2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Signature Identification via Efficient Feature Selection and GPU-based SVM Classifier

Bernardete Ribeiro, *Senior Member, IEEE*, Noel Lopes and João Gonçalves

***Abstract*— The problem of handwritten signature recognition is considered significant in biometrics, in particular for determining the validity of official documents. The rationale consists of creating an off-line classifier to discriminate between fake (forged) and genuine digitalized signatures. In such applications containing thousands of samples machine learning techniques such as Support Vector Machines (SVM) play a preponderant role in overcoming the challenges inherent to this problematic. However, to deal with the computational burden of calculating the large Gram matrix, approaches such as Graphics Processing Units (GPU) computing are required for efficiently processing big image biometric data. In this paper, first, we present an empirical study for efficient feature selection concerning the signature identification problem. Second, an GPU-based SVM classifier that integrates a component of the open source Machine Learning Library (GPUMLib) supporting several kernels is developed. Third, we ran several experiments with improved performance over baseline approaches. From our study, we gain insights in both performance and computational cost under a number of experimental conditions, and conclude that the most appropriate model is usually a trade-off between performance and computational cost for a given experimental setup and dataset.**

I. Introduction

THE problem of handwritten signature recognition is a challenging one that plays an important role in validating many important transactions, such as issued checks, credit card shopping, authorization documents or even contracts. The idea consists of creating an off-line classifier to discriminate between fake (forged) and genuine signatures in a database of digitalized signatures, after identifying the author.

This problem is specially difficult for many reasons. The biometric data is a scanned 2D image. Therefore, unlike in on-line verification, dynamic characteristics, such as velocity, pen pressure and acceleration, which reflect specific individual motion style and are harder to fake can not be obtained. To exacerbate the problem, the biometric features of genuine and forged signatures can be extremely similar. Examples are the shapes, sizes and variations of signatures that lead to a confluence of factors extremely tricky to verify. Moreover, the sheer volume of biometric data in many applications require fast tools for model selection in order to expose better models. Preprocessing of offline handwritten biometric data is complex and motivates the holistic study of many features capable of proper capturing the intra-variational characteristics of the individual signatures and the optimal group of features for building better models.

Support Vector Machine (SVM) based on the structural risk minimization [1], [2] can provide good generalization in many applications. However, significant computational costs arise from calculating the large Gram matrix [3], in particular when big data is involved, as it is the case of most biometric problems. This has motivated research in fast learning methods [4], [5]. Nevertheless, most implementations still fail to take advantage of today multi-core architectures of neither the Central Processing Units (CPUs) nor the more powerful Graphics Processing Units (GPUs).

In this work we focus on a GPU-based SVM classifier, by extending our previous work on Multi-Threaded parallel CPU standalone SVM version (MT-SVM), which builds from scratch an implementation of the Sequential Minimal Optimization (SMO) algorithm [6]. We proceed in two main steps, first, by extracting and selecting features from the GPDS[1] biometric database images (see Figure 1 for examples of genuine and forged signatures in the database) and, second, by using these features to verify a given image. The GPU-based classifier SVM component was integrated in the GPUMLib[2]. Notwithstanding that our solver is for binary SVM, our method includes apart from the commonly used kernel functions, an Universal Kernel Function (UKF) [7] with good generalization properties. Therefore, our GPU-based SVM classifier has important advantages both in terms of computational cost and performance to solve the offline signature recognition problem. Our proposal is effective both in terms of the feature selection and also in the GPU-based SVM classifier. We obtained excellent results on the identification of an individual's signature despite the fact that a generic classifier configuration is difficult to achieve.

The paper is organized as follows. Section II introduces the related work in the area of signature recognition. Section III is devoted to feature extraction and selection. Section IV presents our proposed SVM-GPU component of GPUMLib. In section V the experimental setup is described, including the database, data sampling and system configuration. In Section VI we outline the results and in Section VII further discussion is given. In section VIII we draw the main conclusions.

II. Related Work

Previously, research was by done by Armand et al [8], Blumenstein et al. [9] and Ferrer et al [10] in order to study better

Bernardete Ribeiro is with the Department of Informatics Engineering, University of Coimbra and with the Center of Informatics and Systems of University of Coimbra (CISUC), Portugal (email: bribeiro@dei.uc.pt).

Noel Lopes is with CISUC and with the Polytechnic of Guarda, Portugal (email: noel@ipg.pt).

João Gonçalves is with CISUC, Portugal (email: jcgonc@dei.uc.pt).

[1](Grupo de Procesado Digital de Señales), available at `http://www.gpds.ulpgc.es/download/`.

