Sensors and Materials, Vol. 33, No. 9 (2021) 3037–3052
MYU Tokyo

S & M 2672

Drug Verification System Based on Deep Learning Multiscale Rotating Rectangle Detector and Feature Embedding

Shih-Pang Tseng,[1] Che-Wen Chen,[2*] Wei-Yan Jang,[2] and Jhing-Fa Wang[2]

[1]School of Software, Changzhou College of Information Technology, Changzhou 213164, China
[2]Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan

(Received December 30, 2020; accepted August 13, 2021)

Keywords: drug detection, deep residual network, feature pyramid network, image feature embedded network, medication safety

This research is aimed at the development of automatic drug image verification functions. Our verification system is composed of two stages. The first stage is an arbitrarily axis-aligned object detector, which is mainly based on a deep residual network and feature pyramid network (FPN). The detector predicts the rotation bounding boxes for drugs using multiscale feature maps generated by the FPN. Then, the rotation bounding boxes are axially aligned, and the drug image is cropped according to the axis-aligned bounding box. The second stage is a feature matcher, which is based on a feature embedding network. The embedding feature extracted by the feature embedding network is combined with the geometric feature obtained by the arbitrarily axis-aligned object detector to determine whether the drug in the image belongs to the category specified by the user. The database used in this research is an image database created by imaging drugs provided by domestic local medical centers. Our verification system achieved a false positive rate (FPR) of 0.047% in verification tasks of drugs of 21 categories.

1. Introduction

The Institute of Medicine has reported that medication errors are the single most common type of error in health care, representing 19% of all adverse events and accounting for over 7000 deaths annually.(1) Intensive care units (ICUs) in hospitals face great challenges resulting from medication errors,(2) which account for 78% of serious medical errors.(3) Even minor medication errors can cause severe disability. Moreover, the patients in ICUs are relatively weak, so they cannot tolerate medication errors. The drugs in ICUs are more diverse and complex than those in general wards, and the potential risks of medication errors to patients' lives are greater, making the prevention of medication errors a very important task.

The fatigue and stress of nurses undertaking long shifts in ICUs are potential factors contributing to medication errors owing to the possibility of confusion and misunderstanding at every stage of the drug use process. Possible causes of medication errors include mislabeling due to a similar appearance, similar drug names, the same drug being composed of different

*Corresponding author: e-mail: kfcmax300@gmail.com
https://doi.org/10.18494/SAM.2021.3241

ISSN 0914-4935

formulas, multiple abbreviations of the same concept, confusing abbreviations and symbols, different doses of the same drug, and confusing labeling.[4]

Even experts in drug administration may misidentify drugs due to fatigue. Therefore, additional mechanisms to assist medical personnel in the administration of drugs will be a future trend, which, combined with the vigorous development of AI technology and related hardware, will make it possible to achieve fast and accurate drug identification through the assistance of AI. In the previous research on drug identification, some mechanical equipment or additional measures were often required, or there was no uniform and effective solution for identifying multiple drug categories. Because the problems of different types of packaging, surfaces, placement positions, and angles of drugs need to be solved at the same time, there were certain difficulties in the practical use of previous approaches. Thus, we propose a system with an easy-to-use, efficient, and accurate drug verification function, thereby avoiding the potential risks of medication errors and the irreparable harm they may cause.

2. Related Work

2.1 Traditional methods

Some related studies on drug identification have been reported. In 2015, Yang *et al.*[5] used the barcode attached to the bottle body to identify the category of drug. In their system, a scanner was used to scan the barcode, which had to face the scanner. Che *et al.*[6] calculated a global threshold to binarize a gray-scale image as a feature vector, and then compared the similarity with an image feature vector in a database through template matching. However, their method ignored the problem of the placement angle. Zhang *et al.*[7] proposed a method of extracting the drug region of interest (ROI) based on edge detection, and obtained the diameter, height, and neck position of the bottle by applying predefined rules, and then used a trained Bayesian classifier to identify the drug category. In 2016, Gong *et al.*[8] proposed a way to continuously capture images with label surfaces in the form of conveyor belts, then combine the images to correct the boundary distortion, and then use optical character recognition (OCR) and a scale-invariant feature transform (SIFT) to match the template image and input image. However, in this method, it was necessary to ensure that the label of the drug on the conveyor belt faced the camera and that the bottle was placed upright when the conveyor belt started. In 2017, Xu *et al.*[9] proposed a method involving registering a complete and clear image of a drug in advance, which was expanded into a flat image as a template image through calculation, and a label image was segmented by a clustering algorithm. Their proposed algorithm extracted the feature matrix of a gray-scale image and compared it with the database image feature by cosine similarity. In the above methods, the shape, text, color, or texture on the label was used to identify drugs, but these methods usually had certain limitations, for example, the number of recognizable categories should not be too large or the accuracy was sensitive to the light source and rotation angle.

