Signature Detection, Restoration, and Verification: A Novel Chinese Document Signature Forgery Detection Benchmark

Kaihong Yan, Ying Zhang, Haoran Tang, Chengkai Ren, Jian Zhang, Gaoang Wang*, Hongwei Wang*
Zhejiang University-University of Illinois at Urbana-Champaign Institute, Zhejiang University,
Haining, Zhejiang 314400, China
{kaihong.20,ying.21,haoran.21,chengkai.21,jianzhang,gaoangwang,hongweiwang}@intl.zju.edu.cn

Abstract

Offline signature forgery detection has attracted many researchers in recent years. In real situations, signatures should be detected from the signed documents and verified by the forgery detection system. There are many challenges in the pipeline. First, some signatures have low resolutions and are difficult to be detected. Second, the cropped signatures may contain irrelevant background context of the document, making the signature hard to be verified. Third, some forgery signatures are very similar to genuine ones, increasing the challenge of verification. In addition, most existing datasets do not cover all the pipeline tasks. Moreover, publicly available Chinese-based signature datasets are rare for research purposes. In this paper, we construct a novel Chinese document offline signature forgery detection benchmark, namely ChiSig, which includes all pipeline tasks, i.e.*, signature detection, restoration, and verification. Besides, we extensively compare different deep learning-based approaches in these three tasks. The results show that our proposed dataset can effectively provide solutions for constructing pipeline systems for Chinese document signature forgery detection.*

1. Introduction

The handwritten signature is a common way of authentication. It is widely used in the legitimacy verification of documents such as contracts, forms, and bills. However, the widespread use of signatures has a risk of signature forgery, and more and more scenarios require the identification and authentication of handwritten signatures in scanned documents [12]. As a result, the development of a pipeline system for forgery detection is necessary.

Depending on the signature acquisition, the signature forgery detection systems can be divided into two categories [11], offline and online. The signature in the offline system is a two-dimensional static digital image mostly captured by document scanning. On the contrary, in the online system, the sample signature is described by position, velocity, pen direction, pressure sequence *etc*., which is obtained through a special acquisition device. Online signatures have more distinct features than offline signatures and can be modeled in temporal space and achieve better results [20]. However, the data for online signatures are expensive to capture and do not cover all signature verification scenarios. Contrarily, the data of offline signature is easy to obtain but difficult to verify due to a lack of temporal information [20], a limited amount of features [11], and the noisy background [6].

Most studies [5, 11, 36] for offline signature verification focus on verifying existing clean, noise-free signature images on a public dataset, which do not fit the actual application scenario. It is not easy to get a clean signature in the real document image. The location of the signature is not fixed, and there is an interference with the signature from free objects like logos, stamps, and text in the document. The actual signature obtained will have noise and background texture, leading to a difficult verification task. Therefore, developing a practical signature forgery detection pipeline system should deal with the challenges of signature detection, signature restoration, and signature verification.

In this paper, we construct a novel Chinese document signature forgery detection benchmark for offline signature verification, namely *ChiSig*, which includes all signature forgery detection pipeline tasks, *i.e*., signature detection, restoration and verification. This rich dataset consists of 102 classes of signatures and has been carefully collected and organized to consider the influencing factors present in the signature process. Different baseline methods are also evaluated and compared, thus helping researchers to propose or improve algorithms. The dataset can be accessed at https://github.com/dskezju/Chisig.

The main contributions are summarized as follows. 1) We introduce a novel public available benchmark dataset

*Corresponding authors.

for Chinese signature forgery detection. 2) The introduced dataset includes all tasks for the full pipeline of the signature forgery detection system, which is more realistic than most existing datasets. 3) Several baseline methods are evaluated on the dataset, facilitating researchers for future works.

2. Related Work

2.1. Signature Detection

Detecting the location of signatures on complex scanned documents and cropping the region of interest (ROI) is the primary goal of signature detection. Existing methods for signature detection can be divided into two categories. One is to propose a specialized system to extract features to detect signatures. The other is to model the signature detection task as a standard object detection task and use deep learning-based methods for the task. Specifically, Sharma [28] uses YOLOv2 [24] and Faster R-CNN [8] for signature detection. Hauri [12] studies different object detection methods on signature detection, including YOLOv5 [9], Faster R-CNN [8], and RetinaNet [18], where YOLOv5 outperforms the other models. On account of that, we evaluate the performance of different deep learning-based object detection methods on our proposed dataset.

2.2. Signature Restoration

Signatures on real-world documents often interfere with background contents, such as seals, stamps, handwritten text, printed lines, and printed text, which increase the difficulty of signature verification and influence the signature verification effect. Removing noisy background can be seen as reconstructing an image from a noisy image to a clean one. Therefore, image-to-image translation methods can be well utilized in the task. Some methods assume paired data, *i.e*., the noisy and clean versions of the same signature image, are available in training. For example, DE-GAN [30] utilizes the log loss between paired enhanced image and ground-truth image to optimize discriminator and generator. Some methods can be trained on unpaired data, which is more realistic. For instance, DualGAN [44] builds the loss through two generators to solve the problem of unpaired data. This paper evaluates several deep learning-based methods for signature restoration on our proposed dataset.

