

Offline Signature Verification on Real-World Documents

Deniz Engin^{1*}Alperen Kantarcı^{2*}Seçil Arslan¹Hazım Kemal Ekenel²¹Yapı Kredi Technology²Istanbul Technical University

{deniz.engin, secil.arslan}@ykteknoloji.com.tr

{kantarcia, ekenel}@itu.edu.tr

Abstract

Research on offline signature verification has explored a large variety of methods on multiple signature datasets, which are collected under controlled conditions. However, these datasets may not fully reflect the characteristics of the signatures in some practical use cases. Real-world signatures extracted from the formal documents may contain different types of occlusions, for example, stamps, company seals, ruling lines, and signature boxes. Moreover, they may have very high intra-class variations, where even genuine signatures resemble forgeries. In this paper, we address a real-world writer independent offline signature verification problem, in which, a bank's customers' transaction request documents that contain their occluded signatures are compared with their clean reference signatures. Our proposed method consists of two main components, a stamp cleaning method based on CycleGAN and signature representation based on CNNs. We extensively evaluate different verification setups, fine-tuning strategies, and signature representation approaches to have a thorough analysis of the problem. Moreover, we conduct a human evaluation to show the challenging nature of the problem. We run experiments both on our custom dataset, as well as on the publicly available Tobacco-800 dataset. The experimental results validate the difficulty of offline signature verification on real-world documents. However, by employing the stamp cleaning process, we improve the signature verification performance significantly.

1. Introduction

Handwritten signatures are one of the oldest and most widely used biometric authentication techniques in administrative and financial institutions due to its simplicity and uniqueness [16]. As technology progresses, authentication methods have also evolved. Handwritten signatures are now

*indicates equal contribution

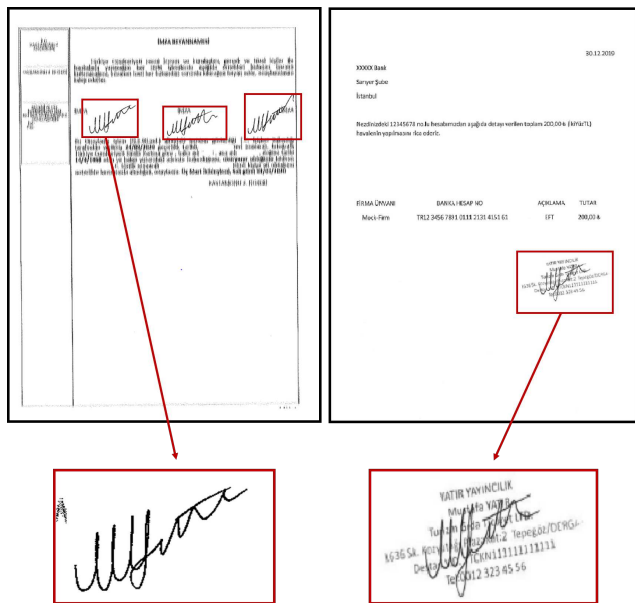


Figure 1: Example of verifying a signature extracted from a signature declaration document (left) and the stamped signature extracted from an order document (right).

categorized as online signatures and offline signatures. Online signatures have much more distinct features than offline signatures; therefore, they are easier to verify [6]. However, capturing online signatures is expensive, and digital systems prefer different authentication methods, such as passwords or personal authentication questions. On the other hand, offline signatures are easy to capture but hard to verify due to the limited amount of features they contain and uncontrolled environmental acquisition conditions.

Offline signature verification task has been a challenge for computer vision research and many different approaches have been proposed to perform the task more accurately [16]. Evaluations of these approaches have been conducted on publicly available datasets such as GPDS-960 [19], GPDS-4000 [5], MCYT [15], and CEDAR [9].

All of these datasets contain genuine signatures of the users with random and skilled forgeries that try to imitate the genuine signature. The collection of the signatures is completed in either single or multiple sessions. People tend to sign very similar signatures when they sign one after another, similar to the one session acquisition, however, signatures differ very much when signatures are collected over time. In real-world applications, signatures of a person can be varied considerably, because people sign a lot of documents in their daily lives, and it is unlikely to sign exactly the same every time. Therefore, datasets, that acquire signatures in a short period, do not capture the high intra-class variety of a person’s signature.

In the literature, both writer dependent and writer independent signature verification methods have been proposed. However, in a real-world signature verification setting user enrollment is very frequent. On account of this reason, writer dependent methods are not feasible to apply. In the writer independent methods, the subjects used for training and testing are different, so no person specific features can be utilized. Writer independent methods try to learn efficient representations of the signatures to distinguish each person, but creating a universal discriminative representation of a signature is challenging and no particular feature extraction method has been found to solve this problem [7].