2.2 Deep learning methods

In 2018, Lee *et al.*[10] proposed the use of VGG-19 deep learning technology for drug recognition. The input image was obtained by photographing handheld drugs on mobile phones. Eleven categories of pill boxes were successfully identified, all of which were flat pill boxes, so there was no need to solve the problem of multiple differently placed feature surfaces. In 2020, Ting *et al.*[11] proposed the use of YOLOv2 deep learning technology to identify blister packs of drugs. They developed a single-stage detection system that included positioning and recognition that trained a model for each side of the blister pack. However, the model trained with text and logo features on the back of the blister pack performed better. The above research on the application of deep learning in drug recognition currently only recognizes specific types of packaging or containers, and directly uses deep learning technology to learn attention to ensure robustness to the effects of changes in the rotation angle.

3. Proposed System

The proposed system architecture is shown in Fig. 1. The system consists of two stages: a multiscale rotating rectangular drug detection system based on a deep residual network followed by a drug matching system based on feature embedding.

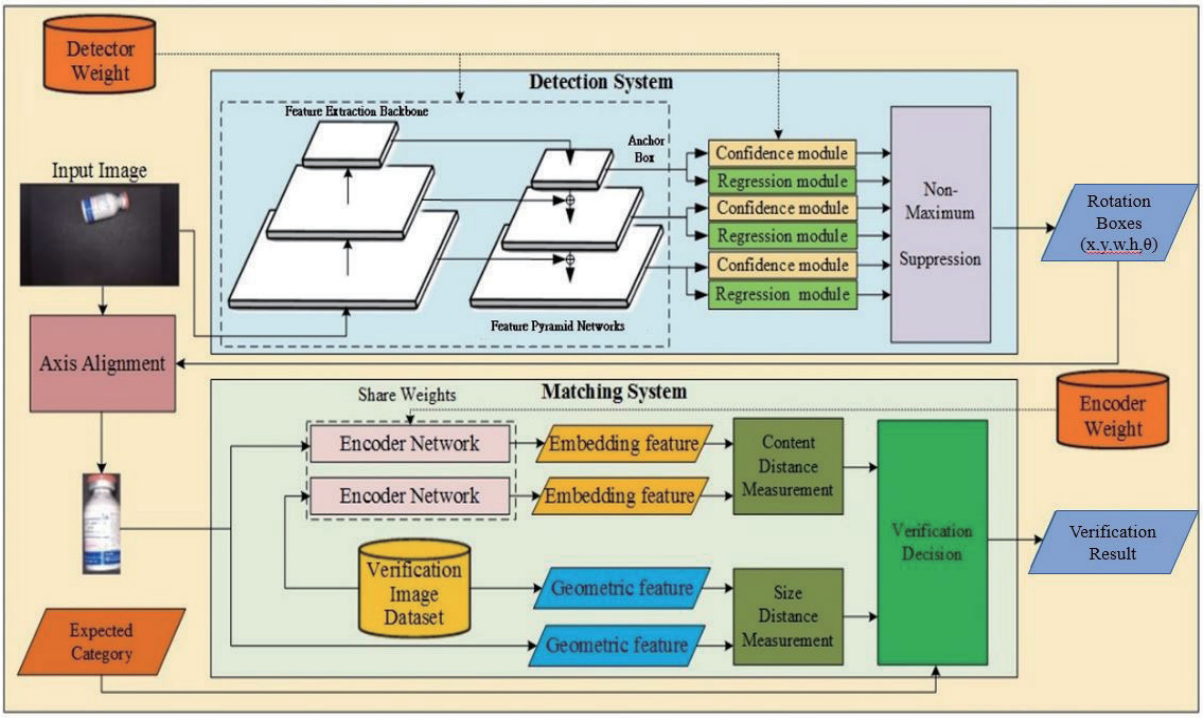

Fig. 1. (Color online) Architecture of the proposed system.

In the drug detection system, the input image is passed through a deep residual network to obtain feature maps of different scales, and a feature pyramid network (FPN) is used to merge features to build a multi-semantic feature map. For each feature map, a confidence module and regression module are used to obtain the confidence and spatial location of all rotation bounding boxes, and then the results of all scale feature maps are input to a non-maximum suppression module to obtain a rotation bounding box with high confidence. After obtaining all the rotation bounding boxes, axis alignment is performed to obtain the image information contained in the axis-aligned bounding boxes.

In the part of the drug matching system, a trained full convolution feature embedding network is used to embed the image output of the drug detection system and the image in the database, then the geometric features and the embedded features are compared separately, and finally, the results and the expected category are input to the decision module to determine whether the input drug image passes a verification.

