Design and Development of a Real-Time Camera-based Smart Cooking Assistant

Hammad Sheikh*, Kiran George†, Tabashir Nobari‡ and Anand Panangadan§
California State University, Fullerton
Fullerton, California 92831, USA

Abstract—**Personalized cooking recipe recommendation systems offer the potential to improve dietary choices for unhoused individuals and those transitioning out of homelessness. However, existing systems often neglect the needs of users with minimal cooking experience, providing little guidance during meal preparation. This study proposes the development of an intelligent cooking assistant system designed to offer real-time, step-by-step support throughout the cooking process. The system integrates a Raspberry Pi 5 mini-computer with a Raspberry Pi AI HAT+ (AI HAT+) and Raspberry Pi AI Camera (AI Camera), strategically mounted above the cooking area to continuously monitor culinary activity. At its core, the assistant utilizes a deep learning image classification model built on Ultralytics' You Only Look Once version 11 (YOLO11) framework, trained on a curated dataset of 1,339 images collected during the preparation of chicken teriyaki and pasta dishes. The model achieved 100% precision and 99% recall of identifying all six cooking states utilized in this work, resulting in an average confidence accuracy of 91% during real-time tests. The system is intended to enable greater culinary independence among individuals with little cooking experience, such as those affected by long-term homelessness.**

Index Terms—**Supportive Housing, Machine Learning, Smart Home, Raspberry Pi, YOLO, Recipe Assistant**

I. Introduction

Cooking is a skill that must be learned. Although most adults are able to learn basic cooking skills and prepare meals following a recipe, there exist subpopulations for whom cooking is a challenge. One such group of people are those transitioning out of homelessness. Homelessness is a major concern in the United States and other parts of the world. Los Angeles County alone was reported to have 75,312 unhoused persons in 2024 [1]. Providing supportive housing, which is subsidized housing provided along with personalized wraparound supportive services such as mental health and substance abuse treatment, is the preferred way to address the crisis [2]. However, individuals having experiences of homelessness often face barriers that make them less likely to seek assistance in developing cooking skills, leading to limited improvements in preparing nutritious meals even when a kitchen becomes available. The unhoused individuals who have recently transitioned out of homelessness and are residing in supportive housing can significantly benefit from personalized guidance in nutrition and cooking. Such support is especially critical given common challenges, including limited income, insufficient cooking skills and knowledge, and restricted access to affordable, nutritious food [3], [4].

Recently, personalized cooking recipe recommendation systems have been developed to offer guidance on nutrition [5], [6]. Such systems can also be based on emerging Artificial Intelligence (AI) technologies [7]. However, a significant limitation of existing recipe recommendation systems is their inability to deliver real-time feedback during the cooking process.

This work develops and validates the hypothesis that a smart "cooking assistant" can be engineered to deliver real-time feedback and support throughout the cooking process. This system is therefore intended to be used *after* a recipe is recommended by personalized recommendation system. The intended outcome is to provide a personalized, interactive, and educational experience that enhances nutritional well-being among individuals transitioning out of homelessness, while simultaneously fostering essential culinary skills and self-sufficiency.

The system has hardware and software components that together perform as an edge device. The hardware consists of a camera mounted below the cooking range hood, looking down on the stove top. The camera points straight down to ensure the user privacy. The camera is connected to an embedded computer (Raspberry Pi 5) which processes the images with a specially trained object recognition and detection neural network model. The labels of the model are mapped to the main steps of a particular recipe. (In this work, we evaluate the system on a *Chicken Teriyaki* recipe.) Thus, the system is able to monitor the current step being performed on the stovetop and prompt the cook to move to the next step after an appropriate amount of time has passed, guiding the cook to follow all the steps of the recipe.

The contributions of this work include (1) the design of an edge device that functions as a smart cooking assistant, (2) development of a Machine Learning model that is trained on a custom dataset to monitor in real-time the cooking process and identify the cooking step of a recipe, and (3) evaluation of the real-world performance of the system.