2.3. Signature Verification

In biometrics and document forensics, offline signature verification is challenging to determine whether a given pair of signatures are genuine or forged. There are two different types of methods for signature verification [11]: writer-dependent and writer-independent. Writer-dependent method lacks the flexibility to verify the new user, while the writer-independent method is more robust and naturally becomes the research focus. Nowadays, using deep learning methods to tackle those signature verification problems has become more popular. SigNet [5] uses Siamese convolutional networks as a feature extractor to learn signature embeddings. Inverse Discriminative Network [36] introduces inverse supervision and a multi-path attention mechanism to resolve the sparse information of signatures. However, due to the lack of public datasets in Chinese, most signature verification systems mainly focus on English, which hinders research on Chinese signature verification. Thus, based on our public dataset, we validate performance for different models and provide benchmark results.

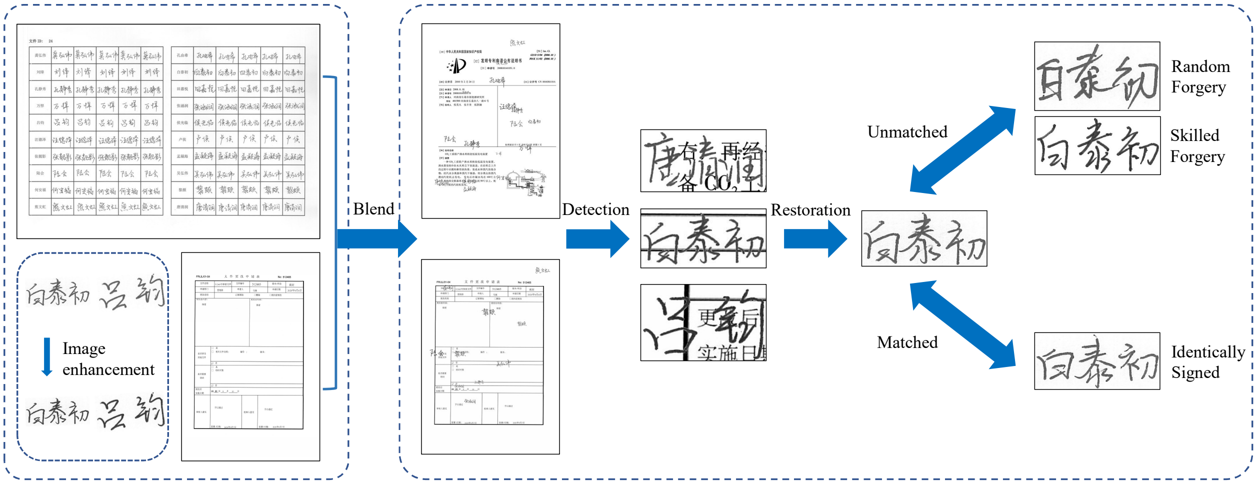
2.4. Offline Signature Datasets

The most commonly used dataset for signature detection is Tobacco800. However, this dataset is in English and is only composed of 1290 document images. To the best of our knowledge, there are no public datasets on signature restoration. Most of the related studies [6] use the method of data synthesis to build datasets.

Extensive datasets exist for signature verification, such as GPDS, CEDAR and BHSig260. Tab. 1 gives more information about the offline signature datasets. However, most of the mentioned datasets are in English characters, and there are not many public Chinese datasets. The lack of public Chinese datasets hinders research on Chinese signature verification. In addition, Chinese characters are more diverse than English, and the writing forms are diverse, which is more difficult for Chinese offline signature verification. Some researchers [14, 36] have studied the verification of Chinese signatures, but the datasets they used are not publicly available. Currently, a Chinese signature dataset that can be obtained is ICDAR 2011 (SigComp2011) [20]. However, the dataset has limited data, with only 1177 signature images and 20 different names for offline signature verification. Unlike existing datasets, ours has 10,242 signature images with 500 different signed names.

2.5. Sketch-oriented deep learning

From another perspective, hand-written characters share similarities with the free-hand sketch, which has been successfully modeled using RNNs [10, 39]. Free-hand sketch has a time-sequence nature [38]. It is a dynamic and temporally extended process. In the sketch classification and modeling task, SketchMate [39], and Multi-graph transformer [41] are designed as a two-branch CNN-RNN network and a transformer to capture both geometric and temporal information. A deep embedding model [40] encodes the static and temporal pattern of sketch stroke to extract semantic vectors as semantic knowledge for sketch retrieval and zero-shot learning. In the multi-modal application, the sketch



(a) (b)

Figure 1. (a) Preparation of our data set. (b) The pipeline system of signature verification.

