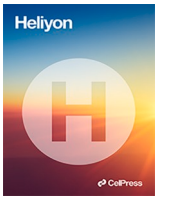
Heliyon 10 (2024) e30486

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

Automated medication verification system (AMVS): System based on edge detection and CNN classification drug on embedded systems

Yen-Jung Chiu

Department of Biomedical Engineering, Ming Chuan University, Taoyuan, 333, Taiwan

ARTICLE INFO

Keywords:
Drug
Image classification
Embedding
Deep learning
Edge detection

ABSTRACT

A novel automated medication verification system (AMVS) aims to address the limitation of manual medication verification among healthcare professionals with a high workload, thereby reducing medication errors in hospitals. Specifically, the manual medication verification process is time-consuming and prone to errors, especially in healthcare settings with high workloads. The proposed system strategy is to streamline and automate this process, enhancing efficiency and reducing medication errors. The system employs deep learning models to swiftly and accurately classify multiple medications within a single image without requiring manual labeling during model construction. It comprises edge detection and classification to verify medication types. Unlike previous studies conducted in open spaces, our study takes place in a closed space to minimize the impact of optical changes on image capture. During the experimental process, the system individually identifies each drug within the image by edge detection method and utilizes a classification model to determine each drug type. Our research has successfully developed a fully automated drug recognition system, achieving an accuracy of over 95 % in identifying drug types and conducting segmentation analyses. Specifically, the system demonstrates an accuracy rate of approximately 96 % for drug sets containing fewer than ten types and 93 % for those with ten types. This verification system builds an image classification model quickly. It holds promising potential in assisting nursing staff during AMVS, thereby reducing the likelihood of medication errors and alleviating the burden on nursing staff.

1. Background

Medication dispensing errors are a common problem, referring to instances where patients may not experience improvements in their condition due to the use of non-prescribed medications, or may suffer adverse reactions, which in severe cases can lead to fatalities [1,2]. To mitigate medication dispensing errors in hospitals, nursing professionals are often required to conduct repeated checks. While this approach indeed reduces errors, it also adds to the workload of nursing staff, particularly in cases where medications have high similarity, making differentiation challenging [3]. Hence, the development of Automated Dispensing Cabinets (ADCs) has been proposed. These cabinets autonomously dispense medications or medical supplies, opening specific compartments to retrieve the correct items, thereby reducing the likelihood of medication errors [4–6]. Jaw-Horng Liou et al. implemented ADCs at the Taichung Veterans General Hospital in Taiwan, where pharmacists place medications in the cabinets upon receiving physician medication

E-mail address: yjchiu@mail.mcu.edu.tw.

https://doi.org/10.1016/j.heliyon.2024.e30486
Received 24 April 2024; Accepted 28 April 2024
Available online 3 May 2024

orders. During the medication delivery process, nurses manually verify various aspects such as meal-specific medication variations, correctness, and dosages [5]. To further minimize errors in medication dispensing and retrieval, existing ADCs systems typically utilize barcodes and Radio-Frequency Identification (RFID) to identify the location of medications. However, since not all medication packaging contains barcodes or RFID, Hsien-Wei Ting et al. developed ADCs integrated with image recognition to verify the placement of packages [6]. Nursing staff maintained optimism about the implementation of ADCs, reducing medication errors [5]. Despite the reduced workload for nursing staff and the decrease in medication dispensing errors facilitated by ADCs, nurses still face considerable pressure during medication delivery processes, as they need to confirm the usage and dosage of each medication, which is about 6 min per patient [5,7]. Moreover, the nurse double-checked, but the medication administration error still occurred [8]. Hence, the aim is to build a mobile system to assist nurses in double-checking doses after pharmacists check by promoting the development of medication verification algorithms and devices called automated medication verification systems (AMVS).

Medication verification can be categorized into recognition of single medications and recognition of multiple medications. In deep learning models, recognition of single medications can be achieved using classification models or object detection techniques [9–11], whereas segmentation methods are employed to build models for cases where multiple medications are present in a single image. Segmentation and object detection typically require manual annotations, with model construction involving original images and mask images. Object detection predicts the category of medications in a single image, while segmentation delineates the edges of each category and predicts their medication types. Models such as YOLO and Mask R–CNN have been applied to recognize multiple medication types in single images [6,12]. In the context of deep learning and the YOLO model, segmentation typically refers to the task of object segmentation in computer vision. YOLO is a real-time object detection system that divides images into a grid and predicts bounding boxes and class probabilities for objects within each grid cell. In contrast, Mask R–CNN has a multi-stage architecture comprising region proposals, bounding box refinement, and mask prediction stages, differing from YOLO's single-stage detection approach. Medication image recognition can be divided into two types: pill recognition and medication packaging recognition. For ADCs, to prevent errors in recognizing look-alike blister packages, it is crucial to verify whether the packaging is correct. Hsien-Wei Ting et al. analyzed packaging for 250 types of medications, comprising a total of 18,000 images of front and back views, and employed YOLO-based models for packaging classification, with training times ranging from 5 to 7 h and F1 accuracies exceeding 90 % [6]. Lu Tan et al. also utilized YOLO-based models for pill classification, albeit with a somewhat lower accuracy rate, achieving around 75 % in F1 score [13]. Recognizing the challenge of manually annotating medication regions, Hyuk-Ju Kwon et al. partitioned their model into two segments: one dedicated to detecting medication regions and the other focused on predicting medication types. They employed a Mask R–CNN model for medication region detection, efficiently identifying all pills within an image and automatically generating distinct image regions for each pill. Subsequently, the second segment of the model was responsible for predicting the types of medications associated with each identified region. Instead of attempting to predict all categories in an image simultaneously, this approach stabilizes the increase in prediction accuracy. However, despite these advancements, the highest accuracy achieved was approximately 90 % when dealing with images containing 5–7 pills each [12].

