applied sciences

MDPI

Article

Keyword Detection Based on RetinaNet and Transfer Learning for Personal Information Protection in Document Images

Guo-Shiang Lin [1,*], Jia-Cheng Tu [2] and Jen-Yung Lin [2]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan
[2] Department of Computer Science and Information Engineering, Da-Yeh University, Changhua 515, Taiwan; r0506005@cloud.dyu.edu.tw (J.-C.T.); jylin@mail.dyu.edu.tw (J.-Y.L.)
* Correspondence: gslin@ncut.edu.tw

Featured Application: The proposed scheme can be utilized to protect personal information in document images.

Abstract: In this paper, a keyword detection scheme is proposed based on deep convolutional neural networks for personal information protection in document images. The proposed scheme is composed of key character detection and lexicon analysis. The first part is the key character detection developed based on RetinaNet and transfer learning. To find the key characters, RetinaNet, which is composed of convolutional layers featuring a pyramid network and two subnets, is exploited to detect key characters within the region of interest in a document image. After the key character detection, the second part is a lexicon analysis, which analyzes and combines several key characters to find the keywords. To train the model of RetinaNet, synthetic image generation and data augmentation are exploited to yield a large image dataset. To evaluate the proposed scheme, many document images are selected for testing, and two performance measurements, IoU (Intersection Over Union) and mAP (Mean Average Precision), are used in this paper. Experimental results show that the mAP rates of the proposed scheme are 85.1% and 85.84% for key character detection and keyword detection, respectively. Furthermore, the proposed scheme is superior to Tesseract OCR (Optical Character Recognition) software for detecting the key characters in document images. The experimental results demonstrate that the proposed method can effectively localize and recognize these keywords within noisy document images with Mandarin Chinese words.

Keywords: document image; deep learning; convolutional neural network; RetinaNet

Citation: Lin, G.-S.; Tu, J.-C.; Lin, J.-Y. Keyword Detection Based on RetinaNet and Transfer Learning for Personal Information Protection in Document Images. *Appl. Sci.* **2021**, *11*, 9528. https://doi.org/10.3390/app11209528

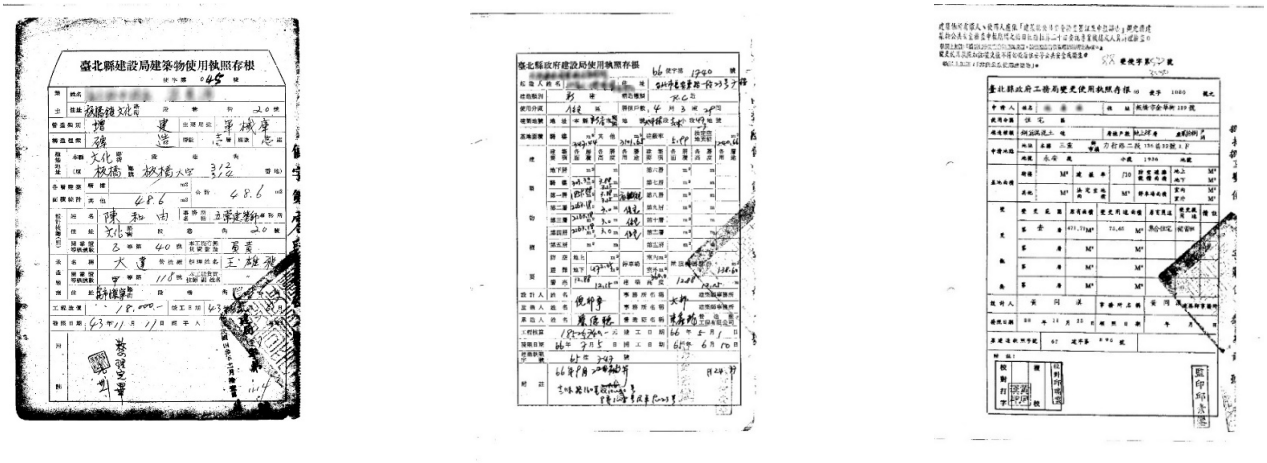
Academic Editor: Amerigo Capria

Received: 31 August 2021
Accepted: 6 October 2021
Published: 13 October 2021

1. Introduction

There is a lot of information from paper documents for human communication. Paper documents often contain typical elements such as text, tables, stamps, and signatures. Due to the advancement of digital technologies, a massive of document images have been digitized in Taiwan, so users can now easily search and access these document images. As for document image searching, some document image retrieval methods [1–3] have been developed. In ref. [1], some representative features extracted from document images by using a Convolutional Neural Network (CNN) were used for a retrieval task. As for personal information protection, since the property rights may be transferred and spatial attributes may be modified, the document image should contain these changes. Accordingly, it is significant that document images can generate a great influence on real estate value in a way that consequently causes great concern for people's property rights. However, some malicious users may not have any difficulties obtaining the property owner's information or even manipulating document images for illegal purposes. This is why document images should be kept secure and private in terms of privacy policies.

Figure 1 illustrates three examples of digitized document images in Taiwan. As we can observe in Figure 1, each document image usually has a page frame, which is the smallest rectangle and encloses most of the foreground elements; some Chinese words are around the page frame, and some noise may exist. The foreground elements in the page frame often include the owner's information, designer's information, and building's information. To avoid deliberately prying or illegal access, it is important to keep the property owner's information private and secure according to the viewpoint of the privacy policy. However, the cost of manually detecting and concealing the property owner's information in document images is very high. This means that automatically detecting the personal information in document images is an important pre-processing step for intelligent and efficient application software. Furthermore, as shown in Figure 1, personal information usually follows specific printed Mandarin Chinese words, such as "業主" (Property Owner), "姓名" (Name), "住址" (Address), etc. It is expected that correctly detecting suitable, specific, printed Mandarin Chinese words can easily pin down personal information in a document image. In this sense, those observations have motivated us to develop a scheme for automatically detecting specific printed Mandarin Chinese words that are useful to localize personal information in document images.



(**a**) (**b**) (**c**)

Figure 1. Three examples of digitized document images in Taiwan: (**a**) case 1, (**b**) case 2, and (**c**) case 3.

According to traditional document processing methods, image layout analysis, which is performed to locate some regions of interest (ROIs), is often designed as a pre-processing step for further analysis and applications such as personal information protection, document classification, document image analysis, and document image retrieval [4,5]. In our previous work [4], we proposed a coarse-to-fine scheme to automatically detect the ROIs for digitized cadastral images in Taiwan. The existing scheme was composed of four parts: pre-processing, skew correction, noise reduction, and ROI localization. After finding the coordinates of the candidate region in the de-skew image with high resolution, ROIs can be localized in the high-resolution images by using the ROI location algorithm in the fine detection. Although ROIs can be localized in the digitized cadastral images, it is still a little complicated to find the correct position of personal information according to the layout information of cadastral images. In ref. [5], a machine-learning-based patient identification recognition method was proposed for a medical information system. In the proposed method [5], the color information is used to identify camera-captured screen images, a bilateral filter is used to reduce the effect of noise in captured screen images, and then the color and spatial information are used to initially and roughly locate the candidate region. After skew correction, a template-matching algorithm is used to find special symbols for locating the ROI. Then, the patient identification information can be

recognized in the ROI. Unfortunately, traditional image layout analysis methods may not deal with noisy document images well. It is expected that poor image layout analysis or document segmentation may affect the following recognition process [6].

Compared with traditional machine-learning methods developed based on hand-crafted features, Deep Neural Networks (DNN) [7,8] have received more and more attention due to their excellent performance in image classification, speech recognition, fraud detection, and so on. Many CNNs, such as AlexNet, VGG, GoogleNet, and ResNet, demonstrate that they have better capabilities to extract multi-level visual features for improving the accuracy of classification models [7–13]. As for document image analysis and scene image analysis, there are some deep-learning-based methods [6,14–18]. For example, a deep-learning approach is proposed based on a fully convolutional network (FCN) [9] for document segmentation [15]. In ref. [16], a fast CNN-based method is proposed to automatically perform layout analysis for document images. In the existing method [16], a document image is segmented into some blocks, and these blocks are classified into three categories, i.e., text, table, and image, based on a CNN. In [6], some deep CNNs such AlexNet, VGG-16, GoogleNet, and ResNet-50 are used to distinguish document images into some categories such as emails, news articles, and invoices for document image classification. In ref. [17], an existing method, EAST, similar to FCN [9], is exploited to detect text regions in scene images. In EAST, multi-level feature maps are extracted, merged, and fed into the output layer for pixel-level predictions of text regions. Unfortunately, the text detection method [17] may not be suitable to analyze these document images, which usually have many Mandarin Chinese words in Figure 1.

As far as we are concerned, object detection plays an important role in vision-based applications such as face detection, traffic sign detection, character/digit detection, person re-identification, animal detection, and meter reading [5,19–22]. So far, in addition to traditional machine-learning-based methods, there are some CNN-based object-detection algorithms, including R-CNN (region-based convolutional neural network), Faster R-CNN, you only look once (YOLO), and a single-shot multiBox detector (SSD), RetinaNet et al. [23]. For example, an automatic meter-reading method was proposed for gas meter reading [20]. The proposed method [20] is composed of three steps: meter detection, digit segmentation, and number recognition. In the three steps, YOLOv3, maximally stable extremal regions (MSER), and a modified VGG network are adopted in [20]. In [22], a two-stage method is proposed for an automatic meter reading. In the first stage, a small model, called fast-YOLO, is used for counter-detection. After counter-detection, digit recognition can be achieved based on CR-NET, CRNN, and multi-task learning in the second stage. For classifying document images, some Mandarin Chinese words such as "業主" (Property Owner) and "起造人姓名" (Name of Applicant) can be used as the special information to distinguish Figure 1a from Figure 1b. This circumstance has inspired us to analyze the document images for finding the specific information via CNNs. Therefore, we aim at developing a CNN-based object-detection scheme to directly detect and recognize some Mandarin Chinese words in noisy document images for document image analysis.

