IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 27, 2020, accepted October 5, 2020, date of publication October 8, 2020, date of current version October 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029526

Deep Feature-Based Three-Stage Detection of Banknotes and Coins for Assisting Visually Impaired People

CHANHUM PARK, SE WOON CHO, NA RAE BAEK, JIHO CHOI, AND KANG RYOUNG PARK, (Member, IEEE)
Division of Electronics and Electrical Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Kang Ryoung Park (parkgr@dongguk.edu)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (MSIT) through the Basic Science Research Program under Grant NRF-2019R1A2C1083813, in part by the NRF funded by the MSIT through the Basic Science Research Program under Grant NRF-2019R1F1A1041123, and in part by the NRF funded by the MSIT through the Basic Science Research Program under Grant NRF-2020R1A2C1006179.

ABSTRACT Owing to the rapid advancements in smartphone technology, there is an emerging need for a technology that can detect banknotes and coins to assist visually impaired people using the cameras embedded in smartphones. Previous studies have mostly used handcrafted feature-based methods, such as scale-invariant feature transform or speeded-up robust features, which cannot produce robust detection results for banknotes or coins captured in various backgrounds and environments. With the recent advancement in deep learning technology, some studies have been conducted on banknote and coin detection using a deep convolutional neural network (CNN). However, these studies also showed degraded performance depending on the changes in background and environment. To overcome these drawbacks, this paper proposes a three-stage detection technology for new banknotes and coins by applying faster region-based CNN, geometric constraints, and the residual network (ResNet). In the experiment performed using the open database of Jordanian dinar (JOD) and 6,400 images of eight types of Korean won banknotes and coins obtained using our smartphones, the proposed method exhibited a better detection performance than the state-of-the-art methods based on handcrafted features and deep features.

INDEX TERMS Smartphone camera, banknote and coin detection, faster R-CNN, geometric constraints, ResNet.

I. INTRODUCTION

With the rapid advancements in technology, smartphone has been widely used in various applications. As one of them, there is an emerging need for a technology that can detect banknotes and coins to assist visually impaired people using the cameras embedded in smartphones [1], [2].

In previous studies on banknote detection, high detection performance was observed by applying speeded-up robust features (SURF) to the banknotes [3]. However, the performance of SURF was significantly degraded when the images captured in complicated and diverse backgrounds were used [4]. In other research [5], the classification of fake banknote using deep learning was proposed, which did not require the pre-classification of banknote images in the denomination and input direction. However, the regions of banknotes were manually segmented from the input image, which requires user's assistance to use this method in actual smartphone. In addition, most previous studies on banknote detection using deep learning have used databases with simple backgrounds or with the application of a slight rotation such that the objects can be easily recognized. Thus, the studies that examine the detection performance using the images captured in various conditions are lacking [6], [7].

Therefore, the problem definitions can be as follows; the difficulties of automatic detection of banknotes in complicated background and lacking of performance evaluations in various experimental conditions. Moreover, considering that coins are commonly used in everyday lives, small-sized coins in addition to banknote should be also regarded as detection classes different from the most previous works.

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

Based on these research motivations, our research objective is the accurate banknote and coin detection and recognition in complicated backgrounds and various experimental conditions. Our research has the following contributions, significances, and advantages compared with the previous works:

- This study is the first deep learning-based approach to detect and recognize bills and coins with images captured by smartphone cameras in complicated background and various experimental conditions for assisting visually impaired people. Different from the most previous works, small-sized coins are also regarded as detection classes in our research.
- To improve the detection performance by VGG-16-based faster region-based convolutional neural network (Faster R-CNN) of the first stage of detection, false positive (FP) candidates are removed by applying post processing of the second stage based on the three features: the width-to-height ratio, detection box size, and detection score.
- The candidates remaining after post processing are divided into coin, bill, and coin and bill sections according to the detection box size; for coin and bill candidates, the verification of the third stage of detection is performed using ResNet-18-based Faster R-CNN to detect the final banknote region.
- Experimental results in various conditions and backgrounds confirm that our method outperforms the state-of-the-art methods for banknote detection. In addition, the self-collected Dongguk Korean Banknote database version 1 (DKB v1) and the developed models with algorithms are disclosed for a fair evaluation by other researchers as shown in [8].

This paper is organized as follows. In Section II, related works are described, and the proposed method is introduced in Section III. In Sections IV and V, we present the experimental results with analysis, and conclusions, respectively.

II. RELATED WORKS

Previous studies on banknote detection can be largely classified into handcrafted feature-based and deep feature-based methods. Several handcrafted feature-based studies examined detection and recognition methods using SURF. SURF was used mainly because it is effective for images with rotation or scaling changes. Compared with other handcrafted feature models, this method requires lower computational costs, which leads to a shorter time for the localization or matching of features. The detection method with SURF involves extracting features within images using the Hessian matrix [3]. Subsequently, approximately 20 images per class are obtained under various conditions, and then the features are matched to verify the detection performance. In addition to this handcrafted feature-based extraction method using entire images, some studies examined more efficient ways of using SURF based on foreground segmentation [1], [2]. These studies distinguished the banknote and background within images using a pixel-based adaptive segmenter (PBAS). Distinguishing foreground and background not only reduces the computation cost and improves the computation speed but also does not extract features from unnecessary areas in the images except the banknote. Moreover, the number of false matches in the recognition results can be reduced if features are not extracted from unnecessary areas. Adaptive boosting (AdaBoost), which is a widely used algorithm [9], involves transforming a weak classifier into a strong classifier through repeated training.

In a previous study [10], banknote recognition was conducted using AdaBoost and SURF-based methods. However, detection and classification processes using SURF are inefficient for finding the desired objects in images with complicated backgrounds or environments. The reason is that the SURF algorithm involves transforming color images into grayscale images to find features; consequently, important information is lost when color images captured by smartphone cameras are used. Moreover, even when the object and background in color images are different, SURF transforms the image into grayscale and recognizes the object and background as similar features, which leads to false detection of objects. Another handcrafted feature-based method involves using the fast radial symmetry (FRS) transform [11], in which geometrical patterns are extracted. The FRS transform is a gradient-based interest operator. The number parts in banknotes became gradients when the FRS transform was applied to regions with radial symmetry. Then, the unique geometrical patterns of the number parts in the denominations of Mexican banknotes were extracted using the extracted gradients. Then, the extracted patterns were applied to the test banknote to classify to which denomination the respective banknote belongs [12].

Another study applied principal component analysis (PCA) [13] as a handcrafted feature-based method [14]. In this study [14], the region of interest (RoI) was extracted based on the number parts located to the left and right of the denominations. The optimal number of eigenimages was determined using the extracted RoIs. Here, a total of 24 eigenimages were obtained by applying six denominations and four RoIs. Subsequently, the banknotes were recognized by applying the Mahalanobis distance using the features extracted based on eigenimages as input. In addition, the studies on banknote recognition [15] were conducted by applying the k-nearest neighbors (KNN) classifier [16] and the decision tree classifier (DTC) [17]. In this study [15], features with RGB values of RB, RG, or GB were extracted for each denomination of Malaysian banknotes. The two algorithms, KNN and DTC, were applied to the three extracted features for training with 10-fold cross validation.

However, these handcrafted feature-based methods [1]–[3], [10], [12], [14], [15] demonstrate a degraded detection or classification performance for banknotes when the images have complicated backgrounds or are captured in various conditions, such as skewness, illumination, and folded

objects. Accordingly, deep feature-based methods have been proposed [6].

A deep CNN (DCNN) extracts features through fast computations using a convolution filter based on a large number of datasets. Unlike handcrafted feature-based methods, a DCNN learns both the object and the background and thus can detect objects more effectively using a large amount of information. Several previous studies focused on the recognition of banknotes for assisting the visually impaired. However, some DCNN models with outstanding recognition and detection performance such as Faster R-CNN and you-look-only-once (YOLO) [18] have numerous layers, which results in several parameters and computations, thus requiring high hardware specifications. In other words, these models are difficult to realize in a wearable device with low specifications. Thus, MobileNet with a relatively lower amount of computations is often used [19]. The amount of computations of MobileNet v1 is three times less than that of GoogleNet [20], and the number of parameters is only 60% of that of the latter. Furthermore, the number of parameters of MobileNet v1 is 30 times less than that of the visual geometry group (VGG)-16 and the amount of computations is approximately 3% of that of the latter.

A previous study [6] examined the recognition performance based on MobileNet for Indian banknotes with an accuracy of approximately 96.6%. Another study [7] performed banknote detection and recognition using a shallow CNN designed by the researchers. In this study, features are extracted through a newly designed CNN network. Then, the output feature map is divided into an S × S grid of cells based on the extracted features. Each cell contains the vector of bounding boxes and the information on class predictions. A grid cell handles prediction in each object in the image; each grid cell predicts the bounding box and class probabilities. Banknote detection and recognition are performed based on the results of these grid cells. In [21], they use YOLO-v3-based banknote detection and recognition method. They collected images of different denominations and augmented those images with different geometric and image transformations. Previous research proposed the currency recognition method based on deep learning, and compared the performances by Faster R-CNN, single shot multibox detector (SSD), and MobileNet [22].

Chowdhury *et al.* proposed the Indian banknote recognition method based on CNN [23]. In their method, the images of currency notes are checked whether they are Indian banknotes or not. In case of Indian banknote, the denomination is classified by k-NN and also classified by CNN. In other research [24], they adopted CNN model based on Alexnet architecture with Chilean bill data which were augmented by translation, rotation, scaling, brightness variation, and etc. They also proposed their method as automatically classifying bill. Jadnav *et al.* proposed the method of currency identification and forged banknote detection using deep learning [25]. They used Saudi and Indian currencies, and extracted features in depth and analyzed the banknote by using deep CNN. As the researches of object detection, previous research proposed YOLO v2 [26]. Using a multi-scale training method, the YOLOv2 model can be operated at varying sizes, providing an easy tradeoff between speed and accuracy.

