FORENSIC SCIENCE JOURNAL SINCE 2002

Forensic Science Journal
2020;19(1):9-22
fsjournal.cpu.edu.tw
DOI:10.6593/FSJ.202012_19(1).0002

An Improvement of the Interpretability for the Deep Learning Based Signature Examination Assistance Method

Hsin-Hsiung Kao [1,2], M.S. ; Che-Yen Wen [1*], Ph.D. ; Kai-Ping Chang [3], M.S.

[1] Department of Forensic Science, Central Police University, 56 Shuren Rd., Guishan Dist., Taoyuan City, Taiwan 333322 R.O.C.
[2] Science and Technology Crime Prevention Center, Criminal Investigation Bureau
[3] Forensics Science Center, Criminal Investigation Bureau

Received: September 23, 2020; Accepted: October 26, 2020.

Abstract

Automatic signature verification has been extensively researched for a long time and has already been used in many fields like banking, security, and other authentication purposes. However, human experts still play a dominant role in the field of forensic handwriting examination. Only a few studies have been conducted on the use of computers in assisting handwriting experts. There are also fewer attempts to perform the examination based on only a single known sample.

We built a deep learning based assistance method for signature examination in our previous work. The method can deal with the problem of signature verification by single known sample, and is based on an explainable deep learning approach (by using deep convolutional neural network, DCNN). This paper is a continuation and refinement of our previous work. We refine the interpretability of the model and present application scenarios for assisting signature examination. After improving the interpretability of the model, the proposed method can be used as an assistant system by providing quantitative results. The visualized heatmaps can also be used to identify genuine or suspicious strokes in disputed signatures.

Keywords: ***handwritten signature verification, convolutional neural network, explainable deep learning, computer assistance system***

Introduction

Handwritten signature verification is one of the most important biometric technology and has been widely applied in commercial authentication and forensic examination. In practice, the signature examination is carried out by trained experts, especially in court. To improve reliability and efficiency, automatic signature verification has been studied [1]. However, compared with other biometric methods, handwritten signature verification has the properties of relatively high intra-class variability (the variability among an individual's genuine signatures) and low inter-class variability (the variability between genuine signatures and skilled forgeries).Even for human examiners, it is not an easy work of distinguishing between genuine signatures and forgeries. For

*Corresponding author: Che-Yen Wen, Department of Forensic Science, Central Police University, No.56, Shuren Rd., Guishan Dist., Taoyuan City 333322, Taiwan (R.O.C.) Fax: 886-3-3275907
E-mail: cwen@mail.cpu.edu.tw

these reasons, although automatic signature verification has been widely studied, it remains a challenging problem.

Automatic off-line signature verification methods can be classified into two categories: handcrafted feature extractors and deep learning approaches [1]. Recently, the deep learning approaches have shown the great capability of image recognition and detection. For example, the convolutional neural network (CNN) is a specialized type of neural networks which apply the convolution function to feature extraction. CNN is widely used in image recognition and other related fields.

Deep learning algorithms are effective when applied to large-scale datasets. Therefore, most of learning based signature verification studies require several (more than one) signature samples to train the networks [2-5]. Khalajzadeh et al. [1] proposed a deep CNN method for author classification, which can directly learn features from signature image pixels. Hafemann et al. [6] used a CNN based method that can extract stable features from variable-size signatures.

Although deep learning-based approaches have shown their great capability in signature verification and other pattern recognition areas, the sample size limitation has led to an critical problem. We can improve reliability of approaches with vast amounts of reference samples, but the cost for sample collecting and system implementation is high. In some applications, such as signature verification, it is usually not easy for us to get a lot of sample data. For these reasons, the idea of using small scale datasets for deep learning has gotten considerable attention in recent years.

Signature verification with single reference signature is a challenging task due to the large intra-class variability. That is the reason that the single reference based approachs are more applicable in practical applications, but it has still attracted less attention. Adamski and Saeed [7] proposed a sampling algorithm to acquire the vector based feature from a preprocessed one-pixel-wide signature. Their algorithm is based upon a traditional handcrafted feature extraction method. Unfortunately, their method can only deal with random forgeries (other unrelated signatures), so it is not capable of detecting skilled forgeries.

Our work tries to address the following two issues that deserve attention. First, handwritten forensic examiners need a suitable computer-aided examination tool that can provide a quantitative assessment. Secondly, it is difficult for front-line investigators to obtain sufficient reference signature samples. When the case is still under preliminary investigation, front-line investigators also need an examination tool to assess the genuineness of the questioned signature samples quickly (based on a small number of samples).

To overcome the problem of insufficient genuine samples when using deep learning methods, we choose an alternative strategy to make our training feasible. First, our method is based upon local features instead of the whole signature image. It is very different from previous signature verification research work. We divide a signature image into many overlapping sub-image blocks and apply a series of data augmentation techniques to initially expand the training samples. More details will be presented in the subsection “*Data preprocessing.*” Since we are dealing with a binary classification task (genuine and forged). We can shift the focus of our system to learn “what is forged signature” and let the networks learn useful features from a lot of forged signatures, which are relatively easy to obtain and can even be created by researchers. Once our system can detect the features of forgeries, our goal of distinguishing between genuine and forged is basically achieved. Secondly, a lot of deep learning-based signature verification methods have already proven their superior performance. However, none of them can be applied to practical forensic document examination. The most important problem is that deep learning methods are generally opaque and lack explainability, which means they cannot explain why and how they make a specific decision (genuine and forged). Consequently, the main purpose of our work is to enhance the explainability by using visual interpretation techniques, and present the visualization results to handwriting experts.