II. Related Work

Artificial Intelligence (AI) techniques, particularly machine vision and image processing, have been widely applied across various aspects of food processing. These methods

*Dept. of Computer Science, hshammads@csu.fullerton.edu
†Dept. of Electrical & Computer Engineering, kgeorge@fullerton.edu
‡Dept. of Public Health, tnobari@fullerton.edu
§Dept. of Computer Science, apanangadan@fullerton.edu

are primarily used for tasks such as identifying food types and quality, grading food products, and detecting defects or foreign objects [8]. A notable contribution in this area is the development of a dataset of Chinese recipes, which includes multiple images representing different stages of cooking. Distinct models were trained independently for specific categories, including the initial, intermediate, and advanced stages of food preparation [9].

However, these approaches do not directly correspond to the core steps involved in home cooking for personal use. Research specifically focused on assistive cooking systems remains limited. One notable project, the Cognitive Orthosis for coOKing (COOK), is a smart tablet application connected to a stove, designed to assist individuals with cognitive impairments during meal preparation [10], [11]. Monitoring and tracking objects during cooking within COOK utilize real-time detection and tracking techniques such as You Only Look Once (YOLO) and the Kernelized Correlation Filter (KCF). The system addresses several challenges, including object disappearance and reappearance, occlusion, and motion blur. Evaluations have demonstrated that combining object detection and tracking data significantly improves the system's ability to trace and identify kitchen utensils [12].

Similarly, the study by Jelodar et al. [13] created a dataset of cooking-related images representing 11 common object states, using a deep learning model based on ResNet for object identification. These systems often require retraining their object detection models to accurately recognize cooking-specific items.

In previous work [14], we had developed a version of the smart "cooking assistant" that incorporated a Raspberry Pi Camera Module 2, a thermal camera, an infrared temperature sensor, and ambient temperature and humidity sensors on a Raspberry Pi 4 with a Coral USB accelerator. The system was also trained to identify the stages of cooking only a simple dish - boiling pasta. These were: (1) Empty Burner, (2) Empty Pot, (3) Pot with Water, (4) Pot with Boiling Water, (5) Pot with Pasta, and (6) Pot with Cooked Pasta. The full dataset comprised 330 images. That system used a Machine Learning model that is deployed remotely (Vertex AI) and was thus not a true edge device. An Edge AI version of the trained model was developed using Vertex AI but had high deployment costs. In this work, we describe a fully local solution capable of supporting model training and deployment on an edge device, thereby removing the dependency on continuous cloud connectivity and significantly reducing operational cost. The current system is also trained to assist in the cooking of more complex dishes and is correspondingly trained on a much larger dataset of images.

III. System Design

A. Hardware Design

This work focuses on developing a fully local solution capable of supporting model training and deployment directly on an edge device. As an edge system, it must be both cost-effective and accurate to reliably monitor the cooking process and provide real-time, step-by-step guidance to the user. Furthermore, the design must be unobtrusive and uphold user privacy.

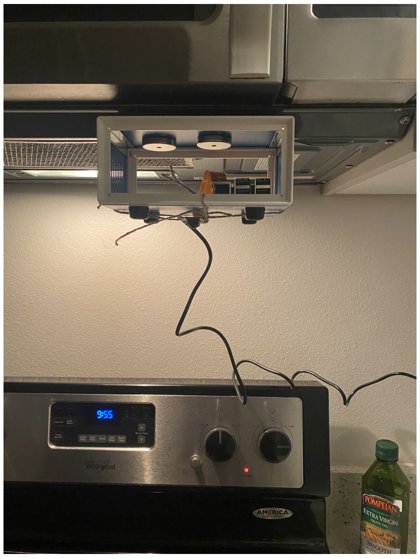


Fig. 1: Hardware setup enclosed in a metal box (with exposed sides), mounted below a range hood using magnets.

