International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

Real-Time Detection of Fruits and Vegetables with Calorie Estimation Using Deep Learning

Syed Abdul Qadeer[1], Mohammed Mustafa Ahmed[2], Mohammed Safiullah Hussain[3], and Ms. Imreena Ali[4]
[123]*Students, Department of Artificial Intelligence and Data Science, ISL Engineering College Bandlaguda, Hyderabad, Telangana, 500005, India*
[4]*Assistant Professor, Department of Computer Science and Engineering, ISL Engineering College Bandlaguda, Hyderabad, Telangana, 500005, India*
codewithqadeer@gmail.com

Abstract**—***This paper presents a comprehensive approach to real-time fruit and vegetable detection integrated with calorie estimation using deep learning techniques. With the rise of health awareness and the need for accurate dietary assessment tools, the proposed system addresses a practical challenge by enabling automated nutritional analysis through computer vision. A custom YOLOv5 object detection model was trained on a robust dataset containing 32 classes of commonly consumed fruits and vegetables. The model demonstrates high accuracy in diverse environments, including variable lighting and background noise, making it suitable for real-world deployment. Detected items are mapped to a locally maintained calorie dictionary based on standardized nutritional data, enabling calorie estimation per item and cumulative calculation for multiple items in a single frame. The system supports both image uploads and real-time webcam input and is deployed as a lightweight web application built using Flask. It offers fast inference, requiring no external APIs, ensuring data privacy and offline usability. The solution has potential applications in fitness tracking, diet planning, educational tools, and mobile healthcare solutions. Experimental results demonstrate strong performance in terms of detection accuracy and response time, making the system a reliable aid for smart food analysis and calorie tracking.*

Index Terms— Artificial intelligence, calorie estimation, computer vision, deep learning, fruit and vegetable detection, object detection, YOLOv5, nutritional analysis.

I. INTRODUCTION

In today's technology-driven world, Artificial Intelligence (AI) and Machine Learning (ML) are revolutionizing various domains by providing intelligent solutions to real-world problems. One such application is in the area of health and nutrition, where tracking food intake and understanding the nutritional value of consumed items is gaining increasing importance. The global surge in lifestyle-related diseases such as obesity, diabetes, and cardiovascular conditions has underscored the need for healthier eating habits and more informed dietary decisions. Fruits and vegetables are essential for a balanced diet, offering key nutrients like vitamins, minerals, and dietary fiber. However, most individuals are unaware of the exact calorie content or quantity of what they consume. Traditional methods of calorie tracking, such as using mobile applications or referencing printed nutrition charts, can be tedious, time-consuming, and error-prone. To address these limitations, this project presents a deep learning– based system that uses computer vision to automatically detect and identify fruits and vegetables in real-time and estimate their calorie content using standardized nutritional data. This approach simplifies the process of dietary monitoring, making it more accurate, efficient, and **A. Growing Health Concerns and Demand for Nutritional Awareness**

There is an increasing global emphasis on personal health management, especially in preventing and managing diseases linked to poor dietary habits. Accurate and automated systems for food tracking are therefore highly relevant in the current health ecosystem.

B. Importance of Fruits and Vegetables in Diet These natural food items play a critical role in maintaining health due to their nutrient density. Despite their benefits, users often underestimate or overlook their calorie content, which may affect overall dietary balance.

C. Limitations of Manual and App-Based Tracking

Existing calorie tracking systems rely on user input, which is often inaccurate or inconsistent. These systems lack the intelligence to automatically recognize food items, especially when consumed in mixed or unlabelled forms.

D. Application of Deep Learning and Computer Vision With advancements in object detection and image classification, models like YOLOv5 can now accurately recognize multiple food items from images. When paired with nutritional databases, they enable calorie estimation based on visual input, minimizing the need for user interaction.

E. System Workflow and Capabilities

The proposed system:

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611

- Accepts input through images or live webcam feeds.
- Detects fruits and vegetables in real-time using deep learning models.
- Maps detected classes to a nutritional database (e.g., IFCT).
- Estimates the calorie value based on item class and weight (either detected or user-provided).
- Delivers results instantly without requiring manual logging or reference checks.