Table 1. Comparison of offline signature datasets.

Dataset	Access	Script	IDs	Samples	Verification	Restoration	Detection
Tobacco800 [17]	Public	Western	-	1,290	-	-	✓
CEDAR [31]	Public	Western	55	2,624	✓	-	-
GPDS [7]	Public	Western	4,000	216,000	✓	-	-
BHSig260 [23]	Public	Bengali, Hindi	260	14,040	✓	-	-
SigComp2011 [20]	Public	Chinese	10	1177	✓	-	-
CSD [14]	Exclusive	Chinese	300	5,400	✓	-	-
Dataset in [36]	Exclusive	Chinese	749	29,000	✓	-	-
Ours	Public	Chinese	102	10,242	✓	✓	✓

is demonstrated on video retrieval with zero-shot learning [42].

The stroke information for handwriting recognition shares similarities with a free-hand sketch. Multivariate time series classification is applied in online handwriting recognition. Joint online handwriting classification [22] employs the cross-entropy loss in combination with distance and similarity losses to learning handwriting style and stroke order information. Sketch2Vec [1] proposes a cross-modal translation between image and vector space using encoder-decoder architectures, which provides a powerful representation for sketch and handwriting data.

3. The Dataset

We introduce the signature forgery detection dataset for three tasks, including detection, restoration and verification. The dataset consists of clean handwritten signatures, synthesized noisy handwritten signatures, and synthesized documents with handwritten signatures. More details are demonstrated in the following subsections and Fig. 1.

Table 2. Statistic of signature detection task, including background images (BG images), signatures (Sigs), documents (Docs), signatures per document (Sigs / Doc).

Split	BG Images	Sigs	Docs	Sigs / Doc
Train set	1,412	6,593	8,472	14
Test set	354	1,648	2,124	15

Table 3. Statistic of signature restoration task.

Split	Synthetic signature image
Train set	8,000
Test set	2,000

Our dataset is more valuable and useful. First, our dataset is in Chinese and has a large amount of data. Furthermore, our dataset provides a different setting that addresses different stages of developing practical signature verification pipeline systems, including signature detection,

Table 4. Statistic of signature verification task. The testing set includes forged (f) and genuine (g) samples.

Split	Number
Train set	1,480
Test set	8,120 (f) + 5,760 (g)

restoration, and verification. We also provide benchmark experiments for the constructed dataset. Tab. 2, Tab. 3 and Tab. 4 provide the descriptions of our dataset in different tasks.

3.1. Data Acquisition

Signature acquisition. First, we use name generation software to randomly generate and obtain 500 Chinese names. To have a diverse distribution of names, we take into account the gender of the name, the distribution of the number of words and the probability of occurrence of the last name. Then, to collect the volunteers' signatures, we create a form with 20 rows and 6 columns, with the first column filling the generated printed names to guide the volunteers, as shown on the top left of Fig. 1. Each form has a unique file id and includes 100 cells that can be signed. We have prepared a total of 83 forms. The names appear randomly in all forms, and each name appears 4 times in all prepared forms. Each form is distinct from the others, *i.e.*, each name is signed by 4 different volunteers, which increases the diversity of obtained signatures. After that, we create another 20 forms for skilled forgery. Suppose selected forms with file id equal to id_1, and we construct skilled forged signatures by having volunteers forge the signatures on these 20 forms and set the file id of these forged forms as $id_1 + 100$. A total of 103 forms with a sample of 10,300 signatures are collected. After excluding invalid signatures, we get 10,242 signatures. Finally, the signatures in the cells are cropped into individual signature images and form a clean signature dataset.

The obtained signature image is named in the following format **name-id-number**.jpg, where "name" represents the name signed by volunteer, "id" represents the file id, and "number" represents the number of signatures. For example, if the name 'A' is signed by four people, and then each person will sign five times to get a total of 20 samples. If we consider the name 'A' signed by each person as a class, We obtained at least four different classes with five samples each class. If a signature image is named with a file id greater than 100, *e.g.* $id_1 > 100$, it means it is a skilled forged one of the signature with file id $id_1 - 100$.

We have collected two different forms of signatures through the above work: random forgery and skilled forgery. These two categories are shown in Fig. 1 and defined as follows.

- **Random Forgery**. A pair of signatures, with the same name with different ids, *e.g.*, name-id_1-# and name-id_2-# (as shown in Fig. 1).
- **Skilled Forgery**. A pair of signatures, with the same name. And the difference between ids is 100, *e.g.*, name-id_1-# and name-$(id_1 + 100)$-# (as shown in Fig. 1).

