IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2023, accepted 21 October 2023, date of publication 25 October 2023, date of current version 31 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3327421

RESEARCH ARTICLE

Improving Automatic Identification of Medications in Transparent Packaging by Glare Removal and Color Correction

CHENG-CHIN CHIANG, (Member, IEEE), YU-YU YANG, WEI-LIN LIU, AND YI-CHENG LIN
Department of Computer Science and Information Engineering, National Dong Hwa University, Shoufeng, Hualian 974, Taiwan

Corresponding author: Cheng-Chin Chiang (ccchiang@gms.ndhu.edu.tw)

ABSTRACT Automated Medication Identification (AMI) systems can significantly streamline the daily tasks of pharmacists. Nonetheless, the image analysis methods utilized by AMI systems often encounter difficulties in real-world settings. For example, glare reflections from transparent packaging and color distortions from different lighting conditions may alter the visual appearances of medication images, thereby degrading the recognition accuracy. This paper proposes an innovative approach to mitigate these issues, incorporating image registration techniques to eliminate glare reflections and correct color discrepancies to improve recognition accuracy. The proposed solutions are integrated with a novel ResDenseNet neural network architecture, which efficiently merges cross-level features via skip connections, harnessing the combined merits of ResNet and DenseNet. Empirical evaluations reveal that this integrated solution significantly elevates the recognition rate from 15.19% to 96.85% for a ResDenseNet model trained on a dataset with limited appearance variations. Furthermore, the ResDenseNet outperforms ResNet and DenseNet by 7.5% and 3.3% in recognition rate, respectively.

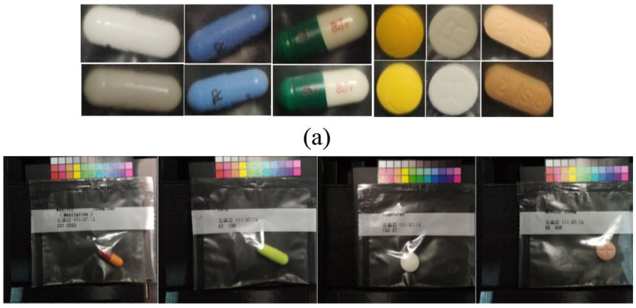
INDEX TERMS Deep learning, medication identification, color correction, glare removal, object detection, YOLO, ResNet, DenseNet, image registration.

I. INTRODUCTION

Automated Medication Identification (AMI) technology has become increasingly important for hospitals facing high demands of medications for patients. It helps pharmacists verify medications and reduce the burden of manual confirmation on each patient's prescriptions. Moreover, an AMI system integrated with medication inventory management systems can further enhance its benefits in accelerating the medication dispensing process, improving medical efficiency, reducing human errors, ensuring patient safety, and becoming a critical tool for enhancing operational efficiency in hospitals [1]. Despite its advantages, AMI faces several technical challenges. One of the significant obstacles is color distortions, which can occur due to different light sources or imaging devices. This can significantly hinder AMI's effectiveness since color is often a critical identification feature for many medications. Additionally, transparent packaging can create glare reflections, obscuring the medications' appearance and making identification more difficult. Fig. 1 illustrates some problematic cases caused by glare reflections and color distortions. Moreover, accurately distinguishing between medications with similar appearances highly depends on defining subtle features and developing algorithms to extract them which pose non-trivial technical challenges.

While several studies on medication identification have achieved high accuracy, challenges such as glare reflections and color distortions have never been addressed [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. To improve the performance of AMI systems in hospitals, our goal in this paper is to propose effective solutions to overcome these challenges. To achieve this, we develop methods for removing glare reflections and correcting distorted colors during image preprocessing.

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.


(a)

(b)

FIGURE 1. Some examples of packaged medications with glare reflections and color distortions: (a) medications with distorted colors; (b) medication packages with glare reflections.

Additionally, we design deep learning models that can learn to extract effective features for distinguishing similar-looking medications.

The main contributions of our work in this paper include:

- designing a glare removal method that repairs pixels disrupted by glare reflections through registration between multiple images taken under different lighting and viewpoints;
- designing a color correction method that corrects distorted colors through registration between the input image and a color checker;
- designing a deep learning model that leverages the strength of both ResNet [11] and DenseNet [12] to extract more effective cross-level features for medication identification.

II. RELATED WORKS

AMI is a specialized research topic with dedicated applications. Few techniques are specifically crafted for AMI, particularly in addressing the challenges and issues mentioned previously. Therefore, we review some general-purpose methods related to the core tasks of AMI systems in this section.

A. GLARE REMOVAL

There can be two approaches to address the glare reflections: non-learning-based methods and learning-based methods. Non-learning-based methods detect and remove glare by specifically human-crafted image processing techniques, while learning-based methods involve training a model to convert input images with glare to output images without glare through automatic data-driven training.

Non-learning-based methods often utilize prior assumptions to distinguish between reflection and background. These assumptions can include ghosting cues [13], depth of field maps [14], [15], and sparsity priors [16]. For example, the retinex-based algorithm [17] uses the distribution of gradients to reduce the reflection component and reveal the underlying scene. Another technique is the polarizer-based approach, which uses external polarizing filters to suppress reflections and reduce glare. However, non-learning-based methods have limitations such as requiring additional hardware costs, being less adaptable to varying lighting conditions, being sensitive to specific application scenarios, and requiring time-consuming parameter tuning to achieve satisfactory performances.

Several deep learning-based algorithms have been proposed to effectively and comprehensively capture reflection properties. For instance, Fan et al. devised CEILNet [18] to eliminate reflections from a solitary image by predicting the edge map and reconstructing the glare-free target image using two subnetworks. CoRRN [19] and CRRN [20] also consisted of sub-networks for extracting background information, estimating the background gradient, and separating the background from reflection layers. Zhang et al. [21] designed one subnetwork to separate reflection and background layers to restore overlooked details with an attention strategy. Abiko and Ikehara [22] proposed a GAN-based algorithm utilizing a UNet++L^4 [23] generator to produce the desired reflection-free image. Li et al. proposed the IBCLN (Iterative Boost Convolutional LSTM Network) [24], which comprises two cascaded blocks, GT and GR, for iteratively refining the estimates of the transmission and reflection layers in an image.

Despite the demonstrated effectiveness of these learning-based glare removal methods, they have several limitations, such as extensive demands in training data, difficulty in adapting to new glare situations, susceptibility to training biases, high computational costs, and long training duration. These limitations motivate further research into developing more robust and efficient glare removal algorithms for AMI systems.

B. COLOR CORRECTION

Two main approaches exist for developing color correction methods: the reference-free-base approach and the reference-based approach. Reference-free methods correct color distortions within an image without using a reference image. These methods include white balance [25], histogram equalization [26], and the gray world algorithm [27]. They adjust the color or brightness distribution of the image to match a presumed normal distribution. However, these methods may not be suitable for practical applications that involve complex and fluctuating lighting conditions. Assuming a generic distribution for the image could lead to significant color disparities if the lighting conditions do not align with the assumption.

Reference-based color correction methods use a color reference image with patterns of known true colors to achieve color correction. This can be accomplished through two kinds of methods: location-independent and location-dependent methods. The location-independent methods determine the color correction model by aligning the global color distributions of the input image and the color checker. As a result, the primary colors appearing in the image would significantly impact the results of color correction. Conversely, the location-dependent methods infer the color correction model based on scene matching between the color-distorted image and the reference image. The accuracy of color correction would highly depend on the accuracy of image registration between the two images. Examples of location-independent methods include Histogram Matching [28] and PCA Color Matching [29]. The method proposed in [30] is an example of the location-dependent method.

Recent studies have utilized deep learning techniques, such as the one presented in [31], to create color correction models based on an extensive collection of samples. Although deep learning methods can produce sophisticated color correction models, they require many training samples. These samples must encompass a broad spectrum of lighting conditions; otherwise, the resulting model may struggle with diverse lighting variations. Given that lighting conditions can vary widely across different application scenarios, gathering sufficient representative samples would pose a challenge. Therefore, utilizing deep learning techniques for color correction may not be cost-effective in practical applications.

C. MEDICATION DETECTION

In recent years, deep learning methodologies have led to significant advancements in object detection, surpassing traditional computer vision approaches in detection accuracy, stability, and speed. Three prominent deep-learning paradigms for object detection are the two-stage, one-stage, and attention-based approaches.

The two-stage approach, introduced by Girshick et al. [32], is based on the framework of Region-based Convolutional Neural Network (R-CNN). This approach involves generating region proposals via a selective search algorithm in the first stage, followed by classification using a Convolutional Neural Network (CNN) in the second stage. Several variants of the R-CNN, such as Fast R-CNN [33], Faster R-CNN [34], and Spatial Pyramid Pooling Network (SPP-Net) [35], have been proposed to address issues such as slow speed and high computational complexity.

The one-stage approach, first presented by Redmon et al. [36], is a real-time object detection technique known as You Only Look Once (YOLO). This approach utilizes a single neural network to simultaneously predict bounding boxes and class probabilities. Another noteworthy one-stage method is the Single Shot Detector (SSD) [37], which also integrates image classification and object detection within a unified neural network.

The attention-based approach is a technique that employs attention mechanisms to identify crucial areas within images. One such method is Reverse Connection with Objectness Prior Networks (RON) [38]. This model enhances the Faster R-CNN algorithm by creating feature maps from different convolutional layers. Attention scores are then used to weigh the level of feature activation within the bounding boxes, thereby improving object detection accuracy in images.

D. MEDICATION IDENTIFICATION

Identifying medications can be a difficult task as medications often bear a striking resemblance to one another. Discriminating between them requires professional expertise to explore their subtle discerning features. One potential solution is employing deep learning models that can automatically learn effective visual features for medication identification.

In a recent study conducted by Delgado et al. [1], four deep networks, namely ResNet50 [2], SqueezeNet [3], and InceptionV3 [4], were compared for identifying 924 types of medication. The results showed that ResNet50 achieved the highest accuracy, with an averaged Top-1 accuracy of 76.3% and an averaged Top-5 accuracy of 94.8%. Another study by Chang et al. [5] developed a system for identifying pills using smart glasses and advanced neural networks. The study evaluated three backbone architectures within the SSD and Faster R-CNN networks, including Mobilenet, Inception Net, and ResNet50. The ResNet50 backbone outperformed the others, achieving a recognition rate of 95.1%, but only for four types of pills with noticeable visual differences. In another study, Wu et al. [6] proposed an Attention-YOLO deep learning model specifically designed for recognizing round pills in images. The model achieved an accuracy of 92.28% for identifying 316 types of round pills.

Ting et al. [7] employed YOLOv2 [39] to identify 250 different medications based on images of their blister packaging. Two models were developed using images of the front and back sides of the packaging, respectively. The back-side model was more accurate, with an accuracy of 96.26%, compared to the accuracy of 94.09% for the front-side model. It is essential to note that the blister packs used in the study are distinct and have no transparent packaging. Chen et al. [8] also developed a system for recognizing medications with distinct packaging. The system had two processing stages. The first stage used a deep residual network and a feature pyramid network (FPN) [9] to identify the rotation of bounding boxes for medications and crop the medication patches accordingly. In the second stage, a feature embedding network combined a derived embedding feature with the geometric feature obtained from the first stage. The system achieved an accuracy of 99.87% when evaluated on 828 samples of 21 types of packaged medications.