3.1 Detection system

3.1.1 Feature extraction backbone

In the object detection system using deep learning, backbones with multiscale feature maps are needed as the basic feature extractor, but because of the need for multiple scales, the depth of the network must be sufficient. The backbones are ResNet-50 and ResNet-101,[12] and these two networks have two basic shortcut connection blocks, the architectures of which are shown in Fig. 2.

The architectures of ResNet-50 and ResNet-101 are shown in Table 1. The deep residual network can determine the required depth of the network through the shortcut connection block structure. The problem of the disappearance of gradients that may occur in deep networks can thus be solved.

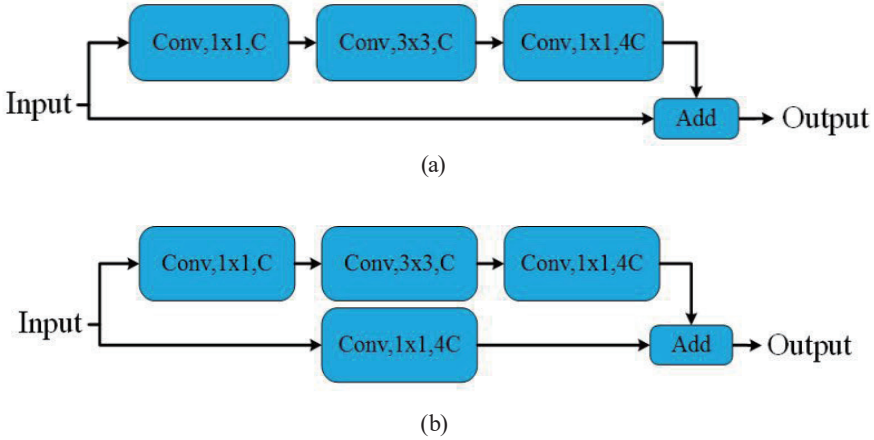

Fig. 2. (Color online) Architectures of shortcut connection blocks. (a) Identity block. (b) Projection block.

Table 1
Deep residual networks.

Network Name	ResNet-50	Network Name	ResNet-101
Conv1(C1)	7×7, $c = 64$, $s = 2$ 3×3 maxpool, $s = 2$	Conv1(C1)	7×7, $c = 64$, $s = 2$ 3×3 maxpool, $s = 2$
Conv2(C2)	$\begin{bmatrix} 1\times1 , 64 \\ 3\times3 , 64 \\ 1\times1 , 256 \end{bmatrix} \times 3$	Conv2(C2)	$\begin{bmatrix} 1\times1 , 64 \\ 3\times3 , 64 \\ 1\times1 , 256 \end{bmatrix} \times 3$
Conv3(C3)	$\begin{bmatrix} 1\times1 , 128 \\ 3\times3 , 128 \\ 1\times1 , 512 \end{bmatrix} \times 4$	Conv3(C3)	$\begin{bmatrix} 1\times1 , 128 \\ 3\times3 , 128 \\ 1\times1 , 512 \end{bmatrix} \times 4$
Conv4(C4)	$\begin{bmatrix} 1\times1 , 256 \\ 3\times3 , 256 \\ 1\times1 , 1024 \end{bmatrix} \times 6$	Conv3(C3)	$\begin{bmatrix} 1\times1 , 256 \\ 3\times3 , 256 \\ 1\times1 , 1024 \end{bmatrix} \times 6$
Conv5(C5)	$\begin{bmatrix} 1\times1 , 512 \\ 3\times3 , 512 \\ 1\times1 , 2048 \end{bmatrix} \times 3$	Conv5(C5)	$\begin{bmatrix} 1\times1 , 512 \\ 3\times3 , 512 \\ 1\times1 , 2048 \end{bmatrix} \times 3$

3.1.2 Feature pyramid network

The architecture of the FPN[13] in the proposed system is shown in Fig. 3. The FPN is used as a feature fusion method after these backbones to generate multi-semantic feature maps. The FPN contains a bottom-up pathway, a lateral connection, and a top-down pathway. The bottom-up pathway is composed of a backbone. The final feature map of each stage is used as our lateral transfer feature map.

The lateral connection uses a 1×1 convolution layer, the main purpose of which is to provide the number of channels required to combine with deeper features, and at the same time combine the information of each channel to obtain new features.

In the path from top to bottom, the deeper feature map is upsampled twice through bilinear interpolation, and then element-wise addition is performed with the shallower lateral connection results. In addition, to detect larger objects, it is necessary to obtain feature maps with a smaller spatial size because the receptive field is relatively large. A convolution with step 2 is used for downsampling. The final feature maps obtained from the backbone are C3, C4, and C5, which are used to generate P3, P4, and P5, then further generate P6 and P7 through the convolution with step 2. Feature map C2 is not used because its size would make the number of calculations huge.