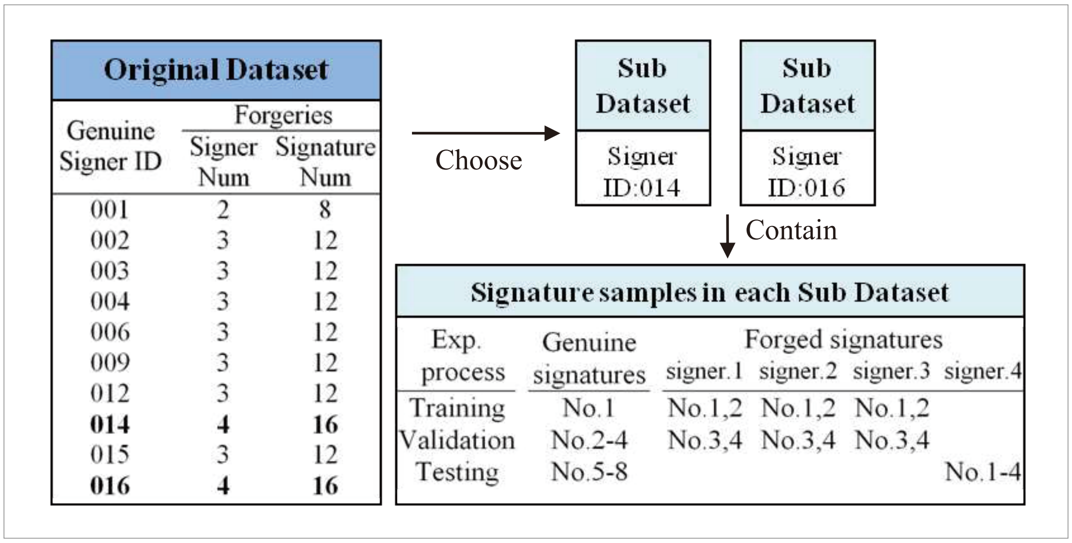
We have built a deep learning based method (with the supervised learning) to automatically detect forged signatures in our previous work [8]. The method is to deal with the problem of signature verification by single known sample. As an assistant system for forensics, the proposed method can classify the disputed signature as the skilled forgery or the genuine one. In this paper, we refine the explainability of the model and present application scenarios for assisting signature examination.

Materials and Methods

Materials

Our experimental signature samples are collected from ICDAR 2011 SigComp dataset [8]. This dataset is used for signature verification competition, and its offline section contains different sample sizes of skilled forgeries for each genuine signer. Each genuine signer corresponds to 2~4 forged signers, and each forged signer contributes four skilled forgeries.

The data collection procedure is based on our previous work [9]. Since ID 014 and ID 016 signers' signatures in the dataset have the largest number of forged reference signatures (16 skilled forgeries from 4 signers), we can use them for both processes of training the network and evaluating the performance, as shown in Fig. 1.



Original Dataset

Genuine Signer ID	Forgeries	
	Signer Num	Signature Num
001	2	8
002	3	12
003	3	12
004	3	12
006	3	12
009	3	12
012	3	12
014	**4**	**16**
015	3	12
016	**4**	**16**

Signature samples in each Sub Dataset

Exp. process	Genuine signatures	Forged signatures			
		signer.1	signer.2	signer.3	signer.4
Training	No.1	No.1,2	No.1,2	No.1,2	
Validation	No.2-4	No.3,4	No.3,4	No.3,4	
Testing	No.5-8				No.1-4

Fig. 1 Data collection: sub-datasets used for the experiment [9,10].

In Fig. 1, each sub-dataset (ID 014 and ID 016) contains eight genuine signatures from a genuine signer (No.1~8 in genuine signatures) and sixteen forged signatures from four different signers (each signer provides four forged signatures). Please note that the last signer (signer 4) is only used to evaluate the effectiveness of the network, and does not participate in the training and verification stages.

CNN Architecture

A convolutional neural network (CNN) [11] is a branch of deep learning networks and has demonstrated great success in many computer vision applications, such as image classification, pattern recognition, object detection, etc. Moreover, in the image recognition area, several researchers have claimed that their CNN approaches can achieve human-level [12] or even superhuman performance [13-15]. In this paper, we use Inception V3 architecture [16] for our experiments. The research work by Simonyan and Zisserman [17] has shown that the depth of CNN network plays a crucial role in classification accuracy. Inception V3 is a very deep CNN architecture developed by Google research team and has already proven its great performance in ImageNet object classification competition [15]. The CNN used in our work consists of following components: convolutional layer, pooling layer, dropout layer, fully-connected layer, and sigmoid layer, as illustrated in Fig. 2.

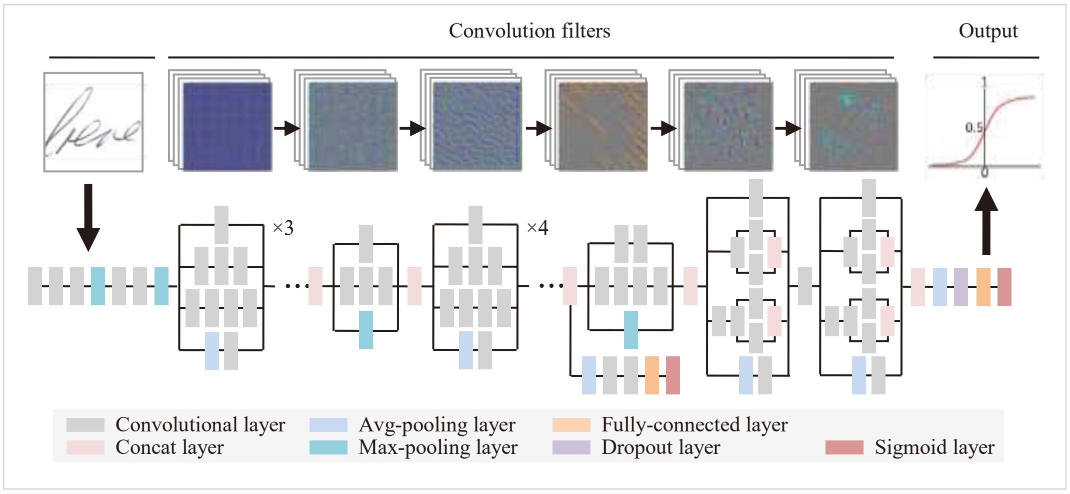


Fig. 2 The schematic diagram of CNN architecture (Inception V3) used in our experiment [16].