Segmentation models like YOLO and Mask R–CNN lack interpretability due to their complex processes. In contrast, image classification provides clearer insights into predictions, allowing users to understand and interpret model outputs effectively [14]. Pre-trained classification models such as ResNet [15], offer the advantage of fine-tuning and optimizing their performance for specific tasks without the need to train from scratch [16]. They have been applied across various fields, including medical images [17], natural language processing [18], and genomic sequences [19], demonstrating their versatility and effectiveness.

In most research involving trained models, GPU acceleration is commonly used to expedite model training due to the lengthy training times. Additionally, segmentation methods require a substantial amount of annotated medication data, which consumes significant human resources. Moreover, re-annotation is necessary when adding new medications, further increasing costs. In summary, previous methods have several drawbacks: 1) while utilizing computers for GPU training is feasible, it costs high; 2) segmentation methods entail significant human resource costs; 3) despite hardware advances, integrating GPU hardware into portable systems remains costly.

The AMVS proposed in this study can rapidly and accurately predict multiple medications in a single image. The model is divided into two stages: the first stage establishes a medication classification model using segments of medication regions, and the second predicts the types of medications in each region, eliminating the need for manual annotations for predicting medication types and regions. Furthermore, in previous studies, image capture, model training, prediction, and system operation were all conducted in open spaces, whereas this study is conducted in a closed space, reducing the impact of optical changes on image capture. If integrated with Hospital Information Systems (HIS) in the future, it holds promising clinical applications. This study proposes a design pattern encompassing hardware design, analysis workflow design, and Internet of Things (IoT) design, effectively reducing costs and human resource consumption.

2. Methods

2.1. Dataset collection

All images in this study were captured using the "Drug Verification Box." For both the validation and training datasets of the edge detection method, each image consists of a single drug, with 30 images taken for each drug, resulting in a total of 300 images across 10 drug categories. In the design of the inference data, each photo will contain multiple drugs, and 50 images will be randomly captured.

2.2. Find contours

The contours function in OpenCV is commonly employed to accurately delineate object boundaries within images. Preceding its application, techniques such as thresholding or the Canny edge detection method are often utilized to enhance the precision of edge detection. The thresholding method facilitates the conversion of grayscale imagery into binary representations, offering a selection of distinct types, including binary, binary_inv, trunc, tozero, tozero_inv, mask, otsu, and triangle [13,20]. Notably, variants suffixed with _inv effectuate inversion after image evaluation, mapping pixel values exceeding the predefined threshold to 0 and the remainder to 255, thus delineating black and white in the grayscale spectrum. Furthermore, the Canny method [21] and the watershed algorithm [22] were considered in the method choice. Process time and accuracy were referred to when deciding on edge detection in this system.

2.3. Drug verification box

The system design for this study incorporates a sealed and light-tight medication box, housing a Raspi 4B with a 4 GB CPU, equipped with a 64 GB memory card, a camera module, and two white LED lights. The medication tray, produced through 3D printing, has dimensions of 100x100 mm. The distance between the camera and the medication tray is set at 70 mm, and the captured image size is 500x500 pixels. In the analysis, images will be resized to 224x224 based on the pre-trained model's training dataset.

To mitigate the impact of external environmental factors on the experiment, all hardware components are enclosed within this medication box. Due to the fully enclosed and dark nature of the medication box, the use of LED light sources ensures stable and consistent lighting, contributing to a reduction in variations caused by other light sources on pixel values. The specified distance between the camera and the medication tray ensures high resolution, capturing images that exclusively include the medication tray. To prevent contact between medications and the medication tray, thus reducing segmentation errors, the outer walls of the medication tray feature an angled design. During practical use, the medication box's closed environment is leveraged with an electromagnetic valve to enhance the convenience of retrieving and placing the medication tray (Fig. 1). In terms of model testing, we employed three devices: Raspi, Intel(R) Xeon(R) Silver 4314 CPU @ 2.40 GHz, and NVIDIA RTX A5000.

2.4. Software design

In the control and algorithm development for the Raspi, we utilized a suite of Python packages. To manage the Raspi's camera module and LED, we employed the gpiozero, signal, and RPi.GPIO packages. This allowed us to control the LED's on/off state, brightness, and frequency, while also adjusting the camera's exposure time and image capture size.

During the image processing phase, we applied the OpenCV package to read images and perform edge detection analysis. This step assisted in extracting crucial edge information from pharmaceutical images, providing features for subsequent classification and segmentation.

For the construction of the deep learning model, we opted for the PyTorch package to design and implement the model. This encompassed the establishment of the model, training, and evaluating its performance in effectively classifying images.

