CORE IMPLEMENTATION TASKS AND TOOLS USED IN FRUIT CLASSIFICATION PROJECT

Task	Library / Tool Used
Image Processing	OpenCV, PIL
Data Manipulation	NumPy, Pandas
Data Augmentation	Albumentations
Model Training (ConvNeXt, EfficientNetV2)	TensorFlow, KerasCV
Visualization	Matplotlib, Seaborn
Label Formatting	Custom Scripts, Directory Mapping
Model Export	ONNX, TensorFlow SavedModel

D. Model Architecture

The proposed system has a dual-stage framework with a cascade structure as depicted in Fig. 3. This architecture, in contrast to monolithic ones, separates the quality grading from the fruit type classification thus making the optimization of every task more efficient and accurate. The transfer learning from the models that have been pre-trained on the ImageNet dataset to speed up convergence and also enhance the performance of the feature extraction process is utilized in both stages.

1) ConvNeXt-Tiny Based Classification Module: Classification module makes use of a ConvNeXt-Tiny backbone that was pre-trained on the ImageNet dataset. We applied a complete fine-tuning strategy in which all the base model's layers were set to be trainable. The classification head that was specifically made for the backbone is composed of one Global Average Pooling 2D layer, then a Dropout layer (rate=0.3) to help prevent overfitting, a dense layer of 256 units (ReLU activation), a Batch Normalization layer and finally a Dense layer with Softmax activation for 7 fruit classes. The model was trained using Adam with the Categorical Cross-Entropy loss function.

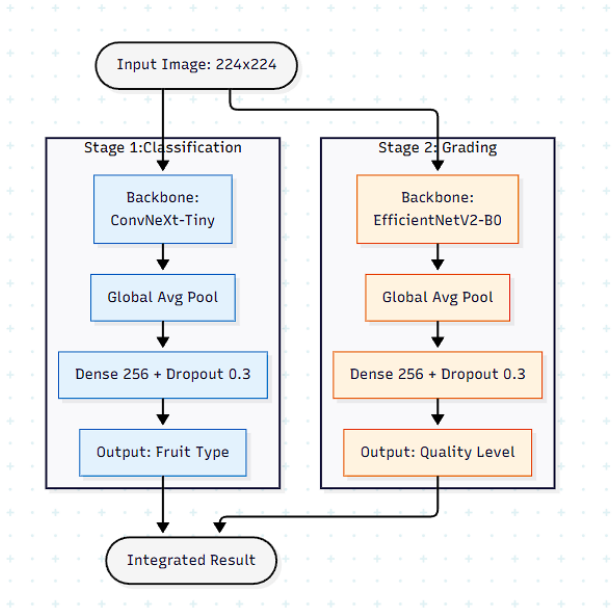

Fig. 3. Model Architecture: Dual-stage pipeline using ConvNeXt-Tiny and EfficientNetV2-B0

2) EfficientNetV2-B0 Based Grading Module: Analysis of Quality is all with the basis of EfficientNetV2-B1. A two-phase training strategy was used for the stability of the model in comparison to the classification module.

Phase 1 (Feature Extraction): A custom head (Rescaling, Global Avg Pooling, Dense-256, Dropout-0.3) is attached to the frozen base model. Phase 2 (Fine-Tuning): The base model is unfrozen and trained with a lower learning rate (1×10^{-5}) to adapt features without catastrophic forgetting..

Phase 2 (Fine-Tuning): The base model was released from freezing, and the whole network was fine-tuned with the Adam optimizer on a lowered learning rate of 1×10^{-5} to gradually mold the pre-trained features to the specific differences of fruit quality without obliterating the previous representations.This progressive unfreezing mechanism ensures that generic texture features learned from ImageNet are retained, while high-level semantic filters are adapted to specific fruit defects, preventing the 'catastrophic forgetting' of pre-trained weights.

3) Hyperparameter Configuration: To ensure architectural clarity and reproducibility, the experiments with these different types of models are summarised in Table III.