However, these deep feature-based methods [6], [7], [21]–[26] demonstrate a degraded performance depending on the changes in the background and environment.

To overcome these drawbacks, this paper proposes a new three-stage detection technology for banknotes and coins by applying Faster R-CNN, geometric constraints, and residual network (ResNet). Moreover, previous handcrafted feature-based and deep feature-based methods consider only bills in the detection and recognition processes, but not coins. However, coins are commonly used in everyday lives; therefore, banknote detection and recognition were performed using a database of both coins and bills in this study.

The advantages and drawbacks of the proposed and previous methods are summarized in Table 1.

III. PROPOSED METHOD

A. OVERVIEW OF THE PROPOSED METHOD

The flowchart of the proposed banknote detection method is shown in Figure 1.

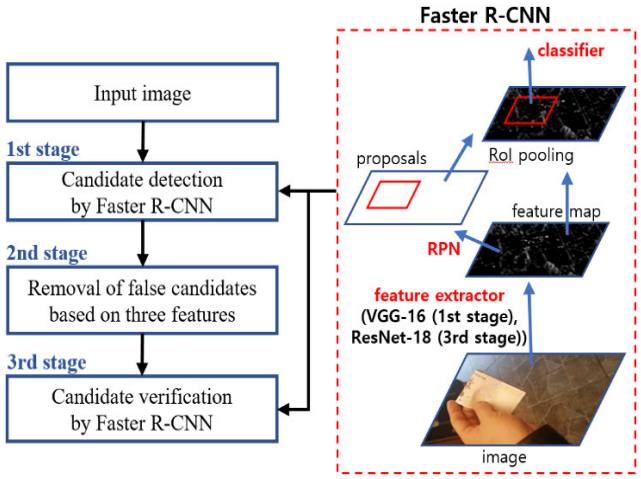


FIGURE 1. Flowchart of the proposed method.

First, the input images were applied to the pretrained Faster R-CNN as the experimental data (first stage). Among the box regions detected through Faster R-CNN, FP candidates were removed by performing post processing when there are multiple detection boxes in the image (second stage). In the post processing step, FP candidates were removed based on the box size, width-to-height ratio of the box, and detection score. Finally, the detection boxes that were not removed in the second stage are classified into coin, coin and bill, and bill candidates according to the detection box size. The coin and bill candidates were verified using the pretrained Faster R-CNN to detect the final banknote regions (third stage).

B. FIRST STAGE OF DETECTION

In the first stage of detection, banknotes were detected using Faster R-CNN. Starting with R-CNN [27], the speed was

TABLE 1. Summary of previous and proposed studies on banknote detection and recognition.

Category	Method	Strength	Weakness
Handcrafted feature based	PBAS + SURF [1,2]	- Can use images captured in various conditions - Fewer computations	Limited scale invariance performance
	Hessian matrix for SURF [3]	Can use images captured in various conditions	- Degraded performance for images with complicated backgrounds - Limited scale invariance performance
	AdaBoost + SURF [10]	- Detection and recognition possible in low illumination or dark environments - Applicable to sunglasses-type capturing devices	Degraded recognition performance for crumpled banknotes
	FRS transform [12]	Banknote detection is possible without separate training	- Uses images obtained from limited environments – Slow processing speed
	PCA [14], KNN + DTC [15]	- High recognition performance - Applicable using mobile devices	Limited regions for obtaining RoIs
Deep feature based	MobileNet [6]	Can be trained with a lightweight model without feature engineering or extensive preprocessing tasks	Uses images obtained from limited environments
	CNN network [7, 21-25]	Performs robust banknote recognition using real scene images with sunlight or artificial light	Did not perform the experiments with crumpled or difficult-to-detect banknotes
	Proposed three-stage method	- Uses images captured in various conditions and backgrounds - Includes banknotes and small-sized coins for experiments	Requires intensive training for two CNNs

improved compared to that of Fast R-CNN [28]. Ultimately, the detection performance was improved compared to that of Faster R-CNN, which has a faster detecting speed [29]. Accordingly, Faster R-CNN was used in this study as shown in Figure 1. VGG-16 [30] was used among the pretrained CNNs as a feature extractor, and the input image size (height × width × channel) was set to 600 × 800 × 3.

After extracting the feature map from VGG-16, region proposals were output through the region proposal network (RPN). RoI pooling was performed using the region proposals and feature map, and, subsequently, the results were used to classify objects and detect the bounding box. Tables 2–4 summarize the detailed architectures of the network used in this study. The feature extractor, whose architecture is summarized in Table 2, has 13 convolutional layers and rectified linear units (ReLUs), and four max pooling layers. In the existing VGG-16 network, the structure up to the last max pooling layer was used. The filter having the width × height of 3 × 3 is used in the convolutional layers. The numbers of paddings and strides are both 1 × 1 so that the size of the output feature map does not change when a convolutional layer is processed. A filter with the size of 2 × 2 × 1 is used in the max pooling layers. A stride of 2 × 2 was applied, and, consequently, the width and height of the output feature map of the convolutional layer were divided into two. The original image used in this study is an RGB image with the size of $1080 \times 1920 \times 3$. The size of the input image is $600 \times 800 \times 3$, which is converted through bilinear interpolation. Through the processes listed in Table 2, a feature map with the size of 38 × 50 × 512 was finally output. The feature map obtained thus is used as an input of the RPN and classifier.

Table 3 summarizes the information of the RPN network, which generates the region proposals. Conv5_3 output, which is the end of the previous feature extractor, is used as an input; it consists of a 3 × 3 convolutional layer and two 1 × 1 convolutional layers. In Conv6, a 3 × 3 window was slid on top of the pixels of the feature maps to perform convolution. There are nine anchor boxes at the center of the sliding window. The feature map extracted through Conv6 becomes the input to the classification and regression layers. The anchor boxes output the objects and background score in the classification layer, and the bounding box regression vector in the regression layer.

As shown in Equations (1) and (2), the bounding box regression vectors were used to transform the anchor boxes into proposal boxes.

$$v_x = \frac{x_{proposal} - x_{anchor}}{w_{anchor}}, \quad v_y = \frac{y_{proposal} - y_{anchor}}{h_{anchor}} \tag{1}$$

$$v_w = log\left(\frac{w_{proposal}}{w_{anchor}}\right), \quad v_h = log\left(\frac{h_{proposal}}{h_{anchor}}\right) \tag{2}$$

In Equations (1) and (2), $x_{proposal}$, $y_{proposal}$, $w_{proposal}$, and $h_{proposal}$ are the center x and y coordinates and the width and height of the proposal box, respectively. x_{anchor}, y_{anchor},

TABLE 2. Architecture of the feature extractor in Figure 1.

Layer type	Number of filters	Size of feature map (height × width × channel)	Kernel size (height × width × channel)	Number of strides	Number of paddings
Input layer [image]		600 × 800 × 3			
Conv1_1 (1st convolutional layer)	64	600 × 800 × 64	3 × 3 × 3	1 × 1	1 × 1
ReLU1_1		600 × 800 × 64			
Conv1_2 (2nd convolutional layer)	64	600 × 800 × 64	3 × 3 × 64	1 × 1	1 × 1
ReLU1_2		600 × 800 × 64			
Max pooling layer	1	300 × 400 × 64	2 × 2 × 1	2 × 2	0 × 0
Conv2_1 (3rd convolutional layer)	128	300 × 400 × 128	3 × 3 × 64	1 × 1	1 × 1
ReLU2_1		300 × 400 × 128			
Conv2_2 (4th convolutional layer)	128	300 × 400 × 128	3 × 3 × 128	1 × 1	1 × 1
ReLU2_2		300 × 400 × 128			
Max pooling layer	1	150 × 200 × 128	2 × 2 × 1	2 × 2	0 × 0
Conv3_1 (5th convolutional layer)	256	150 × 200 × 256	3 × 3 × 128	1 × 1	1 × 1
ReLU3_1		150 × 200 × 256			
Conv3_2 (6th convolutional layer)	256	150 × 200 × 256	3 × 3 × 256	1 × 1	1 × 1
ReLU3_2		150 × 200 × 256			
Conv3_3 (7th convolutional layer)	256	150 × 200 × 256	3 × 3 × 256	1 × 1	1 × 1
ReLU3_3		150 × 200 × 256			
Max pooling layer	1	75 × 100 × 256	2 × 2 × 1	2 × 2	0 × 0
Conv4_1 (8th convolutional layer)	512	75 × 100 × 512	3 × 3 × 256	1 × 1	1 × 1
ReLU4_1		75 × 100 × 512			
Conv4_2 (9th convolutional layer)	512	75 × 100 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU4_2		75 × 100 × 512			
Conv4_3 (10th convolutional layer)	512	75 × 100 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU4_3		75 × 100 × 512			
Max pooling layer	1	38 × 50 × 512	2 × 2 × 1	2 × 2	0 × 0
Conv5_1 (11th convolutional layer)	512	38 × 50 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU5_1		38 × 50 × 512			
Conv5_2 (12th convolutional layer)	512	38 × 50 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU5_2		38 × 50 × 512			
Conv5_3 (13th convolutional layer)	512	38 × 50 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU5_3		38 × 50 × 512			

TABLE 3. Architecture of the RPN in Figure 1.

