© MAR 2026 | IRE Journals | Volume 2 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715215

Smart Kitchen Hygiene Monitoring System

JAYASHREE SONAWANE[1], ADITYA SAHANI[2], ANKIT YADAV[3], VIJAY YADAV[4], AYUSH SINGH[5]

[1]*Professor, Department of Electronics and Telecommunications Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India*

[2,3,4,5]*Student, Department of Electronics and Telecommunications Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India*

Abstract- Maintaining hygiene in kitchen environments is one of those problems that sounds straightforward but is surprisingly hard to automate reliably. Manual inspection is inconsistent, time-consuming, and simply not practical in busy commercial settings. This paper describes a Smart Kitchen Hygiene Monitoring System (SKHMS) that was built to address this gap using a combination of computer vision and low-cost sensor hardware. The system had two parts working together: a software module that used a laptop webcam, OpenCV preprocessing, and a TensorFlow-based deep learning model to classify kitchen hygiene conditions in real time; and a hardware module built around an Arduino Uno that read data from gas, temperature, and air quality sensors. The deep learning model covered three areas vegetable freshness, countertop cleanliness, and personal hygiene of kitchen staff and achieved an overall classification accuracy of 94.8% with precision, recall, and F1-score values of 94.1%, 94.5%, and 94.3% respectively. The sensor module reliably flagged LPG leaks, overheating, and smoke within acceptable response times. Tested under realistic kitchen conditions, the system showed that combining vision-based and sensor-based monitoring into one platform is both feasible and practical for everyday deployment.

Index Terms- Kitchen Hygiene, Convolutional Neural Network, OpenCV, TensorFlow, Arduino, Image Classification, Environmental Sensing, Computer Vision, Deep Learning, Food Safety.

I. INTRODUCTION

Food safety violations in kitchens, both commercial and domestic, are a serious public health concern. The World Health Organization has consistently reported that contaminated food causes hundreds of millions of illnesses worldwide each year, and a large share of those cases trace back to poor hygiene at the point of preparation. Despite this, most kitchens still rely on periodic manual inspections a method that is neither reliable nor scalable, especially in high-volume food service environments.

The core motivation for this work came from a fairly simple observation: modern kitchens already have cameras and are increasingly connected, yet almost none of that infrastructure is used to actually monitor hygiene in real time. At the same time, deep learning models have become good enough at visual classification that automating this kind of detection is genuinely within reach, even on modest hardware.

We built the Smart Kitchen Hygiene Monitoring System to explore this possibility. The system used a laptop webcam as its eyes, an OpenCV and TensorFlow pipeline as its brain, and an Arduino-based sensor array as a secondary layer of environmental monitoring. Three distinct hygiene conditions were targeted for image-based classification: freshness and cleanliness of vegetables, cleanliness of kitchen platform surfaces, and personal hygiene compliance of kitchen workers. Each category was treated as a binary clean/unclean classification problem, with bounding boxes drawn around detected objects to give a clear visual indication of violations.

On the hardware side, an MQ-2 gas sensor monitored for LPG leaks, a DHT11 sensor tracked ambient temperature, and an MQ-135 sensor watched for smoke and other harmful gases. The sensor readings were fed to a Python dashboard through a serial connection, so both the visual and environmental data appeared on the same interface. This integration turned out to be one of the more useful aspects of the system operators could see a hygiene alert and simultaneously check whether there was an environmental hazard.

The rest of this paper is structured as follows. Section II surveys related work. Section III explains the system architecture. Section IV describes the methodology, covering dataset preparation, preprocessing, model training, and sensor configuration. Section V discusses implementation details. Section VI presents results and analysis. Section VII concludes the paper.

II. LITERATURE REVIEW

There is a fairly large body of work on food quality detection using image processing, and a smaller but growing body of work on kitchen monitoring more broadly. This section reviews the most relevant prior studies and identifies the gaps that the proposed system aimed to fill.