In this paper, we focus on offline signature verification in the banking process as one of the real-world application scenarios. In the banks, the customers from enterprise and commercial segments send their banking transaction orders with mainly petition-based documents. These documents are received by the central operation unit of the banks from their fax, scanner, and e-mail channels. The operators are responsible for checking the signature, whether it is the same with the one, which is seen on the signature declaration document of the same customer. This task is illustrated in Figure 1. Due to the requirement of a significant manual workforce, we aim to automatize this process for the documents of the companies that have exactly one authorized employee to sign the documents. It is measured that such types of customers send around 90,000 pages of banking order documents per month in the medium-size banks. The signature verification task with a manual workforce requires approximately 233 person-hours to process these documents. Hence, employing an automatic offline signature verification system provides significant resource efficiency to the central operation unit of banks.

In this work, we collect bank order and signature declaration documents of the customer’s¹. The location of signatures on these documents are annotated manually. This

¹Please note that due to data confidentiality, we cannot publish samples from our real-world dataset. Therefore, to visualize our real-world signature verification problem, imitations of signatures, rubber-stamps, and document images are provided from our dataset.

way, we create a real-world signature dataset. Signatures on order documents can be rubber-stamped or unstamped. Therefore, we need a stamp cleaning method to obtain more clear signatures before the verification process. Inspired from image-to-image translation works in the literature, we utilize the CycleGAN [24] for stamp cleaning. We generate two datasets from the created signature dataset, one for representation learning and the other to run verification tests. These two subsets contain signatures from different individuals. Thus, we train a deep feature extraction network on a completely different set of users than the ones in the test set to have a writer independent feature extractor.

Please note that we cannot make our confidential customer signature dataset of the bank publicly available due to the General Data Protection Regulation (GDPR). Furthermore, we cannot use publicly available signature verification datasets, such as GPDS-960 [19], GPDS-4000 [5], MCYT [15], and CEDAR [9], because our problem differs from the one presented by them regarding data collection and application purpose. Therefore, we also prepare another real-world signature verification setup using the publicly available Tobacco-800 dataset [10, 14] and conduct experiments on it. In order to promote signature verification research on real-world documents, we publish the generated training, validation, and verification splits that we use in this benchmark².

Our main contributions can be summarised as follows:

- We present a comprehensive study on offline signature verification on real-world documents. For this purpose, we both create a custom offline signature verification dataset and a real-world signature verification setup using the publicly available Tobacco-800 dataset.
- We extensively analyze different verification setups, fine-tuning strategies, and signature representation approaches. Moreover, we conduct a human evaluation to show the challenging nature of the problem.
- We formulate the stamp removal task as an unpaired image-to-image translation problem and propose a CycleGAN-based stamp removal method. With the proposed framework, we achieve a significant reduction in the equal error rate.

The remainder of the paper is organized as follows. In Section 2, we review the related work. The proposed method is explained in Section 3. Experimental setups and the corresponding results are presented and discussed in Section 4. Finally, Section 5 concludes the paper.

²<https://github.com/Alpkant/Offline-Signature-Verification-on-Real-World-Documents>

2. Related Work

Noise Cleaning. Signatures on the complex documents often overlap with different parts of the documents, such as stamps, ruling lines, printed and handwritten texts, which are called noise in general. Removal of these parts can be seen as a segmentation problem since segmented parts can be removed to extract a clean signature. [23] proposed a fully convolutional stamp segmentation network to detect different kinds of stamps in the documents. Stamps change a lot between companies and countries; therefore, network training for the specific dataset is essential. Their proposed network has been trained with pixel-level stamp annotations; however, creating a pixel level stamp annotation for real-world documents is not feasible. On account of this, we utilize a noise cleaning method, which does not require pixel-level annotations, and it is trained in an unpaired manner.

In [17], a CycleGAN [24] based scanning artifact removal deep network is proposed to clean documents from a variety of noises, *e.g.*, watermark, background noise, and blur. They train their network on four different datasets for four different noise types, however, these datasets are synthetically created. Our proposed noise-cleaning network has been trained on real-world documents. Moreover, we do not constrain our network to a limited number of noise types or degradations. For example, printed and handwritten texts or stamps are also seen as noise along with the other noise types for our network.

