# AVN: An Adversarial Variation Network Model for Handwritten Signature Verification

Huan Li, *Student Member, IEEE*, Ping Wei, *Member, IEEE*, and Ping Hu

***Abstract*—Handwritten signature verification is a crucial yet challenging problem. While previous studies have made great progress in this problem, they learn signature features passively from given existing data. In this paper, we propose a novel adversarial variation network (AVN) model for handwritten signature verification which mines effective features by actively varying existing data and generating new data. Powered by a proposed novel variation consistency mechanism, the AVN contains three different types of modules unified under one end-to-end framework: the extractor seeks to extract deep discriminative features of handwritten signatures, the discriminator aims to make verification decisions based on the extracted features, and the variator is designed to actively generate signature variants for constructing a more discriminative model. The proposed model is trained in an adversarial way with a min-max loss function, by which the three modules cooperate and compete to enhance the entire model's ability and therefore the signature verification performance is improved. We test the proposed method on four challenging signature datasets of different languages: CEDAR, BHSig-Hindi, BHSig-Bengali, and GPDS Synthetic Signature. Extensive experiments with in-depth discussions validate the effectiveness of the proposed method.**

***Index Terms*—Handwritten signature, variation consistency, adversarial enhancement, neural network.**

## I. Introduction

SIGNATURE modeling is a significant issue in numerous vision and multimedia applications [1]–[4]. With a huge number of signature data emerging worldwide every day, there is a pressing need to develop an intelligent and effective technique for verifying signatures.

The objective of signature verification is to verify the authenticity of a test signature compared with a reference signature. Information sparsity and style vulnerability are two major challenges of signature verification [5]. For signatures in images, signature verification mainly depends on signature stroke structures but not the image colors or intensities. Information sparsity means signature strokes which mainly determine the verification result always only account for a very small proportion of a signature image, for signature strokes are always extremely thin, as shown in Fig. 1. Style vulnerability means a same person may signs arbitrarily with different writing styles on different occasions and on the other hand some intentional forgers can skillfully forge another person's genuine signatures. These two challenges are both related to representing and mining signature stroke features.

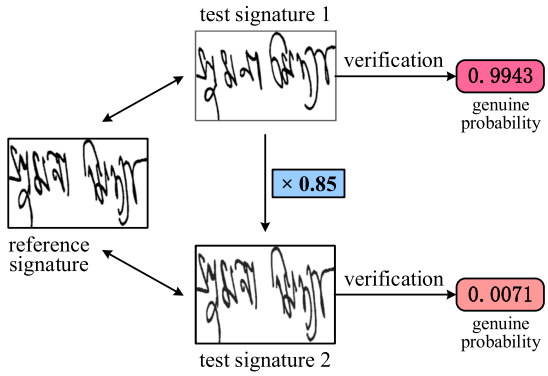


Fig. 1. A signature verification model seeks to determine whether the test signature is genuine compared with the reference signature. The test signature 2 is obtained by multiplying the image pixel values of test signature 1 by a constant 0.85. The same model makes opposite decisions on the two signatures, even if the signature structures were not changed, which is inconsistent with our intuition.

However, modeling the effective stroke features for signature verification is not an effortless task. As shown in Fig. 1, we aim to verify the authenticity of the test signature compared to the reference signature. For the test signature 1, a signature verification model based on neural networks [6] predicts a probability 0.9943 that the test signature is genuine. When the pixel values of the original test signature image slightly vary (multiplying all the pixel values by 0.85), e.g. the test signature 2 in Fig. 1, the model predicts the probability as 0.0071, i.e. the test signature 2 is forged. The same model makes a diametrically opposite verification decision. However, according to our experience, this slight linear variation of pixel values should not change the verification result, because the signature stroke structures and styles do not change, as the test signature 1 and the test signature 2 shown in Fig. 1. We introduce a consistency rule of multiplicative variations in pixel intensity for signature verification, which is defined as *variation consistency*.

***Variation consistency** means for a signature verification model slightly varying the colors or intensities of signature*

Manuscript received March 8, 2020; revised July 25, 2020, October 8, 2020, and December 15, 2020; accepted January 16, 2021. Date of publication February 2, 2021; date of current version February 8, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61876149 and in part by China Postdoctoral Science Foundation under Grant 2018M643657. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfei Cai. *(Corresponding author: Ping Wei.)*

The authors are with the Xi'an Jiaotong University, China (e-mail: lh875056558@stu.xjtu.edu.cn; pingwei@xjtu.edu.cn; helenhu@xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3056217.

Digital Object Identifier 10.1109/TMM.2021.3056217

*images should not change the verification decision of the signatures.*

A desired verification model should meet the variation consistency rule, i.e. it can capture stroke structure information and make the same decisions even the image colors are slightly changed. This observation inspires us that if a model was explicitly trained to make the same decisions for the original signature images and their variants with respect to color, this model would be forced to focus on the stroke information rather than the image colors. Consequently the trained model would gain stronger discriminative and verification abilities.

This paper proposes a novel adversarial variation network (AVN) for writer-independent handwritten signature verification. Our model is writer-independent which means the model is independent of specific writers and one unified model is applied to verification of different signatures. This network contains three major types of modules: extractors, discriminators, and variators. The extractor is used to extract deep features from signature images. The discriminator receives the features and makes verification decisions. The variator aims at generating variants of the original signature images to enhance the entire model. The extractor, discriminator, and variator are integrated into one unified end-to-end framework.

We propose to train the AVN model in an adversarial enhancement way with a new loss function. In the training process, the variator strives to maximize the loss function by producing increasingly complex signature variants online to challenge the extractor and the discriminator. On the adversarial side, the extractor and the discriminator cooperate to minimize the loss function by making correct decisions even being challenged by the generated variants. Based on this min-max game with cooperation and competition, the extractor, discriminator, and variator are all enhanced to improve performance. Since the color of a signature image is varying as the variants are generated in training, the model would be forced to focus on the latent invariant structures and styles of signature strokes. In this way, the effective information for signature verification is captured and mined.

This network is powered by the proposed variation consistency mechanism which is expressed by the variator design in our AVN model. The variator produces variation maps from original extracted features. These variation maps are used to generate variants of the original signature samples to continually challenge the extractor and the discriminator. The effect of the variation consistency is realized by the adversarial enhancement training, by which the extractor, discriminator, and variator are all enhanced in an adversarial min-max game. In this sense, the variation consistency mechanism and the adversarial enhancement training are inseparable aspects, which jointly power the AVN model to improve signature verification performance.

We test our method on four challenging signature datasets of different languages: CEDAR (English) [7], BHSig-Bengali (Bengali) [2], BHSig-Hindi (Hindi) [2], and GPDS Synthetic Signature (Spanish) [8]. Extensive experiments demonstrate the effectiveness of the proposed model.

Our AVN is different from the well-known Generative Adversarial Nets (GAN) [9] or GAN-based models [10], [11]. First, our variator aims to generate variation maps rather than the simulated data samples. Second, our discriminator is used to make final verification decision rather than compute the scores of the generated samples. Third, our model introduces the new variation consistency mechanism. These three aspects differentiate our AVN model from previous methods.

**Three major contributions** are made in this paper compared with other studies:

1) It proposes a novel variation consistency mechanism and a new adversarial variation network model for handwriting signature verification.
2) It proposes a novel adversarial enhancement method with a min-max loss function for training the network and realizing the variation consistency mechanism.
3) It tests the proposed approach on four signature datasets of different languages. The proposed method outperforms the other comparison approaches.

## II. Related Work

We review the related work from the following three main streams of research.

### A. Signature Verification

While previous research efforts have made great progress over the past decades [1]–[3], [5], [12]–[17], there is still much space to address the underlying essential challenges and improve the signature verification performance. The early research efforts extract hand-crafted features and compare the distance between two signatures for verification, such as signature geometric features [1], [18] and local features [12], [19]–[22]. Ferrer *et al.* [23] tried to measure the robustness of gray level features in complex backgrounds and extracted stable features from signatures with different backgrounds. Most previous studies show that hand-crafted features are disturbed by noise and geometric variations. Furthermore, it is difficult for the hand-crafted features to characterize the effective information of signature strokes which involves signature styles and structures. Consequently, the performance of the early approaches still has much room for improvement.

With rapid development of large scale data and hardware resource, various new techniques have been applied to signature verification. One major stream is deep learning and neural network models [3], [11], [14], [24]–[26], which will be detailed in the following section. Other typical methods include meta-learning [27] and one-class method [28]. While these approaches have made impressive progress on signature verification, most of them regard the signification problem as a binary image classification problem and did not specifically consider to mine the information of signature stroke itself but rather the signature image. This may lead to overfitting problems and reduce the model's generalization ability.

### B. Neural Network Models

Neural network models are currently widely used in a variety of tasks, such as recognition [6], [29], [30], detection [31], [32], segmentation [33], [34], and action recognition in videos [35],

[36]. Neural network models for signature verification include two major kinds of frameworks [3], [37]–[39]. The first one is that the reference signature image and the test signature image are concatenated into one single image from which the deep convolutional features are extracted for classification. The second one is extracting deep convolutional features in two streams from the reference signature image and the test signature image respectively. Then the two stream features are concatenated for classification. Siamese neural network [14], [37] extracts deep features of the reference signature image and the test signature image for verification. The analogous two-stream networks were also used in face verification [40], [41] where the two streams of neural network modules share the same architecture and weights to compare the similarity of two face images. Engin *et al.* [42] utilized neural network model to address a real-world writer independent offline signature verification problem.

These previous studies have achieved remarkable progress in signature verification. However, they did not specifically consider the variation mechanism for mining effective stroke information nor formulate it into the network framework. In this paper, we introduce a novel variation consistency mechanism for learning and mining effective signature information.

### C. Adversarial Models

Adversarial models aim to utilize adversarial examples to enhance the models' discriminative or generative abilities. Generative adversarial network (GAN) [9] is one of most well-known adversarial models and has been broadly applied to data generation [4], [10], [43] and other related tasks [44]–[46]. Peng *et al.* [47] utilized an adversarial learning method to formulate a data augmentation network for human pose estimation. The cycle-consistent mechanism was proposed in the CycleGAN model [48] for image-to-image translation. It utilizes a cycle consistency loss to drive the mappings between the original image and the target image to be cycle consistent. This mechanism was later applied to numerous other tasks [49]–[51], such as image-text matching [52], domain adaptation [53], and video understanding [44]. Different from the previous models, our method introduces the new variation consistency mechanism to train a stronger discriminative model. Furthermore, the generator in our model aims to produce true variation maps rather than the fake data samples. Third, the discriminator in our model is to make the final verification decision rather than compute the scores of the generated samples.