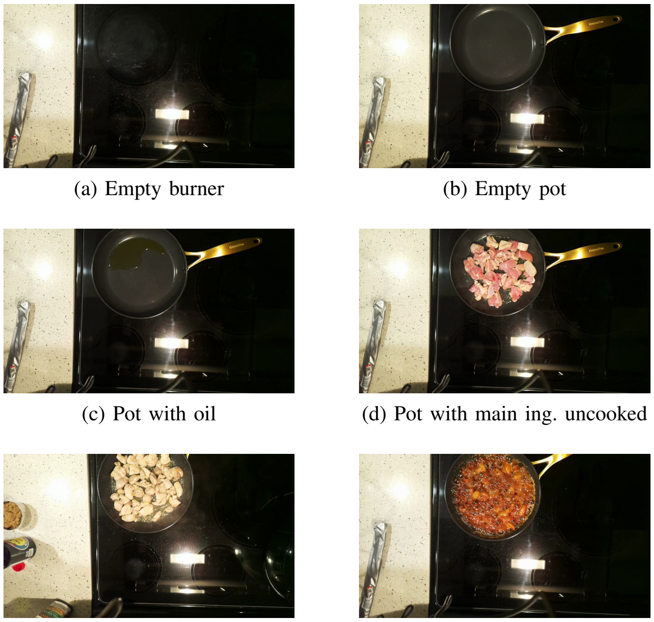
1) Selected Hardware

1) **Raspberry Pi 5:** An affordable and versatile mini-computer capable of running custom operating systems with multiple camera and display options. It can operate headless or with a custom OS, and is well-suited for AI and machine learning at the edge. We utilized the default Raspberry Pi OS.
2) **Raspberry Pi Active Cooler:** An official cooling accessory designed for the the Raspberry Pi 5, featuring a heatsink and a fan that clips onto the board to actively dissipate heat and maintain optimal performance.
3) **Raspberry Pi AI HAT+ (AI HAT+):** An add-on board for the Raspberry Pi 5 that provides enhanced AI processing capabilities using an integrated Hailo-8 AI accelerator, with 26 Tera Operations Per Second (TOPS).
4) **Raspberry Pi AI Camera (AI Camera):** A high-resolution camera module designed for AI applications, featuring 12 MP Sony IMX500 Intelligent Vision Sensor and integrated low-power inference engine for real-time image recognition and computer vision tasks.

Fig. 1 illustrates the complete hardware setup, enclosed in a metal box and mounted below a range hood using magnets.

B. Data Acquisition

With no existing dataset aligning with our unique use case, a smart cooking assistant, and selected recipe, chicken teriyaki, we collected the data ourselves. The chicken



(a) Empty burner

(b) Empty pot

(c) Pot with oil

(d) Pot with main ing. uncooked

(e) Pot with main ing. cooked

(f) Pot with all ingredients

Fig. 2: Samples of representative images of the six classes of the stovetop environment

teriyaki recipe includes these ingredients; oil, chicken, brown sugar, soy sauce, and optional sesame seeds.

Using the default Python library for the AI Camera, we configured the system to simultaneously record video (in one-minute segments) and capture still images (at one-second intervals). Data collection was conducted over the course of a month at various times of day, resulting in approximately 1,300 images along with corresponding video footage. Sample images are shown in Fig. 2.

C. Data Pre-Processing

With the collected data, to closely simulate a real-world deployment scenario, we deliberately chose not to apply any data transformations beyond annotation. Our objective was to evaluate the system's performance - and that of the trained model - using raw, unprocessed input. This approach minimizes computational overhead on the edge device during runtime, as it eliminates the need for real-time data transformation. Instead, the device directly feeds raw camera frames into the model, which is solely responsible for interpreting the cooking state. It is important to note that the current work and model development are based solely on the collected images. Although videos were recorded during data acquisition, they were not utilized in this phase and have been preserved for future enhancements and iterations of the system.