Signature Verification. Like all other computer vision problems, handcrafted features have been widely used in the signature verification. [22] built a support vector machine (SVM) classifier on top of combined local binary patterns (LBP) and histogram of oriented gradients (HOG) features. This approach achieved the highest score in IC-DAR SigWiComp challenge both in 2013 [13] and 2015 [12]. Instead of searching good handcrafted features, deep convolutional neural networks have been utilized to learn feature representations from raw data [4, 21, 11]. In [6], the authors investigated the feature representations of the deep learning models specifically for the signatures. Analysis of the features showed that deep learning models could successfully create good representations of the signatures and able to discriminate the genuine signatures. Also, [7] created a writer independent deep neural convolutional network to prove that learned feature space not only generalizes to unseen users in a dataset but also to the users from other datasets. This is also a good indicator of the applicability of the deep convolutional neural networks to the real-world signature verification task. [20] proposed a multiple stream verification network, which uses original and inverse signatures. They claim that their network focuses more on the signature strokes when original and inverse signatures

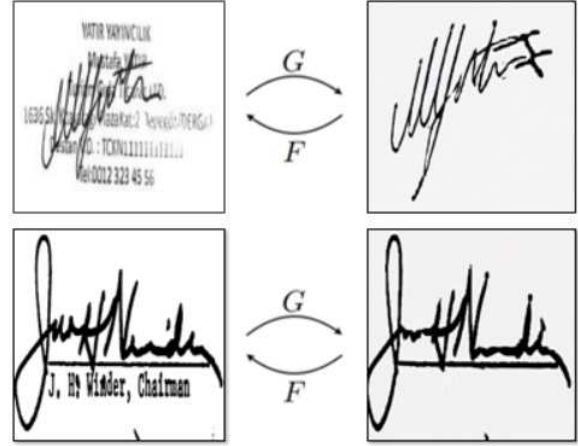


Figure 2: Examples of stamped and cleaned signatures from our dataset (above) and Tobacco-800 (below).

are used together with their inverse streams and multi-path attention modules.

3. Proposed Method

Our proposed system includes two main steps as stamp cleaning and representation learning. In the system, after stamp cleaning, signature representations are extracted. Then similarity between two signature representations is measured and compared to a general threshold to determine whether the signatures belong to the same person or not. In the following subsections, we explain these processes.

3.1. Stamp Cleaning

Signatures on the real-world documents might be stamped, which degrades the verification process. In our dataset, the target signatures generally include a stamp. Thus, a conversion between stamped and unstamped signatures is a critical process for signature verification. For this reason, a stamp cleaning method is necessary. The requirement of an unsupervised method is the primary constraint for the stamp cleaning method due to the difficulty of collecting a large number of stamped and unstamped pairs of signatures from the same users in real-world documents. This limitation motivates us to utilize CycleGAN [24] to perform unpaired image-to-image translation.

We collect a dataset by using the extracted signatures from the documents. There are 1287 signatures extracted from signature declaration documents which are clean, whereas 3607 signatures extracted from the order documents contain stamps. Our aim is to learn the conversion between stamped signatures, X , and unstamped ones, Y . For this purpose, two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$ are defined.

The adversarial loss for mapping function $G : X \rightarrow Y$ is given in Equation 1. Adversarial loss for mapping function $F : Y \rightarrow X$ is also similar to this adversarial loss.

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{cun}}(x)} [\log (1 - D_Y(G(x)))] \quad (1)$$

As an improvement to adversarial loss, cycle consistency loss has been proposed in CycleGAN to compare generated images with input images using the cyclic process. In cycle consistency loss described in Equation 2, the L1 norm is employed to calculate the loss between generated inputs and original inputs.

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{dea}}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{dea}}(y)} [\|G(F(y)) - y\|_1] \quad (2)$$

The full objective of CycleGAN, which consists of adversarial losses in two ways and cycle consistency loss, is given in Equation 3.

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GNN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \quad (3)$$

The sample inputs and outputs of our cleaning process can be seen in Figure 2. Our trained model is able to remove texts successfully on images in both datasets.

3.2. Representation Learning

Writer dependent signature verification models are not feasible for real-world signature verification scenarios where user enrollment is very frequent. Therefore, we should learn writer independent signature representations to verify signatures. For this purpose, we benefit from well-known, successful architectures, namely, VGG-16 [18] and ResNet-50 [8], and their pre-trained models on ImageNet [3]. For each dataset, we fine-tune these networks' models with signatures of the users in the training set. In the verification test set, we have signatures of the users that our networks have never seen before. For each network architecture, we fine-tune three models with different settings: raw signature images, cleaned signature images, and inverse signature images. By changing the input image type, we explore the effect of the cleaned and inverse signature images.

3.3. Verification

Figure 3 illustrates the feature extraction and verification process. Two signatures are fed into the model, and their features are extracted. Cosine similarity is calculated between the extracted features. Finally, the obtained similarity score is thresholded to determine whether the signatures belong to the same person or not.