Layer type	Number of filters	Size of feature map (height × width × channel)	Kernel size (height × width × channel)	Number of strides	Number of paddings
Input layer [Conv5_3]		38 × 50 × 512			
Conv6 (14th convolutional layer)	512	38 × 50 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU6		38 × 50 × 512			
Classification (convolutional layer)	18	38 × 50 × 18	1 × 1 × 512	1 × 1	0 × 0
Softmax		38 × 50 × 18			
Regression (convolutional layer)	36	38 × 50 × 36	1 × 1 × 512	1 × 1	0 × 0

w_{anchor}, and h_{anchor} are the center x and y coordinates and the width and height of the anchor box, respectively. Moreover, v_x, v_y, v_w, and v_h were obtained as the outputs of the regression convolution layer by training the RPN. The anchor box values were transformed into the proposal box values through the output of the regression layer.

Only the boxes above the intersection over union (IoU) threshold through non-maximum suppression (NMS) based on the score obtained from the classification layer remained as the proposal boxes. Here, the top 300 boxes become the region proposals and the input to the classifier in Table 4. The RPN and classifier were trained first during the training of Faster R-CNN. For the classification, nine classes (i.e., 10, 50, 100, 500, 1000, 5000, 10000, 50000 KRW, and background) were trained for KRW. Meanwhile, 10 classes (i.e., 1 qirsh, 5, 10 piastres, 1/4, 1/2, 1, 5, 10, 20 dinars, and background) were trained for JOD. When training the RPN and classifier, the weight was also trained so that the loss function value is minimized.

$$L\left(p_i, p_i^*, v_i, v_i^*\right) = \frac{\sum_i L_{cls}\left(p_i, p_i^*\right)}{N_{cls}} + \sigma \frac{\sum_i p^* L_{reg}\left(v_i, v_i^*\right)}{N_{reg}} \tag{3}$$

In Equation (3), i is the index of the mini-batch, p_i is the probability of whether anchor i is the object or background, and p_i^* is the ground-truth label, which has the value of 1 when the anchor is the banknote object or 0 when it is the background. L_{cls} is the classification loss function, which denotes the log loss of the class. v_i is the bounding box regression vector of the anchor box, which corresponds to Equations (1) and (2) mentioned above. v_i^* is the bounding box regression vector for the ground truth related to the respective class.

TABLE 4. Architecture of the classifier in Figure 1.

Layer type	Output size
Input layer [Conv 5_3] [region proposals]	 38 × 50 × 512 300 × 4
RoI pooling layer	7 × 7 × 512 × 300
Fc6 (1st fully connected layer) (ReLU) (Dropout)	4096 × 300
Fc7 (2nd fully connected layer) (ReLU) (Dropout)	4096 × 300
Classification (fully connected layer) (Softmax)	The number of classes × 300
Regression (fully connected layer)	4 × 300

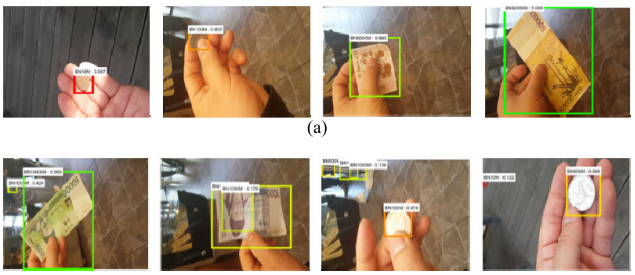


FIGURE 2. Examples of detected boxes by Faster R-CNN. (a) Correctly detected cases and (b) incorrectly detected cases.

L_{reg} is the smooth L1 loss function and is only used for positive anchors, or when $p_i^* = 1$. N_{cls} and N_{reg} are the mini-batch size and anchor locations, respectively, and the two loss functions are normalized. Accordingly, as the two loss functions, L_{cls} and L_{reg}, were trained, uniform balancing occurs through σ. When the two loss functions were minimized through balancing, the banknote location was detected, and the denomination of the detected banknote was distinguished.

Table 4 summarizes the architecture of the network, which detects objects using a feature map and the RPN. A feature map of 38 × 50 × 512, which is the output of VGG-16, and 300 region proposals of the RPN were used as the input. A fixed feature map of 7 × 7 was obtained in the RoI pooling layer. The newly extracted feature map was used to obtain a 4096-dimensional feature vector through the fully connected layers FC6 and FC7.

The respective vector becomes an input to the classification and regression layers, which used the probability of 9 classes for KRW and 10 classes for JOD. In the classification layer, each proposal was classified into multiple classes. In the regression layer, the proposal box was transformed into the predict box by outputting the bounding box regression vector. The final detection result is obtained through NMS.

C. SECOND STAGE OF DETECTION

In this study, all the cases of an object being detected from parts that are not banknote objects or other classes being detected from objects are counted as FPs. When a banknote object was not detected in the image, the case was counted as a false negative (FN). As shown in Figure 2(b), incorrectly detected boxes are present due to Faster R-CNN. Therefore, FPs were removed in the second stage of detection for improving the detection performance obtained with Faster R-CNN. Specifically, the FPs generated when the FN error is set to be minimized in the first stage of detection by Faster R-CNN were removed in the second stage of detection. The following three handcrafted feature-based post processing methods were applied to remove the FPs in the second stage of detection.

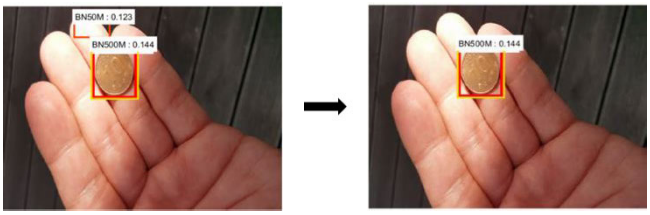


FIGURE 3. Examples of removing FPs based on the width-to-height ratio.

First, the FP box was removed by post processing according to the width-to-height ratio of the detected box. In general, the ground-truth box of coins is closer to a square than the ground-truth box of bills, which is a rectangle. The width-to-height ratio of coins in this study is close to 1; thus, a minimum value is not required. Using the width-to-height ratio of bills, the threshold range of the minimum and maximum ratios was determined using the training data. When the width-to-height ratio of the detection box is not within the threshold, the detection box was removed as an FP. Figure 3 shows an image in which FPs have been removed after the first post processing.

Some FPs may be present even after the first post processing. To remove these FPs, the second post processing method involves removing FPs based on the detection box size (the number of pixels in the detected box). As shown in Figure 4, the minimum and maximum ranges of the coin box size and bill box size observed in the training data were obtained. Then, based on the class label of the detected box, the box size was removed as an FP if it does not fall within the range of coin box size or bill box size. Figure 5 shows an image in which FPs have been removed after the second post processing.

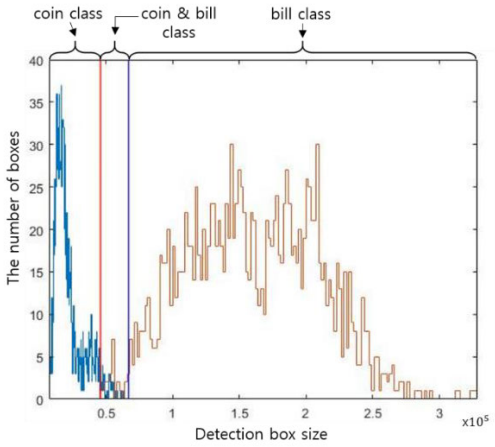


FIGURE 4. Distributions of detection box size of coin, coin and bill, and bill classes.

Some FPs may remain even after the second post processing. As shown in the left image of Figure 6, there may be two detection boxes in a banknote object. To address this problem, when the IoU of the detection boxes is 50% or greater, the third post processing method removed the remaining FPs by treating only the first ranked objects as true positives (TPs) based on the detection score obtained from Faster R-CNN. Here, TP refers to the case where the banknote in the input image has been correctly detected according to the corresponding class. This post processing was not applied to the candidates that have been detected as coins and was only applied to the candidates that have been detected as bills. The reason is that TP was removed along with coins when the FPs were removed based on the detection score of Faster R-CNN only, as the coin size is small. Figure 6 shows an image in which FPs have been removed after the third post processing.

D. THIRD STAGE OF DETECTION

As explained in Section III.C, the distributions of coin class (class 1), coin and bill class (class 2), and bill class (class 3) based on the detection box size with threshold information, shown in Figure 4, were obtained from the training data. Furthermore, a Faster R-CNN-based banknote detector, which uses the ResNet-18 as the feature extractor, was trained using the training data of each class.

Subsequently, it was determined to which class the data belong among the three classes (classes 1–3) in Figure 4 based on the detection box size obtained after the second stage of detection during the testing process. The third stage of detection was performed using the pretrained Faster R-CNN only for the class data belonging to class 1 or class 3. The training and testing performance can be improved if a separate Faster R-CNN is used as the form and size of class 1 and class 3 vary. However, the third stage of detection was not performed for class 2 because the training and testing performance of Faster R-CNN is not guaranteed for the data with mixed coins and bills. Therefore, in this study, the last fully connected layer in the ResNet-18 [31] pretrained with

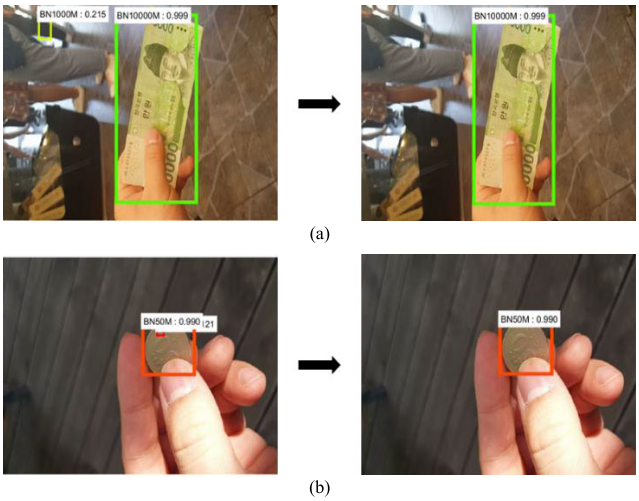


FIGURE 5. Examples of removing FPs based on the detection box size. The upper and lower images show the cases of bill and coin, respectively.

