International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

AI Based Home Security Camera System

Chitapong Wechtaisong[1], Sunanta Radomboon[1], Pratiparn Chawala[1], Kanokporn Boonjubut[2], Thanaphon Sakonphattharasorn[2], Anawin Pechbooranin[1]
[1]Institute of Engineering, Suranaree University of Technology, Nakhon Ratchasima, Thailand-30000
[2]Faculty of Industrial Technology, Nakhon Ratchasima Rajabhat University, Nakhon Ratchasima, Thailand-30000

Abstract— *This article presents the design and development of an AI based security camera system. The objective is to recognize and classify faces in a database using artificial intelligence and to send intrusion alerts for unidentified faces through LINE Notify. The security camera system employs the YOLOv5 model for human detection and utilizes face recognition technology for face detection and identification. To process data, the system uses a Raspberry Pi 4 Model B board in conjunction with a Full HD 1080P webcam for image capture. Results from experiments involving 11 volunteers show that the AI based security camera system can accurately classify detected faces with an average accuracy of 91.72% and a mean average precision of 89.54%.*

Keywords— *Artificial intelligence, Human detection, Face recognition.*

I. INTRODUCTION

At present, technological advancements are progressing rapidly, leading to the continuous invention and development of new technologies that enhance human convenience. One such technology is Artificial Intelligence (AI), which involves equipping machines and computers with capabilities through algorithms and statistical tools to create intelligent software capable of emulating complex human abilities, such as recognition, differentiation, reasoning, and decision-making.

The authors place significant emphasis on safety, as the protection of life and property is a fundamental concern for all individuals. A society with low levels of security faces increased risks of loss of life and property. Consequently, a high-quality security system is essential. Security systems employing closed-circuit television (CCTV) cameras are widely adopted as a basic measure for capturing images and recording videos, which can be instrumental in identifying intruders. However, relying solely on CCTV cameras may not effectively prevent unauthorized intrusions.

To address this limitation, the team has integrated Human Detection technology, which identifies human presence, and Face Recognition technology, which recognizes and differentiates faces based on facial structures, into a security system. This system is implemented using a Raspberry Pi board and a camera module, with programming in Python, HTML, and CSS. Tools such as Visual Studio Code, GitHub, and Thonny were utilized to develop a security camera system capable of detecting intrusions by comparing faces in the captured images against a pre-stored database. This allows the system to effectively alert users to unauthorized intrusions.

A. Objectives

- To recognize and classify faces using artificial intelligence technology.
- To alert unauthorized intrusions by individuals whose facial data is not present in the database via LINE Notify.

B. Scope of the Study

- The system was tested using a Raspberry Pi board with a 4-core 64-bit CPU and 8GB RAM, a notebook with an Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, 64-bit, and a Full HD 1080P DI01 webcam.
- The brightness level in the detection area was set between 5500 and 7000 lux.
- The distance between the object and the camera was defined within the range of 1 to 2.5 meters.

II. RELATED THEORY AND RESEARCH WORKS

In this section, we review the relevant literature and theoretical foundations pertaining to the proposed AI-based home security camera system.

A. Related Research Works

Research on security camera systems utilizing face recognition and human detection technologies frequently incorporates object detection techniques to identify entities such as humans, faces, animals, or objects. Popular object detection methods include R-CNN, Fast R-CNN, SPP-net, and YOLO. However, a key challenge associated with these techniques is their insufficient processing speed for real-time applications.

In 2016, Redmon J. and colleagues introduced a real-time integrated object detection model known as YOLO (You Only Look Once) [1][2]. This model leverages deep learning techniques, specifically Convolutional Neural Networks (CNNs) [3], as its foundational framework. YOLO performs object detection by resizing an input image into an S × S grid and dividing it into n × n cells, where a single neural network determines whether the detected object belongs to a specific dataset class. YOLO achieves a real-time image processing speed of 45 frames per second, while a smaller version can process images at up to 155 frames per second. Although YOLO may occasionally mispredict object locations, it has a lower likelihood of generating false positives. Following its release, YOLO gained widespread adoption. For instance, Chun L. Z. et al. (2020) employed YOLOv3, which integrates the Darknet-53 network and Viola-Jones face recognition, to detect human faces in complex environments [4]. Similarly,

Chitapong Wechtaisong, Sunanta Radomboon, Pratiparn Chawala, and Kanokporn Boonjubut, Thanaphon Sakonphattharasorn, Anawin Pechbooranin, "AI Based Home Security Camera System," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 1, pp. 12-17, 2025.

Menon S. et al. (2021) utilized YOLOv3 for customized real-time face recognition, demonstrating superior speed compared to R-CNN and Fast R-CNN [5].

