A REVIEW PAPER ON Smart Kitchen Assistant for Ingredient Detection and Recipe Guidance Using Edge Artificial Intelligence

Vijay Chakole[1] ,Ayush Sarode[2],Raju Malekar[3],Muskan Kulkarni [4],Ravisha Kolhe[5] , Khushi Chakole[6]

[1]Head f Dept , Dept of Electronics and Telecommunication, KDK College of Engineering, Maharashtra, India
[2345]UG student, Dept of Electronics and Telecommunication, KDK College of Engineering, Maharashtra, India

---***---

Abstract - *The rapid advancement of embedded computing and artificial intelligence has significantly influenced the automation of domestic environments. Among various home activities, cooking remains a routine yet decision-intensive task that often requires users to determine suitable recipes based on available ingredients. This research presents the design and implementation of a Smart Kitchen Assistant capable of detecting food ingredients using image processing techniques and providing appropriate recipe guidance through an interactive display interface. The proposed system operates entirely on edge hardware using a Raspberry Pi, a camera module, and locally stored data resources, thereby eliminating dependence on internet connectivity. A lightweight convolutional neural network model is deployed using an optimized inference engine to enable real-time ingredient classification on resource-constrained hardware. Recipes are retrieved from an offline structured database and presented via a user-friendly touch interface, while a secondary display provides system feedback. The study discusses architectural design, processing workflow, hardware-software integration, algorithm formulation, and experimental observations. Results indicate that edge-based processing achieves acceptable accuracy and response time for domestic applications while ensuring privacy and reliability. The proposed system demonstrates the feasibility of developing an affordable intelligent cooking assistant suitable for smart home environments.*

***Key Words*: Smart kitchen, Ingredient detection, Edge computing, Embedded vision, Raspberry Pi, Recipe recommendation, Artificial intelligence.**

1. INTRODUCTION

The integration of intelligent systems into everyday domestic appliances has transformed the concept of smart homes from theoretical models to practical implementations. Cooking, being an essential daily activity, presents significant opportunities for automation and intelligent assistance. Many individuals experience uncertainty when selecting meals based on limited available ingredients, leading either to repetitive cooking patterns or increased food wastage. Conventional solutions such as recipe books or mobile applications require manual input and active searching, which may not provide immediate or intuitive assistance.

Recent advancements in compact computing platforms and machine learning algorithms have enabled the development of systems capable of performing complex image recognition tasks directly on embedded hardware. Edge computing, which refers to processing data locally rather than transmitting it to remote servers, offers significant advantages including low latency, enhanced privacy, and reduced dependency on network infrastructure. These characteristics make it particularly suitable for domestic kitchen environments where reliability and responsiveness are critical.

The Smart Kitchen Assistant proposed in this research is designed to bridge the gap between ingredient availability and recipe decision-making. By combining computer vision with embedded artificial intelligence, the system automatically detects ingredients placed before a camera and retrieves suitable recipes stored in a local database. The objective of this work is to design a compact, affordable, and efficient prototype capable of operating independently without cloud support, thereby demonstrating the practicality of intelligent edge-based cooking assistance.

2. SYSTEM DESIGN AND ARCHITECTURE

The overall architecture of the proposed system centers on a Raspberry Pi single-board computer functioning as the primary processing unit. The Raspberry Pi is selected due to its balance between computational capability, cost-effectiveness, and compatibility with peripheral devices. A web camera is connected to capture images of ingredients placed within a predefined detection region. The captured data is processed locally using an optimized inference model.

The system incorporates a touch-enabled display that serves as the primary user interface. This interface allows users to initiate scanning, view detection results, and read detailed recipe instructions. In addition to the main display, an auxiliary Inter-Integrated Circuit (I2C) Liquid Crystal Display provides concise system status updates such as readiness, processing state, or detection confirmation. All operational software components, including the operating system, trained neural network model, and recipe database, are stored on a microSD card to ensure completely offline functionality.

The architecture is modular in nature. The image acquisition module captures frames at the user's request. The preprocessing module prepares the captured image by resizing, normalization, and conversion into a format compatible with the neural network model. The classification module performs inference and outputs the most probable ingredient label. Finally, the recipe retrieval module searches a structured database and displays the corresponding cooking procedure.

3.METHODOLOGYANDALGORITHMIC FRAMEWORK

The operational workflow of the Smart Kitchen Assistant follows a systematic sequence beginning with user interaction and concluding with recipe display. When the user places an ingredient under the camera and initiates the detection process, the camera captures a high-resolution image. The captured frame undergoes preprocessing to remove unnecessary background variations and standardize input dimensions for neural network inference.