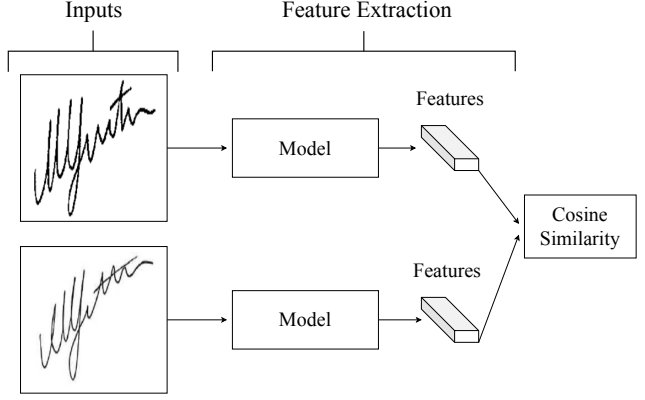


Figure 3: Feature extraction and verification process.

More specifically, the first fully-connected layer of VGG-16 and the second last convolution layer of ResNet-50 are chosen for feature extraction. Accordingly, we obtain a feature vector with size of 4096 from VGG-16 and a feature vector size of 25088 from ResNet-50. Then, we employ cosine similarity to measure the similarity between extracted feature vectors. After calculating the similarity for a pair, a label is assigned according to a specified threshold.

In this paper, we present the results in terms of global equal error rate ($\text{EER}_{\text{global}}$), based on a global threshold value, and ROC curves. Defining a threshold value for each user is not feasible for a real-world signature verification system, where new users enroll frequently with a few samples provided in a single session.

4. Experimental Results

In this section, we first present the datasets and the experimental setups. Then, we will give information about the implementation details. Finally, the objective and subjective evaluation results are provided and discussed.

4.1. Datasets

We collect signatures from two sources: order documents and signature declaration documents. A sample signature declaration document and an order document can be seen in Figure 1. *Order documents* include the transaction order of the customers and must be signed by them. Customers must also declare their signatures on *signature declaration documents*. According to the regulations, each person signs three times on signature declaration documents. Signatures extracted from the signature declaration documents are named *reference signatures* of the customers and these are unstamped signatures. On the other hand, signatures extracted from the transaction order documents are named *target signatures*. These signatures can be rubber-stamped or unstamped, which are named as *stamped*, and *unstamped* signatures, respectively.

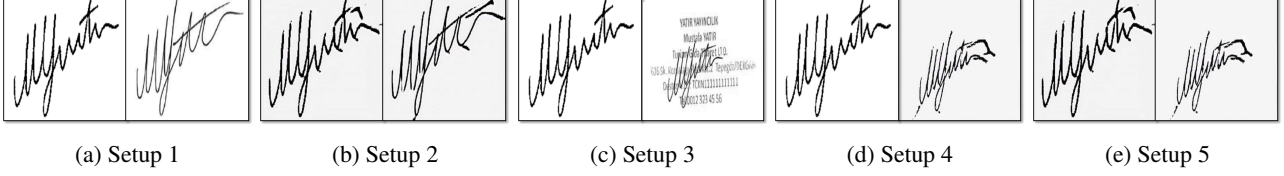


Figure 4: Example pairs of verification test setups. (a) Setup 1: Reference - Unstamped Target, (b) Setup 2: Cleaned Reference - Cleaned Unstamped Target, (c) Setup 3: Reference - Stamped Target, (d) Setup 4: Reference - Cleaned Stamped Target, (e) Setup 5: Cleaned Reference - Cleaned Stamped Target

Table 1: Verification test setups

Test setups	Signature Pairs	
Setup 1	Reference	Unstamped Target
Setup 2	Cleaned Reference	Cleaned Unstamped Target
Setup 3	Reference	Stamped Target
Setup 4	Reference	Cleaned Stamped Target
Setup 5	Cleaned Reference	Cleaned Stamped Target

Our dataset is categorized into two sub-datasets: (i) representation learning dataset, (ii) verification test dataset. The representation learning dataset is utilized for training a model to learn signature representations. The verification test dataset includes signature pairs (reference signatures and target signatures) to evaluate the signature verification performance. In these datasets, we selected the individuals from whom the bank has received a high number of orders. These two subsets contain different sets of customers, that is, a customer’s signatures are included in only one of these two subsets leading to a person independent setup.

Representation Learning Dataset: This dataset consists of 109 individuals’ signatures. After applying data augmentation, such as thickening, rotation, and random distortion, each individual has at least 80 signatures. In total, we have approximately 9K signatures. Finally, we split this dataset randomly into training, validation, and test sets with a proportion of 70%, 15%, and 15%, respectively.