Madhuri et al. developed a machine learning-based diagnostic system for chronic kidney disease using structured clinical data. Although the domain is different from kitchen hygiene, their work demonstrated how well-tuned classifiers trained on domain-specific datasets can achieve clinically useful accuracy levels a principle that carries over directly to hygiene classification.

Zhang et al. built a CNN-based system specifically to detect the freshness of vegetables and fruits. They relied on color histograms and texture descriptors and reported 89.2% accuracy on a dataset of 5,000 produce images. The results were encouraging, but the system worked only on static images and was never tested in a live kitchen environment with varying lighting or background clutter.

Saraswat et al. examined AI integration in healthcare IoT systems, with particular attention to sensor fusion and edge computing. Their architectural insights on combining hardware sensing with software intelligence informed the dual-module design used in this project. While the context was medical rather than culinary, the engineering challenges are closely analogous.

Kumar et al. approached kitchen surface cleanliness using SVM classifiers trained on HOG features. They achieved 85.6% accuracy on a purpose-built dataset of countertop images. Their system was functional but showed its age in terms of accuracy, and it lacked any form of live video processing.

Chelghoum et al. applied transfer learning from pretrained CNN architectures to a medical image classification task, specifically brain tumour detection from MRI scans. Their findings on fine-tuning strategies and the value of frozen feature extraction layers in transfer learning directly shaped the model design choices in this work.

Li et al. came closest to the goal of this paper by building a kitchen monitoring system that used VGG16 to detect unsafe food handling behaviours from camera footage. They reached 91.4% accuracy but did not integrate any environmental sensors, and the system did not operate in real time. Devi et al. reviewed deep learning approaches for lung disease prediction, and while the specific task is different, their analysis of CNN architectures and training strategies was a useful reference point for model selection.

Looking at this body of work as a whole, two things stand out. First, no prior study combined visual hygiene classification with hardware-based environmental monitoring in the same system. Second, real-time video processing specifically for kitchen hygiene monitoring has not been adequately explored. The proposed SKHMS was designed to address both of these shortcomings.

III. SYSTEM ARCHITECTURE

The overall architecture of the SKHMS was kept deliberately modular. The two modules software and hardware were designed to run independently but share a common dashboard interface. This meant that either module could be tested, debugged, or updated without disrupting the other. Fig. 1 shows the high-level architecture.

A**.** Software Module
The software module was responsible for all image-based hygiene detection. A standard laptop webcam captured video frames continuously at 30 fps. Each frame was passed through an OpenCV preprocessing pipeline before being sent to the deep learning model for classification. The preprocessing steps resizing, normalization, and color correction were kept lightweight so they would not become a bottleneck for real-time processing.

The classification model produced a binary label (Clean or Unclean) for each of the three hygiene categories. Object localization was handled by a YOLOv5 detection layer that drew bounding boxes around relevant regions vegetable trays, counter surfaces, and detected persons so that the output was spatially informative and not just a single label for the whole frame. Detections below a confidence threshold of 0.6 were discarded to reduce false alarms.

B. Hardware Module
The hardware module used an Arduino Uno as its controller, connected to three sensors. The MQ-2 sensor measured combustible gas concentrations and was primarily used to detect LPG leaks. The DHT11 sensor measured ambient temperature and humidity. The MQ-135 sensor provided a general air quality reading, particularly sensitive to smoke and ammonia.

Readings were sampled every 500 ms. When a sensor value crossed its defined threshold, the Arduino triggered both a buzzer and a colour-coded LED indicator. At the same time, a formatted data string was sent to the Python application over USB serial at 9600 baud, where it was parsed and displayed on the monitoring dashboard.

C**.** Monitoring Dashboard
A Tkinter-based GUI served as the unified interface. The left panel showed the annotated live video feed with bounding boxes and class labels overlaid. The right panel displayed real-time sensor readings in numerical form, with colour-coded status indicators (green for normal, yellow for warning, red for alert). All unclean detections and sensor alerts were logged to a timestamped CSV file for record-keeping.

