ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

OFFLINE SIGNATURE VERIFICATION USING MACHINE LEARNING

Graduation Project Final Report

Burak M. Gonultas
040140003

Department: Computer Engineering
Division: Computer Engineering

Advisor : Assoc. Prof. Dr. Hazim Kemal EKENEL

May 2019

ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

OFFLINE SIGNATURE VERIFICATION USING MACHINE LEARNING

Graduation Project Final Report

Burak M. Gonultas
040140003

Department: Computer Engineering
Division: Computer Engineering

Advisor : Assoc. Prof. Dr. Hazim Kemal EKENEL

May 2019

Statement of Authenticity

I hereby declare that in this study

1. all the content influenced from external references are cited clearly and in detail,
2. and all the remaining sections, especially the theoretical studies and implemented software/hardware that constitute the fundamental essence of this study is originated by my/our individual authenticity.

Istanbul, 19.05.2019

Burak Mert Gonultas

OFFLINE SIGNATURE VERIFICATION USING MACHINE LEARNING

(SUMMARY)

Autonomous signature verification has been quite a challenge for the researchers working in the signal processing domain, since hand signatures are unique specifiers created by humans and also verified by other humans by visual inspection. In the signal processing domain, signature verification problems are divided into two subcategories: online and offline signature verification. Online signature verification is the verification procedure of a hand signature generated by using pressure sensitive media, e.g. a pressure sensitive screen or a pressure sensitive pen. Since these devices are able to capture most of the dynamic features unobservable by human eyes which make forgeries even more difficult, online verification is a problem that has been solved with relatively high accuracy. On the other hand, offline signature verification is a more formidable task, since no dynamic information is available to the system; yet the system is expected to outperform expert visual analysis.

Recent developments in the machine learning domain focusing on sense data, i.e. in the sub-domains of computer vision, natural language processing and speech processing have significantly improved the state-of-the-art accuracy metrics in those areas and provided many possible solutions to problems like offline signature verification. However, the problem differs from generic binary classification problems by nature, due to the lack of so called “skilled-forgery signatures”. Almost every time, binary (also multi-class) classification models are trained on numerous positive and negative examples so that they can learn to discriminate between those two. However, for offline signature verification, it would be extremely unrealistic to feed the network skilled forgeries of a specific authentic signature, since those forgeries are almost never available under real conditions. To work around those constraints, a model must be trained that generalizes to such conditions and ideally, the model must be able to infer what a negative example is, without being shown a negative example.

During the development and evaluation phases, various offline signature verification datasets were used, those were: GPDS-Synthetic 4000, GPDS, MCYT-75 and CEDAR. Unfortunately, due to new European Data Regulations, CEDAR and GPDS datasets were inaccessible, therefore during the research, only pre-extracted features from those two datasets were used, which limited the development focusing those two datasets. The development was carried out using state-of-the-art Python libraries. GPU supported training of the models also gave significant boost during development.

The results were compiled in two different sections, the first experiments were carried out using available datasets, GPDS-Synthetic 4000 and MCYT-75. The proposed method scored 6.95% Equal Error Rate for global thresholds and 2.52% for user thresholds Equal Error Rate on GPDS-Synthetic 4000; closely matching 6.57% global thresholds Equal Error Rate of the state-of-the-art solution and surpassing 6.13% user thresholds Equal Error Rate achieved by the same solution by a significant margin. However, the generalization ability of a feature extraction model trained on GPDS-Synthetic 4000 turned out to be much poorer than models trained on GPDS, achieving only 9% Equal Error Rate using a global threshold and 4% using user specific thresholds versus 2.87% Equal Error Rate of the state-of-the-art

solution. The second subset of experiments were carried out the pre-extracted features by other researchers, on the CEDAR and MCYT-75 dataset, the achieved Equal Error Rate, False Acceptance and False Rejection Ratios using the pre-extracted features by the state-of-the art solution. The error scores on MCYT-75 were comparable while surpassing all the other proposed solutions in the literature. However, the achieved Equal Error Rate calculated using user-specific thresholds on CEDAR dataset was 2.75% versus 4.63% proposed by the state-of-the-art solution, which is a significant improvement.

MAKİNE ÖĞRENMESİ KULLARAK ÇEVRİMDIŞI İMZA DOĞRULAMA

(ÖZET)

Otonom imza doğrulama, sinyal işleme alanında çalışan araştırmacılar için zorlayıcı bir problem olarak varlığını sürdürmektedir, zira elle atılan imzalar, atan şahsı temsil eden, insan tarafından yaratılmış ve imza sahibine özgüdürler. Özgünlük doğrulamaları ise görsel inceleme sonucu yine insanlar tarafından yapılır. Sinyal işleme alanında imza doğrulama problemleri çevrimiçi ve çevrimdışı olmak üzere ikiye ayrılır. Çevrimiçi imza doğrulama prosedürü basınca duyarlı ortamlarla, örneğin basınca duyarlı tablet veya kalem gibi yaratılan imzaların doğrulanmasıdır. Bu cihazlar insan gözünün ayırt edemeyeceği dinamik detayları yakalayabildiklerinden bu cihazlarda atılan imzaları taklit etmek daha zordur. Bu sebepten çevrimiçi imza doğrulama problem günümüzde yüksek başarıyla çözülmüş bir problemdir. Öte yandan çevrimdışı imza doğrulama daha karmaşık bir problemdir zira sistemin dinamik detaylar hakkında bir fikri yoktur. Buna rağmen sistemin usta bir göz tarafından yapılan görsel analizden daha başarılı olması beklenir.

Makine öğrenmesinin duyusal veri üzerindeki yakın tarihli gelişmeleri sayesinde yapay görme, doğal dil işleme ve konuşma işleme alanlarındaki başarım ölçütlerinde kayda değer iyileşmeler olmuştur ve bu alanlardaki çevrimdışı imza doğrulamaya benzer problemlere birçok çözüm önerilmiştir. Yine de, söz edilen problem yaygın ikili sınıflandırma problemlerinden doğası gereği farklıdır zira "usta taklit imza" olarak adlandırabileceğimiz imza örnekleri sayıca azdır. Neredeyse her zaman ikili sınıflandırıcı modeller pozitif ve negatif örnekler üzerinde eğitilerek modellerin bunları ayırt etmesi beklenir. Ancak çevrimdışı imza doğrulama problem için taklit imzaların sisteme eğitim sırasında verilmesi gerçekçi değildir zira bu imzalar gerçek koşullarda neredeyse hiçbir zaman kulanılabilir değildir. Bu kısıtlamaları aşmak adına genelleme kabiliyeti yüksek ve negatif örnekleri görmeden dahi bunlara dair bir çıkarım yapabilen bir model eğitilmelidir.