Some other studies in machine learning use adversarial examples [29], [54], [55] to boost the security of the methods. Hafemann *et al.* [56] evaluated the impact of the different attack types in signature verification. These studies did not design variation consistency mechanism nor adversarial network architectures for signature verification. Wei *et al.* [5] introduced inverse streams to extract effective information of signatures and force the original-inverse signature pairs to have the same labels. We introduce a variation consistency mechanism and the corresponding adversarial learning scheme into signature verification, which pushes the model to extract effective features of signatures to improve the signature verification performance.

## III. Model

In this section, we will introduce the problem representation, the ideas behind the model, and the architecture of the proposed adversarial variation network in detail.

Suppose $\mathbf{r}$ is a reference signature image and $\mathbf{x}$ is a test signature image. The model aims to decide if $\mathbf{x}$ is genuine or forged compared to $\mathbf{r}$. The output $y \in \{0, 1\}$ is the verification decision label where $y = 1$ indicates the test signature is *genuine* and $y = 0$ means *forged*. The signature verification problem can be represented as

$$y = f(\mathbf{x}, \mathbf{r}; \boldsymbol{\theta}), \tag{1}$$

where $f(\cdot)$ is a decision function which maps the input signature images $\mathbf{x}$ and $\mathbf{r}$ to a verification label $y$. $\boldsymbol{\theta}$ is the set of parameters to be learned from training samples.

It should be noted that the signature verification problem defined in Eq. (1) is similar to but different from traditional binary classification problems in computer vision. Firstly, the traditional binary classification problem generally has one input while the signature verification problem has two inputs $\mathbf{x}$ and $\mathbf{r}$. Secondly, in most traditional binary classification problems, the primary objective is to model and represent the appearance patterns of input data which are based on colors, intensities, and textures. In the signature verification problem, the key task is to characterize the style difference between the bilateral inputs. Writing style is an indescribable attribute defined by signature strokes rather than colors or intensities. Thus how to mine signature stroke features from images is the key point for signature verification.

### A. Variation Consistency Mechanism

General scene images have rich information of colors and textures which play essential roles in discriminating objects in the images. Different from those images, the effective information of signature images is very sparse as signatures are mainly composed of strokes with monotonous colors. On the other hand, signature images collected from different scenarios or different time have considerable variations of colors or intensities. How to overcome these challenges and extract the effective stroke information are critical issues for signature verification.

We propose a novel variation consistency mechanism to address this problem. This mechanism is based on the observation that slightly varying the image colors or intensities should not change the signature verification decision results as signature verification mainly depends on signature stroke structures or styles rather than image colors or intensities. This idea is illustrated in Fig. 2. The original inputs of a test signature image and a reference signature image obtain a verification decision label $y_0$. By varying the intensities of the original reference image and the test image, $m$ reference-test image pairs are generated and their verification decision labels are $y_1, y_2, \ldots, y_m$, respectively. These generated reference-test pairs should have the same verification labels as the intensity variation did not change the

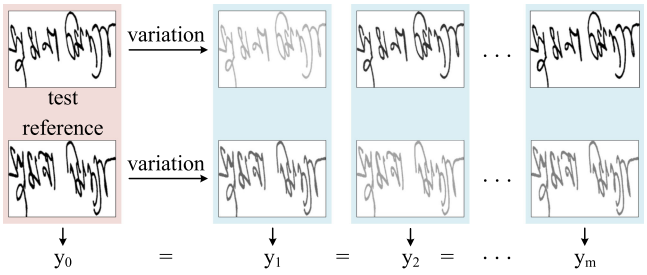


Fig. 2. Illustration of variation consistency mechanism. Slightly changing the image intensities should not change the verification decisions.

signature structures nor writing styles, i.e. $y_1 = y_2 = \cdots = y_m$, as shown in Fig. 2.

Another accompanying problem is how to generate the variants of the signature images. A natural and simple choice is to manually generate these variants offline, such as multiplying the original pixel values by a constant weight or a group of random weights. However, such hand-crafted operations may not produce desired results since the variations of the original images are manually defined but not learned from data. Another more effective method is to generate the variants by learning, i.e. incorporating the generation process into the model and producing the variant images online, by which the generation modules can be trained with other parts of the entire model in an end-to-end framework. We adopt the latter one and propose a new variant generation method based on neural networks. Our later ablation study about these two kinds of variant generation will prove that the proposed variation method is advantageous to the hand-crafted methods.

### *B. Architecture*

Fig. 3 shows the architecture of the proposed adversarial variation network (AVN). The network contains three major types of functional modules: the feature extractor $F$, the discriminator $D$, and the variator $V$, as the different blocks shown in Fig. 3. The whole architecture contains multiple extractors, discriminators, and variators. All the same type modules share the structures and parameters. In the following contents, we will elaborate on the modules first, and then introduce the computing scheme of the architecture.

*1) Feature Extractor $F$:* The feature extractor $F$ extracts deep convolutional features from signature images:

$$\begin{cases} \mathbf{b} = F(\mathbf{x}) \\ \mathbf{d} = F(\mathbf{r}), \end{cases} \tag{2}$$

where $\mathbf{b}$ and $\mathbf{d}$ are the extracted feature maps of the test signature image and the reference image, respectively.

Fig. 4(a) shows the inner structure of a feature extractor. The signature images input to the feature extractor are all resized to the same size of $155 \times 220$. A feature extractor is composed of four cascaded convolutional modules that inspired by the VGG net [6], as shown in Fig. 4(a). Each convolutional module contains two convolutional layers (kernel size $3 \times 3$ and stride 1) with ReLu activation function and one max-pooling layer (kernel size $2 \times 2$ and stride 2). As shown in Fig. 4(a), the kernel numbers of the four modules are 32, 64, 96, and 128, respectively. The feature map sizes after pooling layers of the four modules are $78 \times 110$, $39 \times 55$, $20 \times 28$, and $10 \times 14$, respectively. All the feature extractors $F$ in the AVN architecture share the structures and parameters that are learned to extract effective signature stroke features.

*2) Discriminator $D$:* As shown in Fig. 3, the discriminator $D$ takes the extracted signature features $\mathbf{b}$ and $\mathbf{d}$ as inputs and outputs the verification decision probability $D(\mathbf{b}, \mathbf{d}) \in [0, 1]$, where $D(\mathbf{b}, \mathbf{d}) \geq 0.5$ indicates the test signature is genuine and $D(\mathbf{b}, \mathbf{d}) < 0.5$ indicates forged.

Fig. 4(b) shows the structure of the discriminator $D$. It is composed of the convolutional module and the fully-connected layers. The features extracted from the test signature and the reference signature are concatenated and put into a convolutional module with two convolutional layers and one max-pooling layer. The kernel size, stride, and kernel number of the convolutional layer are $3 \times 3$, 1, and 256, respectively. Through a global average pooling (GAP) layer and a fully-connected layers with 256 units, the features are used to compute the verification decision.

*3) Variator $V$:* The variator $V$ is designed to produce variation maps which are applied to the original input signature images to generate the variants of the signature images. $V$ takes the feature maps $\mathbf{b}$ and $\mathbf{d}$ as inputs and outputs the variation maps $V(\mathbf{b})$ and $V(\mathbf{d})$, respectively. $V(\mathbf{b})$ and $V(\mathbf{d})$ are weight maps with the same sizes of the input images $\mathbf{x}$ and $\mathbf{r}$. As shown in Fig. 3, the variants of the original input reference and test signature images are obtained by element-wise multiplying the original images by the variation maps,

$$\begin{cases} \tilde{\mathbf{x}} = V(\mathbf{b}) \otimes \mathbf{x} \\ \tilde{\mathbf{r}} = V(\mathbf{d}) \otimes \mathbf{r}, \end{cases} \tag{3}$$

where $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$ are the variants of original test signature image $\mathbf{x}$ and the reference signature image $\mathbf{r}$, respectively. The symbol $\otimes$ denotes element-wise multiplication.

The variation maps $V(\mathbf{b})$ and $V(\mathbf{d})$ re-weight the pixel intensities of the original images. In this way the intensities of the original images are changed. The element values of $V(\mathbf{b})$ and $V(\mathbf{d})$ are not necessarily uniform, which leads to nonlinear variation of the original signature images.

Fig. 4(c) shows the inner structure of $V$. It receives the feature map $\mathbf{b}$ or $\mathbf{d}$ as input. After the global average pooling (GAP) and a fully-connected layer, the intermediate feature is reshaped to $80 \times 112$. Then an up-sampling with nearest neighbor operation is performed and a convolutional layer with sigmoid activation function is used to output the weight mask $V(\mathbf{b})$ or $V(\mathbf{d})$, which has the same size with the original input signature images.

*4) Computing Scheme:* In our AVN architecture, the computing schemes for inference and training are different. As shown in Fig. 3, the inference scheme contains the solid-line modules and connections, and the learning scheme contains all the modules and connections.

In inference, two streams of the feature extractor $F$ receive the test signature image $\mathbf{x}$ and the reference signature image $\mathbf{r}$, and produce the deep convolutional features $\mathbf{b}$ and $\mathbf{d}$, respectively.