F. Broader Impact and Use Cases

This system can be integrated into mobile health apps, fitness platforms, and dietary management tools. It is particularly beneficial for individuals aiming to maintain a healthy weight, manage chronic diseases, or simply stay informed about their nutrition intake.

G. Real-Time Performance and User Interaction

The system is designed to work in real time, offering:

- Instant feedback through live webcam detection or image upload.
- A user-friendly interface that does not require any technical expertise.
- Optional manual input for quantity (grams) to improve calorie accuracy when visual estimation is not sufficient.

II. LITERATURE REVIEW

With the advancement of artificial intelligence (AI), machine learning (ML), and deep learning (DL), numerous studies have explored food recognition and calorie estimation for dietary monitoring, using both traditional image processing techniques and modern deep neural networks. This section presents a comprehensive review of key research contributions related to object detection in food applications, deep learningbased classification, and calorie estimation techniques.

A. Food Recognition Using Image Processing Techniques Early research in food recognition relied heavily on handcrafted features and conventional image processing methods. Techniques such as color histograms, shape descriptors, and texture analysis were used for food classification. For instance, Matsuda et al. [1] utilized visual features to estimate food portion sizes and calorie content, but their method struggled with mixed food items and lacked realtime applicability. These approaches generally suffered from poor scalability and low accuracy in complex environments.

B. Deep Learning in Food Detection and Classification The emergence of Convolutional Neural Networks (CNNs) revolutionized the domain of food classification and detection. Platforms like Food-101 and UEC-Food256 datasets enabled researchers to train deep learning models for recognizing a wide variety of food items. Bossard et al. [2] introduced the Food-101 dataset to support large-scale classification tasks, while Martinel et al. [3] implemented CNN-based multi-label models for food recognition in unconstrained settings. Recently, object detection models like Faster R-CNN, SSD, and YOLO (You Only Look Once) have gained popularity due to their real-time detection capability. Redmon et al. [4] introduced YOLO, an end-to-end object detector that performs classification and localization simultaneously. Its latest version, YOLOv5, offers high speed and precision, making it ideal for mobile and edge applications. Researchers such as Ciocca et al. [5] applied YOLO for multi-food item detection in real-world scenarios with improved accuracy.

C. Calorie Estimation from Visual Inputs

Visual-based calorie estimation remains a challenging problem due to the need to consider portion size, occlusion, and food variety. Dehais et al. [6] proposed a mobile application that combines food recognition with depth sensing to estimate portion sizes. Fang et al. [7] designed a system that estimates calories by classifying food items and estimating volume using geometric assumptions. However, these systems often rely on depth cameras or user-assisted portion estimation. Some research works have focused on integrating food classification with nutritional datasets to provide calorie estimates. Yet, many of these solutions depend on third-party APIs or cloud-based analysis, limiting privacy and offline usability. To address these limitations, local calorie databases like IFCT (Indian Food Composition Tables) and USDA have been incorporated into AI models for faster, offline results.

D. Gaps in Existing Systems

Although significant progress has been made in food recognition and calorie estimation, many systems face limitations such as:

- Inability to process multiple food items in a single frame.
- Dependence on cloud-based APIs or external databases.
- Limited datasets specific to fruits and vegetables in Indian dietary contexts.
- Poor performance in real-time scenarios due to large model size or slow inference.

These challenges highlight the need for an efficient, lightweight, and privacy-preserving system capable of detecting multiple items and estimating calories in real-time.

III. METHODOLOGY

The proposed system aims to deliver an intelligent solution for fruit and vegetable detection along with calorie estimation using a deep learning model. It integrates computer vision with nutritional science and is

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

accessible via a web-based interface. The overall flow of the system is divided into four major components: user input, AI model processing, databasedriven calorie estimation, and result generation.