The remainder of this paper is organized as follows. Section 2 describes the system description of the proposed keyword detection scheme. Sections 3 and 4 elaborate the key character detection and lexicon analysis of the proposed scheme, respectively. Section 5 shows experimental results, and Section 6 gives a discussion. Section 7 draws a conclusion and future work.

2. System Description

As mentioned in the previous section, some specific Mandarin Chinese words in document images can be considered as objects. Consider an input document image $X = \{x(i,j)|1 \le i \le N_W, 1 \le j \le N_H\}$ of size $N_W \times N_H$, where $x(i,j)$ denotes the pixel value at the coordinate (i,j), N_W and N_H represent the width and height of X, respectively. The goal of the proposed scheme is to detect keywords, O^{KW}, in X. As we know, a keyword (i.e., specific word) is often composed of some key characters (i.e., specific characters) in

Mandarin Chinese, i.e., $O^{KW} = \cup O^{KC}$, where O^{KC} denotes the key character. For example, the keyword "姓名" (Name) is composed of two key characters, "姓" (Surname) and "名" (First Name), in Mandarin Chinese. This means that a keyword can be found after analyzing these detected key characters. Therefore, it is important to efficiently localize and analyze the key characters from X for the proposed keyword-detection scheme.

As shown in Figure 1, detecting key characters in digitized document images should consider some issues. First, document quality and digitization processing may affect the image properties of digitized document images. As shown in Figure 1a, the visual quality of Figure 1a is poor, and it is tough to analyze degraded document images to detect key characters. This means that the image quality and spatial resolutions of scanned document images are not the same. Second, marginal noise, artificial noise, and random noise may exist in digitized document images. As shown in Figure 1, there is marginal noise around the paper frames, and a seal occurs to be artificial noise on the bottom-right part of Figure 1a. It is no doubt that analyzing the layout of a noisy document image is difficult. Third, these documents may have some popular keywords, but some seldom happen. For example, the keyword "姓名" (Name) usually exists in document images, but "申請人" (Applicant) may only happen in some kinds of documents. Fourth, keywords may appear in some possible locations and with different fonts. For instance, the keywords "姓名" (Name) have different fonts in Figure 1.

To consider the above issues and the efficiency requirement for keyword detection, the block diagram of the proposed scheme is shown in Figure 2. As shown in Figure 2, the proposed scheme is composed of two parts: key character detection and lexicon analysis.

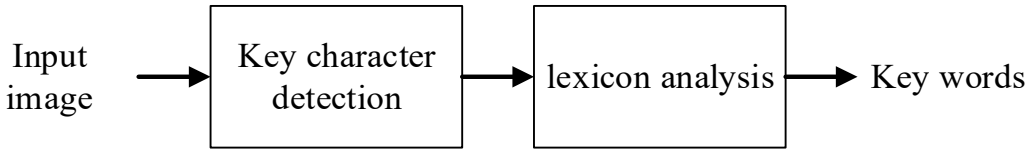


Figure 2. Methodological illustration of the proposed scheme.

The first part is designed to effectively detect the key characters in a document image. As mentioned in Section 1, CNN-based object-detection methods can be used to localize key characters. These CNN-based object-detection methods can be classified into two classes: one stage and two stages. For two-stage methods, the first step is to find many region proposals, and then these region proposals are analyzed, located, and recognized in the other step. Some frameworks, such as R-CNN, Fast R-CNN, and Faster R-CNN, are two-stage methods [24–26]. On the contrary, one-stage methods [11–13,27], such as SSD, RetinaNet, and YOLO, perform region proposal searching, region proposal localization, and region proposal recognition at the same stage. Though there are more proposals in the one-stage methods compared with the two-stage methods, the one-stage object-detection methods still have a shorter computational time.

In addition, key characters with different frequencies of occurrence may occur in a document image, and the number of the key characters may vary in document images. As we can see in Figure 1, some key characters, e.g., "名" (Name), often exist, but some, e.g., "業" (Property), seldom exist in document images. This means that class imbalance may happen so that some key characters with low appearance frequencies may not be detected well. As we know, the one-stage, CNN-based object-detection method, RetinaNet [27] with a novel loss function, can not only detect objects but also reduce the impact of class imbalance on object classification. From the view of object-detection performance, RetinaNet outperforms SSD, YOLOv2, and YOLOv3 for the COCO dataset [28]. From the view of computation complexity, an object-detection network usually has higher computational complexity compared with an image classification network [28]. For one-stage object-detection networks, RetinaNet is superior to YOLOv3 in terms of FLOPs (floating-point operation) and mAP (mean average precision) for object detection [28]. This means that RetinaNet is a better choice of object detection in terms of computational complexity

and object-detection performance. Thus, we develop the key character-detection method based on RetinaNet [27].

After key character detection, the other part is to analyze these detected key characters to obtain the keywords in a document image. Then, the detected keywords can be exploited for personal information protection and document image classification. In the following, we elaborate on each part of the proposed scheme.

3. Key Character Detection

The proposed key-character-detection algorithm is composed of two parts: ROI localization and CNN-based key character detection. We describe the two parts in the following.

3.1. ROI Localization

In real environments, the spatial resolutions of digitalized document images are often different in different cities of Taiwan. It is expected that the computational complexity of dealing with digitalized document images with different spatial resolutions is high. Furthermore, as we can see in Figure 1, the personal information of owners is usually located in the top part of a document image. The observation can be considered as a priori information that reduces the search space for key character detection. It is expected that the candidate number of the key characters can be reduced if the key-character-detection algorithm only processes the specific region, which is the region of interest (ROI). Figure 3 illustrates the ROI within an input image. According to our experiences, an ROI whose size is $N_W^R \times N_H^R$ pixels can be localized in the top part of each input image. Furthermore, since ROI is smaller than the input image, it is expected that the system efficiency of the proposed scheme can be effectively raised.

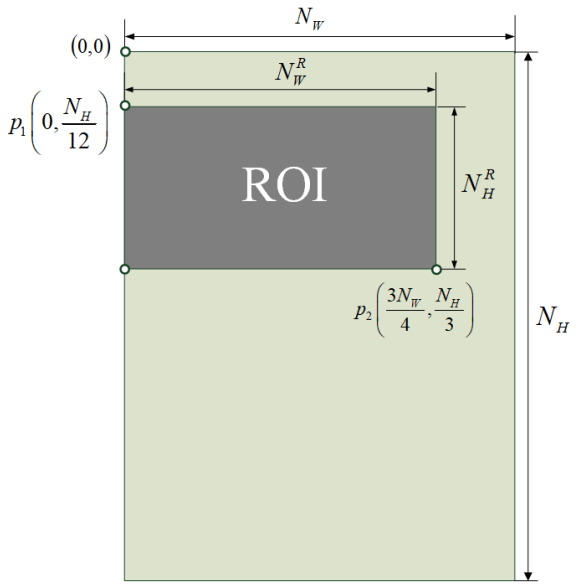


Figure 3. Illustration of ROI within an input image.

3.2. CNN-Based Key Character Detection

Assume there are N^{KC} key characters determined in the proposed scheme. As mentioned in Section 2, RetinaNet is adopted to detect key characters in ROIs.

3.2.1. RetinaNet Architecture

Figure 4 illustrates the block diagram of RetinaNet [27]. As shown in Figure 4, the RetinaNet architecture has some sub-parts: ResNet [8], Feature Pyramid Network (FPN) [29], classification subnet, and box regression subnet. In the first sub-part, ResNet, which is one of the famous CNNs, is exploited to extract multi-level feature maps from different convolutional layers for image classification and object detection. In the second subpart,

the concept of image pyramid used in SIFT, SURF, HOG, YOLOv3 [30], and YOLOv4 [31] is adopted here. To raise the performance of detecting objects with different sizes, FPN is adopted in RetinaNet. The main reason is that FPN is utilized to combine multi-scaled feature maps and then obtain rich feature maps. In addition to feature-point extraction and object detection, FPN is also used to achieve instance segmentation [32]. After FPN, the classification subnet based on the full convolution network (FCN) predicts the probability of a present object to obtain the object class information in the multi-scale feature maps. Similarly, the box regression subnet is used to estimate the coordinate information of objects with different sizes in the multi-scale feature maps. Then, it is expected that RetinaNet can effectively detect key characters with different sizes for document images.

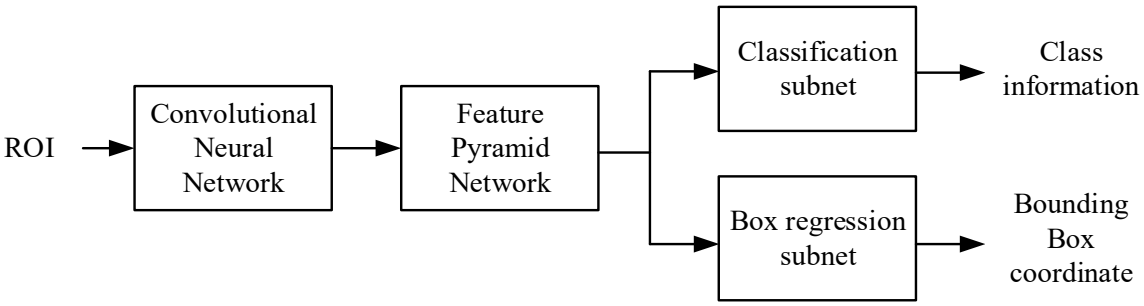


Figure 4. Illustration of key character detection based on RetinaNet.