The convolutional and pooling layers run the feature extraction work (will be described later). The concat layer merges the multiple inputs from the previous layers. The fully connected layer is basically a traditional multilayer perceptron (MLP) neural network that used for the classification task. In the dropout layer, about 50% of perceptrons will be temporarily ignored (drop out) randomly during the training process. This technique is widely used to prevent neural networks from overfitting.

Finally, considering that our network model is designed to solve the binary classification problem (genuine and forged). We use the sigmoid function as the output layer to handle our binary classification outputs (instead of the softmax function, which is originally used in Inception v3 architecture). The sigmoid function generates an s-shaped curve whose values lie between 0 and 1. It can be defined as follow:

$$\text{p} = \text{Sigmoid}\,(x) = \frac{1}{1 + e^{-x}}\ .$$

In this work, the output value of sigmoid layer (p) represents the probability of genuine signature, and the probability of the forged category is (1-p).

Convolutional Layer

The CNN architecture is mainly constituted by convolutional layers [18]. Each convolutional layer contains multiple convolution filters to extract high-level features from low-level information. In our signature verification system, the convolution filter can gradually detect edges, corners, connection points, and other higher-order features (e.g. quality of strokes) from the original signature image.

The convolution filter can be seen as a sliding-window to path over the entire image. The sliding area is multiplied by the filters and its sum is saved as a new feature map pixel. Also, to prevent imperfect border overlays, the border pixels are filled with zero values, as shown in Fig. 3.

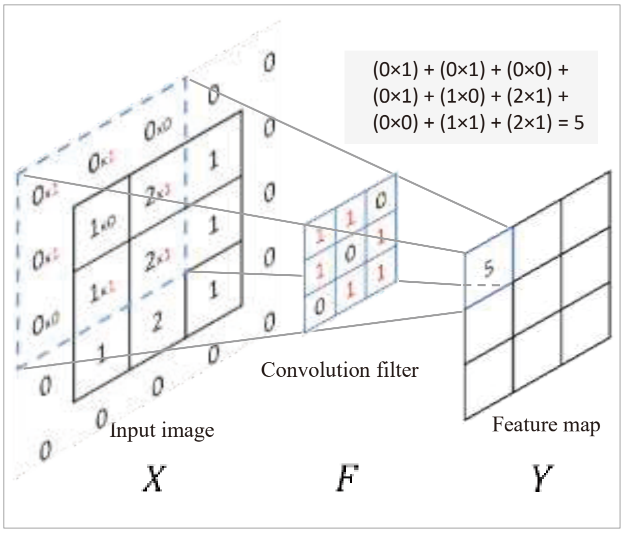


Fig. 3 An example of a convolution filter with zero padding.

Pooling Layer

Pooling layers play an important role in CNN [19-21]. In a CNN, intermediate layers serve as the features generator for the previous layer and flow their outputs to the next layer. Due to the large number of convolution operations in CNN, the size of the feature map will grow dramatically and greatly increase the computation cost. Therefore, we need the pooling layer to reduce the size of feature maps. And this leads to a faster convergence rate as well as a better performance for training networks.

In this paper, we use two different types of pooling layers: max-pooling and average-pooling, as shown in Fig. 4.

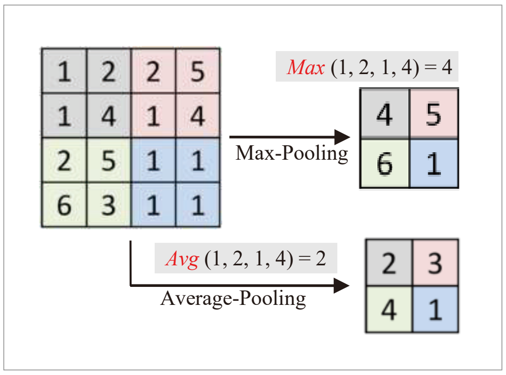


Fig. 4 The pooling process with a 2×2 filter.

Fig. 4 illustrates the difference between the max-pooling and average-pooling processes. Theoretically, the max-pooling process can reduce background noise and extract the most important texture features. Whereas the average-pooling process extracts feature more smoothly, and thus more overall information can be reserved to the next layer for further feature extraction.

Explainable Deep Learning

Most deep learning-based models are complex and work as a black-box. This results in deep neural networks' decisions are generally opaque. In order to make deep learning more explainable, the related research is called explainable AI (or XAI) and has attracted more and more attention in recent years. The gradient-based visual saliency method [22] is one of the most important approaches for XAI in the computer vision field. Its main idea is to visualize the decision-making process by marking the image pixels which are sensitive to the neural networks. This is the main idea of the saliency map.

The saliency map is generated by calculating the gradient of category-specific scores from a given classifier. The gradient indicates how much the change in a pixel will influence the classifier output. In our work, the gradient map itself can be regarded as a saliency map. The saliency map can be used to check whether the network's decision is consistent with human cognition, as shown in Fig. 5.