Geliştirme ve değerlendirme fazlarında çeşitli çevrimdışı imza doğrulama veri kümeleri kullanılmıştır, bunlar: GPDS-Sentetik 4000, GPDS, MCYT-75 ve CEDAR veri kümeleridir. Ne yazık ki AB Kişisel Verileri Koruma Kanunları uyarınca CEDAR ve GPDS veri kümeleri erişime kapatılmıştır, bu sebepten ötürü, bu veri kümeleri üzerinde yapılan araştırma esnasında daha önceki araştırmacılar tarafından çıkartılmış öznitelik verileri kullanılmıştır ve bu iki veri kümesi için yapılan geliştirmeler bu sebepten ötürü sınırlı kalmıştır. Geliştirmeler modern Python dili kütüphaneleri kullanılarak yapılmıştır. Grafik işlemcisi kullanılarak yapılan model eğitimleri sayesinde geliştirmeler önemli ölçüde hızlandırılmıştır.

Sonuçlar iki ayrı grupta derlenmiştir, birinci gruptaki deneyler mevcutta kullanılabilir veri kümeleri olan GPDS-Sentetik 4000 ve MCYT-75 üzerinde gerçekleştirilmiştir. Bu projede sunulan metod evrensel sınır değerinin kullanıldığı şartlarda %6.95 isabet oranı, kullancıya has sınır değerleri kullanıldığında ise %2.52 eşit hata oranı yakalamıştır, Bu sonuçlar akademideki en iyi sonuç olan %6.57 evrensel sınır değeri eşit hata oranına yakın bir değerdir, öte yandan %6.13 olan kullanıcı özelinde belirlenmiş sınır değerindeki eşit hata oranından hissedilir derecede üstündür. Fakat GPDS-Sentetik 4000 üzerinde eğitilmiş bir öznitelik çıkartıcı modelin genelleme kabiliyeti zayıftır. Bu modelle çıkartılan özniteliklerle yapılan eğitim sonrası testlerde evrensel sınır değeri ile %9 eşit hata oranı, kullanıcı özelinde

tanımlanmış sınır değerleriyle ise %4 eşit hata oranı yakalanmıştır ve bu oran akademideki en iyi çözüm olan %2.87 değerinden önemli miktarda yüksektir. İkinci gruptaki deneyler ise diğer araştırmacılar tarafından CEDAR ve MCYT-75 veri kümelerinden çıkarılmış öznitelik verileri üzerinde gerçekleştirilmiştir. MCYT veri kümesi üzerinde yapılan geliştirmeler için eşit hata oranı, yanlış kabul ve yanlış ret oranları literatürdeki en başarılı örnek ile karşılaştırılabilir düzeydedir ve diğer çalışmalara göre fark edilir ölçüde üstündür. CEDAR veri kümesi üzerinde yapılan geliştirmelerde ise, şahıs eşik değerlerine bakılarak yapılan eşit hata oranı hesaplamasında literatürdeki en üstün sonuç %4.63 hata oranı verirken geliştirme sonucu %2.75 hata oranına erişilmiş ve önemli miktarda iyileşme sağlanmıştır.

Contents

1. Introduction and Project Summary

1.1. Problem Definition

Autonomous handwritten signature verification problem still exists as a formidable research problem for the signal processing research community, where the goal is to design a system that is able to discriminate between a genuine signature (i.e. generated by the claimed, unique person) and a forgery (i.e. generated by a "skilled" impostor). This issue proved to be a challenging one for the offline scenario, where only scanned signature images are available to the system and dynamic information about the signing procedure is inaccessible (or non-existent) to the system. Defining discriminative features and extracting them for signatures (just like for most of the computer vision problems) is a difficult task, hence, representing a signature using a feature set is a difficult goal. In the past decade, many proposals have been made in this subdomain and especially proposals using the power of Deep Neural Networks to automate the highly important feature extraction phases have made great contributions to this research area.

In the core of the problem lies the difficulty of finding a discriminative representation for signatures that directly correlates with the classification performance of a signature verification system. This is especially very important during the detection phase of "skilled" forgeries, which are made targeting a particular, unique person.

1.2. Proposed Solution

The final proposed solution is a correct implementation of the initial proposal made on the Interim Report, where Convolutional Neural Networks (CNN) have been used for extracting features from the handwritten signature images and Support-vector machine (SVM) classifiers using non-linear kernels have been used for the final classification using the extracted features from the handwritten signature subsets of interest. A very important point worth mentioning is the fact that the CNNs have been trained only using genuine signature subsets from the datasets, since in real conditions, access to skilled forgery of any handwritten signature is not a very easy task and accessing to a database of skilled forgeries of certain set of genuine forgeries is virtually impossible; unless they are generated for the specific problem. Even then, using generated skilled forgeries would be also quite ineffective for the system since they would also carry a certain pattern behind them during the forgery process, inapplicable for the real world forgery scenarios. The models trained using forgery signatures from the datasets also proved this theory by resulting in similar or worse performance compared to the models trained only using genuine forgeries. However, since CNNs are classifiers that require negative examples to learn a decision function, genuine signatures from other users are given as negative examples during the training for a certain user. Ultimately, the learned functions by the CNNs are used to extract feature vectors from signature images, essentially a group of floating-point numbers to be classified by Support-vector Machines. Here, it must be mentioned that the number of genuine examples (in the form of feature vectors per signature) available to the SVM classifier is a parameter which greatly effects the classification accuracy. The underlying reason for that is the existing visual cues for a personal signature and their closeness in the feature space. Since a biometric data like handwritten signature can contain variations, simply feeding more examples to the model makes it understand possible variations.

