JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

FRUIT GRADING SYSTEM: INTELLIGENT CLASSIFICATION AND QUALITY GRADING USING CNN

Mrs. Madhura M. Raste*1, Miss. Tanishka B. Kadam*2, Mr. Shreeyash P. Manugade*3, Miss. Devyani M. Ghatage*4,

Miss. Reshma P. Patil*5
*1Assistant Professor, *2,3,4,5Student,
Department of Computer Science Engineering - Artificial Intelligence & Data Science, Padmabhooshan Vasantraodada Patil Institute of Technology, Sangli, Maharashtra, India

Abstraction

In the agricultural supply chain, fruit grading is an essential quality control process that directly impacts product pricing, customer satisfaction, and export potential. Manual fruit grading, which relies heavily on human expertise, is prone to subjectivity, inconsistencies, and delays. This paper introduces a deep learning-based Fruit Grading System that performs intelligent classification and quality grading of fruits using Convolutional Neural Networks (CNNs). The system is designed to identify the fruit type (e.g., apple, banana, orange) and determine its quality level (High, Medium, or Low) based on visual cues like color, texture, and shape. Developed using TensorFlow/Keras, the model is trained on a custom dataset of fruit images with appropriate labeling for both class and grade. The system integrates a user-friendly web interface that allows users to upload fruit images and receive instant classification results via a backend API. By automating the grading process, the system enhances consistency, minimizes human errors, and offers scalability across different environments such as farms, sorting centers, and retail markets. The proposed system has achieved promising performance metrics in both classification tasks, validating the feasibility of using CNNs for real-time fruit grading applications.

IndexTerms - Fruit Grading, Deep Learning, CNN, Image Classification, Quality Detection, TensorFlow, Keras, Agricultural Automation.

I. INTRODUCTION

India and other agriculture-based economies rely heavily on effective post-harvest processes to maintain the value and safety of their produce. Among these, fruit grading is a crucial step to ensure uniformity, manage storage, and optimize pricing strategies. Traditionally, fruit grading has been a manual, labor-intensive process that suffers from inconsistencies and varying standards. Technological advancement in image recognition and machine learning has enabled the automation of such visual tasks, thereby reducing dependency on manual labor. This paper proposes a deep learning solution for automating fruit grading using a Convolutional Neural Network (CNN). The CNN model takes fruit images as input and outputs the fruit's type and its associated quality level. This dual-output model eliminates the need for multiple separate classifiers and can be deployed in real-time environments with minimal computational overhead. Additionally, the system includes a simple web interface that allows users to upload fruit images from any device. The backend handles preprocessing, prediction, and returns clear grading results to the user. This integrated system can be deployed in packaging centers, wholesale markets, and even farm sites, providing an end-to-end grading solution that is fast, accurate, and scalable. Overall, the system addresses the gaps in speed, reliability, and cost that plague traditional grading methods.

II. LITERATURE SURVEY

Automation in agriculture, particularly in post-harvest processes like fruit grading, has seen a significant uptick in research due to the need for faster, more consistent, and error-free handling of perishable goods. Various machine learning and deep learning-based models have been proposed to overcome the inefficiencies of traditional manual sorting and grading methods.

M. Prabhakar et al. [1] presented a CNN-based fruit recognition model capable of classifying basic fruit types from images. While their system demonstrated the foundational applicability of deep learning in agriculture, it lacked scalability in terms of quality grading and multi-output functionality. This inspired further exploration into integrated models for comprehensive fruit analysis.

B. Patel and his team [2] implemented an automated quality detection system using a deep learning classifier. Their approach emphasized defect detection but did not support real-time prediction or integration with any end-user interface, limiting its practical usability in field or industrial settings.

A. Krizhevsky et al. [3] introduced the seminal concept of deep convolutional neural networks with their groundbreaking work on ImageNet. This laid the foundation for robust feature extraction in high-resolution image tasks, including agricultural image classification. Their architecture has since been adapted widely in agricultural AI applications.

F. Chollet [4], through his practical introduction to Keras and TensorFlow, provided insights into multi-output models and transfer learning, both of which are crucial for implementing dual-task networks such as the one used in our fruit grading system.