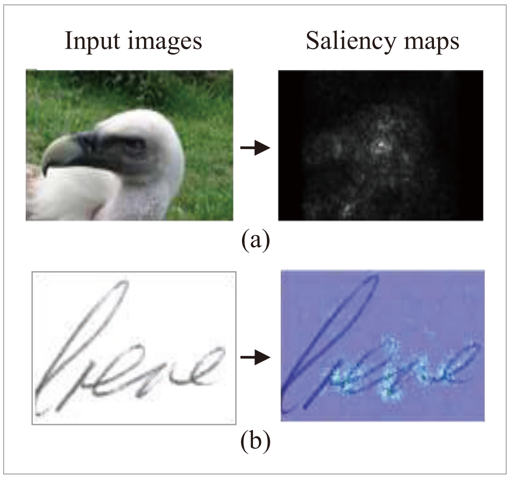


Fig. 5 Saliency maps for the CNN classifiers. (a) A Bird image and its saliency map (marked by grayscale), taken from [22]; (b) a signature image and its saliency map in our experiment (marked by jet-colormap).

Experimental Design

Since this paper is to extend our previous work, we summarize the experimental design and network training results briefly [9]. The proposed network system is designed to be used under the condition of only single genuine reference sample. Before that, the system is trained by sufficient local features from additional forgeries. In order to check whether our system can learn some useful and ubiquitous forgery features (existing in signatures of different forged signers) that can be used to detect forgeries. We design six experiments by using two sub-datasets with different genuine signers, and then arrange forgery samples of different sizes for controlled experiments, as shown in Table 1.

Table 1 The summary of the experiment with different sample sizes [9].

Sub dataset	Group	Exp No.	Exp. Stage	Genuine Signatures	Forged signatures from			
					signer 1	signer 2	signer 3	signer 4
ID 14 (Exp.1-3) / ID 16 (Exp.4-6)	Control Group	Exp.1,4	Training	No.1	No.1,2	No.1,2	No.1,2	
			Validation	No.2−4	No.3,4	No.3,4	No.3,4	
			Testing	No.5−8				No.1−4
	Experiment Group	Exp.2,5	Training	No.1	No.1,2	No.1,2		
			Validation	No.2−4	No.3,4	No.3,4		
			Testing	No.5−8				No.1−4
		Exp.3,6	Training	No.1	No.1,2			
			Validation	No.2−4	No.3,4			
			Testing	No.5−8				No.1−4

In Exp.1 and Exp.4, we use the complete dataset (with three forged signers and six forgeries) as our control group. While the remaining experiments with different forgery sample sizes are served as the experiment group to confirm our assumption.

The "Development environment" and "Data Preprocessing" can be found in [8].

Experimental Results

Networks Performance

In the training process, we use a popular optimization technique, called the stochastic gradient descent (SGD) algorithm [23], to optimize our networks. After network training and validation, theatrically we can get a well-trained classification model. To further verify whether our networks can be used by the completely unknown signer. We test the networks with new signers' signatures that are not present before (not available in the training and verification stages). Then we use the following equations [24] to evaluate the network performance during the training, verification, and testing stages.

$$\text{ACC} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \times 100\%$$

$$\text{FRR} = \frac{\text{FN}}{\text{TP+FN}} \times 100\%$$

$$\text{FAR} = \frac{\text{FP}}{\text{FP+TN}} \times 100\%$$

where ACC = Accuracy, TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative, FRR = false rejection rate (or type I error), and FAR = false acceptance rate (or type II error). Thus, higher accuracy and lower error rate (FRR/FAR) can be considered as better performance, as summarized in Table 2.

Table 2 The summary of training results [9].

Sub dataset ID	14			16		
Exp. ID	**Exp.1**	Exp.2	Exp.3	**Exp.4**	Exp.5	Exp.6
Forged author Num.	**3**	2	1	**3**	2	1
Forged signature Num.	**6**	4	2	**6**	4	2
Train accuracy (%)	**99.93**	100	99.67	**100**	99.97	100
Validation accuracy (%)	**100**	100	95.66	**98.96**	97.56	99.98
Test accuracy (%)	**99.96**	99.98	76.93	**94.37**	90.23	90.85
Test FAR (%)	**0**	0	27.81	**5.88**	15.16	2.83
Test FRR (%)	**0.22**	0.07	1.47	**5.34**	3.66	16.31

Visualization

Most deep learning methods are generally opaque and lack explainability. We try to understand the process by visualization. Fortunately, since our samples are images, it is easier for us to visualize the processed results of convolution filters and saliency maps.

Visualization (Part-1): The Processed Results of Convolution Filters

The visualization figures of the convolved results can help us to understand the CNN feature extraction process. The Inception V3 network has 6848 filters totally [25]. There are several filters in each convolution layer. In order to visualize the convolved results of the filters, we input some grayscale random noise images to the network, as shown in Fig. 6. Then we let the filters optimize (convolve) these random images, and get the convolved results. Fig. 7(a)-(f) show the convolved results of the first nine filters of some layers. We can see some “spots” and “line textures” in Fig. 7(a) and (b), respectively. As the depth of the network is increased, these filters create increasingly complex patterns. From layer to layer, the convolution filters can gradually capture the higher-level of abstraction features.