2. Comparative Literature Survey

The problem of signature verification relies heavily on the feature extraction for the binary and multi-class classification task both. First examples in the literature relied on the geometric features like basic descriptors such as the signature height, width, height to width ratio and covered area or more advanced descriptors proposed by H. Baltzakis and N. Papamarkos [1] like count of endpoints and closed loops. Some authors also generated local features by dividing the signatures in a grid and extracting cell features such as pixel density within grids like A. El-Yacoubi et al. [2]. During forensic document examination static features like Calibre ratio of height / width of the image, proportion, referring to the signature symmetry, alignment and spacing etc. are used, Oliviera et al. [3] investigated the problem using those features. Many of the researchers tried approaching the feature extraction problem using directional features, which define the direction of the pen strokes in the signature. Sabourin [4] and Drouhard [5] used the directional probability density function extracted from the gradient of the signature outline. This method was applied in combination with a grid-splitting of multiple scales by Rivard et al. [6]. A different approach using pyramid histogram of oriented gradients was proposed by Zhang et al. [7] which represents local characteristics of an image by a histogram of edge directions using multiple scales. An interested method called Extended Shadow Code was proposed by Sabourin et al. [4], [8], a grid is set on top of the signature image containing horizontal, vertical and diagonal bars with fixed number of bins. Pixels in the signature image are then projected to the nearest bar in each direction, adding up to the value of that bin. The total sum of each bin is the descriptor of the signature, in a similar fashion to a direction histogram. This type of extractor is also adopted by Rivard [9] and Eskander [10] with changing resolutions, ultimately achieving significant results for classification.

Recent developments in the Deep Learning domain empowered by Neural Network Architectures [11] have rendered hand-engineering of features obsolete in most of the sub-domains of the pattern recognition. The deep learning approach in literature has been implemented for the signature verification task in two different main paths. The first approach is to learn writer-independent features from a subset of users and then training writer-dependent classifiers [12], [13], [14]. An alternative approach is to learn feature representations and a complementary writer independent system at once, empowered by metric learning method [15]. Hafemann et al. [12] proposes a writer-independent approach. In this approach, a development set D is used to learn a feature representation function $\varphi(X)$ using CNNs. The trained network, representing the function $\varphi(X)$ is then used as a feature extractor on an exploitation set E (rather than hand-engineering of features) to train the writer-independent classifiers. The authors extended their work [14] and proposed a more extensive architecture, where CNNs are trained using both genuine signatures and skilled forgeries, trying to learn how to discriminate between users and also trying to learn how to discriminate between genuine signatures and skilled forgeries. Zhang et al. [16] used Generative Adversarial Networks for learning the features from a subset of users, in a similar fashion to the subset D proposed by Hafemann et al. [12]. However, the architecture is vastly different, two networks are trained, one for the generation task and the other for the discrimination task between a real signature and a machine-generated one. After training the networks, the convolutional layers of the discriminator network are proposed as feature extractors for new signatures.

The method proposed by Rantzsch et al. [15] uses metric learning, a method employed by Siamese networks. In this method, the system is trained to learn distances between signatures

and tuples are fed to the network during training, composed of a reference signature, a genuine signature and a forgery, either skilled or random. For best score, the system is expected to minimize the distance between the reference and the genuine and maximize the distance between the reference and the forgery.

The proposed solution is this project is a more generalized version of the Siamese networks approach proposed by Rantzsch et al. [15], where distance metrics are to be learned using two identical CNNs with identical weights. The shortcoming of this Siamese networks approach comes from the twin-networks architecture of the Siamese networks where signatures are compared in pairs. However, comparing signatures in batches, in a one vs. many fashion is a far better approach for this problem, due to high intra-class variance of personal genuine signatures. It also entirely possible to increase the CNN number in the Siamese networks architecture, however due to increasing model and loss function complexity, this is not a very preferable method. Using machine learning classifiers that are able to do one versus all classification perform better in terms of performance in that regard. Overall, the proposed approach in this project converges to the method proposed by Hafemann et al [12], using the term metric-learning to describe the method by a few changes.

3. Datasets

The development has been carried out using the dataset created by Ferrer et al. [17] [18] named as GPDSsyntheticSignature database, which in an offline signature database. It contains data from 4000 synthetic individuals, 24 genuine and 30 forgeries per individual. The pens used generating signatures were modeled differently for each signature. The generalization performance was assessed using the MCYT-75 dataset created by Ortega-Garcia et al. [19], consisting of 75 real individuals, having 20 genuine and 25 forgery hand signatures provided per individual.

The extracted features were provided by Hafemann et al. [13] for GPDS created by Blumenstein et al. [20], for MCYT-75 created by Ortega-Garcia et al. [19] and for CEDAR created by Kalera et al. [21] The genuine-forgery distribution per individual is the same for GPDS as GPDSsyntheticSignature (GPDS-Synthetic 4000). For CEDAR, 55 individuals were contained in the database, having 24 genuine and 24 forged signatures for each individual.

4. Data Preprocessing

The signatures provided in the datasets are already extracted from the corresponding documents, therefore signature detection is not included in the preprocessing. Still, due to fixed-size input characteristic of the Neural Network architectures, some preprocessing is necessary since the signature sizes are varying. An example signature is given below as Figure 1, taken from the GPDS-Synthetic 4000 dataset, first signature of user 49. A corresponding forgery is also given as Figure 2.

Figure 1: Example genuine signature

Figure 2: Example forged signature

The preprocessing starts with the size normalization, which is done by centering the images in a canvas that is as large as the largest height and width dimensions (separately), which is essentially the same process as padding an image to match a certain size. The maximum dimensions within GPDS-Synthetic 4000 are 1338x2973. The next step is doing an OTSU thresholding operation [22] proposed by Otsu et al., to clear the image of noise. During the thresholding, the background pixels are set to white with 255 intensity (8-bits) and the foreground pixels (that represent the signature itself) are left in grayscale as is. Then an inversion operation is carried out, setting the background pixels in black and the foreground pixels in inverse grayscale. An example signature before the preprocessing operation and the preprocessed version of the same signature is presented as Figures 3 and 4. Finally, the images are direct resized to the required size which is 170x242. An important point here is that the CNN input size is 150x220, the signature size is chosen

larger than the input size in order to enable random crop data augmentation during the training phase. During testing, a center crop of 150x220 is used as input.

