2016 9th International Symposium on Computational Intelligence and Design

Multi-Phase Offline Signature Verification System Using Deep Convolutional Generative Adversarial Networks

Zehua Zhang[1], Xiangqian Liu[1], Yan Cui[2],
1. Institute of Information Science, Beijing Jiaotong University, Beijing, China
2. Beijing University of Chemical Technology, Beijing, China
15120383@bjtu.edu.cn, xqliu@bjtu.edu.cn, cuiyan20@126.com

***Abstract*—We present an offline signature verification system with new architecture, using Deep Convolutional Generative Adversarial Networks (DCGANs) to learn features unsupervised instead of using hand-crafted features. The advantage of this architecture is that system has a robust generalization ability. Besides, aiming at the conflict between convenience and accuracy, a hybrid Writer-Independent-Writer-Dependent classifier is used, which is an approach compromise two different kind of classifier. We test our method in GPDSsyntheticSignature database, which is an updated version of GPDS-960. In order to make it's convenient to compare with other works, we also use GPDS-960 to test our method. Our experimental results show that, the accuracy of proposed method is satisfactory. Due to this kind of architecture, the more query samples are tested and enrolled into the system, the more accurate our system will be conceptually.**

Keywords*— *offline signature verification; DCGANs; Writer-Independent-Writer-Dependent classifier

I. INTRODUCTION

Biometrics technology is used in a wide variety of security applications. Handwritten signature is playing a very special role in the wide series of biological characteristics. Up to this day, it's still a hot topic of computer vision, mostly due to its widespread use to verify a person's identity in legal, such as financial and administrative areas. Generally speaking, we can distinguish between two different categories of verification system: online and offline. In the first case, signature signal is captures during the writing process, thus we can get some dynamic information of signature, such as the stroke order, the pressure applied and the speed. However, in offline system, the signature what we get is a static image.

In the research of signature verification, forgeries are often classified in three types: random, simple and skilled forgeries. In the case of random forgeries, the forger has no information about the user's signature or even don't know his name, forgeries present a very different overall shape. In the case of simple forgeries, the forger has knowledge of the user's name, but not about the user's signature. In this case, the forgery may present more similarities to the genuine signature. In skilled forgeries, the forger has access for both the user's name and signature, and often practices imitating the user's signature [1]. Handwritten signature verification is still plagued by people mostly due to signatures are showing high intra-class variability and low inter-class variability. Compared to physical biometric traits, such as fingerprint or iris, handwritten signatures from the same user often show a large variability between samples, even much discriminative from same user write in different period. This problem is illustrated in **Figure 1**. In the **Figure 2**, we present low inter-class variability.

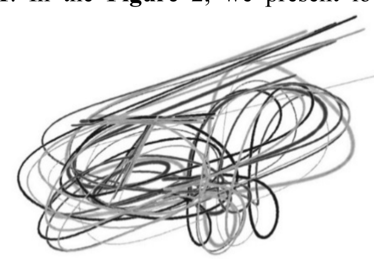

Figure 1. Ten genuine signatures from one user, we can notice a high intra-class variability.

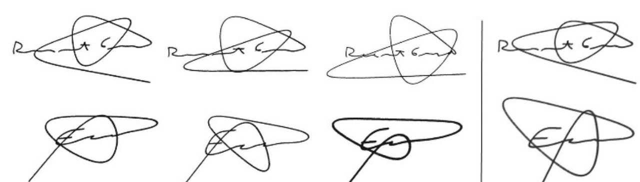

Figure 2. Each row contains 3 genuine signatures from the same user and a skilled forgery. It shows low inter-class variability.

Our system is addressed as a Pattern Recognition problem, as normal Pattern Recognition problem like, following steps are taken: pre-preprocessing, feature extraction, and training a classifier. Feature extraction and pattern recognition undoubtedly constitute essential components of a signature verification system. Most of the research effort in this area has been devoted to obtaining a good feature representation for signatures [4]. As with several problems in computer vision, it is often hard to design good feature extractors, and the choice of which feature descriptors to use is problem-dependent. As for offline signature verification system, although many approaches have been explored, such as Pseudo-dynamic features [14], geometric features [16,17], mathematical transformations features [18,19,20], and texture features [20] also used in verification system. There is not a widely accepted "best" feature to model the problem. Thus some authors used Artificial Neural Network (ANN) to learn

2473-3547/16 $31.00 © 2016 IEEE
DOI 10.1109/ISCID.2016.138

features. However, as the common drawback of ANN, layers in the networks are limited.