Verification Test Dataset: We have two sets of test pairs of signatures from 178 individuals: unstamped pairs and stamped pairs, which consist of reference and target signatures. Unstamped pairs of signatures contain 2609 pairs, which consist of 1001 matched pairs and 1608 mismatched pairs. On the other hand, stamped pairs of signatures contain 2630 pairs, which have 1022 matched pairs and 1608 mismatched pairs. Five different experimental setups are prepared in order to assess the effects of different cases as listed in Table 1. Corresponding sample pairs of these setups can be seen in Figure 4. Please note that the signature images in this figure are resized for visualization pur-

poses. In the first setup, we compare a reference signature with an unstamped signature. In the second setup, we apply our stamp cleaning method both on the reference and unstamped target signature. This is to evaluate the effect of performing a stamp cleaning process when both reference and target signature does not contain any stamps. This could happen, since, at the moment, we do not employ a stamp detection method and apply stamp cleaning on all signatures extracted from the order documents. Stamped target and reference signature are compared in the third setup. This setup is to observe the degree of performance loss when the target signature contains a stamp. In the fourth setup only the stamped target signature is cleaned. This setup is to assess the effect of stamp removal on signature verification performance. Finally, in the fifth setup, both reference and stamped target signatures are cleaned. This setup is to observe the effect of slight artifacts from the cleaning process on the verification performance. Moreover, we could have defined another setup consisting of stamped reference signatures and stamped target signatures. In this case, we should have added a stamp on the reference signature by generating a random stamp; however, the generated stamp cannot be the identical stamp with the target signatures. Since different stamps on the reference and target signatures lead to a decrease in the similarity of these signatures, this is not an appropriate setup for our problem.

Tobacco-800 dataset [10, 14] is a publicly available subset of 42 million pages of documents that are scanned with various equipment. It contains real-world documents and unlike most of the publicly available signature datasets, it contains noises and artifacts, such as stamps, handwritten texts, and ruling lines, on the signatures. Figure 5 shows example signatures of different users from the Tobacco-800 dataset. The resolution of the documents varies between 150 and 300 DPI. All signatures are manually annotated in this dataset. Also, the identification of the users has been done manually by considering the signers’ names in the document. There are some mislabeled or unidentified signatures. These mislabeled signatures and signatures without user identities have been removed from the dataset. In the end, 746 signatures of 130 users remained. The number of signatures for each user varies, for example, some users

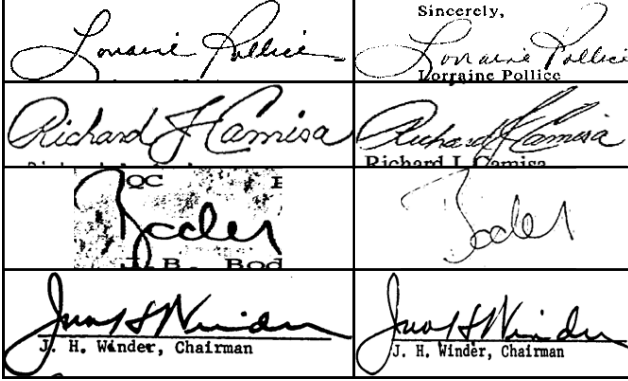


Figure 5: Example signatures of four different individuals from Tobacco-800 dataset.

have just one signature. We use randomly selected 60 users to perform representation learning. After applying the same data augmentation strategies with our dataset, we obtain approximately 4200 signatures in total for training.

To perform a writer independent signature verification, we use the remaining 70 users for the test set. 41 of these users only have one signature; therefore, they are only used to generate negative pairs. The remaining 29 users have a minimum of two and a maximum of seven signatures. From these user signatures, we generate all possible positive pairs, which are 166 in the test set. We randomly create the same number of negative pairs by using all the test users. In total, we formed 332 signature pairs.

4.2. Implementation Details

We implement our models in Tensorflow [1] and Keras [2] frameworks. We train our model with NVIDIA GTX 1080Ti graphics card. We perform fine-tuning on ResNet-50 and VGG-16 models with batch size of 32 and 64, respectively. We utilize the SGD optimizer with momentum. The learning rate in the initialization varies in the range of 0.001 and 0.0001. Early stopping is employed by controlling validation loss for specified consecutive epochs.

4.3. Objective Evaluation

We run experiments using five different test setups, three different use of fine-tuning data, and three different representations of signature images –original, cleaned, and inverse– as input.

Effects of cleaned input images. We investigate the effectiveness of the stamp cleaning process on signature verification. We train VGG-16 and ResNet-50 on raw input images and cleaned input images, separately. The models trained on the cleaned input images are denoted as VGG-16_{clean} and ResNet50_{clean}. We then test these models on five test setups and compare the results. According to Ta-

ble 2, the experimental results indicate that the stamps lead to significant degradation of the performance. For example, the obtained EER_{global} with the VGG-16 model is 0.18, when there are no stamps in the target signatures. The EER_{global} increases dramatically to 0.33, when the target signatures contain stamps. However, the cleaning process compensates for this performance loss to a large extent and brings the EER_{global} down to 0.23. This observation is consistent in all the experiments, therefore, independent of the used network model, fine-tuning data, and the input image representation. VGG-16_{clean} model is found to be better than the others in almost all test setups on our dataset.