Figure 3: Example signature

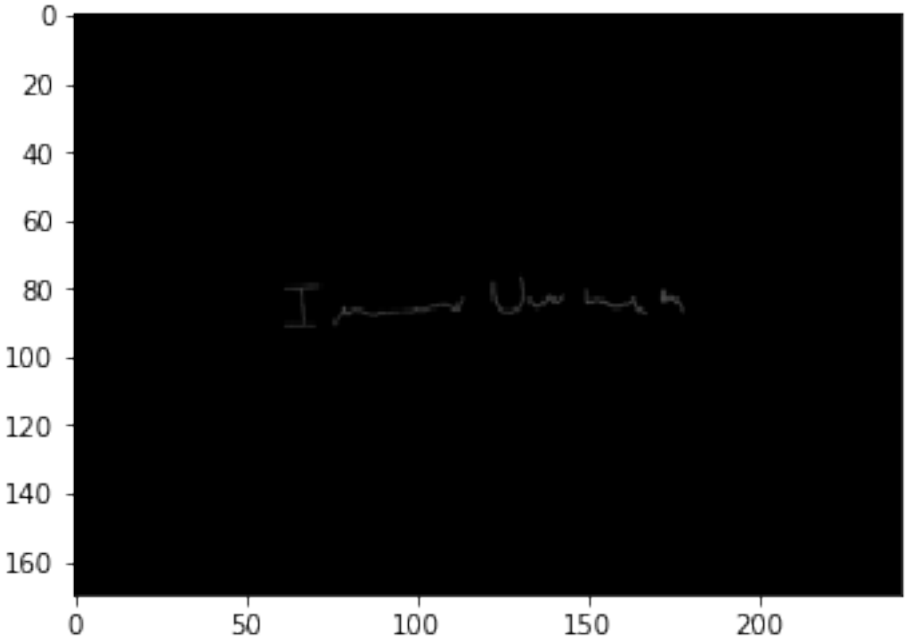

Figure 4: Example preprocessed signature

5. Initial Hypothesis and Feature Learning

5.1. Convolutional Neural Networks

Since CNNs have been more successful than hand engineered features in most of the state-of-the-art proposals in the academy, CNNs are chosen as the machine learning models in this proposal.

CNNs are multilayer Neural Networks consisting of several stacked convolutional layers with different kernel sizes, separated by pooling layers that are responsible of downsampling and summarizing the output of the convolutional layers, before passing the information to deeper layers. The nonlinearity function units are chosen as rectified linear units (ReLU) proposed by Nair et al. [23]. The nonlinearity functions are the key units for the architecture since they enable the model to learn nonlinear functions.

5.2. Siamese Networks

Siamese Networks are networks that connect two identical subnetworks that are usually CNNs. The name comes from the fact the two subnetworks are twins, sharing same architecture, same configurations, same parameters and same hyperparameters. They have been successfully applied for dimensionality reduction problems. These two subnetworks are joined together by a special loss function, usually representing the Euclidean distance between feature representations generated by two subnetworks. It is expected for examples that are in the same class to be as close as possible and examples that are in different classes to be as spread out as possible. The loss function to achieve that is named contrastive loss proposed by Chopra et al. [24], presented in Equation 1.

$$L(s_1, s_2, y) = \alpha(1-y)D_w^2 + \beta y \max(0, m - D_w)^2 \qquad (1)$$

The parameters s_1 and s_2 are the two samples (handwritten signature images), y is the binary indicator whether the two samples belong to the same class or not. α and β are the two coefficient constants and m is the margin. D_w is the Euclidean distance computed between the feature vectors. Using this particular loss function, Siamese networks aim to bring inputs of same class together and inputs of different classes away from each other.

While the Siamese networks are very useful for similar applications, their computational performance proved to be somewhat lacking for this particular problem since they are usually trying to do one versus one comparisons, using loss triplets; while we need to compare a signature as one versus many, due to large intra-class variance, and also wish the system to accept new users without a large re-training of the networks. However, the biggest issue was their inferior performance after a training without forged signatures. Siamese network model accuracy was around 66%, confirming the experiments carried out by Dey et al. [25].

5.3. Convolutional Neural Network Architecture

The network architecture employed in this paper was based on the proposal made by Krizhevsky et al. [26] also known as the AlexNet. Both the final proposal and the networks used in building the Siamese network are the same. The detailed breakdown of the architecture is presented in Table 1.

Table 1: CNN Architecture

Layer	Size	Parameters
Input	1x150x220	
Convolution	96x11x11	stride = 4
Max. Pool	96x3x3	stride = 2
Convolution	256x5x5	stride = 1, pad = 2
Max. Pool	256x3x3	stride = 2
Convolution	384x3x3	stride = 1, pad = 1
Convolution	384x3x3	stride = 1, pad =1
Convolution	256x3x3	stride = 1, pad =1
Max. Pool	256x3x3	stride = 2
Fully Connected	2048	
Fully Connected	2048	

In addition to given layers, each layer in Table 1 is combined with a batch normalization operation, proposed by Ioffe et al. [27] for regularization.

6. Developed Approach

Due to reasons listed above, hereby a more efficient method has been developed in order to accomplish the signature verification task, as stated in section 5.2, a better supervised learning method is needed in order to better handle the intra-class variance of the handwritten signature representations. The central idea is as follows: just like they do in Siamese network architecture, CNNs are able to learn feature representations for handwritten signatures, that also contain visual cues that represent the intra-class variance and genuine versus forgery variance. Using that method, the system will also be able to accept new user enrollments without a large computational overhead, since CNNs are to be trained only once and used repetitively in order to extract features from new signatures. The features that represent handwritten signatures are then fed into another classifier, Support-vector machine proposed by Cortes and Vapnik [28]. Formally, we want the CNN architecture to learn a function f(X) that can do embedding, which means we want a function that takes an input X as a grayscale image and projects it to a representation space, where signatures of the same class are close together and signatures of different classes are well separated in this new representation space. The inputs are tuples of (X,y) where X is the grayscale image and y is the user number; which means that fundamentally, the network should discriminate between signatures of different users. The loss function to declare is named cross-entropy and presented below as Equation 2.

$$L = -\sum_{j=0}^{n} y_{ij} \log P(y_j | X_i) \qquad \textbf{(2)}$$

In the Equation 2, y_{ij} is a binary parameter that represents whether the input signature X_i belongs to the user j. $P(y_j | X_i)$ is the assigned probability for some class j by the network to X_i. This loss function is minimized using gradient descent method. Minimization of this loss function should enable the network distinguish between signatures of different users. The last layer of the CNN architecture contains a softmax unit, a linear classifier in order to generate class probabilities by taking the feature representations as input. The key idea of metric learning is looking at those feature representations that are fed to softmax classifier and using another classifier to do predictions with small number of positive examples.