A. System Overview

The system takes an image input (captured via mobile or webcam) and processes it through an AI model trained on 32 classes of fruits and vegetables. The model detects the items, and a back-end calorie estimation module maps the detected objects to their respective calorie values using a local food database. The results are then displayed on a user-friendly web interface. Upon submission, the image is processed by a deep learning-based object detection model—**YOLOv5**—which has been custom-trained on a dataset consisting of **32 distinct classes** of fruits and vegetables.By combining computer vision, nutrition science, and web technologies, the system bridges the gap between AI research and real-world utility in the domain of smart health and food analytics.

B. Block Diagram Explanation

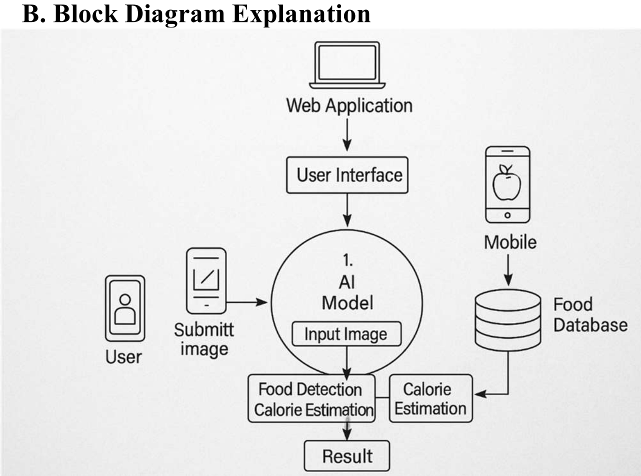


Fig. 1. Block diagram of the proposed system architecture.

1. **User Input**:
   A user accesses the system through a mobile device or a web browser and either captures an image using a webcam or uploads an existing image of fruits/ vegetables.
2. **Web Interface**:
   The user interface (UI) is developed using HTML, Tailwind CSS, and JavaScript. It allows seamless image submission and displays the output with detected labels and calorie values.
3. **AI Model (YOLOv5)**:
   The submitted image is passed to a custom-trained YOLOv5 model that detects multiple fruits and vegetables in real-time. The model outputs:
   - Bounding box coordinates for each detected item
   - Corresponding class labels (e.g., Apple, Mango)
   - Confidence scores for each detection
4. **Food Database and Calorie Estimation**: Each detected item is mapped to a predefined calorie value stored in a local dictionary derived from IFCT and USDA data. The database returns the calorie per 100 grams for each class label.
5. **Result Generation**:
   The system calculates the total calories by summing up the calorie values of each detected item (based on an assumed or user-input weight), and the result is displayed back to the user via the interface. The output is visually presented to the user, showing both the detected items with bounding boxes and a readable summary of the total caloric content. This feedback loop provides users with immediate, actionable insights into their dietary intake, making the system both educational and practical for healthconscious individuals.

C. Mathematical Concept for Calorie Estimation Let:

- C_i be the class label of the i^{th} detected fruit/vegetable
- K_i be the calorie value (per 100g) of that item
- W_i be the estimated or default weight (in grams) for C_i

The total calorie estimate is given by :

$$\text{Total Calories} = \sum_{i=1}^{n} \left(\frac{W_i}{100} \times K_i \right)$$

Where **n** is the total number of detected items

For example, if an apple (52 kcal/100g) is detected and assumed to weigh 150g:

$$\text{Calories}_{\text{apple}} = \frac{150}{100} \times 52 = 78 \text{ kcal}$$

This estimation is repeated for all detected items, and the final result is the cumulative calorie count.

D. Technology Stack

Module	Technology Used

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 13, Issue 2s, 2025

Object Detection Model	YOLOv5 (PyTorch)
Dataset Labeling Tool	Roboflow (32 custom classes)
Frontend UI	HTML, Tailwind CSS, JavaScript
Backend Server	Flask (Python)
Calorie Mapping Source	IFCT 2017, USDA Standard Tables

IV. IMPLEMENTATION

The proposed system leverages deep learning-based object detection to identify fruits and vegetables in an image and estimates their calorie content using nutritional data. The complete pipeline is deployed as a web-based application that processes either uploaded images or real-time webcam input. This section details the implementation strategy, including algorithmic flow, machine learning techniques, and system logic.