We split obtained signatures into training and test sets to be used later in the signature verification benchmark. The train set and the test set are divided in different ways and we ensure that the volunteers in the training and test sets are disjointed, *i.e.*, the volunteers who signed the test set do not appear in the training set. For the training set, one name is signed by different volunteers. We consider the name signed by each person as a class. Each class has five samples. Recall that every signature had been named in the format of **name-id-number**.jpg. Thus, the class label is defined by the **name-id** of each signature image. For the test set, each sample consists of a pair of signatures with the same name and a class label that shows whether they are forged. We select signatures with 29 different file ids to make a test set so that random forgeries and skilled forgeries are included in the set. We generate signature pairs for the test set by combining all signatures and assign class labels. The rest of the signatures are used to construct the train set. The task of signature verification is to train the feature extractor on the training set and evaluate the model performance on the test set for given pairs of signatures.

Background document acquisition. We collect various document background images to blend the signature into the document background. We obtain background document images from the XFUND dataset [43]. The XFUND dataset is a multilingual form understanding dataset containing 199 scanned pages, which can increase the diversity of our dataset. We do not consider using those background images with resolutions lower than 300dpi to improve the robustness of the model and adapt to more complex document environments. We use the images from the XFUND dataset as a part of the background images. In addition, the availability of scanned documents, such as public patents and Chinese national industry standards, also provides another important source of background document images. Finally, we collect 1,766 background images to synthesize the signature dataset with the noise and signature detection dataset.

3.2. Data Synthesis

The signed documents are synthesized by blending the collected background documents and clean signatures. The synthesized data can be used for signature detection and restoration tasks. First, the signature is cut out from the collected form. Then, image enhancement technology is used for the cropped signature image. After that, Otsu's method, a binarization method, is applied to the signature image to obtain the signature mask. Finally, some signatures are placed completely randomly in the background to simulate real signature scenarios and improve the robustness of the model.

The location and number of signature placement and background repetitions are considered, as shown in Fig. 1. We randomly select 10 to 20 signatures and place them in random locations of the background images. Each background image is used six times to increase the dataset size. Experiments show that entirely random placement of signatures in the background image achieves better results than interfering with the placement position. We believe that this is beneficial for the model robustness of the signature detection model.

Several image enhancements are adopted before combining signatures with background images to improve the authenticity of synthetic data. We scale signatures to random sizes within specific ranges and rotate signatures to mimic the uncertainty of signature size and angle. The signature size is set to $3\% - 15\%$ of the background image size, and the signature is rotated left and right within $12°$. The randomness of data augmentation operations such as random scaling and random rotation obeys normal distribution. Moreover, we also perform color and brightness adjustments on the signature images and darken the signature color. After adding the signatures with the backgrounds, the synthetic document images with multiple signatures will be used for signature detection. We crop the signatures that do not overlap with other signatures from the document images, as shown in Fig. 1. Signatures without noise added are not considered for signature restoration. We construct pairs of signature images with and without the background contents for the signature restoration benchmark.

4. Benchmark

The dataset includes three benchmarks, detection, restoration, and verification. The selected evaluation methods are all strong baselines and representative methods in the related fields. In addition, the stability and robustness of the selected methods have been verified. Details for each task are provided in the following subsections.

4.1. Detection

To illustrate the effect of our constructed dataset on the signature detection task, we provide benchmark experiments for signature detection. We follow the MS COCO [19] dataset format for evaluation. We build the benchmark of our dataset using the Open MMLab detection toolbox [3], a unifying framework for object detection.

Evaluation metrics. We use the same evaluation metrics as MS COCO Dataset, including average precision (AP) and average recall (AR). We report AP with the IoU from 0.5 to 0.95. The APs at $IoU = 0.50$ and $IoU = 0.75$ are denoted as AP_{50} and AP_{75}, respectively. Considering the object size, the APs for median objects ($32^2 < area < 96^2$) and large objects ($area > 96^2$) are denoted as $\mathrm{AP}^{\mathrm{median}}$ and $\mathrm{AP}^{\mathrm{large}}$, respectively. Accordingly, we report AR with the IoU from 0.5 to 0.95 and denote AR for median and large objects as $\mathrm{AR}^{\mathrm{median}}$ and $\mathrm{AR}^{\mathrm{large}}$.

Evaluated methods. Three representative methods of signature detection are used for evaluation, including Faster R-CNN [26], YOLOv3 [25] and DETR [2], they represent the two-stage detector, one-stage detector, transformer-based detector, respectively. Specifically, Faster R-CNN introduces Region Proposal Network (RPN) to generate region proposals and uses Fast R-CNN [8] framework for detection. Faster R-CNN merges RPN and Fast R-CNN by sharing their convolutional features to generate region proposals. YOLOv3 updates to YOLOv2 [24] by using feature pyramid networks for detecting objects across scales. YOLOv3 also introduces Darknet-53 for feature extraction. DETR treats object detection as a direct set prediction problem, which can be trained end-to-end with a set loss function that performs bipartite matching between predicted and ground-truth objects. The architecture of DETR includes a CNN backbone to extract features and a transformer [33] architecture to make the final prediction.

