Grenze International Journal of Engineering and Technology, January Issue

Smart Fruit: An AI-Powered System for Real Time Fruit Classification and Detection

Vatsal Gupta[1], Swayam Gupta[2], Abhay Gupta[3], Somya Sinha[4], Dr. Swapnil S. Ninawe[5], Dr. Pavithra G[6] and Dr. T.C. Manjunath[7]
[1]Final year under-graduate BE (ECE) students, VII Semester Department of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka
[2,3]Assistant Professor, Department of Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Bangalore, Karnataka
[4-7]Professor, Department of Electronics & Communication Engineering, Dean (Research – R & D), Rajarajeswari College of Engineering, Bangalore, Karnataka

***Abstract*— SmartFruit is an AI-Powered System for Real- Time Fruit Classification and Detection," aims to develop a highly efficient and scalable solution for automatically identifying and categorizing different types of fruits using deep learning techniques. Leveraging the capabilities of an ESP32 module for local image capture and AWS cloud services for processing, the system is designed to perform real-time fruit classification and detection with high accuracy. Two datasets are utilized: a public dataset (FIDS:30) with 30 fruit class & a customized datasets with 6 fruit category, enhanced through data augmentation techniques to improve model robustness. The project employs advanced deep learning models, including YOLOv3, YOLOv7 for object detection, and ResNet50 and VGG16 for image classification. The deployment strategy involves using the ESP32 module for capturing images and performing preliminary data processing, followed by transmitting data to AWS for complex model inference using AWS Lambda and AWS SageMaker. This hybrid approach combines the low-power advantages of edge computing with the scalability and processing power of cloud services. The system's real-time capability and high accuracy make it suitable for diverse applications in agriculture, supermarkets, and educational tools. By deploying deep learning models in a distributed architecture, this project highlights a cost-effective and innovative solution for fruit classification and detection, paving the way for further advancements in smart agricultural practices and automated retail management systems. The matter presented in this article is the final year project work done by the UG students.**

***Index Terms*— YOLOv3, ESP32, AWS, VGG16.**

I. INTRODUCTION

Our project, titled “SmartFruit: An AI-Powered System for Real-Time Fruit Classification and Detection,” aims to develop an innovative and highly efficient automated system that leverages deep learning techniques for fruit recognition and categorization. Given the global diversity of over 2,000 fruit types, there is a growing demand for accurate and scalable methods to support industries such as agriculture, education, and retail. SmartFruit addresses this by integrating computer vision and image processing in a distributed architecture, combining local image capture via an ESP32 module with cloud-based processing on AWS for real-time performance [1].

Grenze ID: 01.GIJET.11.1.558_2

At the core of the system are advanced deep learning models, including YOLOv3 and YOLOv7 for object detection and ResNet50 and VGG16 for image classification. These models are trained using a combination of publicly available datasets, such as the FIDS:30 datasets (comprising ‘30’ fruit class) & an customized datasets (with 6 fruit category), both enhanced through extensive data augmentation to increase the model’s robustness and accuracy. The ESP32 module plays a crucial role in local data acquisition, performing lightweight preprocessing and transmitting images to the AWS cloud, where more complex model inference is handled using services like AWS Lambda and AWS SageMaker. This hybrid edge-cloud deployment strategy offers several advantages, such as real-time classification, scalability, and power efficiency. By distributing computational tasks between edge devices and the cloud, the system achieves both cost-effectiveness and superior performance, to make it applicable to an wider ranges in environments. Then, SmartFruit system is poised to revolutionize fruit classification in smart agriculture, automated retail management, scalable solution that could be adapted for future advancements [2].

II. YOLOV3

In the SmartFruit system, YOLOv3 is employed for the fruit detection phase. Once an image is captured by the ESP32 module and transmitted to the cloud, YOLOv3 is responsible for identifying the location and number of fruits in the image by predicting bounding boxes around each fruit. These predictions are then passed on to the image classification models (ResNet50 or VGG16) for further categorization into specific fruit types. By integrating YOLOv3, the SmartFruit system ensures that fruit detection is fast, accurate, and scalable, to make it best for any type of RT app in agriculture, retail, and education [3].