Heo et al. [10] developed a pill retrieval system that combines models for image classification and text recognition to retrieve similar pills from a large database. The system uses YOLOv5 [9] to extract and recognize characters imprinted on the pill and ResNet-32 to classify using the pill's visual features, such as shape, color, and form. The search results of each query are then obtained based on the recognized texts and visual features. The system achieved a Top-1 accuracy of

85.65% and a Top-3 accuracy of 92.35% when evaluated on a MFDS dataset [10].

While previous studies on medication identification have achieved remarkable recognition rates, they have predominantly focused on medications without any packaging or with easily recognizable packaging. There are no glare reflections and color distortions caused by transparent packaging that impede the recognition rates. Hence, these issues have never been addressed in previous studies.

Based on the survey of the general methodologies for the four core tasks in developing an AMI system, we can see that previous glare removal methods, both non-learning and learning, have limitations and are not tailored for medication identification tasks. The non-learning approach, relying upon a specific model separating the background from glare, is less adaptable due to inherent presumptions. In contrast, the learning approach faces restrictions from training bias and computational efficiency. In color correction, reference-free methods, which rely on preset assumptions, are not well-suited for diverse real-world conditions. Reference-based methods, meanwhile, necessitate the accurate registration between the input and standard reference images.

Our study aims to tackle the challenges posed by glare reflections and color distortions in AMI, setting our work apart from previous studies and underlining its significance. In light of the limitations of previous works, we design an automated imaging platform to capture images from multiple viewpoints under multiple lighting conditions. The controlled imaging environment of the platform allows for stable image registration, thereby enhancing both glare removal and color correction. Besides, both ResNet and DenseNet architectures employ skip connections to effectively integrate features across different granularity levels. To leverage the strengths of both ResNet and DenseNet, we propose a ResDenseNet model to achieve a recognition rate superior to that of either individual model.

III. THE PROPOSED METHOD

A. SYSTEM COMPONENTS AND OPERATIONS

The proposed AMI system, as shown in Fig. 2, comprises a client-side subsystem and a server-side subsystem. The client-side subsystem manages an automatic imaging apparatus that captures images of medications in transparent packages. The server-side subsystem performs four key operations on the captured images: glare removal, medication detection, color correction, and medication identification. To support the proposed methods of glare removal and color correction for medications in transparent packaging, we build an automatic imaging apparatus utilizing a medication conveyor belt, two LED light sources, and a smartphone, as shown in Fig. 3. The smartphone, accompanied by a designated app, controls an ESP32 microcontroller to operate the conveyor belt via Bluetooth. The app alternates between different light sources via the ESP32 controller as the conveyor belt moves medication packages to predefined positions. This setup enables the automatic acquisition of medication package images from multiple viewpoints under different light conditions.

After capturing images of medication packages under different viewpoints and lighting conditions, the client-side subsystem sends these images to the server-side subsystem via WiFi. Once the images are received, the glare removal module in the server-side subsystem removes the glare reflections and then sends the glare-removed image to the detection module. After receiving the glare-removed image, the detection module detects and extracts the color checker from the image, which is required by the color correction module. On receiving the extracted patch of the color checker, the color correction module performs the color correction on the glare-removed image and then sends the color-corrected image back to the detection module. The detection module detects medications on the color-corrected image and crops them into individual patches. Afterward, each cropped medication patch is sent to the identification module for recognition. Finally, the identification result is relayed to the client-side app via WiFi for displaying on the smartphone.

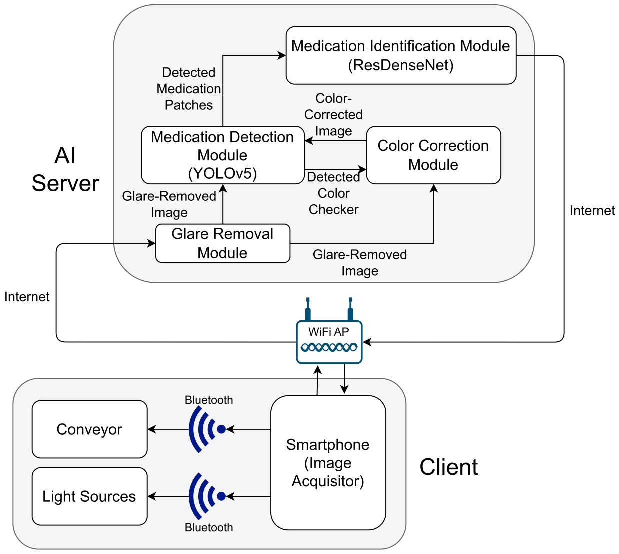


FIGURE 2. The diagram of components in the proposed AMI system.

B. PREPROCESSING

1) GLARE REMOVAL

Generally, there can be two strategies to mitigate the detrimental effects of glare reflections on transparent packages, including:

1) re-positioning the light source to prevent direct illumination on the package surface might help if the glare is due to lighting, or
2) capturing the image of the medication package from a different viewpoint could reduce or eliminate the glare.

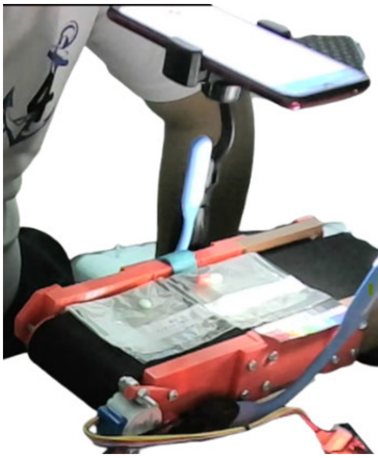


FIGURE 3. The developed apparatus for automatic image capturing of medication packages.

Following these two strategies, we implement the apparatus illustrated in Fig. 3 to capture multiple images of packaged medications from different viewpoints under varying lighting conditions. We then perform glare removal by registration between the captured multiple images. With this apparatus's well-controlled imaging environment, the image registration becomes stable and accurate to identify the positional correspondences among the images which allows us to rectify pixels disrupted by glare reflections using their non-glared counterparts from alternate images. The detailed method is presented in the following.

Given any pair of images I_m and I_n, we detect SIFT feature points [40] on them and match these feature points to obtain a set of pairs of matched feature points, $MP = \{(\mathbf{p}_i^{(m)}, \mathbf{p}_i^{(n)})\}_{i=1}^{K}$, where $\mathbf{p}_i^{(m)}$ and $\mathbf{p}_i^{(n)}$ represent the homogeneous coordinates of the i-th pair of matched feature points. We apply the RANSAC algorithm [41], as listed in Algorithm 1, to iteratively estimate a homography $H^*_{m,n}$ for mapping any coordinate $\mathbf{p}^{(n)}$ on I_n to its corresponding point $\mathbf{q}$ on I_m. In each iteration of this algorithm, the function **RandomSample**() in Step 5 randomly samples k ($k \geq 4$) pairs of matched feature points from MP into a set S. With S, the function **EstimateHomography**() in Step 6 estimates the homography $H^*_{m,n}$ by employing the Direct Linear Transform algorithm [42]. Consequently, any pair of matched feature points $(\mathbf{p}, \mathbf{q})$ in MP is considered an inlier if it satisfies the following condition:

$$\|H \cdot \mathbf{q} - \mathbf{p}\| \leq \epsilon, \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm. This condition states that the mapped position of $\mathbf{q}$ is close to $\mathbf{p}$ within a predetermined threshold ϵ (e.g. 3 pixels) with respect to H, meaning that $\mathbf{p}$ and $\mathbf{q}$ validly identify a pair of corresponding pixels on I_m and I_n, respectively. This condition creates a set of inliers Ω in Step 7 in each iteration. During the iterations, the best homography $H^*_{m,n}$ is determined by the estimated homography that identifies the most inliers, as shown in Steps 8-10 of Algorithm 1.

Algorithm 1 RANSAC Image Registration

```
Require: The Homogeneous coordinates of the set of matched pair of feature points on $I_m$ and $I_n$, $MP = \{(\mathbf{p}_i^{(m)}, \mathbf{p}_i^{(n)})\}_{i=1}^{K}$
Require: The identification numbers of $I_m$ and $I_n$, $m$ and $n$
Require: A maximum number of iterations, $T$
Require: The number of matched points for estimating a homography matrix, $k$
Require: A threshold for inliers, $\epsilon$
Ensure: The homography matrix, $H^*_{m,n}$
 1: procedure Ransac($m, n, MP, T, k, \epsilon$)
 2:     $\Omega = \{\}$
 3:     $M = 0$
 4:     for $i = 1$ to $T$ do
 5:         $S$ = RandomSample($MP, k$)
 6:         $H$ = EstimateHomography($S$)
 7:         $\Omega = \{(\mathbf{p}, \mathbf{q}) \in F \mid \|H \cdot \mathbf{q} - \mathbf{p}\| \leq \epsilon\}$
 8:         if $|\Omega| > M$ then
 9:             $H^*_{m,n} = H$
10:             $M = |\Omega|$
11:         end if
12:     end forreturn $H^*_{m,n}$
13: end procedure
```

Given M images, I_j for $j = 1, \ldots, M$, of a medication package captured by our imaging apparatus, let $I_j(\mathbf{p})$ denote the intensity value of the pixel at position $\mathbf{p}$ in image I_j. To repair a glared pixel $\mathbf{p}$ in image I_j, we first compute a mean intensity for this pixel $\mathbf{p}$ from all corresponding pixels of $\mathbf{p}$ on other images by

$$\mu_j(\mathbf{p}) = \frac{\sum_{k=1}^{M} \mathbb{1}(H^*_{k,j} \cdot \mathbf{p} \in I_k) I_k(H^*_{k,j} \cdot \mathbf{p})}{\sum_{k=1}^{M} \mathbb{1}(H^*_{k,j} \cdot \mathbf{p} \in I_k)}. \tag{2}$$

where $\mathbb{1}(\mathbf{q} \in I_k)$ denotes an indicator function that outputs one if the pixel position $\mathbf{q}$ falls within the image I_k, and 0 otherwise. Note that $(H^*_{k,j} \cdot \mathbf{p})$ maps the position of $\mathbf{p}$ on I_j to the position of its corresponding point on I_k using the homography $H^*_{k,j}$. We can then repair each pixel at $\mathbf{p}$ in I_j by updating $I_j(\mathbf{p})$ as follows:

$$I_j(\mathbf{p}) = \begin{cases} \min\limits_{k}\{I_k(H^*_{k,j} \cdot \mathbf{p})\}, & \text{if } I_j(\mathbf{p}) - \mu_j(\mathbf{p}) \geq \tau \\ I_j(\mathbf{p}), & \text{otherwise} \end{cases} \tag{3}$$

where τ is a threshold, 30, determined empirically. Equation (3) states that if any pixel $\mathbf{p}$ on I_j has a brightness level exceeding the computed mean intensity $\mu_j(\mathbf{p})$ by the threshold τ, then it is judged as a glared pixel and can be repaired with the lowest intensity among those of the corresponding pixels of $\mathbf{p}$. Algorithm 2 shows the steps to repair the glared pixels. The function **ComputeMean**() in this algorithm computes the mean intensity according to equation (2). Both the **DetectSIFT**() and **Match**() are