1) Data Annotation

Since no pre-processing is applied to the data, annotation becomes a critical component of the workflow. Using the collected dataset, we segmented the cooking process into six distinct stages (classes), each representing a clearly observable state of the stovetop environment:

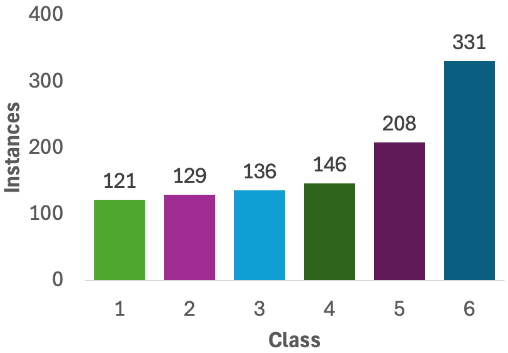


Fig. 3: Distribution of the training dataset for the six classes of the stovetop environment

1) Empty Burner
2) Empty Pot
3) Pot with Oil
4) Pot with Main Ingredient Uncooked
5) Pot with Main Ingredient Cooked
6) Pot with All Ingredients

We annotated the dataset using the labelImg tool, assigning each image to one of the defined classes. This process generated corresponding .txt label files, containing both the class identifier and bounding box coordinates.

Following annotation, the dataset was prepared for model training by partitioning it into training and validation subsets. In accordance with standard machine learning practices, the dataset was split using an 80:20 ratio. This resulted in 1,071 images allocated to the training set. The distribution of training data across the six annotated classes is illustrated in Fig. 3.

D. Model Development

Among popular object detection models, YOLO has gained particular prominence. Unlike traditional object detection methods, YOLO uses a single unified network to simultaneously detect and classify objects, greatly simplifying the detection pipeline compared to earlier approaches [15]. Our smart "cooking assistant" system builds upon YOLO11S, demonstrating promising results, as discussed below.

YOLO version 11 was selected as it is specifically optimized for deployment on resource-constrained devices such as the Raspberry Pi. The model was trained for 80 epochs on a locally available PC equipped with an internal GPU. Due to hardware constraints, the number of training epochs was capped at 80.

The YOLO11S model comprises 181 layers, with approximately 9.4 million parameters and an equivalent number of trainable gradients. With training, 493 out of 499 components from the pretrained weights were successfully

Open Source, https://github.com/raspberrypi/picamera2/tree/main

Open Source, https://github.com/HumanSignal/labelImg

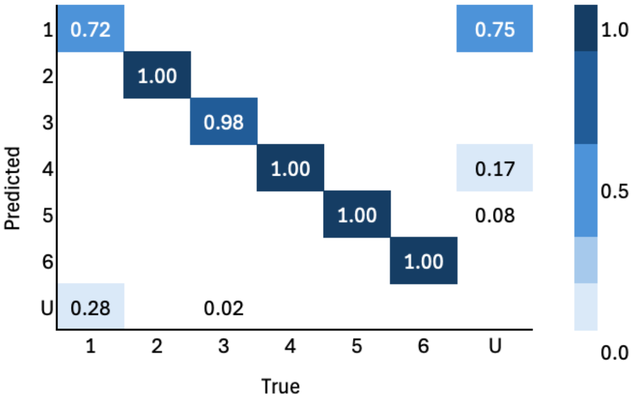


Fig. 4: Confusion Matrix (Normalized) for the six classes of the stovetop environment

transferred. This effective weight transfer enables the model to leverage prior knowledge, thereby improving fine-tuning efficiency and reducing overall training time. The fine-tuning process, over 80 epochs, required approximately 3.4 hours and resulted in the generation of two sets of model weights, best and last, as outlined previously. Notably, the best set of weights was utilized.