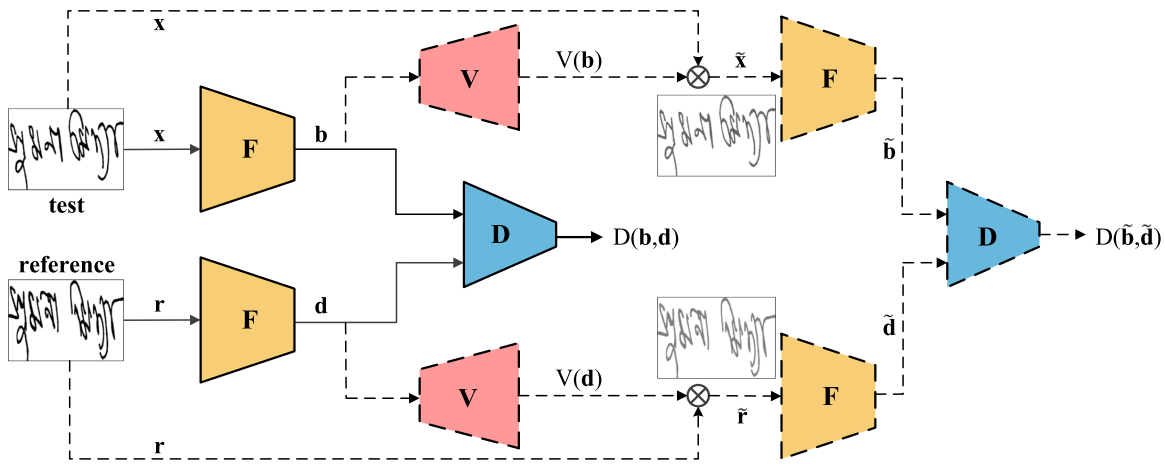


Fig. 3. Architecture of the proposed adversarial variation network (AVN). The inference scheme contains the solid-line modules and connections, and the learning scheme contains all the modules and connections. The four feature extractors $F$, the two discriminators $D$, and the two variators $V$ share the parameters, respectively.

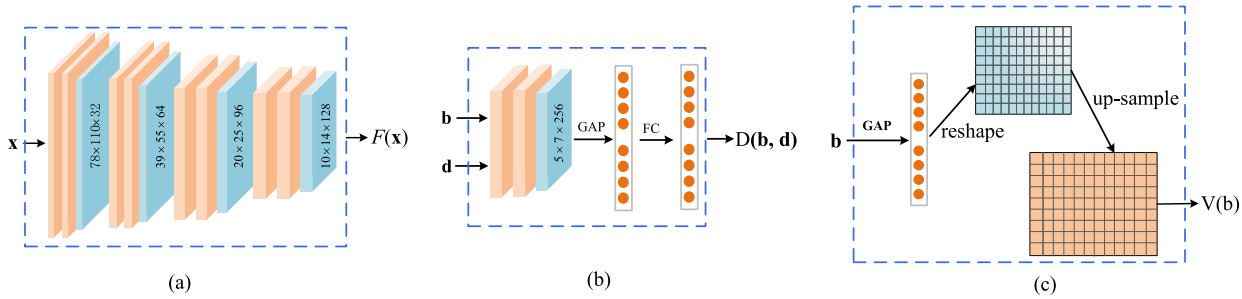


Fig. 4. The structures of $F$, $D$, and $V$. (a) Extractor $F$. (b) Discriminator $D$. (c) Variator $V$. The input is the feature from $F$. The output is a variation map.

The feature maps $\mathbf{b}$ and $\mathbf{d}$ are fed to the discriminator $D$ to compute the final verification result. This process only involves the original images $\mathbf{x}$, $\mathbf{r}$, extractor $F$, and discriminator $D$, without using the variator $V$ or any variants $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$. This process is shown as the solid-line modules connected by the solid lines in Fig. 3.

In training, all the modules in the architecture are involved. Two streams of the variator $V$ receive the feature maps $\mathbf{b}$ and $\mathbf{d}$ and generate the variation maps $V(\mathbf{b})$ and $V(\mathbf{d})$, respectively. The original signature images $\mathbf{x}$ and $\mathbf{r}$ are multiplied by the variation maps to generate the variants $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$, respectively. The variants $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$ are taken as the new test signature image and the reference signature image which are input to the feature extractor respectively to produce the corresponding feature maps $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{d}}$. Taking $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{d}}$ as inputs, the discriminator outputs the verification decision probability $D(\tilde{\mathbf{b}}, \tilde{\mathbf{d}})$ with respect to the variants $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$, as shown in Fig. 3. It should be noted that in the model training, generating variants $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{r}}$ and making verification decisions on the variants are all online processes. A series of variants and decisions are produced as the training proceeds.

An important point in our computing scheme is that the same type modules share the parameters. All the feature extractors addressing the reference signature, the test signature, and their variants share the structures and parameters. In training, the discriminators for original signature images and the variants share the parameters. The variators for the reference signature and the test signature also share the parameters. The parameter sharing between the reference signature modules and the test signature modules pushes the model to focus on the effective information such as stroke styles and structures. Furthermore, the feature extractors and the discriminator are challenged by both the original signature images and a series of the variants, which strengthens the abilities of the feature extractors and the discriminator for signature verification.

## IV. Learning

In the AVN architecture, the extractor aims at extracting the most effective features of signatures and the discriminator seeks to make correct verification decisions. The variator generates variants of original signature images to challenge the extractor and the discriminator. All these modules are unified in an end-to-end framework. We propose a novel adversarial enhancement method to train the model.

### A. Adversarial Enhancement Learning

The adversarial enhancement learning is a strategy for training the model parameters via a cooperation-competition game among the different network modules. In the model training process, the variator produces diverse variants of the original training signature samples online to challenge the extractor and

the discriminator; on the other hand, the extractor and the discriminator cooperatively strive to make correct decisions even being constantly challenged. This strategy is formulated with an adversarial min-max loss optimization, in which the variator seeks to maximize the loss while the extractor and the discriminator cooperate to minimize the loss. In this adversarial game with cooperation and competition, the variator, the extractor, and the discriminator all strive to enhance their own abilities so that they will not be defeated by another side. Finally they become increasingly substantial and the signature verification performance is improved.

The well-known GAN [9] method proposes a min-max loss function to train generative models in an adversarial way. The idea of our adversarial enhancement learning is inspired by but different from GAN. Our model introduces the new variation consistency mechanism, which aims to train a stronger discriminative model while GAN aims to produce a reasonable generator. Second, our variator aims to generate true variation maps rather than the fake data samples. Third, our discriminator is used to make final verification decision rather than compute the truth scores of the generated samples.

### *B. Adversarial Min-Max Loss Function*

Suppose $\{(\mathbf{x}_i, \mathbf{r}_i, y_i) \mid i = 1, \ldots, N\}$ is a set of $N$ training samples, where $\mathbf{x}_i$ and $\mathbf{r}_i$ are the $i$th test signature image and reference signature image, respectively. $y_i \in \{0, 1\}$ is a binary ground truth label of the sample $(\mathbf{x}_i, \mathbf{r}_i)$, where 1 indicates the test signature $\mathbf{x}_i$ is genuine compared with $\mathbf{r}_i$ and 0 indicates forged. With these training samples, we aim to learn the network parameters contained in $F$, $D$, and $V$. Since the learning process involves both original samples and the variants, our loss function is composed of two parts: the loss for the original samples and the loss for the variants.

For the $i$th original sample $(\mathbf{x}_i, \mathbf{r}_i, y_i)$, $F(\mathbf{x}_i)$ and $F(\mathbf{r}_i)$ are the features output from the extractor $F$. $D(F(\mathbf{x}_i), F(\mathbf{r}_i))$ is the signature verification probability predicted by the discriminator. We use the cross entropy function of binary classification to formulate the loss for verification decision. The loss function $L_O(\mathbf{x}_i, \mathbf{r}_i, y_i)$ of verification decision for the $i$th original sample is,

$$\begin{aligned} L_O(\mathbf{x}_i, \mathbf{r}_i, y_i) = &-y_i \ln D(F(\mathbf{x}_i), F(\mathbf{r}_i)) \\ &-(1 - y_i) \ln(1 - D(F(\mathbf{x}_i), F(\mathbf{r}_i))). \end{aligned} \quad (4)$$

Corresponding to the original sample $\mathbf{x}_i$ and $\mathbf{r}_i$, suppose $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{r}}_i$ are the variants generated by the variator online, respectively. According to the variation consistency mechanism, we would like to force the model predicting the same label $y_i$ for $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i)$ with the original signature images $(\mathbf{x}_i, \mathbf{r}_i)$. The verification decision probability for the variants $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i)$ is $D(F(\tilde{\mathbf{x}}_i), F(\tilde{\mathbf{r}}_i))$. Thus the loss function $\boldsymbol{L}_V(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i, y_i)$ of verification decision for the variants $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i)$ is defined as

$$\begin{aligned} \boldsymbol{L}_V(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i, y_i) = &-y_i \ln D(F(\tilde{\mathbf{x}}_i), F(\tilde{\mathbf{r}}_i)) \\ &-(1 - y_i) \ln(1 - D(F(\tilde{\mathbf{x}}_i), F(\tilde{\mathbf{r}}_i))), \end{aligned} \quad (5)$$

**Algorithm 1:** Alternate Min-Max Optimization Algorithm.

1: **Initialization:** $F$, $D$, $V$.
2: **Discrimination optimization:**
set $V$ as $V^*$ in the last iteration, optimizing $F$ and $D$ by $(F, D)^* = \arg\min_{F,D} \boldsymbol{L}(F, D, V^*)$;
3: **Variation optimization:**
set $F$, $D$ as $F^*$, $D^*$ in the last iteration, optimizing $V$ by $V^* = \arg\max_{V} \boldsymbol{L}(F^*, D^*, V)$.
4: **Convergence check**. Iteratively repeat step 2 and step 3 until the maximum iteration step is satisfied.

where the variants $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{r}}_i$ are generated by the variator $V$, i.e. $\tilde{\mathbf{x}}_i = V(F(\mathbf{x}_i)) \otimes \mathbf{x_i}$ and $\tilde{\mathbf{r}}_i = V(F(\mathbf{r}_i)) \otimes \mathbf{r_i}$, as defined in Eq. (3).

In Eq. (4), the arguments to be optimized are the feature extractor $F$ and the discriminator $D$. In Eq. (5), in addition to the arguments $F$ and $D$, the variator $V$ should also be optimized.

The loss function for all the training samples is

$$\boldsymbol{L}(F, D, V) = \sum_{i}^{N} \{\boldsymbol{L}_O(\mathbf{x}_i, \mathbf{r}_i, y_i) + \lambda \boldsymbol{L}_V(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i, y_i)\}, \quad (6)$$

where the weight $\lambda$ balancing the two terms is a hyper-parameter. This loss function contains two parts. $\boldsymbol{L}_O(\mathbf{x}_i, \mathbf{r}_i, y_i)$ corresponds to the original reference and test signature samples. $\boldsymbol{L}_V(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i, y_i)$ characterizes the variants produced by the variator. The hyper-parameter $\lambda$ balances the weights of the two parts.