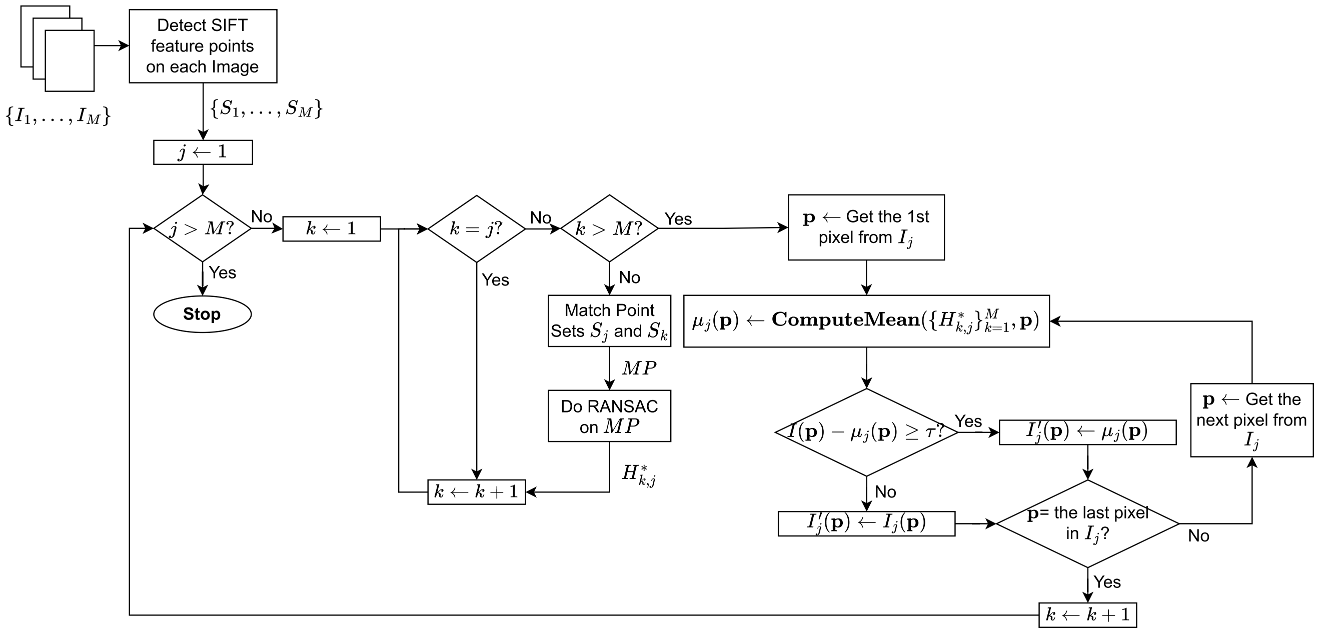

$\{I_1, \dots, I_M\}$
Detect SIFT feature points on each Image
$\{S_1, \dots, S_M\}$
$j \leftarrow 1$
$j > M?$
No
Yes
Stop
$k \leftarrow 1$
$k = j?$
No
Yes
$k > M?$
Yes
No
Match Point Sets S_j and S_k
MP
Do RANSAC on MP
$H^*_{k,j}$
$k \leftarrow k + 1$
$\mathbf{p} \leftarrow$ Get the 1st pixel from I_j
$\mu_j(\mathbf{p}) \leftarrow \textbf{ComputeMean}(\{H^*_{k,j}\}^M_{k=1}, \mathbf{p})$
$I(\mathbf{p}) - \mu_j(\mathbf{p}) \geq \tau?$
Yes
$I'_j(\mathbf{p}) \leftarrow \mu_j(\mathbf{p})$
No
$I'_j(\mathbf{p}) \leftarrow I_j(\mathbf{p})$
$\mathbf{p}$= the last pixel in I_j?
No
Yes
$\mathbf{p} \leftarrow$ Get the next pixel from I_j
$k \leftarrow k + 1$

FIGURE 4. The flowchart of the proposed glare removal method.

implemented using built-in functions in OpenCV. Fig. 4 shows the flowchart of the glare removal method.

```
Algorithm 2 Glare Removal Algorithm
Require: The captured images to be processed: $I_1, \dots, I_M$
Require: The threshold parameter: $\tau$
Ensure: The output images with glare reflections removed: $I'_1, \dots, I'_M$
procedure RemoveGlare($[I_1, \dots, I_M], \tau$)
    for $j$ from 1 to $M$ do        ▷ Detect SIFT on each image
        $S_j = \textbf{DetectSIFT}(I_j)$
    end for
    for $j$ from 1 to $M$ do        ▷ Repair images one by one
        for $k$ from 1 to $M$ and $k \neq j$ do
            $MP = \textbf{Match}(S_k, S_j)$
            $H^*_{k,j} = \textbf{Ransac}(k, j, MP, 100, 4, 3)$
        end for
        for $\mathbf{p} \in I_j$ do
            $\mu_j(\mathbf{p}) = \textbf{ComputeMean}(\{H^*_{k,j}\}^M_{k=1}, \mathbf{p})$
            if $I_j(\mathbf{p}) - \mu_j(\mathbf{p}) \geq \tau$ then
                $I'_j(\mathbf{p}) = \min_k\{I_k(H^*_{k,j} \cdot \mathbf{p})\}$.
            else
                $I'_j(\mathbf{p}) = I_j(\mathbf{p})$.
            end if
        end for
    end for
    return $I'_1, \dots, I'_M$.
end procedure
```

2) COLOR CORRECTION

Our method for color correction employs the reference-based approach, incorporating the use of a standard color checker.

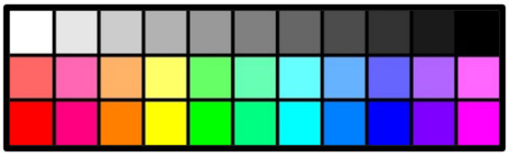


FIGURE 5. The color checker used for our color correction.

As shown in Fig. 5, this color checker contains a broad spectrum of colors and is placed strategically within the camera's field of view on each captured image. Let C_R be the referenced image of this standard color checker which has no color distortions. Assume that the color checker in a color-distorted image I is represented by C_I. The color correction aims to infer a color correction function, denoted as $f()$, to remap the color-distorted pixels in C_I back to the colors of their undistorted counterparts in C_R.

Like the glare removal method, finding the pixel correspondences between C_I and C_R requires the image registration between them. We exploit a YOLOv5 neural network to detect the color checker and crop it, denoting it as C_I, from the input image. Subsequently, the **Ransac** algorithm, as detailed in Algorithm 2, is employed to find a homography $T \in \mathbb{R}^{3\times3}$ which identifies the position of the corresponding pixel $\mathbf{q}$ in C_I for any given pixel position $\mathbf{p}$ in C_R by $(T \cdot \mathbf{p})$.

Once the homography T between C_I and C_R is derived, we then map a set of uniformly sampled reference pixels, denoted by $P = \{\mathbf{p}_i | \mathbf{p}_i \in C_R, i = 1, \dots, N\}$, on C_R to their corresponding pixels, denoted by $Q = \{\mathbf{q}_i | \mathbf{q}_i = T \cdot \mathbf{p}_i, \forall \mathbf{p}_i \in P\}$, on C_I. Let $\mathbf{c}_I(\mathbf{q}) = [r^I_{\mathbf{q}}, g^I_{\mathbf{q}}, b^I_{\mathbf{q}}]^T$ denote the RGB color tuple of the pixel $\mathbf{q}$ in I. In this study, we assume the color correction function $f()$ to be a linear transformation which can be expressed as:

$$f(\mathbf{c}_I(\mathbf{q})) = F \cdot \mathbf{c}_I(\mathbf{q}) \in \mathbb{R}^3, \tag{4}$$

where F is a 3×3 transformation matrix. To minimize the differences between the corrected colors and their corresponding true colors, the color correction function $f()$ must ideally minimize the following objective function:

$$L(F) = \sum_{i=1}^{N} \|\mathbf{c}_R(\mathbf{p}_i) - F \cdot \mathbf{c}_I(\mathbf{q}_i)\|^2 \tag{5}$$

where $\|\cdot\|$ represents the Euclidean norm. Let F be represented as

$$F = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix}. \tag{6}$$

To solve the least squares problem given by equation (5), we can first reshape F into a vector $\mathbf{f} \in \mathbb{R}^9$ and then obtain $\mathbf{f}$ using the pseudo-inverse operation [43] as follows:

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = (X_I^T X_I)^{-1} X_I^T \mathbf{c}_R(P), \tag{7}$$

where

$$X_I = \begin{bmatrix} \mathbf{c}_I^T(\mathbf{q}_1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_I^T(\mathbf{q}_1) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}_I^T(\mathbf{q}_1) \\ \mathbf{c}_I^T(\mathbf{q}_2) & & \\ \mathbf{0} & \mathbf{c}_I^T(\mathbf{q}_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}_I^T(\mathbf{q}_2) \\ \vdots & \vdots & \vdots \\ \mathbf{c}_I^T(\mathbf{q}_N) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_I^T(\mathbf{q}_N) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{c}_I^T(\mathbf{q}_N) \end{bmatrix} \in \mathbb{R}^{3N \times 9}, \tag{8}$$

and $\mathbf{c}_R(P) = \mathbf{c}_R(\mathbf{p}_1) \oplus \mathbf{c}_R(\mathbf{p}_2) \oplus \cdots \oplus \mathbf{c}_R(\mathbf{p}_N) \in \mathbb{R}^{3N}$ represents the ordered concatenation of the color tuples of all pixels in P. The detailed steps of the above color correction procedure are listed in Algorithm 3 and the data flow of this algorithm is illustrated in Fig. 6. The **PrepareX**() and **PrepareR**() functions in Algorithm 3 construct the matrix X_I in equation (8) and the vector $\mathbf{c}_R(P)$, respectively. The function **PseudoInv**() performs the computation presented in equation (7) and **Reshape**() reshapes the vector $\mathbf{f}$ as a 3×3 homography matrix F presented in equation (6). Fig. 7 illustrates an example of the registration between C_I and C_R.

The above method allows for the efficient computation of the matrix F. Moreover, due to the registration between C_I and C_R, our color correction method is adaptable to various imaging conditions in real-world scenarios. Furthermore, the method involves no collection and training of a large volume of comprehensive samples required in those learning-based methods.