Benchmark results. The detection results are reported in Tab. 5. And two sample results for signature detection using DETR are shown in Fig. 2 and Fig. 3. Comparing these three models, Faster R-CNN outperforms other models because two-stage detectors usually outperform one-stage detectors as two-stages allow the model to learn flexible region proposals. As for DETR, transformer-based models usually require more data to achieve better results, and our dataset is limited considering the complexity of DETR. Despite different approaches, all these three models can achieve a precision of 0.990 when we set IoU=0.50, and the average recalls are all in a narrow range of 0.814 to 0.857, suggesting that signature detection can be successfully done with the state-of-the-art object detection methods.

4.2. Restoration

Evaluation metrics. We use four metrics, including Peak signal-to-noise ratio (PSNR), Structural Similarity Index (SSIM), Fréchet Inception Distance (FID), and Learned Perceptual Image Patch Similarity (LPIPS) for signature restoration evaluation.

PSNR provides an evaluation at the pixel level and is based on the error between corresponding pixels. PSNR measures the similarity between the original image and the processed image. The larger the PSNR value, the less dis-

Table 5. Results for signature detection.

Method	AP	AP_{50}	AP_{75}	AP^{median}	AP^{large}	AR	AR^{median}	AR^{large}
Faster R-CNN	**0.819**	**0.990**	**0.954**	**0.720**	**0.823**	**0.857**	**0.756**	**0.860**
YOLOv3	0.784	**0.990**	0.947	0.688	0.801	0.847	0.747	0.849
DETR	0.765	**0.990**	0.930	0.612	0.768	0.814	0.686	0.818

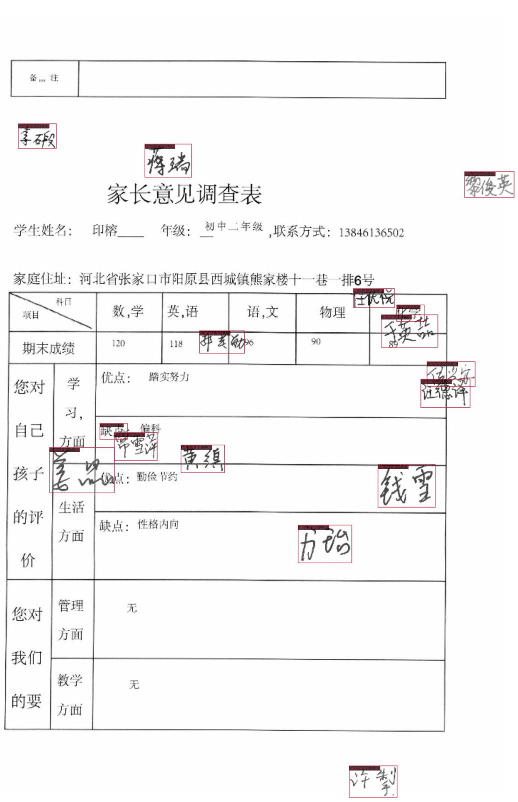


Figure 2. A tabular form sample for synthetic signature detection.

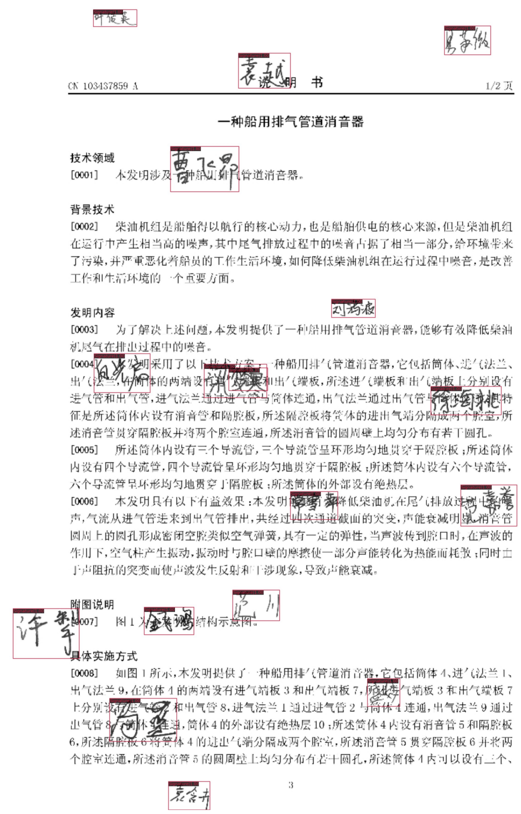


Figure 3. A technology patent sample for synthetic signature detection.

tortion and the higher the resemblance of the two images. Following is the definition of PSNR. We use MAX to denote the maximum image color value. If the size of the evaluation image is $M \times N$, PSNR would be computed as follow:

$$PSNR = 10 \log \left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}} \right), \tag{1}$$

where

$$\mathrm{MSE} = \frac{\sum_{\mathrm{x=1}}^{\mathrm{M}} \sum_{\mathrm{y=1}}^{\mathrm{N}} (\mathrm{I(x,y)} - \mathrm{G(x,y)})^2}{\mathrm{MN}}, \tag{2}$$

where I is the ground-truth image and G is the generated image.