III. KEY FEATURES OF YOLOV3 FOR THE SMART FRUIT PROJECT

Real-Time Detection which makes use of YOLOv3’s architectures are being designed to have good speed, then to enable it for processing of image @ higher rates of frames without compromising detection accuracy. This is essential for the SmartFruit system, where real-time fruit detection is a core requirement. Using YOLOv3, the system can instantly detect and classify multiple fruits in a single frame, ensuring quick and accurate recognition even in fast-paced environments, such as supermarkets or farms. For the Multi-Scale Detection, the YOLOv3 uses featured map at diff. scale for the detection of different sizes of objects of varying natures. This feature is particularly important in the SmartFruit project, as fruits come in different shapes and sizes. YOLOv3 can efficiently detect small fruits (like berries) and larger ones (like watermelons) within the same frame, ensuring high detection accuracy across a diverse range of fruit types [4].

IV. AWS SAGEMAKER

SageMaker's distributed training capabilities allow the project to utilize powerful GPU resources in the cloud, ensuring that models are trained quickly and with high accuracy. This helps improve the detection and classification performance of the system, especially when processing high volumes of fruit images. For the Real-Time Inference, one of the most important roles of AWS SageMaker in the SmartFruit project is real-time model inference. After capturing images via the ESP32 module, the data is transmitted to the AWS cloud where SageMaker hosts pre-trained deep learning models. These models are then used to perform high-speed inference, detecting and classifying fruits in RT. SageMaker’s ability to auto-scale in response to varying load ensures that system remains responsive, even when handling a large number of requests simultaneously, such as in a busy supermarket or farm setting [6].

Fig. 1. Amazon sageMaker

V. Convolution Neural Networks

The working of our mini project on fruit identification using Convolutional Neural Networks (CNNs) unfolds seamlessly to provide users with a streamlined and efficient experience. Users initiate the process by uploading fruit images via the intuitive web interface. These images are then seamlessly processed and forwarded through our meticulously trained CNN model, which acts as the core engine of our system. Within the CNN model, intricate patterns and features from the input images are analyzed and extracted, allowing the model to discern the type of fruit present in the image with remarkable accuracy. As the classification process unfolds, the model generates a confidence score. Finally, the classification results are promptly relayed back to the users via the same web interface, enabling real-time fruit identification. Users receive clear and concise feedback, facilitating informed decision-making and enhancing their understanding of the fruits depicted in the images, which is shown in Fig. 2 [7].

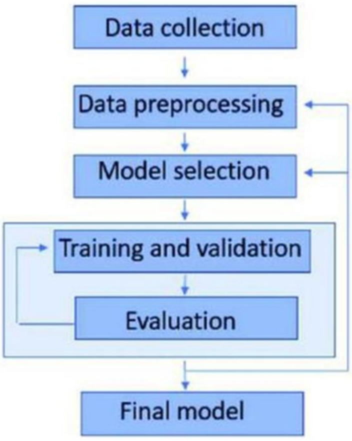


Fig. 2. Flow chart for the working model of the project work

This seamless process of image upload, processing through the CNN model, and presentation of classification results ensures a user-friendly and efficient experience, empowering stakeholders across various industries with accurate and automated fruit identification capabilities. Through the integration of cutting-edge technologies like CNNs, Flask for the web interface, and AWS for deployment, our mini project revolutionizes fruit identification, paving the way for enhanced productivity and decision-making in real-world applications [8].