Algorithm 3 Color Correction Algorithm

```
Require: Color distorted image: $I$
Require: Reference image of the color checker: $C_R$
Require: Uniformly sampled pixels on the reference image
    $C_R$: $P$
Ensure: Color corrected image: $I'$
procedure ColorCorrect($I, C_R$)
    $C_I$ = YoLoDetect($I$)
    $P'$ = DetectSIFT($C_I$)
    $Q'$ = DetectSIFT($C_R$)
    $MP$ = Match($P', Q'$)
    $T$ = RANSAC($I, C_R, MP, 100, 4, 3$)
    $Q = \{T \cdot \mathbf{p}_i | \forall \mathbf{p}_i \in P\}$
    $X_I$ = PrepareX($Q, C_I$)
    $\mathbf{c}_R$ = PrepareR($P, C_R$)
    $\mathbf{f}$ = PseudoInv($X_I, \mathbf{c}_R$)
    $F$ = Reshape($\mathbf{f}$)
    for $\mathbf{q} \in I$ do
        $[r_{\mathbf{q}}^{I'}, g_{\mathbf{q}}^{I'}, b_{\mathbf{q}}^{I'}]^T = F \cdot [r_{\mathbf{q}}^{I}, g_{\mathbf{q}}^{I}, b_{\mathbf{q}}^{I}]^T$.
    end for
    return $I'$.
end procedure
```

C. MEDICATION DETECTION

Medication detection can be accomplished using object detection technology and deep learning is currently the most groundbreaking approach in this area. Among several neural network models, we have chosen YOLOv5 as our preferred medication detector due to its impressive features, including real-time speed, no region proposal requirement, superior detection of small objects, and integration of object recognition with detection. Although YOLOv5 can also be used to identify medications, it may fall short in characterizing finer details such as markings, imprints, or labels that are crucial for fine-grained identification of similar-looking medications. Therefore, we direct YOLOv5's task solely toward medication detection. Namely, all 218 types of medications to be identified by our AMI system are labeled as a single object category on training YOLOv5 for detection. In addition to the color checker required by our color correction method, the total number of object

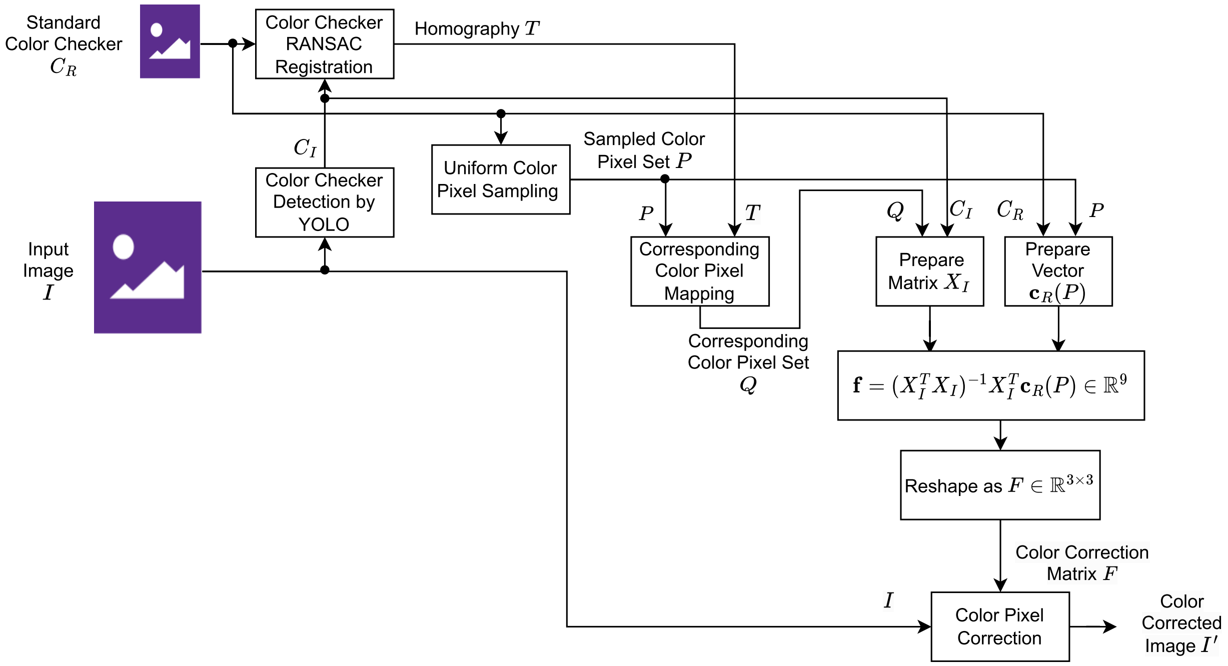


FIGURE 6. The data flow of the proposed color correction method.

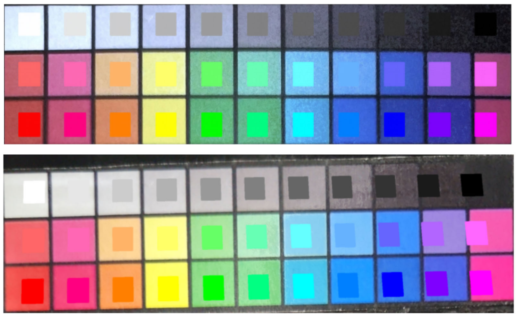


FIGURE 7. The top image shows the standard color checker C_R, while the bottom image shows the detected one C_I. The estimated homography is used to map the smaller colored squares centered at uniformly sampled points $\mathbf{p}_i$'s inside the color grids on C_R to their corresponding quadrilaterals with the same color centered at points $\mathbf{q}_i$'s on C_I.

categories to train YOLOv5 for detection is only 2, thereby significantly reducing our efforts needed to label the data for training.

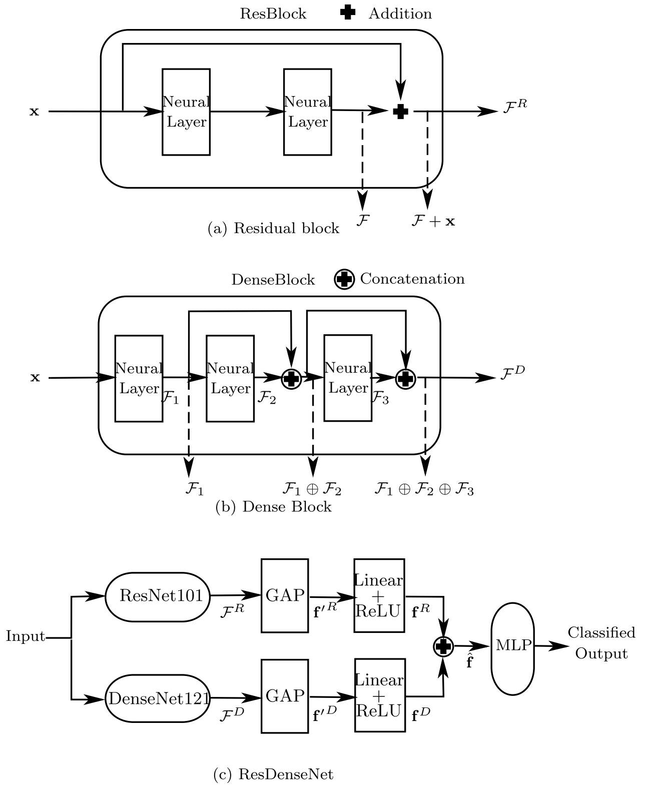
D. MEDICATION IDENTIFICATION

ResNet comprises successive "residual blocks," whereas DenseNet comprises successive "dense blocks." Every residual block in ResNet employs skip connections, allowing the direct addition of its input to the output of the block, as depicted in Fig. 8(a). This mechanism provides a more immediate pathway for integrating results from adjacent residual blocks. Similarly, within a DenseNet, each dense block concatenates the features extracted by a neural layer with those extracted by all preceding layers via skip connections, as shown in Fig. 8(b). This also facilitates efficient feature fusion across different resolution levels. In summary, through skip connections, both ResNet and DenseNet are highly suitable for learning the integration of cross-level features which assist in discerning subtle differences between similar medications. To combine the strength of these two models, we propose a ResDenseNet model to integrate both networks as backbones for feature extraction in the identification module of our AMI system.

The ResDenseNet consists of two backbones, ResNet and DenseNet, employed for feature extraction, as depicted in Fig. 8(c). Each backbone uses its final block to produce a feature map, denoted as $\mathcal{F}^B = \{F^B_{c,h,w}\} \in \mathbb{R}^{C\times H\times W}$, where C, H, and W refer to the channel, height, and width of the feature map, respectively. Note that the notation B can denote either R (representing ResNet) or D (representing DenseNet). Each feature map is then aggregated into a feature vector, $\mathbf{f}'^B = [f_1'^B, \ldots, f_C'^B]^T \in \mathbb{R}^C$, through Global Average Pooling (GAP). The i-th element, $f_i'^B$, of $\mathbf{f}'^B$ computed from the GAP operation is calculated as follows:

$$f_i'^B = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathcal{F}^B_{i,h,w}, \quad \text{for } i = 1, \ldots, C. \tag{9}$$

A linear layer, accompanied by the ReLU activation, is added after the GAP operation to transform $\mathbf{f}'^B$ into another feature vector $\mathbf{f}^B$ with a specified dimension. Upon obtaining the two feature vectors, $\mathbf{f}^R$ and $\mathbf{f}^D$, from the two linear layers, a final feature vector, $\hat{\mathbf{f}}$, is formed by concatenating them, i.e., $\hat{\mathbf{f}} = \mathbf{f}^R \oplus \mathbf{f}^D$, where $\oplus$ represents the



(a) Residual block

(b) Dense Block

(c) ResDenseNet

FIGURE 8. The architecture of the proposed ResDenseNet: (a) the residual block used in ResNet; (b) the dense block used in DenseNet; (c) the proposed ResDenseNet.

concatenation operator. Subsequently, this final feature vector is supplied to a multilayer perceptron (MLP) for medication identification.

IV. EXPERIMENTAL RESULTS

A. DATASETS AND ENVIRONMENTAL SETUP

We have received 218 packaged medications from a hospital, each containing a single type of medication. This packaging arrangement allows us to label image samples more easily. We use these medication packages to create three datasets. The first dataset, called the pretraining dataset, is acquired using several smartphones under general imaging conditions. The second dataset, called the pragmatic dataset, is created with three different smartphones that can be mounted on our automated imaging apparatus, exhibiting fewer variations under a controlled imaging environment. The third dataset, called the challenging dataset, is obtained by replacing the smartphone, as well as the lighting source, in the imaging apparatus with an entirely different one. Each dataset has its distinct purpose in our experiments. The pretraining dataset is used exclusively to pre-train the YOLOv5 model to handle general imaging conditions. Given that YOLOv5's medication detection precedes medication identification, its performance could potentially limit the overall system's recognition rate. Therefore, we hypothesize that training data with more variations in imaging conditions can enhance detection accuracy. The pragmatic dataset is partitioned into training and testing subsets, serving for finetuning/assessing YOLOv5 and training/testing the three deep learning models for medication identification. On the other hand, the challenging dataset is exclusively used to evaluate the adaptability of the three models to varied imaging conditions. Hence, it is not used in any training process. Table 1 outlines the characteristics of these three datasets, including the sizes, the numbers of training/testing samples, their intended purposes, and imaging conditions.