[2]`http://gpumlib.sourceforge.net/`

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

Fig. 1: Examples of genuine and forged signatures from the GPDS database.

features from the original dataset. More information regarding the extraction algorithms are referenced in their papers. In the scope of biometric analysis, an important problem is to distinguish between genuine and forged signatures, which is a hard task. The continued motivation to investigate this problem may be attributed in part to its challenging nature which depends on various factors such as behavioral characteristics like mood, fatigue, energy, etc.. Feature extraction and pattern recognition comprise methods that indubitably have proven to be effective for setting up a signature verification system. Research has been very intensive in the last years and many approaches have been devised using mainly discriminative techniques [8], [10], [11], [12], [13]. In a recent work [14] the authors propose a generative approach based on deep learning. A deep neural network is trained by the contrastive divergence method introduced by Hinton [15]. The study was conducted in the biometric database GPDS mentioned above, although in a reduced number of image folders. However, due to the high computational cost, in this paper we present a GPU-based effective model selection methodology that contributes to circumvent this problem and a robust classifier capable of handling many different groups of features. The robustness brings a good trade-off between the False Positive Rate (FPR) and False Discovery Rate (FDR) while the computational cost is substantially reduced. In the next section further information of the database is given, followed by the explanation of the feature extraction and selection methodology.

TABLE I: Number of attributes per feature.

Feature	Attributes
Best Fit	4
Discrete Cosine Transform (DCT)	5
Geometric Parameters (Cartesian)	180
Geometric Parameters (Polar)	192
Gravity Center	1
Histogram Frequencies (hist)	6
K-Means	10
Maximum Intensity Points (maxint)	1
Modified Direction Feature (MDF)	160
Six-fold-Surface	6
Three-fold-Surface	3
Wavelet Transform Feature	12

III. Feature Extraction and Selection

A. Dataset

The database contains data from 300 individuals. For each individual there are 54 signatures (24 genuine plus 30 forgeries). The 24 genuine specimens of each signer were collected in single day writing sessions. The forgeries were produced under the following conditions: The forger imitates a genuine signature from the static image of the genuine signature (scanned at 300 DPI) and the forger is allowed to practice writing the signature for as long as s/he wishes. Each forger has to imitate three signatures of five signers in a single day writing session. The genuine signature shown to each forger is chosen randomly from the 24 genuine ones. Therefore, for each genuine signature, there are 30 simple forgeries made by 10 forgers from 10 different genuine specimens. Globally, the dataset consists of 16200 handwritten off-line signature recognition (each signature is a 649×462 pixels image). Additional information on this database can be found in Ferrer et al. [10].

For the test set we used 9 images and the remaining 45 for the training set. Both training and testing sets were randomly generated from the initial data, being the test set composed of 4 genuine signatures and 5 forged. The experiments were run 10 times per configuration.

In Table I we present the number of attributes for each extracted feature from the image dataset. For that purpose we used a tool developed in [16], [14].

B. Feature Extraction

The whole set of images was centered and borders were added to obtain equal sized objects. The image dataset has a very big size which make it impractical to use directly in memory. In this case, it would be composed of $16200 \times 649 \times 642 = 6.749.859.600$ pixels (roughly 6.75 Gigapixels). As each pixel would be translated to a single precision floating type (four bytes per pixel) its storage would require 25.15 GB of RAM.

In this kind of problems, methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) or non-linear dimensional reduction methods (KPCA), among others, would be unfeasible to use here for a number of reasons. In particular, because they would be blind towards the characteristics of the specific individuals. Hence, feature extraction techniques are an essential step of a signature verification system in order to achieve a good performance The following methods give a reasonable coverage of the broad set of features (see Table I) to tackle this kind of problems [8], [10] [14]. The results yielded using each one of these methods are discussed later in Section VI.

Gravity Center Angle

This feature consists in dividing an image using a centered vertical axis, cropping two equally sized sections of the image. The "Centroid" of each section is calculated and the angle of the vector between the two centroids is used as the final feature.

Maximum Intensity Points

This feature returns either the line or the column of the image containing the greatest number of black pixels.

Tri-fold surface

This feature corresponds on the proportion of pixels contained in three vertical divisions of the image. Therefore it represents the dispersion of pixels within the three sections.

Six-fold surface

Following the last feature this extends the algorithm by additionally dividing the image in three horizontal sections. However, the horizontal divisions of each vertical section are done using the gravity center of the vertical sections.