SSIM [35] is used to measure the closeness of two images and estimate image quality degradation. To calculate SSIM, we need a reference image and an image processed by the corresponding image. SSIM uses three contrast modules to evaluate the images: brightness, contrast, and structure. Specifically, the mean is used to estimate brightness, the standard deviation as an estimate of contrast, and the covariance as a measure of structural similarity.

FID [13] is also a measure of similarity between two images. By loading the pre-trained Inception network and using the activation function output value of the last pooling layer as the feature vector, the distance between the ground-truth image and the generated image at the feature level can be calculated as FID. A lower FID means that the two distributions are close.

LPIPS [45] is used to estimate the difference between two images. LPIPS also requires a pre-trained network to extract features and a linear layer on top of the classification

Table 6. Results for signature restoration.

Method	PSNR	SSIM	FID	LPIPS
Autoencoder	22.70	0.9011	151.82	0.1018
Pix2pix	**34.47**	0.9765	5.68	0.0344
CycleGAN	31.95	**0.9820**	**5.25**	**0.0259**

network. The distance between features is calculated by learning the reverse mapping of the generated image to the ground-truth image. The lower the value of LPIPS, the more similar the two images are.

Evaluated methods. We evaluate the following representative signature restoration methods, including Denoising autoencoder [34], Pix2pix [16], and CycleGAN [46]. Specifically, the Denoising autoencoder introduces a straightforward training principle on top of the autoencoder, which is based on the goal of revoking a corruption process. Pix2pix takes conditional adversarial networks [21] as the approach for image-to-image translation problems, which can learn the mapping from input to output. CycleGAN can achieve style transfer from the source domain to the target domain without establishing a one-to-one mapping between training data, employing the cycle-consistency loss for regularization.

Benchmark results. The results of the enumerated methods mentioned above are given in Tab. 6. It could be observed that CycleGAN and Pix2pix get better performance than the Denoising autoencoder. The reason is that CycleGAN and Pix2pix utilize the mechanism of adversarial neural networks. Pix2pix achieves better performance in PSNR but worse in SSIM, FID and LPIPS, suggesting that Pix2pix's goal is to get closer to the target image, but it ignores human vision to a certain extent. Some sample results for signature restoration are shown in Fig. 4, which are generated to assess qualitative performance. We observe that the three models have obtained relatively good results. In particular, Pix2pix and CycleGAN have successfully removed the background noise artifact in the generated images.

4.3. Verification

In this section, we use four embedding methods for signature verification on our dataset. After getting the embeddings of two signatures, we compare two signatures by calculating the cosine similarity of two embeddings to estimate whether the signatures are signed by the same person.

Evaluation metrics. We use three metrics for evaluation, including the Accuracy (Acc), Equal Error Rate (EER), and True Acceptance Rate (TAR) when the False Acceptance Rate (FAR) is equal to $1e{-}3$. These metrics are defined as follows.

$$FAR = \frac{\text{Number of false accepted}}{\text{Number of forged}} \tag{3}$$

$$FPR = \frac{\text{Number of false rejected}}{\text{Number of genuine}} \tag{4}$$

$$TAR = 1 - FPR \tag{5}$$

Evaluated methods. We evaluate ResNet50-IR-SE [4], InceptionResNet [32], ResNeXt50 [37] and VGG16 [29] for embedding networks. Specifically, ResNet50-IR-SE is introduced in the Arcface [4], which is based on the original ResNet50 with improved residual block and squeeze and excitation block [15]. InceptionResNet [32] combines Inception module and residual blocks and it is used for face verification in [27]. ResNeXt50 is a modularized network architecture for image classification presented by Xie *et al.* [37]. It aggregates a set of transformations with the same topology, which improves performance by maintaining the same complexity as the original ResNet50. VGG16 is a widespread and classic network for extracting image features, consisting of 13 convolutional layers and 3 fully connected layers, showing the benefits of deep networks.

Training loss functions. We adopt two commonly used loss functions, *i.e.*, Softmax loss and ArcFace loss, for learning the embedding networks. Specifically, the Softmax loss is defined as follows,

$$L = -\frac{1}{N}\sum_{i}^{N} log\frac{e^{x_i}}{\sum_j e^{x_j}} \tag{6}$$

where x_i stands for the predicted logits with index i. ArcFace Loss is an additive angular margin loss, defined as follows,

$$L = -\frac{1}{N}\sum_{i}^{N} log\frac{e^{s(cos(m_1\theta_{y_i+m_2}-m_3))}}{e^{s(cos(m_1\theta_{y_i+m_2}-m_3))} + \sum_{j=1,j\neq y_i}^{n} e^{scos\theta_j}} \tag{7}$$

where m_1, m_2, m_3, s are hyper-parameters, and $cos\theta_i$ stands for the predicted logits with index i.