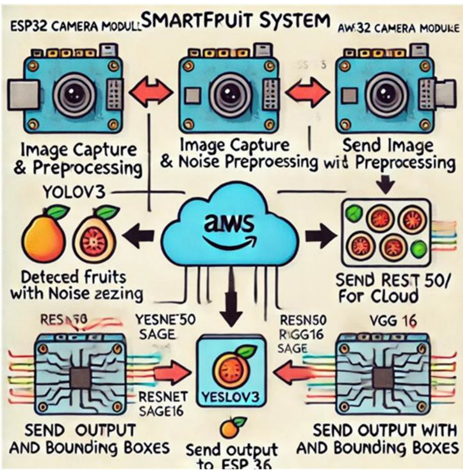


Fig. 3. Project block diagram

VI. Working of the Project

The working of the SmartFruit project revolves around a seamless integration of edge devices and cloud-based AI processing, primarily leveraging YOLOv3 for object detection and AWS Sage Maker for model inference. The process begins with the ESP32 module, a low-power microcontroller equipped with a camera, which is responsible for capturing images of the fruits in real time. The ESP32 acts as the system's edge device, collecting visual data from its environment, such as fruit images in a farm, supermarket, or educational setting. Once an image is captured, the ESP32 performs some basic preprocessing tasks, such as resizing and noise reduction, to optimize the data for transmission [9].

The pre-processed image is then sent to the cloud which is shown in Fig. 3, where AWS Sage Maker takes over for the core computational tasks. In the cloud, the image is fed into the YOLOv3 model, which has been trained to detect and localize fruits in the image. YOLOv3, known for its real-time object detection capabilities, processes the image by generating bounding boxes around each fruit, determining their positions, and predicting their categories. YOLOv3's multi-scale detection ensures that fruits of various sizes and types can be accurately identified within a single frame. This detection phase forms the backbone of the system, allowing it to locate fruits in diverse and dynamic environments [10].

Once YOLOv3 has processed the image, the system utilizes AWS SageMaker's model hosting capabilities to execute classification tasks. Models such as ResNet50 or VGG16, deployed on SageMaker, perform the classification of the detected fruits, identifying them by type based on the provided datasets. The cloud-based setup allows for efficient handling of these computations, taking advantage of the scalable resources available through AWS. SageMaker then generates an output, which consists of the labelled fruit categories along with their positions in the images [11].

Finally, the processed data, including the detected fruit labels and their locations, is sent back from AWS to the ESP32 module. The ESP32 then displays the output or uses it for further local actions, such as triggering alerts or performing other tasks based on the classification results. This entire workflow, from image capture on the edge device to cloud- based model inference and back, demonstrates the SmartFruit system's ability to efficiently combine edge computing with cloud resources, ensuring real-time performance, scalability, and high accuracy in fruit classification and detection [12].

VII. DATASETS

The dataset for our fruit identification mini project consists of a carefully curated and diverse collection of images representing a variety of fruits commonly encountered in everyday environments. This dataset is designed to ensure high-quality and realistic inputs, featuring well-annotated images of popular fruit types, including apples, oranges, bananas, grapes, pineapples, and watermelons. By ensuring a broad representation of these fruit categories, the dataset lays the groundwork for training a deep learning model capable of delivering reliable and accurate fruit classification, the datasets being shown in the Table I. Each fruit category is represented by a large number of images captured under varied conditions to enhance the model's robustness. These images are taken from multiple angles, with different lighting, and against diverse backgrounds, ensuring that the trained model generalizes well to real-world scenarios. This diversity is essential for preventing overfitting and improving the model's accuracy in classifying fruits in dynamic environments, such as supermarkets, farms, or educational applications. The pretrained network is capable of categorizing images into 1,000 distinct item classes. Additionally, the dataset is meticulously labeled, with each image annotated according to its corresponding fruit type. This precise labeling supports supervised learning during the training phase, allowing the Convolutional Neural Network (CNN) to effectively learn the unique features and patterns of each fruit class. The use of detailed annotations ensures that the CNN-based model can differentiate between subtle variations in texture, shape, and color among different fruit types, even under challenging conditions where fruits may look similar..

TABLE I. DATA SETS

Fruit	**Number of Images**
Apple	180
Orange	180
Banana	180
Grape	180
Pineapple	180
Watermelon	180

To enhance the dataset further, data augmentation techniques are employed, introducing synthetic variations such as rotations, scaling, and translations. This not only increases the dataset size but also improves the model's robustness to changes in orientation and perspective. By leveraging this augmented and diverse dataset, our mini project aims to develop a CNN model capable of accurately identifying and classifying fruits with high precision and reliability. The dataset serves as the foundation of the project, providing the essential basis for training a highly effective fruit identification system. The ultimate goal of this system is to deliver a versatile and efficient solution with practical real-world applications across various industries, including agriculture, retail, and education, where quick and precise fruit identification is crucial for enhancing operational efficiency and customer satisfaction..