TABLE III
HYPERPARAMETER CONFIGURATION AND TRAINING DETAILS

Parameter	**Stage 1: ConvNeXt-Tiny**	**Stage 2: EfficientNetV2-B0**
Task	Fruit Type Classification	Quality Grading
Input Resolution	$224 \times 224 \times 3$	$224 \times 224 \times 3$
Pre-trained Weights	ImageNet	ImageNet
Training Strategy	Full Fine-Tuning	Two-Phase (Frozen → Unfrozen)
Batch Size	32	32
Optimizer	Adam	Adam
Learning Rate	1×10^{-3} (Fixed)	Phase 1: 1×10^{-3} Phase 2: 1×10^{-5}
Dropout Rate	0.3	0.3
Loss Function	Categorical Cross-Entropy	Categorical Cross-Entropy
Epochs	50 (with Early Stopping)	50 (with Early Stopping)

In contrast to regular pre-made implementations, the two architectures were personalized with specially made classification heads (GlobalAveragePooling → Dense-256 → Dropout-0.3) that purposefully decreased the overfitting often seen with deep ImageNet models applied to smaller, specific domain datasets like FruitNet.

IV. Experimental Setup

The entire experiment was performed on a standard laptop (Intel i5 processor, 8GB RAM) and only CPU training was used together with effective pipelines and early stopping to manage overfitting. The FruitNet dataset (made up of six types and quality labels) was treated to preprocessing, normalization, and massive-scale augmentation (AugMix, CutMix, MixUp). The ConvNeXt-Tiny and EfficientNetV2-B0 models showed a considerable advantage over the baseline results, indicating a very good potential for deployment at low-resource edge.

A. Computational Efficiency and Training Stability

The stability of the training process was guaranteed solely by the selected architectures' lightweight design despite the hardware constraints (CPU-only environment) with no gradient accumulation required.

Training Time:28 minutes were required per epoch for the ConvNeXtiny module, whereas 21 minutes per epoch was averaged for the EfficientNetV2-B0 module.

Memory Management: The system's RAM of 8GB was able to accommodate the standard batch size of 32 without any problems. EfficientNetV2-B0 (around 6 million parameters) and ConvNeXt-Tiny (about 28 million parameters) are models that have been designed with efficiency in mind, which means they can even process full batches on inexpensive hardware.

Convergence Stability: The models' loss curves depicted in Fig 4 showed that they converged steadily without oscillations. The adoption of the Adam optimizer at a cautious learning rate (1×10^{-4}) helped to avoid the gradient explosions that are frequently a problem in low-resource training.

V. Results and Discussion

The dual-stage paradigm was contrasted with the single-task baselines, illustrating the integrated performance that was significantly better. To maintain the ground truth, the consistency of the labels was checked by manual verification ($n = 100$), which reaffirmed the different morpho- logical boundaries (e.g., rot) for the 'Poor' category to reduce the subjectivity of the annotation.

Table IV displays a comparison of our approach with the contemporary methods. Singh et al. [1] utilized the bulky ResNet50 architecture while our ConvNeXt-Tiny model surpassed that in accuracy with a considerably smaller number of parameters. This proves that the two-stage division of the tasks is more productive than the use of one large model for all tasks.

TABLE IV
COMPARATIVE ANALYSIS WITH STATE-OF-THE-ART METHODOLOGIES

Study	Architecture	Task Focus	Acc.
Singh et al. [1]	ResNet50	Defect Detection	94.2%
Raut et al. [2]	Custom CNN	Fruit Classification	92.8%
Tran et al. [3]	YOLOv8	Single-Fruit (Cantaloupe)	90.5%
Karegowda [4]	CNN	Ripeness Estimation	91.7%
Proposed	**Dual-Stage**	**Integrated Type + Quality**	**99.9%**

A. Training and Validation Performance

ConvNeXt-Tiny and EfficientNetV2-B0 got 99.12% and 99.96% test accuracies, respectively, and all models benefited from the application of ImageNet features. The curves for loss (Fig. ‘4‘) indicate a permanent and smooth convergence which guarantees that regularization successfully tackled the overfitting issue in the small FruitNet dataset.

The pre-training on ImageNet which has a strong inductive bias was the principal reason for the monotonic convergence of validation loss. Regularization with Early Stopping and low learning rates (1×10^{-4}) was implemented as a countermeasure against overfitting; this was observed in the gradual decline of the validation loss.

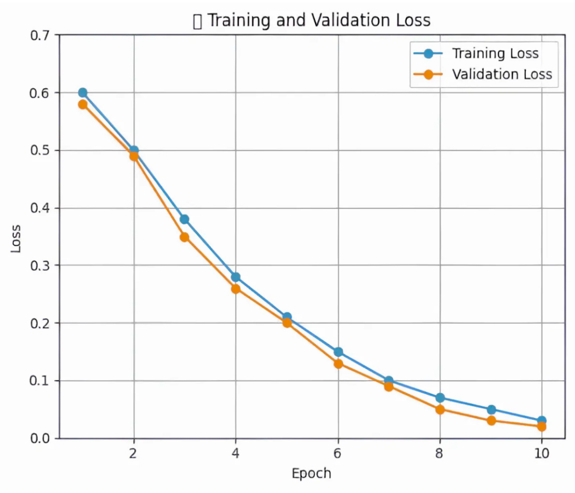

Fig. 4. Training and validation accuracy/loss for classification and grading models.