Benchmark results. In this experiment, we employ the test set in Section 3.1. We vary the threshold and record FAR, FRR and Accuracy every turn. The results are shown in Tab. 7. ResNet50-IR-SE recalibrates channel-wise feature responses by explicitly modeling interdependencies between channels and achieves the best TAR. Combining the Inception architecture with residual connections, InceptionResnet shows the best performance on EER and ACC. ResNeXt50 uses group convolution and performs better than VGG16. ResNet50-IR-SE outperforms other models on TAR When FAR is equal to $1e{-}3$, no matter that the

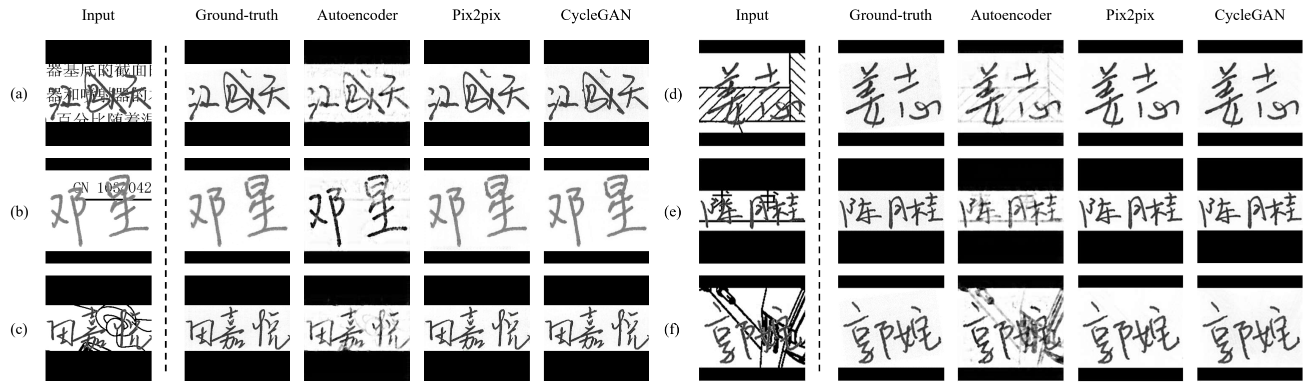


Figure 4. Samples of signature restoration.

Table 7. Results for signature verification

Networks	Loss	EER	TAR	ACC
ResNet50-IR-SE	Softmax	0.092	**0.304**	0.909
InceptionResnet	Softmax	**0.066**	0.281	**0.936**
ResNeXt50	Softmax	0.152	0.043	0.855
VGG16	Softmax	0.245	0.029	0.763
ResNet50-IR-SE	ArcFace	0.089	**0.328**	0.914
InceptionResnet	ArcFace	**0.071**	0.169	**0.931**
ResNeXt50	ArcFace	0.117	0.081	0.884
VGG16	ArcFace	0.275	0.019	0.732

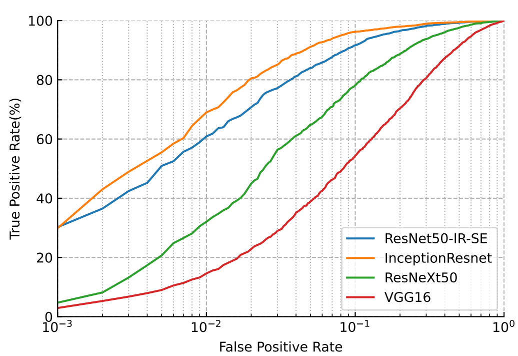


Figure 5. ROC curves of four methods with Softmax loss function.

Softmax loss function or ArcFace loss function is used. However, InceptionResnet has the lowest EER and accuracy with the Softmax loss function. Generally, ResNet50-IR-SE with ArcFace loss outperforms others among all five backbones, and VGG16 performs poorly on all settings. InceptionResnet is almost up to the performance of ResNet50-IR-SE. ResNeXt50 gains good performance on the test set.

From Figs. 5 and 6, we see clearly ResNet50-IR-SE as backbone outperforms other backbones with ArcFace loss function. However, with the Softmax loss function, the InceptionResnet backbone achieves better performance than ResNet50-IR-SE. When the figure's x-axis is scaled by log, we know all four backbone networks get low TAR when the FAR is low. Until the FAR reaches 1% the TAR remains at a low level. ResNet50-IR-SE, InceptionResnet and ResNeXt50's performance is optimized after using the ArcFace loss function, but VGG16 performs worse on ArcaFace loss.