3.1.3 Anchor box

In each spatial position of the output feature map, anchor boxes with different scales, aspect ratios, and angles are generated to meet objects of different shapes and placement angles. The sizes of the anchor box are defined as $\{32^2, 64^2, 128^2, 256^2, 512^2\}$, and these boxes are set in the

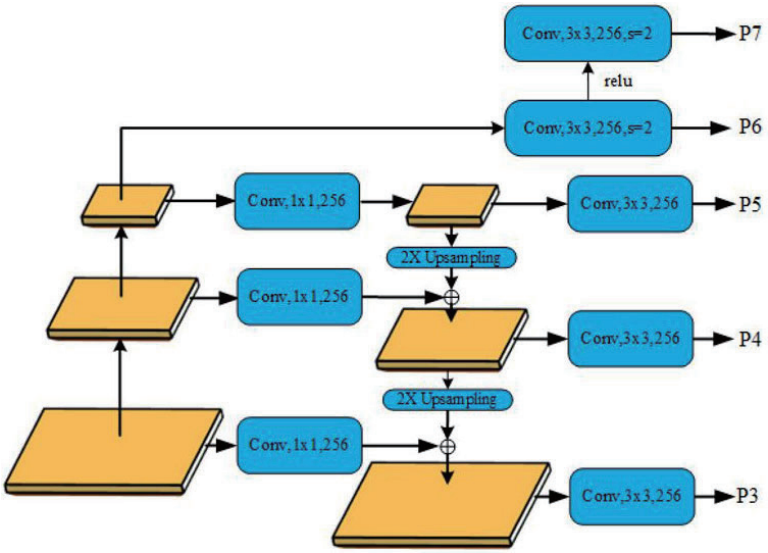

Fig. 3. (Color online) Architecture of the FPN.

$\{P_3, P_4, P_5, P_6, P_7\}$ layers of the feature pyramid, respectively. In each feature map, the height: width ratio for each anchor box is set to {1, 1/2, 2, 1/3.5, 3.5} with different scales $\{2^0, 2^{1/3}, 2^{2/3}\}$ and different angles {−85, −75, −65, −55, −45, −35, −25, −15, −5} to increase the density of anchor boxes.

3.1.4 Regression module and confidence module

For the output feature map of each layer of the feature pyramid P_3, P_4, P_5, P_6, P_7, we connected the regression module and then the confidence module. The architectures of these two modules are shown in Fig. 4. The regression module generates a bounding box spatial offset based on each anchor of all A anchors for each position on the feature map. Thus, the number of channels of the output is $5A$, which is the total length of the rotating bounding box vector (x, y, w, h, θ) corresponding to each anchor box. Similarly, the confidence module generates the confidence of each category of all A anchors for each position on the feature map of different spatial sizes. The only difference between the regression module and the confidence module is that the number of channels of the last connected 3×3 convolution layer is KA, where K is the total number of categories and KA is the total length of the confidence vector for the categories of all the rotation bounding boxes.

3.1.5 Post-processing

The spatial position and confidence of the prediction bounding box generated by the regression module and confidence module at each layer of the feature pyramid must be post-processed to obtain the final output. First, the prediction box predicted using the model is an offset relative to the anchor box, so it must be converted to an absolute value. The output of the

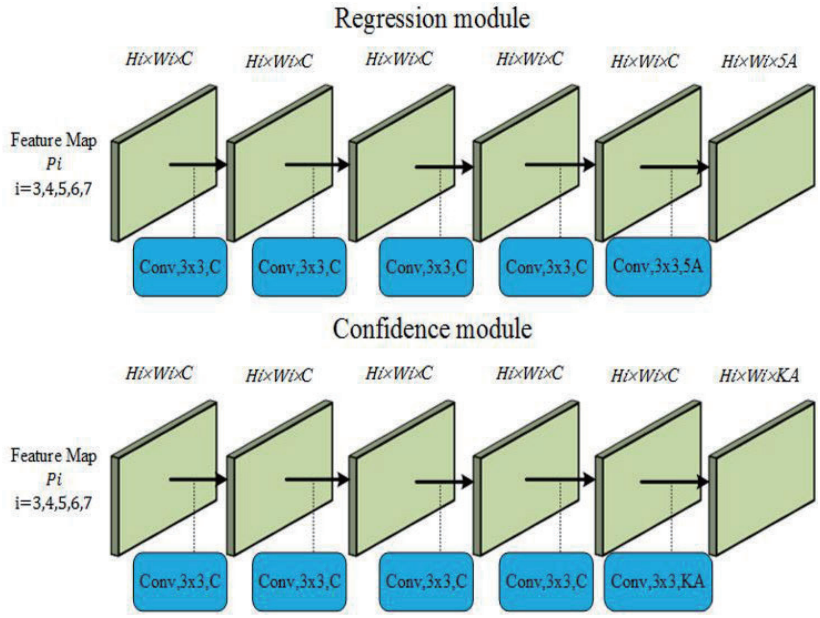

Fig. 4. (Color online) Architectures of regression and confidence modules.