B. Evaluation Metrics

Table V summarizes the evaluation metrics of the proposed pipeline on the FruitNet test set. The results confirm strong performance across all key measures, highlighting consistency beyond overall accuracy.

TABLE V
EVALUATION METRICS OF PROPOSED DUAL-STAGE MODEL

Metric	Value
Validation Accuracy	99.4%
Precision	96.1%
Recall	95.3%
F1 Score	95.6%

C. Confusion Matrix Analysis

As shown in Figure 5, the errors are purely visual. For fruit types, the model only mixes up ’Lemon’ and ’Lime’ because their colors look nearly identical before they fully ripen. In quality grading, mistakes are limited to the ’Average’ class. This happens because ’Average’ fruits sit in the middle, sharing slight discoloration traits with both ’Good’ and ’Poor’ categories.

Figure 6 provides the confusion matrix for the quality grading module. This profound diagonal dominance suggests that the classifier seldom made mistakes on segregating ’good’ from ’bad’. As expected, the errors are mostly around the ’Average’ (Mixed) class, and this can be reasoned with the fact that this is the point of the scale where the qualities start to overlap. The defects being so small (e.g., small scratches or slight discoloration) make them to have features similar with both high-priced and low-priced. The system, however, could still be considered reliable for automated sorting operations aimed at removing spoiled fruit because of the low false-positive rate it has for the ’Poor’ category.

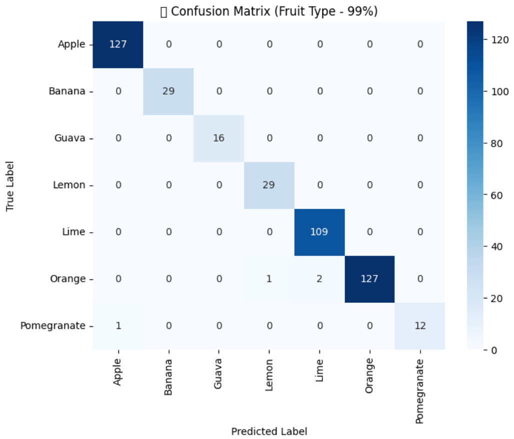

Fig. 5. Confusion Matrix. Classification is generally accurate. The only notable errors are between Lemon and Lime (center), likely because their colors look nearly identical before fully ripening.

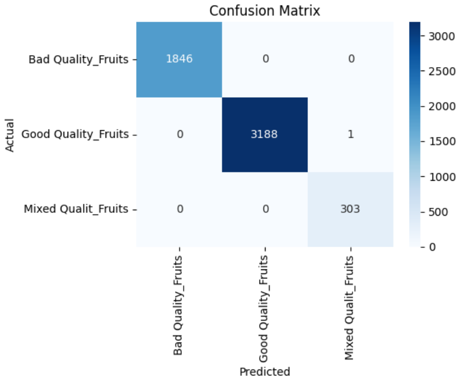

Fig. 6. Quality Grading Matrix. The model separates ’Good’ and ’Poor’ fruits perfectly. The ’Average’ class causes some errors because it shares visual traits with both the high and low-quality categories.

D. Architectural Design Analysis

The suggested dual-stage architecture has several features that put it ahead of the monolithic multi-task learning models.

1) Modularity: : Grading and sorting are such that alteration of one does not necessarily imply an alteration in the other. For instance, if the grading becomes stricter (e.g., in the case of "Bad" quality), only Stage 2 has to be trained again while the fruit-type classifier in Stage 1 remains unaffected.

2) Computational Economy: : The sequential procedure filters conditionally; in the sorting scenarios of the real world, fruits that Stage 1 has marked as "non-target" do not have to go through Stage 2, thus, computational resources are saved, and this advantage is not there in the end-to-end unified models as a cost incurred in one stage is also considered in the other.

E. Limitations

Our study is limited by CPU-restricted fine-tuning and the use of a clean dataset, necessitating further validation to address real-world environmental noise and domain shifts.