ROC curves for all the models are plotted in Figure 6. Each ROC curve includes the results of five test setups to compare the effects of the cleaning process. As can also be observed from the ROC curves, when *Stamped Target Signatures* are cleaned, the performance increases. When *Unstamped Target Signatures* are cleaned without necessity, the performance does not get affected much. Due to the slight artifacts caused by the cleaning process, applying stamp removal also on the clean reference signature leads to either a slight performance improvement or does not change the performance, depending on the experimental setup.

We then evaluate our best performing models on the cleaned test pairs of the Tobacco-800 dataset. That is, we train VGG-16 and VGG-16_{cleaned} models on the Tobacco-800 and cleaned Tobacco-800 training sets, respectively. As can be seen from Table 2, since the Tobacco-800 dataset also consists of real-world documents, the results are similar to the ones that we have obtained on our custom dataset, which validates the difficulty of the problem.

Effects of inverse input images. For offline signature verification, signature images are digitalized by the scanners. Original images contain a white background and black or blue signatures when scanned. In signature verification literature, we notice that most of the work use binarized signature images with black background and white signatures instead of directly using binarized signature images with white background and black signatures. Therefore, we trained our models with both original and inverse images to see the effect of image representation on the performance. From Table 2, it can be observed that image representation does not affect the verification accuracy significantly.

To investigate the effect of image representation further, we visualize the five most activated convolution filters of the last convolutional layer for the VGG-16 model. Figure 7 shows that both models, either trained with original or inverse images, learn similar features from the signatures. Visualizations indicate that most activated five convolutional filters concentrate on the same regions of the signatures.

Table 2: Signature verification results

Test Setups	EER_{global}					
	VGG-16	VGG-16 _{cleaned}	VGG-16 _{inverse}	ResNet-50	ResNet-50 _{cleaned}	ResNet-50 _{inverse}
Reference Signature - Unstamped Target Signature	0.18	0.16	0.18	0.20	0.20	0.20
Cleaned Reference Signature - Cleaned Unstamped Target Signature	0.18	0.17	0.18	0.19	0.19	0.20
Reference Signature - Stamped Target Signature	0.33	0.31	0.32	0.34	0.34	0.34
Reference Signature - Cleaned Stamped Target Signature	0.23	0.23	0.22	0.26	0.26	0.26
Cleaned Reference Signature - Cleaned Stamped Target Signature	0.22	0.22	0.23	0.26	0.26	0.23
Tobacco-800	0.24	0.17	-	-	-	-