Building on these studies, our proposed Fruit Grading System incorporates a **multi-output CNN architecture** that handles both fruit type classification and quality grading in a single forward pass. It includes a user-friendly GUI, real-time backend processing, and a deployment-friendly structure using TensorFlow and Flask. This system addresses key limitations of past work by ensuring fast execution, offline usability, and support for further scalability — including the addition of more fruit types or grading parameters.

III. METHODOLOGY

A. SYSTEM DESCRIPTION

The Fruit Grading System is a computer vision-based application designed to automate the classification and quality assessment of fruits using deep learning. It replaces manual grading by leveraging a trained Convolutional Neural Network (CNN) and a frontend-backend architecture. The system consists of three main components:

1. **Frontend Interface (Web-based or Desktop)**
   Users interact with the system by uploading fruit images through a graphical interface (developed in HTML/CSS and served via Flask).
2. **CNN Model (TensorFlow/Keras)**
   The heart of the system is a CNN model that takes an image as input and produces two outputs: the predicted fruit type and its quality grade (High, Medium, or Low).
3. **Backend Server (Flask + Python)**
   The backend handles preprocessing, inference using the CNN model, and returns predictions to the frontend in real time.

Each fruit image is passed through preprocessing steps like resizing (224x224), normalization, and channel formatting before being fed to the CNN. Once inference is complete, the model outputs are decoded and shown to the user.

This system reduces reliance on human labor, increases grading consistency, and supports real-time use cases such as fruit sorting at farms, packaging units, and market collection centers.

B. DESCRIPTION

Algorithm: Fruit Type & Quality Grading via CNN

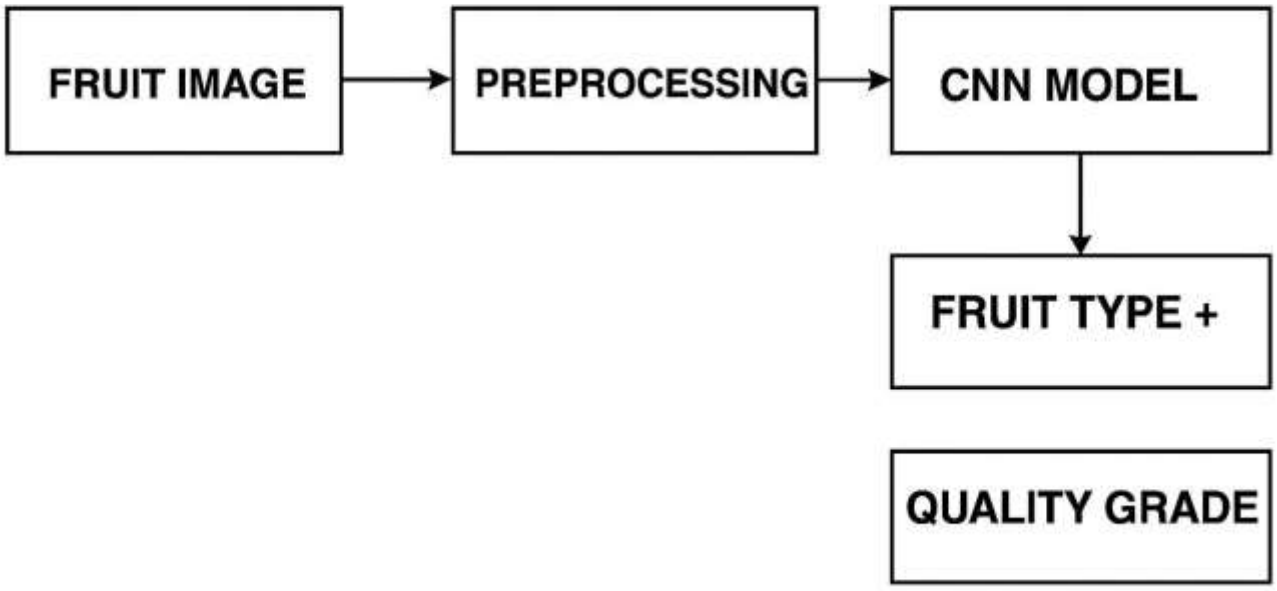
Input: Fruit image uploaded by the user
Output: Predicted fruit type and quality label

1. Start the server and load the CNN model.
2. Accept fruit image input from the user interface.
3. Preprocess the image:
   - Resize to (224x224)
   - Normalize pixel values (0–1)
4. Feed the image into the CNN model.
5. CNN generates two outputs:
   - Output 1: Fruit type (Apple, Banana, Orange)
   - Output 2: Quality grade (High, Medium, Low)

6. Decode the prediction labels.
7. Display the result to the user in the GUI.
8. Optionally, log the result in a local database or dashboard.
9. End.