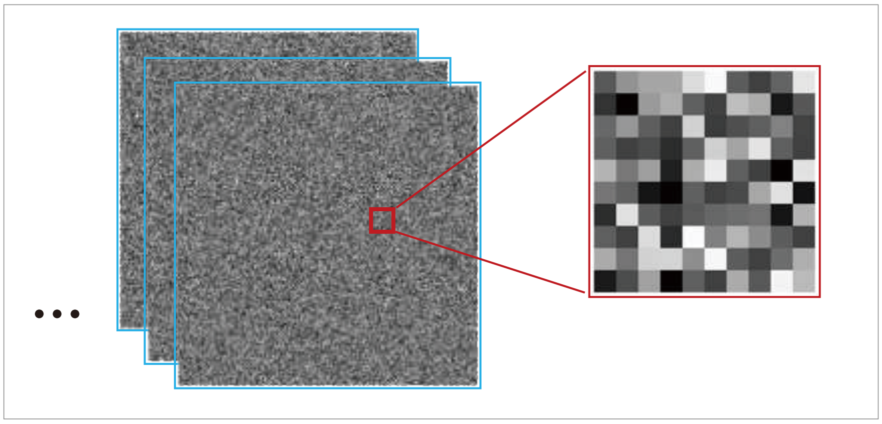
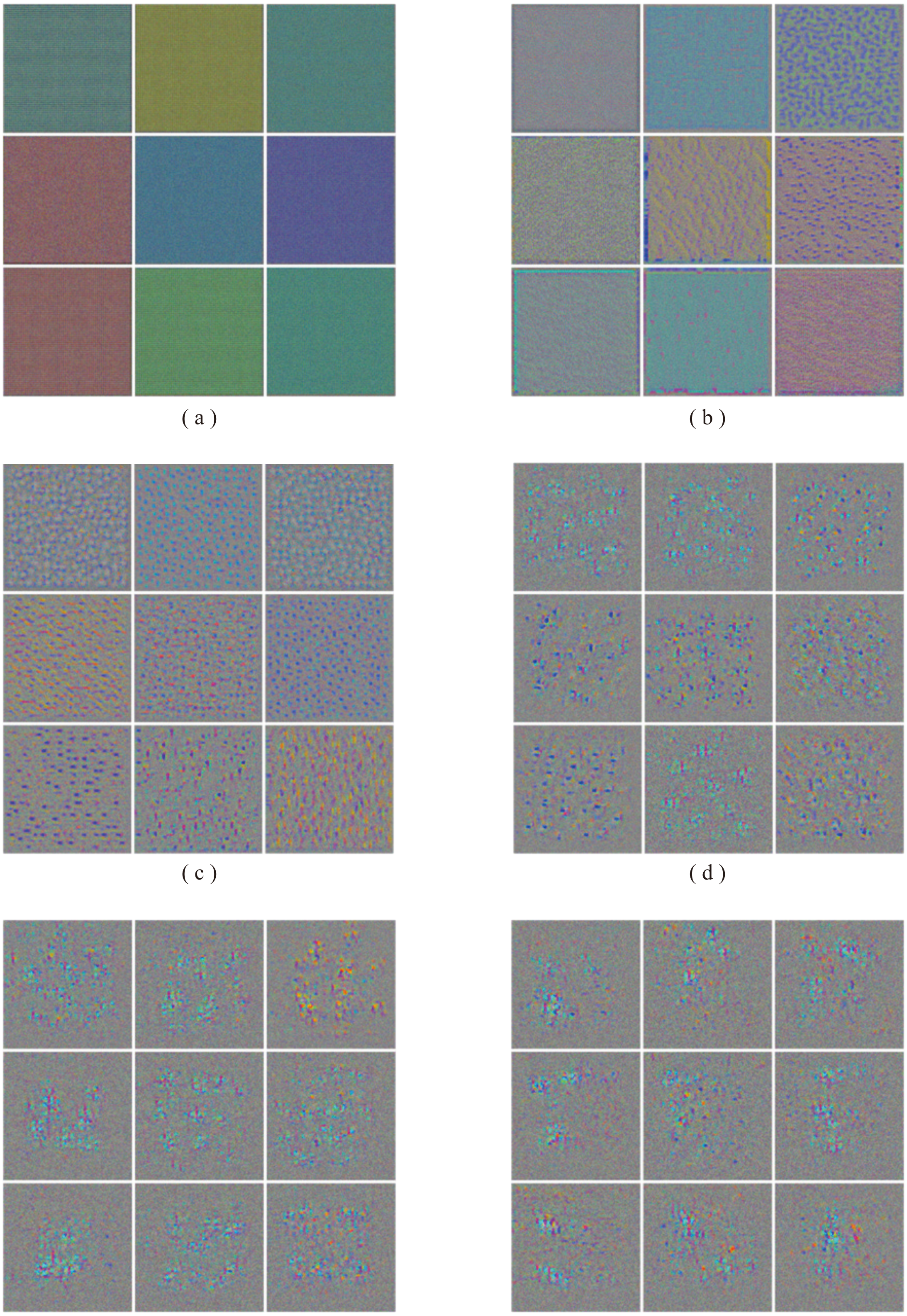


Fig. 6 The input grayscale random noise images for the visualization of the convolved results of the filters.



(a)

(b)

(c)

(d)

(e)

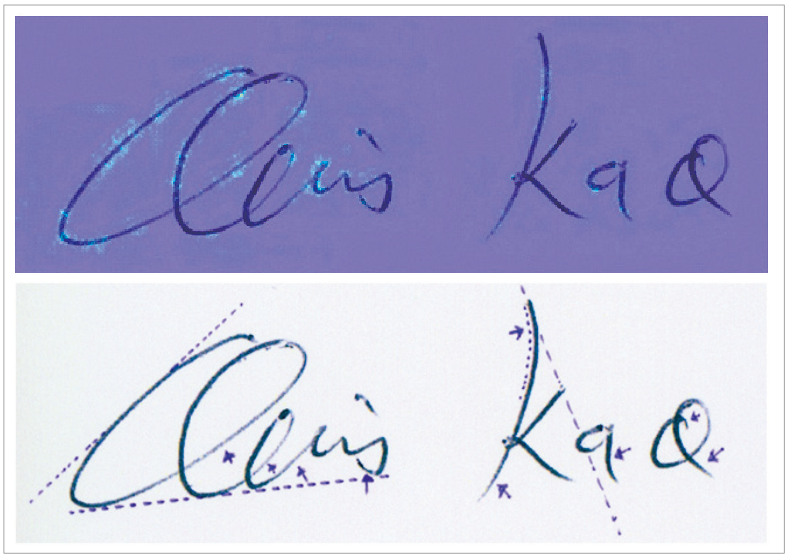
(f)

Fig. 7 The convolved results of the first nine filters of some layers: (a) layer 1; (b) layer 3; (c) layer 4; (d) layer 10; (e) layer 40; (f) layer 94.