FIGURE 6. Examples of removing FPs based on detection score.

the ImageNet database [32] was removed to be used as the feature extractor of Faster R-CNN.

Table 5 shows the architecture of ResNet-18. The input image has the size of $224 \times 224 \times 3$ in the input layer of ResNet-18. In the first convolution, the output of size $112 \times 112 \times 64$ is produced through 64 filters with the size of $7 \times 7 \times 3$. A feature map of size $56 \times 56 \times 64$, which is half the aforementioned size, is obtained through max pooling. This feature map becomes the input to Conv2, and it then passes through four convolutional layers with the filter size of $3 \times 3 \times 64$. In Conv3, filters with the sizes of $3 \times 3 \times 128$, $3 \times 3 \times 64$, and $1 \times 1 \times 64$ were employed. The feature information of small objects such as coins can be delivered to the next layer by using the shortcut and residual block of ResNet. Similarly, the final feature map of size $7 \times 7 \times 512$ can be obtained through Conv3, Conv4, and Conv5. As shown in Figure 1, the result was input to the RPN and classifier to perform the final banknote detection. The architectures of the RPN and classifier are summarized in Tables 6 and 7, respectively.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL ENVIRONMENT

There is a lack of an open database of banknote images captured using smartphone cameras. Therefore, the experiment in this study was conducted using the Dongguk Korean Banknote database version 1 (DKB v1). The DKB v1 contains

TABLE 5. Architecture of ResNet-18.

Layer type	Number of filters	Size of feature map (height × width × channel)	Kernel size (height × width × channel)	Number of strides	Number of paddings
Input layer [image]		224 × 224 × 3			
Conv1	64	112 × 112 × 64	7 × 7 × 3	2 × 2	3 × 3
Max pooling layer	1	56 × 56 × 64	3 × 3 × 1	2 × 2	0 × 0
Conv2_1 (Res2a)	64	56 × 56 × 64	3 × 3 × 64	1 × 1	1 × 1
Conv2_2 (Res2a)	64	56 × 56 × 64	3 × 3 × 64	1 × 1	1 × 1
Conv2_3 (Res2b)	64	56 × 56 × 64	3 × 3 × 64	1 × 1	1 × 1
Conv2_4 (Res2b)	64	56 × 56 × 64	3 × 3 × 64	1 × 1	1 × 1
Conv3_1 (Res3a)	128	56 × 56 × 64	3 × 3 × 64	2 × 2	1 × 1
Conv3_2 (Res3a)	128	28 × 28 × 128	3 × 3 × 128	1 × 1	1 × 1
Conv3_3 (Res3a_shortcut)	128	28 × 28 × 64	1 × 1 × 64	2 × 2	0 × 0
Conv3_4 (Res3b)	128	28 × 28 × 128	3 × 3 × 128	1 × 1	1 × 1
Conv3_5 (Res3b)	128	28 × 28 × 128	3 × 3 × 128	1 × 1	1 × 1
Conv4_1 (Res4a)	256	28 × 28 × 128	3 × 3 × 128	2 × 2	1 × 1
Conv4_2 (Res4a)	256	14 × 14 × 256	3 × 3 × 256	1 × 1	1 × 1
Conv4_3 (Res4a_shortcut)	256	14 × 14 × 128	1 × 1 × 128	2 × 2	0 × 0
Conv4_4 (Res4b)	256	14 × 14 × 256	3 × 3 × 256	1 × 1	1 × 1
Conv4_5 (Res4b)	256	14 × 14 × 256	3 × 3 × 256	1 × 1	1 × 1
Conv5_1 (Res5a)	512	14 × 14 × 256	3 × 3 × 256	2 × 2	1 × 1
Conv5_2 (Res5a)	512	7 × 7 × 512	3 × 3 × 512	1 × 1	1 × 1
Conv5_3 (Res5a_shortcut)	512	7 × 7 × 256	1 × 1 × 256	2 × 2	0 × 0
Conv5_4 (Res5b)	512	7 × 7 × 512	3 × 3 × 512	1 × 1	1 × 1
Conv5_5 (Res5b)	512	7 × 7 × 512	3 × 3 × 512	1 × 1	1 × 1

TABLE 6. Architecture of the RPN.

Layer type	Number o f filters	Size of feature map (heigh t × width × channel)	Kernel size (height × w idth × channel)	Number of strides	Number of paddings
Input layer [Conv5_5]		7 × 7 × 512			
Conv6 (1^{4th} convolutional layer)	512	7 × 7 × 512	3 × 3 × 512	1 × 1	1 × 1
ReLU6		7 × 7 × 512			
Classification (convolutional layer)	18	7 × 7 × 18	1 × 1 × 512	1 × 1	0 × 0
Softmax		7 × 7 × 18			
Regression (convolutional layer)	36	7 × 7 × 36	1 × 1 × 512	1 × 1	0 × 0

eight classes, namely, 10, 50, 100, 500, 1000, 5000, 10000, and 50000 KRW, with each class having 800 images, yielding a total of 6,400 images. The images were captured using the frontal viewing camera of Galaxy Note 5 [33]. The images of the banknotes were captured from various distances. To reflect the real-world environment as closely as possible, the images were captured under conditions of various locations, lighting, and cases where the bills were randomly folded. The size of the obtained image is 1920 × 1080 pixels. Figure 7 shows the images in the DKB v1.

Furthermore, the experiment was conducted using the open database of JOD [34] to verify whether the proposed algorithm can be applied to various types of banknote images. The JOD open database contains nine classes (i.e., 1 qirsh, 5, 10 piastres, 1/4, 1/2, 1, 5, 10, 20 dinars), yielding a total of 330 images. The size of the obtained image is 3264 × 2448 pixels. Figure 8 shows the images in the open database of JOD. As shown in Figure 8(b), the open database of JOD includes several images of street quality such as bills being severely folded or occluded and with poor quality due to uneven lighting or light saturation. The algorithm proposed in this paper was trained and tested using a desktop computer equipped with Intel® Core™ i7-950 CPU@3.07GHz, 20 GB memory, and NVIDIA GeForce GTX1070 graphics (1920 compute unified device architecture (CUDA) cores) [35]. The algorithm was realized using MATLAB R2019a and R2017b version [36] based on compute unified device architecture (CUDA) (Version 10.0) [37] with the CUDA deep neural network library (cuDNN) (version 7.1.4) [38].

B. TRAINING OF THE PROPOSED METHOD

As explained in Section III.B, Faster R-CNN was used in the first stage of detection in this study. For end-to-end training, two-fold cross validation was performed by dividing 6,400 images of the DKB v1 into 3,200 images each for the training and testing (validation) sets. In other words, 3,200 images were used for the training, and the remaining 3,200 images were used for testing (validation) in the first fold. In the second fold, the two subsets for training and testing (validation) were switched. The average value of accuracy of the two tests was used as the final performance indicator. Faster R-CNN was trained with the four-step alternating training method [29]. By using the stochastic gradient descent [39], we determined the weight value that minimizes the difference

TABLE 7. Architecture of the classifier.

Layer type	Output size
Input layer	
[Conv 5_5]	7 × 7 × 512
[region proposals]	300 × 4
RoI pooling layer	7 × 7 × 512 × 300
Fc6 (1st fully connected layer) (ReLU) (Dropout)	4096 × 300
Fc7 (2nd fully connected layer) (ReLU) (Dropout)	4096 × 300
Classification (fully connected layer) (Softmax)	The number of classes × 300
Regression (fully connected layer)	4 × 300

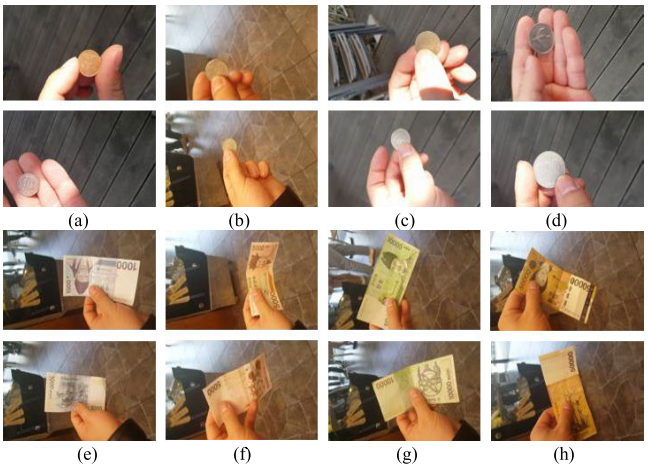


FIGURE 7. Examples of images in DKB v1. (a)–(h) show 10, 50, 100, 500, 1000, 5000, 10000, and 50000 KRW, respectively, with the top and bottom images showing the front and rear sides, respectively.

between the training result and the ground-truth value. The parameters used in this study for training are as follows: base learning rate = 0.001, batch size = 1, gamma = 0.1, momentum = 0.9, weight decay = 0.0005, and number of iterations = 120,000. Figure 9 shows the training loss graphs when the DKB v1 was used. Figure 10 shows the training loss graphs when the JOD open database was used. In both Figures 9 and 10, the training loss converged as the number of iterations increased, which indicates that Faster R-CNN was sufficiently trained in this study.