Standard signature verification systems are Writer-Dependent (WD) classification, which is trained for per-user using his genuine signatures [4,24], some authors name this architecture as specific classifier [13]. As for Writer-Independent (WI) classification, which involves only one single classifier for all users, it can seem like a universal classifier. Both of two architectures have coin's two sides. Although the accuracy of WD classifier is better than WI classifier, however, it's inconvenient to ask a user to provide enough number of signature samples to design personal WD classifier. WI classifier is a user-convenient approach, it can be trained by a development database that contain real user that enrolled the system. But on the other hand, it reduces the accuracy. Hence, the classifier implies a trade-off between accuracy and user convenience.

In order to overcome the difficulties and inadequacies of previous work. We propose a multi-phase architecture for offline signature verification system, which extracts feature use an unsupervised way, and works together with a hybrid Writer-Independent-Writer-Dependent classifier. The next section will give the details of our method, the experimental results and comparison will be presented in section Ⅲ. The problem remaining and future works will be discussed in section Ⅳ.

II. PROPOSED METHOD

In this paper, we propose a multi-phase approach, which contains an autonomic feature learning phase, following by a hybrid Writer-Dependent-Writer-Independent double phases. The architecture of our system is illustrated in **Figure 3**.

In feature learning phase, we base our research on recent successful applications of purely un-supervised learning models for computer vision, we present an offline signature verification system using Deep Convolutional Generative Adversarial Networks (DCGANs) [8] that have certain architectural constraints, and demonstrate that it is a strong candidate for unsupervised learning. Own to this method, unlabeled samples can be used and features can be extracted by itself instead of hand-crafted.

During classification phase, how to balance between this two sides become our concerns. A WI-WD system makes it possible to balance between two approaches, which compromises between the pros and cons. Due to this architecture, we find a balance point between convenience and accuracy. Thus we will use a hybrid Writer-Independent-Writer-Dependent classifier in our system.

As **Figure 3** shows, firstly, we train the Deep Convolutional Generative Adversarial Network (DCGANs) for learning feature representations. After extracting features, a WI classifier will be employed in the system first. Once enough samples are collected for a specific user, WD classifier is used to verify signatures for the specific user. The details will be presented in the following contents.

We tested our architecture using two datasets: GPDS-960 and GPDSsyntheticSignature. Actually latter database has replaced previous database now, the reason why we use GPDS-960 is easily compare with previous works, and test system's ability of generalization.

A. Feature Extraction

Learning feature representations from large number of labeled data is broadly used. However, recently unsupervised feature learning has been an active area. In this work, a class of unsupervised CNNs called deep convolutional generative adversarial networks is employed. Radford et al. [8] proposed one way to extract good image representations using Generative Adversarial Networks (GANs) which proposed by Goodfellow et al. in [25]. We use parts of the generator and discriminator networks as feature extractors for our supervised tasks.

An example model architecture that proposed in [8] is showing in **Figure 4**. Stable Deep Convolutional GANs following such rules: a) Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator). b) Use batchnorm in both the generator and the discriminator. c) Remove fully connected hidden layers for deeper architectures. d) Use ReLU activation in generator for all layers except for the output, which uses Tanh. e) Use LeakyReLU activation in the discriminator for all layers. For the details, you can refer to [8].

The main idea of GANs is training two networks at the same time, one of them we call it as generator $G(z)$, another one named as discriminator $D(y)$. When put a signal (such as an image) into $D(y)$, it will output a scalar to indicate that image y seems to be the "natural" or not. $D(y)$ can be seen as a kind of energy function, when y is an actual sample (in our system, it's genuine signature) this function will use a very low value (for example, close to zero). When y is not a real sample (forgery), it will adopt a positive. As for $G(z)$, z is a vector generally, the function of generator is generate an image to train $D(y)$ in order to form the correct model.

In the case of training, we do our best to make the output D minimize via tune the parameters. However, $G(z)$ will self-training to generate images, in order to deceive D, try to make D believe that generation is the real image. In other words, network is trying to minimize the output of the D, however D is trying to maximize itself. So this kind of network is called adversarial networks.