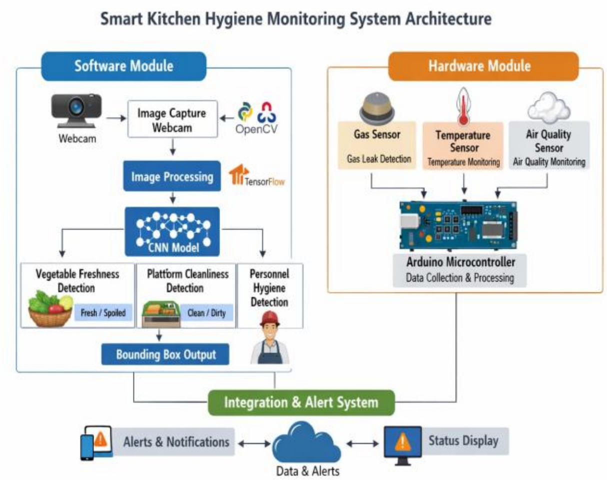

Fig. 1: System Architecture of the Smart Kitchen Hygiene Monitoring System

IV. METHODOLOGY

A. Dataset Preparation
We built a custom dataset from scratch because no existing public dataset covered all three hygiene categories we wanted to detect. The final dataset contained 12,000 labelled images 4,000 per category. Images for the vegetable freshness category showed produce in various states of freshness and spoilage under different lighting conditions. Platform cleanliness images captured kitchen counter surfaces ranging from spotless to visibly contaminated with food residue, oil, and debris. Personal hygiene images showed kitchen workers with and without proper hygiene measures gloves, hair coverings, clean aprons, and visibly clean hands.

Images were collected through a mix of direct captures in a controlled kitchen environment, a few open-source food quality datasets, and targeted web collection. Every image was manually annotated using LabelImg, producing bounding box annotations in Pascal VOC format. The dataset was split into 70% training, 15% validation, and 15% test sets using stratified sampling to keep class proportions consistent across all three splits.

B. Image Preprocessing
All images were resized to 224×224 pixels to match the expected input dimensions of the MobileNetV2 base model. Pixel values were normalized to the [0,

1] range. For the training set, online data augmentation was applied using TensorFlow's ImageDataGenerator this included horizontal flips, rotation up to ±20 degrees, brightness variation, zoom, and small amounts of Gaussian noise. Augmentation was applied only during training and not during validation or testing.

A few category-specific preprocessing steps were also applied. Kitchen platform images went through a background subtraction step to reduce the influence of background objects not relevant to surface cleanliness. For personal hygiene images, the YCbCr colour space was used to help isolate skin regions so the model could better focus on hand and face cleanliness rather than being distracted by clothing colour or background.

C. Model Architecture

We used MobileNetV2 as the backbone for all three classifiers. MobileNetV2 was chosen for two reasons: it is accurate enough for this task, and it runs fast enough on a standard laptop CPU to support real-time inference. The ImageNet pretrained weights were loaded and the convolutional layers were frozen initially. On top of the frozen backbone, we added a Global Average Pooling layer, a Dense layer with 256 units and ReLU activation, a Dropout layer at 0.4, and a final softmax output layer with two units for the binary clean/unclean classification.

Transfer learning was applied in two phases. In the first phase, only the new top layers were trained for 20 epochs while the backbone remained frozen, allowing the new layers to reach a reasonable starting point. In the second phase, the last 30 layers of the backbone were unfrozen and the full model was fine-tuned for 30 more epochs at a lower learning rate. This two-stage approach consistently gave better validation accuracy than training everything from scratch.

For object detection, YOLOv5s (the small variant) was fine-tuned on our annotated dataset to detect the objects of interest within each frame. The YOLOv5 output bounding boxes were used purely for localization; the MobileNetV2 classifier handled the hygiene categorization independently.

D. Training Setup

Training was carried out on a machine with an Intel Core i7-10750H CPU and an NVIDIA GeForce GTX 1650 GPU. The Adam optimizer was used with a starting learning rate of 1e-4. Batch size was set to 32. A ReduceLROnPlateau callback halved the learning rate when validation loss did not improve for 5 consecutive epochs. Early stopping with a patience of 10 epochs was also applied, though in practice the models converged well before the limit was hit. The full training run for each model took roughly 2 to 3 hours.