7. Experimentation Environment and Experiment Design

Full experiments have been carried out using GPDS-Synthetic 4000 and MCYT-75 datasets. CNN model has been trained on GPDS-Synthetic 4000 and its generalization ability at feature extraction has been tested on MCYT-75. To give a decent real life performance benchmark, the CNN has been trained on a 1000 user split from GPDS-Synthetic 4000 and 12 signatures per split for the multiclass classification problem. Thereafter, the development for SVM Classifiers using feature vectors have been carried out on a separate 200 user split. During this step, 12 genuine signatures are used as positive examples. Signatures from the whole CNN training and SVM training split (1200 users) have been used as negative examples. The best performing parameters for the SVM Classifiers are discovered as Radial Basis Function kernel, balanced class weights (inversely proportional to their example number), and scaled gamma, which is given as the Equation 3 and penalty parameter (C) equal to 1. It must be reminded that no forgeries are used for training, since it would not be a realistic benchmark including a training phase with forgeries.

$$\gamma = \frac{1}{n_{features} * Var(X)} \quad \textbf{(3)}$$

Where $n_{features}$ is the number of elements in the feature vector and Var(X) is the variance within. The MCYT-75 dataset is used as a performance estimator for real life conditions, therefore no split was made dividing the users. Also, the accuracy of the developed SVM classifiers have also been measured using the extracted features provided by Hafemann et al. [13] for GPDS, MCYT-75 and CEDAR, a more detailed breakdown of the testing conditions and splits have been given on Table 2.

Table 2: Training and Testing splits

Dataset	Training Split		Testing Split
	Genuine	Random Forg.	
GPDS-Synthetic 4000	$n \in \{1,...,12\}$	1200 x 14	(10 genuine, 10 forgery)
MCYT- 75	$n \in \{1,...,10\}$	74 x 15	(5 genuine, 15 forgery)
GPDS	$n \in \{1,...,12\}$	581 x 14	(10 genuine, 10 forgery)
CEDAR	$n \in \{1,...,12\}$	54 x 12	(10 genuine, 10 forgery)

8. Comparative Evaluation and Discussion

8.1. Evaluation of the System

CNNs are used to extract feature representations of the signature images and SVM Classifiers are trained to classify those representations. The performance of the classifiers is represented using the following metrics: False Rejection Rate (FRR), False Acceptance Rate (FAR), Equal Error Rates are also reported in two different formats, EER_{global} and EER_{user}, EER_{global} is the Equal Error Rate reported by selecting a decision threshold for the decision function outputs, only by looking at the global Equal Error point, EER_{global} brings this method to user-level, further optimizing the accuracy. However, these two EER metrics are not realistic and exist purely to enable comparison, since in ideal conditions, the classifiers and hyperparameters are tuned during the validation phase, not during the testing phase. Therefore, looking at those during the test phase would be a data leak.

During the training phase, the CNN is trained for 60 epochs, using batches of 32 images, with early stopping enabled. Further details are available through the corresponding public online source code repository [29]. The training classification loss, training classification accuracy, validation classification loss and validation classification accuracy are represented as Figure 5, 6, 7 and 8, respectively.

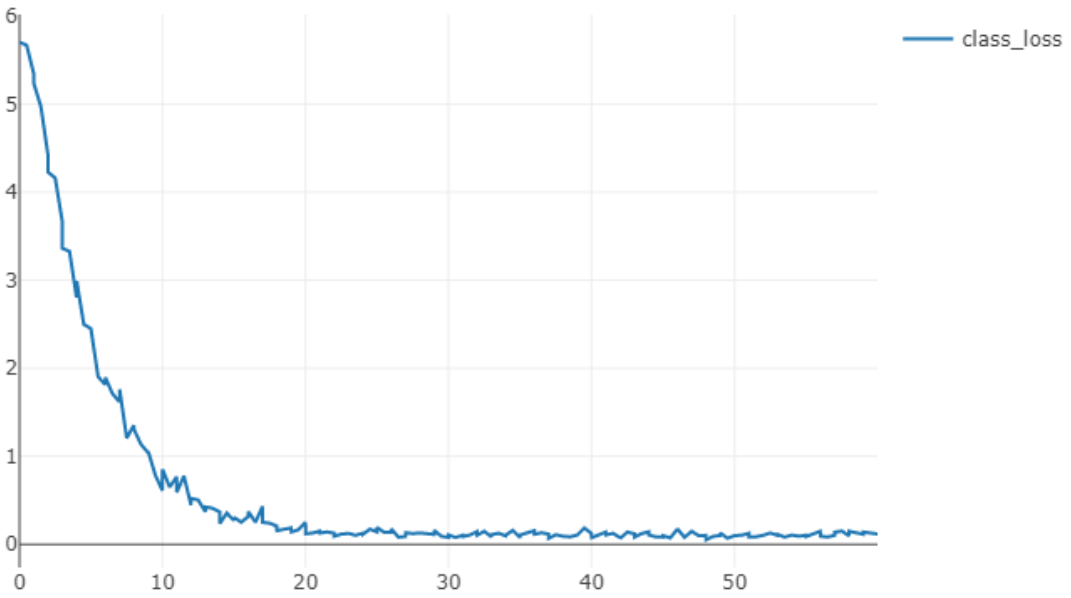

Figure 5: Classification loss/epochs.