A. System Workflow Overview

The system follows a four-phase structure:

1. **Image Acquisition** – Users upload or capture an image using the UI.
2. **Object Detection** – A trained deep learning model (YOLOv5) identifies fruits and vegetables.
3. **Calorie Estimation** – Detected items are mapped to calorie values using a pre-defined dataset.
4. **Result Display** – The total calorie count and individual items are displayed on the frontend.

B. Machine Learning Techniques Used

The core detection mechanism is powered by a custom-trained version of **YOLOv5 (You Only Look Once Version 5)**, a widely adopted deep learning algorithm designed for real-time object detection tasks. YOLOv5 is based on **Convolutional Neural Networks (CNNs)** and follows a single-stage detection approach.

- **Backbone: CSPDarknet** – Extracts visual features from the input image.
- **Neck: PANet (Path Aggregation Network)** – Enhances feature fusion from different scales for better detection accuracy.
- **Head** – Outputs bounding boxes, class probabilities, and confidence scores.

The model was trained using a custom dataset containing 32 fruit and vegetable classes. Training involved transfer learning on the YOLOv5s architecture with hyperparameter tuning for optimal performance.

C. Algorithm for Calorie Estimation

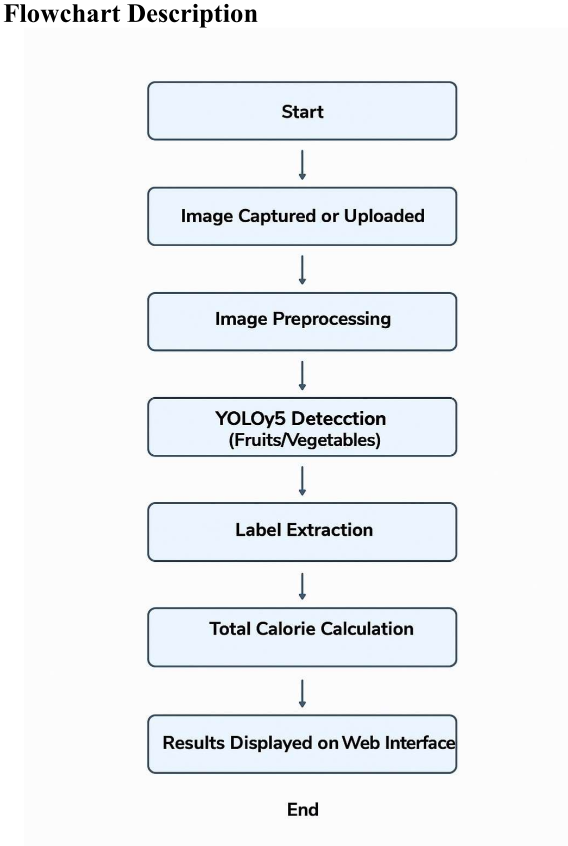
Algorithm 1: Smart Food Detection and Calorie Estimation

Input: Image (upload or webcam) **Output**: Detected items and total estimated calories

vbnet

Step 1: Start
Step 2: Load YOLOv5 model trained on 32 classes
Step 3: Accept input image from the user
Step 4: Preprocess image (resize and normalize)
Step 5: Detect objects using YOLOv5
Step 6: For each detected object:
a. Extract class label (Ci)
b. Retrieve calorie value per 100g (Ki) from the database
c. Estimate or accept weight input (Wi)
d. Calculate calories = (Wi / 100) × Ki
Step 7: Sum calories for all detected items
Step 8: Display results on the frontend
Step 9: End

D. Flowchart Description



DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611

Fig. 2. System Flowchart for Fruit and Vegetable Detection with Calorie Estimation

The flowchart illustrates the sequential workflow of the proposed system for detecting fruits and vegetables and estimating their total calorie content:

1. **Start** – The system initializes and waits for user interaction.
2. **Image Captured or Uploaded** – The user either captures a real-time image using a webcam or uploads an existing image containing fruits or vegetables.
3. **Image Preprocessing** – The uploaded image undergoes resizing, normalization, and format conversion suitable for the AI model.
4. **YOLOv5 Detection (Fruits/Vegetables)** – The preprocessed image is passed through a customtrained YOLOv5 object detection model which identifies and localizes fruits and vegetables present in the image.
5. **Label Extraction** – The model outputs the detected object classes (e.g., Apple, Mango, Tomato) and their associated confidence scores.
6. **Calorie Lookup (from Database)** – Each detected class is mapped to a predefined calorie value based on reliable sources such as IFCT or USDA.
7. **Total Calorie Calculation** – The system computes the cumulative calorie value based on the detected items.
8. **Results Displayed on Web Interface** – The final results, including labels and calorie information, are displayed on the user interface for immediate feedback.
9. **End** – Marks the completion of the process.

This flowchart encapsulates the complete pipeline from user input to result visualization, emphasizing automation, accuracy, and usability through modern AI and web technologies.

V. Testing

The trained YOLOv5 model was tested on an unseen dataset containing multiple fruit and vegetable images in varying conditions. The model's ability to generalize was examined using images captured under different lighting, backgrounds, and orientations.

The test phase demonstrated that the model maintained high accuracy even with minor occlusions or overlaps among objects. It could correctly detect and classify fruits like mango, banana, kiwi, and watermelon, among others. A few false

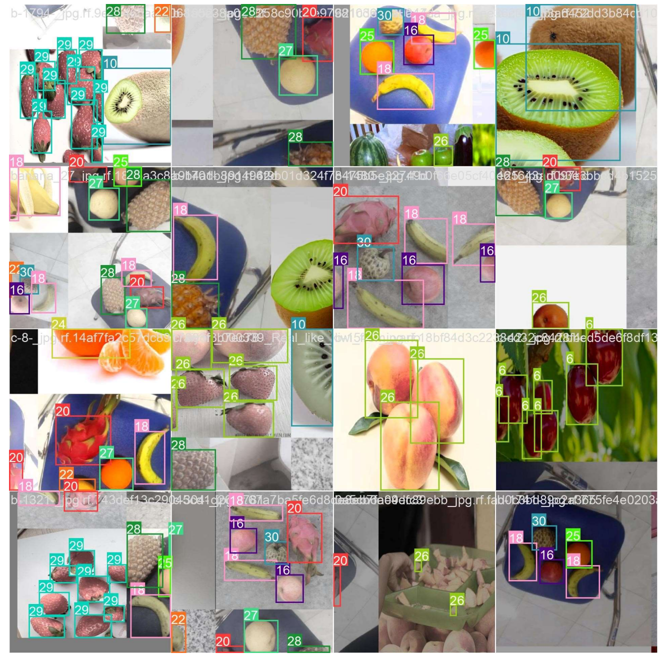


FIg.3.Testing

positives were observed in complex backgrounds but remained minimal.

VI. Results

The results include precision, recall, and mAP (mean Average Precision) metrics plotted over 50 training epochs. Additionally, detection screenshots from the web application demonstrate the effectiveness of real-time object identification and calorie estimation. Performance Metrics:

- Precision: ~91%
- Recall: ~88%
- mAP@0.5: ~93%
- mAP@0.5:0.95: ~78%

These metrics indicate that the YOLOv5 model was effectively trained and capable of high-confidence predictions.