B. WI-WD Classification

In this paper, a solution compromise between the pros and cons of WI and WD classification will be used. Which is proposed by Eskander et al. in [11]. In order to train classifier, a development database should be used. So we start by partitioning the dataset into two distinct sets: Development set D and Exploitation set E. Both of two are a sub-dataset from the enrolled users. In the classifier training phase, the set D is used to learn the feature representation for signatures, and set E contains the rest of users and work as a test database for evaluating the performance of the system. Firstly, we start to train a WI classifier with development set D, through operation, signature samples are collected and stored in the user profile. Once enough samples which are used to adapt the universal (WI) classifier to this user are collected for a specific user, following the WD classifier will be used to verify

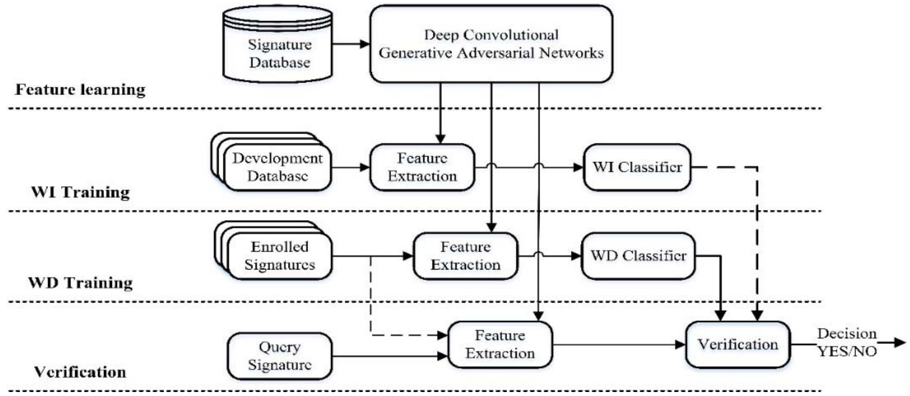

Figure 3. The architecture of Multi-Phase Offline Signature Verification System.

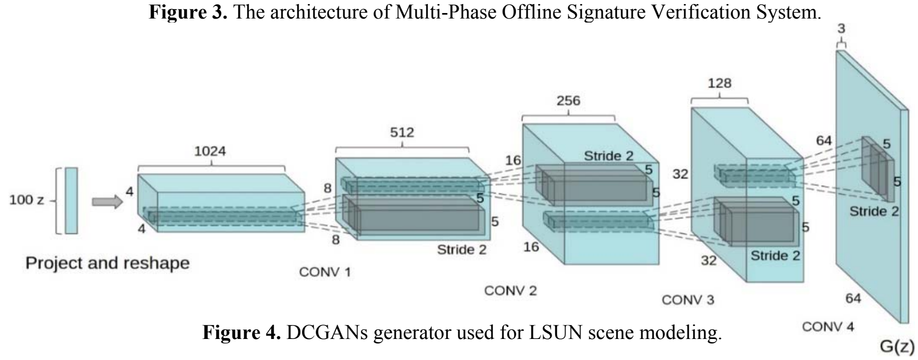

Figure 4. DCGANs generator used for LSUN scene modeling.

signatures for this specific user from this time on. While the universal classifier compares the query samples to the stored user signature templates, the user-specific (WD) classifier only uses the query sample to produce the classification decision. We illustrate this procedure in **Figure 3**, a verification system is designed combine WI classifier with WD classifier.

A Gentle Adaboost is employed in our system to separate different features extracted from signatures. Adaboost classifier is also used by some authors like Juan Hu et al. in [14], however, we are inspired by Friedman's work in [26], we use Gentle Adaboost instead of Real Adaboost. Each weak classifier of Real Adaboost output the probability of sample belongs to a certain class, then map the probability value of 0 and 1 to Real domain with a logarithmic function, the final output of classifier is the sum of entire mapping functions. However, Gentle AdaBoost will do a weighted regression additionally, based on the least squares at each iteration, the sum of all of the regression functions is used as the final output of classifier. For this reason, Gentle Adaboost is more accurate than Real Adaboost. Generate a Gentle Adaboost should follow below steps [26]:

a) Start with a weight $w_i = 1/N,\ i = 1, 2 \ldots N,\ F(x) = 0$.

b) Repeat for $m = 1, 2 \ldots M$:

(1) Fit the regression function $f_m(x)$ by weighted least-squares of y_i to x_i with weights w_i.