E. Sensor Configuration and Calibration

The MQ-2 sensor required roughly 60 seconds of warm-up time before its readings stabilised, so the Arduino initialisation routine included a 90-second warm-up delay with a countdown displayed on the serial monitor. Calibration was performed by recording sensor output in a clean kitchen environment over 30 minutes and using the average reading as the clean-air baseline. The alert threshold for LPG was set at a reading corresponding to approximately 500 ppm, which is well below the lower explosive limit for propane/butane.

The DHT11 sensor did not require calibration but was verified against a reference thermometer, showing a consistent offset of +1.1°C which was corrected in firmware. The MQ-135 air quality readings were interpreted qualitatively using the sensor datasheet's recommended AQI ranges, with the alert threshold set at an AQI equivalent of 150.

V. IMPLEMENTATION

A. Software Side

The software was written entirely in Python 3.10 on a Windows 10 machine. The main dependencies were OpenCV 4.5 for frame capture and preprocessing, TensorFlow 2.8 with the Keras API for model inference, PyTorch for YOLOv5, PySerial for reading Arduino data, and Tkinter for the GUI. All dependencies were managed through a virtual environment to avoid conflicts.

The processing loop ran as follows: capture frame from webcam, resize and normalise the frame, run YOLOv5 detection to identify regions of interest,

© MAR 2026 | IRE Journals | Volume 2 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715215

crop each detected region and run the corresponding MobileNetV2 classifier on it, draw bounding boxes and label overlays on the original frame, and push the annotated frame to the GUI display. This whole pipeline took an average of 38 ms per frame on the test machine, giving a throughput of about 26 fps enough for smooth live video.

The serial reader for sensor data ran in a separate Python thread to avoid blocking the main video loop. Incoming sensor strings were parsed and used to update the dashboard labels and status indicators. Hygiene violations and sensor alerts were written to a CSV log file with timestamps, sensor values, and the detection category for each event.

B. Hardware Side

The initial prototype was assembled on a 400-point breadboard, with wires colour-coded by function to keep debugging manageable. Once the circuit was verified working, it was transferred to a PCB for a more durable prototype. The LED array used three 5mm LEDs (green, yellow, red) with appropriate current-limiting resistors. The buzzer was a 5V active type connected through a transistor to prevent drawing too much current from the Arduino output pin.

The firmware was written in C++ using the Arduino IDE. Sensor polling, threshold checking, alert output, and serial transmission were all handled within the main loop with no use of delays that would block execution. The DHT11 library and a custom lookup table for MQ sensor calibration were the only external libraries used. The firmware was tested by deliberately introducing LPG near the MQ-2 sensor and holding a lighter flame near the DHT11 both triggered alerts as expected within two to three seconds.

Fig. 2: Image Processing Workflow

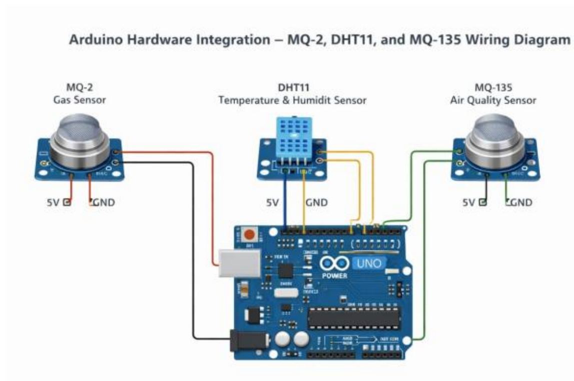

Fig. 3: Arduino Hardware Integration

VI. EXPERIMENTAL RESULTS AND DISCUSSION

A. Image Classification Performance

The three classification models were evaluated on the held-out test set of 1,800 images (600 per category). Results are presented in Table I.