A lightweight convolutional neural network model is employed to perform ingredient classification. The choice of a compact model is essential to ensure real-time performance on resource-limited hardware. The inference process calculates probability scores for predefined ingredient categories, and the label with the highest confidence is selected as the output. Model optimization techniques such as quantization are applied to reduce computational load and memory usage while maintaining acceptable accuracy.

Following classification, the detected ingredient label is used as a key to query the local recipe database. The database is structured in JSON format to allow efficient search and retrieval operations. If a matching recipe is found, the system displays preparation steps, required quantities, and estimated cooking time on the touch interface. In cases where no match exists, the system prompts the user accordingly. The entire process is designed to complete within a few seconds to ensure seamless user experience.

4.IMPLEMENTATIONANDEXPERIMENTAL OBSERVATIONS

The hardware components are assembled within a compact enclosure designed to withstand typical kitchen conditions. Stable power regulation is ensured through appropriate voltage management modules to provide uninterrupted operation. The camera positioning is optimized to minimize shadow effects and improve detection reliability under indoor lighting conditions.

The software environment is developed using Python programming language due to its extensive support for image processing and machine learning libraries. OpenCV is used for image handling and preprocessing tasks, while TensorFlow Lite is employed to execute the optimized neural network model. The graphical user interface is developed with emphasis on clarity, readability, and ease of navigation.

Experimental evaluation focuses on response time, classification accuracy, and robustness under varying lighting conditions. Initial testing with a limited dataset of common vegetables and fruits indicates that the system achieves satisfactory recognition accuracy for clearly visible single ingredients. The average detection time, including preprocessing and inference, remains within two seconds. Although performance decreases under poor lighting or when ingredients are partially occluded, the system demonstrates consistent functionality under normal indoor conditions.

5. DISCUSSION

The implementation of ingredient detection using edge computing offers several advantages over traditional cloud-based solutions. By eliminating the need for external server communication, the system reduces latency and enhances user privacy. Moreover, offline functionality ensures reliability even in environments with unstable internet connectivity. The affordability of the hardware components further supports the feasibility of large-scale adoption in domestic settings.

However, certain limitations remain. The current model focuses primarily on single-object classification and does not fully address multi-ingredient or mixed-item scenarios. Dataset diversity significantly influences recognition accuracy, and expanding the training dataset would improve robustness. Future enhancements may incorporate object detection algorithms capable of identifying multiple ingredients simultaneously, as well as adaptive learning mechanisms that refine predictions based on user feedback.

6. CONCLUSION

This research presents the design and development of a Smart Kitchen Assistant that integrates embedded hardware with lightweight artificial intelligence for automated ingredient detection and recipe guidance. The proposed edge-based architecture successfully demonstrates that real-time image classification and database retrieval can be achieved using affordable single-board computing platforms. Experimental observations confirm the practicality of the approach for domestic applications while highlighting areas for future improvement. With further dataset expansion, optimization, and integration of multi-object detection techniques, the system can evolve into a comprehensive intelligent cooking assistant for smart home environments.

REFERENCES

[1]A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.

[2] M. Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[3] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proceedings of the IEEE CVPR, 2017, pp. 1251–1258.

[4]K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," International Conference on Learning Representations (ICLR), 2015.

[5] C. Szegedy et al., "Going Deeper with Convolutions," Proceedings of the IEEE CVPR, 2015, pp. 1–9.

[6] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Google Research, 2016.

[7] B. Zoph et al., "Learning Transferable Architectures for Scalable Image Recognition," Proceedings of the IEEE CVPR, 2018.

[8] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE CVPR, 2016, pp. 779–788.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, 2012.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[11]W. Shi et al., "Edge Computing: Vision and Challenges," IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.

[12]M. Satyanarayanan, "The Emergence of Edge Computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[13] H. Chen et al., "Deep Learning on Embedded Systems: A Survey," IEEE Transactions on Emerging Topics in Computing, 2020.

[14] S. Han et al., "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," International Conference on Learning Representations, 2016.

[15]J. Dean et al., "Large Scale Distributed Deep Networks," Advances in Neural Information Processing Systems, 2012.

[16 R. Want, "An Introduction to RFID Technology," IEEE Pervasive Computing, vol. 5, no. 1, pp. 25–33, 2006.

[17 S. K. Datta et al., "Smart Home Automation Using IoT," IEEE Consumer Electronics Magazine, vol. 5, no. 1, pp. 48–55, 2016.

[18] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," Proceedings of the IEEE CVPR, 2001.

[19] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[20] N. Lane et al., "Deep Learning for Mobile and Embedded Devices: A Survey," IEEE Pervasive Computing, vol. 17, no. 3, pp. 35–44, 2018.