Further advancements were made by Mantau, A. et al. (2022), who employed YOLOv5 and transfer learning with thermal imaging data from UAVs to detect humans in surveillance systems [6]. Their study utilized RGB and thermal infrared data to develop an optimized model for deployment on the Jetson Nano module. Experimental results showed that transfer learning using the MS COCO dataset significantly enhanced YOLOv5's performance in detecting humans and objects in RGBT datasets.

Kortli Y. et al. (2020) conducted a survey on face recognition systems, focusing on three key techniques: Local Approaches (processing entire facial images), Holistic Approaches (feature extraction-based processing), and Hybrid Approaches (combining both methods) [7]. Their evaluation highlighted Local Approaches, such as CNN-based methods, as the most effective in terms of discrimination, complexity, and accuracy.

Drawing from the aforementioned research, this project incorporates the capabilities of the YOLO model for object detection, specifically in human and facial detection. Additionally, Local Approaches are employed to develop a face recognition model. While YOLOv3 demonstrates limitations in processing speed and accuracy, this study opts for YOLOv5 combined with transfer learning to enhance performance and improve the precision of human face detection in security camera systems.

B. *Convolutional Neural Networks (CNNs)*

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing data with a grid-like topology, such as images and videos. The architecture of CNNs is composed of three main types of layers: Convolutional Layers, Pooling Layers, and Fully Connected Layers.

The Convolutional Layer is the core building block of a CNN. It consists of a set of learnable filters (or kernels) that slide over the input data to perform convolution operations. These filters are responsible for detecting local patterns such as edges, textures, and shapes. As the filters move across the input, they produce feature maps that capture the spatial hierarchies in the data. This process enables the network to learn and recognize complex patterns from raw input data [9].

Pooling Layers are used to reduce the spatial dimensions of the feature maps, thereby decreasing the number of parameters and computational load in the network. Common pooling operations include Max Pooling and Average Pooling. Max Pooling selects the maximum value from a patch of the feature map, while Average Pooling computes the average value. This dimensionality reduction helps in making the model more efficient and less prone to overfitting by retaining the most important features [9].

Fully Connected Layers, typically found at the end of the CNN architecture, are used to perform high-level reasoning and classification. Each neuron in a Fully Connected Layer is connected to every neuron in the previous layer, allowing the network to integrate the features extracted by the convolutional and pooling layers. The final layer often uses a Softmax activation function to output probabilities for each class, enabling the network to make predictions [10].

CNNs have demonstrated remarkable success in various computer vision tasks, including image classification, object detection, and image segmentation. Their ability to automatically learn and extract features from raw data makes them a powerful tool in the field of deep learning [11]

C. *Face Recognition*

This study incorporates face recognition technology to detect and identify individuals within a security camera system. Face recognition technology is designed to recognize human facial structures through the analysis of facial images used to train the model. The process involves detecting a face within an image, comparing it against a database, and identifying the individual if a match is found. Depending on the facial recognition approach, the comparison may involve analyzing either the entire face or specific facial features.

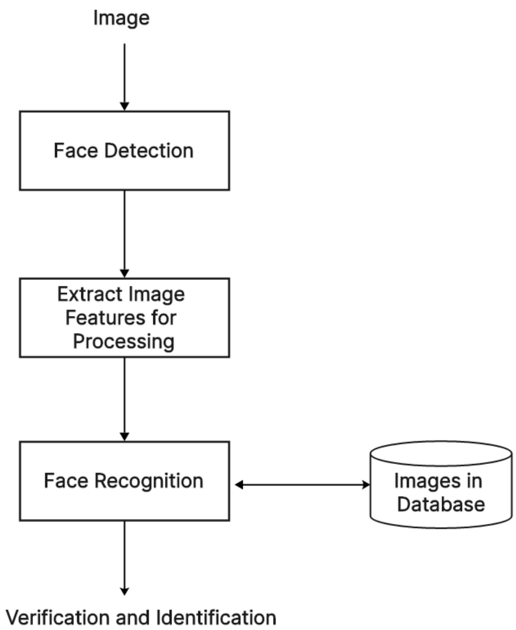

Fig. 1 Structure of Face Recognition

D. *YOLOV5*

The YOLOv5 model has been employed in this study for human detection in the security camera system. YOLO, which stands for "You Only Look Once," divides an input image into s × s grids and uses an algorithm to calculate the probability of each grid cell containing an object from the dataset. This model is widely used for object detection due to its speed and high accuracy.

YOLO processes an image in a single pass, labeling the image by dividing it into an s × s grid and assigning labels to each grid cell. This approach enables YOLO to efficiently and accurately detect objects within an image.