To correctly classify and localize the key characters, the total loss function of RetinaNet is the sum of the classification loss and the bounding box regression loss for network training. For object classification, a common loss function, cross-entropy loss, is described as follows:

$$CE(p,y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{if } y = 0 \end{cases} \quad (1)$$

where $y = \{0,1\}$ denotes whether the ground-truth class exists, $p \in [0,1]$ represents the probability output of the proposed scheme with y = 1 (i.e., the ground-truth class exists), and $\log(\cdot)$ denotes the logarithmic function. Furthermore, to avoid class imbalance and easy/hard example problems during the training, the loss weight for well-classified samples (i.e., easy samples) can be reduced, and the weight of hard samples can be increased. Then, the focal loss function in RetinaNet is designed as the classification loss in the following:

$$FLoss(p,y) = \alpha \cdot (1-p_t)^{\gamma} \cdot CE(p,y) \quad (2)$$

where α is a weighting factor, $p_t = y \cdot p + (1-y)(1-p)$, and γ is a constant used in a modulating factor $(1-p_t)^{\gamma}$. As we can see in Equation (2), the effect of the modulating factor is enlarged when γ is increased. This means that the focal loss contains more errors from the hard samples, and then RetinaNet can learn more experiences from the hard samples for raising the performance of object detection when γ is bigger. On the other hand, RetinaNet is also needed to evaluate whether an object is correctly located. To evaluate whether each true object is detected by a bounding box, the smooth L_1 loss [26] applied as the loss function to the box regression subnet for object localization is described as follows:

$$L_{1-smooth}(z) = \begin{cases} |z| & \text{if } |z| > \beta \\ \frac{1}{|\beta|} z^2 & \text{if } |z| \leq \beta \end{cases} \quad (3)$$

where z denotes the input of the smooth L_1 loss and β is a hyper-parameter. It is expected that the detected bounding box is close to the true object when the smooth L_1 loss is small. Therefore, the trained RetinaNet can be obtained by minimizing the total loss function.

3.2.2. Model Training Procedure

Though RetinaNet is suitably utilized for key character detection, the following problem is how to efficiently train a RetinaNet model and simultaneously prevent the issue of model overfitting. So far, RetinaNet has been used to deal with some applications such as pedestrian detection, face mask detection, and pill detection in real environments. As we know, most deeper models usually need a lot of data and corresponding labeled information for supervised learning. Unfortunately, it is usually expensive work to manually obtain the ground-truth labeling information of a large training set for model learning. Fortunately, transfer learning [33] is a machine-learning technique where a model trained on one task is re-used on a new related task even under the situation of limited annotated data. It is expected that transfer learning is an efficient way to obtain a suitable RetinaNet model for document images. Though transfer learning is an efficient way to achieve model learning, the following issues are how to efficiently obtain the training dataset with annotation information and re-train the RetinaNet.

Figure 5 illustrates the model training for key character detection via transfer learning and synthetic images. First, one approach to reducing the cost of obtaining the ground-truth labeling information is to generate synthetic data with annotation information [34,35]. In fact, it is easy to render a lot of synthetic images with synthesized key characters. Here, some fonts are exploited to synthesize the key characters and then render a large training dataset via synthetic image generation and data augmentation for model learning. The other is that ImageNet is one of the most widely used image datasets in many applications. It is expected that the convolutional layers of a pre-trained RetinaNet based on ImageNet should contain a lot of rich knowledge extracted from natural images. Therefore, our model learning strategy is that RetinaNet is initialized by a pre-trained model with ImageNet, and then the training set of synthetic images, including these key characters, is utilized to efficiently re-train the RetinaNet for document images. After transfer learning, the re-trained RetinaNet can be exploited to detect the key characters in unseen ROIs.

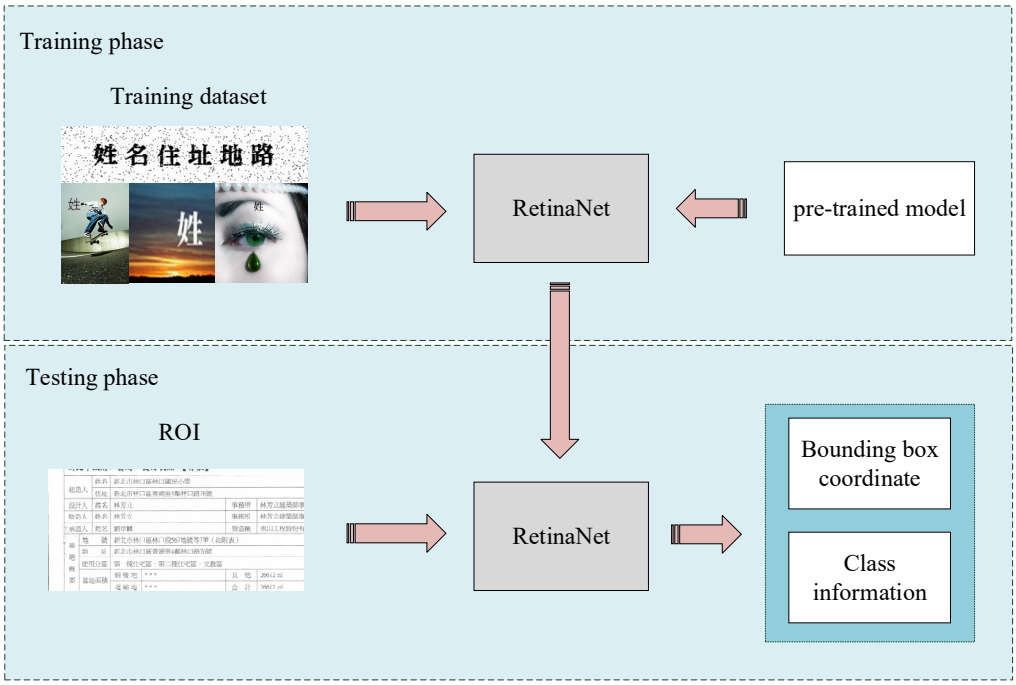


Figure 5. Illustration of model training for key character detection via transfer learning and synthetic images.

4. Lexicon Analysis

Since a document image should have some key characters, these detected characters may be located within the ROI after key character detection. In fact, some detected characters may not be helpful for finding personal information. Then, the following question is related to how to analyze the detected key characters to obtain the correct keywords in ROIs.

In fact, a word is often composed of two or three characters in Mandarin Chinese. When trying to find meaningful keywords in a document image, there are two problems. The first one is that two or three arbitrary characters may not be a meaningful word in Mandarin Chinese. For example, the keyword "業主" (Property Owner) composed of "業" (Property) and "主" (Owner) is a meaningful word, but the combination of "址" (Address) and "主" (Owner) is non-meaningful in Mandarin Chinses. In addition, the order of two or three characters is also important for the combination of a meaningful word. For example, the combination of "姓名" (Surname–First Name) is meaningful, but the combination of "名姓" (First Name–Surname) is not. This means that a lexicon analysis is necessary in the proposed scheme.

On the other hand, the spatial relation of two or three key characters could be horizontal and vertical. As shown in Figure 1a, the keywords may be arranged in the horizontal or vertical direction. For example, the two keywords, "業主" (Property Owner) and "住址" (Address), are in the vertical and horizontal directions, respectively. This means that the lexicon analysis algorithm should be performed in the horizontal and vertical directions. Furthermore, the observation that the spatial distance of two or three key characters may be different can be observed in Figure 1. For example, the spatial distance between "姓" (Surname) and "名" (First Name) in Figure 1a is smaller than that in Figure 1b. To find two or three characters to generate a meaningful word, a searching algorithm is proposed here. Therefore, a lexicon analysis algorithm is necessary to find and analyze the combination of some possible key characters for obtaining correct keywords.

To devise a searching algorithm in the lexicon analysis, an observation that the distance between two characters is usually equal to multiple times a character's width with a document in Taiwan can be utilized. This means that a character's width can be considered as a prior information for developing the searching algorithm. To estimate a character's width, the bounding boxes enclosing the detected characters are collected during the training phase, and then their width information is analyzed to estimate the size information as a prior information.

Figure 6 illustrates the lexicon analysis. Assume there are N^{KW} keywords determined in the proposed scheme. Then, the lexicon analysis algorithm in the horizontal direction is described as follows:

(L1) Select one possible key character O_1^{KC} as the first one and then measure its center $\overline{p}_1^C$, whose coordinate is $(\overline{x}_1, \overline{y}_1)$. For example, "姓" (Surname) is the first key character.

(L2) Determine whether another key character O_i^{KC} is around O_1^{KC}. According to the prior width information of a character, the positions whose horizontal coordinate is around the position $\overline{y}_1 + j \cdot N_W^{Ch}$ are checked whether a key character exists, where j is an integer and N_W^{Ch} is the estimated width of a key character. If there is a key character around the positions, the second character is considered as the second one O_2^{KC}. For example, "名" (First Name) is the second character. Then, the two key characters are considered as the first keyword candidate $O_1^{KW} = O_1^{KC} \cup O_2^{KC}$. For example, "姓名" (Name) is the candidate of the first keyword.

(L3) Decide whether the combination of the two key characters matches one of the pre-defined keywords. If the first keyword candidate O_1^{KW} matches one of the pre-defined keywords, O_1^{KW} is a detected keyword. For example, "姓名" (Name) is meaningful, and it is also one of the pre-defined keywords.

(L4) Repeat Steps (L2) to (L3) to obtain a keyword if the combination of these key characters matches one of the pre-defined keywords. For example, "住址" (Address) is another pre-defined keyword.

(L5) Repeat Steps (L1) to (L4) to detect all of the keywords.