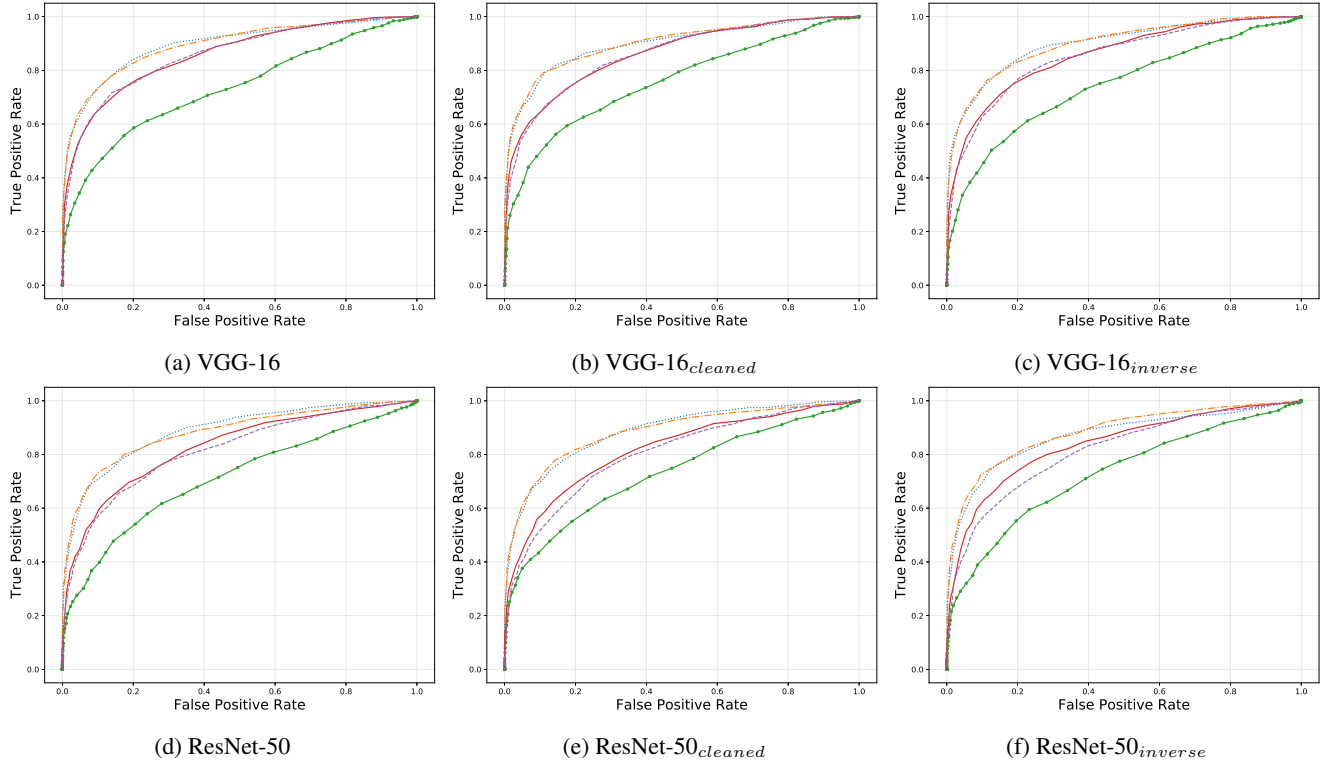


Figure 6: ROC curves of the model on different experimental setups.

4.4. Subjective Evaluation

To assess the difficulty of the problem, we also perform a subjective evaluation by 18 volunteers. We randomly select 360 pairs from our dataset. The subjective evaluation test set includes 180 reference - stamped pairs of signature and 180 reference - unstamped pairs of signatures. These 360 pairs are divided equally into six subsets. Each participant is shown 60 pairs and expected to decide whether the shown signature pair belongs to the same individual or not. This way, each pair is evaluated by three individuals. We provide human evaluation results via majority voting and

individual. For majority voting, we assign the human prediction for each pair to whichever prediction is in the majority in the human prediction set. On the other hand, for individual results, we assume having 1080 pairs of signatures and evaluate the prediction of each individual separately.

Table 3: Result of subjective evaluation

Evaluation Method	Accuracy (%)		
	Human	VGG-16 _{cleaned}	ResNet-50 _{cleaned}
Majority Voting	91.66	76.38	75.00
Individual	89.25		

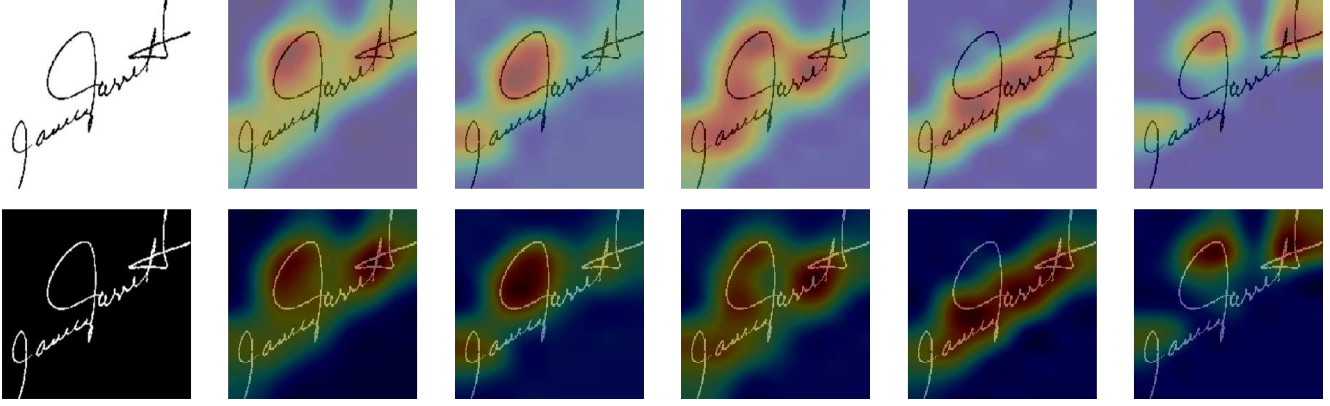


Figure 7: Response maps with five different filters that have produced highest energy activations in the last convolution layer of VGG-16 network when networks are trained with original signature images where background is white and signature is black (top left) and trained with inverse signature images where background is black and signature is white (bottom left).

In order to compare human vs. machine performance, we also run signature verification experiments with the proposed system on the selected 360 pairs for the subjective evaluation. The models fine-tuned on the cleaned signatures, namely VGG-16_{cleaned} and ResNet-50_{cleaned}, are chosen to extract features. EER_{global} is calculated on these pairs, and the threshold value according to this EER_{global} is used to calculate the accuracy of the models.