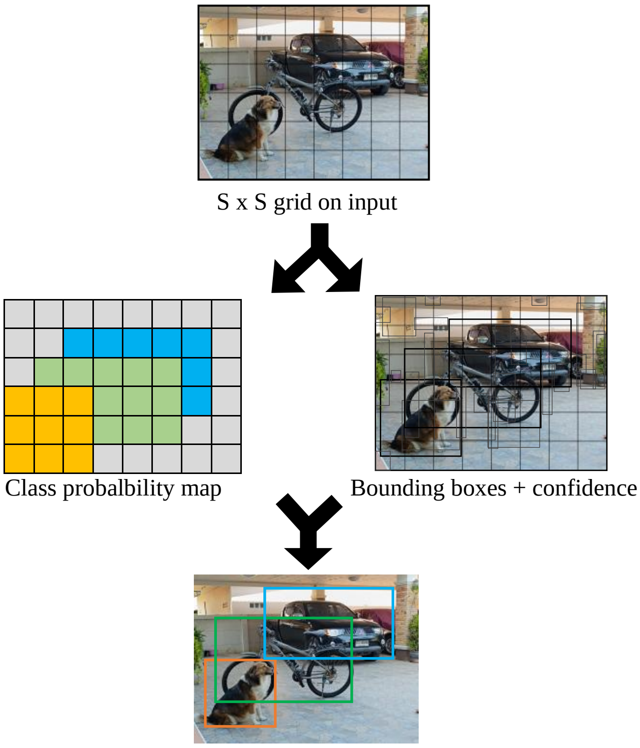

Fig. 2 YOLO Model Workflow [1]

III. DESIGN

The system design includes key components such as a Human Detection Program for identifying individuals, a Facial Recognition Program for precise identification, and an AI-Based Security Camera Website for data management. It also incorporates a LINE Notify Alert System for real-time notifications, Data Storage for logging essential information, and Wi-Fi Network Connectivity for seamless system integration. These elements combine to deliver an intelligent and efficient surveillance solution.

A. Human Detection Program

The developers designed a human detection program using the YOLOv5 model for object detection, filtering only the human class. When the program detects a human, the system draws a bounding box around the object in the frame.

B. Facial Recognition Program

The facial recognition program is designed using the Face Recognition library as the primary tool for comparing the encodings of detected faces with the encodings of known faces stored in a database. When the program identifies a person with matching data, it assigns the corresponding name to that individual. If an unknown individual is detected, the system draws a red bounding box and labels the person as "Unknown."

C. Development of an AI-Based Security Camera Website

The website for displaying camera feeds and accessing recorded data was developed using HTML and JavaScript for structuring the interface and CSS for styling. The design focuses on seamless access to information captured by the system.

D. LINE Notify Alert System For the LINE Notify alert function, the developers designed a system that allows users to update their LINE account credentials through the "Setting" page on the website. If the system detects an unknown individual, it sends an alert to the registered LINE account specified in the initial setup.

E. Data Storage

The security camera system stores the date, time, and location of unknown individuals in a MySQLite database. Each time an unknown face is detected, the system logs the date, time, and image file path into the .db file.

F. Wi-Fi Network Connectivity

To enable Wi-Fi network connectivity, the developers created a function allowing users to input Wi-Fi credentials through the "Setting" page on the website. This enables seamless connection to the network.

IV. OPERATION OF THE PROPOSED SYSTEM

The AI-based security camera system is developed using Python programming language and operates on a Raspberry Pi 4 board paired with a Full HD 1080p webcam. When the program initiates, the system activates the camera to capture and display real-time video on a dedicated website. The system detects humans within a specified bounding box in the video feed. Upon detecting a human, the program performs facial recognition to compare the detected face against a database of known individuals.

If the detected individual is not found in the database, the system triggers an alert via LINE Notify and records the date, time, and image location in a MySQLite database. The recorded data is subsequently displayed on the website's "Database" page for user access and review. This comprehensive process ensures efficient monitoring and data management within the security system.

V. EXPERIMENTAL PROCEDURES

Following the design and development of the program and hardware, the experimental data were collected as follows:

A. Installation of Equipment

The designed system, consisting of either a laptop or a Raspberry Pi setup, was installed, and the camera was positioned appropriately for the intended location. The camera was installed at a height not exceeding 10 meters, and the ambient light intensity was measured to ensure it ranged between 5500 and 7000 Lux during each trial.

B. Initial Testing with Unregistered Faces

Eleven volunteers participated in the experiment. In the first phase, the volunteers walked past the camera without their facial data being registered in the system. The program's performance was observed, and the results were recorded.