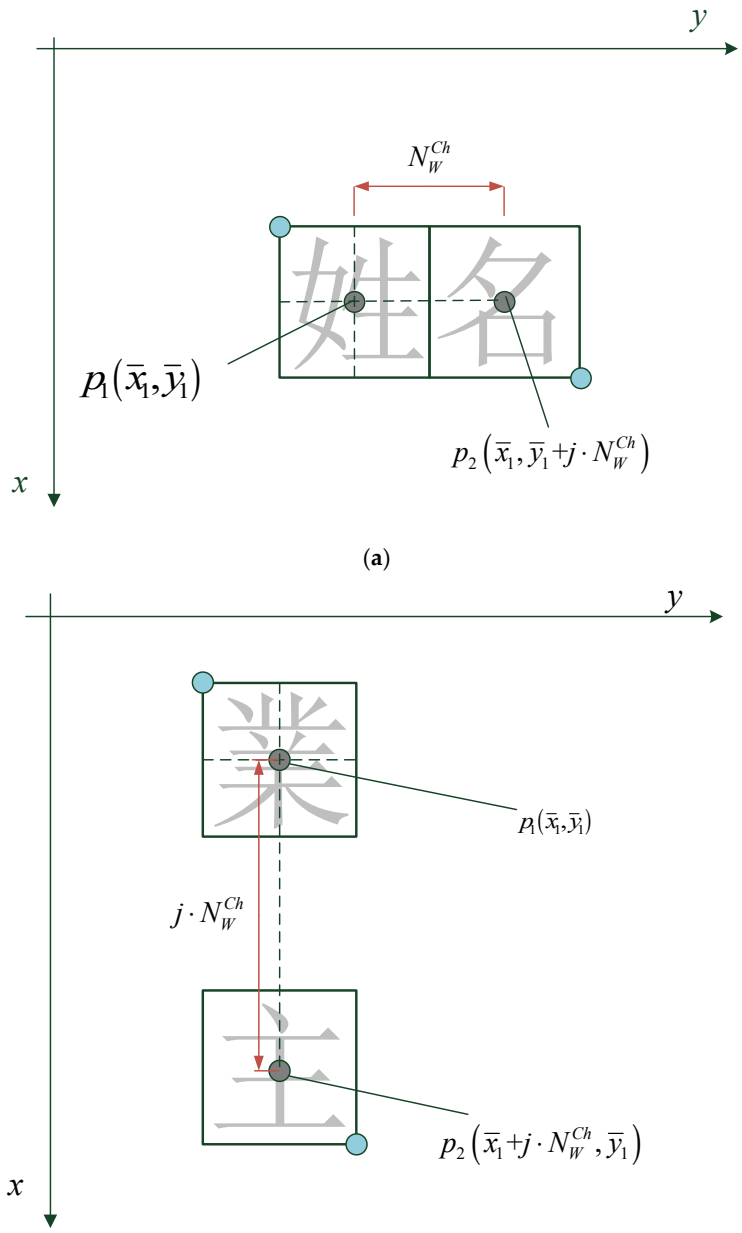


Figure 6. Illustration of lexicon analysis: (**a**) horizontal direction and (**b**) vertical direction.

Similar to Steps (L1) to (L5), the lexicon analysis algorithm can be performed in the vertical direction to detect some keywords, e.g., "業主" (Property Owner). After lexicon analysis, these keywords can be detected in document images.

5. Experimental Results

To evaluate the performance of our proposed scheme, a total of 124 digitized document images whose sizes are from 824 × 1169 to 2503 × 3571 pixels are collected and contain several kinds of document images. Figure 7 shows some examples of document images for performance evaluation. Note that due to personal privacy, personal information is blurred for protection. As shown in Figure 7, the visual quality may be poor, and the image content of page frames may be different among the test images. As we can observe in Figure 7d,e, the keyword "申請人" (Applicant) has different fonts. In addition, the experiments were

performed on AMD Ryzen 7 with 32 GB RAM and one graphic card (Nvidia GTX 1080 Ti). The backbone of RetinaNet is ResNet-50, and its input size is limited to 1333 × 800 pixels in the proposed scheme. Lots of synthetic images are yielded to train the RetinaNet, and then a testing set of 112 images are used for performance evaluation. Here, the optimizer for network model learning is the adaptive moment estimation (Adam) algorithm, and the learning rate is 0.00001 [36]. In the following, we conducted some experiments to evaluate the performance of the proposed keyword detection scheme for document images.

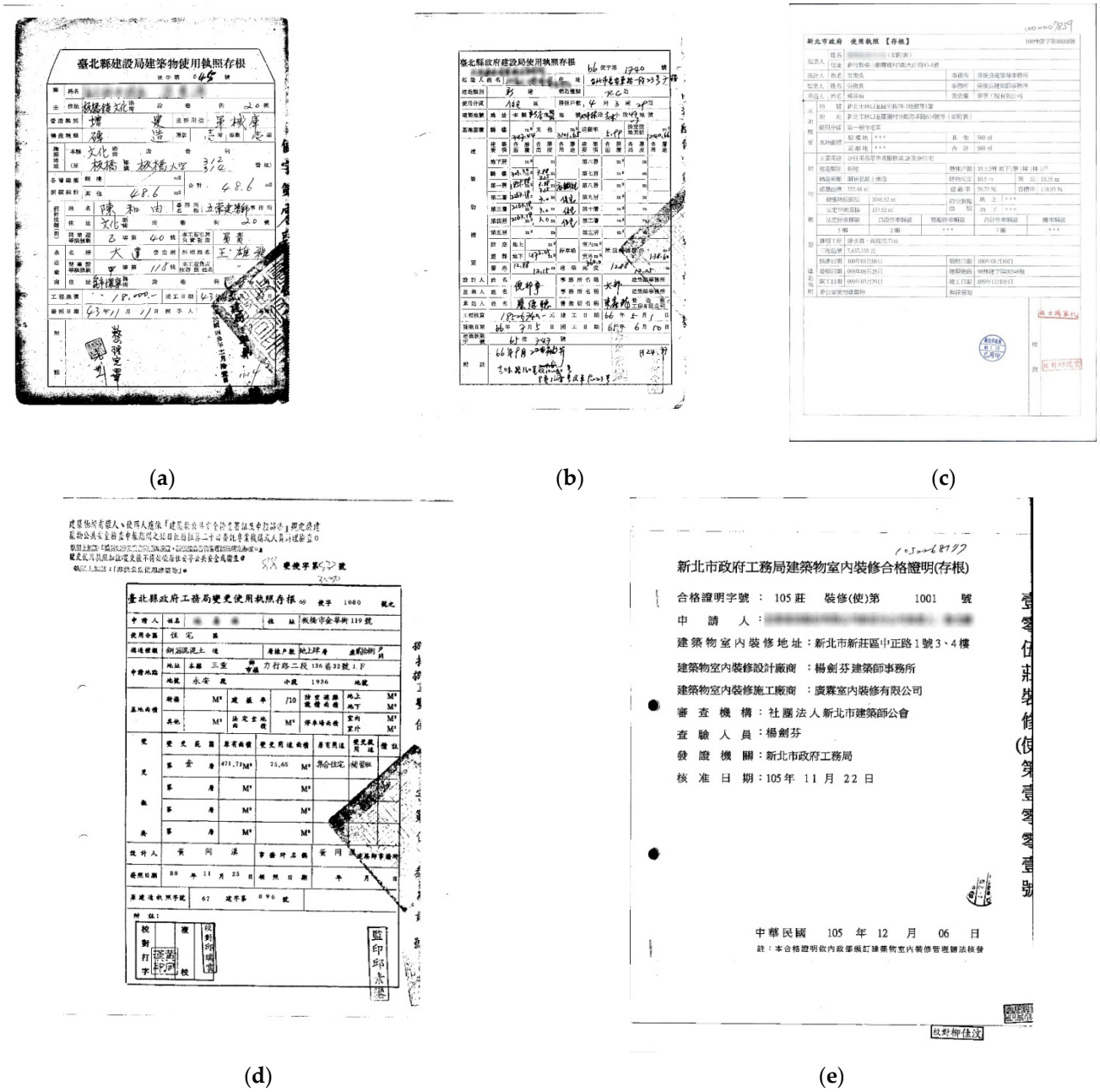


Figure 7. Some examples of document images in Taiwan: (**a**) case 1, (**b**) case 2, (**c**) case 3, (**d**) case 4, and (**e**) case 5.

5.1. Data Augmentation

As we know, the more data a machine-learning-based scheme can access, the more powerful the scheme can be. As mentioned in Section 3.2.2, synthetic images are yielded for re-training the pre-trained RetinaNet model. Then, a total of 57 fonts such as Microsoft

PMingLiu and Microsoft DFKai-SB are utilized to synthesize these key characters for synthetic image generation.

In fact, another generic and accepted way for augmenting image data is to perform geometric transformation [34]. However, we need other operations to simulate more training samples on noisy document images. To consider real environments, some operations, such as morphological dilation, noise adding, and image smoothing, are also adopted here. Figure 8 illustrates the examples of data augmentation for the key character "姓" (Surname). Figure 8b–d is generated by using morphological dilation, noise adding, and smoothing, respectively. Furthermore, some key characters are collected from document images. Then, we generate 174,726 synthetic images with a single key character and simple backgrounds. To simulate the real environment of dealing with document images, we also created 24,994 synthetic images with multiple key characters and simulated backgrounds. After synthetic image generation and data augmentation, these augmented images are divided into the training and validation sets for model training. The ratio between the training and validation sets is 80:20, respectively.