Regarding the environmental setup, Table 2 details the hardware/software configurations for model training and testing. The architecture of the network layers and the number of model parameters for ResNet, DenseNet, and ResDenseNet are enumerated in Table 3.

1) TRAINING FOR MEDICATION DETECTION

We use the pretraining dataset to pretrain YOLOv5. To augment the training data, we apply randomized transformations, including scaling, cropping, rotation, Gaussian blur, color jittering, grayscale conversion, and histogram equalization. All are supported by either YOLOv5 or PyTorch. Having been trained on the pretraining dataset, YOLOv5 is already capable of accurately detecting most medications. To further enhance its detection capability, we identified 128 medication types that show room for improvement in detection performance from the test results of the pretraining dataset. We then randomly select 28 training samples per type from the pragmatic dataset to bolster the training on these 128 types of medications. Our primary consideration for adopting this approach is to alleviate the burden of manual data annotation for too many types of medications.

2) TRAINING FOR MEDICATION IDENTIFICATION

Our ResDenseNet incorporates ResNet101 and DenseNet121 as backbone networks for feature extraction. The dimension of each input medication patch to the network is 160 × 160 pixels. As listed in Table 1, we use a pragmatic dataset consisting of 6104 samples (28 samples per type) for training and 2616 samples (12 samples per type) for testing. We follow two approaches to train each model for different purposes of experiments, including

- Uniform Training: Under this training approach, each training sample undergoes glare removal and color correction, thereby achieving a more uniform appearance. This training approach can evaluate how the proposed glare removal and color correction methods affect the recognition rate of a model trained by samples with uniform and limited variations in appearance.
- Diverse Training: Under this training approach, each training sample does not undergo glare removal and color correction, resulting in a diverse range of variations in appearance. This training approach can assess

TABLE 1. The three datasets used in the experiments.

Dataset Name	Size	Samples per Type	# of Training Samples	# of Testing Samples	Purpose	Imaging Condition
Pretraining Dataset	17,440	80	12,208	5,232	Detection pretraining	Smartphones A, B, C under general environments
Pragmatic Dataset	8,720	40	2,384	5,136	Detection finetuning/testing	Smartphones A, B, C and lighting 1, 2 under automatic imaging apparatus
			6,104	2,616	Identification training/testing	
Challenge Dataset	872	2	0	872	Identification testing on a different imaging condition	Smartphone D and lighting 3 under automatic imaging apparatus

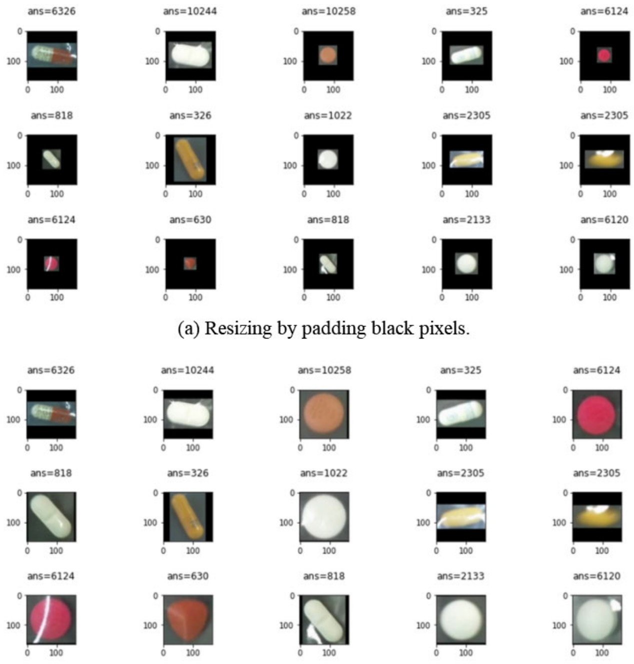
TABLE 2. The configurations of hardware/software/training parameters for the experiments.

Hardware	CPU: i7-12700K GPU: RTX3090=24G RAM: 80G DDR4 3200MHz ROM: Gen4 SSD
Software	Python: 3.9.16 PyTorch: 1.12.1+cu116 OpenCV: 4.7.0.72 THOP
Training Parameters for YOLO	Learning Rate: 0.01 Optimizer: SGD Batch Size: 32
Training Parameters for ResNet/DenseNet/ResDenseNet	Learning Rate: 2e-5 Optimizer: Adam (β_1=0.9,β_2=0.999 ϵ:1e-9) Batch Size: 48

Data Augmentation (PyTorch)	RandomRotation(180, expand=True, center=None,fill=None)
	ColorJitter(brightness=(0.85,1.15), contrast=(0.85,1.15), saturation=(0.85,1.15), hue=(-0.1,0.1))
	GaussianBlur(3,(1,3))
	RandomCrop()
	Grayscale Conversion (YOLOv5),
	Histogram Equalization (YOLOv5)

a model's ability to handle glare reflections and color distortions through data-driven training of diversified samples.

Each medication patch, which is cropped from YOLOv5's detection, is typically smaller than 160 × 160 pixels under the operation of our automatic imaging apparatus. To be compatible with the input dimension of ResNet, DenseNet, and ResDenseNet, we pad each patch with black pixels, as shown in Fig. 9(a). This padding process preserves the aspect ratio and relative sizes between different medications, which is vital because different dosages of the same medication must be treated as distinct types in the hospital's identification task. The only visual difference between them is size. Proportional scaling would lose the relative sizes between different medication types, as illustrated in Fig. 9(b). During the training process, the training epochs continue until the recognition rate on the training set reaches 98%. The learning curves of these three models are depicted in Fig. 10. The curves indicate that ResDenseNet achieves the most rapid learning.



(a) Resizing by padding black pixels.

(b) Resizing by proportional scaling followed by padding black pixels.

FIGURE 9. The resizing of image samples by padding black pixels.

B. EXPERIMENTAL RESULTS FOR PREPROCESSING

The pragmatic dataset is used to test the effectiveness of our proposed glare removal and color correction methods. Fig. 11 presents some results of the glare removal process. The two leftmost images in each row depict the two captured input images from a test sample, with partial areas significantly disrupted by glare reflections. The two rightmost images exhibit the results after applying glare removal to the respective input images. According to the outcomes, restoring

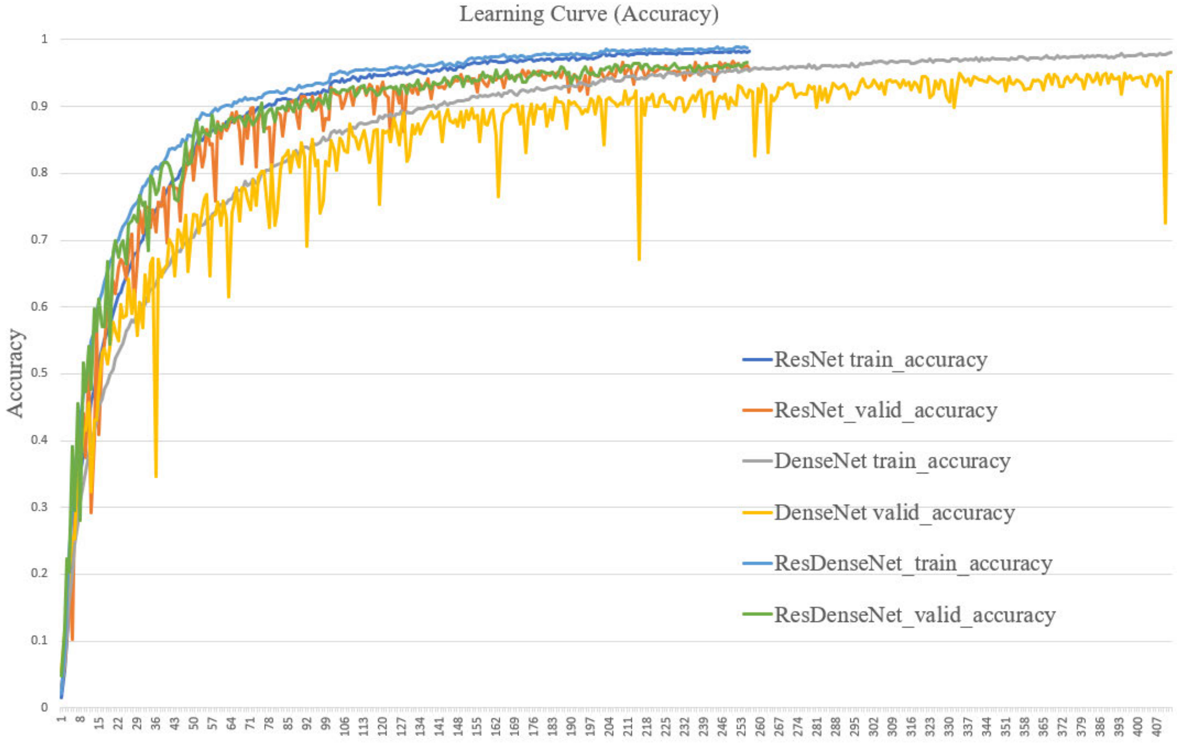


FIGURE 10. **The learning curves of ResNet, DenseNet, and ResDenseNet.**

glared pixels in one image by using the corresponding non-glared pixels from the other can notably reduce glare reflections.

To verify the effectiveness of image registration done before the glared pixel repairing, we illustrate the registration results of three example cases in Fig. 12. As shown in this figure, the registrations are pretty accurate thanks to the grid patterns of the color checker.

In the experiment of color correction, we apply four distinct color tones of lighting sources to induce different color distortions. We then conduct color correction on these four color-distorted images. After color correction, we examine whether the results of these four images achieve color consistency to verify the effectiveness of our color correction method. Fig. 13 displays some color correction results. In each odd row, columns one through four show image samples of one medication package, while columns five through eight show image samples of another. Images in even rows show the color-corrected versions of their corresponding color-distorted images in the odd rows above them. The results show that our color correction method maintains color constancy effectively in all four images of each medication package. The satisfactory correction results require accurate registration between the captured color checker and the standard color checker. Fig. 14 shows the registration of color checkers for three example cases. Again, the registration results are pretty good because of the grid patterns on the color checkers.