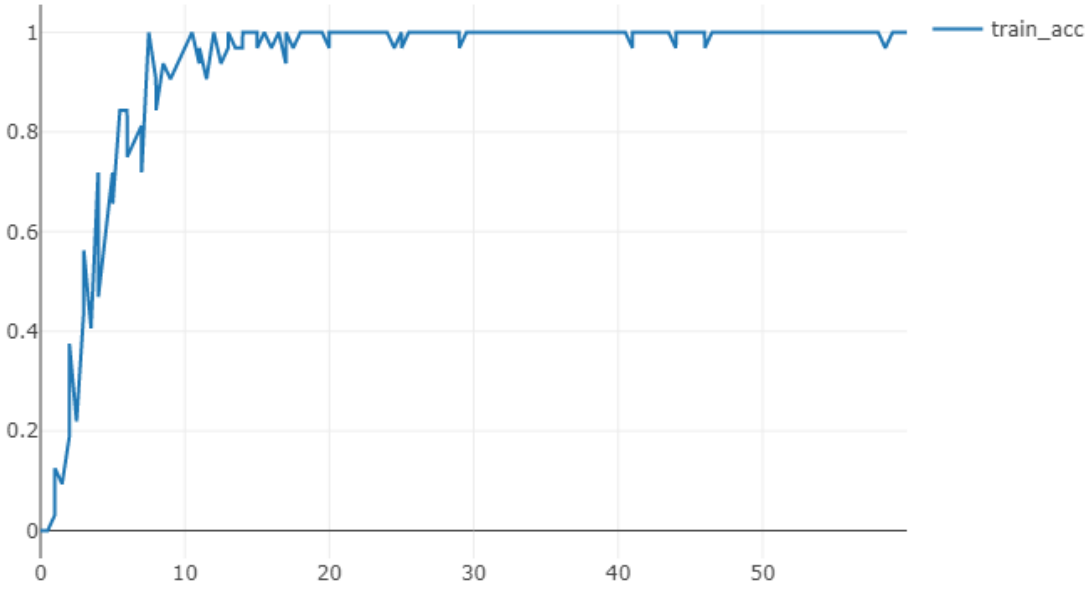

Figure 6: Classification accuracy/epochs.

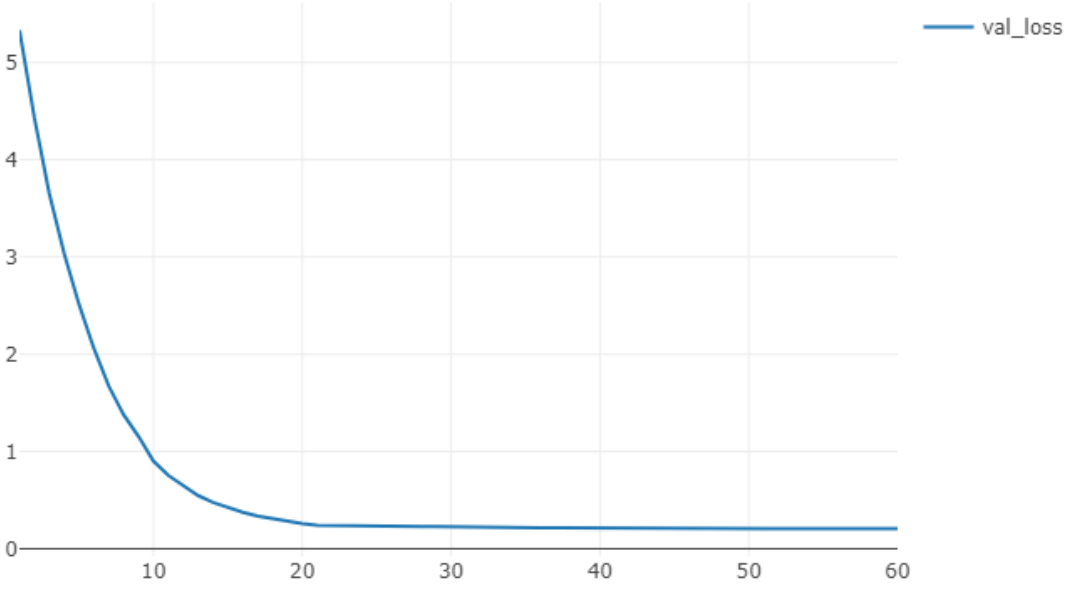

Figure 7: Validation loss/epochs.

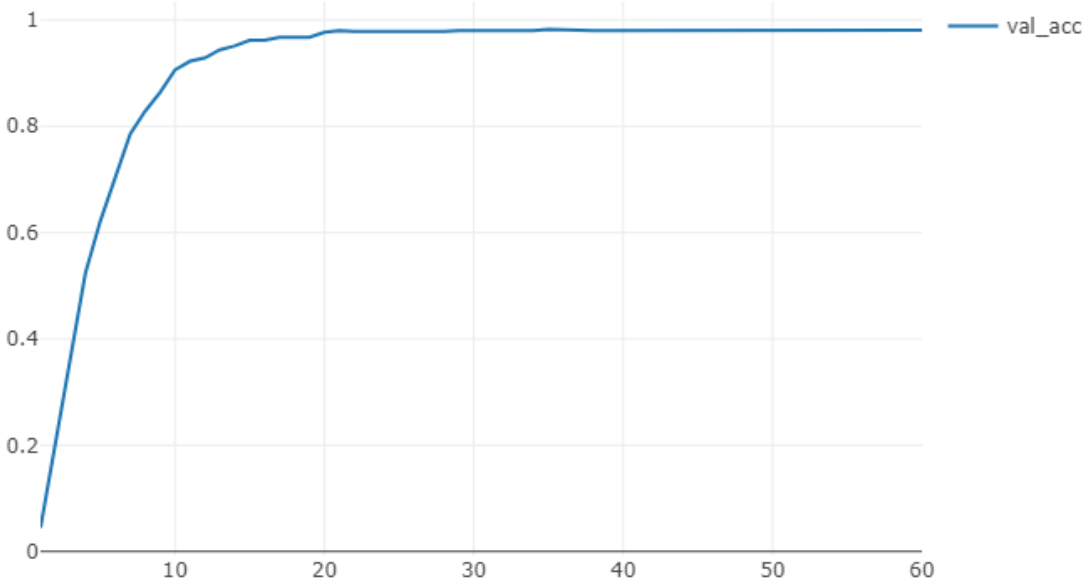

Figure 7: Validation accuracy/epochs.

In Table 3, the solutions to the signature verification problem in the academic literature using GPDS-Synthetic 4000 are presented. The proposed solution using 10 genuine samples also scores 7.17% False Rejection and 6.83% False Acceptance Ratio, using the thresholds discovered during the training and validation phases. Each phase has been repeated 10 times in order to decrease randomness involved.

Table 3: Performance on GPDS-Synthetic 4000, using 10 genuine 10 skilled forgery samples.

Reference	EER(%)
Dutta et al. [30]	26.33
Dey et al.[25]	22.24
Soleimani et al. [31]	13.30
Ferrer et al.,[32]	16.44
Serdouk et al.,[33]	16.68
Zhang et al.[34]	14.79
Proposed	**6.96**

In Table 4, the solutions to the same problem using MCYT-75 dataset are presented. It must be mentioned that CNN is not trained on this dataset in this proposal in order to benchmark its generalization ability. Since the other researchers opted for EER_{user}, the proposal error rate is reported accordingly.