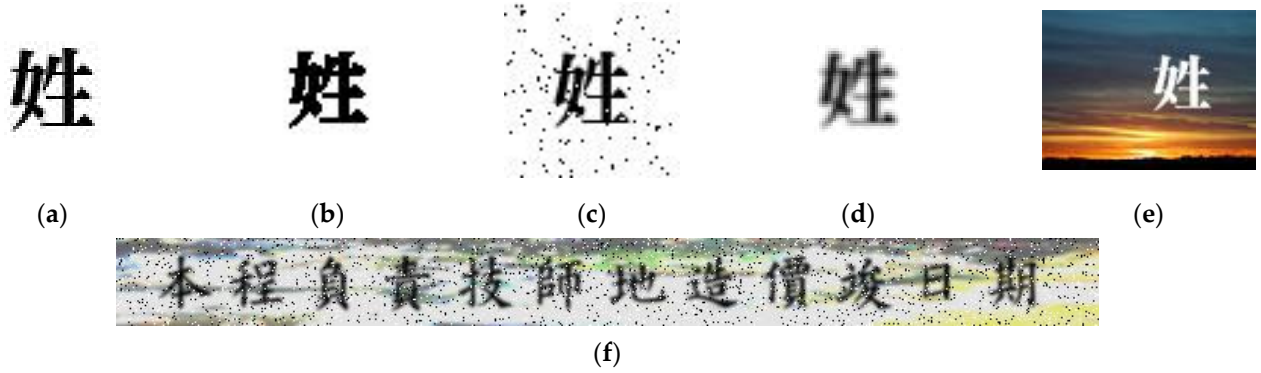


Figure 8. Examples of data augmentation: (**a**) possible font (**b**) morphological dilation, (**c**) noise adding, (**d**) smoothing, (**e**) different background, and (**f**) mixed.

5.2. Performance Index

Since key character detection is a critical process in the proposed scheme, it is important to evaluate whether the proposed scheme can locate and recognize key characters effectively in document images. Here, some measurements are selected for performance evaluation. First, IoU (intersection over union) [12] is adopted as a performance index to evaluate whether the proposed scheme can locate key characters correctly. The definition of IoU [10] is described as follows:

$$\text{IoU} = \frac{\Omega^T \cap \Omega^P}{\Omega^T \cup \Omega^P} \tag{4}$$

where Ω^T, Ω^P, $\cap$, and $\cup$ represent the exact bounding box, the predicted bounding box, intersection operator, and union operator, respectively. As shown in Equation (4), IoU means the amount of overlap between the predicted and ground truth bounding box. It is expected that the higher the IoU is, the better the performance will be. In fact, IoU is usually used when computing mAP [1,28] for performance evaluation of object detection.

Next, we explain the performance indices of recognizing key characters used in the experiment. The recall and precision rates are widely used to measure the performance of shot change detection [37,38]. We utilize them to evaluate the performance of the proposed scheme. The precision rate is the ratio of correct detections to the total number of detected key characters. The definition of precision is as follows:

$$\text{Precison} = \frac{N_C}{N_C + N_F} \tag{5}$$

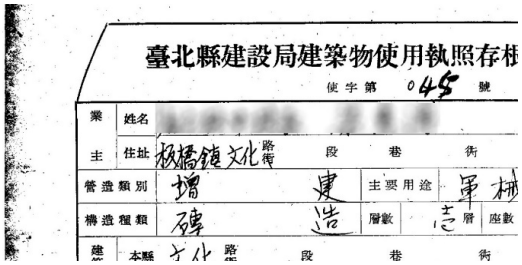
where N_C and N_F are the numbers of correct detections and false alarms, respectively, and $(N_C + N_F)$ is the total number of detected key characters. Theoretically, if a detection scheme achieves high precision rates, its performance is considered good. Here, average precision (AP) is also exploited. Furthermore, since there are some key characters that should be detected in document images, mAP [1,28] is also utilized to evaluate whether a scheme can simultaneously detect these key characters successfully. It is expected that the higher the mAP is, the better the performance will be.

5.3. Analysis of the Proposed Scheme

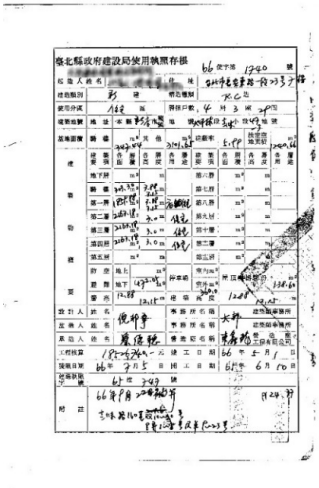
As mentioned in Section 4, the width of a character is useful information for lexicon analysis. Here, we assume that the bounding box of each key character is near a square. Some key characters are analyzed manually. After sorting the area values of these key characters, the median value of these sorted area values is 2200 pixels as the estimated area value of a character, and then the estimated width of a character is about 46 pixels, i.e., $N_W^{Ch} = 46$, as the prior information in the lexicon analysis.

Figure 9 shows the results of ROI localization by using the proposed scheme. The left and right columns of Figure 9 show the original document image and their corresponding ROIs. As we can see in the right column of Figure 9, these ROIs contain the keywords for further analysis. The experimental results demonstrate that the proposed scheme can effectively localize the ROIs in the digitized document images.

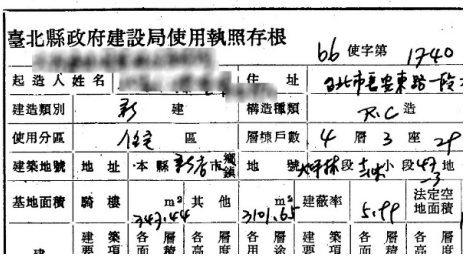
(**a1**)



(**b1**)

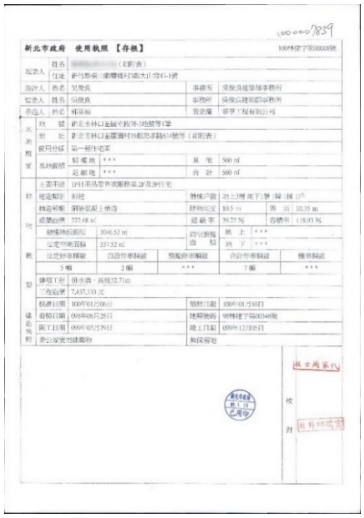


(**a2**)



(**b2**)

Figure 9. *Cont.*



(**a3**)

新北市政府 使用執照 【存根】

起造人	姓名	（如附表）		
	住址	新竹縣橫山鄉豐鄉村3鄰大山背43-1號		
設計人	姓名	吳俊良	事務所	吳俊良建築師事務
監造人	姓名	吳俊良	事務所	吳俊良建築師事務
承造人	姓名	郭芬茹	營造廠	華亨工程有限公司
基地概要	地號	新北市林口區國宅段78-3地號等1筆		
	地址	新北市林口區麗園村19鄰忠孝路614號等（如附表）		
	使用分區	第一種住宅區		
	基地面積	騎樓地 ***	其他	560 ㎡
		退縮地 ***	合計	560 ㎡

(**b3**)

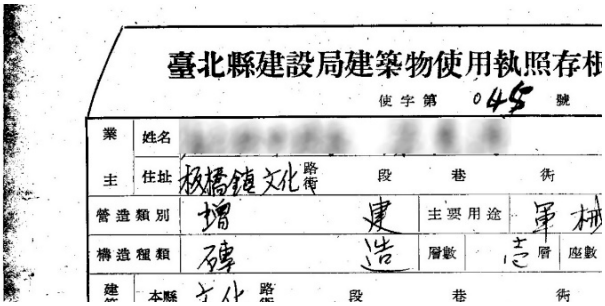
Figure 9. Results of ROI initialization; (**a1**–**a3**): input, (**b1**–**b3**): output.

5.3.1. Key Character Detection

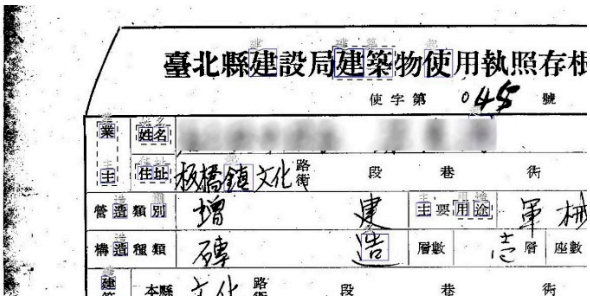
Key character detection is an important step in the proposed scheme. Here, N^{KC} is 16, and we evaluate the performance of the proposed character detection algorithm. Figure 10 illustrates the results of the proposed key character detection and keyword-detection algorithms. The left and right columns of Figure 10 are the ROIs and the results, respectively. Figure 11 illustrates the local detection results of the proposed scheme. In Figures 10 and 11, each rectangle with a solid line is the bounding box of RetinaNet for key character detection, and each rectangle with a dashed line is the output of the proposed keyword detection. As we can see in Figure 10, these ROIs have different image contents, some ROIs may have noise, and the quality of some key characters, e.g., "姓" (Surname) and "名" (First Name), may vary in the ROIs. Though the image quality of some ROIs may not be good, the proposed scheme can detect and recognize the key characters in document images with different contents. From Figure 10b4,b5, the key characters, "申" (Apply), "請" (Apply), and "人" (Applicant), can be detected well even when their fonts are different in ROIs. This means that ResNet in RetinaNet can effectively extract the useful and representative features from the training data, and these features are robust even when noise or poor visual quality happens.

Furthermore, though the key characters "建" (Build) have different sizes in Figure 10b1,b2 most can be correctly detected by using the proposed scheme. This means that ResNet can extract useful feature maps, and FPN is useful to yield the multi-scale feature maps for detecting key characters with different sizes. Therefore, the results show that the proposed scheme based on RetinaNet can detect these key characters well, even in situations of poor visual quality, different fonts, and different sizes.

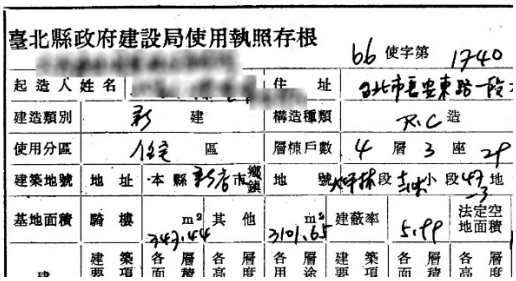
Figure 12 shows two cases with incorrect detection results. The left and right columns of Figure 12 are the ROIs and the key character detection results, respectively. Dashed-line circles are used to point out wrong detection results. As we can observe in Figure 12b1, the visual quality of some key characters "申" (Apply) is poor, so they cannot be detected correctly. On the top of Figure 12b2, one key character, "築" (Building), is not detected. In addition, some key characters are incorrectly classified in Figure 12. For example, two characters "第" (prefix indicating ordinal number) and "第" (Floor) are misclassified as the key characters "築" (Building) and "用" (Use) in Figure 12b1, respectively. Similarly, one character "中" (Median) and one digit "0" are wrongly determined as key characters "申" (Apply) and "用" (Use) in Figure 12b2, respectively.