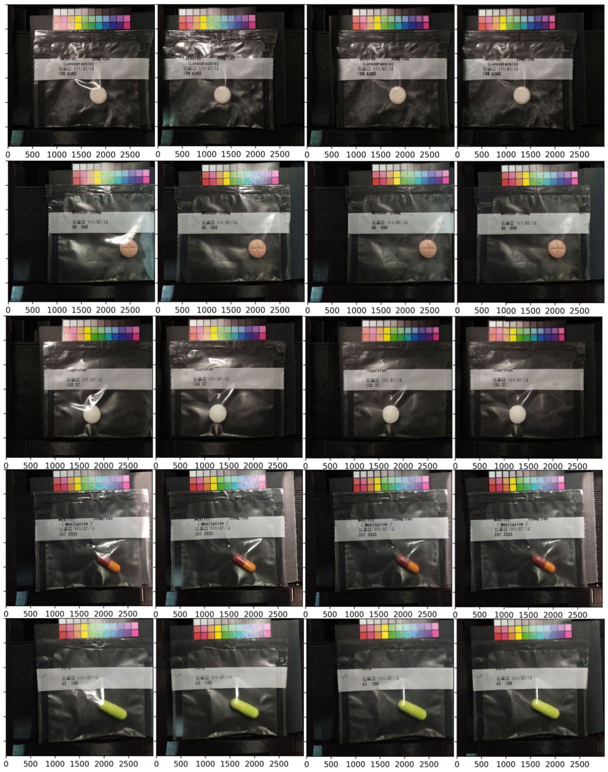


FIGURE 11. **Some results of glare removal. The two leftmost images in each row depict the two captured input images from a test sample. The two rightmost images exhibit the results after applying glare removal to the respective input images.**

TABLE 3. The layer constructions of ResNet101, DenseNet121, and ResDenseNet.

Layers	Output Size	ResNet101	DenseNet121	ResDenseNet	
Input	160 × 160	channel=3			
Conv1	160 × 160	5 × 5, 64			
Conv2_x	80 × 80	3 × 3 max pool			
		$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times6$	$\begin{bmatrix}1\times1,64\\3\times3,64\\1\times1,256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times6$
Conv3_x	40 × 40	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times12$	$\begin{bmatrix}1\times1,128\\3\times3,128\\1\times1,512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times12$
Conv4_x	20 × 20	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times24$	$\begin{bmatrix}1\times1,256\\3\times3,256\\1\times1,1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times24$
Conv5_x	10 × 10	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times16$	$\begin{bmatrix}1\times1,512\\3\times3,512\\1\times1,2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1\\3\times3\end{bmatrix}\times16$
Classification Layer	1 × 1	global average pool			
	-	218FC, softmax	218FC, softmax	512FC	512FC
				Concate, 218FC, softmax	
Total Numbuer of Parameters		7.18M	42.95M	51.25M	

TABLE 4. The confusion matrices for the testing experiment on YOLOv5. Column 2 shows the results of testing samples without preprocessing, while Column 3 shows the results of testing samples with preprocessing.

	Test on Samples w/o Preprocessing			Test on Samples w/ Preprocessing		
	True Medication	True Color Checker	True Background	True Medication	True Color Checker	True Background
Medication Detected	5131	0	4	5134	0	1
Color Checker Detected	0	5136	0	0	5136	0
Background Detected	1	0	0	1	0	0

C. EXPERIMENTAL RESULTS FOR MEDICATION DETECTION

We conduct two experiments using the pragmatic dataset to evaluate the performance of YOLOv5's medication detection. The first experiment tests YOLOv5 using samples without glare removal and color correction. The purpose is to assess its ability to handle glare reflections and color distortions through data-driven learning. In contrast, the second experiment tests YOLOv5 with glare-removed samples to verify whether our glare-removal method improves YOLOv5's detection. The testing samples for YOLOv5 cannot undergo color correction because it requires the cropped color checker from YOLOv5's detection.

We selected 5136 samples, which are not involved in finetuning YOLOv5, as the testing set. Table 4 presents the confusion matrices for the two experiments. In the first experiment, YOLOv5 fails to detect one medication in a sample because the color of the medication closely resembles the background color, as shown in Fig. 17. Additionally, there are four false positives caused by glare reflections. However, the results in the second experiment indicate that our glare removal method can correctly remove three of the four false positives. Fig. 15 and Fig. 16 show four example cases' detection and identification results.

D. EXPERIMENTAL RESULTS FOR MEDICATION IDENTIFICATION

As detailed in Section IV-B, we employ two distinct training approaches: uniform training, which entails training on samples that have undergone glare removal and color correction, and diverse training, which involves training on samples without glare removal and color correction. Based on these two training approaches, we evaluate the models using the following five options:

- **Option I**: Recognizing test samples without preprocessing by training models using uniform training.
- **Option II**: Recognizing test samples that undergo only glare removal by training models using uniform training.
- **Option III**: Recognizing test samples that undergo only color correction by training models using uniform training.
- **Option IV**: Recognizing test samples that undergo both glare removal and color correction by training models using uniform training.

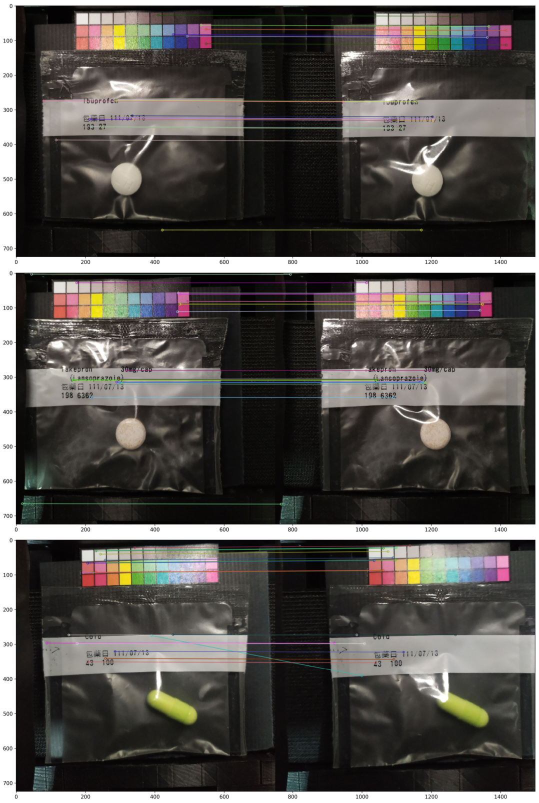


FIGURE 12. The results of image registration in glare removal for three example cases. Each pair of matched corresponding points is connected with a colored line. The results are quite accurate because of the regular grid patterns of the color checker.

- **Option V**: Recognizing test samples without preprocessing by training models using diverse training.

Tests with Options I through IV aim to examine the effect of the two proposed preprocessing methods on the recognition accuracy of the models trained with samples of uniform appearance variations. Tests with Option V aim to evaluate the neural network's ability to resist diverse variations purely through data-driven training. The recognition accuracies of ResNet, DenseNet, and ResDenseNet are assessed to see if ResDenseNet outperforms both ResNet and DenseNet.

Table 5 shows the Top-1, Top-3, and Top-5 recognition rates for tests of the five options on the pragmatic dataset. The Top-K recognition rate is defined as

$$\text{Top-}K \text{ recognition rate} = \frac{\sum_{x_i \in D} \mathbb{1}(rank(x_i) \leq K)}{|D|} \quad (10)$$

where x_i denotes the i^{th} sample in the testing set D, $|D|$ denotes the size of D, $rank(x_i)$ denotes the ranking of x_i among the candidates of the identification output, and $\mathbb{1}(c)$ is 1 if condition c is true, and 0 otherwise. The five columns, with the identifiers A1∼A5 at the bottom row, correspond to the tests with the above five options, respectively, using the pragmatic testing set. Test A1 tests the three models trained with samples without glare removal and color correction. Unfortunately, the Top-1 recognition rates achieved by the three models are below 16% in this test, indicating that these models trained only with samples of uniform variations have very weak resistance to various glare reflections and color distortions. Moreover, DenseNet performs worse than ResNet in Test A1, implying that estimating its larger number of model parameters from only samples of uniform variations acquires poorer generalization ability. The proposed ResDenseNet, which combines ResNet and DenseNet, achieves a recognition rate that situates between the two individual models. This outcome can be attributed to the ResNet's remedies to the DenseNet's weakness to some extent.

When testing samples undergoing glare removal in Test A2, we find that the Top-1 recognition rates ranged from 10.98% to 15.42%, which are even lower than the recognition rates evaluated in Test A1. This result indicates that our glare removal method may occasionally repair some normal pixels with improper pixels, e.g. color-distorted pixels, from other images. Consequently, some samples correctly recognized in Test A1 are misclassified, thereby causing a decreased recognition rate in Test A2. To rectify this, applying color correction after glare removal can eliminate the side effects and improve the recognition rate.

In Test A3, applying color correction to the test samples significantly reduces color variability across samples, leading to a substantial increase in Top-1 recognition rates of the three models to 80.65%, 83.76%, and 89.21%, respectively. These results underscore the critical role of color in medication identification. Since both the testing set and the training set now exhibit a comparable level of diversity in appearance changes, DenseNet121 quickly outperforms ResNet101 due to its better integration of cross-level features. As a result, ResDenseNet also begins to extract superior cross-level features through the DenseNet121 and experiences a significant increase in recognition accuracy, making its recognition rate the highest among the three models.

Test A4 performs both glare removal and color correction on the test samples and all three models show further improved Top-1 recognition rates, ranging from 89.33% to 96.85%. ResDenseNet remains in the leading position. Table 5 shows that all three models perform better in Test A4 than in Test A5, implying that the glare removal and color correction enable a model trained with samples of uniform variations to outperform a model trained with samples of diverse variations.

We also test the three models on the challenge dataset, mentioned in Section IV-A, to assess their adaptability to different imaging conditions. Using this dataset, we repeat the tests of the above five options for each model. These five tests are identified as B1∼B5 in Table 5, respectively. The achieved rates shown in Test B1 again indicate that the models cannot well handle interferences not present in the training dataset because the samples do not undergo