2.5. Pre-trained models

The architecture of the model is based on established models successfully applied in large-scale image classification tasks, employing a fine-tuned strategy to construct a multi-drug classification model, including ResNet18, ResNet50 [15], VGG11 [23],

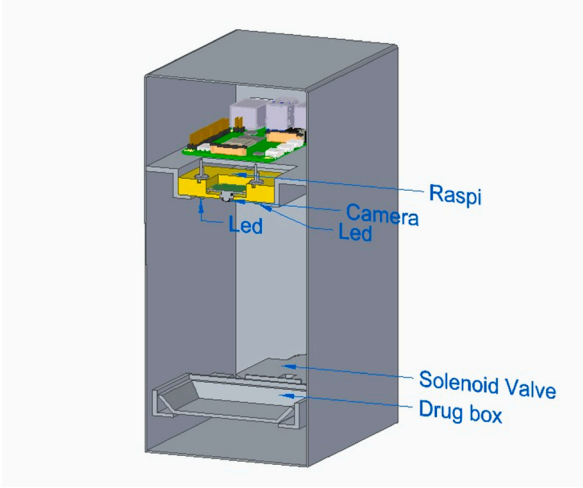


Fig. 1. Design of the identification system.

AlexNet [24], GoogLeNet [25], SqueezeNet [26], ShuffleNet [27], and MobileNetV3_Small [28]. During model training, we implemented early stopping 20 times based on the values of the mean of cross entropy and classification accuracy to determine when to halt the training process.

2.5.1. Statistic method

In the training process, g represents the number of samples in each minibatch, t_m^c takes values of 0 or 1, indicating whether the sample belongs to class c, p_m^c represents the probability of each sample being predicted as class c, and TP_m indicates the number of the correct classification. g represents the number of images.

For results presentation, we utilized a confusion matrix. In model inference, we employed SA average accuracy, where SA_k denotes the accuracy of each image, N represents the number of images. Single-image accuracy assesses the multi-label prediction accuracy of a single image, with n indicating the size of multi-labels, i representing the number of drugs in a single image, and TP_l indicating the number of correct category predictions.

$$Mean\ of\ cross\ entropy = -\frac{1}{g}\sum_{m=1}^{g} t_m^c \times \log\left(p_m^c\right)$$

$$Classification\ accuracy = \sum_{m=1}^{g} \frac{TP_m}{g}$$

$$SA\ average\ accuracy = \frac{\sum_{k=1}^{N} SA_k}{N}$$

$$Single-image\ accuracy\ (SA) = \sum_{l=1}^{i} \frac{TP_l}{n}$$

Furthermore, the results include the calculation of sensitivity and specificity metrics. Sensitivity represents the ratio of accurately classified instances of a specific class to all instances of that class. Specificity measures the ratio of accurately classified instances not belonging to a specific class among all instances in the dataset that do not belong to that class.

3. Study design

In this study, all photographs were captured using a single device, which contained various pharmaceuticals and health

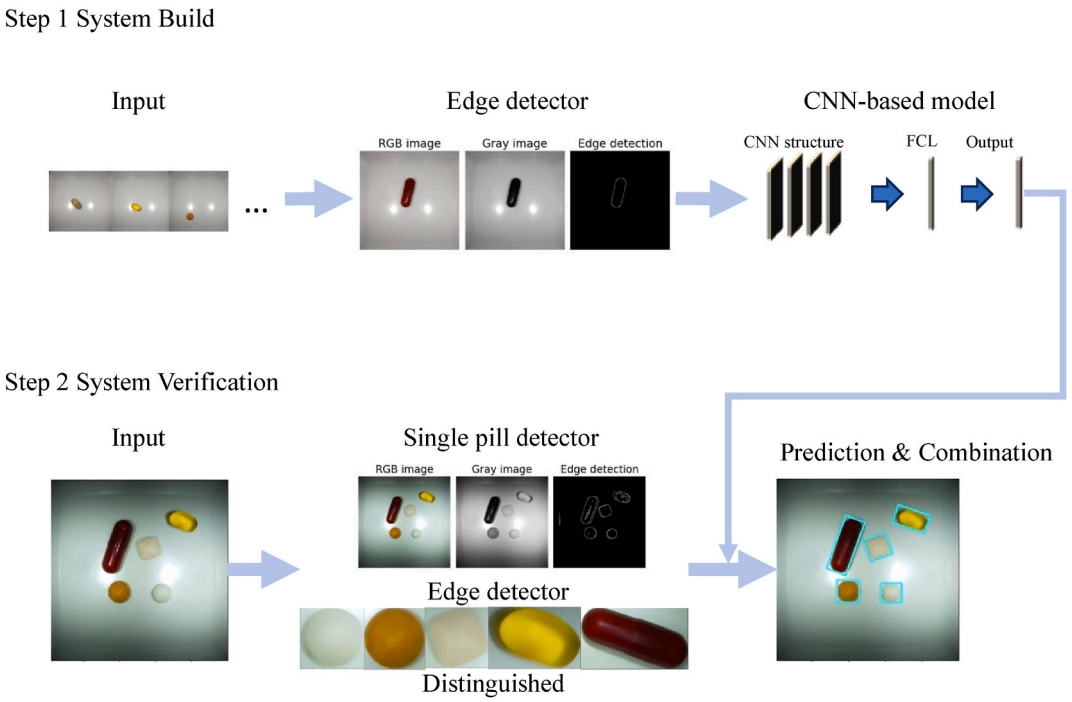


Fig. 2. Workflow contains the process of model train and verification. For step 1, the CNN-based model was trained on a single pill image. That area was only a pill with no background. For step 2, the system used the same detector algorithm to find each pill and predict the pill type.

supplements. The method involves applying a multi-class classification model and edge detection methods for the automated training and inference analysis of the Automated Medication Verification System (AMVS). In Step 1, the workflow for model development is described. The training process involves using images of single pills. Initially, the edge detection method is applied to extract medication positions and form the training dataset. Subsequently, a CNN-based model, utilizing pre-trained techniques, is trained for classification (Fig. 2, step 1).