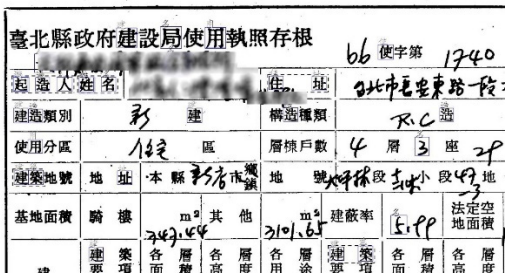
(**a1**)



(**b1**)

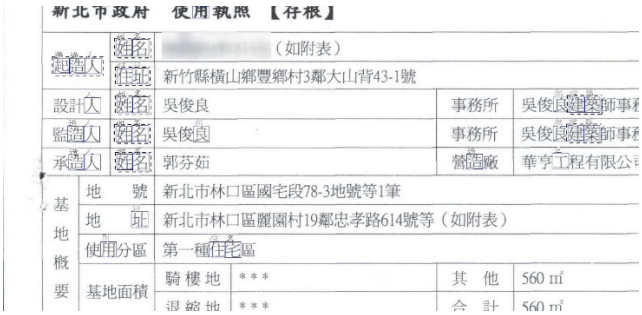


(**a2**)

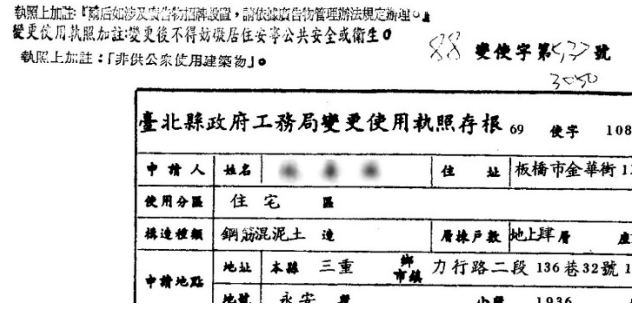


(**b2**)

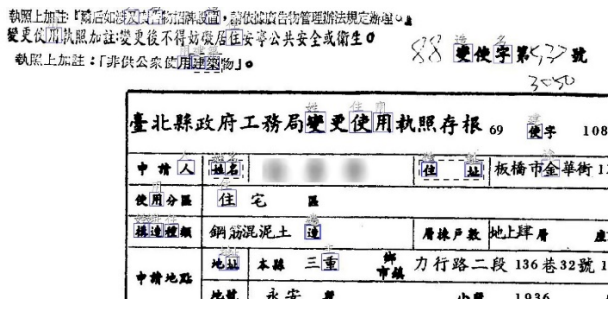
(**a3**)



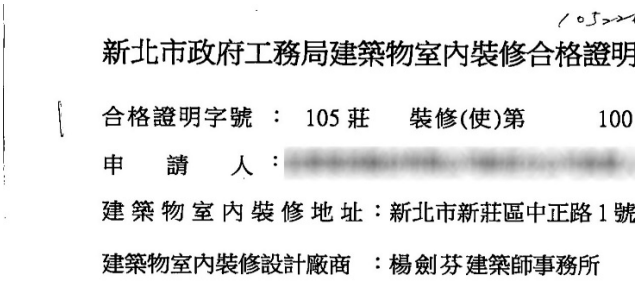
(**b3**)



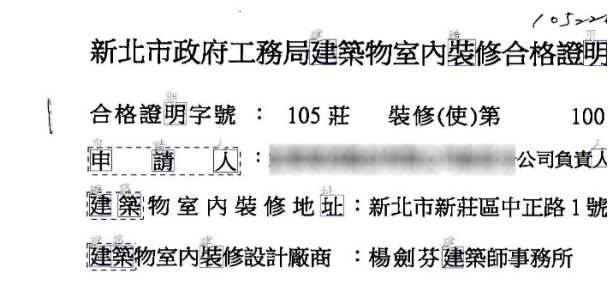
(**a4**)



(**b4**)

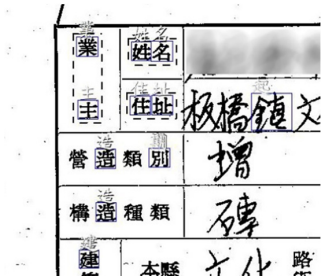


(**a5**)

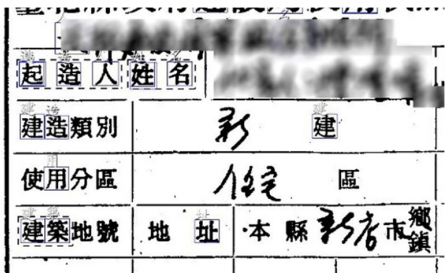


(**b5**)

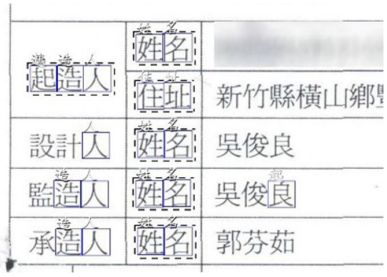
Figure 10. Results of key character detection and keyword detection. (**a1**–**a5**): ROI; (**b1**–**b5**): results.



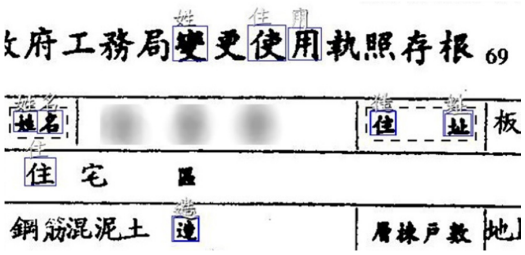
(**Lb1**)



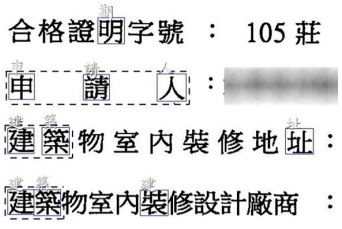
(**Lb2**)



(**Lb3**)

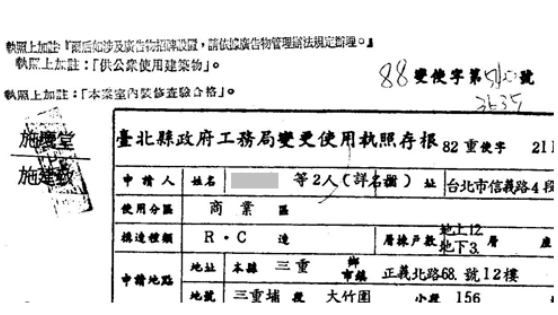


(**Lb4**)



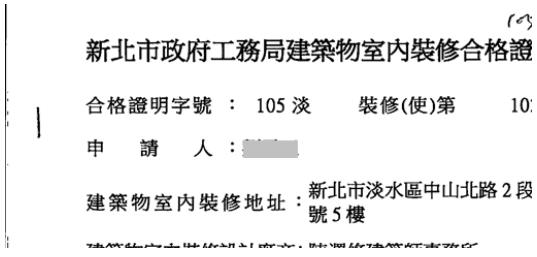
(**Lb5**)

Figure 11. Local detection results of Figure 10b1–b5 for the proposed scheme: (**Lb1–Lb5**): local results of Figure 10b1–b5.

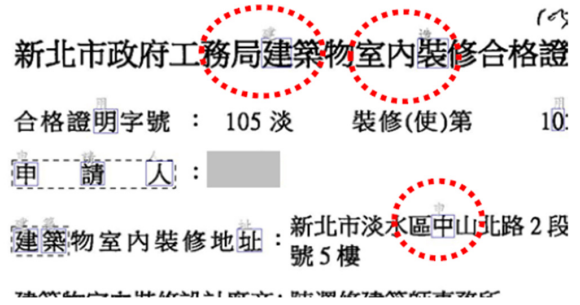


(**a1**)

(**b1**)



(**a2**)



(**b2**)

Figure 12. Two cases with wrongly detected characters in the key character detection. (**a1–a2**): ROIs; (**b1–b2**): results.

Here, mAP is utilized to assess the performance of the proposed scheme for all key characters. To compute the mAP rate, the thresholds of confidence and IoU are determined as 0.3 and 0.5, respectively. Figure 13 shows the AP rates of the proposed key-character-detection algorithm. As we can observe in Figure 13, the AP rates of the common key characters, "姓" (Surname), "名" (First Name), "住" (Live), and "址" (Site), are at least 0.9. In addition, as shown in Figure 13, most AP rates are higher than 0.82, except for two key characters, "申" (Apply) and "築" (Building). According to Figure 10a4, Figure 11, and Figure 12, the main reason is that the visual quality of the two key characters, "申" (Apply) and "築" (Building), is poor, which consequently causes the lower AP rates of the proposed key-character-detection algorithm. Furthermore, the mAP of the proposed scheme is 85.1% for all of the key characters. The experimental results show that the proposed scheme based on RetinaNet can locate and recognize the key characters successfully, even when the quality of input images is not good.