prediction box is defined by the model $t = (t_x, t_y, t_w, t_h, t_\theta)$, the anchor box $a = (a_x, a_y, a_w, a_h, a_\theta)$, and the actual value $b = (b_x, b_y, b_w, b_h, b_\theta)$, and the conversion formula is given by the following equations.

$$b_x = a_w t_x + a_x \tag{1}$$

$$b_y = a_h t_y + a_y \tag{2}$$

$$b_w = a_w \exp(t_w) \tag{3}$$

$$b_h = a_h \exp(t_h) \tag{4}$$

$$b_\theta = t_\theta \times 180/\pi + a_\theta \tag{5}$$

In actual situations, there may be multiple predicted bounding boxes that contain the same object in the image, but this is unlikely to occur, so we must use non-maximum suppression to delete these redundant boxes. The algorithm is defined as Fig. 5.

3.1.6 Multitask loss function

Because there are multiple tasks to be performed, the multitask loss function is defined as

$$L(t,g,p,y) = \sum_{l=3}^{7} \left(L_{reg}^{pl}(t,g) + L_{con}^{pl}(p,y) \right), \tag{6}$$

```
Input : B = {b_1, ..., b_N}, S = {s_1, ..., s_N}, N_t
        B is the list of initial detection boxes
        S is the list of scores corresponding to detection boxes
        N_t is the NMS threshold
begin
        D ← {}
        while B ≠ empty do
            m ← argmax S
            M ← b_m
            D ← D ∪ M; B ← B − M
            for b_i in B do
                if iou(M, b_i) ≥ N_t then
                    B ← B − b_i; S ← S − s_i
            end
        end
        return D, S
end
```

Fig. 5. Algorithm of non-maximum suppression.

where t is the rotation bounding box predicted using the regression module, g is the label of the regression task, p is the confidence for each rotation bounding box predicted using the confidence module, and y is the label of the confidence task.

The regression loss is defined as Eq. (7), which refers to IOU smooth L1 loss.[14] The regression task involves maximizing the area of intersection between the bounding box and the ground truth.

$$L_{reg}(t,g)=\sum_{i\in x,y,w,h,\theta}\frac{smooth_{L1}(t_i-v_i)}{\left|smooth_{L1}(t_i-v_i)\right|}\left|-\log\left(IOU(b,g)+\varepsilon\right)\right| \tag{7}$$

Here, v is the label of the regression task in the offset relative to the anchor box and b is the prediction box converted into an absolute value from t, which is the offset of the anchor box.

The confidence loss is defined as Eq. (8), which is referred to as focal loss.[15] Focal loss reduces the loss of samples that are easy to classify (such as background), and increases the loss of samples that are more difficult to classify (objects of interest), which is equivalent to solving the problem of category imbalance.

$$L_{con}(p,y)=\frac{1}{p}\sum_{i=1}-\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{8}$$

Here, p_t is the probability of correctly predicting that a drug belongs to the ground-truth class. γ is the modulation factor, and α_t is the weighting factor.

3.2 Matching system

3.2.1 Feature embedding network

A very important part of the matching system is to obtain a global feature description of the image, which may be its color or texture. In our system, the feature description is obtained through a fully convolutional encoder network. The architecture of the encoder is shown in the left of Fig. 6, and feature extraction is performed through four identical and continuous CONV blocks. The architecture of a CONV block is shown in the right of Fig. 6.

The spatial range of the image of each category after feature embedding will be within a certain distance, and the feature distribution of each category must have a center point, called the prototype. The training algorithm of the feature embedding network is shown in Fig. 7. During training, two encoders are used, which share weights. One of the encoder inputs is the support set, and the average value of the corresponding output is used as the prototype. The other encoder input is the query set. It inputs the query set feature and prototype calculation distance into the loss function and then adjusts the distance between them according to the label. Through this training method, the distance between image features of the same category is reduced by the loss function, and the distance between features of different categories is increased.

3.2.2 Feature distance measurement

In the inference stage, we compare the embedded features and geometric features of the input image with those of the image in our database, so a definition of similarity or distance is required. The following equation is used to calculate the similarity of embedded features:

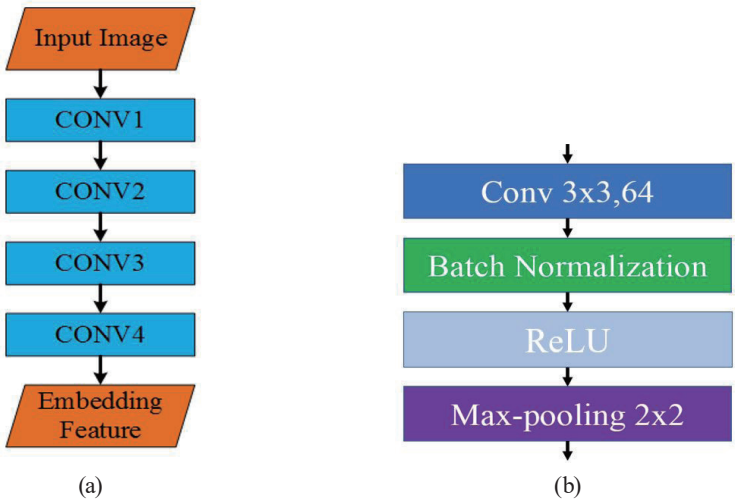

Fig. 6. (Color online) Image feature embedding network. (a) Encoder. (b) CONV block.

Training Algorithm

Definition:

N is the number of examples in the training set
K is the number of classes in the training set
$N_c \leq K$ is the number of classes per episode
N_s is the number of support examples per class
N_q is the number of query examples per class
RANDOMSAMPLE(S,N) denotes a set of N elements chosen uniformly at random from set S, without replacement

Input:

Training set $D = \{(x_1, y_1), \cdots, (x_N, y_N)\}$, where each $y_i \in \{1, \cdots, K\}$.
D_k denotes the subset of D containing all elements (x_i, y_i) such that y_i=k.

Output:

The loss J for a randomly generated training episode

$V \leftarrow RANDOMSAMPLE(\{1, \cdots, K\}, N_c)$ ∇ select class indices for episode
for k in $\{1, \cdots, N_c\}$ **do**
  $S_k \leftarrow RANDOMSAMPLE(D_{V_k}, N_s)$ ∇ select support set examples
  $Q_k \leftarrow RANDOMSAMPLE(D_{V_k} \backslash S_k, N_q)$ ∇ select query set examples
  $c_k \leftarrow \frac{1}{N_c} \sum_{(x_i, y_i) \in S_k} f_\emptyset(x_i)$ ∇ compute prototype for every class
end for
J ← 0 ∇ initialize loss
for k *in* $\{1, \cdots, N_c\}$ **do**
  for (x, y) in Q_k **do**
  $J \leftarrow J + \frac{1}{N_c N_q} [d(f_\emptyset(x), c_k) + log \sum_{k'} \exp(-d(f_\emptyset(x), c_{k'}))]$ ∇ update loss
  end for
end for

Fig. 7. Training algorithm of feature embedding network.

$$sim_{emb} = p_{\varnothing}\left(y = k \mid x\right) = \frac{\exp\left(-d\left(f_{\varnothing}\left(x\right),\ c_k\right)\right)}{\sum_{k'}^{N} \exp\left(-d\left(f_{\varnothing}\left(x\right),\ c_{k'}\right)\right)}, \tag{9}$$

where k is the expected category, N is the total number of categories, $d(x, y)$ is the function used to calculate the distance between the features (we use the Euclidean distance), c_k is the prototype, and $f_\emptyset(x)$ is the embedded feature of the input image obtained by the encoder. The following equations are used to calculate the similarity of geometric features:

$$sim_{geo\_H} = \frac{min\left(feat_H^k, feat_H\right)}{max\left(feat_H^k, feat_H\right)}, \tag{10}$$

$$sim_{geo\_W} = \frac{min\left(feat_W^k, feat_W\right)}{max\left(feat_W^k, feat_W\right)}, \tag{11}$$

where superscript k represents the feature of the prototype of the expected category. The method used to compare the similarly of geometric features is to compare the long side with the long side and the short side with the short side. The long and short sides produce similarity separately, and when the numerator is small and the denominator is large, the similarity of geometric features is low.

3.2.3 Verification decision

In the decision of whether to pass the verification, the corresponding thresholds are set to give a certain fault tolerance space, and the decision rules are as follows:

$$sim_{emb} \geq 0.5 , \tag{12}$$

$$sim_{geo\_H} \geq 0.85 , \tag{13}$$

$$sim_{geo\_W} \geq 0.85 . \tag{14}$$

The above rules must all be satisfied in this system for the identification of the drug to be considered as verified, that is, the embedded features and geometric features have a certain level of similarity.

4. Experiments

The drug image verification dataset was collected in cooperation with Kaohsiung Veterans General Hospital (KVGH). To maintain a certain image quality, we used a Logitech C920 Pro webcam to collect the images, which had a resolution of 1920 × 1080 pixels. The database included images taken with various placement positions and placement angles. The drug image verification dataset contained 21 categories of drugs from KVGH. Since our system is composed of two subsystems for training, our dataset was also divided into two parts, namely, the detection dataset and matching dataset, for the training of the individual subsystems. A sample image from each dataset is shown in Fig. 8. The matching dataset was cropped by our trained detection system, and we evaluated our drug verification system by applying this matching dataset to the matching system.