C. Testing with Registered Faces

Volunteers had their photographs taken and uploaded into

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

the system via the website. They then walked past the camera again, maintaining a distance of 1–2.5 meters. The system's performance in recognizing faces was evaluated, and the results were documented.

Fig. 3 Workflow of the AI-Based Security Camera System

D. *Performance Evaluation*

The system's efficiency was assessed by calculating key performance metrics, including Accuracy, Recall, and mean Average Precision (mAP) at 50 - 95% Intersection over Union (IoU). These metrics provided a quantitative evaluation of the AI-based security camera system's effectiveness.

VI. EXPERIMENTAL RESULTS

The experiment was designed with two approaches: the first utilized a notebook computer for processing the program with the Human Detection function, while the second employed a Raspberry Pi board to process the program without this function.

A. *Experimental Results Using a Laptop Computer*

The processing was conducted on an ASUS EXPERTBOOK P2451FA notebook paired with a Full HD 1080p DI01 webcam. The notebook specifications are as follows:

- CPU: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz
- Operating System: 64-bit Windows 11 Home Single Language
- RAM: 12.0 GB (11.8 GB usable)

TABLE I. Confusion Matrix based on "Unknown" and "Known" events.

	Actual = Known	**Actual = Unknown**
Predicted = Known	TP = 33	FP = 3
Predicted = Unknown	FN = 0	TN = 30

Model Performance Metrics

- Accuracy: 0.9536
- Recall: 1.0000
- mAP: 0.9318

The program successfully transmitted data to the website without issues. The model accurately identified individuals without registered facial data as "Unknown" when they appeared within 1–2.5 meters of the camera. However, in cases where unregistered individuals had facial features resembling registered users with over 50% confidence, the system occasionally took longer to process and sometimes made incorrect predictions.

For example, in cases reflected in Tables 1, 3, and 9, the system incorrectly identified unregistered individuals as Participant 1 (Mr. A). Upon review, Participants 1, 2, and 5 shared similar round facial shapes, while Participants 1 and 2 wore glasses, and Participants 1 and 5 had the same hairstyle. This limitation likely stems from insufficient training data or high facial similarity in terms of inter-eye distance and facial components.

TABLE II. Model Performance Results Based on the Frequency of "Unknown" and "Participant n" Events

Participant	Accuracy	Recall	mAP
1	0.83	1.00	0.75
2	0.83	1.00	0.75
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	0.83	1.00	0.75
6	1.00	1.00	1.00
7	1.00	1.00	1.00
8	1.00	1.00	1.00
9	1.00	1.00	1.00
10	1.00	1.00	1.00
11	1.00	1.00	1.00
Average	0.9536	1.00	0.9318
S.D.	0.0794	0.0000	0.1168

To address this issue, the number and variety of facial images for each user should be increased to improve the program's ability to distinguish between similar faces. For registered individuals, the program consistently recognized their faces and reported their identities along with a confidence percentage when they appeared within 1–2.5

Chitapong Wechtaisong, Sunanta Radomboon, Pratiparn Chawala, and Kanokporn Boonjubut, Thanaphon Sakonphattharasorn, Anawin Pechbooranin, "AI Based Home Security Camera System," *International Research Journal of Advanced Engineering and Science*, Volume 10, Issue 1, pp. 12-17, 2025.

meters of the camera.

B. Experimental Results Using Raspberry Pi Board

The system utilized a Raspberry Pi 4 Model B board paired with a Full HD 1080p DI01 webcam for data processing. The specifications of the Raspberry Pi are as follows:

- Model: Raspberry Pi 4 Model B
- Processor: 64-bit Quad-core Cortex-A72
- RAM: 8GB LPDDR4
- CPU Cores: 4

TABLE III. Confusion Matrix based on "Unknown" and "Known" events.

	Actual = Known	Actual = Unknown
Predicted = Known	TP = 33	FP = 7
Predicted = Unknown	FN = 0	TN = 26

Model Performance Metrics

- Accuracy: 0.8809
- Recall: 1.0000
- mAP: 0.8590

The program successfully transmitted data to the website and operated faster than the previous version after functional upgrades.

The model demonstrated high accuracy in identifying unregistered faces as "Unknown" when they appeared within 1–2.5 meters of the camera. However, if an unregistered face resembled a registered one by more than 50% confidence, the processing time increased, and occasional misclassification occurred. For example, in cases reflected in Tables 32, 40, 42, and 44, the system misclassified unregistered individuals as Participants 1, 2, and 5 (referred to as Mr. A, Ms. B, and Mr. E).