C. CLASS ACTIVATION MAP OF FEATURE USING THE PROPOSED METHOD

In this subsection, the features obtained through the proposed network are analyzed through the class activation map (CAM) of the third-stage detector of Figure 1 for the input images. The regions whose color is closed to red represent the important features extracted by our network. This kind of analysis with CAM images has been widely adopted in the researches of deep learning-based image processing and recognition [40]. CAMs were obtained from Conv1, Conv2_4, Conv3_5, Conv4_5, and Conv5_5 of the feature extractor in Table 5. As shown in Figure 11, the CAM was obtained by considering the images of coins and bills of the DKB v1 and JOD open database as the input.

FIGURE 8. Examples of images in JOD. Images of (a) good quality and (b) bad quality.

Figure 11 illustrates that more abstract CAM images were obtained as the convolutional layer became deeper. Both coins and bills had more highly activated results (marked in red) in the object region compared with the background. These results indicate that the proposed network generates feature maps from which banknotes can be easily detected.

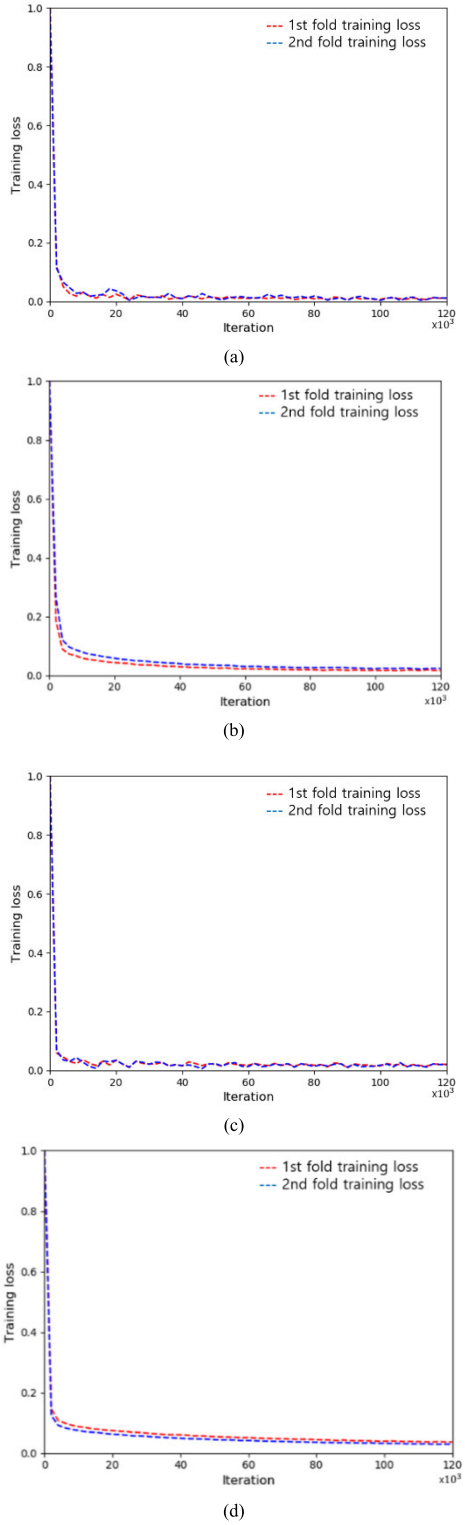


FIGURE 9. Training loss graphs of Faster R-CNN with DKB v1. (a) RPN training of stage-1, (b) classifier training of stage-1, (c) RPN training of stage-2, and (d) classifier training of stage-2.

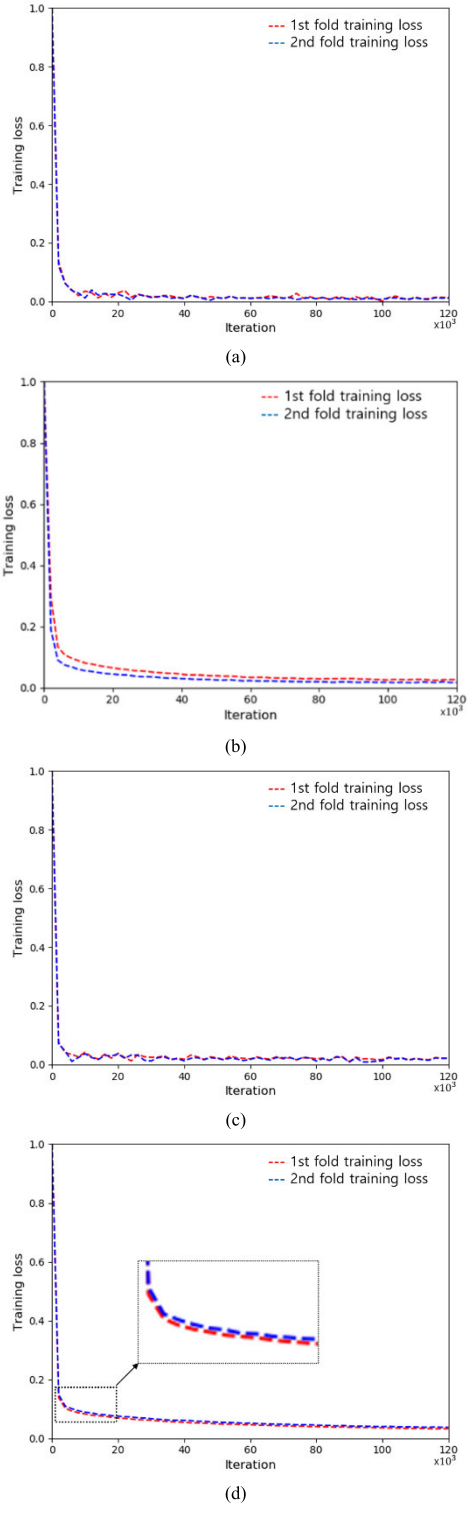


FIGURE 10. Training loss graphs of Faster R-CNN with JOD. (a) RPN training of stage-1, (b) classifier training of stage-1, (c) RPN training of stage-2, and (d) classifier training of stage-2.

D. TESTING OF THE PROPOSED METHOD (ABLATION STUDY)

1) FIRST STAGE OF DETECTION

TP, FP, and FN were computed based on the IoU value between the detected box and the ground-truth box using the proposed method to evaluate its testing performance. Based on these results, the detection accuracy was calculated by determining the recall, precision, and F1 score [41], [42]

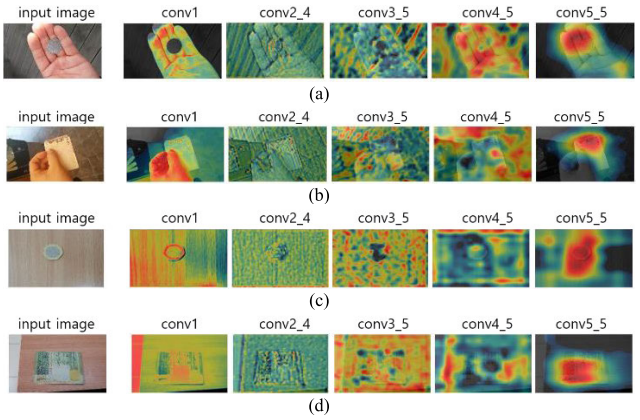


FIGURE 11. CAM images from Conv1, Conv2_4, Conv3_5, Conv4_5, and Conv5_5 of the feature extractor of the third-stage detector in the case of (a) coin of DKB v1, (b) bill of DKB v1, and (c) coin of JOD, (d) bill of JOD.

according to Equations (4)–(6).

$$\text{Precision} = \frac{P}{P+Q} \tag{4}$$

$$\text{Recall} = \frac{P}{P+R} \tag{5}$$

P, Q, and R represent the numbers of TPs, FPs, and FNs, respectively.

$$\text{F1 score} = \frac{2 \text{ precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

Table 8 lists the recall, precision, and F1 score values, as the output detection threshold of Faster R-CNN is adjusted from 0.1 to 0.5 in the DKB v1. As listed in Table 8, with the increase in threshold, the recall value decreased, whereas the precision value increased. The criteria for determining TP become stricter with an increase in the threshold; thus, R of Equation (5) increases, but Q of Equation (4) decreases. R is minimized even when Q increases in the first stage of detection, and, then, Q is reduced in the following step. Therefore, the output detection threshold of 0.1 with the highest recall value in Table 8 was used for Faster R-CNN.

TABLE 8. Accuracies of the first stage of detection with DKB v1 according to the increase in the output detection threshold of Faster R-CNN.

Output detection threshold	Recall	Precision	F1 score
0.1	**0.8553**	**0.8432**	**0.8492**
0.2	0.8399	0.9049	0.8711
0.3	0.8242	0.9283	0.8731
0.4	0.8175	0.9466	0.8773
0.5	0.8049	0.9623	0.8766

Table 9 lists the accuracies of the first stage of detection for the coin, bill, and coin and bill sub-datasets. As summarized in Table 9, the first stage of detection for bills has a higher accuracy because the object size is larger than that of coins and the features are more easily observed.

TABLE 9. Accuracies of the first stage of detection with DKB v1 in the case of coin, bill, and coin + bill sub-datasets.