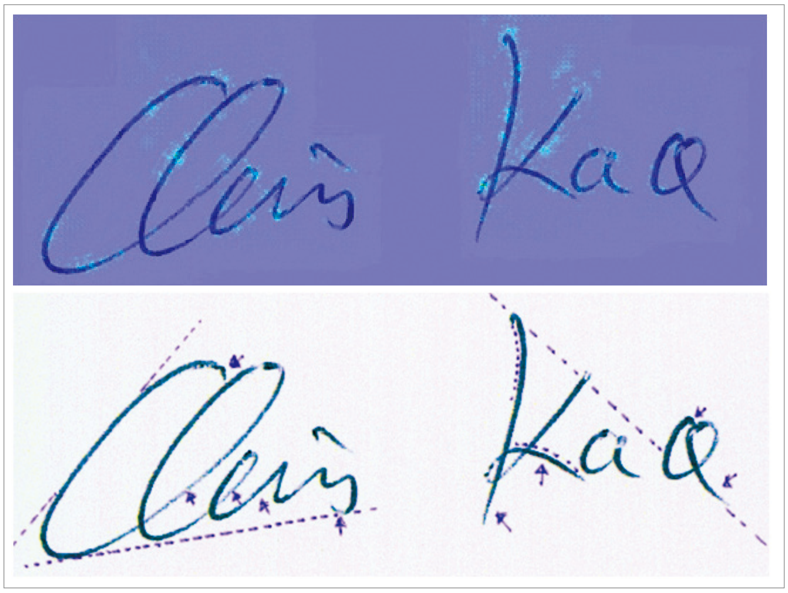
Visualization (Part-2): Saliency Maps

We generate the saliency maps of questioned signatures and use them to make our method more explainable. These visualized results are based on the last fully-connected layer of neural networks, and can be used to check whether the network's decisions are reasonable. In this part of experiment, we arranged three genuine signatures and three forgeries for verification (based on only single known sample). We also compare the results with human recognition by a forensic document examiner of Taiwan's Criminal Investigation Bureau (CIB). The comparison of CNN architecture (Inception V3) visualization outputs and expert's examination results are presented in Fig. 8(a)-(f). In the CIB forensic document examiner's result (the bottom figures of Fig. 8(a)-(f)), the blue arrows and dotted lines indicate the "similarities" in certain features, such as the stroke fluency, spacing, slant, relative positioning and formation of the stokes between the questioned signatures with the genuine signature. While the red ones indicate the "discrepancies" between them.

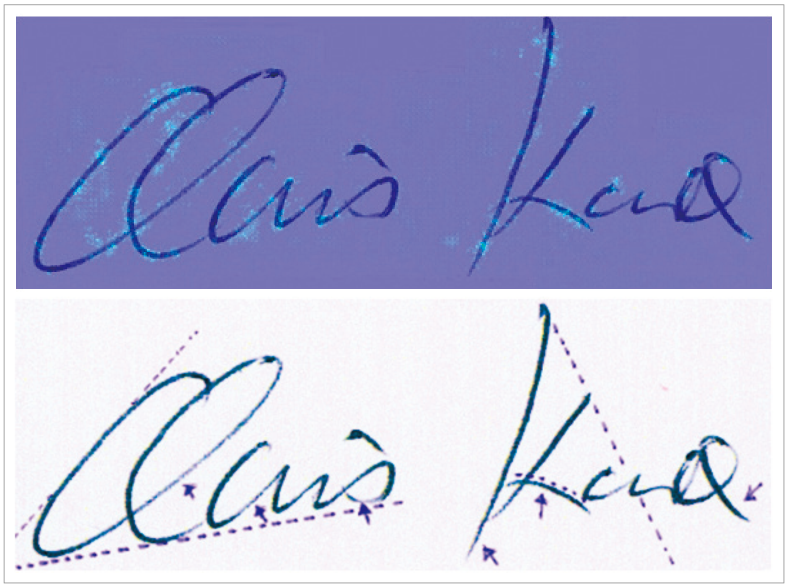
For clearer visualization, we utilize the jet-colormap to represent intensity, then impose the saliency map on the original signature image (the top figures of Fig. 8(a)-(f)). The jet-colormap returns color temperature from blue, green, yellow to red. It represents the intensity values between 0 to 1, the color scheme depicted in Fig. 8(g).