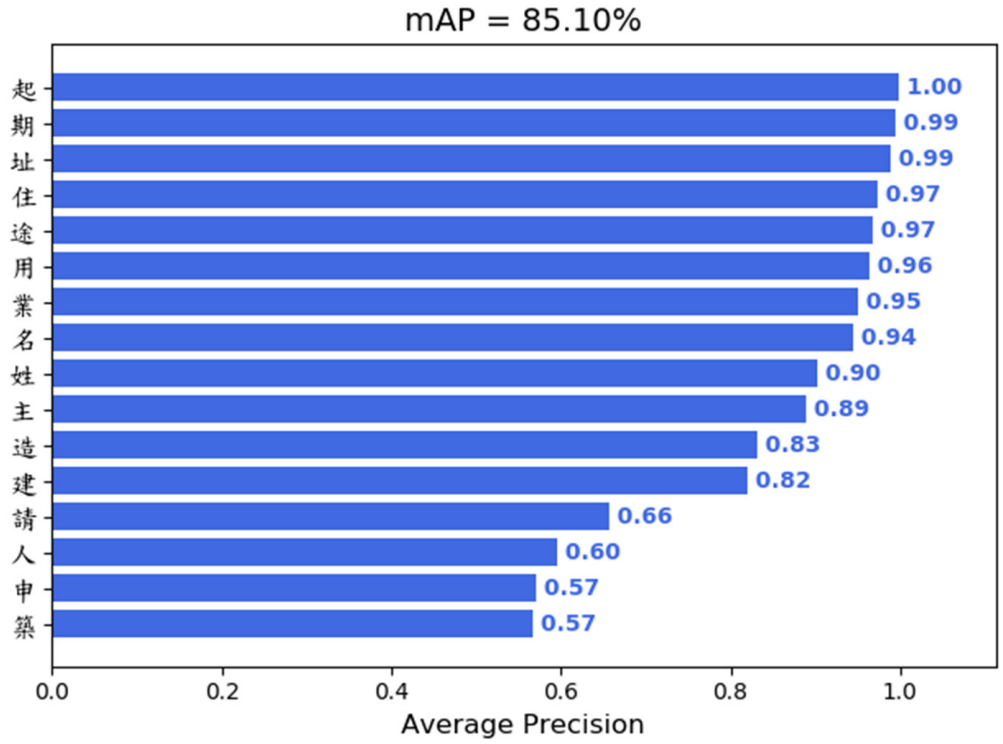


Figure 13. The AP rates of the proposed key-character-detection algorithm.

5.3.2. Keyword Detection

Here, N^{KW} is seven, and we evaluate the performance of the proposed keyword-detection algorithm. Figure 10 also shows the results of the proposed scheme for keyword detection. In the right columns of Figure 10, the dashed rectangle is the result of the proposed scheme for keyword detection. The local results are also shown in Figure 11. The threshold of IoU is set to 0.5. As we can see in Figure 10, some keywords, e.g., "姓名" (Name), "住址" (Address), in Figure 10b1 and some keywords, e.g., "申請人" (Applicant), Figure 10b4,b5, are arranged in the horizontal direction. These keywords, such as "姓名" (Name), "住址" (Address), and "申請人" (Applicant), can be located and recognized well. Furthermore, "業主" (Property Owner), in Figure 10b1, is arranged in the vertical direction. The keyword "業主" (Property Owner) can also be detected well. This means that the lexicon analysis is useful to detect keywords in horizontal and vertical directions.

Figure 14 illustrates the AP results of the proposed keyword-detection algorithm. As shown in Figure 14, most AP rates are higher than 0.89, except for two keywords, "申請人" (Applicant) and "建築" (Building). According to Figures 10–13, the main reason is that the proposed scheme may not detect the two key characters, "申" (Apply) and "築"

(Building), of poor quality, as well as the two keywords, "申請人" (Applicant) and "建築" (Building). In addition, the mAP of the proposed scheme is 85.84%, as shown in Figure 14. This means that the experimental result shows that the lexicon analysis can effectively find some suitable key characters, and then the keywords can be obtained by analyzing the combination of these key characters. Therefore, the experimental results demonstrate that the proposed keyword-detection algorithm can effectively locate and recognize the keywords in document images with different image qualities.

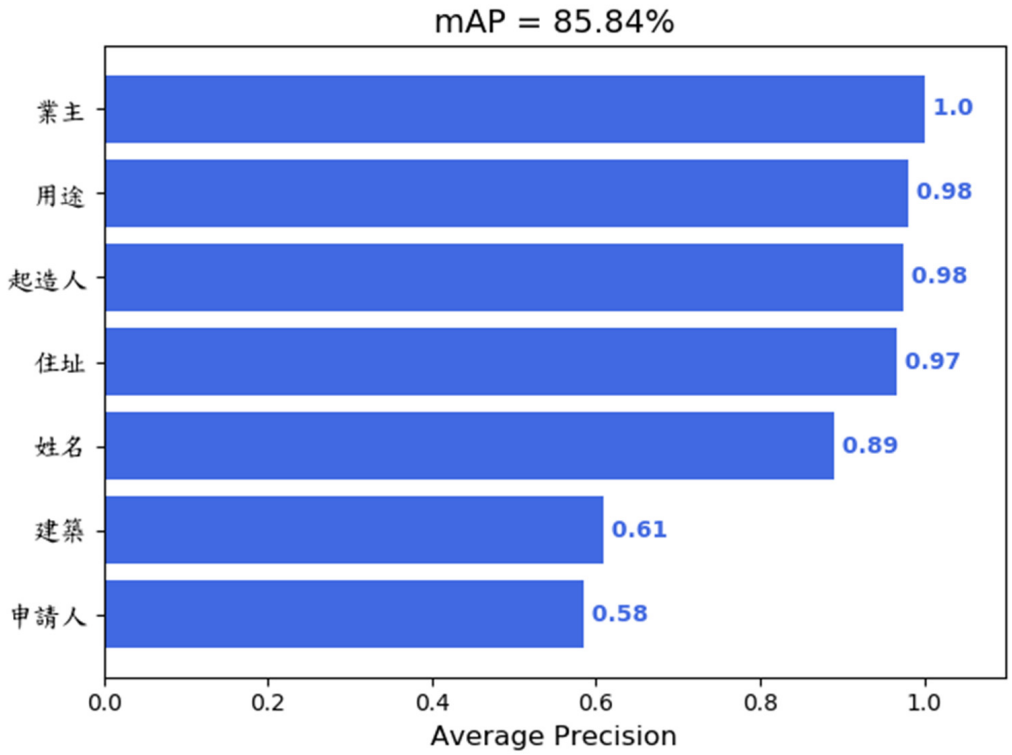


Figure 14. The AP results of the proposed keyword-detection algorithm.

As for the execution time, the proposed scheme can deal with each document image about 0.2 frames per second. It is expected that the execution time of the proposed scheme can be reduced if our program is optimized or a powerful graphics card is used.

5.4. Comparison with Tesseract for Key Character Recognition

As we know, Tesseract OCR (Optical Character Recognition) software provided by Google is well-known for character recognition. To compute the AP rate, the threshold of IoU is determined as 0.5. Here, there are 112 ROIs extracted from the testing set for performance evaluation. Figure 15 shows the AP rates of the proposed scheme and Tesseract OCR software. As we can see in Figure 15, the proposed scheme can detect these key characters from noisy document images and is compared with Tesseract OCR software. Furthermore, the AP rates of the Tesseract OCR tool are at most 36.2%, shown in Figure 15. This means that the Tesseract OCR tool cannot deal with noisy document images well.

The AP rates of the proposed scheme are at least 56% for key characters in noisy document images. Therefore, the experimental results demonstrate that the proposed scheme is superior to Tesseract OCR software for detecting key characters in noisy document images.

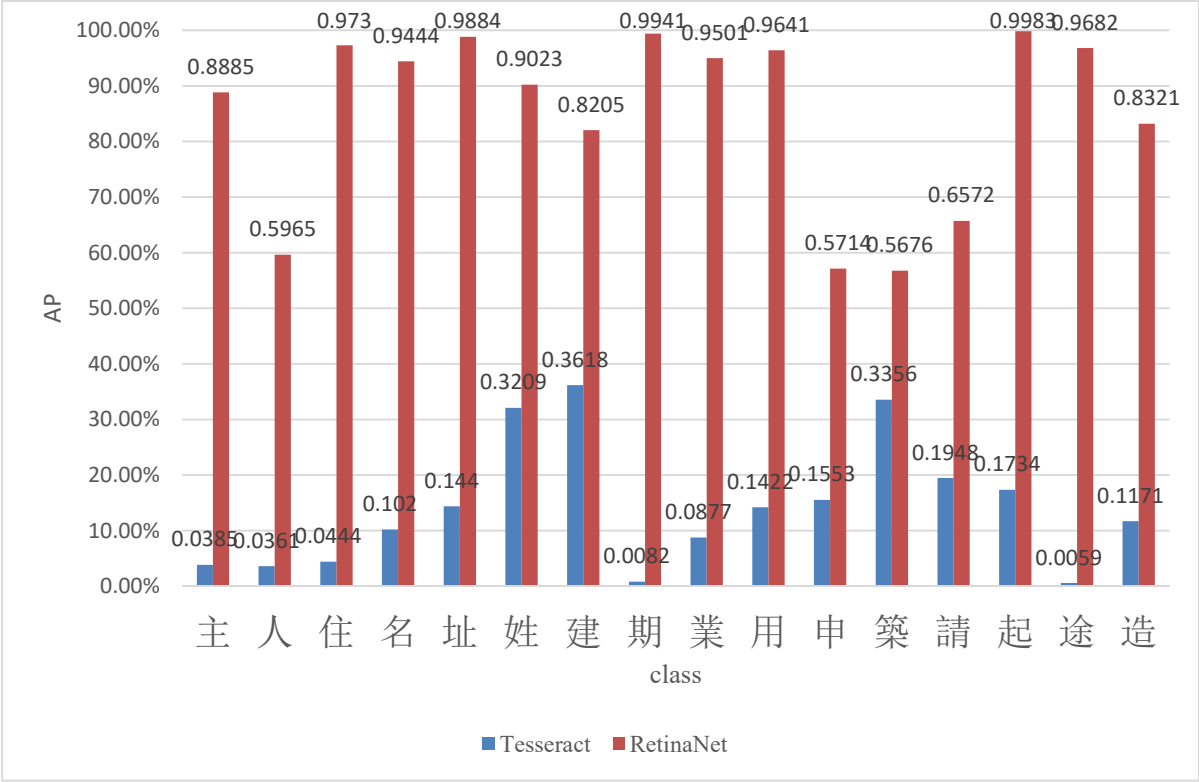


Figure 15. The AP results of the proposed scheme and Tesseract OCR software.