IV. Results and Discussion

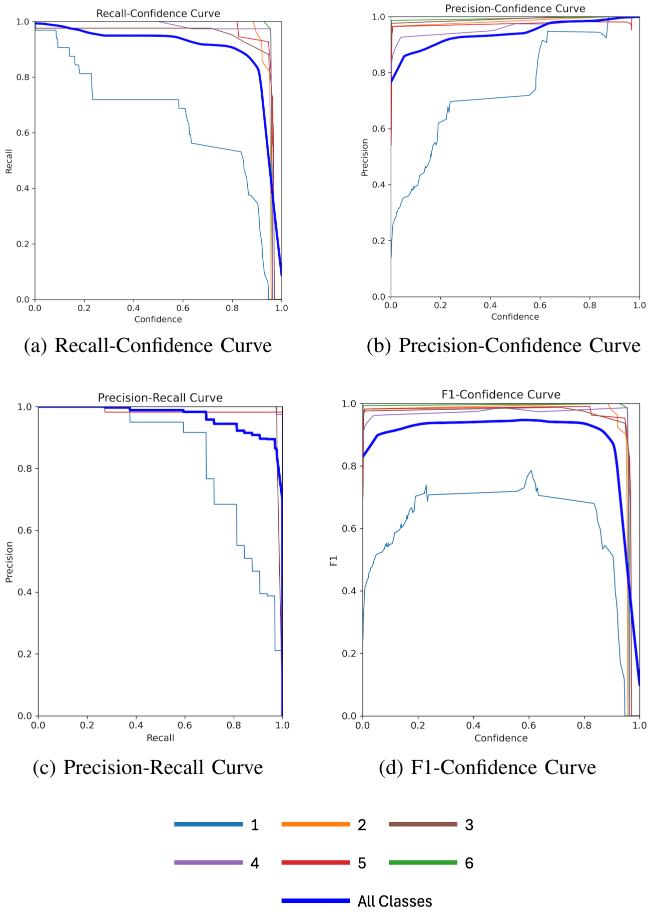
A. *Model Test and Evaluation*

Following model training and fine-tuning on the annotated dataset, the initial evaluation metric analyzed was the confusion matrix, as shown in Fig. 4. Notably, Class 1 (empty burner) exhibited lower performance, which was anticipated due to the visual similarity between the induction stovetop surface and the background. However, this limitation is not critical, as the model still effectively identifies the absence of a cooking pot - an outcome sufficient for maintaining the integrity of the cooking step classification.

In addition to the confusion matrix, we evaluated the model using standard object detection metrics: precision, recall, F1, mean Average Precision at Intersection over Union (IoU) threshold 0.5 (mAP50), and mean Average Precision calculated at varying IoU thresholds, ranging from 0.50 to 0.95 (mAP50-95). mAP50 is a measure of the model's accuracy considering only the "easy" detections while mAP50-95 gives a comprehensive view of the model's performance across different levels of detection difficulty. All of these metrics provide a comprehensive understanding of model performance across all six cooking states. As shown in Table I and Fig. 5, the model achieved a precision of at least 0.92 at confidence of 0.8. The recall is 0.55 for Class 1 (empty burner) at confidence of 0.8 while the recall is at least 0.93 for the other classes. These values indicate a strong ability to correctly identify most cooking stages with minimal false positives or negatives. The mAP@0.5 score, calculated across all classes, was 0.96, which reflects a high level of accuracy in both object localization and classification.

TABLE I: Precision, Recall, F1, and Confidence Evaluation Metrics results

	Classes					
Metric	**1**	**2**	**3**	**4**	**5**	**6**
Prec.@IoU=0.5	0.83	1.00	0.99	0.99	0.98	1.00
Prec.@Conf.=0.8	0.92	1.00	1.00	1.00	1.00	0.97
Recall@Conf.=0.8	0.55	1.00	0.93	1.00	0.98	1.00
F1@Conf.=0.8	0.69	1.00	0.96	1.00	0.99	0.98



(a) Recall-Confidence Curve

(b) Precision-Confidence Curve

(c) Precision-Recall Curve

(d) F1-Confidence Curve

Fig. 5: Precision, Recall, and F1 Scores at all values of confidence levels

These results suggest that the YOLO11S model, even when trained on a dataset without pre-processing, performs robustly and generalizes well across the selected cooking scenarios. The use of pre-trained weights, combined with focused fine-tuning, played a significant role in achieving these outcomes.

Fig. 6 presents the model's performance across 80 epochs, highlighting key metrics including precision, recall, mAP50 and mAP50-95. The results indicate that performance stabilizes as training progresses, plateauing toward the final epochs. Notably, all four metrics consistently exceed 0.90, demonstrating the high accuracy and overall quality of the model.