(2) Update $F(x) \leftarrow F(x) + f_m(x)$.

(3) Update $w_i \leftarrow w_i exp(-y_i f_m(x))$ and renormalize.

c) Output the classifier:

$$\text{sign}[F(x)] = \text{sign}[\sum_{m=1}^{M} f_m(x)]$$

III. Experiment and Results

A. Database

Two large publicly available databases are employed in our system. The first one is GPDS-960 [9], this dataset contains 24 genuine signatures and 30 skilled forgeries per user, from 881 users, which were captured in a single session. The second one is GPDSsyntheticSignature, it obtains data from 4000 synthetic individuals: 24 genuine signatures for each individual, plus 30 skilled forgeries of his/her signature. All the signatures were generated with different modeled pens. In our experiment, only GPDS-960 is used for train system, however, in the verification phase, both of two databases are tested. According to the performance of our system works on a database which is never trained on to evaluate the ability of generalization.

B. Features Learning

One common technique for evaluating the quality of unsupervised representation learning algorithms is to apply them as a feature extractor on supervised datasets and evaluate the performance of linear models fitted on top of these features. To evaluate the quality of the representations learned by DCGANs for supervised tasks, we train system on GPDS-960 and then use the discriminator's convolutional features from all layers, these features are then flattened and concatenated to form a vector and a hybrid WI-WD classifier is trained on the top of them.

In this paper, all weights are initialized from a zero-centered Normal distribution with standard deviation 0.02. In the LeakyReLU, the slope of the leak is set to 0.2 in all models. Learning rate we select 0.0005, the value of momentum term β1 use 0.5.

C. Classifier Training

We split GPDS-960 into two parts: The first part (S_I) contains signatures of the first 381 users, a subset of this part is used as development set for the WI training, and the remaining of this part is used as exploitation set for performance evaluation. Details showing in **Table 1**. The second part (S_D) contains signatures of the last 500 users, this part is used for the WD training and seeing in **Table 2**. To evaluate the impact of different number of sample signatures per user, we trained the WD classifiers using a variable number of signatures from the enrolled users. Training and testing procedure is taken 10 epoches for the purpose of obtaining reliable results. For testing, all of signatures we selected from the users, ensuring they were not used for training the current classifier.

Table 1. The part of WI dataset S_I: 381 × (24 genuine + 30 skilled forgeries)

	Intra-Class	Inter-Class
Development	14 genuine signatures from S_I	14 random signatures from S_I
	14×381=5334 samples	14×381=5334 samples
Exploitation	10 untrained genuine signatures from S_I	30 skilled forgeries and 10 random forgeries from S_I
	10×381=3810 samples	40×381=15240 samples

Table 2. The part of WD dataset S_D: 500 × (24 genuine + 30 skilled forgeries)

	Intra-Class	Inter-Class
Development	14 genuine signatures from S_D	14 training signatures from S_I
	14×500=7000 samples	14×381=5334 samples
Exploitation	genuine signatures subusets of S_D sizes	30 skilled forgeries and 10 random forgeries from S_D
	4, 8, 12, 14 samples	40×500=20000 samples

D. Results

In this paper, we train and test our system on GPDS-960, meanwhile test on GPDSsyntheticSignature database to test ability of generalization. Such performance index will be used to evaluate the accuracy of classifier in this paper: The AUC of ROC curves, the False Rejection Rate (FRR), False Acceptance Rate (FAR) and Average Error Rate (AER).

ROC curves is broadly used to evaluate performance of a binary classifier, and AUC (Area Under roc Curves) can perceivably reflect a system is good (value close to 1) or bad (value close to 0). FRR means the rate of a positive sample is rejected by classifier falsely, as for FAR, it means the rate of a negative sample is accepted by classifier falsely. AER is defined as:

$$AER = \frac{FRR + FAR_{random} + FAR_{simple} + FAR_{skilled}}{4}$$

Where FAR_{random}, FAR_{simple} and $FAR_{skilled}$ are the false accept rates when verifying random, simple and skilled forgeries respectively. However, for GPDS database, we can consider only FAR_{random} and $FAR_{skilled}$.

An overall review present in **Table 3**. Firstly, we compare our results with different authors. Different strategies of split database, features selection, type of classification with diff erent classifier, which lead to different accuracy. These results shows that Writer-Dependent approach performs better than Writer-Independent approach broadly. However this is not surprising because the Writer-Dependent classifier is trained for each users particularly, however the Writer-Independent classifier is designed as a universal model for all users.