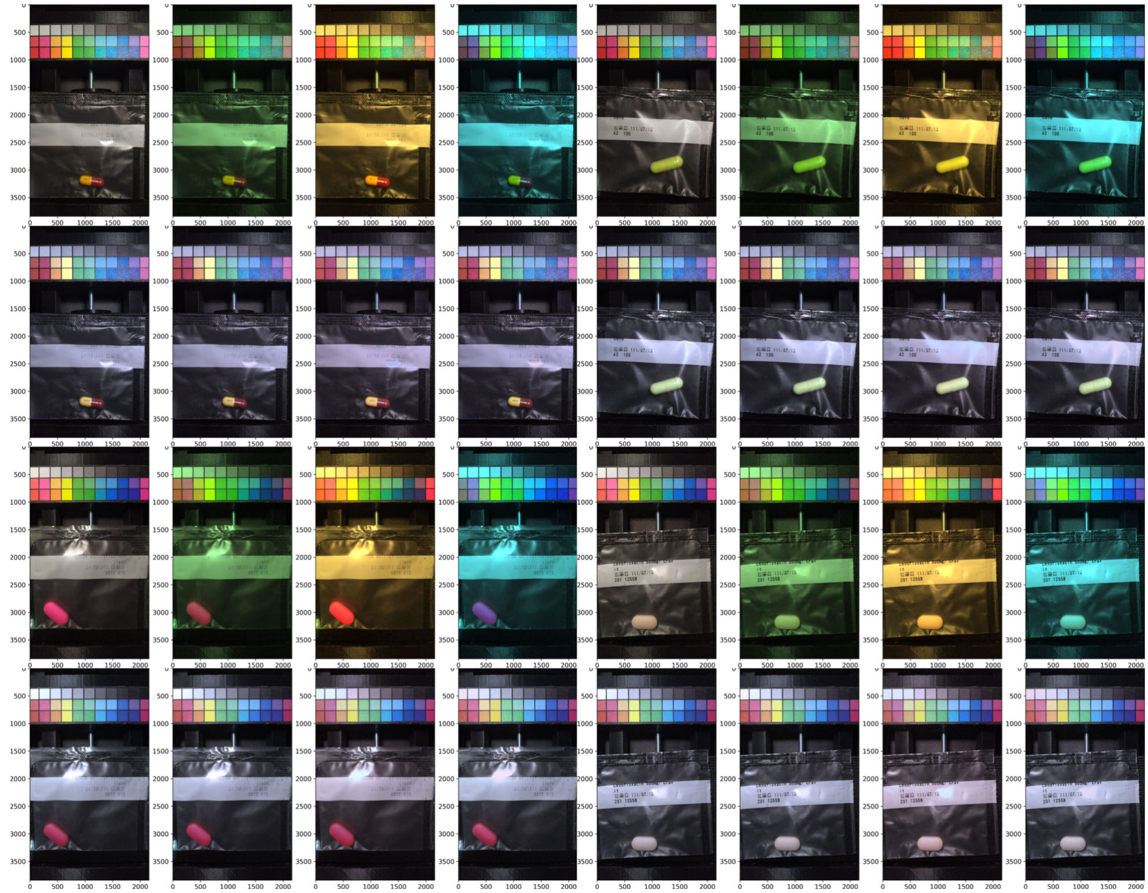


FIGURE 13. Some results of color correction. Odd rows show input image samples, and even rows show the results of color correction,

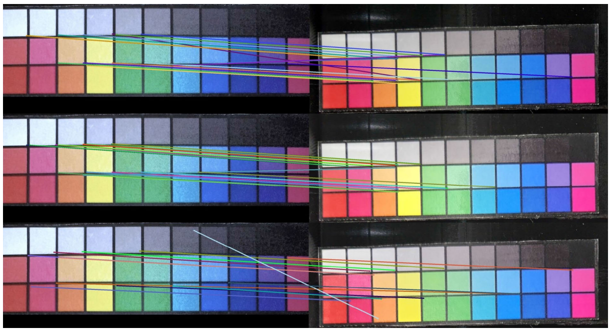


FIGURE 14. The results of color checker registration in color correction for three example cases. Each pair of matched corresponding points is connected with a colored line.

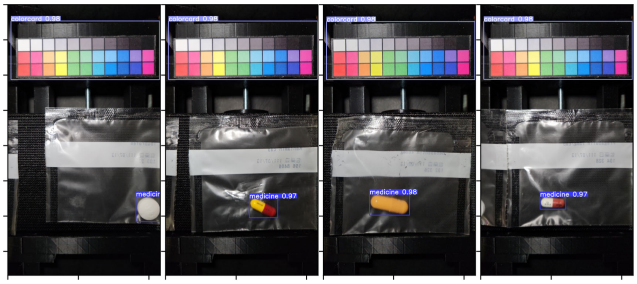


FIGURE 15. The results of YOLOv5's detection for four example cases.

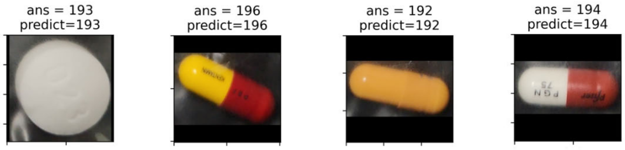


FIGURE 16. The identification results for the four example cases given in Fig. 15 using ResDenseNet.

glare removal and color correction. However, in Test B4, it becomes evident that applying our proposed preprocessing methods still significantly improves the ResDenseNet's Top-1 recognition rates from 15.65% in Test B1 to 82.83%. On the contrary, as shown in the results of Test B5, the models trained with diverse samples attain Top-1 recognition rates of merely around 34.23∼35.86%, indicating that their adaptability to new imaging conditions cannot be improved solely by data-driven training. These results verify that our proposed glare removal and color correction methods significantly improve each model's adaptability even though the model is trained with only a dataset that encompasses uniform appearance variations.

TABLE 5. Evaluated Top-1, Top-3, and Top-5 recognition rates of different recognition models using different testing options.

Testing Dataset		*Pragmatic Testing Set*					*Challenge Testing Set*				
Glare Removal		no	yes	no	yes	no	no	yes	no	yes	no
Color Correction		no	no	yes	yes	no	no	no	yes	yes	no
Training Dataset		*Pragmatic Training Set*					*Pragmatic Training Set*				
Glare Removal		yes	yes	yes	yes	no	yes	yes	yes	yes	no
Color Correction		yes	yes	yes	yes	no	yes	yes	yes	yes	no
ResNet	Top-1 Recog. Rate	**15.93**	**15.42**	80.65	89.33	88.43	**16.94**	**16.36**	64.95	78.50	34.23
DenseNet		11.68	10.98	83.76	93.57	91.08	16.12	15.30	68.57	79.56	35.16
ResDenseNet		15.19	14.60	**89.21**	**96.85**	**91.98**	15.65	14.95	**73.13**	**82.83**	**35.86**
ResNet	Top-3 Recog. Rate	**29.89**	**28.74**	95.60	98.44	97.98	**29.21**	**27.57**	85.98	93.81	58.29
DenseNet		20.99	19.86	96.87	99.42	98.68	28.39	24.18	86.33	94.39	59.23
ResDenseNet		26.17	24.81	**97.62**	**99.81**	**98.99**	26.64	23.71	**90.65**	**96.50**	**60.16**
ResNet	Top-5 Recog. Rate	**36.45**	**35.36**	98.68	99.38	99.61	**35.63**	**33.88**	94.04	96.96	69.86
DenseNet		25.97	24.81	98.21	99.81	99.65	30.61	27.57	93.93	97.08	70.79
ResDenseNet		32.83	31.46	**98.93**	**99.96**	**99.73**	31.66	28.97	**94.51**	**97.43**	**72.90**
Test Ientifier		A1	A2	A3	A4	A5	B1	B2	B3	B4	B5

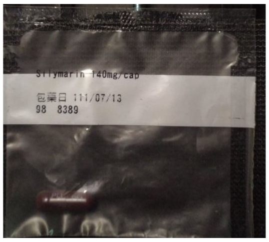


FIGURE 17. The case missed by YOLOv5's detection.

E. DISCUSSIONS

The experimental results indicate that our glare removal technique is notably effective when applied to images of medication packages taken under our designed imaging apparatus. The key to the technique's success is utilizing two distinct images captured from closely situated positions, specifically 0.5mm to 1cm apart. This spatial arrangement was empirically determined to minimize the likelihood of glare reflections affecting the same regions in both images and to maintain a sufficient overlap between the two images for stable registration, thereby enabling the method to repair glared pixels in one image using the non-glared pixels from the other. Our other findings reveal that utilizing a color checker with regular grid patterns in the imaging setup significantly improves the stability of image registration, especially under the controlled environment that our imaging apparatus offers. Since the transparent medication packages would not obscure the color checker under the operation of the imaging apparatus, its grid patterns remain consistently visible, thereby contributing to stable registrations. The correction results maintain a remarkable level of color constancy across all test samples. This can be primarily attributed to a good color correction function estimated from the stable and accurate registration between the reference color checker and the captured ones. Again, this is facilitated by the color checker's regular grid patterns which also can reduce the demand for a sophisticated method for feature detection.

As to medication detection, we have some findings from the experimental results.

- Focusing the YOLOv5's detection solely on two primary object categories, namely, the medication and the color checker, reduces the complexity of distinguishing subtle features among many different types of medications, thereby enhancing its detection accuracy.
- Pre-training YOLOv5 using the pretraining dataset is crucial in bolstering its ability to manage various appearance variations effectively. Without such pretraining, the detection accuracy of YOLOv5 would not achieve the elevated levels attained in our experiments.
- Our glare removal feature can mitigate false positives caused by glare interference, effectively addressing a limitation on YOLOv5's data-driven learning.

Analyzing the results of medication identification, our observations are itemized as follows.

- The recognition rate of ResDenseNet soars from 15.19% in Test A1 to 89.21% in Test A3. This improvement underscores the pivotal role of color fidelity in medication identification tasks, thereby emphasizing the importance of the color correction method in enhancing medication recognition.
- The results of Tests A3, A4, B3, and B4 show that color correction plays a crucial role in boosting the performance of ResNet and DenseNet, thereby facilitating the effective integration of both networks in our ResDenseNet to improve performance further.

- Applying glare removal and color correction in Test B4 significantly improved the Top-1 recognition rates to around 78.5% $\sim$ 82.83%. Interestingly, relying solely on data-driven training did not improve adaptability, as shown in Test B5, with rates only reaching around 34.23 $\sim$ 35.86%. This finding reiterates that our proposed preprocessing techniques are instrumental in enhancing model adaptability beyond what can be achieved through diversifying the dataset alone.
- The attained Top-5 recognition rate of our system is as high as 99.96%, implying that our system also has good potential to serve as an excellent medication retrieval system for searching similar medications through image queries.
- Suppose that a further improvement on the Top-1 recognition rate (96.85%) is required. A possible approach is to concentrate solely on re-ranking the top five candidates of each recognition outcome, disregarding the other 213 types of medications, through another stage of a finer-grained identification process.

Regarding the execution speed of this system, we use the system to recognize 218 images and count the processing time for each processing stage. We have tabulated the average percentage of total processing time taken by each processing stage, as shown in Table 6. According to the tabulated data, our system recognizes about one image per second. Though the recognition speed is not very impressive, it still matches the speed of the conveyor's mechanical transportation and wireless transmission of images on our automatic imaging apparatus. The data from the table indicates that color correction and glare removal take up most of the processing time. These two operations not only require feature point detection and matching but also necessitate updating all pixels in the entire image. Moreover, they cannot benefit from GPU acceleration, making them more time-consuming. To assess the computation costs of the three recognition models, we use the THOP package to count the computational operations in ResNet, DenseNet, and ResDenseNet models, obtaining 4.22G, 5.91G, and 17.53G in terms of MACs (Multiply-Accumulate Operations), respectively. The higher computation costs of ResDenseNet come from the independent operations performed in ResNet and DenseNet backbones. To reduce the costs, inventing an architecture to facilitate the sharing of neural layers between the two backbones could be a possible solution.

While the current system establishes a robust foundation for AMI, it has limitations. These include mutual occlusions of medications in a package, which can compromise detection and identification accuracy, and the need for adaptability to accommodate new types of medications, which requires retraining all deep learning models. Another limitation lies in detecting medications whose colors closely match the background, as observed in our YOLOv5 experiments. Addressing these limitations will be essential for refining the system's capabilities and ensuring its adaptability and reliability in diverse conditions.