Results of human evaluation, along with the accuracies obtained by the models, are given in Table 3. The results show the challenging nature of the task as even humans cannot predict all the pairs correctly. Model accuracies on this subset are lower than the ones obtained on the overall test set in Table 2, which indicates that the chosen subset includes harder pairs. Comparing human and model performances, it is clear that we still need further improvements in the system to match human performance.

5. Conclusion

In this paper, we have presented a comprehensive study on writer independent offline signature verification in a real-world scenario, where occluded signatures of a bank’s customers’ are verified against their clean reference signatures. We have proposed a CycleGAN based stamp removal method to clean signatures before feeding them to a CNN model to extract the signature representation. We have compared different verification setups, fine-tuning strategies, and signature representation approaches and analyzed their effects. In order to show the difficulty of the problem, we have also conducted a human evaluation. We have shown the challenging nature of the problem and effectiveness of our proposed stamp cleaning method in our experiments both on our custom dataset and on publicly available Tobacco-800 dataset.

Acknowledgements. We would like to thank our colleagues from Applied AI and R&D Department of Yapı Kredi Technology, especially Ali Yeşilkanat and Mehmet Yasin Akpınar for their support and valuable comments.

References

- [1] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 6
- [2] F. Chollet et al. Keras. <https://keras.io>, 2015. 6
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009. 4
- [4] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal. SigNet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017. 3
- [5] M. A. Ferrer, M. Diaz, C. Carmona-Duarte, and A. Morales. A behavioral handwriting model for static and dynamic signature synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1041–1053, 2017. 1, 2
- [6] L. G. Hafemann, R. Sabourin, and Luiz S. Oliveira. Analyzing features learned for offline signature verification using deep CNNs. In *International Conference on Pattern Recognition (ICPR)*, pages 2989–2994, 2016. 1, 3
- [7] L. G. Hafemann, R. Sabourin, and L. S. Oliveira. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176, 2017. 2, 3
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4
- [9] M. K. Kalera, S. Srihari, and A. Xu. Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(07):1339–1360, 2004. 1, 2

- [10] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. Building a test collection for complex document information processing. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 665–666, 2006. 2, 5
- [11] C. Li, F. Lin, Z. Wang, G. Yu, L. Yuan, and H. Wang. DeepHSV: User-independent offline signature verification using two-channel CNN. In *ICDAR*, 2019. 3
- [12] M. I. Malik, S. Ahmed, A. Marcelli, U. Pal, M. Blumenstein, L. Alewijnse, and M. Liwicki. ICDAR 2015 competition on signature verification and writer identification for on-and off-line skilled forgeries (SigWiComp 2015). In *ICDAR*, pages 1186–1190, 2015. 3
- [13] M. I. Malik, M. Liwicki, L. Alewijnse, W. Ohyama, M. Blumenstein, and B. Found. ICDAR 2013 competitions on signature verification and writer identification for on-and offline skilled forgeries (SigWiComp 2013). In *ICDAR*, 2013. 3
- [14] University of Maryland, Laboratory for Language, and Media Processing (LAMP). Tobacco-800 signatures and logos dataset. <http://lamp.cfar.umd.edu>, 2006. 2, 5
- [15] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J. Igarza, C. Vivaracho, D. Escudero, and Q. Moro. MCYT baseline corpus: a bimodal biometric database. *IEEE Proceedings-Vision, Image and Signal Processing*, 150(6):395–401, 2003. 1, 2
- [16] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000. 1
- [17] M. Sharma, A. Verma, and L. Vig. Learning to clean: A GAN perspective. In *ACCV*, pages 174–185, 2018. 3
- [18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 2014. 4
- [19] J. Vargas-Bonilla, M. Ferrer, C. Travieso, and J. Alonso. Off-line handwritten signature GPDS-960 corpus. *ICDAR*, 2:764–768, 2007. 1, 2
- [20] P. Wei, H. Li, and P. Hu. Inverse discriminative networks for handwritten signature verification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5764–5772, 2019. 3
- [21] M. Yilmaz and K. Ozturk. Hybrid user-independent and user-dependent offline signature verification with a two-channel CNN. In *CVPRW*, pages 639–6398, 2018. 3
- [22] M. B. Yilmaz, B. Yanikoglu, C. Tirkaz, and A. Kholmatov. Offline signature verification using classifier combination of HOG and LBP features. In *International Joint Conference on Biometrics*, pages 1–7, 2011. 3
- [23] J. Younas, M. Z. Afzal, M. I. Malik, F. Shafait, P. Lukowicz, and S. Ahmed. D-StaR: A generic method for stamp segmentation from document images. In *ICDAR*, volume 1, pages 248–253, 2017. 3
- [24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2, 3