Table 3. Overall review the results of employing different methods

	Database	Feature	Classifier	AER(%)	Mean AUC
Hafemann et al. [4]	GPDS-160	CNN_GPDS_{norm}	WD SVM_{RBF}	10.4	0.9459
	GPDS-300	CNN_GPDS_{norm}	WD SVM_{RBF}	14.84	0.9251
Guebai et al.	GPDS-160	Curvelet Transform	WD OC-SVM	15.95	—
Kumar et al. [22]	GPDS	Texture Imformation	WI SVM	13.76	—
Eskander et al. [11]	GPDS	ESC&DPDF	WI-WD BFS	13.96~17.82	—
Proposed Method	GPDS	DCGANs	WI-WD Gentle Adaboost	12.57~16.08	0.9172
	GPDSsynthetic Signature	DCGANs	WI-WD Gentle Adaboost	14.79~17.38	0.8836

Kumar et al. proposed a WI system trained with forgery and hand-crafted feature in [24], this system achieved an acceptable result. **Table 4** presents detail results of WI system and WI-WD system. We can notice that, WI-WD system include our proposed system, FRR, FAR and AER is a dynamic range, which because the train size of WD classifier is different.

According to the **Table 4**, we can find that the FRR of our system ranges between 17.2% and 23.73%, the FAR of random and skilled forgery verification ranges 0.0017%~0.0037%, and 16.05% ~25.1% respectively, and the accuracy of our system ranges between 12.57% and 16.08%. Generally speaking, our system outperforms Eskander et al. [11] proposed system, and based on the point of balancing between pure WI and WD method.

Besides, we test our system on a never trained database: GPDSsyntheticSignature. From the **Table 4**, we can see the FRR ranges between 18.74% and 26.13%, the FAR of random and skilled forgery verification ranges 0.0037% ~ 0.0077%, and 17.35% ~ 26.74% respectively, and the accuracy of our system ranges between 14.79% and 17.38%. Although the accuracy on this database is the worst compared with other results. However, this result can be acceptable as might have been expected. In theory, a sufficiently large corpus of database could provide an unlimited training set and an unsupervised feature extractor trained on this massive corpus should outperform the much smaller supervised training set. In this paper, we train our network on the GPDS-960, even so,

Table 4. Detail results of WI svstem and WI-WD svstem

	Kumar et al. [24]	Eskander et al. [11]	Proposed Method	
Database	GPDS	GPDS	GPDS	GPDSsyntheticSignature
Feature	Texture Imformation	ESC&DPDF	DCGANs	DCGANs
Classifier	WI SVM	WI-WD BFS	WI-WD Gentle Adaboost	WI-WD Gentle Adaboost
FRR(%)	13.76	18.06~26.42	17.2~23.73	18.74~26.13
FAR(%)	13.76	random:0.0031~0.0056 skilled:18.17~27.04	random:0.0017~0.0037 skilled:16.05~25.1	random:0.0037~0.0077 skilled:17.35~26.74
AER(%)	13.76	13.96~17.82	12.57~16.08	14.79~17.38
Mean AUC	—	—	0.9172	0.8836

samples are not enough comparatively. It is worth mentioning that the more query samples are tested and enrolled into the system, the more accurate our system will be since the WI-WD scheme.

IV. Conclusion

In this paper, we present a system with multi-phase architecture for offline signature verification. System is composed by an unsupervised feature extraction phase, and a hybrid WI-WD classification double phases. We test our system on GPDS-960 and GPDS-syntheticSignature, the results demonstrate the rationality and the domain robustness of the system. Our method is promising, even though our system doesn't achieve performance close to the state-of-the-art for GPDS, because it combines conveniences and robustness.

Generative Adversarial Networks is an exciting event for DL. It is the most worth looking forward to, in the near future, and is the most likely a breakthrough in the field. A lot of short term applications have been presented, we believe this is a promising direction for future exploration. In the long run we believe that unsupervised feature learning of signature verification has the potential to outperform supervised training as the unsupervised approach has access to unlimited data. Our work is the first serious step in this direction.