Sub-dataset	Recall	Precision	F1 score
Coin	0.7221	0.6715	0.6959
Bill	0.9592	0.9315	0.9451
Coin + bill	**0.8553**	**0.8432**	**0.8492**

TABLE 10. Accuracies of the first stage of detection with JOD according to the increase in the output detection threshold of Faster R-CNN.

Output detection threshold	Recall	Precision	F1 score
0.1	**0.8294**	**0.8523**	**0.8407**
0.2	0.8112	0.8803	0.8443
0.3	0.8016	0.8993	0.8476
0.4	0.7935	0.9257	0.8545
0.5	0.7817	0.9447	0.8555

Table 10 lists the recall, precision, and F1 score values, as the output detection threshold of Faster R-CNN is adjusted from 0.1 to 0.5 in the JOD open database. As listed in Table 10, with the increase in the threshold, the recall value decreased whereas the precision value increased. As explained previously, R of Equation (5) is minimized even when Q of Equation (4) increases in the first stage of detection, and, then, Q of Equation (4) is reduced in the following step. Therefore, the output detection threshold of 0.1 with the highest recall value in Table 10 was used for Faster R-CNN.

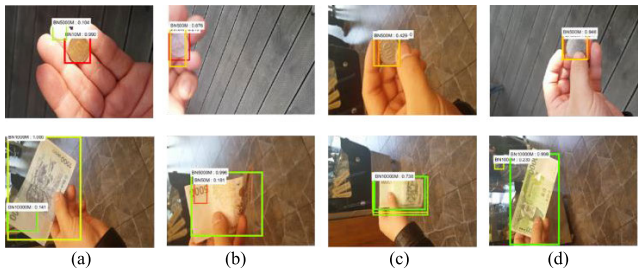


FIGURE 12. Results of the first stage of detection.

Figure 12 illustrates the result images of the first stage of detection. As shown in Figure 12, the result images contain several FPs in addition to TP. These FPs are removed during the second and third stages of detection.

2) SECOND STAGE OF DETECTION

The FPs present after the first stage of detection are removed sequentially based on the width-to-height ratio of the box (1st step), box size (2nd step), and detection score of Faster R-CNN (third step), as explained in Section III.C. Table 11 lists the accuracies of the second stage of detection per step for the DKB v1 when the first, second, and third steps are applied sequentially. As summarized in Table 11, the precision value and F1 score were the highest when steps 1–3 were applied. Moreover, the second post processing step applied to both

TABLE 11. Accuracies of the second stage of detection with DKB v1.

Method		Recall	Precision	F1 score
1st step	Coin + bill	0.8553	0.8472	0.8512
1st step + 2nd step	Coin	0.8553	0.8544	0.8548
	Bill	0.8553	0.8505	0.8529
	Coin + bill	0.8553	0.8602	0.8577
1st step + 2nd step + 3rd step	**Bill**	**0.8553**	**0.8747**	**0.8649**
	Coin + bill	0.8147	0.9355	0.8709

TABLE 12. Accuracies of the second stage of detection with DKB v1 in the case of coin, bill, and coin + bill sub-datasets.

Sub-dataset	Recall	Precision	F1 score
Coin	0.7221	0.7013	0.7115
Bill	0.9592	0.9817	0.9703
Coin + bill	**0.8553**	**0.8747**	**0.8649**

coins and bills resulted in a higher precision and F1 score for the same recall value, compared with when applied to either coins or bills. The accuracies of applying the third post processing step only to bills and to both coins and bills were compared, as presented in Table 11. The precision and F1 score for applying the third post processing step to both coins and bills were slightly higher than those when applying the third post processing step to only bills, but the recall value also decreased. The reason is that TP was removed along with coins when the FPs are removed based on the detection score of Faster R-CNN only, as the coin size is small. This study used a method where the FPs are removed in the subsequent step by maintaining the recall as high as possible in each step; therefore, the third post processing step is applied only to bills as explained in Section III.C.

Table 12 lists the accuracies of the second stage of detection for the coin, bill, and coin and bill sub-datasets in the DKB v1. As summarized in Table 12, the second stage of detection for bills has a higher accuracy because the object size is larger than that of coins and the features are more easily observed.

Table 13 lists the accuracies of the second stage of detection of Figure 1 per step for the JOD open database when the first, second, and third steps are applied sequentially. As summarized in Table 13, the precision value and F1 score were the highest when steps 1–3 were applied. Moreover, the second post processing step applied to both coins and bills resulted in a higher precision and F1 score for the same recall value, compared with when applied to either coins or bills. The accuracies of applying the third post processing step only to bills and to both coins and bills were compared, as summarized in Table 13. The precision and F1 score for applying the third post processing step to both coins and bills were slightly higher than those when applying the third post processing step to only bills, but the recall value also decreased. This study used a method where the FPs are removed in the subsequent step by maintaining the recall as high as possible in each step; therefore, the third post processing step is applied only to bills.

TABLE 13. Accuracies of the second stage of detection with JOD.

Method		Recall	Precision	F1 score
1st step	Coin + bill	0.8294	0.8553	0.8422
1st step + 2nd step	Coin	0.8294	0.8595	0.8442
	Bill	0.8294	0.8643	0.8465
	Coin + bill	0.8294	0.8717	0.8500
1st step + 2nd step + 3rd step	**Bill**	**0.8294**	**0.8823**	**0.8550**
	Coin + bill	0.7948	0.9381	0.8605

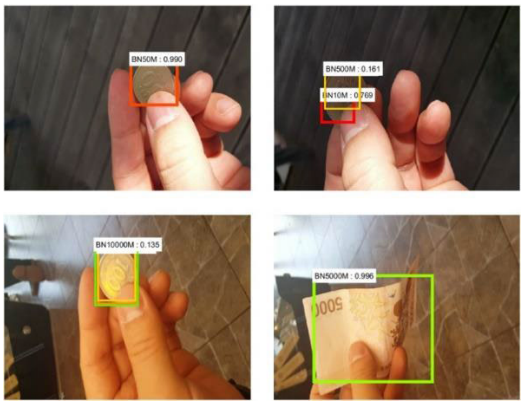


FIGURE 13. Results of the second stage of detection.

TABLE 14. Accuracies of the third stage of detection with DKB v1.

Sub-dataset	Recall	Precision	F1 score
Coin	0.9250	0.9548	0.9397
Bill	0.9769	0.9880	0.9824
Coin + bill	**0.9486**	**0.9721**	**0.96**

Figure 13 illustrates the result images of the second stage of detection. As shown in Figure 13, the result images contain FPs (yellow box in the upper right image and green box in the lower left image) in addition to TP. These FPs are removed in the third stage of detection.

3) THIRD STAGE OF DETECTION

The accuracy of the third stage of detection was computed. Table 14 lists the accuracies of the third stage of detection of Figure 1 for the coin, bill, and coin and bill sub-datasets in the DKB v1. As summarized in Table 14, the third stage of detection for bills has a higher accuracy because the object size is larger than that of coins and the features are more easily observed.

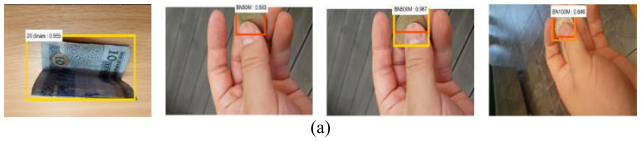
Table 15 list the accuracies of the third stage of detection of Figure 1 for the coin, bill, and coin & bill sub-datasets in the JOD open database. As summarized in Table 15, the third stage of detection for bills has a higher accuracy because the object size is larger than that of coins and the features are more easily observed.

Figure 14 shows examples of correct detection for all the three stages of detection performed using the proposed method. As shown in this figure, the correct detection results

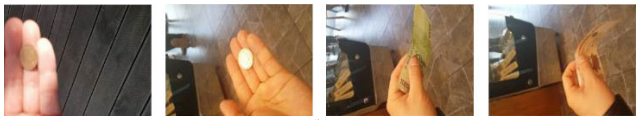
TABLE 15. Accuracies of the third stage of detection with JOD.

Sub-dataset	Recall	Precision	F1 score
Coin	0.9076	0.9211	0.9143
Bill	0.9512	0.9747	0.9628
Coin + bill	**0.9356**	**0.9604**	**0.9478**

FIGURE 14. Examples of correct detection using our method.



(a)



(b)

FIGURE 15. Examples of incorrect detection by our method. (a) FP cases and (b) FN cases.

were still obtained when banknotes were folded, some parts were occluded in the input image, or there was severe light saturation on the surface.

Figure 15 shows examples of incorrect detection for all the three stages of detection performed using the proposed method. As shown in this figure, the incorrect detection results were obtained when banknotes were severely folded, the object was slanted, some parts were not in the input image or were occluded, or there was severe light saturation on the surface.

By summary, our method produces the minimum number of FN (causing the higher recall of Equation (5)) in the first stage of Figure 1 although the number of FP is also increased (causing the lower precision of Equation (4)). The results are shown as the bold numbers of Tables 8 and 9 in case of DKB v1, and those of Table 10 in case of JOD. Then, the number of FP is further reduced by the second stage of Figure 1 (causing the higher precision of Equation (4) and F1 score of Equation (6) while the recall is maintained). The results are shown as the bold numbers of Tables 11 and 12 in case of DKB v1, and those of Table 13 in case of JOD. Finally, the images including only the candidates of classes 1 (coin) and 3 (bill) of Figure 4 are processed by the third stage of detection of Figure 1 (causing the higher recall, precision, and F1 score). The results are shown as the bold numbers of Table 14 in case of DKB v1 and those of Table 15 in case of JOD.