In the experimental comparison, the Euclidean distance was used to compare the false positive rate (FPR) under different training parameters. The experimental results are shown in Table 2, where n-way represents the number of categories in each training support test and n-shot represents the number of images in a single category for each training support set. The experimental results show that when n-way is relatively high but n-shot is relatively low, the FPR will be high. A higher n-shot reduces the FPR, and under our experimental comparisons, the final 10-way 20-shot training obtained the lowest FPR (0.047%).

(a) (b)

Fig. 8. (Color online) Sample images taken from datasets. (a) Image from detection dataset. (b) Images from matching dataset.

Table 2
Results of different training methods for our system.

Training method	FPR (%)
5-way 5-shot	0.104
5-way 20-shot	0.063
10-way 5-shot	0.112
10-way 20-shot	0.047
20-way 5-shot	0.132
20-way 20-shot	0.063

Table 3
Comparison with other image feature embedding methods.

Training method	FPR (%)
Siamese network[16]	0.151
Triplet network[17]	0.169
Ours (cosine distance)	0.124
Ours (Euclidean distance)	0.047

The experimental results of a comparison with other methods are shown in Table 3. The proposed method is divided into feature distance calculation methods using the cosine distance or Euclidean distance. It can be seen from the experimental results that the Siamese network and triplet network, which respectively use dual samples or triples in a single training, have higher FPR values than our method. That is, feature embedding of our dataset will help improve the FPR if we consider multiple category features for training.

The results with the lowest FPR in Table 3 are shown as a confusion matrix in Table 4. We evaluate the results with the following common metrics:

- Precision = TP / (TP + FP) = 99.03%
- Recall = TP / (TP + FN) = 98.32%
- Accuracy = (TP + TN) / (TP + FP + FN + TN) = 99.87%
- FPR = FP / (FP + TN) = 0.047%
- FRR = 1 − Recall = 1.68%

Here, TP is true positive, FP is false positive, TN is true negative, and FN is false negative.

It can be seen that the FPR for our method with the Euclidean distance is 0.047%, which means that the incorrect categorization of a drug can be effectively avoided. The FRR is 1.68%, which means that most drugs of the correct category can pass verification, giving the system high efficiency.

Table 4
Confusion matrix.

$N = 17514$	Positive samples	Negative samples
Verification passed	820 (TP)	8 (FP)
Verification not passed	14 (FN)	16672 (TN)

	class0	class1	class2	class3	class4	class5	class6	class7	class8	class9	class10	class11	class12	class13	class14	class15	class16	class17	class18	class19	class20	Not pass
class0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
class1	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
class2	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
class3	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
class4	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
class5	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
class6	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
class7	0	0	0	0	0	0	0	38	2	0	0	0	0	0	0	0	0	0	0	0	0	0
class8	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0
class9	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	3
class10	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0
class11	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0
class12	0	0	0	0	0	0	0	0	0	0	0	0	38	2	0	0	0	0	0	0	0	0
class13	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	1
class14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0
class15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
class16	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	38	0	0	0	0	0
class17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	1
class18	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0
class19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0
class20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	2

Fig. 9. (Color online) Detailed confusion matrix.

Fig. 10. (Color online) FN samples.

We analyzed the incorrectly identified samples in the experiments. Figure 9 is a detailed confusion matrix for all 21 classes; note that the table cannot display the number of TNs.

Figure 10 shows some FN samples, that is, positive samples that the system believed should not be verified. The FN for the image on the left may be due to a detection angle error or shooting exposure. The detection error for the image on the right may have been due to the poor focus of the label and its closeness to the boundary.

The pair of images in Fig. 11(a) have the same bottle body and the labels have the same background color. The pair of images in Fig. 11(b) show bottles with no features on the surface,

(a) (b) (c) (d)

Fig. 11. (Color online) Misidentified cases.

resulting in the images being very similar. The pair of images in Fig. 11(c) also have no obvious differences in the color features. The pair of images in Fig. 11(d) have the same bottle body and the same background color of the label, and the words on the label are similar in color.

5. Conclusions

In this study, we propose a drug image verification system, which includes an automatic drug detection system and a matching system. The dataset used contains drugs of 21 categories for performance evaluation of verification tasks. The database used in this research is an image database created by imaging drugs provided by domestic local medical centers. Our verification system achieved an FPR of 0.047% in verification tasks of drugs of 21 categories. Furthermore, the proposed system can be integrated into a mobile phone, so that users can easily identify a drug at any time. In the future, we will attempt to achieve a higher recognition rate for our proposed drug image verification system by collecting more databases and make the system more practical.