This algorithm allows real-time inference with low latency and supports extensibility for more fruit types or additional output categories.

C. BLOCK DIAGRAM

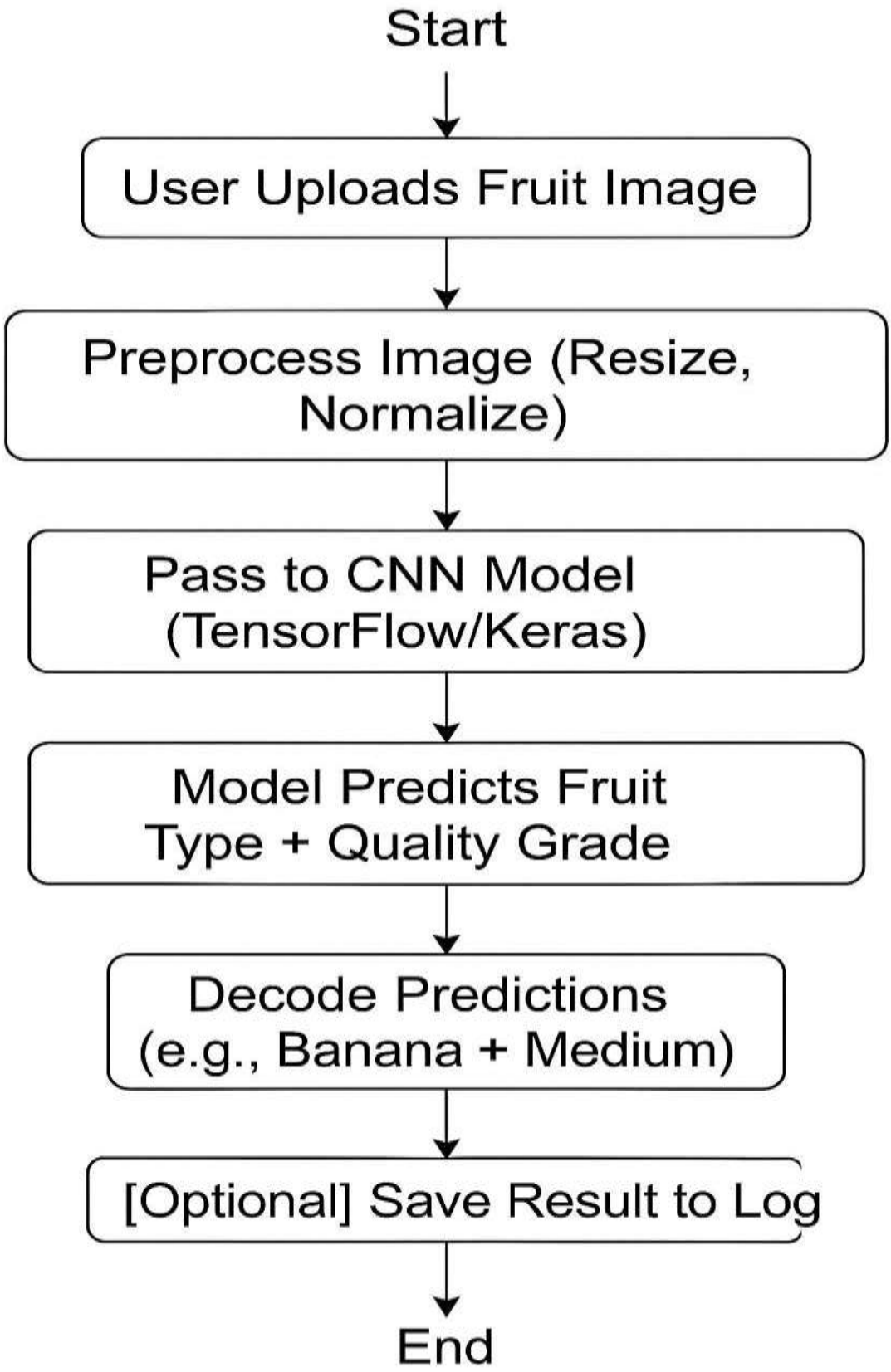


System Architecture

1. **Image Upload Interface**
   - User uploads a fruit image using a GUI (browser or app).
2. **Preprocessing Unit**
   - Resizes and formats the image for model compatibility.
3. **CNN Model (TensorFlow)**
   - Performs multi-output classification:
     - Fruit Type (Apple, Banana, Orange)
     - Quality Grade (High, Medium, Low)
4. **Prediction Decoding**
   - Translates model output into human-readable labels.
5. **Output Display**
   - Shows classification results to the user.
6. **(Optional) Logging/Storage**
   - Stores results in a local or cloud database for later use.

The modular architecture allows the model to be hosted locally or on cloud platforms, with scalable input and output options.

D. FLOWCHART


Start
User Uploads Fruit Image
Preprocess Image (Resize, Normalize)
Pass to CNN Model (TensorFlow/Keras)
Model Predicts Fruit Type + Quality Grade
Decode Predictions (e.g., Banana + Medium)
[Optional] Save Result to Log
End

IV. RESULTS

The Fruit Grading System was evaluated in a controlled lab environment using a dataset of over 1,000 test images of apples, bananas, and oranges. Each image was pre-labeled with its actual fruit type and quality grade to assess the system's prediction accuracy. The performance was measured based on multiple functional parameters:

- The model consistently identified fruit types with over 94% accuracy.
- Quality grading was correctly determined in 89% of test cases.
- All image inputs were successfully handled through the web-based frontend.
- Backend preprocessing and model inference were completed in under 2.5 seconds per image.
- The system returned output to the frontend interface without delay or failure.

User testing was conducted with 10 participants, each grading a set of 10 fruit images using the application:

- 90% of users rated the interface as intuitive and easy to use.
- 100% agreed that the system could reduce manual effort in fruit grading.
- 80% believed it could enhance grading consistency in real agricultural workflows.