This limitation can be attributed to insufficient facial data uploaded for training or high similarity in facial features, such as inter-eye distance, facial structure, makeup, or hairstyle. Addressing this issue requires increasing the diversity and volume of training data for each individual to improve the system's discrimination ability.

In cases where individuals with registered data approached the camera within 1–2.5 meters, the program consistently recognized their faces and correctly identified them with confidence values.

TABLE IV. Model Performance Results Based on the Frequency of "Unknown" and "Participant n" Events

Participant	Accuracy	Recall	mAP
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	0.83	1.00	0.75
6	0.83	1.00	0.75
7	1.00	1.00	1.00
8	1.00	1.00	1.00
9	0.60	1.00	0.60
10	0.60	1.00	0.60
11	0.83	1.00	0.75
Average	0.8809	1.0000	0.8590
S.D.	0.1583	0.0000	0.1700

VII. CONCLUSION

Based on the testing and analysis of the AI-based security camera system, the program effectively detects humans entering the monitored area. The facial recognition program successfully identifies human faces and determines whether they belong to known individuals. If the system detects an individual whose facial data is not in the database, it captures the face image and sends an intrusion alert, along with the captured image, to LINE Notify on the LINE application.

In an experiment involving 11 volunteers, the system accurately identified the individuals during each trial. Across 132 tests, the system achieved an average accuracy of 91.72%, an average recall of 100%, and a mean Average Precision of 89.54%.

When processing data on both a Notebook and a Raspberry Pi 4, the system correctly identified True Positives and True Negatives in 122 instances, accounting for 92.42%. However, minor errors were observed, with 10 False Positives (7.58%), where the system misidentified unknown faces as known individuals. Despite these minor inaccuracies, the experimental results demonstrate that the AI-based security camera system exhibits high accuracy and precision.

ACKNOWLEDGMENT

This research article on an artificial intelligence-based home security camera system has been successfully completed thanks to the generous support of colleagues and seniors in the field of Telecommunications Engineering. We extend our heartfelt gratitude to all volunteers who participated in providing facial data for testing the security camera system. Special thanks are also due to the Instrument Center of Suranaree University of Technology for their invaluable support in providing the equipment necessary for the development and testing of the security camera system, enabling its successful completion.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, 2016. doi: 10.1109/CVPR.2016.91.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, 2010. doi: 10.1109/TPAMI.2009.167.

[3] Y. Fan, Y. Luo, and X. Chen, "Research on face recognition technology based on improved YOLO deep convolution neural network," Journal of Physics: Conference Series, vol. 1982, no. 1, p. 012010, 2021. doi: 10.1088/1742-6596/1982/1/012010.

[4] L. Z. Chun, L. Dian, J. Y. Zhi, W. Jing, and C. Zhang, "Face detection in complex environments," International Journal of Computational Intelligence Systems, vol. 13, no. 1, pp. 1153–1163, 2020. doi: 10.2991/ijcis.d.200805.002.

[5] S. M. Menon, A. George, A. N., and J. James, "Custom face recognition using YOLO.V3," in Proceedings of the International Conference on Signal Processing and Communication (ICSPC), pp. 1–5, 2021. doi: 10.1109/ICSPC51351.2021.9451684.

[6] A. J. Mantau, I. W. Widayat, J.-S. Leu, and M. Köppen, "A human-detection method based on YOLOv5 and transfer learning using thermal image data from UAV perspective for surveillance system," Drones, vol. 6, no. 10, p. 290, 2022. doi: 10.3390/drones6100290.

[7] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, “Face recognition systems: A survey,” Sensors (Basel, Switzerland), vol. 20, no. 2, p. 342, 2020. doi: 10.3390/s20020342.

[8] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” Journal of Big Data, vol. 8, no. 1, pp. 1–53, Mar. 2021. [Online]. Available: https://journalofbigdata. springeropen.com /articles/10.1186/s40537-021-00444-8

[9] I. Alqatawneh, R. Deng, K. Rabeyee, Z. Chao, F. Gu, A. D. Ball, “A Developed Convolutional Neural Network Architecture for Condition Monitoring,” 2021 26th International Conference on Automation and Computing (ICAC), IEEE, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9594171/

[10] S. Cong, Y. Zhou, “A review of convolutional neural network architectures and their optimizations,” Artificial Intelligence Review, vol. 55, no. 2, pp. 1289–1307, Feb. 2022. [Online]. Available: https://link.springer.com/article/10.1007/s10462-022-10213-5

[11] [4] M. D. Zeiler and R. Fergus, “An Introduction to Convolutional Neural Networks,” arXiv preprint arXiv:1511.08458, 2015. [Online]. Available: https://arxiv.org/abs/1511.08458