Table 4: Performance on MCYT-75, using 10 genuine 10 skilled forgery samples.

Reference	EER(%)
Gilperez et al.[37]	6.44
Vargas et al.[36]	7.08
Ooi et al. [35]	9.87
Soleimani et al.,[31]	9.86
Hafemann et al.[14]	**2.87**
Proposed	4.00

8.2. Evaluation of the Feature Classifiers

Since some of the feature vectors are provided by Hafemann et al. [14], other types of feature classifiers have been developed and evaluated in this proposal, using the same training and testing conditions, on GPDS, MCYT-75 and CEDAR datasets. Their comparative evaluations are provided on Table 5, 6 and 7; respectively, using EER_{user}.

Table 5: Performance on GPDS using pre-extracted features for the proposal.

Reference	# genuine samples	EER(%)
Hu and Chen[38]	10	7.66
Guerbai et al.[39]	12	15.07
Serdouk et al.[40]	16	12.52
Soleimani et al.[31]	10	20.94
Yilmaz.[41]	12	6.97
Hafemann et al.[14]	10	1.69
Proposed	**10**	**1.33**

Table 6: Performance on MCYT-75, 10 samples each, included using pre-extracted features for the proposal.

Reference	EER(%)
Gilperez et al.[37]	6.44
Vargas et al.[36]	7.08
Ooi et al. [35]	9.87
Soleimani et al.,[31]	9.86
Hafemann et al.[14]	**2.87**
Proposed	4.00
Proposed(pre-extracted)	3.14

Table 7: Performance on CEDAR using pre-extracted features for the proposal.

Reference	# genuine samples	EER(%)
Chen and Srihari	16	7.9
Bharatri and Shekar	12	15.07
Guerbai et al.	12	5.6
Hafemann et al.[14]	12	4.63
Proposed	**12**	**2.87**

9. Conclusion and Future Work

In this work, a deep learning based approach for handwritten signature feature learning using CNNs; combined with a machine learning based approach powered by SVMs for image data classification using the features learned by the CNNs has been presented. It has been shown that features learned by deep learning architectures are much stronger than hand-crafted features and they are able to capture visual cues that create the distinction between a genuine signature and a forged one. Using these techniques, it is entirely possible to outperform the proposals that rely on hand-engineering by a very comfortable margin; it is also possible match and slightly outperform the state-of-the-art proposals in the academic literature. It must be also mentioned that these methods also work with small amount of data just like hand-engineered features, which is relatively uncommon for pure deep learning based proposals. During the development, the Python programming language [42] and its state-of-the-art scientific computing libraries, numpy [43], IPython [44], matplotlib [45], pandas [46], scipy [47], Cython [48] and PyTorch [49] have been used.

The experiments carried out on MCYT-75 prove that features learned on a synthetic dataset, GPDS-Synthetic 4000 generalize well to a biometric dataset, although Equal Error Rate is significantly worse than the performance of the SVM Classifiers on the biometric datasets, using pre-extracted features. It is entirely possible to analyze the reason for that, because it is highly likely for GPDS-Synthetic 4000 to be a much harder dataset compared to GPDS, MCYT-75 and CEDAR. However, this analysis should be made on the generative side, rather than the discriminative side.

There has been some exploratory research made regarding the CNN architectures, trying deeper networks, editing the final layer that dictates the loss, despite their somewhat better performance on the dataset they get trained on, the features they extract lose their generalization capability and they perform significant worse on biometric datasets.

Many more user classifiers have been tested instead of SVMs: different types of decision tree based models and multilayer perceptrons of different sizes. However, their performances were vastly inferior to SVMs. Also, SVMs with linear kernels also work quite well for the problem, performing slightly worse than the Radial Basis Function kernel based SVMs, indicating that the data is also linearly separable.

As future work, Generative Adversarial Networks are very promising for this domain, especially for creating forgery signatures, due to their non-availability under real conditions, since in this work it has been proven that the availability of a forged signature during the training phase vastly improves the classification performance of the signature verification system.

10. References

[1] H. Baltzakis and N. Papamarkos. A new signature verification technique based on a two-stage neural network classifier. *Engineering applications of Artificial intelligence*, 14(1):95–103, February 2001.

[2] A. El-Yacoubi, E. J. R. Justino, R. Sabourin, and F. Bortolozzi. Offline signature verification using HMMs and cross-validation. In *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, volume 2. IEEE, 2000.

[3] Luiz S. Oliveira, Edson Justino, Cinthia Freitas, and Robert Sabourin. The graphology applied to signature verification. In *12th Conference of the International Graphonomics Society*, pages 286–290, 2005.

[4] R. Sabourin and Jean-Pierre Drouhard. Off-line signature verification using directional PDF and neural networks. In *International Conference on Pattern Recognition*, pages 321–325, August 1992.

[5] J.-P. Drouhard, Robert Sabourin, and Mario Godbout. A neural network approach to off-line signature verification using directional PDF. *Pattern Recognition*, 29(3):415–424, 1996.

[6] Dominique Rivard, Eric Granger, and Robert Sabourin. Multi-feature extraction and selection in writer-independent off-line signature verification. *Int. Journal on Doc. Analysis and Recognition*, 16(1), 2013.

[7] Bailing Zhang. Off-line signature verification and identification by pyramid histogram of oriented gradients. *International Journal of Intelligent Computing and Cybernetics*, 3(4):611–630, 2010.

[8] R. Sabourin and G. Genest. An extended-shadow-code based approach for off-line signature verification. I. Evaluation of the bar mask definition. In *Int. Conference on Pattern Recognition*, October 1994.

[9] Dominique Rivard, Eric Granger, and Robert Sabourin. Multi-feature extraction and selection in writer-independent off-line signature verification. *Int. Journal on Doc. Analysis and Recognition*, 16(1), 2013.

[10] G.S. Eskander, R. Sabourin, and E. Granger. Hybrid writer-independentwriter-dependent offline signature verification system. *IET Biometrics*, 2(4):169–181, December 2013.

[11] Yoshua Bengio. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[12] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Analyzing features learned for offline signature verification using Deep CNNs. In *International Conference on Pattern Recognition*, pages 2989–2994, 2016.