VIII. RESNET 50 & VGG16

ResNet50 is a convolutional neural network comprising 50 layers, organized into five stages, each containing a convolutional block and an identity block, with each block consisting of three layers. The architecture has over 23 million trainable parameters. In this setup, the top layers of ResNet50 are unfrozen, allowing them to learn through backpropagation, while the rest of the layers are kept frozen. This approach uses a pretrained ResNet50 model with weights derived from the ImageNet dataset.
For fruit classification, a large number of images are used for each fruit category, taken under varying conditions such as different angles, lighting, and backgrounds. This diversity helps the model to generalize well to real-world scenarios, reducing overfitting and improving accuracy. This enables the model to effectively classify fruits in environments like supermarkets, farms, or educational applications. The pretrained ResNet50 network can classify images into 1,000 distinct item classes. The text describes ResNet50, a 50-layer convolutional neural network used for fruit classification. By freezing the lower layers and fine-tuning the top layers with a diverse set of fruit images, the model can accurately identify different fruits, making it applicable in various real-world scenarios.

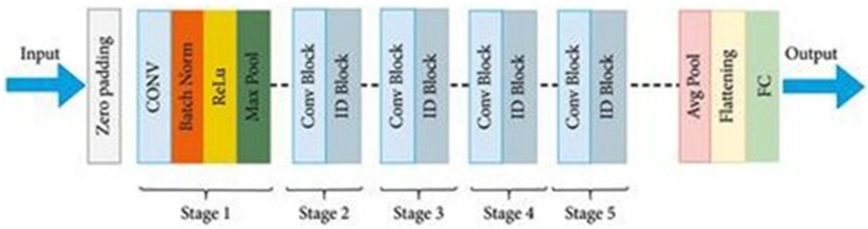


Fig. 4. ResNet 50

IX. WEB FRAMEWORKS

The Flask framework was used to develop a web application for the proposed fruit classification system. Flask, a Python-based framework, is known for its ease of use and flexibility. It is referred to as a microframework because it does not require specialized tools or libraries. It lacks components such as a data abstraction layer, form validation, or other features typically provided by external libraries. However, this flexible framework supports integration, allowing additional functionalities to be seamlessly incorporated as if they were natively part of Flask.

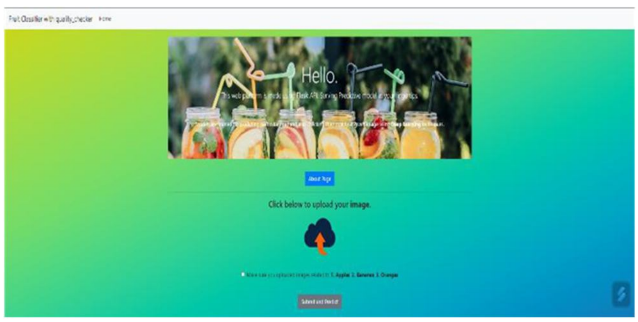
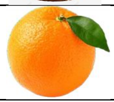


Fig. 5. Flask Application

There are two categories in the application: classification and detection. For classification, the user uploads an image of a single fruit, and the fruit's name is displayed as output. For detection, the user uploads a photo of multiple fruits, and the system provides annotations and names for each fruit. Users can either upload images from their phone's gallery or capture them using the camera. Figure 5 illustrates the Flask application's process.