For verification decision, we hope the extractor and discriminator make correct decisions for both the original signature image pairs and the variant pairs, i.e. the predicted probabilities $D(F(\mathbf{x}_i), F(\mathbf{r}_i))$ and $D(F(\tilde{\mathbf{x}}_i), F(\tilde{\mathbf{r}}_i))$ should approach the ground truth label as much as possible. Thus, the loss functions $\boldsymbol{L}_O$ and $\boldsymbol{L}_V$ should be minimized as much as possible. For variant generation, we hope the variator can help to generate more challenging signature samples, e.g. a genuine sample appear to have some characteristics of the forged ones and a forged sample seems to be a genuine one. In this sense, the generated variants may cause a weak discriminator to make incorrect verification decisions, and therefore the loss function $\boldsymbol{L}_V$ should be maximized with respect to the variator $V$.

Thus, the model parameters are learned by solving the following min-max optimization problem,

$$(F, D, V)^* = \arg\min_{F,D} \max_{V} \boldsymbol{L}(F, D, V). \quad (7)$$

By optimizing Eq. (7), the neural network parameters can be learned from the training samples.

### *C. Optimization*

In Eq. (7), there are two adversarial sides where $F$ and $D$ aim to minimize the loss function while $V$ strives to maximize it. We adopt an alternate scheme to solve the optimization problem of Eq. (7). First, we fix the parameters of the variator $V$

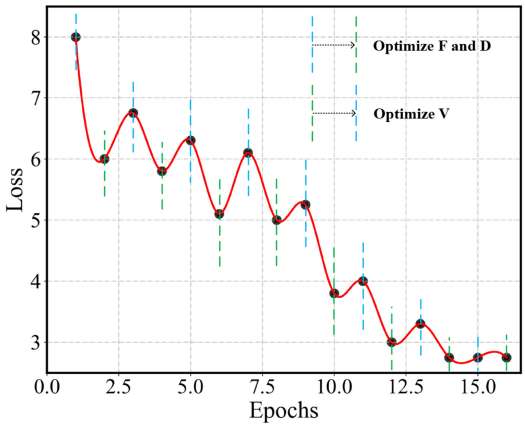


Fig. 5. Illustration of min-max optimization with respect to $F$, $D$, and $V$.

and train the extractor $F$ and the discriminator $D$ by minimizing the loss function which involves both $\boldsymbol{L}_O(\mathbf{x}_i, \mathbf{r}_i, y_i)$ and $\boldsymbol{L}_V(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i, y_i)$. After some steps, the parameters of $F$ and $D$ are fixed and the variator $V$ is trained by maximizing the loss function only involving $\boldsymbol{L}_V(\tilde{\mathbf{x}}_i, \tilde{\mathbf{r}}_i, y_i)$, which pushes $V$ to generate variants to challenge the extractor and discriminator. The two processes iteratively alternate until the maximum iteration step is achieved. This optimization scheme is summarized in Algorithm 1. In Algorithm 1, the two optimization problems are solved using the Stochastic Gradient Descent (SGD) algorithm. The maximum iteration step is defined empirically. Fig. 5 illustrates the min-max optimization process with respect to $F$, $D$, and $V$ with a toy example.

With this alternate adversarial min-max game, the variator, the extractor, and the discriminator all enhance their own abilities to avoid being defeated. Finally they become stronger and the signature verification performance is improved.

## V. Experiments

The proposed model is tested on four different languages signature datasets respectively: CEDAR [7] written in English, BHSig-Bengali [2] signed in Bengali, BHSig-Hindi [2] signed in Hindi, and GPDS Synthetic Signature Database [8] that belongs to Spanish. We also carry out studies including comparison with traditional data augmentation, ablation studies of adversarial learning, and the robustness to disturbance. These extensive experiments and discussions in depth prove the effectiveness and strength of our proposed method.

### A. Experimental Setup

*1) Evaluation Metrics:* In our model, a positive sample is composed of a reference signature and a genuine signature; a negative sample is composed of a reference signature and a forged signature. The evaluation metrics are computed based on these positive and negative samples. We use five metrics to evaluate our method and compare it with other approaches: False Rejection Rate (FRR), False Acceptance Rate (FAR), Equal Error Rate (EER), Accuracy (Acc), and Area Under the Curve (AUC). The number of false rejections divided by the number of positive samples is the FRR. FAR is computed as the ratio of the number of false acceptances over the total number of negative samples. EER is an integrated metric that is the error for FAR = FRR, where the lower EER means the model has a better performance. Accuracy is the percentage of test samples which are correctly predicted in all the test samples. AUC is also a comprehensive metric that is defined as the area under the receiver operating characteristic (ROC) curve.

Our method is writer-independent which we mark as WI in experimental results tables, such as [3], [57]. We also list results of some writer-dependent methods, which are marked as WD, such as [2], [39], [58]. Writer-independent (WI) methods use just one model for all test users and writer-dependent (WD) methods train a specific model for each user. Therefore WD methods usually perform much better than WI methods but needs training samples for each person, which is impractical and can not be generalized to unobserved people. It should be noted that WD methods have different experimental protocols from WI methods. Firstly, datasets are used in different ways. The WI methods divide all the writers into two parts. Some writers' signatures are used to train the model and the rest writers' signature samples are used as the test samples. The WD methods divide every user's signature samples into two parts. Secondly, the evaluation metric computation is different. The WI methods compute evaluation metrics based on the number of positive and negative samples, while the WD methods compute evaluation metrics based on the number of the genuine signatures and the forged signatures. Since the WD methods adopt different training methods and usages of data from WI methods, we list the WD methods here just as reference baselines. The other differences among the methods are also marked in the comparison tables. RN denotes the number of reference samples to verify a test signature. DT denotes the decision threshold to make the verification decision, where 'G' indicates a global decision threshold for all users and 'U' means using different decision threshold for each user. SK describes whether using the skillfully forged signatures or not in the training data, where 'Y' and 'N' represents 'Yes' and 'No' respectively.

*2) Implementation Details:* In experiments, all input signature images are pre-processed using the OTSU algorithm [59] and non-standard Binarization, which transform the background pixel values to 255 and maintain the original pixel values for signature strokes. Then all the images are resized to the same scale of $155 \times 220$. During the training process, we randomly selected about two thousand paired signature samples (one thousand positive samples and one negative samples) from training set as validation set that was used to set the hyper-parameters.

The proposed model is trained based on TensorFlow 1.8.0 framework with NVIDIA 1080Ti. We use a SGD optimization method with a base learning rate of 0.01 to train our model, where the decay factor is set as 0.5 and the batch size 32. The hyper-parameters of batch normalization are set as $\epsilon = 10^{-5}$ and decay 0.99.

### B. CEDAR Dataset

The CEDAR dataset [7] contains 55 people's signatures, and each person has 24 genuine and 24 forged signatures. The signature samples are gray images. Following previous studies, 50

TABLE I
SIGNATURE VERIFICATION COMPARISON ON CEDAR DATASET (%)

| Method | Type | RN | DT | SK | FRR | FAR | EER | Acc |
|---|---|---|---|---|---|---|---|---|
| Morphology [60] | WI | 1 | G | Y | 12.39 | 11.23 | 11.59 | 88.19 |
| Surroundness [61] | WI | 1 | G | Y | 8.33 | 8.33 | - | 91.67 |
| Chain Code [62] | WD | 12 | U | N | 9.36 | 7.84 | - | 92.16 |
| Graph Matching [58] | WD | 16 | U | N | 7.7 | 8.2 | - | 92.1 |
| SigNet-F [39] | WD | 12 | U | N | - | - | 4.63 | - |
| OC-SVM [63] | WD | 12 | U | Y | - | - | 5.60 | - |
| HOCCNN [68] | WD | 12 | U | N | 4.79 | 5.07 | 4.94 | - |
| PDSN [64] | WD | 5 | U | N | - | - | 4.37 | - |
| Triplet Nets-Graph [65] | WD | 10 | G | N | 12.21 | 12.35 | 12.27 | - |
| OSIVCN [66] | WD | 14 | U | Y | - | - | - | 90.00 |
| Ensemble Learning [67] | WI | 1 | G | Y | - | - | - | 92.00 |
| **Our AVN** | **WI** | **1** | **G** | **Y** | **4.42** | **3.26** | **3.77** | **96.16** |

people's signature samples are used to train the model and the rest 5 people's signature samples as the testing set. We pair a reference signature with a genuine signature as a positive sample and a reference signature with a forged signature as a negative sample. In this way, each person has 276 ($24 \times 23/2$) positive samples. For sample balance, we combine the genuine signatures and the forged signatures and randomly select 276 reference-forgery pairs as negative samples for each person. Thus, we have a total of 2760 test samples on CEDAR.

We compare our AVN method with several other approaches: Morphology [60], Surroundness [61], Chain Code [62], Graph Matching [58], SigNet-F [39], OC-SVM [63], PDSN [64], Triplet Nets-Graph [65], OSIVCN [66], Ensemble Learning [67] and HOCCNN [68]. Table I shows the experiment results of different approaches.

On this dataset, our AVN method achieves FRR, FAR, EER, Acc as 4.42%, 3.26%, 3.77%, and 96.16%, respectively, which outperforms other comparison approaches under all metrics. Furthermore, even if comparing with writer-dependent methods, our writer-independent AVN method still has better performance, such as the recently proposed HOCCNN [68], which achieves FRR, FAR, EER as 4.79%, 5.07%, and 4.94%, respectively. These results prove the strength of our AVN method.

The major reasons why our AVN method outperforms other comparison approaches are the variation consistency and the corresponding adversarial enhancement training. In fact, the base network of our AVN model is composed of just two convolutional blocks for feature extraction and fully-connected layers for decision output, which is a rather simple network structure. Under this circumstance, our AVN model still achieves better performance than other comparison approaches, which sufficiently proves the advantage and potential of the proposed framework.