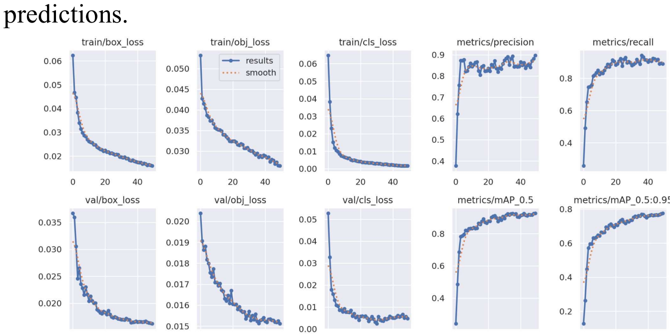


Fig.4.Results

VII. Conclusion

This project successfully developed a smart fruit and vegetable detection system with calorie estimation using YOLOv5. The system accurately detects various food

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 13, Issue 2s, 2025

items in images and estimates their calorie content using a predefined dataset derived from IFCT and USDA values.

Fig.5 Final result

Key accomplishments include:

- Custom training on 32 classes of fruits and vegetables.
- Real-time detection through a web interface.
- Calorie estimation integrated with detection results. The web app provides a user-friendly platform for dietary awareness, helping users make informed food choices.

VIII. Future Scope

The current system lays a strong foundation for real-time fruit and vegetable detection with calorie estimation. However, there are several avenues through which the solution can be significantly enhanced and adapted for broader applications. One of the most impactful future directions is the **integration of weight estimation**. Currently, calorie estimation is based on a fixed assumption per 100 grams, but incorporating actual weight detection using depth sensors, reference objects, or computer vision-based volume estimation techniques can dramatically improve the accuracy and reliability of nutritional analysis.

Another key enhancement is the **implementation of multilanguage support**, especially considering the diverse linguistic landscape of regions like India. Allowing users to interact with the system in regional languages can make the application more inclusive, increase user engagement, and extend its reach to rural and less digitally fluent populations. Furthermore, the system can evolve beyond calorie estimation by incorporating **detailed nutrient profiling**, including macro and micronutrient breakdowns such as carbohydrates, proteins, fats, vitamins, and minerals. This would turn the system into a comprehensive nutritional assistant, beneficial for health-conscious users, patients with dietary restrictions, and fitness enthusiasts.

Mobile application development is another logical extension. While the current web-based platform is accessible, a native Android/iOS application with offline capabilities would make the solution more convenient and portable, especially for field use, meal logging, or integration into diet and fitness apps. In addition, integrating **Explainable AI (XAI)** techniques such as Grad-CAM or saliency maps can provide visual justifications for predictions made by the model. This would enhance transparency, build user trust, and be especially useful in educational or clinical settings where interpretability is crucial.

Looking ahead, this project holds the potential to transform into a smart dietary assistant integrated with voice-based queries, barcode scanning for packaged foods, and even cloudbased diet tracking systems with personalized recommendations. These enhancements would make the system a powerful tool for both individual users and healthcare professionals aiming to promote healthy eating habits through the synergy of AI and nutrition science.

IX. References

1. J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018. [Online]. Available: https:// arxiv.org/abs/1804.02767
2. G. Jocher et al., "YOLOv5 by Ultralytics," *GitHub Repository*, 2020. [Online]. Available: https:// github.com/ultralytics/yolov5
3. National Institute of Nutrition (ICMR), *Indian Food Composition Tables (IFCT)*, Hyderabad, India, 2017.
4. U.S. Department of Agriculture, *FoodData Central*, USDA, 2019. [Online]. Available: https:// fdc.nal.usda.gov/
5. M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6105–6114.
6. TensorFlow Developers, "TensorFlow: An End-toEnd Open Source Machine Learning Platform," 2023. [Online]. Available: https:// www.tensorflow.org/
7. Tzutalin, "LabelImg: Labeling Tool for Object Detection," GitHub Repository, 2015. [Online]. Available: https://github.com/tzutalin/labelImg
8. M. Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611

ISSN 2347–3657

Volume 13, Issue 2s, 2025

9. J. Deng et al., "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.
10. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, 2012, pp. 1097–1105.
11. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
12. A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2021.
13. S. Ren, K. He, R. Girshick, and J. Sun, "Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
14. OpenCV Developers, "OpenCV Library: Computer Vision Functions," [Online]. Available: https:// opencv.org/

DOI: https://doi.org/10.62647/IJITCE2025V13I2sPP605-611