6. Discussions

To re-train the model of RetinaNet with a pre-trained model, synthetic image generation and data augmentation are exploited to yield a large image dataset. Then, a testing set of 112 images is used for performance evaluation.

As for key character detection, Figure 13 shows the AP rates of the proposed key-character-detection algorithm. As shown in Figure 13, most AP rates are higher than 0.82, except for two key characters, "申" (Apply) and "築" (Building). According to Figures 10a4, 11, and 12, the main reason is that the visual quality of the two key characters, "申" (Apply) and "築" (Building), is poor, which consequently causes the lower AP rates of the proposed key-character-detection algorithm. Furthermore, the mAP of the proposed scheme is 85.1% for all of the key characters. The experimental results show that the proposed scheme based on RetinaNet can locate and recognize the key characters successfully even though when the quality of input images is not good.

As for keyword detection, Figure 14 illustrates the AP results of the proposed keyword-detection algorithm. As shown in Figure 14, most AP rates are higher than 0.89, except for two keywords, "申請人" (Applicant) and "建築" (Building). According to Figures 10–13, the main reason is that the proposed scheme may not detect the two key characters, "申" (Apply) and "築" (Building), of poor quality, as well as the two keywords, "申請人" (Applicant) and "建築" (Building). In addition, the mAP of the proposed scheme is 85.84%, as shown in Figure 14. This means that the experimental result shows that the lexicon analysis can effectively find some suitable key characters, and the keywords can be obtained by analyzing the combination of these key characters. Therefore, the experimental results demonstrate that the proposed keyword-detection algorithm can effectively locate and recognize the keywords in document images with different image qualities.

The execution time of the proposed scheme is about 0.2 frames per second for the experiment platform. It is expected that the execution time of the proposed scheme can be reduced if our program is optimized or a powerful graphics card is used.

Since the proposed scheme is developed for some specific document images, such as building use permits in Taiwan, the key characters and the keywords are determined according to our experiences and analysis. This means that the proposed scheme is only suitable to deal with some specific document images, such as building use permits in Taiwan.

7. Conclusions and Future Work

In this paper, a keyword detection scheme was proposed based on deep convolutional neural networks for personal information protection in document images. The proposed scheme is composed of key character detection and lexicon analysis. The first part is key character detection developed based on RetinaNet and transfer learning. To find the key characters, RetinaNet, composed of convolutional layers, feature pyramid network, and two subnets, is exploited to detect key characters within the region of interest in a document image. After the key character detection, lexicon analysis is used where the detected key characters are analyzed, and it combines them to obtain the keywords.

To train the model of RetinaNet, synthetic image generation and data augmentation are exploited to yield a large image dataset. To evaluate the proposed scheme, IoU (Intersection Over Union) and mAP (Mean Average Precision) are utilized as performance measurements. For performance evaluation, many document images with different types are selected for testing. The experimental results show that the proposed scheme can detect these key characters well even under situations of poor visual quality, different fonts, and different sizes. The mAP rates of the proposed scheme are 85.1% and 85.84% for key character detection and keyword detection, respectively. Furthermore, the proposed scheme is superior to Tesseract OCR (Optical Character Recognition) software for detecting the key characters in document images with Mandarin Chinese words. The experimental results demonstrate that the proposed method can effectively localize and recognize these keywords within document images with Mandarin Chinese words.

Since the proposed scheme is developed for specific document images, such as building use permits in Taiwan, the future work should increase the number of keywords to deal with more kinds of document images in Taiwan. Furthermore, detecting and analyzing the keywords in documental images can be utilized for document image analysis and document classification.

Author Contributions: Conceptualization, G.-S.L.; Data curation, J.-C.T.; Formal analysis, G.-S.L. and J.-Y.L.; Methodology, G.-S.L.; Software, J.-C.T.; Supervision, G.-S.L. All authors have read and approved the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by Ministry of Science and Technology (MOST), Taiwan, under the Grants MOST 108-2622-E-167-012-CC3. In addition, the authors wish to acknowledge the help of SyncThreads Technology Inc., Taiwan, in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wiggers, K.L.; Britto, A.S.; Heutte, L.; Koerich, A.L.; Oliveira, L.E.S. Document Image Retrieval Using Deep Features. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
2. Zagoris, K.; Ergina, K.; Papamarkos, N. A Document Image Retrieval System. *Eng. Appl. Artif. Intell.* **2010**, *23*, 872–879. [CrossRef]
3. Shin, C.K.; Doermann, D.S. Document Image Retrieval Based on Layout Structural Similarity. In Proceedings of the International Conference on Image Processing, Computer Vision, & Pattern Recognition, Las Vegas, NV, USA, 26–29 June 2006.
4. Lin, G.-S.; Tuan, N.-M.; Chen, W.-J. Detecting Region of Interest for Cadastral Images in Taiwan. *Multimed. Tools Appl.* **2017**, *76*, 25369–25389. [CrossRef]
5. Lin, G.-S.; Chai, S.-K.; Li, H.-M.; Lin, J.-Y. Vision-Based Patient Identification Recognition Based on Image Content Analysis and Support Vector Machine for Medical Information System. *EURASIP J. Adv. Signal. Process.* **2020**, *2020*, 1–15. [CrossRef]

6. Afzal, M.Z.; Kölsch, A.; Ahmed, S.; Liwicki, M. Cutting the error by half: Investigation of very deep CNN and advanced training strategies for document image classification. In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017.
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst. NIPS* **2012**, *25*, 1097–1105. [CrossRef]
8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* **2015**, arXiv:1512.03385.
9. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. *arXiv* **2014**, arXiv:1411.4038.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards realtime object detection with region proposal networks. *Adv. Neural Inf. Process. Systems.* **2015**, *28*, 91–99. [CrossRef]
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. SSD: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the CVPR, Las Vegas, NV, USA, 26 June–1 July 2016.
13. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the CVPR, Honolulu, HI, USA, 21–26 July 2017.
14. Kölsch, A.; Afzal, M.Z.; Ebbecke, M.; Liwicki, M. Real-Time Document Image Classification using Deep CNN and Extreme Learning Machines. *arXiv* **2017**, arXiv:1711.05862v1.
15. Oliveira, S.A.; Seguin, B.; Kaplan, F. dhSegment: A generic deep-learning approach for document segmentation. *arXiv* **2018**, arXiv:1804.10371v1.
16. Oliveira, D.A.B.; Viana, M.P. Fast CNN-based document layout analysis. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 1173–1180.
17. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. EAST: An Efficient and Accurate Scene Text Detector. *arXiv* **2017**, arXiv:1704.03155v2.
18. Guo, J.; Chang, L.; Lee, H. DDSnet: A Deep Document Segmentation with Hybrid Blocks Architecture Network. In Proceedings of the International Symposium on Computer, Consumer and Control (IS3C), Taichung City, Taiwan, 13–16 November 2020.
19. Iqbal, A.; Basit, A.; Ali, I.; Babar, J.; Ullah, I. Automated Meter Reading Detection Using Inception with Single Shot Multi-Box Detector. *Intell. Autom. Soft Comput.* **2021**, *27*, 299–309. [CrossRef]
20. Son, C.; Park, S.; Lee, J.; Paik, J. Deep Learning–based Number Detection and Recognition for Gas Meter Reading. *IEIE Trans. Smart Process. Comput.* **2019**, *8*, 367–372. [CrossRef]
21. Gómez, L.; Rusinol, M.; Karatzas, D. Cutting Sayre's Knot: Reading Scene Text without Segmentation. Application to Utility Meters. In Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria, 24–27 April 2018.
22. Laroca, R.; Barroso, V.; Diniz, M.A.; Gonçalves, G.R.; Schwartz, W.R.; Menotti, D. Convolutional Neural Networks for Automatic Meter Reading. *arXiv* **2019**, arXiv:1902.09600v1. [CrossRef]
23. Zhao, Z.; Zheng, P.; Xu, S.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef]
24. Shrivastava, A.; Gupta, A.; Girshick, R. Training region based object detectors with online hard example mining. In Proceedings of the CVPR, Las Vegas, NV, USA, 26 June–1 July 2016.
25. Shrivastava, A.; Sukthankar, R.; Malik, J.; Gupta, A. Beyond skip connections: Top-down modulation for object detection. *arXiv* **2016**, arXiv:1612.06851.
26. Gishick, R. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083v2.
27. Lin, T.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**, arXiv:1708.02002v2.
28. Li, Y.; Ren, F. Light-Weight RetinaNet for Object Detection. *arXiv* **2019**, arXiv:1905.10011v1.
29. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2017**, arXiv:1612.03144.
30. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767v1.
31. Bochkovskiy, A.; Wang, C.; Liao, H.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934v1.
32. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
33. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
34. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
35. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic Data for Text Localisation in Natural Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
36. Kingma, P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2005; pp. 1–13.
37. Lin, G.S.; Chang, M.K.; Chiu, S.T. A feature-based Scheme for detecting and classifying video-shot transitions based on spatio-temporal analysis and fuzzy classification. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 1179–1200. [CrossRef]
38. Lin, G.S.; Chang, M.K.; Chen, Y.L. A passive-blind scheme for image forgery detection based on content-adaptive quantization table estimation. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 421–434. [CrossRef]