During the verification phase, each image encompasses multiple medications. The edge detection method is employed to identify regions containing candidate pill regions. These regions are then padded, resized, and subjected to classification by the trained model to predict the drug type. The outcomes are aggregated to depict the classification of medications within each region, thus completing the medication segmentation analysis (Fig. 2, step 2).

4. Results

4.1. Compared the performance of edge detection

The results demonstrate that all methods, except for Canny, effectively isolate the drug regions while excluding the background (Fig. 3 A). Canny achieves accurate selections in roughly 60 % of the images but may encounter errors in identifying regions for different drug types (Fig. 3 B). Furthermore, each image requires only 0.01 s for processing. Comparative t-tests reveal that the watershed algorithm incurs lower time costs compared to other methods, showcasing the lowest time consumption (Fig. 4). Additionally, when assessing the time costs associated with the watershed algorithm, each method analyzed 300 images, conducted 10 times on a Raspi, with an average completion time of approximately 0.011 s (Fig. 5 A, Supplementary Table 1). Moreover, as the dataset size increases incrementally, with the addition of one type at a time (30 images), the time and quantity exhibit a linear relationship (Fig. 5 B, R^2 is 0.884).

4.2. The model structure and training image preprocessing

Following the final layer of the eight models, a fully connected layer known as a dense layer is added for linear transformation. The input size of this dense layer is set to 512 for ResNet18, 2048 for ResNet50, and 1000 for other models. The output size is uniformly set to 11, representing the categories of drugs and background (Fig. 6, Supplementary Fig. 1). Moreover, to account for the potential inclusion of background images in segmentation methods (Supplementary Fig. 2), we introduced a background image category for prediction (Fig. 7). Two datasets with different preprocessing processes were used for model training: one utilized original image data (Fig. 7 A), while the other employed edge detection methods on drug images and expanded the image pixels to 500 × 500 using padding techniques (Fig. 7 B). To reduce training and inference costs, images in the classification stage were resized to 224x224. During model validation, 20 % of images were randomly selected from all images for validation, resulting in a validation accuracy of 100 % (Supplementary Table 2).

4.3. Multi-drug verification results: model accuracy comparison

Firstly, we utilized VGG11 to demonstrate the effect of implementing Edge Detection Processing (EDP) with and without the padding method (Fig. 7). Remarkably, when solely employing EDP, two pills were devoid of predicted labels (Supplementary Fig. 2 A and C). In contrast, employing EDP with the padding method led to successful pill predictions (Supplementary Fig. 2 B and D). Specifically, when using the original images as training data, pills numbered 2, 6, 9, and 10 could not be accurately predicted.

Next, among the eight classification models utilizing EDP, we assessed their performance on images containing 10 different drugs and images with 3–5 drugs, categorized as “All,” “Less than 10 pills,” and “10 pills,” respectively. The “All” results were derived from the combined outcomes of “Less than 10 pills” and “10 pills.” VGG11 demonstrated the highest SA average accuracy of approximately 93 % across all images, effectively discerning between images with and without ten pills. Specifically, the VGG11 model achieved an

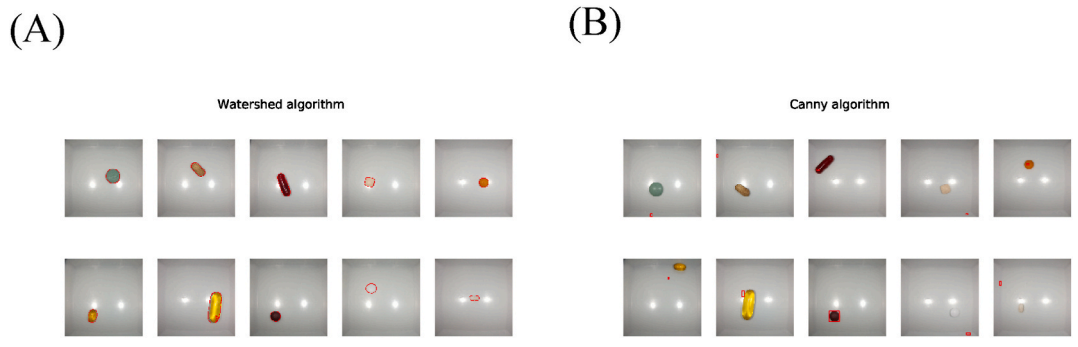


Fig. 3. The regions and time taken for different edge detection methods were analyzed. (A) Represents the results obtained using the watershed algorithm. (B) Depicts the outcomes from the canny algorithm.