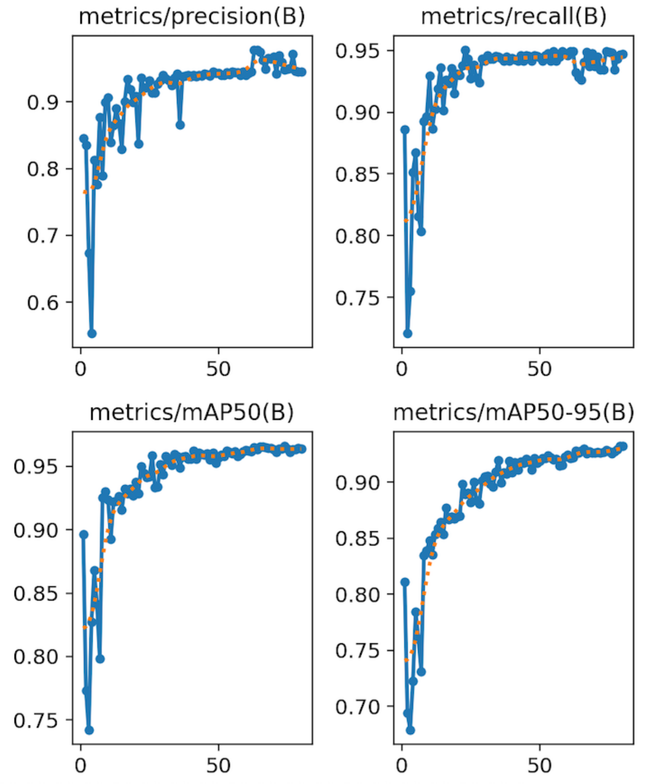


Fig. 6: Metrics progression by training epochs

TABLE II: Confidence Levels (%) of classes detection during UAT

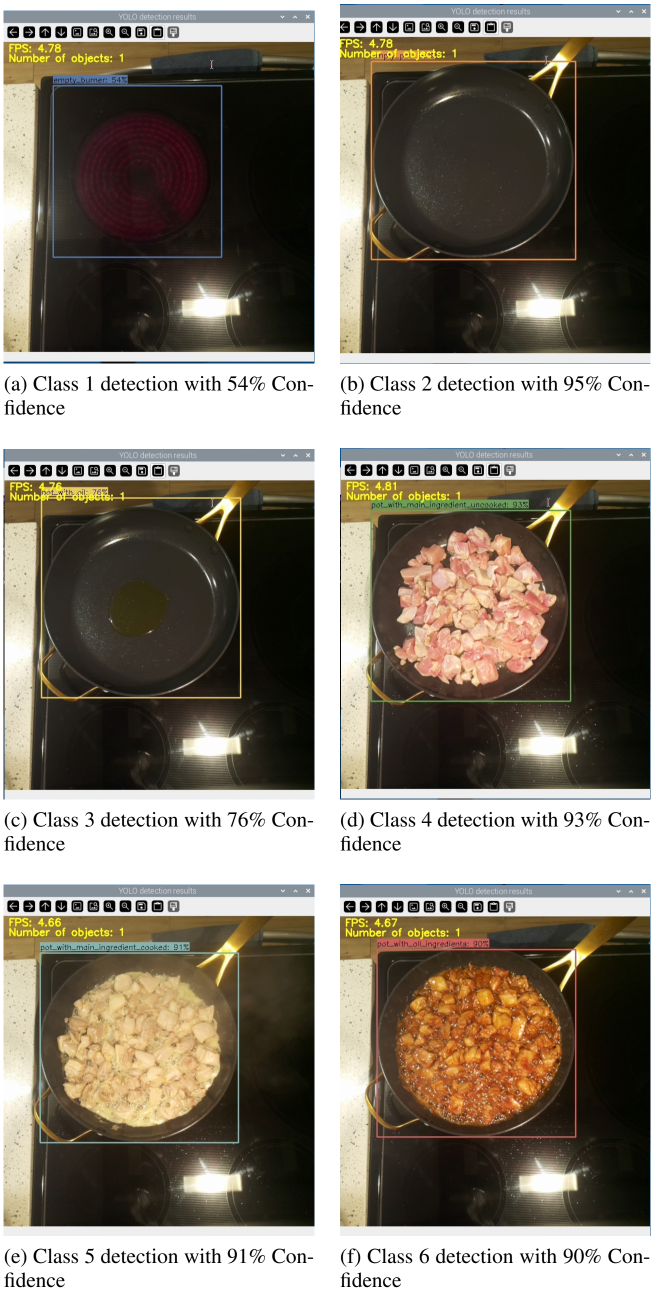
	Classes					
Test	**1**	**2**	**3**	**4**	**5**	**6**
Real time test env.	54	95	76	93	91	90
External media sources	75	95	-	-	-	-