[13] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Writer independent feature learning for Offline Signature Verification using Deep Convolutional Neural Networks. In *International Joint Conference on Neural Networks*, pages 2576–2583, July 2016.

[14] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70:163–176, October 2017.

[15] H. Rantzsch, H. Yang, and C. Meinel. Signature embedding: Writer independent offline signature verification with deep metric learning. In *Advances in Visual Computing*. Springer, 2016.

[16] Z. Zhang, X. Liu, and Y. Cui. Multi-phase offline signature verification system using deep convolutional generative adversarial networks. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, volume 02, pages 103–107, 2016.

[17] M. A. Ferrer, M. Diaz-Cabrera, A. Morales, (2015), "Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.37, n.3, pp. 667-680.
[18] M. A. Ferrer, M. Diaz-Cabrera, A. Morales, "Synthetic Off-Line Signature Image Generation", 6th IAPR International Conference on Biometrics, Madrid, 4-7 June 2013, pp. 1 - 7. doi: 10.1109/ICB.2013.6612969
[19] Ortega-Garcia, J., Fierrez-Aguilar, J., Simon, D., Gonzalez, J., Faundez-Zanuy, M., Espinosa, V., ... & Escudero, D. (2003). MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings-Vision, Image and Signal Processing*, *150*(6), 395-401.
[20] M. Blumenstein, Miguel A. Ferrer, J.F. Vargas, \93 The 4NSigComp2010 off-line signature verification competition: Scenario 2 \94, in proceedings of 12th International Conference on Frontiers in Handwriting Recognition, ISSBN: 978-0-7695-4221-8, pp. 721-726, Kolkata, India, 16-18 November 2010.
[21] M. K. Kalera, S. Srihari, A. Xu, Offline signature verification and identification using distance statistics, International Journal of Pattern Recognition and Artificial Intelligence 18 (07) (2004) 1339–1360. doi:10.1142/S0218001404003630.
[22] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and 33 Cybernetics 9 (1) 62–66. doi:10.1109/TSMC.1979.4310076.
[23] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.
[24] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: CVPR, 2005, pp. 539–546.
[25] Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J., & Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*.
[26] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
[27] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate
Shift, in: Proceedings of The 32nd International Conference on Machine Learning, 2015, pp. 448–456.
[28] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.
[29] Gonultas, B. M. (n.d.). Sigver_v2. Retrieved May 19, 2019, from github.com/gonultasbu/sigver_v2
[30] A. Dutta, U. Pal, J. Llad´os, Compact correlated features for writer independent signature verification, in: ICPR, 2016, pp. 3411–3416.
[31] Soleimani, A., Araabi, B. N., & Fouladi, K. (2016). Deep Multitask Metric Learning for Offline Signature
Verification. Pattern Recognition Letters, 80, 84-90. doi: 10.1016/j.patrec.2016.05.023
[32] Ferrer, M. A., Diaz-Cabrera, M., & Morales, A. (2015). Static signature synthesis: A neuromotor inspired approach for biometrics. *IEEE Transactions on pattern analysis and machine intelligence, 37*(3), 667-680.
[33] Serdouk, Y., Nemmour, H., & Chibani, Y. (2017). Handwritten signature verification using the quad-tree histogram of templates and a Support Vector-based artificial immune classification. *Image and Vision Computing, 66*, 26-35.

[34] Zhang, Z., Liu, X., & Cui, Y. (2016). *Multi-phase Offline Signature Verification System Using Deep Convolutional Generative Adversarial Networks.* Paper presented at the Computational Intelligence and Design (ISCID), 2016 9th International Symposium on.
[35] S. Y. Ooi, A. B. J. Teoh, Y. H. Pang, B. Y. Hiew, Image-based handwritten signature verification using hybrid methods of discrete radon transform, principal component analysis and probabilistic neural network 40 274–282. doi:10.1016/j.asoc.2015.11.039.
[36] J. F. Vargas, M. A. Ferrer, C. M. Travieso, J. B. Alonso, Off-line signature verification based on grey level information using texture features, Pattern Recognition 44 (2) (2011) 375–385. doi: 10.1016/j.patcog. 2010.07.028.
[37] A. Gilperez, F. Alonso-Fernandez, S. Pecharroman, J. Fierrez, J. Ortega-Garcia, Off-line signature verification using contour features, in: 11th International Conference on Frontiers in Handwriting Recognition, Montreal, Quebec-Canada, August 19-21, 2008, CENPARMI, Concordia University, 2008.
[38] J. Hu, Y. Chen, Offline Signature Verification Using Real Adaboost Classifier Combination of Pseudo-dynamic Features, in: Document Analysis and Recognition, 12th International Conference on, 2013, pp. 1345–1349. doi:10.1109/ICDAR.2013.272.
[39] Y. Guerbai, Y. Chibani, B. Hadjadji, The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters, Pattern Recognition 48 (1) (2015) 103–113. doi:10. 1016/j.patcog.2014.07.016.
[40] Y. Serdouk, H. Nemmour, Y. Chibani, New gradient features for off-line handwritten signature verification, in: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), 2015, pp. 1–4. doi:10.1109/INISTA.2015.7276751.
[41] M. B. Yilmaz, B. Yanikoglu, Score level fusion of classifiers in off-line signature verification, Information Fusion 32, Part B (2016) 109–119. doi: 10.1016/j.inffus.2016.02.003.
[42] Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org
[43] Travis E, Oliphant. A guide to NumPy, USA: Trelgol Publishing, (2006).
[44] Fernando Pérez and Brian E. Granger. IPython: A System for Interactive Scientific Computing, Computing in Science & Engineering, 9, 21-29 (2007), DOI:10.1109/MCSE.2007.53 (publisher link)
[45] John D. Hunter. Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9, 90-95 (2007),DOI:10.1109/MCSE.2007.55 (publisher link)
[46] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010) (publisher link)
[47] Travis E. Oliphant. Python for Scientific Computing, Computing in Science & Engineering, 9, 10-20 (2007),DOI:10.1109/MCSE.2007.58 (publisher link)
[48] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn and Kurt Smith. Cython: The Best of Both Worlds, Computing in Science and Engineering, 13, 31-39 (2011), DOI:10.1109/MCSE.2010.118 (publisher link)
[49] Paszke, A., Gross, S., Chintala, S., & Chanan, G. (2017). Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, *6*.