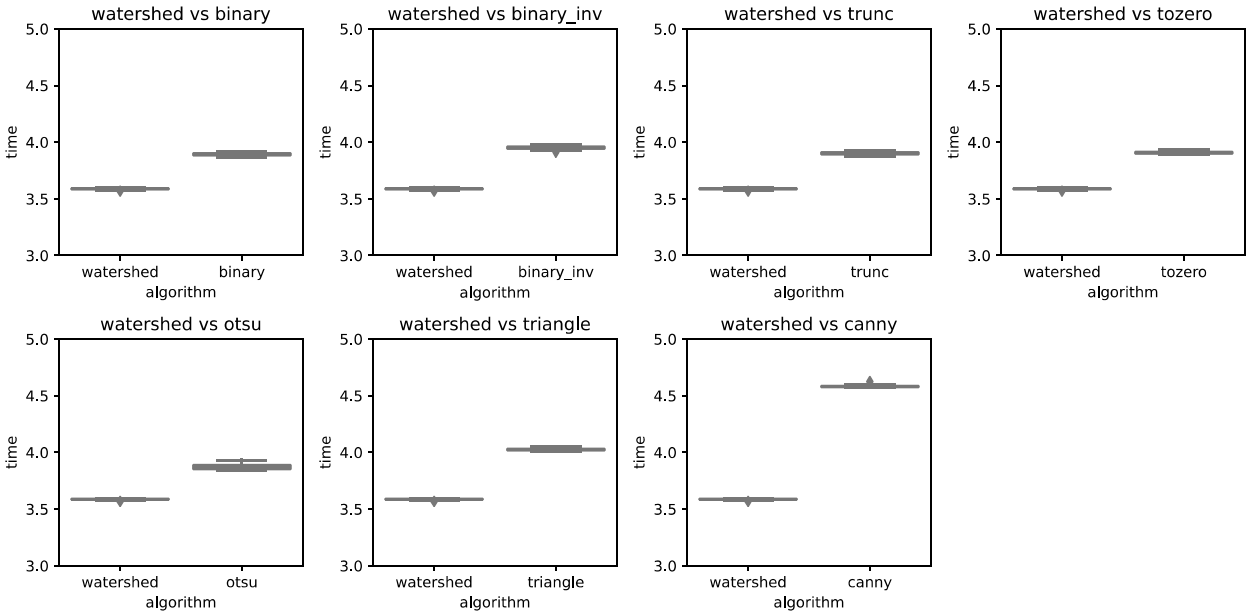


Fig. 4. The computational time of edge detection was compared between the watershed algorithm and other methods.

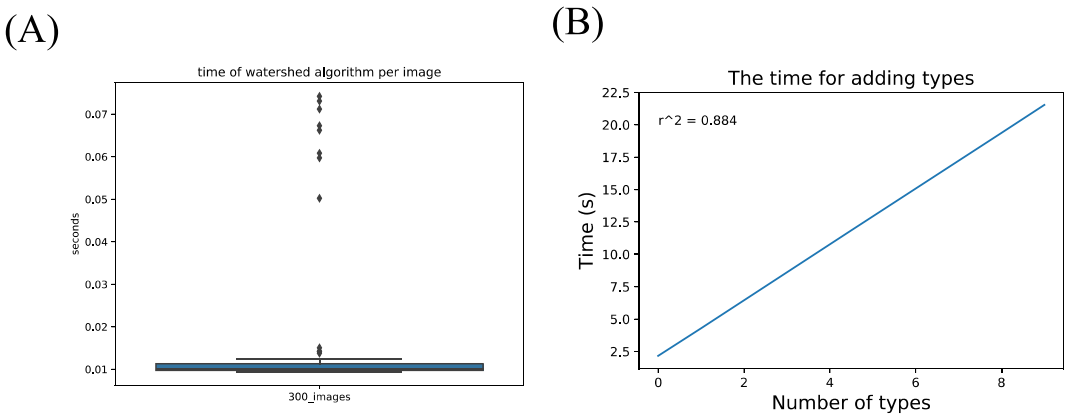


Fig. 5. (A) The boxplot shows the average time of each image taken over 10 trials. (B) The line plot illustrates the correlation between the processing time and the number of types added, as indicated by the line plot.

accuracy of approximately 96 % for images without ten pills. In comparison, the AlexNet model achieved an accuracy of around 97 % for images without ten pills, while VGG11 achieved approximately 96 % (Table 1). The images encompassed randomized drug types and positions, totaling 50 images to assess research process efficiency, including 15 images with 10 pills. Among these, 35 images were predicted with 100 % accuracy, while the remaining 15 contained one incorrectly identified pill. SA average accuracy for other models ranged from 60 % to 80 %. Additionally, to evaluate model efficacy in analyzing images with untrained pills, photographs were taken with either one or two unknown pills per image, resulting in a noticeable decrease in accuracy (Supplementary Table 3). This decline occurred because non-existent drugs had the potential to be misidentified as one of the known drug types.

Additionally, sensitivity and specificity were employed to assess the performance of predicting each drug type. The confusion matrix depicted the correspondence between the actual and predicted drug types for each pill from the inference images (Supplementary Table 4). Sensitivity and specificity values were provided for each pill type. The findings revealed that all specificities exceeded 90 %, with four sensitivities at 100 % (Supplementary Table 5).