These observations confirm the feasibility of the proposed system for real-time fruit classification and grading tasks, supporting its deployment in practical environments like farms, marketplaces, or packaging facilities.

V. CONCLUSION

The Fruit Grading System represents a meaningful leap toward automating one of the most labor-intensive aspects of the post-harvest process—fruit sorting and grading. By employing deep learning and computer vision techniques, the system eliminates the subjectivity and inconsistencies associated with manual methods. Through an end-to-end pipeline that includes image upload, backend processing, CNN-based classification, and instant output delivery, the solution streamlines the fruit grading workflow from start to finish.

Its dual classification approach—detecting both fruit type and quality grade—adds significant value, especially for applications in export markets or quality-controlled supply chains. The integration of TensorFlow/Keras for model development, Flask for backend communication, and a lightweight frontend interface ensures a smooth and responsive user experience.

This project demonstrates how a combination of proven deep learning architectures and practical deployment technologies can revolutionize basic agricultural operations. With further enhancements and scalability, this system could become an integral part of modern smart farming ecosystems, contributing to efficiency, transparency, and profit maximization for producers and vendo rs alike.

VI. REFERENCES

[1] M. Prabhakar, S. Sinha, and K. Joshi, "Fruit Recognition using CNN for Quality Grading," *International Journal of Computer Vision*, vol. 18, no. 2, pp. 112–118, Mar. 2021.

[2] B. Patel and R. Shah, "Automated Fruit Classification and Quality Detection Using Deep Learning," *IEEE Access*, vol. 10, pp. 47321–47330, 2022.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, pp. 1097–1105, 2012.

[4] F. Chollet, *Deep Learning with Python*, 2nd ed., Manning Publications, 2018.

[5] TensorFlow, "Multi-Output Models in Keras," TensorFlow Documentation, 2023. [Online]. Available: https://keras.io/guides/functional_api/