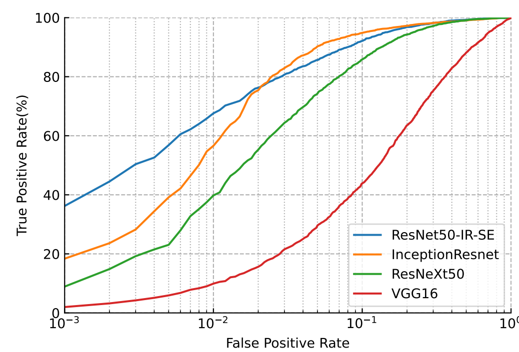


Figure 6. ROC curves of four methods with ArcFace loss function.

5. Discussion

In this paper, we present a novel Chinese document signature forgery detection dataset with three tasks, including signature detection, restoration and verification. We state the dataset construction and compare the proposed dataset with the existing signature verification datasets. We evaluate several strong baseline methods on the dataset for the three tasks, showing the capability of these representative methods. We hope the proposed new benchmark can facilitate future research on Chinese signature forgery detection.

References

[1] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Yongxin Yang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Vectorization and rasterization: Self-supervised learning for sketch and handwriting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5672–5681, 2021. 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 5

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 7

[5] Sounak Dey, Anjan Dutta, J Ignacio Toledo, Suman K Ghosh, Josep Lladós, and Umapada Pal. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017. 1, 2

[6] Deniz Engin, Alperen Kantarci, Seçil Arslan, and Hazim Kemel Ekenel. Offline signature verification on real-world documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 808–809, 2020. 1, 2

[7] Miguel A Ferrer, Moises Diaz-Cabrera, and Aythami Morales. Static signature synthesis: A neuromotor inspired approach for biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):667–680, 2014. 3

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 2, 5

[9] Glenn Jocher et. al. yolov5. https://github.com/ultralytics/yolov5, 2021. 2

[10] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017. 2

[11] M Muzaffar Hameed, Rodina Ahmad, Miss Laiha Mat Kiah, and Ghulam Murtaza. Machine learning-based offline signature verification systems: A systematic review. *Signal Processing: Image Communication*, 93:116139, 2021. 1, 2

[12] Marcel René Hauri. *Detecting signatures in scanned document images*. PhD thesis, Vienna University of Technology, 2021. 1, 2

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017. 6

[14] Juan Hu, Zhenhua Guo, Zhenyin Fan, and Youbin Chen. Offline signature verification using local features and decision trees. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(03):1753001, 2017. 2, 3

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 7

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 7

[17] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–666, 2006. 3

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 2

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 5

[20] Marcus Liwicki, Muhammad Imran Malik, C Elisa Van Den Heuvel, Xiaohong Chen, Charles Berger, Reinoud Stoel, Michael Blumenstein, and Bryan Found. Signature verification competition for online and offline skilled forgeries (sigcomp2011). In *2011 International Conference on Document Analysis and Recognition*, pages 1480–1484. IEEE, 2011. 1, 2, 3

[21] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 7

[22] Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, and Christopher Mutschler. Joint classification and trajectory regression of online handwriting using a multi-task learning approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 266–276, 2022. 3

[23] Srikanta Pal, Alireza Alaei, Umapada Pal, and Michael Blumenstein. Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 72–77. IEEE, 2016. 3

[24] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017. 2, 5

[25] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 5

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 5

[27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 7

[28] Nabin Sharma, Ranju Mandal, Rabi Sharma, Umapada Pal, and Michael Blumenstein. Signature and logo detection using deep cnn for document image retrieval. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 416–422. IEEE, 2018. 2

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[30] Mohamed Ali Souibgui and Yousri Kessentini. De-gan: A conditional generative adversarial network for document enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[31] Harish Srinivasan, Sargur N Srihari, and Matthew J Beal. Machine learning for signature verification. In *Computer Vision, Graphics and Image Processing*, pages 761–775. Springer, 2006. 3

[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI Conference on Artificial Intelligence*, 2017. 7

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 5

[34] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, 2008. 7

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 6

[36] Ping Wei, Huan Li, and Ping Hu. Inverse discriminative networks for handwritten signature verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5764–5772, 2019. 1, 2, 3

[37] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017. 7

[38] Peng Xu, Timothy M Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang. Deep learning for free-hand sketch: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[39] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, and Jun Guo. Sketchmate: Deep hashing for million-scale human sketch retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8090–8098, 2018. 2

[40] Peng Xu, Yongye Huang, Tongtong Yuan, Tao Xiang, Timothy M Hospedales, Yi-Zhe Song, and Liang Wang. On learning semantic representations for large-scale abstract sketches. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9):3366–3379, 2020. 2

[41] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multigraph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 2

[42] Peng Xu, Kun Liu, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, Jun Guo, and Yi-Zhe Song. Fine-grained instance-level sketch-based video retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1995–2007, 2020. 3

[43] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*, 2021. 4

[44] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2849–2857, 2017. 2

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 6

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 7