### C. BHSig-Bengali Dataset

BHSig260 dataset [2] consists of the BHSig-Bengali subset and the BHSig-Hindi subset. All the signature samples are binary images. BHSig-Bengali subset is comprised of 100 people's signature images signed in Bengali and each signer has 24 genuine and 30 forged signatures. Following previous studies, 50 people's signature are utilized to train our model and the rest 50 people's signature are used as testing set. For each person, we have 276 reference-genuine pairs as positive samples and 276 reference-forged pairs as negative samples. Totally, we have 27 600 pair samples for test.

TABLE II
SIGNATURE VERIFICATION COMPARISON ON BHSIG-BENGALI DATASET (%)

| Method | Type | RN | DT | SK | FRR | FAR | EER | Acc |
|---|---|---|---|---|---|---|---|---|
| SigNet [3] | WI | 1 | G | Y | 13.89 | 13.89 | - | 86.11 |
| Texture Feature [2] | WD | 12 | U | N | 33.82 | 33.82 | - | 66.18 |
| Correlated Feature [57] | WI | 1 | G | Y | 14.43 | 15.78 | - | - |
| FHTF [69] | WD | 12 | G | N | 23.10 | 18.42 | - | |
| IsRFsM [70] | WD | 12 | G | N | 30.12 | 16.18 | - | |
| DeepHSV [71] | WI | 1 | G | Y | - | - | 11.92 | 88.08 |
| **Our AVN** | **WI** | **1** | **G** | **Y** | **7.33** | **5.07** | **6.14** | **93.80** |

TABLE III
SIGNATURE VERIFICATION COMPARISON ON BHSIG-HINDI DATASET (%)

| Method | Type | RN | DT | SK | FRR | FAR | EER | Acc |
|---|---|---|---|---|---|---|---|---|
| SigNet [3] | WI | 1 | G | Y | 15.36 | 15.36 | - | 84.64 |
| Texture Feature [2] | WD | 12 | U | N | 24.47 | 24.47 | - | 75.53 |
| Correlated Feature [57] | WI | 1 | G | Y | 15.09 | 13.10 | - | - |
| FHTF [69] | WD | 16 | G | Y | 11.46 | 10.36 | - | - |
| IsRFsM [70] | WI | 1 | G | Y | 30.12 | 16.18 | - | - |
| Ensemble Learning [67] | WI | 1 | G | Y | - | - | - | 86.00 |
| DeepHSV [71] | WI | 1 | G | Y | - | - | 13.34 | 86.66 |
| **Our AVN** | **WI** | **1** | **G** | **Y** | **5.91** | **5.46** | **5.65** | **94.32** |

On this dataset, we compare our AVN model with six other approaches: SigNet [3], Correlated Feature [57], FHTF [69], IsRFsM [70], DeepHSV [71] and Texture Feature [2]. The experiment results are shown in Table II. Our model achieves FRR, FAR, EER, and Acc of 7.33%, 5.07%, 6.14%, and 93.80%, respectively, which outperform other approaches by a large margin. These results validate the effectiveness of our method.

### D. BHSig-Hindi Dataset

BHSig-Hindi Dataset is another subset of BHSig260 that includes 160 people's signature images written in Hindi. Each signer also has 24 genuine signatures and 30 forged signatures. Following other studies, 100 people's signatures are used as training samples and the rest 60 people's signatures as testing set. For training, we can obtain $100 \times 276$ reference-genuine pairs as positive samples and $100 \times 276$ reference-forgery pairs as negative samples. For testing set, there are a total of 32 400 ($2 \times 60 \times 276$) pair samples for evaluation.

The experimental results of BHSig-Hindi are presented in Table III. We compare our model with the previous approaches, including SigNet [3], Correlated Feature [57], FHTF [69], IsRFsM [70], Ensemble Learning [67], DeepHSV [71] and Texture Feature [2]. It is obvious that the proposed method achieves great improvement upon the previous methods under all metrics. The proposed model obtains 94.32% Acc, which outperforms

TABLE IV
SIGNATURE VERIFICATION COMPARISON ON GPDS DATASET (%)

| Method | Type | RN | DT | Sk | FRR | FAR | EER | Acc |
|---|---|---|---|---|---|---|---|---|
| Correlated Feature [57] | WI | 1 | G | Y | 27.62 | 28.34 | - | 73.67 |
| HOT [72] | WD | 10 | U | N | 23.41 | 11.96 | 18.15 | - |
| DeepHSV [71] | WI | 1 | G | Y | - | - | 9.95 | 90.05 |
| **Our AVN** | **WI** | **1** | **G** | **Y** | **7.58** | **11.78** | **9.77** | **90.32** |

the existing models by a considerable margin. These results show the advantage of our method over other comparison approaches.

### E. GPDS Synthetic Signature Database

GPDS Synthetic Signature Database [8] contains signatures signed in Spanish. It contains 4000 people's signatures, which are gray images. Each person has 24 genuine signature samples and 30 forged signature samples. Therefore it has a total of $4000 \times 24 = 96,\ 000$ genuine signature samples and $4000 \times 30 = 120,\ 000$ forged signature samples, which makes this dataset one of the largest signature verification dataset. Furthermore, it is also considered to be one of the most challenging dataset for signature verification.

Following previous work, we use 3200 people's samples for training and the rest 800 people's samples for testing. For each person, we have 276 reference-genuine pairs and 276 reference-forgery pairs. There are a total of 883 200 training pairs and 220 800 testing pairs.

We compare our method with other approaches: Correlated Feature [57], HOT [72], DeepHSV [71]. Table IV shows the result comparison. On this dataset, our method achieves an Acc of 90.32% and 9.77% EER respectively. This is a remarkable improvement considering the large scale of this dataset.

### F. Ablation Analysis of Variation Consistency

Our AVN model introduces variation consistency mechanism with adversarial enhancement learning for signature verification. In this section, we compare the proposed AVN model with the base network (BaseNet) model which does not adopt the variation consistency mechanism. The BaseNet model used in our experiments accepts the same preprocessed signature image pairs as inputs and uses the same $F$ to extract features as well as $D$ to make decisions. This BaseNet model is trained in end-to-end ways. The difference from our AVN model is that the BaseNet does not have variators nor train variant samples. We carried out the ablation experiments on CEDAR Dataset, BHSig-Bengali Dataset, BHSig-Hindi, and GPDS Synthetic Signature Database. We use three comprehensive evaluation metrics EER, Acc, and AUC to compare the different methods.

Table V shows the experiment results. It is quite clear that our AVN model outperforms the BaseNet model under all metrics on all four datasets, which proves the effectiveness and advantage of the introduced variation consistency mechanism.

### G. Comparison With Data Augmentation

Our AVN model uses the rariator to produce variants of the original signature images in training. The variator generates variation maps which are multiplied to the original signature images to produce the variant signature images. To some extent, this is like a strategy of data augmentation. Data augmentation is a widely used technique to expand training datasets. We would like to compare our AVN model with data augmentation technique and clarify that our variation generation strategy is superior to the traditional data augmentation.

To compare with the traditional data augmentation method, we remove the variators from our AVN model and train the base model with data augmentation. We compare our method with two data augmentation methods. The first one is UDA - the uniform variation data augmentation. In this method, we vary the intensities of the original signature images by multiplying the pixel intensities of the original image with a random value between 0.5 and 1. For different images the random values are different. The second one is RDA - the random variation data augmentation. Here we vary the intensities of the original signature images by element-wise multiplying the original image with a random value matrix. The values of the matrix are randomly sampled from the range between 0.5 and 1. For different images the random matrices are different. These new generated images are taken as augmentation data to put with the original training data together to train the base model without the variators. The testing set remains unchanged. Here we limit the multiplying factor between 0.5 and 1 because too small multiplying factors in [0, 0.5) may blur the images which is unfair for comparison.

Table VI shows the results of comparison with the two data augmentation methods. Fig. 6 shows the ROC curves of the base model (BaseNet), uniform variation data augmentation (UDA), random variation data augmentation (RDA), and our AVN method on the four testing datasets. Our method outperforms the traditional data augmentation methods by a large margin, which proves the strength and effectiveness of our method.

### H. Comparison of Robustness to Variation

The performance of a signature verification model is prone to be influenced by image intensity variations. In this section, we compare the performance of our method with other methods when changing the intensities of signature images in testing set, by which we can compare the different methods' robustness to intensity variation. We compare our method with other three methods - BaseNet, UDA, and RDA. BaseNet is the basic network structure of our AVN model without variators. It is trained without data augmentation. UDA is the BaseNet model trained by uniform variation data augmentation strategy, as defined in Section V-G. RDA is the BaseNet model trained by random variation data augmentation.

We conduct this comparison by varying the intensities of signature images. The variation is realized by element-wise multiplying an image with a variation matrix. The variation matrix has the same size with the image and each element is a value between 0 and 1. To measure the effect of different variation levels on the performance of the methods, we define five continuous

TABLE V
ABLATION ANALYSIS OF VARIATION CONSISTENCY (%)

| Model | CEDAR | | | BHSig-Bengali | | | BHSig-Hindi | | | GPDS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EER | Acc | AUC | EER | Acc | AUC | EER | Acc | AUC | EER | Acc | AUC |
| BaseNet | 5.51 | 93.71 | 98.84 | 8.26 | 91.28 | 97.40 | 8.97 | 90.74 | 96.84 | 9.96 | 90.09 | 96.47 |
| **Our AVN** | **3.77** | **96.16** | **99.25** | **6.14** | **93.80** | **98.51** | **5.65** | **94.32** | **98.73** | **9.77** | **90.32** | **96.57** |

TABLE VI
COMPARISON WITH DATA AUGMENTATION (DA) (%)

| Model | CEDAR | | | BHSig-Bengali | | | BHSig-Hindi | | | GPDS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EER | Acc | AUC | EER | Acc | AUC | EER | Acc | AUC | EER | Acc | AUC |
| UDA | 5.14 | 95.11 | 99.10 | 9.39 | 90.55 | 97.10 | 10.17 | 89.60 | 96.06 | 10.70 | 89.95 | 95.95 |
| RDA | 7.10 | 93.08 | 98.14 | 8.16 | 91.19 | 97.33 | 9.95 | 89.87 | 96.43 | 11.14 | 88.92 | 95.69 |
| **Our AVN** | **3.77** | **96.16** | **99.25** | **6.14** | **93.80** | **98.51** | **5.65** | **94.32** | **98.73** | **9.77** | **90.32** | **96.57** |