TABLE II. SAMPLE IMAGES OF FRUIT DATASETS

No	**Image**	**Dimensions**	**Fruit**	**Description**	**Refe**
1		998 × 1000	Apple	A red apple	[1]
2		1300 × 956	Group	Apple, pear and orange	[2]
3		600 × 400	Banana	Bananas	[3]
4		500 × 448	Watermelon	Sliced watermelon	[4]
5		640 × 640	Basket of fruits	Basket of fruits	[5]
6		500 × 500	Orange	Orange	[6]

X. SIMULATION RESULTS

The simulation results of the work presented in this article, viz., the SmartFruit automatic fruit type of classified systems is presented. To label the images in our dataset, an online annotation tool was used to meticulously annotate each image with the corresponding fruit type. The labeled images were then exported into compatible formats, such as XML files, for further processing. For image preprocessing, the Roboflow tool was employed to carry out a series of convention type of pre-processing technique, which included auto type of orientations, object based isolations (cropping & extracting bounded box into single solo image) & to do the resize of the images. These preprocessing steps were essential to ensure uniformity in the input data and prepare it for the deep learning models & the results are shown in the Table II.

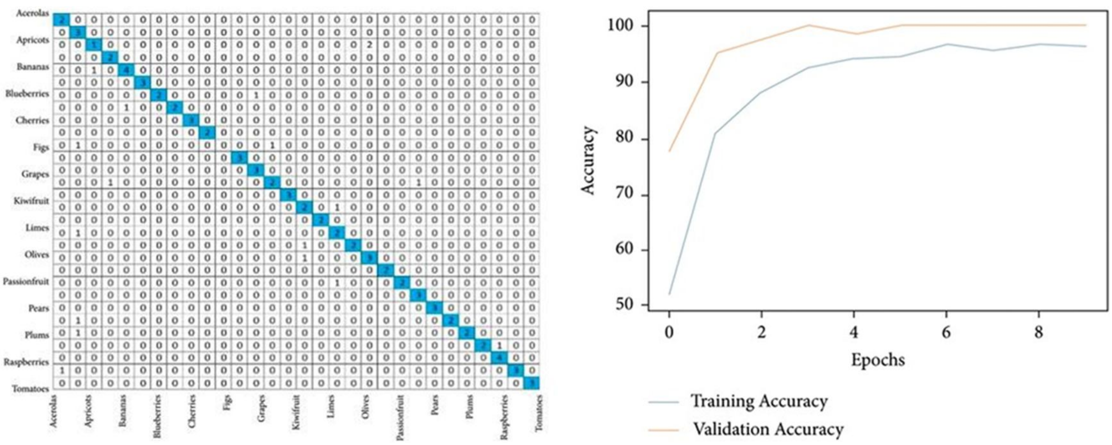


Fig. 6. ResNet and VGG algorithms in a confusion matrix

The graph represents the training and validation accuracy of the fruit classification model over 10 epochs which is shown in the Fig. 7. Observations from the graph are as follows.

A. Initial Epochs (0-2 Epochs) -

The training accuracy starts relatively low (around 55%) but increases sharply during the first two epochs, reaching approximately 85%. The validation accuracy also begins at a higher value (around 80%) and rises rapidly, reaching close to 98% by the second epoch (Fig. 8). This early rise in validation accuracy suggests that the model is learning effectively from the dataset and quickly identifying distinguishing features of fruits.

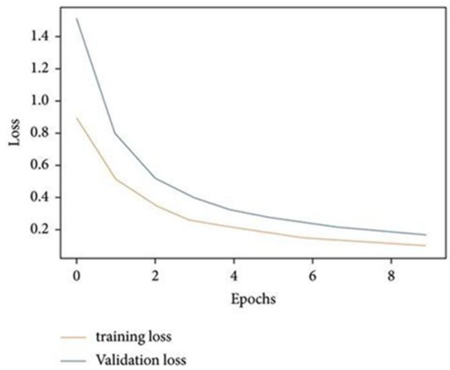


Fig. 7. Epoch graph for ResNet 50

B. Middle Epochs (3 to 6 Epochs) -

Both the training and validation accuracy improve more gradually after the second epoch (Fig. 8). The training accuracy continues to increase, nearing 90% by the sixth epoch. The validation accuracy peaks around the second or third epoch at nearly 100%, and then stabilizes.

C. Later Epochs (6-9 Epochs)

After the sixth epoch, the training accuracy reaches a plateau, fluctuating slightly around 90-92%. This suggests that the model is converging and further training may not significantly improve performance. The Table III gives the technologies used in our project works.