E. COMPARISONS WITH THE STATE-OF-THE-ART METHODS

In the following experiment, the accuracy of the proposed method was compared with those of the state-of-the-art methods. The performance of the proposed method was compared with those of the SURF-based banknote detection based on handcrafted features [1]–[3], [10], Faster R-CNN-based banknote detection based on deep features [22], MobileNet-based banknote detection [6], and YOLO v2 [26] and YOLO v3-based detection methods [21]. As summarized in Table 16, the proposed method exhibited a higher accuracy than the state-of-the-art methods for the DKB v1. As also summarized in Table 17, the proposed method exhibited a higher accuracy than the state-of-the-art

TABLE 16. Comparisons with the state-of-the-art methods with DKB v1.

Method		Sub-dataset	Recall	Precision	F1 score
Handcrafted feature	SURF [1-3,10]	Coin	0.6792	0.7031	0.6909
		Bill	0.8133	0.8912	0.8505
		Coin + bill	0.7252	0.8304	0.7742
Deep feature	Faster R-CNN [22]	Coin	0.7221	0.6715	0.6959
		Bill	0.9592	0.9315	0.9451
		Coin + bill	0.8553	0.8432	0.8492
	MobileNet [6]	Coin	0.7011	0.7552	0.7271
		Bill	0.8532	0.9314	0.8906
		Coin + bill	0.7513	0.8159	0.7823
	YOLO v2 [26]	Coin	0.8413	0.891	0.8654
		Bill	0.9642	0.9735	0.9688
		Coin + bill	0.8931	0.9368	0.9143
	YOLO v3 [21]	Coin	0.8437	0.9012	0.8715
		Bill	0.9233	0.9577	0.9402
		Coin + bill	0.8782	0.9338	0.9049
	Ours	Coin	0.9250	0.9548	0.9397
		Bill	0.9769	0.9880	0.9824
		Coin + bill	0.9486	0.9721	0.96

TABLE 17. Comparisons with the state-of-the-art methods with JOD.

Method		Sub-dataset	Recall	Precision	F1 score
Handcrafted feature	SURF [1-3,10]	Coin	0.7532	0.7798	0.7663
		Bill	0.8839	0.9007	0.8922
		Coin + bill	0.7954	0.8332	0.8139
Deep feature	Faster R-CNN [22]	Coin	0.7352	0.7411	0.7381
		Bill	0.9104	0.9207	0.9155
		Coin + bill	0.8294	0.8523	0.8407
	MobileNet [6]	Coin	0.7621	0.7831	0.7725
		Bill	0.8604	0.9119	0.8854
		Coin + bill	0.8105	0.8639	0.8363
	YOLO v2 [26]	Coin	0.8579	0.9132	0.8847
		Bill	0.9658	0.9612	0.9635
		Coin + bill	0.9132	0.9397	0.9263
	YOLO v3 [21]	Coin	0.8431	0.9014	0.8713
		Bill	0.9532	0.9712	0.9621
		Coin + bill	0.9014	0.9361	0.9184
	Ours	Coin	0.9076	0.9211	0.9143
		Bill	0.9512	0.9747	0.9628
		Coin + bill	0.9356	0.9604	0.9478

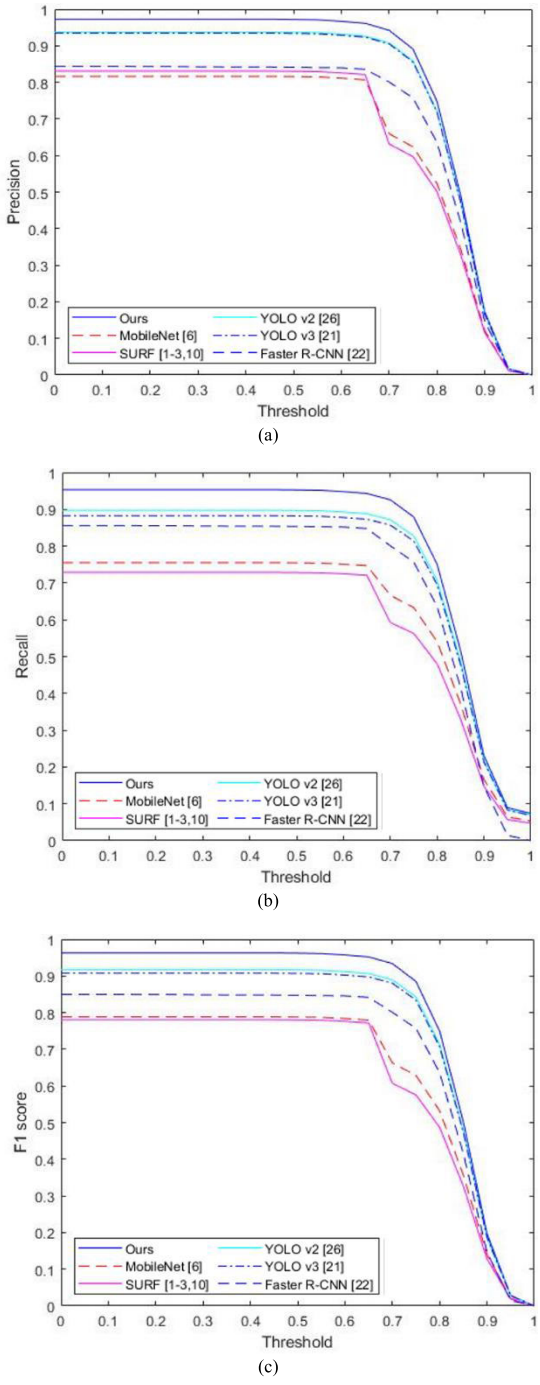


FIGURE 16. Comparative graphs versus IoU thresholds: (a) precision, (b) recall, and (c) F1 score with DKB v1.

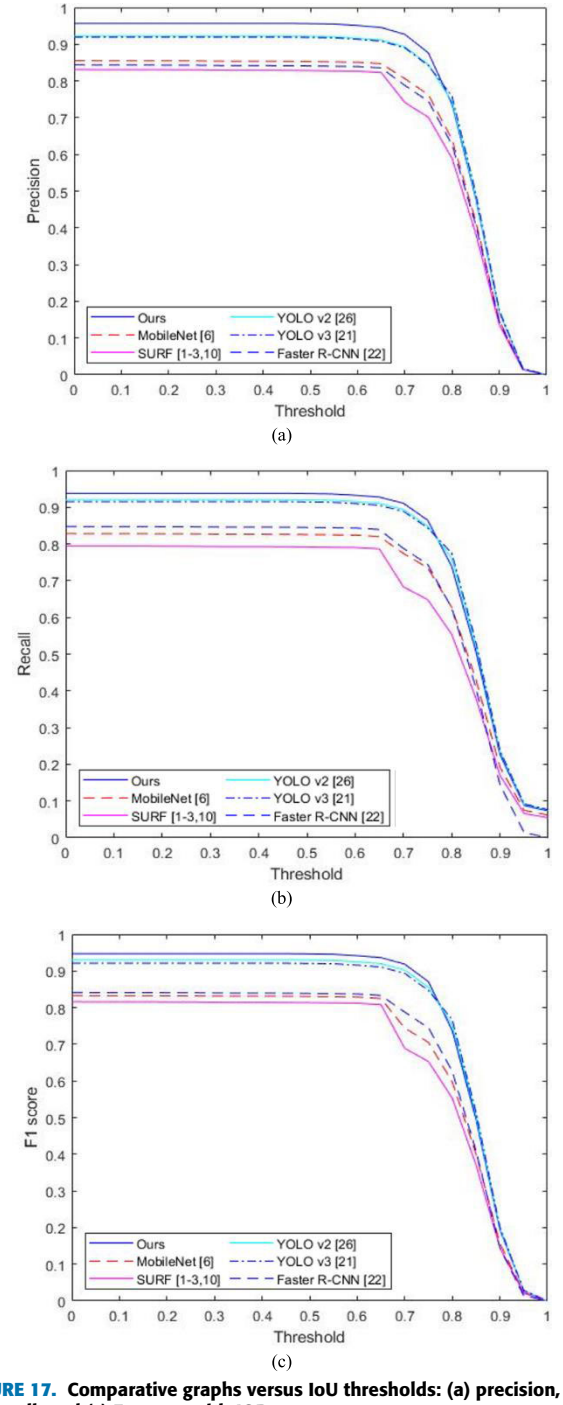


FIGURE 17. Comparative graphs versus IoU thresholds: (a) precision, (b) recall, and (c) F1 score with JOD.

methods for the JOD open database. The SURF-based detection method based on handcrafted features had a lower accuracy than the other deep feature-based detection methods, as summarized in Tables 16 and 17. MobileNet-based detection, which is based on deep features, uses a small number of layers. Thus, it shows a lower accuracy for detecting small objects such as coins and ultimately demonstrates a lower accuracy than the Faster R-CNN-based and YOLO-based detection methods. The YOLO-based detection methods exhibited a higher accuracy than the Faster R-CNN-based method, but the former had a lower detection performance than the proposed method.

Figures 16 and 17 present the graphs of the changes in precision, recall, and F1 score according to the IoU threshold of the proposed method and the state-of-the-art methods for the DKB v1 and JOD open databases, respectively. As shown in these figures, the proposed method had a higher detection accuracy than the state-of-the-art methods for all IoU thresholds.