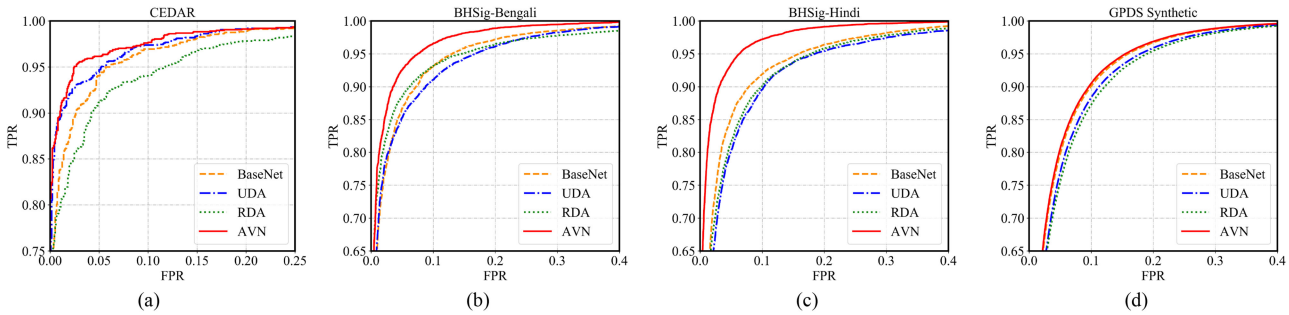


Fig. 6. The ROC curves of the base model (BaseNet), uniform variation data augmentation (UDA), random variation data augmentation (RDA), and our AVN method on the four testing datasets. (a) CEDAR dataset. (b) BHSig-Bengali dataset. (c) BHSig-Hindi dataset. (d) GPDS dataset.

TABLE VII
COMPARISON OF ROBUSTNESS TO UNIFORM VARIATION (ACCURACY (%))

| Variation Range | CEDAR | | | | BHSig-Bengali | | | | BHSig-Hindi | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BaseNet | UDA | RDA | AVN | BaseNet | UDA | RDA | AVN | BaseNet | UDA | RDA | AVN |
| 0.1-0.6 | 74.19 | 74.46 | 70.35 | **75.85** | 50.00 | 75.08 | 56.51 | **79.77** | 50.00 | 87.73 | 84.81 | **89.93** |
| 0.6-0.7 | 79.88 | 87.98 | 86.39 | **89.90** | 50.32 | 89.45 | 81.23 | **92.59** | 50.51 | 89.34 | 88.16 | **94.00** |
| 0.7-0.8 | 89.21 | 93.90 | 89.32 | **94.00** | 60.53 | 89.85 | 86.16 | **93.78** | 58.89 | 89.43 | 89.28 | **94.16** |
| 0.8-0.9 | 91.54 | 95.38 | 91.29 | **95.48** | 81.14 | 90.15 | 88.92 | **94.10** | 75.95 | 89.54 | 89.86 | **94.26** |
| 0.9-1.0 | 93.17 | 95.44 | 92.41 | **96.04** | 89.62 | 90.38 | 90.63 | **93.95** | 87.31 | 89.59 | 90.02 | **94.31** |
| 1.0 | 93.71 | 95.11 | 93.08 | **96.16** | 91.28 | 90.55 | 91.19 | **93.80** | 90.74 | 89.60 | 89.87 | **94.32** |

variation ranges [0.1, 0.6), [0.6, 0.7), [0.7, 0.8), [0.8, 0.9), and [0.9, 1.0). The elements of a variation matrix are randomly sampled from these variation ranges. By sampling from different ranges, the variation levels are changed. For example, multiplying all the pixels of an image with 1 will not change the image, which means there is no variation; multiplying all the pixels of an image with a value sampled from the range [0.1, 0.6) will considerably change the image, which causes large variation to the image.

We vary signature images of the testing set in two ways. The first one is uniform variation. In this case, an image is multiplied element-wise by a variation matrix whose elements are the same, i.e. the intensities of the image are uniformly changed. For an image, one element value is randomly sampled from the variation ranges. For different images the element values (i.e. variation matrices) are different. The second variation method is random variation. In this case, an image is multiplied element-wise by a variation matrix whose elements are different. These elements are also randomly sampled from the variation ranges. And also the variation matrices are different for different images. Fig. 7 illustrates the uniform variation, the random variation, and the corresponding signatures.

We conduct these experiments on the CEDAR, BHSig-Bengali, and BHSig-Hindi datasets. By changing the variation ranges, we compared the performance of our method with other methods at different variation levels. Table VII shows the accuracy comparison in the case of uniform variation and Table VIII shows the results of random variation. These two tables show that for uniform or random variation, in most cases our method outperforms the other methods by a large margin at all variation

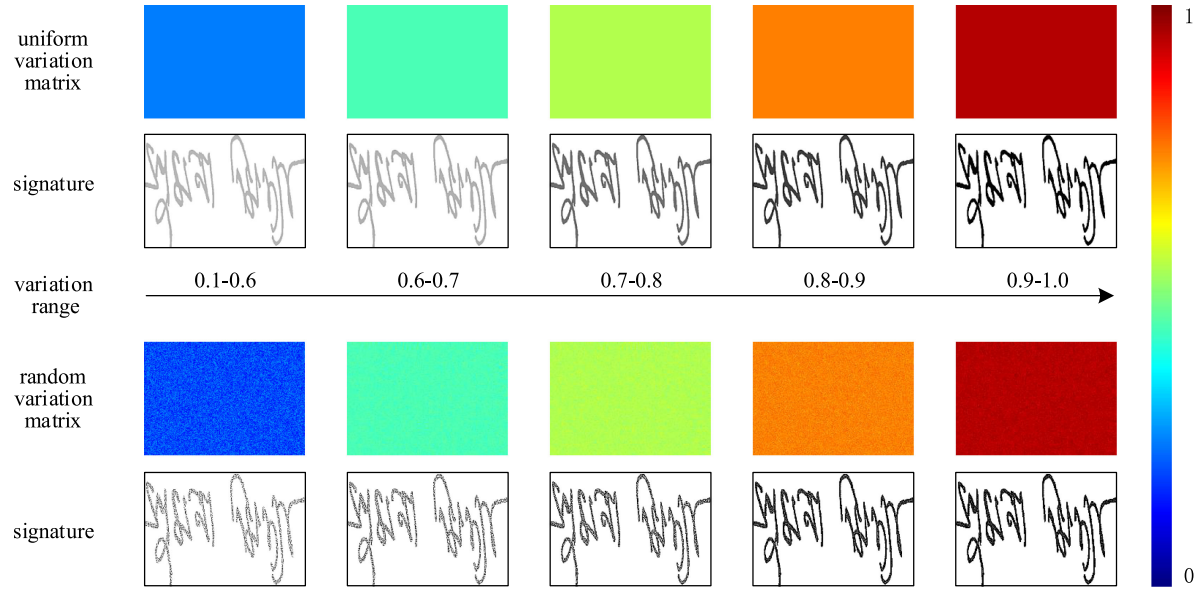


Fig. 7. Visualization of the uniform variation and random variation.

TABLE VIII
COMPARISON OF ROBUSTNESS TO RANDOM VARIATION (ACCURACY (%))

| Variation Range | CEDAR | | | | BHSig-Bengali | | | | BHSig-Hindi | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BaseNet | UDA | RDA | AVN | BaseNet | UDA | RDA | AVN | BaseNet | UDA | RDA | AVN |
| 0.1-0.6 | 60.88 | 51.01 | **75.13** | 59.67 | 50.00 | 56.86 | 59.14 | **73.05** | 50.00 | 88.89 | 84.29 | **91.12** |
| 0.6-0.7 | 75.51 | **86.78** | 86.38 | 85.25 | 50.15 | 89.42 | 81.96 | **92.52** | 50.14 | 89.40 | 88.26 | **94.00** |
| 0.7-0.8 | 82.29 | **92.01** | 89.63 | 89.24 | 57.98 | 89.77 | 86.45 | **93.78** | 58.05 | 89.54 | 89.35 | **94.17** |
| 0.8-0.9 | 86.86 | **92.43** | 91.71 | 92.33 | 80.89 | 90.07 | 89.06 | **94.07** | 76.71 | 89.62 | 89.95 | **94.26** |
| 0.9-1.0 | 90.24 | 92.20 | 92.53 | **94.66** | 88.49 | 90.33 | 90.69 | **93.93** | 88.58 | 89.64 | 90.05 | **94.30** |
| 1.0 | 93.71 | 95.11 | 93.08 | **96.16** | 91.28 | 90.55 | 90.19 | **93.80** | 90.74 | 89.60 | 89.87 | **94.32** |

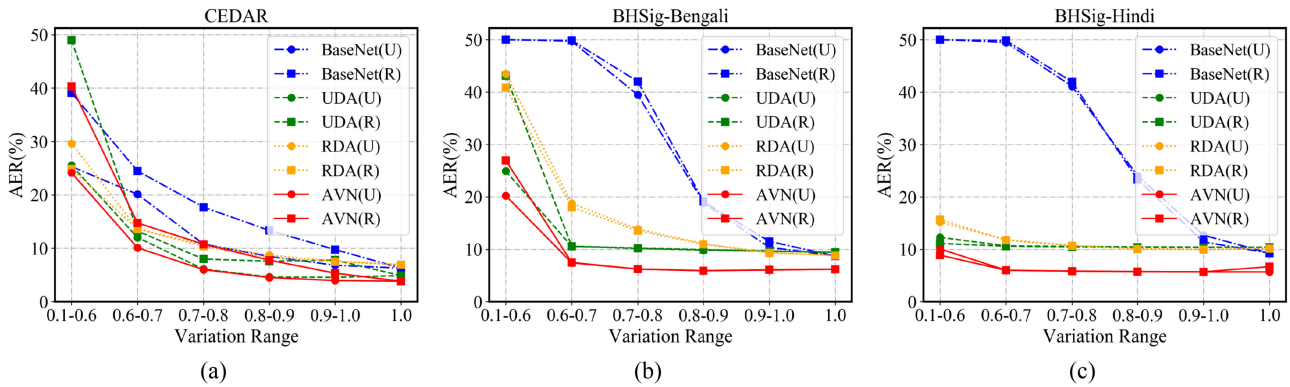


Fig. 8. The average error rate (AER(%)) curves of different methods to uniform and random variations. 'U' and 'R' represent 'uniform' and 'random' respectively. 'BaseNet (U)' means the BaseNet model is tested on the uniform variation testing data. (a) CEDAR. (b) BHSig-Bengali. (c) BHSig-Hindi.

levels. Furthermore, compared to the baseline method, the accuracy of our method decreases slower as the variation range decreases or the image variation increases. Fig. 8 shows the average error rate (AER) curves of these methods. These figures show that for both uniform variation or random variation our AVN method has much lower average error rate. These results prove the advantage and robustness of our method when the input signature images are influenced by intensity variation.