References

1 R. M. Kruer, A. S. Jarrell, and A. Latif: Clin. Pharmacol. Adv. Appl. **6** (2014) 117.
2 D. J. Cullen, B. J. Sweitzer, D. W. Bates, E. Burdick, A. Edmondson, and L. L. Leape: Critical Care Med. **25** (1997) 1289.
3 I. Cho, H. Park, Y. J. Choi, M. H. Hwang, and D. W. Bates: PLOS ONE **9** (2014) 1. https://doi.org/10.1371/journal.pone.0114243
4 S. Farzi, A. Irajpour, M. Saghaei, and H. Ravaghi: J. Res. Pharm. Pract. **6** (2017) 158. https://doi.org/
5 H. B. Yang, L. Xiang, L. Chen, and W. L. Zhou: Chin. J. Hosp. Pharm. **35** (2015) 2142. https://doi.org/10.13286/j.cnki.chinhosppharmacyj.2015.23.19
6 X. S. Che, Y. Li, and D. J. Liu: Int. Conf. Testing and Measurement Techniques (2015) 377. https://doi.org/10.1201/b18470-84
7 R. Zhang, Y. He, P. Zhang, Y. Hu, Y. Cao, and J. Zhang: 2015 IEEE Int. Conf. Information and Automation (2015) 1157. https://doi.org/10.1109/ICInfA.2015.7279461
8 M. Gong, X. Zhang, and T. Ni: 2016 2nd Int. Conf. Artificial Intelligence and Industrial Engineering (AIIE, 2016) 227. https://doi.org/10.2991/aiie-16.2016.53
9 H. Xu, H. Liu, and D. Liu: 2017 3rd Int. Conf. Control, Automation and Robotics (ICCAR) (2017) 410. https://doi.org/10.1109/ICCAR.2017.7942728

10 S. Lee, S. Jung, and H. Song: J. Comput. Sci. Eng. **12** (2018) 149. https://doi.org/10.5626/JCSE.2018.12.4.149
11 H.-W. Ting, S.-L. Chung, C.-F. Chen, H.-Y. Chiu, and Y.-W. Hsieh: BMC Health Services Res. **20** (2020) 1. https://doi.org/10.1186/s12913-020-05166-w
12 K. He, X. Zhang, S. Ren, and J. Sun: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2016) 770. https://doi.org/10.1109/CVPR.2016.90
13 T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2017) 2117. https://doi.org/10.1109/CVPR.2017.106
14 X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu: Proc. IEEE/CVF Int. Conf. Computer Vision (2019) 8232. https://doi.org/10.1109/ICCV.2019.00832
15 T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár: Proc. IEEE Int. Conf. Computer Vision (2017) 2980. https://doi.org/10.1109/ICCV.2017.324
16 J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah: Adv. Neural Inf. Process. Syst. **6** (1993) 737. https://doi.org/10.1142/S0218001493000339
17 F. Schroff, D. Kalenichenko, and J. Philbin: Proc. IEEE Conf. Computer Vision and Pattern Recognition (2015) 815. https://doi.org/10.1109/CVPR.2015.7298682

About the Authors

Shih-Pang Tseng received his B.S. and M.S. degrees from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, and his Ph.D. degree from the Department of Computer Science Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan. He is now a professor at Changzhou College of Information Technology. His current research interests include deep learning, NLP, robotics, and learning technology. (tsengshihpang@ccit.js.cn)

Che-Wen Chen received his B.S. degree from the Industry Engineering Management Department, Yuan Ze University, Taoyuan, Taiwan, in 2008, his M.S. degree from the Department of Civil Engineering, National Cheng Kung University, Tainan, Taiwan, in 2012, and his Ph.D. degree from the Department of Electrical Engineering, National Cheng Kung University, in 2020. His current research interests include deep learning, NLP, robotics, and data science. (kfcmax300@gmail.com)

Wei-Yan Jang received his B.S. degree from the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, in 2018 and his M.S. degree from the Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, in 2020. His current research interests include deep learning and related applications of computer vision. (a0916231808@gmail.com)

Jhing-Fa Wang received his B.S. and M.S. degrees from the Department of Electrical Engineering, National Cheng Kung University (NCKU), Tainan, Taiwan, in 1973 and 1979, respectively, and his Ph.D. degree from the Department of Computer Science and Electrical Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, in 1983. He is an IEEE Life Fellow and was the chair of the IEEE Tainan Section and the Coordinator/Chapter, Region 10, IEEE. He is currently the chair and a distinguished professor in the Department of Electrical Engineering, NCKU. He developed a Mandarin speech recognition system called Venus-Dictate, which is recognized as a pioneering system in Taiwan. He is currently leading a research group of different disciplines for the development of advanced ubiquitous media for created cyberspace. He has authored nearly 150 papers in journals published by the IEEE, the Society for Industrial and Applied Mathematics (SIAM), the Institute of Electronics, Information and Communication Engineers, and the Institute of Electrical Engineers, and approximately 250 conference papers. (jameswangjf@gmail.com)