The MobileNet, SqueezeNet, and ShuffleNet were categorized as lightweight models. Despite this classification, their performance did not exceed that of VGG11. However, among these lightweight models, the SqueezeNet model demonstrated superior performance. Notably, when the number of pills was less than 10, the SA average accuracy reached 91 %, surpassing both ResNet18 and ResNet50 in all scenarios involving any number of pills.

Time in training VGG11-based model from different devices.

The results demonstrate that VGG11 achieves superior accuracy compared to other methods. To evaluate the model's analysis time

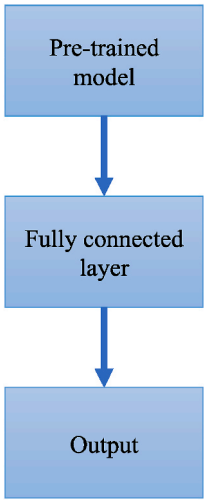


Fig. 6. The architecture utilized pre-trained models to identify drugs.

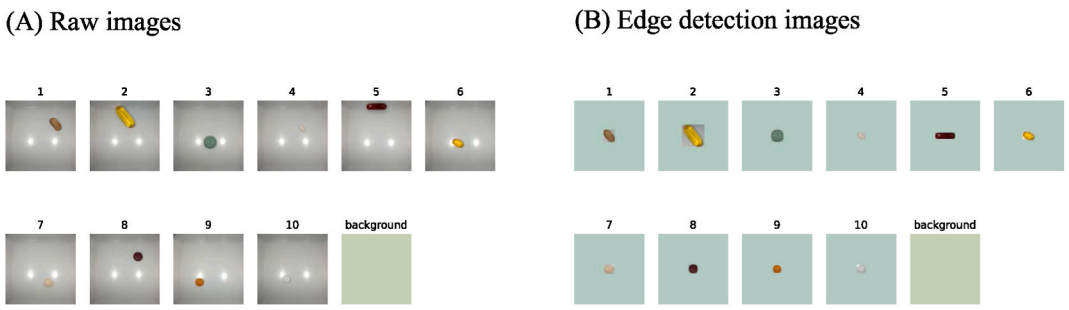


Fig. 7. The images and their corresponding labels were used for training the model.

Table 1
The SA average accuracy for classification with watershed algorithm in three epochs.

Model		Batch Size		
		4	16	32
resnet18	All	80 %	64 %	63 %
	Less than 10 pills	85 %	69 %	68 %
	10 pills	70 %	57 %	52 %
resnet50	All	79 %	70 %	78 %
	Less than 10 pills	85 %	75 %	84 %
	10 pills	65 %	57 %	59 %
vgg11	All	95 %	92 %	91 %
	Less than 10 pills	96 %	96 %	95 %
	10 pills	93 %	83 %	84 %
alexnet	All	91 %	92 %	92 %
	Less than 10 pills	96 %	97 %	97 %
	10 pills	77 %	82 %	80 %
mobilenet_v3_small	All	64 %	64 %	74 %
	Less than 10 pills	71 %	71 %	86 %
	10 pills	46 %	46 %	46 %
googlenet	All	73 %	72 %	76 %
	Less than 10 pills	76 %	77 %	82 %
	10 pills	66 %	60 %	64 %
squeezenet	All	76 %	87 %	87 %
	Less than 10 pills	80 %	91 %	90 %
	10 pills	68 %	78 %	80 %
shufflenet	All	70 %	66 %	64 %
	Less than 10 pills	76 %	78 %	74 %
	10 pills	58 %	38 %	40 %

within the system, we conducted a comparison across three devices: GPU, CPU, and Raspi. Our findings reveal that GPU delivers the highest performance, while Raspi exhibits the poorest performance. During a single training session comprising 16 epochs, GPU completes the training in 3 s, CPU in 5 s, and Raspi in 1033 s (Supplementary Fig. 3 A). Regarding image analysis, both GPU and CPU can process each image within 1 s. However, when utilizing Raspi, processing an image with 5 pills takes approximately 15 s, and with 10 pills, it takes about 30 s (Supplementary Fig. 3 B).

5. Discussion

To alleviate the burden on healthcare professionals, we developed a fully automated drug recognition system leveraging neural network models, achieving over 93 % accuracy in identifying drug types and conducting segmentation analyses. Unlike models utilizing YOLO, our approach reduces the cost of manual annotation and computational time while enabling the addition of drug types at any time. Our system incorporates the edge detection method and utilizes the pre-trained VGG11 model for drug recognition. While our method combines traditional edge detection with a CNN approach, it represents a significant advancement, achieving 95 % accuracy, surpassing the typical 90 % accuracy of YOLO models [6,12]. This breakthrough eliminates the need for resource-intensive YOLO models while enabling accurate predictions for multiple classes within a single image.

In applying deep learning models, we employed eight common image model architectures. The VGG11 model emerged as the optimal choice, demonstrating superior predictive performance across images containing 3–10 pills. While AlexNet achieved an overall accuracy of up to 92 %, its accuracy dropped to 80 % when confronted with images containing 10 pills, compared to VGG11's 86 % accuracy in such scenarios. Additionally, the model was trained using 30 images per type. The performance across different image sizes revealed that the accuracy did not improve beyond 30 images per type (Supplementary Fig. 4).