TABLE 6. The percentage of time taken by each processing stage.

Stage	Time Taken (seconds)				Percentage
	ResNet	DenseNet	ResDenseNet	Total	
Glare Removal	111.00	108.38	111.316	330.7	41.7%
YOLOv5 Detection	11.71	10.63	10.65	32.99	4.2%
Color Correction	138.21	136.82	138.29	413.32	52.2%
ResDenseNet Identification	3.93	4.45	6.75	15.13	1.9%
Total	264.85	260.28	267.01		

V. CONCLUDING REMARKS

In this paper, we present solutions to address the challenges of glare reflection and color distortion in identifying medications with transparent packaging. Our solutions incorporate two preprocessing methods: one for glare reflection removal and the other for color distortion correction. Both methods employ image registration techniques to establish locational correspondences of pixels across multiple images.

Through extensive experimentation, we have demonstrated that our preprocessing methods allow deep learning models trained with only a dataset of preprocessed samples to achieve recognition rates comparable or even superior to those trained with datasets encompassing diverse appearance variations. Furthermore, our proposed preprocessing methods enable the models to better adapt to varying imaging conditions.

For our medication identification module, we propose the ResDenseNet architecture to combine the strengths of the ResNet and DenseNet networks, specifically their ability to integrate cross-level features through skip connections. As a result, our ResDenseNet can better distinguish medications, achieving a higher recognition rate than either of the individual networks.

Aiming at removing the limitations of the current system, future work will focus on implementing advanced techniques such as generative deep learning models to recover missing appearances of medications due to mutual occlusion, few-shot learning to adapt the system to new types of medications with limited labeled data, and the utilization of multiple lighting sources with distinct color tones to increase color differences on the different surface materials of medications and the background. We believe these forthcoming advancements will significantly enhance the system's accuracy and adaptability, making it more viable and deployable in real-world scenarios.

ACKNOWLEDGMENT

The authors would like to thank the Pharmacy Department Staff with Taiwan's Mennonite Christian Hospital for their invaluable assistance in providing medication samples with transparent packaging and related consultations, which greatly facilitated the smooth progress of this study.

REFERENCES

[1] N. L. Delgado, N. Usuyama, A. K. Hall, R. J. Hazen, M. Ma, S. Sahu, and J. Lundin, "Fast and accurate medication identification," *NPJ Digit. Med.*, vol. 2, no. 1, p. 10, Feb. 2019.

[2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[3] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[5] W.-J. Chang, L.-B. Chen, C.-H. Hsu, J.-H. Chen, T.-C. Yang, and C.-P. Lin, "MedGlasses: A wearable smart-glasses-based drug pill recognition system using deep learning for visually impaired chronic patients," *IEEE Access*, vol. 8, pp. 17013–17024, 2020.

[6] P.-T. Wu, T.-Y. Sun, J.-C. Lin, and C.-L. Chin, "Round pill shape recognition system based on AY deep learning model," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan*, Jul. 2022, pp. 553–554.

[7] H.-W. Ting, S.-L. Chung, C.-F. Chen, H.-Y. Chiu, and Y.-W. Hsieh, "A drug identification model developed using deep learning technologies: Experience of a medical center in Taiwan," *BMC Health Services Res.*, vol. 20, no. 1, pp. 1–9, Dec. 2020.

[8] C.-W. Chen, A.-C. Tsai, W.-Y. Chang, H.-F. Huang, and J.-F. Wang, "Arbitrary axis-aligned and multi-scale drug recognition system," in *Proc. 8th Int. Conf. Orange Technol. (ICOT)*, Dec. 2020, pp. 1–4.

[9] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[10] J. Heo, Y. Kang, S. Lee, D.-H. Jeong, and K.-M. Kim, "An accurate deep learning–based system for automatic pill identification: Model development and validation," *J. Med. Internet Res.*, vol. 25, Jan. 2023, Art. no. e41043.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 770–778.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[13] Y. Shih, D. Krishnan, F. Durand, and W. T. Freeman, "Reflection removal using ghosting cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3193–3201.

[14] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, "Depth from shading, defocus, and correspondence using light-field angular coherence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1940–1948.

[15] R. Wan, B. Shi, T. A. Hwee, and A. C. Kot, "Depth of field guided reflection removal," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 21–25.

[16] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *Proc. ACM SIGGRAPH Papers (SIGGRAPH)*, 2006, pp. 787–794.

[17] X. Fu, Y. Sun, M. LiWang, Y. Huang, X.-P. Zhang, and X. Ding, "A novel retinex based approach for image enhancement with illumination adjustment," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 1190–1194.

[18] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "A generic deep architecture for single image reflection removal and image smoothing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3258–3267.

[19] R. Wan, B. Shi, H. Li, L.-Y. Duan, A.-H. Tan, and A. C. Kot, "CoRRN: Cooperative reflection removal network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 12, pp. 2969–2982, Dec. 2020.

[20] R. Wan, B. Shi, L.-Y. Duan, A.-H. Tan, and A. C. Kot, "CRRN: Multi-scale guided concurrent reflection removal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4777–4785.

[21] X. Zhang, R. Ng, and Q. Chen, "Single image reflection separation with perceptual losses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4786–4794.

[22] R. Abiko and M. Ikehara, "Single image reflection removal based on GAN with gradient constraint," *IEEE Access*, vol. 7, pp. 148790–148799, 2019.

[23] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Proc. Deep Learn. Med. Image Anal. (DLMIA) Workshop*, Jul. 2018, pp. 3–11.

[24] C. Li, Y. Yang, K. He, S. Lin, and J. E. Hopcroft, "Single image reflection removal through cascaded refinement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3562–3571.

[25] Y.-C. Liu, W.-H. Chan, and Y.-Q. Chen, "Automatic white balance for digital still camera," *IEEE Trans. Consum. Electron.*, vol. 41, no. 3, pp. 460–466, Aug. 1995.

[26] O. Patel, Y. P. S. Maravi, and S. Sharma, "A comparative study of histogram equalization based image enhancement techniques for brightness preservation and contrast enhancement," 2013, *arXiv:1311.4033*.

[27] J. Cepeda-Negrete and R. E. Sanchez-Yanez, "Gray-world assumption on perceptual color spaces," in *Image and Video Technology* (Lecture Notes in Computer Science), R. Klette, M. Rivera, and S. Satoh, Eds. Berlin, Germany: Springer, 2014, pp. 493–504.

[28] H. Niu, Q. Lu, and C. Wang, "Color correction based on histogram matching and polynomial regression for image stitching," in *Proc. IEEE 3rd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2018, pp. 257–261.

[29] C.-H. Hsu, Z.-W. Chen, and C.-C. Chiang, "Region-based color correction of images," in *Proc. 3rd Int. Conf. Inf. Technol. Appl. (ICITA)*, Jul. 2005, pp. 710–715.

[30] V. Senthilkumaran, "Color correction using color checkers," in *Proc. 1st Int. Conf. Combinat. Optim., ICCAP, December 7–8, Chennai, India*, Dec. 2021, pp. 1–9.

[31] Z. Lou, T. Gevers, N. Hu, and M. P. Lucassen, "Color constancy by deep learning," in *Proc. Brit. Mach. Vis. Conf.*, 2015, p. 76.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[33] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, *arXiv:1506.01497*.

[35] P. Purkait, C. Zhao, and C. Zach, "SPP-Net: Deep absolute pose regression with synthetic views," 2017, *arXiv:1712.03452*.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[37] W. Liu, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 21–37.

[38] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5244–5252.

[39] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[40] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

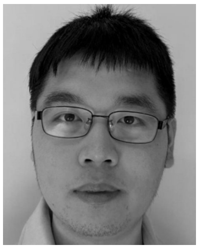
[41] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[42] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

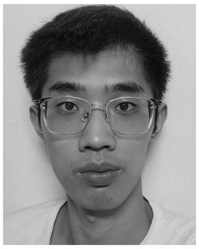
[43] J. C. A. Barata and M. S. Hussein, "The Moore–Penrose pseudoinverse: A tutorial review of the theory," *Brazilian J. Phys.*, vol. 42, nos. 1–2, pp. 146–165, Apr. 2012.

CHENG-CHIN CHIANG (Member, IEEE) received the B.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1986, and the M.S. and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1988 and 1993, respectively.

He was a Researcher with the Computer and Communication Research Laboratory, Advanced Technology Center, Industrial Technology Research Institute, Taiwan, from 1993 to 2000, participating in developing cutting-edge technologies in handwriting Chinese character recognition, image retrieval, and virtual reality. He obtained many patents that were transferred to well-known companies in the industry. In 2000, he joined the Department of Computer Science and Information Engineering, National Dong Hwa University, officially beginning his academic research and teaching career. More than 20 years of service at this university, he has held various positions, including the Director of the Cloud Computing and Digital Content Center, the Department Head of Computer Science and Information Engineering, and the Vice Dean of the College of Science and Engineering. He is currently the Executive Director of the Artificial Intelligence Office and funds the Machine Intelligence Technology Laboratory (MiT Laboratory), Department of Computer Science and Information Engineering. His research interests include deep learning, pattern recognition, multimedia systems, and human–machine interactions. He has received several prestigious awards for his work, including the Long-Term Doctorial Thesis Award from Acer Corporation, in 1993, the Outstanding Research Achievements Award from ITRI, in 1996, and the Outstanding Young Engineer Award from the Chinese Institute of Engineers, in 2000.



YU-YU YANG is currently pursuing the bachelor’s degree in computer science and information engineering with National Dong Hwa University, Hualien, Taiwan. His research interests include machine learning, deep learning, and reinforcement learning. He has published several papers in these fields. In May 2023, he was awarded first place in the Undergraduate Project Competition held by the Department of CSIE, National Dong Hwa University, for his graduation project. Additionally, starting in August 2023, he initiated an AI research project that was approved and funded by the National Science and Technology Council of Taiwan.



WEI-LIN LIU is currently pursuing the bachelor’s degree in computer science and information engineering with National Dong Hwa University, Taiwan. He is also a Research Assistant with the Machine Intelligence Technology Laboratory, where he has been actively involved in exploring the potential of cutting-edge AI technologies in real-world applications. His research interests include deep learning, artificial intelligence, image processing, and computer vision. He possesses extensive experience in advanced image processing and has contributed significantly to developing preprocessing methods used in image recognition applications. He is expected to complete the bachelor’s degree, in 2024, and is enthusiastic about continuing his research in the field of AI to develop innovative solutions that can benefit society.

YI-CHENG LIN is currently pursuing the B.S. degree with the Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan. His research interests include the IoT technology, AI applications, mobile software development, and mechanical hardware design. From 2022 to 2023, he was a Research Assistant with National Dong Hwa University, where he won first place in the Independent Study Project Competition of the Department of Information Engineering, in 2023, in recognition of his research abilities. Moreover, he was selected for the Youth Link Special Topic Grant Program by the Hualien County Government, in 2023.