B. *User Acceptance Test and Deployment*

While the training results and evaluation metrics for the model were promising, it is essential to conduct User Acceptance Testing (UAT) to assess the system's performance in real-world scenarios. We converted the trained model to the NCNN format to optimize inference performance on our edge system. The converted model was subsequently deployed to our hardware system.

To carry out UAT, we executed a series of structured evaluation steps, as detailed below.

1) **Real Time Cooking Process Analysis in Controlled Test Environment:** We deployed the system to analyze the complete chicken teriyaki recipe in real time, within the same controlled environment where the training data was originally collected. The process was successful, with representative results shown in Fig. 7 and Table II.
2) **Evaluation with External Media Sources:** To assess the model's generalizability, we recorded images and videos of various cooking steps using an iPhone, independent of the system's native camera. These external media files were then processed through the deployed system. The results were successful, as demonstrated in Table II.



(a) Class 1 detection with 54% Confidence

(b) Class 2 detection with 95% Confidence

(c) Class 3 detection with 76% Confidence

(d) Class 4 detection with 93% Confidence

(e) Class 5 detection with 91% Confidence

(f) Class 6 detection with 90% Confidence

Fig. 7: Images with the confidence Levels (%) of the detected class during UAT

C. *Assumptions and Limitations*

Several key assumptions underlie the development of this project, each significantly influencing its overall scope. Foremost among these is the issue of privacy. Given the unique experiences and vulnerabilities of the target population, it is challenging to predict their perceptions of privacy accurately. For example, if a hardware malfunction occurs that requires in-person technical support and the affected individual declines access, the smart cooking assistant would become inoperative. Furthermore, there exists a persistent stigma that government entities may use such devices for surveillance, a perception that can be difficult to dispel. At this stage, the project proceeds under the assumption that a majority of users will be willing to adopt and incorporate the smart cooking assistant into their daily routines.

Another critical assumption concerns the availability of Wi-Fi and user access to a smartphone. As the system currently lacks a physical display, user interaction is designed to occur exclusively through a smartphone application. This strategy also complements other initiatives funded under the same grant, such as the development of a smart "medicine dispenser," supporting a centralized and cohesive user experience across devices.

Additionally, the design assumes the presence of a range hood situated directly above the stove, providing a stable location for mounting the smart "cooking assistant". It is also assumed that the camera will have a clear, square view of the stovetop area where cooking activities occur, which is essential for accurate monitoring and feedback.

The project also faces two primary limitations. The first involves hardware constraints. The prototype is built on a Raspberry Pi 5 mini-computer, which offers limited processing and memory capabilities. As the system evolves, these resource limitations must be carefully managed to maintain consistent performance. The second limitation pertains to the training dataset. The current YOLO11 model has been trained on a relatively small and narrowly focused dataset. As a result, if users deviate significantly from predefined recipe steps, the model's performance may degrade. Moreover, the limited dataset constrains the range of recipes the system can currently support. Addressing this limitation through dataset expansion and diversification is a key goal for future system iterations to enhance adaptability and user experience.