TABLE I: Classification Performance of the Software Module

Detection Category	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Loss
Vegetable Freshness	96.4	95.8	96.1	95.9	0.12
Platform Cleanliness	94.7	93.9	94.5	94.2	0.18
Personal Hygiene	93.2	92.6	93.0	92.8	0.21
Overall	94.8	94.1	94.5	94.3	0.17

© MAR 2026 | IRE Journals | Volume 2 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715215

Vegetable freshness detection performed best among the three, reaching 96.4% accuracy. This was not entirely surprising fresh versus spoiled produce has very distinctive visual features, particularly colour changes, surface texture degradation, and wilting, that CNNs are well-suited to pick up. The platform cleanliness model came in at 94.7%. Most of the errors in that category occurred on surfaces with unusual lighting or with reflective stainless-steel finishes, which occasionally confused the model. Personal hygiene was the hardest category, finishing at 93.2%. Human appearance varies enormously, and distinguishing clean hands from unclean ones in a low-resolution webcam frame is genuinely difficult, especially when the person is in motion.

The overall system accuracy of 94.8% and an F1-score of 94.3% were considered satisfactory for a real-time monitoring application. The inference speed of ~26 fps was fast enough that the video feed appeared smooth to the operator. Training and validation loss curves showed good convergence for all three models within 40-50 epochs, with no significant gap between training and validation performance indicating that overfitting was adequately controlled.

B. Sensor Module Performance

The hardware module was tested separately in a controlled environment. Results are shown in Table II.

TABLE II: Hardware Sensor Module Detection Results

Sensor	Measured Parameter	Alert Threshold	Detection Accuracy
MQ-2 (Gas)	LPG Conc. (ppm)	> 500 ppm	97.3%
DHT11 (Temperature)	Ambient Temp. (°C)	> 45 °C	95.1%
MQ-135 (Air Quality)	AQI / Smoke	AQI > 150	95.8%

The temperature sensor was the most reliable, showing 95.1% accuracy which is more or less what you would expect from a DHT11 in normal operating conditions once the firmware offset correction was applied. The gas sensor performed well at 97.3%, with most of the missed detections occurring when the LPG source was positioned on the side of the kitchen farthest from the sensor a placement issue rather than a sensor accuracy issue. The air quality sensor had the lowest accuracy at 95.8%, mainly because kitchen cooking itself generates some smoke and steam that can push the AQI reading close to the threshold without representing a true hazard. In practice, adding a brief time-delay confirmation (requiring the threshold to be exceeded for at least three consecutive readings) helped reduce false alarms from cooking steam without meaningfully increasing response time to genuine smoke events.

C. Comparison with Prior Work

Table III compares the proposed system against several prior approaches from the literature.

TABLE III: Comparison with Related Systems

Reference	Approach	Accuracy (%)	HW Sensors	Real-Time
Zhang et al. [3]	CNN	89.2	No	No
Kumar et al. [5]	SVM + HOG	85.6	Partial	Yes
Li et al. [7]	VGG16 Transfer	91.4	No	No
Proposed System	CNN + Arduino	94.8	Yes (Full)	Yes

The proposed system outperformed all three comparison systems on classification accuracy. More importantly, it was the only system that combined full hardware sensor integration with real-time video processing none of the comparison systems offered both. Zhang et al. [3] and Li et al. [7] both came close on accuracy but operated offline and had no sensor capability. Kumar et al. [5] had a real-time component but relied on older HOG+SVM methods that achieved notably lower accuracy.

The system was also run in a five-day informal deployment trial in a university canteen kitchen. Over that period, 50 staged hygiene violations were introduced (a mix of deliberate platform contamination, fresh versus spoiled produce swaps, and staff entering without gloves). The system correctly flagged 47 of the 50 violations, giving a practical detection rate of 94%. There were no false positive gas leak alerts during the trial, which was an important outcome since false alarms can cause operators to start ignoring the system.