Acknowledgment

This paper is partially supported by the research "High Speed Maglev Transportation Engineering Integrated System" (No. 2013BAG19B01) of National Key Technology Research and Development Program during the Twelfth Five-Year Guideline period of China

References

[1] Hafemann, Luiz G., Robert Sabourin et al. "Offline handwritten signature verification - literature review." *arXiv preprint arXiv: 1507.07909* (2015).

[2] Li, Yin, et al. "Unsupervised learning of edges." *arXiv preprint arXiv: 1511.04166* (2015).

[3] Gallo, Orazio, et al. "Retrieving Gray-Level Information from a Binary Sensor and its Application to Gesture Detection." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2015.

[4] Hafemann, Luiz G., Robert Sabourin, et al. "Writer-independent Feature Learning for Offline Signature Verification using Deep Convolutional Neural Networks." *arXiv preprint arXiv:1604.00974* (2016).

[5] Alam, Md Khorshed, and Aisha Hassan Abdalla. "An evaluation on offline signature verification using artificial neural network approach." *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*. IEEE, 2013.

[6] Quraishi, Md Iqbal, Arindam Das, and Saikat Roy. "A novel signature verification and authentication system using image transformations and Artificial Neural Network." *Computer and Information Technology (WCCIT), 2013 World Congress on*. IEEE, 2013.

[7] Walker, Jacob, Abhinav Gupta, and Martial Hebert. "Patch to the future: Unsupervised visual prediction." *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014.

[8] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

[9] Vargas, J. Francisco, et al. "Off-line handwritten signature GPDS-960 corpus." *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. 2007.

[10] Rivard, Dominique, Eric Granger et al. "Multi-feature extraction and selection in writer-independent off-line signature verification." *International Journal on Document Analysis and Recognition (IJDAR)* 16.1 (2013): 83-103.

[11] Eskander, George S. and Robert Sabourin et al. "Hybrid writer-independent–writer-dependent offline signature verification system." IET biometrics 2.4 (2013): 169-181.

[12] Waknaghat, P. O. "A Preliminary Study on Various Off-line Hand Written Signature Verification Approaches." (2010).

[13] Ribeiro, Bernardete and Ivo Gonçalves et al. "Deep learning networks for off-line handwritten signature recognition." Iberoamerican Congress on Pattern Recognition. Springer Berlin Heidelberg, 2011.

[14] Hu, Juan, and Youbin Chen. "Offline signature verification using real adaboost classifier combination of pseudo-dynamic features." 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013.

[15] Sharma, Meenakshi, and Kavita Khanna. "Offline Signature Verification Using Supervised and Unsupervised Neural Networks." (2014).

[16] Justino, Edson JR, et al. "An off-line signature verification system using HMM and graphometric features." Fourth IAPR International Workshop on Document Analysis Systems (DAS), Rio de. 2000.

[17] Baltzakis, H., and N. Papamarkos. "A new signature verification technique based on a two-stage neural network classifier." Engineering applications of Artificial intelligence 14.1 (2001): 95-103.

[18] Pourshahabi, Muhammad Reza, Mohamad Hoseyn Sigari, and Hamid Reza Pourreza. "Offline handwritten signature identification and verification using contourlet transform." Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of. IEEE, 2009.

[19] Deng, Peter Shaohua, et al. "Wavelet-based off-line handwritten signature verification." Computer vision and image understanding 76.3 (1999): 173-190.

[20] Serdouk, Yasmine, Hassiba Nemmour, and Youcef Chibani. "Combination of OC-LBP and Longest Run Features for Off-Line Signature Verification."Signal-Image Technology and Internet-Based Systems (SITIS), 2014 Tenth International Conference on. IEEE, 2014.

[21] Bertolini, Diego, et al. "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers." Pattern Recognition43.1 (2010): 387-396.

[22] Kumar, Rajesh, J. D. Sharma, and Bhabatosh Chanda. "Writer-independent off-line signature verification using surroundedness feature." Pattern recognition letters 33.3 (2012): 301-308.

[23] Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets." Neural computation 18.7 (2006): 1527-1554.

[24] Kumar, Rajesh, J. D. Sharma, and Bhabatosh Chanda. "Writer-independent off-line signature verification using surroundedness feature." Pattern recognition letters 33.3 (2012): 301-308.

[25] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in Neural Information Processing Systems. 2014.

[26] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. "Additive logistic regression: a statistical view of boosting." The annals of statistics 28.2 (2000): 337-407.