V. Conclusions and Future Work

This research presents the design, development, and evaluation of a smart "cooking assistant" system, intended to enhance the home cooking experience through AI-powered visual recognition and real-time guidance. Leveraging the capabilities of the Raspberry Pi 5, AI HAT+, and AI Camera, we developed a cost-effective, privacy-conscious edge device capable of recognizing critical stages of a cooking process without reliance on continuous cloud connectivity. The system was trained using a custom-collected and annotated dataset, and the YOLO11S model was selected and optimized for edge deployment.

All model inference and state recognition are executed entirely on the system, ensuring the system remains a self-contained, closed-loop solution that prioritizes user privacy. Comprehensive evaluation - including model performance analysis, and user acceptance testing in real-world scenarios - demonstrated the system's reliability in classifying cooking states and delivering seamless, real-time guidance.

While results from this study are promising, several avenues for improvement have been identified. Feedback from live demonstrations emphasized the need to expand the dataset, refine model performance in visually ambiguous scenarios, and introduce greater personalization through a user interface. Future work will include the incorporation of additional recipes, improved model robustness under varying lighting conditions and kitchen layouts, and integration of the previously collected video data. Additionally, thermal imaging data is to be incorporated to improve detection of more nuanced cooking states, such as, empty burners, pot with water or pot with boiling water, particularly where visual cues are insufficient.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 2125654.

References

[1] H. I. County of Los Angeles, "Homeless initiative." [Online]. Available: https://www.lahsa.org/documents?id=8170-los-angeles-county-hc2024-data-summary

[2] United States Interagency Council on Homelessness, "Supportive housing," 2017, https://www.usich.gov/solutions/housing/supportive-housing (accessed: February 26, 2018).

[3] E. F. Sprake, J. M. Russell, and M. E. Barker, "Food choice and nutrient intake amongst homeless people," *Journal of Human Nutrition and Dietetics 27.3*, pp. 242–250, 2014.

[4] J. L. Wiecha, J. T. Dwyer, and M. Dunn-Strohecker, "Nutrition and health services needs among the homeless," *Public Health Reports 106.4*, p. 364, 1991.

[5] F. Pecune, L. Callebert, and S. Marsella, "A recommender system for healthy and personalized recipes recommendations," *HealthRecSys@ RecSys*, 2020.

[6] M. Ge, F. Ricci, and D. Massimo, "Health-aware food recommender system," *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015.

[7] R. Yera, A. A. Alzahrani, and L. Martínez, "Exploring post-hoc agnostic models for explainable cooking recipe recommendations," *Knowledge-Based Systems 251*, vol. 109216, 2022.

[8] L. Zhu and et al, "Deep learning and machine vision for food processing: A survey," *Current Research in Food Science 4*, pp. 233–249, 2021.

[9] Y. Zhang, Y. Yamakata, and K. Tajima, "Stage-aware recognition method for foodstuffs changing in appearance in different cooking stages on chinese recipe," *Proceedings of DEIM*, vol. C13-3, 2021.

[10] S. Giroux and et al, "Cognitive assistance to meal preparation: design, implementation, and assessment in a living lab," *2015 AAAI Spring Symposium Series*, 2015.

[11] A. Yaddaden and et al, "Using a cognitive orthosis to support older adults during meal preparation: Clinicians' perspective on cook technology," *Journal of Rehabilitation and Assistive Technologies Engineering 7*, vol. 2055668320909074, 2020.

[12] H. Ngankam and et al, "Real-time multiple object tracking for safe cooking activities," *International Conference on Smart Homes and Health Telematics. Cham: Springer Nature Switzerland*, 2023.

[13] A. B. Jelodar, M. S. Salekin, and Y. Sun, "Identifying object states in cooking-related images," *arXiv preprint arXiv*, vol. 1805.06956, 2018.

[14] G. Ruiz, S. C. Kilambi, P. Soni, K. George, and A. Panangadan, "Design of a multisensor system for a smart cooking assistant," *In IEEE International Conference on Artificial Intelligence x Medicine, Health, and Care (AIMHC)*, vol. IEEC, 2024.

[15] J. Redmon and et al, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.