### *I. Discussion*

**1) How effective is the variation consistency mechanism?** Table I, Table II, Table III, and Table IV present the comparison results of the proposed AVN model with other approaches on the datasets CEDAR [7], BHSig-Bengali [2], BHSig-Hindi [2], and GPDS Synthetic Signature [8], respectively. These tables show that our model outperforms other approaches by large margins. Table V shows the ablation study of variation

consistency mechanism, which compares the models with and without variation consistency mechanism. This table shows that the proposed model with variation consistency mechanism performs much better than the baseline model without variation consistency mechanism.

These results prove that the proposed variation consistency mechanism is effective for improving signature verification performance, which can be explained from two aspects. First, compared to other comparison approaches, our AVN model impressively improves the signature verification performance. One of the major reasons lies in the variation consistency mechanism. The base network of our AVN model is a very simple neural network structure which is composed of just two convolutional blocks for feature extraction and two fully-connected layers for decision output. The performance of this base network is even lower than some other comparison models, as shown in Table V, Table I, Table II, Table III, and Table IV. By incorporating the variation consistency mechanism with the corresponding adversarial enhancement training, the performance is impressively improved compared to other comparison methods. This is a strong indicator that the variation consistency mechanism is effective. Second, in the ablation study as shown in Table V, compared to the base network without variation consistency mechanism, our AVN model with this mechanism impressively boosts the performance on all the testing datasets. This directly indicates that the introduced variation consistency mechanism is effective for improving performance of signature verification.

The major reasons why our model is effective lie in three aspects. First, our model adopts the neural network modules to extract basic image features. Second, our model introduces the variation consistency mechanism and the corresponding variator design. Third, our model introduces the adversarial enhancement training method. The aspects jointly improve the signature verification performance.

**2) How robust is our AVN model to image intensity variation?**

Table VII and Table VIII present the comparisons of the proposed model with the baseline model and two data augmentation methods when the signature image intensities or colors are varied. From these two tables, we can observe that as the variation increases, the performances of our method, data augmentation methods, and the baseline method are all reduced. However, the performance reduction speed of our method is much slower than the baseline method, especially when the variation is not large. Although the reduction speed of data augmentation methods are also slower than the baseline method, the performances of data augmentation methods are lower than our methods almost in all variation ranges. Second, the random variation harms the methods more than the uniform variation. However, our method behaves much better than the other comparison methods. For example, on the BHSig-Hindi dataset with random variation, when the variation range becomes from 1 to the values smaller than 0.7, the accuracy of the baseline method decreases from 90.74% to 50.14%, while the accuracy of our method becomes from 94.32% to 94.00%. This is a very decent result considering that the random variations with the ranges smaller than 0.6 may have destroyed the patterns of the original signature images. Fig. 8 shows the average error rate (AER) curves of these methods with the uniform and random variations. These figures show that for uniform variation or random variation our AVN method has much lower average error rates than the other methods. Furthermore, the curves of the comparison methods are steeper than the curves of our method, which indicates that the variation has a smaller influence over our method than the comparison methods.

These results show that our method is more robust to image intensity variation than the comparison methods. The major reason is that our method introduces the variation consistency mechanism and the corresponding adversarial enhancement learning method. With this mechanism, our model actively varies the original input images to challenge the model, which not only improves the model's performance in signature verification, but also enhances the model's ability to resist variation.

**3) How does variation consistency differ from data augmentation?**

Data augmentation [73] is a widely-used technique for augmenting data in training. In our AVN model, the variation consistency mechanism produces new variants of the original input signature images for training. In this sense, our variation consistency mechanism has conceptual affinities with data augmentation.

However, our variation consistency mechanism is different from the existing data augmentation [73]–[76], for the following reasons. First, the goal of existing data augmentation methods is to solve the overfitting problem or enhance the model's ability to resist some transformations, while the major goal of our variation consistency mechanism is to push the model to mine effective signature stroke information. Second, the new data generation of our variation consistency mechanism is an online process in training while previous data augmentation techniques usually generate new data offline. Such online processes are more flexible. Third, the generation of new data in our variation consistency mechanism is an "active" process, i.e. it learns to generate the desired variants of the original data which is conducive to enhancing the model's discriminative ability. Existing data augmentation usually produces additional training data by some manually defined rules, such as rotating and flipping, which is stochastic and nondirectional. Finally, the variation consistency mechanism is linked to an adversarial enhancement training method, by which the model's discriminative ability is enhanced. Existing data augmentation is not necessarily related to such a training technique.

We also use an ablation study to analyse the advantage of our variation consistency mechanism compared to the traditional data augmentation. Table VI shows the results of comparison. Fig. 6 shows the ROC curves of our method and the two data augmentation methods. The results demonstrate the advantage of the proposed method over traditional data augmentation in signature verification.

**4) What is the potential of our AVN method?**

Our model incorporates variation consistency mechanism into an ordinary base network model for signature verification. On some datasets, compared with some sophisticatedly-designed neural network methods, the results of our method are not state-of-art. The major reason is that the base network used in our model is a rather simple and ordinary structure, which

is composed of only two convolutional blocks for feature extraction and two fully-connected layers for decision. Under this circumstance, our AVN model still achieves better performance than most of other approaches. In this sense, our method would have great potential by using more sophisticated base modules, such as Residual Network [30].

In this work, variation consistency mechanism is used to mine effective stroke features for signature verification. The essence of variation consistency mechanism is actively generating variants of original data to train a model with desired attributes. Thus we believe it can be generalized to other tasks and play promising roles in those tasks. The potential applications include the following aspects. First, it can be applied to some multimedia applications such as joint parsing of texts and images. One of the challenging problems in text modeling is the variation of descriptions with the same semantic meaning. The variation consistency mechanism is a possible means for this challenging problem. Second, it can be generalized to some other biometric techniques in security, such as recognition of fingerprint, iris, and face. The frameworks of these applications are similar to signature verification and therefore the variation consistency mechanism can be easily generalized. Third, it can be applied to neural network models for object and scene understanding in computer vision. In object recognition, a well trained neural network model is often prone to be harmed by small disturbance. If the variation consistency mechanism was incorporated into object recognition, the disturbance issue might be alleviated and therefore the object recognition performance would be improved.

## VI. Conclusion

In this paper, we present a novel adversarial variation network for writer-independent signature verification which contains three functional modules: the extractor seeks to extract deep discriminative features of handwritten signatures, the discriminator aims to make verification decisions based on the extracted features, and the variator is designed to actively generate signature variants for constructing a more discriminative model. These modules are unified under a variation consistency framework and learned with an adversarial enhancement learning in an end-to-end way. The proposed model is tested on four challenging datasets of different languages.

Experimental results show that our method improve signature verification performance. Furthermore, the introduced variation consistency mechanism outperforms the traditional data augmentation and the proposed method is more robust to image variation than the comparison methods. We also analyse the possible applications of our method in other fields such as multimedia, biometrics, and computer vision, which suggests the great potential of our method.

The future work will focus on the generative models of signature verification and extending our method to other multimedia, biometrics, and vision problems.

## References


[1] A. El-Yacoubi, E. J. R. Justino, R. Sabourin, and F. Bortolozzi, "Off-line signature verification using HMMs and cross-validation," in *Proc. IEEE Signal Process. Soc. Workshop*, 2000, pp. 859–868.

[2] S. Pal, A. Alaei, U. Pal, and M. Blumenstein, "Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset," in *Proc. Int. Assoc. Pattern Recognit. Workshop DAS*, 2016, pp. 72–77.

[3] S. Dey *et al.*, "SigNet: Convolutional siamese network for writer independent offline signature verification," 2017, *arXiv:1707.02131*.

[4] M. Yuan and Y. Peng, "CKD: Cross-task knowledge distillation for text-to-image synthesis," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1955–1968, Aug. 2020.

[5] P. Wei, H. Li, and P. Hu, "Inverse discriminative networks for handwritten signature verification," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5757–5765.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[7] M. K. Kalera, S. Srihari, and A. Xu, "Off-line signature verification and identification using distance statistics," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 7, pp. 1339–1360, 2004.

[8] M. A. Ferrer, M. Diaz-Cabrera, and A. Morales, "Static signature synthesis: A. neuromotor inspired approach for biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 667–680, Jul. 2014..

[9] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014.

[10] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *Proc. Neural Inf. Process. Syst. Deep Learn. Workshop*, 2014.

[11] Z. Zhang, X. Liu, and Y. Cui, "Multi-phase offline signature verification system using deep convolutional generative adversarial networks," in *Proc. Int. Symp. Comput. Intell. Des.*, 2017, pp. 103–107.

[12] J. Ruiz-Del-Solar, C. Devia, P. Loncomilla, and F. Concha, "Offline signature verification using local interest points and descriptors," in *Iberoamerican Congr. Pattern Recognit.: Prog. Pattern Recogn., Image Anal. Appl.*, 2008, pp. 22–29.

[13] M. Yılmaz and K. Öztürk, "Recurrent binary patterns and cnns for offline signature verification," in *Proc. Future Technol. Conf.*, 2020, pp. 417–434.

[14] V. Ruiz, I. Linares, A. Sanchez, and J. F. Velez, "Off-line handwritten signature verification using compositional synthetic generation of signatures and siamese neural networks," *Neurocomputing*, vol. 374, pp. 30–41, Jan. 2020.