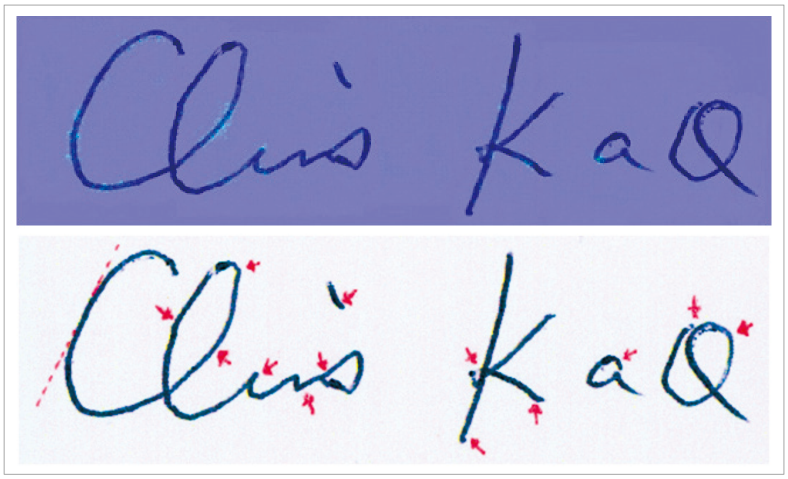
(a) genuine signature



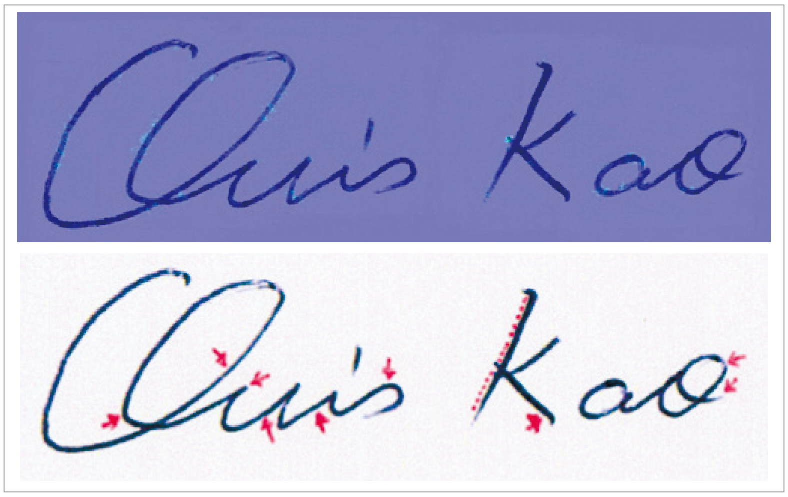
(b) genuine signature



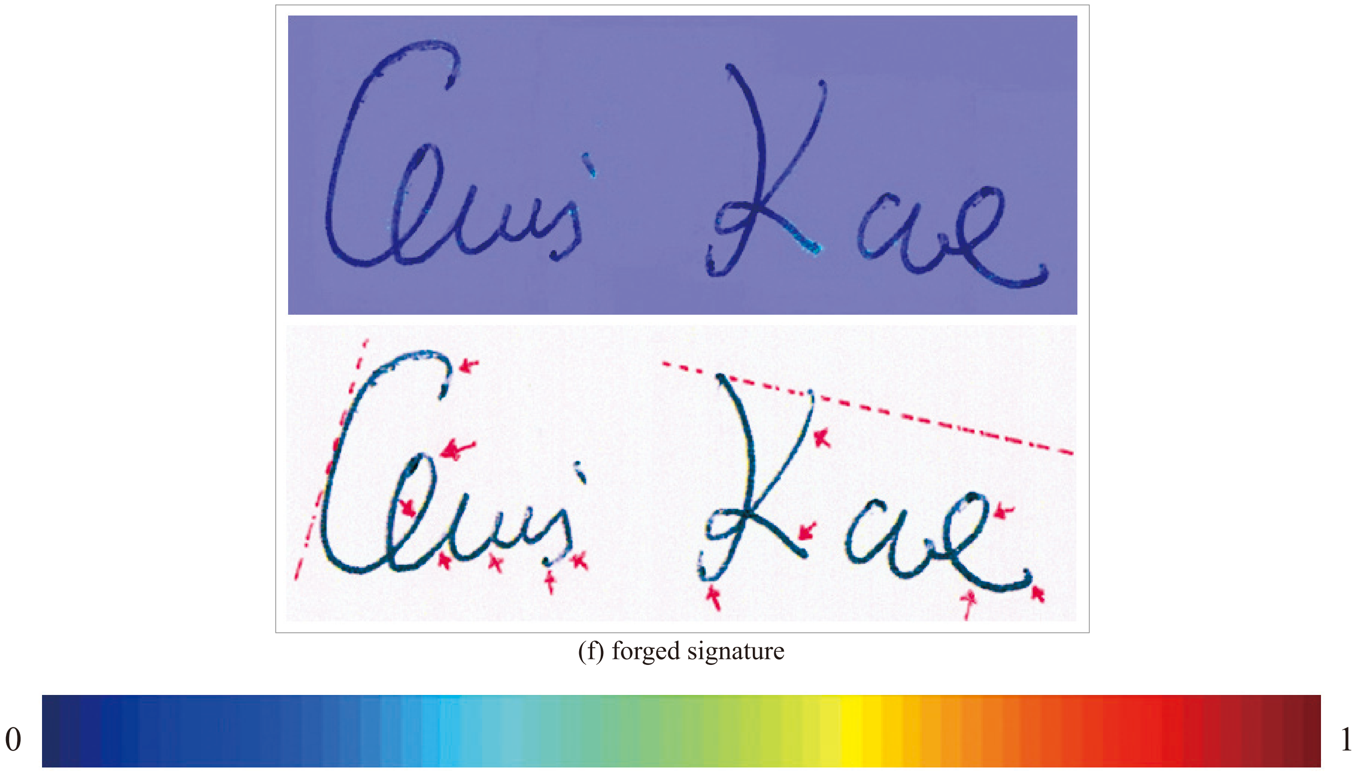
(c) genuine signature



(d) forged signature



(e) forged signature



(f) forged signature

(g) jet-colormap scheme

Fig. 8 The saliency maps and manual handwriting examination results of questioned signatures: (a-c) the genuine signatures, (d-f) the skilled forgeries form three different signers, (g) jet-colormap scheme.