In the VGG11 model, training with a CPU device took approximately 18 min, while training with a GPU took only 14 min. However, training on a Raspi device required about 3 days. Although training time with the MobileNet model on a Raspi could be reduced to 3 h, it came at the cost of accuracy. To overcome this limitation, we propose integrating IoT technology by combining computers with Raspi devices. This approach entails sending images to the computer for model training and then utilizing the Raspi for developing a portable pill recognition system. By adopting this hybrid method, we aim to achieve both efficiency and accuracy in drug recognition while ensuring portability. Moreover, despite their reputation for efficiency on mobile devices, these lightweight models did not outperform the VGG11 in our study. While they may offer advantages in computational resources and memory usage, their accuracy in identifying multiple drug classes was inferior to VGG11. This suggests that, for our specific task of medication classification, the deeper architecture of VGG11 yielded superior results.

Among the eight tested edge detection methods, only canny failed to accurately select the drug area, while the rest yielded favorable results. However, the watershed algorithm notably consumed less time compared to other methods. Therefore, for model inference, we exclusively utilized the watershed algorithm for analysis.

In our system, if a drug present in the image is not recognized, it may be misidentified as another drug type. This discrepancy could be attributed to the model calculating the ratio of drugs to the background during model construction (Fig. 7 A). Through edge detection analysis, most background information is removed, leading to the loss of drug-to-background ratio information, and resulting in prediction accuracy errors. To preserve the ratio between image background and drugs, we applied the EDP method to training and predicting images (Fig. 7 B), demonstrating an improvement in identifying similar drugs and predicting background images, thereby enhancing prediction accuracy. Moreover, when testing images containing unrecognizable drugs, the model may misclassify the drugs as any type in the database, leading to erroneous recognition (Supplementary Table 2).

Compared to the Coral Dev Board or Nvidia Jetson Nano, the Raspberry Pi offers a more cost-effective solution. However, the larger size of the Nvidia Jetson Nano must be considered in medicine cabinet design. Regarding model development, the Raspberry Pi provides versatility by supporting various model architectures such as TensorFlow and PyTorch, while the Coral Dev Board primarily supports standard models like EfficientNet, Inception, MobileNet, and ResNet-50. Additionally, the Raspberry Pi boasts the lowest power consumption [29,30]. Furthermore, deploying TensorFlow-developed models to TPU edge requires additional conversion steps. Research also suggests that there is no significant difference in inference time for image analysis without GPU usage. Hence, it's more practical to develop models using PyTorch, train them on a computer, and then deploy them to the Raspberry Pi for image capture, inference, and IoT integration. We propose that hospital information staff use the medicine box to capture images of individual medications. These photos would then be transmitted via the IoT system to a computer equipped with a GPU for training purposes. Once training is complete, the model can be transferred back to the Raspberry Pi, allowing medical staff to conduct medication double-checks using the medicine box. The compact design of this medicine box also lends itself well to integration into a mobile medical workstation.

To prevent medication errors, nursing staff are mandated to double-check the appearance, dosage, and patient information with another trained professional during inpatient drug distribution. With the introduction of our medication box system, nursing staff can conveniently place medications into the box, scan the patient's wristband barcode to verify medication details, and ensure accuracy through image analysis. Moreover, the information system enables staff to compare actual medications with photographed ones via the screen interface, alleviating the need to memorize various medicines. Therefore, the Automated Medication Verification System (AMVS) is essential.

6. Conclusion

This study presents a framework for easily constructing a multiple-pill classification model accessible to all. Our system swiftly and

accurately identifies and verifies manually retrieved medications, thereby reducing manual verification time and potential errors, consequently enhancing overall medication delivery efficiency. While most images achieve a 100 % accuracy rate in identifying multiple drug classes, variations in accuracy are observed in drug segmentation, particularly when drug edges are in proximity, potentially due to object adjacency. Future research endeavors should concentrate on addressing this issue. Despite not achieving a 100 % recognition rate, our system remains advantageous. We have optimized it to maximize recognition rates, with caregivers performing secondary verification to ensure accuracy. Moreover, our system identifies errors, facilitating prompt correction and mitigating potential risks. Comprehensive assessment of the system's effectiveness through further clinical trials is essential.

Data availability

The code is accessible at https://github.com/holiday01/Automated-Medication-Verification-System.

CRediT authorship contribution statement

Yen-Jung Chiu: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The author confirms no competing financial interests or personal relationships in this work.

Acknowledgements

This work was supported by the grants from the National Science and Technology Council (NSTC), Taiwan (NSTC 112-2222-E-130-003). The funding bodies had no role in the design of the study, collection, analysis, interpretation of data, or the writing of the manuscript.

Abbreviations

ADCs	Automated Dispensing Cabinets
RFID	Radio-Frequency Identification
AMVS	Automated Medication Verification Systems
HIS	Hospital Information Systems
IoT	Internet of Things
EDP	Edge Detection Processing
Raspi	Raspberry Pi

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.heliyon.2024.e30486.