[15] E. N. Zois, A. Alexandridis, and G. Economou, "Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets," *Expert Syst. Appl.*, vol. 125, pp. 14–32, Jul. 2019.

[16] E. N. Zois, E. Zervas, D. Tsourounis, and G. Economou, "Sequential motif profiles and topological plots for offline signature verification," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13245–13255.

[17] E. N. Zois, M. Papagiannopoulou, D. Tsourounis, and G. Economou, "Hierarchical dictionary learning and sparse coding for static signature verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 545–54510

[18] H. Baltzakis and N. Papamarkos, "A new signature verification technique based on a two-stage neural network classifier," *Eng. Appl. Artif. Intell.*, vol. 14, no. 1, pp. 95–103, Feb. 2001.

[19] M. B. Yılmaz and B. Yanıkoğlu, "Score level fusion of classifiers in off-line signature verification," *Inf. Fusion*, vol. 32, pp. 109–119, Nov. 2016.

[20] Y. Serdouk, H. Nemmour, and Y. Chibani, "Combination of OC-LBP and longest run features for off-line signature verification," in *Proc. 10th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, 2014, pp. 84–88.

[21] A. Alaei, S. Pal, U. Pal, and M. Blumenstein, "An efficient signature verification method based on an interval symbolic representation and a fuzzy similarity measure," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2360–2372, Oct. 2017.

[22] Y. Serdouk, H. Nemmour, and Y. Chibani, "Orthogonal combination and rotation invariant of local binary patterns for off-line handwritten signature verification," in *Proc. Int. Conf. Telecommun.*, 2015, p. 75.

[23] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordonez, "Robustness of offline signature verification based on gray level features," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 3, pp. 966–977, Jun. 2012.

[24] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Writer-Independent feature learning for offline signature verification using deep convolutional neural networks," in *Int. Joint Conf. Neural Netw.*, 2016, pp. 2576–2583.

[25] N. N. Mohammad Rezaei, "Persian signature verification using fully convolutional networks," in *Proc. Nat. Conf. New Res. Elect. Comput. Eng.*, 2017.

[26] M. Berkay, Yilmaz, and K. Öztürk, "Hybrid user-independent and user-dependent offline signature verification with a two-channel CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 639–6398.

[27] L. G. Hafemann, R. Sabourin, and L. Soares de Oliveira, "Meta-learning for fast classifier adaptation to new users of signature verification systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1735–1745, 2020.

[28] A. Hamadene and Y. Chibani, "One-class writer-independent offline signature verification using feature dissimilarity thresholding," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1226–1238, Jun. 2016.

[29] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2014, *arXiv:1312.6199*.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[31] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015, *arXiv:1506.01497*.

[32] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, "Spatio-temporal attention networks for action recognition and detection," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2990–3001, Nov. 2020.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[35] Y. Liu, P. Wei, and S. Zhu, "Jointly recognizing object fluents and tasks in egocentric videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2943–2951.

[36] Y. Yi, H. Wang, and Q. Li, "Affective video content analysis with adaptive fusion recurrent network," *IEEE Trans. Multimedia*, vol. 22, no. 9, pp. 2454–2466, Sep. 2020.

[37] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 1993, pp. 737–744.

[38] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4353–4361.

[39] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognit.*, vol. 70, pp. 163–176, Oct. 2017.

[40] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, With Application to Face Verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 539–546.

[41] L. Liu, C. Xiong, H. Zhang, Z. Niu, M. Wang, and S. Yan, "Deep aging face verification with large gaps," *IEEE Trans. Multimedia*, vol. 18, no. 1, pp. 64–75, Jan. 2016.

[42] D. Engin, A. Kantarci, S. Arslan, and H. K. Ekenel, "Offline signature verification on real-world documents," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 3518–3526.

[43] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. Int. Conf. Learn. Representations*, 2016.

[44] L. Yuan, E. H. F. Tay, P. Li, and J. Feng, "Unsupervised video summarization with cycle-consistent adversarial lstm networks," *IEEE Trans. Multimedia*, vol. 22, no. 10, pp. 2711–2722, 2019.

[45] Y. Tang and X. Wu, "Salient object detection using cascaded convolutional neural networks and adversarial learning," *IEEE Trans. Multimedia*, vol. 21, no. 9, pp. 2237–2247, Sep. 2019.

[46] Y. Luo, P. Liu, G. Tao, J.-q. Yu, and Y. Yang, "Adversarial style mining for one-shot unsupervised domain adaptation," in *Proc. Neural Inf. Process. Syst.*, 2020.

[47] X. Peng, Z. Tang, F. Yang, R. S. Feris, and D. Metaxas, "Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation," in *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2226–2234.

[48] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[49] L. C. Tiao, E. V. Bonilla, and F. T. Ramos, "Cycle-consistent adversarial learning as approximate bayesian inference," in *Proc. Int. Conf. Mach. Learn. Workshop*, 2018.

[50] D. Kuang, "Cycle-consistent training for reducing negative jacobian determinant in deep registration networks," in *Proc. Simul. Synth. Med. Imag.*, 2019.

[51] F. Zhao *et al.*, "Harmonization of infant cortical thickness using surface-to-surface cycle-consistent adversarial networks," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2019, pp 475–483.

[52] Y. Liu, Y. Guo, L. Liu, E. M. Bakker, and M. S. Lew, "Cyclematch: A cycle-consistent embedding network for image-text matching," *Pattern Recognit.* vol. 93, pp. 365–379, Sep. 2019.

[53] J. Hoffman *et al.*, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018.

[54] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.

[55] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017.

[56] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Characterizing and evaluating adversarial examples for offline handwritten signature verification," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 8, pp. 2153–2166, Aug. 2019.

[57] A. Dutta, U. Pal, and J. Lladós, "Compact correlated features for writer independent signature verification," in *Proc. Int. Conf. Pattern Recognit.*, 2016, pp. 3422–3427.

[58] S. Chen and S. Srihari, "A new off-line signature verification method based on graph," in *Proc. Int. Conf. Pattern Recognit.*, 2006, pp. 869–872

[59] N. Ohtsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[60] R. Kumar, L. Kundu, B. Chanda, and J. D. Sharma, "A writer-independent off-line signature verification system based on signature morphology," in *Proc. Int. Conf. Intell. Interactive Technol. Multimedia*, 2010, pp. 261–265

[61] R. Kumar, J. D. Sharma, and B. Chanda, "Writer-independent off-line signature verification using surroundedness feature," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 301–308, Feb. 2012.

[62] R. K. Bharathi and B. H. Shekar, "Off-line signature verification based on chain code histogram and support vector machine," in *Proc. Int. Conf. Adv. Comput., Commun. Informat.*, 2013, pp. 2063–2068.

[63] Y. Guerbai, Y. Chibani, and B. Hadjadji, "The effective use of the one-class SVM classifier for handwritten signature verification based on writer independent parameters," *Patt. Recognit.*, vol. 48, 2015, pp. 103–113.

[64] S. Lai and L. Jin, "Learning discriminative feature hierarchies for off-line signature verification," in *Proc. Int. Conf. Front. Handwriting Recognit.*, 2018, pp. 175–180.

[65] P. Maergner *et al.*, "Combining graph edit distance and triplet networks for offline signature verification," *Pattern Recognit. Lett.*, vol. 125, pp. 527–533, Jul. 2019.

[66] D. Gumusbas and T. Yildirim, "Offline signature identification and verification using capsule network," in *Proc. IEEE Int. Symp. Innov. Intell. Syst. Appl.*, 2019, pp. 1–5.

[67] S. D. Das, H. Ladia, V. Kumar, S. Mishra, "Writer independent offline signature recognition using ensemble learning," in *Proc. Int. Conf. Data Sci., Mach. Learn. Appl.*, 2019.

[68] S. Shariatmadari, S. Emadi, and Y. Akbari, "Patch-based offline signature verification using one-class hierarchical deep learning," *Int. J. Document Anal. Recognit.*, vol. 22, pp. 375–385, 2019.

[69] A. Bhunia, A. Alaei, and P. Roy, "Signature verification approach using fusion of hybrid texture features," *Neural Comput. Appl.*, vol. 31, pp. 8737–8748, 2019.

[70] A. Alaei, S. Pal, U. Pal, and M. Blumenstein, "An efficient signature verification method based on an interval symbolic representation and a fuzzy similarity measure," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2360–2372, Oct. 2017.

[71] C. Li, F. Lin, Z. Wang, G. Yu, L. Yuan, and H. Wang, "Deephsv: User-independent offline signature verification using two-channel CNN," in *Proc. Int. Conf. Document Anal. Recognit.*, 2019, pp. 166–171.

[72] Y. Serdouk, H. Nemmour, and Y. Chibani, "Handwritten signature verification using the quad-tree histogram of templates and a support vector-based artificial immune classification," *Image Vis. Comput.*, vol. 66, pp. 26–35, Oct. 2017.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012.

[74] A. V. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, and A. A. Kalinin, "Albumentations: Fast and Flexible Image Augmentations," *Information*, vol. 11, no. 2, 2020.

[75] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Mach. Learn.*, 2018.

[76] A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, and A. V. Dalca, "Data augmentation using learned transforms for one-shot medical image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8535–8545.

**Huan Li** (Student Member, IEEE) received the B.E. degree in communication engineering from the Xi'an University of Science and Technology, Xi'an, China. He is currently working toward the doctor's degree with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include the statistical learning and deep learning methods for biometrics and object identification.

**Ping Wei** (Member, IEEE) received the B.E. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China. He is currently an Associate Professor with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. He has been a Postdoctoral Researcher with Center for Vision, Cognition, Learning, and Autonomy (VCLA), University of California, Los Angeles (UCLA). His research interests include computer vision, machine learning, and computational cognition. He is a Co-Organizer of the International Workshop on Vision Meets Cognition: Functionality, Physics, Intents and Causality at CVPR 2017 and 2018, respectively.

**Ping Hu** received the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China. She is currently a Professor with the School of Management, Xi'an Jiaotong University. She has been a Visiting Scholar with Alberta University, Canada. Her research interests include financial system security, data mining from knowledge dissemination, and social network of enterprises and entrepreneurs. She has authored or coauthored more than one hundred academic papers, books, and reports in related areas.