Discussion

After we visualize the CNN networks' operation in Figs. 7 and 8, the convolved results of filtering and saliency maps can provide some insights into our signature examination work:

1. Overall, we can see that the verification processes and results of our network are reasonable. The visualization results mainly come from the signature itself rather than other unrelated noise (e.g. paper background or optical scanner device).
2. The saliency maps can show which strokes are important to the network's decision (especially in those hotspot areas). For example, we find that the turning point and the intersection of the strokes are often used as important features for CNN signature verification.
3. CNN is particularly good at scrutinizing the stroke quality. Since most forged signatures show signs of hesitation, tremor, re-touching, pen-stops in the position where they are not expected, we can find awkwardly formed stokes in the saliency maps.
4. After comparing the saliency maps and expert's examination results, we can find the human expert is better at examining the overall formation of the signatures. Considering that our CNN (Inception V3) system performs verification based on local feature blocks, such comparison results make sense.
5. Due to the limited capacity of human attention, the forensic document examiner only marks certain features on both genuine and forged signatures which are sufficient for him to reach a conclusion. However, the computer can tirelessly

mark all detected features without missing, and make quantitative measurements possible.

Conclusions

In this paper, we extend our previous work of using the deep learning based assistance method for signature examination. We use the visualization figures of the convolved results to help users to understand the CNN feature extraction process. We also generate the saliency maps of questioned signatures and use them to make our method more explainable. The proposed method can be used for preliminary automatic signature verification. Besides, it can easily work with human experts to obtain direct and strong effectiveness. In the future work, we expect that follow-up research can assist human experts to strengthen their cognitive abilities in the decision-making process.

References

1. Khalajzadeh H, Mansouri M, Teshnehlab M. Persian signature verification using convolutional neural networks. International Journal of Engineering Research and Technology 2012; 1(2):7-12.
2. Hafemann LG, Sabourin R, Oliveira LS. Offline handwritten signature verification—literature review. 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 1-8, Montreal, QC, Canada, 28 November-1 December 2017.
3. Alvarez G, Sheffer B, Bryant M. *Offline signature verification with convolutional neural networks*; Technical report; Stanford University: Stanford, CA, USA, 2016.
4. Zhang X-Y, Bengio Y, Liu C-L. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. Pattern Recognition 2017; 61:348-60.
5. Diaz M, Ferrer M-A, Impedovo D, Malik M-I, Pirlo G, Plamondon R. A perspective analysis of handwritten signature technology. ACM Computing Surveys (CSUR) 2019; 51(6):1-39.
6. Hafemann L-G, Oliveira L-S, Sabourin R. Fixed-sized representation learning from offline handwritten signatures of different sizes. International Journal on Document Analysis and Recognition (IJDAR) 2018; 21(3):219-32.
7. Adamski M, Saeed K. Signature verification by only single genuine sample in offline and online systems. AIP Conference Proceedings, 180011-180015, Rhodes, Greece, 22–28 September 2015.
8. Liwicki M, Malik MI, Van Den Heuvel CE, Chen X, Berger C, Stoel R, Blumenstein M, Found B. Signature verification competition for online and offline skilled forgeries (sigcomp2011). 2011 International Conference on Document Analysis and Recognition, 1480-1484, Beijing, China, 18-21 September 2011.
9. Kao H-H, Wen C-Y. An Offline Signature Verification and Forgery Detection Method Based on a Single Known Sample and an Explainable Deep Learning Approach. Applied Sciences 2020; 10(11):3716-31.
10. Kao H-H, Wen C-Y. A Deep Learning-Based Offline Signature Verification Method by Single Known Sample. Journal of Chung Cheng Institute of Technology 2020; 49(1):23-34.
11. LeCun Y, Boser B, Denker J-S, Henderson D, Howard R-E, Hubbard W, Jackel L-D. Backpropagation applied to handwritten zip code recognition. Neural computation 1989; 1(4):541-51.
12. Taigman Y, Yang M, Ranzato MA, Wolf L. Deepface: Closing the gap to human-level performance in face verification. Proceedings of the IEEE conference on computer vision and pattern recognition, 1701-08.
13. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision, 1026-34.
14. Zhong Z, Jin L, Xie Z. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 846-50.
15. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M. Imagenet large scale visual recognition challenge. International journal of computer vision 2015; 115(3):211-52.
16. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer

vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2818-2826, Las Vegas, NV, USA, 27-30 June 2016.

17. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv 2014:arXiv:1409.1556.
18. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 580-587, Columbus, OH, USA, 23-28 June 2014.
19. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA, USA: MIT press; 2016.
20. Scherer D, Müller A, Behnke S. Evaluation of pooling operations in convolutional architectures for object recognition. International conference on artificial neural networks, 92-101, Thessaloniki, Greece, 15-18 September 2010.
21. Jarrett K, Kavukcuoglu K, Ranzato MA, LeCun Y. What is the best multi-stage architecture for object recognition? 2009 IEEE 12th international conference on computer vision, 2146-2153.
22. Simonyan K, Vedaldi A, Zisserman A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 2013.
23. Bottou L. Large-scale machine learning with stochastic gradient descent. Proceedings of COMPSTAT’2010, 177-186, Paris, France, 22–27 August 2010.
24. Scharcanski J, Proença H, Du E. Signal and image processing for biometrics. Indianapolis, IN, USA: Springer-Verlag; 2014.
25. Bertin P, Hashir M, Weiss M, Frappier V, Perkins TJ, Boucher G, Cohen JP. Analysis of Gene Interaction Graphs as Prior Knowledge for Machine Learning Models. arXiv preprint arXiv:1905.02295 2019.