V. CONCLUSION

In this study, a new method was proposed for banknote detection with banknote images captured in complicated backgrounds and various environments using a smartphone camera. To improve the detection performance in the VGG-16-based Faster R-CNN of the first stage of detection, post processing methods were applied as the second stage of detection based on the three features, namely, the width-to-height ratio, detection box size, and detection score, to remove the FP candidates. For the candidates remaining after the post processing, verification was performed as the third stage of detection by the ResNet-18-based Faster R-CNN to detect the final banknote region. Furthermore, the self-collected DKB v1 and the developed models with algorithms were disclosed for a fair evaluation by other researchers as shown in [8]. When the experiments were conducted with the DKB v1 and JOD open databases, high detection performance was obtained for bills, but FP detection errors were produced for coins.

Further studies would be conducted on deep networks that can detect small objects such as coins in images more accurately. In addition, a shallow network-based detection method that can shorten the processing time would be examined.

REFERENCES

[1] G. A. R. Sanchez, "A computer vision-based banknote recognition system for the blind with an accuracy of 98% on smartphone videos," *J. Korea Soc. Comput. Inf.*, vol. 24, pp. 67–72, Jun. 2019.

[2] G. A. R. Sanchez, Y. J. Uh, K. Lim, and H. Byun, "Fast banknote recognition for the blind on real-life mobile videos," in *Proc. Korean Comput. Conf.*, Jeju Island, South Korea, Jun. 2015, pp. 835–837.

[3] F. M. Hasanuzzaman, X. Yang, and Y. Tian, "Robust and effective component-based banknote recognition by SURF features," in *Proc. 20th Annu. Wireless Opt. Commun. Conf. (WOCC)*, Newark, NJ, USA, Apr. 2011, pp. 1–6.

[4] Y. Li, C. Yang, L. Zhang, R. Xia, L. Fan, and W. Xie, "A novel SURF based on a unified model of appearance and motion-variation," *IEEE Access*, vol. 6, pp. 31065–31076, Jun. 2018.

[5] T. D. Pham, C. Park, D. T. Nguyen, G. Batchuluun, and K. R. Park, "Deep learning-based fake-banknote detection for the visually impaired people using visible-light images captured by smartphone cameras," *IEEE Access*, vol. 8, pp. 63144–63161, Apr. 2020.

[6] S. Mittal and S. Mittal, "Indian banknote recognition using convolutional neural network," in *Proc. 3rd Int. Conf. Internet Things, Smart Innov. Usages (IoT-SIU)*, Bhimtal, India, Feb. 2018, pp. 1–6.

[7] D. G. Pérez and E. B. Corrochano, "Recognition system for Euro and Mexican banknotes based on deep learning with real scene images," *Computación y Sistemas*, vol. 22, no. 4, pp. 1065–1076, Dec. 2018.

[8] DM Lab. (2020). *Dongguk Korean Banknote Database Version1 (DKB V1) and CNN Models for Banknote Detection*. Accessed: Mar. 1, 2020. [Online]. Available: http://dm.dgu.edu/link.html

[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Kauai, HI, USA, Dec. 2001, pp. I-511–I-518.

[10] L. D. Dunai, M. C. Pérez, G. P. Fajarnés, and I. L. Lengua, "Euro banknote recognition system for blind people," *Sensors*, vol. 17, no. 1, pp. 1–15, Jan. 2017.

[11] J. Liang and S. Y. Yuen, "A novel saliency prediction method based on fast radial symmetry transform and its generalization," *Cognit. Comput.*, vol. 8, no. 4, pp. 693–702, Aug. 2016.

[12] A. R. Domínguez, C. L. Alvarez, and E. B. Corrochano, "Automated banknote identification method for the visually impaired," in *Proc. Prog. Pattern Recognit. Image Anal. Comput. Vis. Appl.*, Puerto Vallarta, Mexico, Nov. 2014, pp. 572–579.

[13] A. I. Ahmed, J. P. Chiverton, D. L. Ndzi, and V. M. Becerra, "Speaker recognition using PCA-based feature transformation," *Speech Commun.*, vol. 110, pp. 33–46, Jul. 2019.

[14] F. Grijalva, J. C. Rodriguez, J. Larco, and L. Orozco, "Smartphone recognition of the U.S. Banknotes' denomination, for visually impaired people," in *Proc. IEEE ANDESCON*, Bogota, Colombia, Sep. 2010, pp. 1–6.

[15] N. A. J. Sufri, N. A. Rahmad, M. A. As'ari, N. A. Zakaria, M. N. Jamaludin, L. H. Ismail, and N. H. Mahmood, "Image based ringgit banknote recognition for visually impaired," *J. Telecomm. Electron. Comput. Eng.*, vol. 9, nos. 3–9, pp. 103–111, 2017.

[16] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers: 2nd edition (with Python examples)," 2020, *arXiv:2004.04523*. [Online]. Available: http://arxiv.org/abs/2004.04523

[17] N. Dey, S. Borah, R. Babo, and A. S. Ashour, "Classification and analysis of Facebook metrics dataset using supervised classifiers," in *Social Network Analytics: Computational Research Methods and Techniques*. Cambridge, MA, USA: Academic, 2019, pp. 1–267.

[18] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[19] 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*. [Online]. Available: http://arxiv.org/abs/1704.04861

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[21] R. C. Joshi, S. Yadav, and M. K. Dutta, "YOLO-v3 based currency detection and recognition system for visually impaired persons," in *Proc. Int. Conf. Contemp. Comput. Appl. (IC3A)*, Lucknow, India, Feb. 2020, pp. 280–285.

[22] Q. Zhang, "Currency recognition using deep learning," M.S. thesis, Dept. Comput. Inf. Sci., Auckland Univ. Technol., Auckland, New Zealand, 2018.

[23] U. R. Chowdhury, S. Jana, and R. Parekh, "Automated system for Indian banknote recognition using image processing and deep learning," in *Proc. Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, Gunupur, India, Mar. 2020, pp. 1–5.

[24] D. San Martin and D. Manzano, "A deep learning model for Chilean bills classification," 2019, *arXiv:1912.12120*. [Online]. Available: http://arxiv.org/abs/1912.12120

[25] M. Jadhav, Y. K. Sharma, and G. M. Bhandari, "Currency identification and forged banknote detection using deep learning," in *Proc. Int. Conf. Innov. Trends Adv. Eng. Technol. (ICITAET)*, Shegaon, India, Dec. 2019, pp. 178–183.

[26] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*. [Online]. Available: http://arxiv.org/abs/1612.08242

[27] 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.

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

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–14.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[32] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10687–10698.

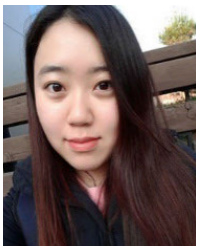
[33] Samsung Electronics Co. (2020). *Galaxy Note 5*. Accessed: Aug. 2, 2020. [Online]. Available: https://www.samsung.com/global/galaxy/galaxy-note5/

[34] I. A. Doush and S. AL-Btoush, "Currency recognition using a smartphone: Comparison between color SIFT and gray scale SIFT algorithms," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 4, pp. 484–492, Oct. 2017.

[35] NVIDIA Corp. (2020). *GeForce GTX 1070*. Accessed: Mar. 1, 2020. [Online]. Available: https://www.nvidia.com/en-in/geforce/products/10series/geforce-gtx-1070/
[36] The Mathworks. (2020). *Matlab*. Accessed: Mar. 10, 2020. [Online]. Available: https://www.mathworks.com/
[37] NVIDIA Corp. (2020). *CUDA*. Accessed: Apr. 12, 2020. [Online]. Available: https://en.wikipedia.org/wiki/CUDA
[38] NVIDIA Corp. (2020). *cuDNN*. Accessed: Apr. 20, 2020. [Online]. Available: https://developer.nvidia.com/cudnn
[39] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 421–436.
[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 618–626.
[41] T. M. Hoang, S. H. Nam, and K. R. Park, "Enhanced detection and recognition of road markings based on adaptive region of interest and deep learning," *IEEE Access*, vol. 7, pp. 109817–109832, Aug. 2019.
[42] WIKIPEDIA. (2020). *Precision and Recall*. Accessed: Mar. 1, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Precision_and_recall

CHANHUM PARK received the B.S. degree in electronics and electrical engineering from Dongguk University, Seoul, South Korea, in 2018, where he is currently pursuing the combined M.S. and Ph.D. degrees. His research interests include banknote and pattern recognition. He designed the banknote detection method and wrote the original paper.

SE WOON CHO received the B.S. degree in electronics and electrical engineering from Dongguk University, Seoul, South Korea, in 2017, where he is currently pursuing the combined M.S. and Ph.D. degrees in electronics and electrical engineering. His research interests include biometrics and pattern recognition. He helped the implementation of Faster R-CNN.



NA RAE BAEK received the B.S. degree in electronics and electrical engineering from Dongguk University, Seoul, South Korea, in 2017, where she is currently pursuing the combined M.S. and Ph.D. degrees in electronics and electrical engineering. Her research interests include biometrics and pattern recognition. She helped the experiments and analysis.

JIHO CHOI received the B.S. degree in business administration from Dongguk University, Seoul, South Korea, in 2016, where he is currently pursuing the combined M.S. and Ph.D. degrees in electronics and electrical engineering. His research interests include biometrics and pattern recognition. He helped the experiments and analysis.

KANG RYOUNG PARK (Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Yonsei University, Seoul, South Korea, in 1994 and 1996, respectively, and the Ph.D. degree in electrical and computer engineering from Yonsei University, in 2000. He has been a Professor with the Division of Electronics and Electrical Engineering, Dongguk University, since March 2013. His research interests include image processing and deep learning. He supervised this study and helped the revision of original paper.