References

[1] R.A. Tariq, R. Vashisht, A. Sinha, Y. Scherbak, Medication dispensing errors and prevention, in: *StatPearls*. Treasure Island (FL), 2023.
[2] G. Reiner, S.L. Pierce, J. Flynn, Wrong drug and wrong dose dispensing errors identified in pharmacist professional liability claims, J. Am. Pharmaceut. Assoc. 60 (5) (2003) e50–e56, https://doi.org/10.1016/j.japh.2020.02.027. Sep-Oct 2020.
[3] M. Bengtsson, A.I. Ekedahl, K. Sjostrom, Errors linked to medication management in nursing homes: an interview study, BMC Nurs. 20 (1) (Apr 29 2021) 69, https://doi.org/10.1186/s12912-021-00587-2.
[4] C.A. Lester, J. Li, Y. Ding, B. Rowell, J. Yang, R.A. Kontar, Performance evaluation of a prescription medication image classification model: an observational cohort, NPJ Digit Med 4 (1) (Jul 27 2021) 118, https://doi.org/10.1038/s41746-021-00483-8.
[5] J.H. Liou, et al., Effect of an automated dispensing cabinet system on drug distribution effectiveness in a surgical unit, Heliyon 9 (11) (Nov 2023) e21668, https://doi.org/10.1016/j.heliyon.2023.e21668.
[6] H.W. Ting, S.L. Chung, C.F. Chen, H.Y. Chiu, Y.W. Hsieh, A drug identification model developed using deep learning technologies: experience of a medical center in Taiwan, BMC Health Serv. Res. 20 (1) (Apr 15 2020) 312, https://doi.org/10.1186/s12913-020-05166-w.
[7] J.I. Westbrook, et al., Associations between double-checking and medication administration errors: a direct observational study of paediatric inpatients, BMJ Qual. Saf. 30 (4) (Apr 2021) 320–330, https://doi.org/10.1136/bmjqs-2020-011473.
[8] A.K. Koyama, C.S. Maddox, L. Li, T. Bucknall, J.I. Westbrook, Effectiveness of double checking to reduce medication administration errors: a systematic review, BMJ Qual. Saf. 29 (7) (Jul 2020) 595–603, https://doi.org/10.1136/bmjqs-2019-009552.
[9] N. Larios Delgado, et al., Fast and accurate medication identification, NPJ Digit Med 2 (2019) 10, https://doi.org/10.1038/s41746-019-0086-0.
[10] J. Heo, Y. Kang, S. Lee, D.H. Jeong, K.M. Kim, An accurate deep learning-based system for automatic pill identification: model development and validation, J. Med. Internet Res. 25 (Jan 13 2023) e41043, https://doi.org/10.2196/41043.
[11] Y.F. Wong, H.T. Ng, K.Y. Leung, K.Y. Chan, S.Y. Chan, C.C. Loy, Development of fine-grained pill identification algorithm using deep convolutional network, J. Biomed. Inf. 74 (Oct 2017) 130–136, https://doi.org/10.1016/j.jbi.2017.09.005.

[12] H.-J. Kwon, H.-G. Kim, S.-H. Lee, Pill detection model for medicine Inspection based on deep learning, Chemosensors 10 (1) (2022) 4 [Online]. Available: https://www.mdpi.com/2227-9040/10/1/4.
[13] L. Tan, T. Huangfu, L. Wu, W. Chen, Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification, BMC Med. Inf. Decis. Making 21 (1) (Nov 22 2021) 324, https://doi.org/10.1186/s12911-021-01691-8.
[14] E. Pintelas, I.E. Livieris, XSC—an eXplainable image segmentation and classification framework: a case study on skin cancer, Electronics 12 (17) (2023) 3551.
[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[16] F. Bozkurt, Skin lesion classification on dermatoscopic images using effective data augmentation and pre-trained deep learning approach, Multimed. Tool. Appl. 82 (12) (2023) 18985–19003.
[17] H.E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M.E. Maros, T. Ganslandt, Transfer learning for medical image classification: a literature review, BMC Med. Imag. 22 (1) (Apr 13 2022) 69, https://doi.org/10.1186/s12880-022-00793-7.
[18] H. Wang, J. Li, H. Wu, E. Hovy, Y. Sun, Pre-trained language models and their applications, Engineering (2022).
[19] B. Song, Z. Li, X. Lin, J. Wang, T. Wang, X. Fu, Pretraining model for biological sequence data, Brief Funct Genomics 20 (3) (Jun 9 2021) 181–195, https://doi.org/10.1093/bfgp/elab025.
[20] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Pearson/Prentice Hall, 2010.
[21] W. McIlhagga, The canny edge detector revisited, Int. J. Comput. Vis. 91 (3) (2011/02/01 2011) 251–261, https://doi.org/10.1007/s11263-010-0392-0.
[22] H. Liu, et al., Application of an improved watershed algorithm based on distance map reconstruction in bean image segmentation, Heliyon 9 (4) (Apr 2023) e15097, https://doi.org/10.1016/j.heliyon.2023.e15097.
[23] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 *arXiv preprint arXiv:1409.1556*.
[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Presented at the Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada, 2012.
[25] C. Szegedy, et al., Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[26] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size, arXiv preprint arXiv:1602.07360 (2016).
[27] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.
[28] A. Howard, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.
[29] A. Biglari, W. Tang, A review of embedded machine learning based on hardware, application, and sensing scheme, Sensors 23 (4) (Feb 14 2023), https://doi.org/10.3390/s23042131.
[30] A. Garcia-Perez, R. Minon, A.I. Torre-Bastida, E. Zulueta-Guerrero, Analysing edge computing devices for the deployment of embedded AI, Sensors 23 (23) (Nov 29 2023), https://doi.org/10.3390/s23239495.