XI. CONCLUSIONS & FUTURE SCOPES

Here, in the proposed research, we developed a very high efficient and robustness based fruit classification system using deep learning techniques, with an emphasis on real-time performance and scalability. The system leverages the YOLOv3 model for fruit detection and CNN architectures like ResNet50 and VGG16 for classification, achieving high accuracy through the use of a diverse, well-annotated dataset. Integration with the ESP32 module for image capture and AWS SageMaker for cloud- based processing enables seamless edge-cloud collaboration, allowing for real-time detection and classification in various environments. The system's ability to generalize well to different lighting conditions, angles, and backgrounds highlights its versatility for real-world applications in agriculture, retail, and education, the result screen in Fig. 7.

REFERENCES

[1] Dr. Swapnil S. Ninawe, Vatsal Gupta, Somya Sinha, Swayam Gupta, Abhay Gupta, “Fruit Classification using CCN Techniques May 2024” Association of Computer Electrical Electronics and Communication Engineers, ACE2024-621F

[2] Nur-E-Aznin Mimma, Sumon Ahmed, Tahsin Rahman, and Riasat Khan, "Fruits Classification and Detection Application Using Deep Learning," Scientific Programming, 2022.

[3] H. Muresan and M. Oltean, "Fruit Recognition from Images Using Deep Learning," Technical Report, Babes-Bolyai University, Cluj-Napoca, Romania, 2017.

[4] W.C. Seng and S.H. Mirisaee, "A New Method for Fruits Recognition System," Proceedings of the International Conference on Electrical Engineering and Informatics, Selangor, Malaysia, 2009

[5] Kumar, Sarath & Srinivasan, Mohanambigai & Alexander, Dr.S.Albert & Ravi, Samikannu & Narasimha Rao, Dasari & Antony Raj, Raymon. (2021). A Technical Review on Self-Healing Control Strategy for Smart Grid Power Systems. IOP Conference Series: Materials Science and Engineering. 1055. 012153.

[6] N. Lakshmi, G. Pavithra and T.C. Manjunath, “CMOS Implementation of Multipath Fully Differential OTA with Dual Flipped Voltage Follower in 50 nm and 75 nm CMOS Technologies using Cadence Tool,” IEEE International Conference on Distributed Computing and Optimization Techniques (ICDCOT), Bengaluru, India, 2024, pp. 1-8, doi: 10.1109/ICDCOT61034.2024.10515482.

[7] M. Suguna, M. Sangeetha, N. Shanmugapriya, G. Pavithra and V. Anandkumar, “AI Based Ontology-Driven Information Retrieval for Healthcare Information System,” IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1872-1875, doi: 10.1109/IC2PCT60090.2024.10486267

[8] V.K. Suhasini et.al., “Detection of Skin Cancer using Artificial Intelligence & Machine Learning Concepts,” IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Goa, India, 2022, pp. 343-347, doi: 10.1109/ICCCMLA56841.2022.9989146.

[9] G. Pavithra, K. Nikhil, M. Sreeraam, S. Pranjal, Sachin and T.C. Manjunath, “Design of a Smart Moving Road Divider for Effective Control of The Traffic Problems in Densely Populated Traffic Zones,” IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, India, 2021, pp. 1-4, doi: 10.1109/ICMNWC52512.2021.9688392.
[10] T.C. Manjunath, G. Pavithra and B.G. Nagaraj, “Design & simulation of the workspace for a stationary robot system,” IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, 2016, pp. 1-5, doi: 10.1109/R10-HTC.2016.7906828.
[11] J. Rajesh, P. Prasanna, R. Akshith, T.C. Manjunath and G. Pavithra, “Conceptual view of a smart tonopen for biomedical engineering applications,” IEEE International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2017, pp. 636-639, doi: 10.1109/ICCONS.2017.8250540.
[12] G. Pavithra, T.C. Manjunath, D. Lamani and G. Anushree, “Hardware Implementation of Glaucoma using A PIC Micro-Controller – A Novel Concept for a Normal Case of the Eye Disease,” IEEE International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India, 2017, pp. 1104-1109, doi: 10.1109/CTCEEC.2017.845515