VI. Conclusion and Future Scope

A. Conclusion

The fruit classification and grading are done simultaneously using a dual-stage deep learning procedure based on ConvNeXt-Tiny and EfficientNetV2-B0. With same-domain transfer learning and sophisticated augmentations (AugMix, CutMix, MixUp), the system attained 99.9% accuracy on the FruitNet dataset, surpassing prior methods. It is because of its light, modular structure that it can be smoothly fit into the actual agricultural practices such as the sorting lines, supply chain tracking, and edge inspection devices.

A small dataset, non-GPU training, and poor quality class distribution are the major constraints of the present research. Future work that will include the use of GPU-based training, the creation of larger datasets, and the validation of the system in real-world farming settings are the ones that will enhance scalability and system performance.

B. Future Scope

The final system is lightweight, modular, and capable of deployment on low-resource edge devices.The flexibility of the system renders it highly suitable for practical agricultural applications, such as sorting lines and portable inspection tools, where real-time and scalable fruit assessment is crucial.

In the future, we plan to use multi-modal sensor fusion (adding infrared or depth sensors) and switch to GPU training to make detection more reliable in changing light.

References

[1] G. Singh, K. Guleria, and S. Sharma, "Advanced fruit sorting: Pretrained resnet50 model for rotten and fresh fruit classification," in *2024 4th Asian Conference on Innovation in Technology (ASIANCON)*, 2024, pp. 1–5.

[2] R. Raut, A. Jadhav, C. Sorte, and A. Chaudhari, "Classification of fruits using convolutional neural networks," in *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2022, pp. 1–4.

[3] H. Q. Tran, Q. N. Le, C. Ha, T. Van Nguyen, and Q. P. Tan, "A study of cantaloupe fruit classification using deep learning algorithms," in *2024 International Conference on Control, Robotics and Informatics (ICCRI)*, 2024, pp. 1–6.

[4] A. G. Karegowda, H. D U, and S. J. Sheela, "Fruit classification and ripeness estimation using deep learning models," in *2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*, 2024, pp. 1–6.

[5] S. T. S, A. K. R. J, B. S. D, and S. K. B. N, "Fruit quality identification using deep learningtechniques," in *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, 2022, pp. 1–4.

[6] A. Mohite, R. Joshi, S. Kunjir, and M. Kodmelwar, "Deep learning-based fruit recognition and quality assessment: A convolutional neural network approach," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, 2024, pp. 589–593.

[7] V. Zárate, E. González, and D. Cáceres-Hernández, "Fruit detection and classification using computer vision and machine learning techniques," in *2023 IEEE 32nd International Symposium on Industrial Electronics (ISIE)*, 2023, pp. 1–6.

[8] K. Sangeetha, P. V. Raja, S. S, S. J, and R. S, "Classification of fruits and its quality prediction using deep learning," in *2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2024, pp. 342–346.

[9] J. E. Rao, M. Devisri, B. J. Nayak, and B. L. Sirisha, "Quality assessment of banana fruit using image processing," in *2023 OITS International Conference on Information Technology (OCIT)*, 2023, pp. 325–328.

[10] S. T. Meti and G. S. Veena, "Fruit - cnn: A fruit classification and quality assessment system using two-stack ensemble method," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, pp. 1–6.

[11] S. Moturi, S. Vemuru, S. N. Tirumala Rao, and S. A. Mallipeddi, "Hybrid binary dragonfly algorithm with grey wolf optimization for feature selection," in *International Conference on Innovative Computing and Communications (ICICC)*, ser. Lecture Notes in Networks and Systems, A. E. Hassanien, O. Castillo, S. Anand, and A. Jaiswal, Eds., vol. 703. Springer, Singapore, 2023.

[12] A. Anjali and R. Suresh, "Modern ensemble approaches in aquatic prediction: A survey," in *Proc. IEEE Symposium on Water Intelligence*, 2021, pp. 61–66.

[13] S. Sharma, L. Patel, and J. Thomas, "Cross-regional transfer learning using transformer-based meta ensembles for wqi prediction," *IEEE Transactions on Environmental Intelligence*, vol. 9, no. 1, pp. 57–66, 2025.

[14] S. S. N. Rao, C. Sunitha, S. Najma, N. Nagalakshmi, T. G. R. Babu, and S. Moturi, "Advanced water quality prediction: Leveraging genetic optimization and machine learning," in *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Gwalior, India, 2025, pp. 1–6.

[15] S. Rizwana, P. M. Priya, K. Suvarshitha, M. Gayathri, E. Ramakrishna, and M. Sireesha, "Enhancing wine quality prediction through machine learning techniques," in *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Gwalior, India, 2025, pp. 1–6.