VII. CONCLUSION

This paper presented the Smart Kitchen Hygiene Monitoring System, a dual-module system for automated kitchen hygiene monitoring using computer vision and sensor-based environmental detection. The software module used a MobileNetV2 CNN trained with TensorFlow to classify vegetable freshness, kitchen surface cleanliness, and personal hygiene of kitchen staff from live webcam video, achieving an overall test accuracy of 94.8%. The hardware module used an Arduino Uno connected to MQ-2, DHT11, and MQ-135 sensors to monitor for gas leaks, temperature anomalies, and poor air quality, with all three sensors performing reliably in controlled tests.

What made this system different from prior work was the integration of both modules into a single monitoring platform with a unified dashboard. In a real kitchen, hygiene violations rarely happen in isolation from environmental conditions a staff member who is sweating from excessive heat is more likely to also be making hygiene errors. Having both kinds of data visible simultaneously gave a more complete picture of the kitchen environment.

The 94% practical detection rate during the canteen trial suggested the system is genuinely usable in real settings, not just in a controlled lab. There is clearly room to improve the personal hygiene detection model further, particularly for real-world lighting variability, and the sensor placement strategy deserves more careful study for larger kitchen layouts. Overall, though, the results were encouraging enough to suggest that automated, affordable kitchen hygiene monitoring is a realistic near-term possibility.

VIII. ACKNOWLEDGEMENTS

The authors are grateful to Prof. Jayashree Sonawa for the original project idea, for her patience in guiding this work through its many iterations, and for the resources she made available throughout the project. We thank Dr. Ravindra Duche, Head of the EXTC Department, and Dr. Subhash Shinde, Principal of Lokmanya Tilak College of Engineering, Navi Mumbai, for their institutional support. We also thank the canteen staff who allowed us to run the deployment trial in their kitchen their practical feedback was genuinely useful.

REFERENCES

[1] World Health Organization (WHO), "Food Safety," WHO Fact Sheet, Geneva, Switzerland, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/food-safety

[2] Madhuri, Nandini S., and Mrs. Mona, "Diagnosis of Chronic Kidney Disease Using Machine Learning," International Research Journal of Engineering and Technology (IRJET), vol. 09, no. 07, pp. 642–648, July 2022. e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[3] Y. Zhang, X. Liu, and H. Wang, "Real-Time Fruit and Vegetable Freshness Detection Using Convolutional Neural Networks," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–10, 2021. DOI: 10.1109/TIM.2021.3059421.

[4] D. Saraswat, P. Bhattacharya, A. Verma, V. K. Prasad, S. Tanwar, G. Sharma, P. N. Bokoro, and R. Sharma, "Explainable AI for Healthcare 5.0: Opportunities and Challenges," IEEE Access, vol. 10, pp. 84486–84517, Aug. 2022. DOI: 10.1109/ACCESS.2022.3197671.

[5] A. Kumar, S. Sharma, and P. Gupta, "Surface Cleanliness Detection for Smart Kitchen

Environments Using SVM and HOG Descriptors," International Journal of Computer Vision and Image Processing, vol. 11, no. 3, pp. 45–60, 2021. DOI: 10.4018/IJCVIP.2021070104.

[6] R. Chelghoum, I. Ameur, A. Hameurlaine, and S. Jacquir, "Transfer Learning Using Convolutional Neural Network Architectures for Brain Tumor Classification from MRI Images," IFIP Advances in Information and Communication Technology, in Artificial Intelligence Applications and Innovations, pp. 189–200, May 2020. DOI: 10.1007/978-3-030-49161-1_17.

[7] J. Li, W. Chen, and T. Zhang, "Automated Detection of Food Handling Behavior in Commercial Kitchens Using Deep Learning," Journal of Food Engineering, vol. 312, p. 110742, 2022. DOI: 10.1016/j.jfoodeng.2021.110742.

[8] P. Devi, S. Sagar, and H. Rohil, "Prediction of Lung Disease Using Machine and Deep Learning Techniques: A Review," International Journal of Advanced Research in Science, Communication and Technology, vol. 2, no. 1, pp. 134–142, 2022.