Best Fit

This feature represents the angle between the signature and the horizontal axis. Therefore, two lines are interpolated using the two centroids, one on the left side of the image and the other on the right side. Additionally, the authors added the proportion of pixels inside each centroid to represent the dimension of the centroids.

Geometric Parameters (Polar)

This feature characterizes the distribution of the image radially, starting on its gravity center. Therefore, the image is divided into equally sized angular sections using equidistant points on the outer edge of the image. Extracted features are the distance of each point to the center, its angle with the center and the proportion of black pixels contained in each section.

Geometric Parameters (Cartesian)

This feature identifies the image's morphology by doing an analysis using two Cartesian axis. The same principle behind the extraction of the polar features is applied, however using sections evenly distributed in a rectangle centered in the image.

Modified Direction Feature (MDF)

This feature identifies the direction of the different segments composing the signature's line and the location of the areas were pixels change from white to black. This is done either vertically or horizontally.

K-Means

Another feature which identifies the position of the main elements of the signature is K-Means clustering using the image's black pixels. The number of clusters is fixed for all the images and set to five.

Histogram Frequencies

In order to characterize the signature's intensity variations either vertically and horizontally this feature calculates the vertical and horizontal frequency histograms of the pixels in each image. These frequencies are calculated using the Fast Fourier Transform.

Discrete Cosine Transform (DCT) Frequencies

This feature uses the two dimensional Discrete Cosine Transform (DCT) to change the initial amplitude-time space to a new amplitude-frequency space, therefore representing the intensity in frequencies of the initial image. It is the same algorithm behind JPEG and some video compression codecs.

IV. GPU-BASED SVM CLASSIFIER

We have developed a GPU SVM component to accelerate the computations inherent to the determination of the large Gram matrix. The resulting SVM component integrates the GPUMLib (GPU Machine Learning Library) software [17], [18]. GPUMLib is an efficient GPU machine learning library, implemented in CUDA (Compute Unified Device Architecture), that aims at providing the building blocks for the development of high-performance Machine Learning (ML) software, by taking advantage of the GPU enormous computational power [17], [18].

A. SVM formulation

Given a set of n training points in a d dimensional feature space $\mathbf{x} \in \mathbb{R}^d$ each associated with a label $y_i \in \{-1, 1\}$ the binary soft-margin kernel SVM solves a linearly convex quadratic problem. The classification of a given sample $\mathbf{z}$ uses a subset of the training set upholding the support vectors. The SVM classification task is given by (1):

$$y(\mathbf{z}) = \text{sign}\left(\sum_{i=1}^{n_{SV}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{z}) + b\right) \tag{1}$$

where n_{SV} is the number of support vectors, α_i the Lagrange multipliers, K the kernel function and b the bias.

The SVM component of this library has been developed with flexible characteristics such as the most common used kernels (linear, Radial Basis Function (RBF), polynomial). In addition it incorporates a robust and generic Universal Kernel Function (UKF) kernel, which can excel greatly the generalization performance of SVM models using other kernels [7].

Listing 1: CUDA device function for computing the UKF kernel dot product.

```
__device__ cudafloat DotProductKernelUKF(int i0,
  int i1, cudafloat * samples, int n_samples,
  int num_dimensions, cudafloat * kernel_args)
{
  // K(x1,x2) = a(||x1 - x2||^2 + b^2)^-c
  cudafloat sum_dif_squared = 0;
  for (int i = 0; i < num_dimensions; i++) {
    cudafloat x0_i = samples[n_samples * i0 + i];
    cudafloat x1_i = samples[n_samples * i1 + i];

    cudafloat _dif = x0_i - x1_i;
    cudafloat _dif_sq = _dif * _dif;
    sum_dif_squared += _dif_sq;
  }

  cudafloat a = kernel_args[0];
  cudafloat b = kernel_args[1];
  cudafloat c = kernel_args[2];

  return a * CUDA_POW(sum_dif_squared + b * b, -c);
}
```

The Universal Kernel Function (UKF) which has been proved to satisfy Mercer kernel [7] conditions is defined as follows:

$$K(\mathbf{u},\mathbf{v}) = L(\|\mathbf{u}-\mathbf{v}\|^2 + \sigma^2)^{-\alpha} \tag{2}$$

where L is a normalization constant, $\sigma > 0$ is the kernel width and $\alpha > 0$ controls the decreasing speed around zero. This kernel aims to gather nearby points, in a higher dimension space, since they are strongly correlated. Hence, it can provide a small number of SVs and thus speeds up both the training and classification tasks. Additionally, it can yield better generalization performance [19].

B. SVMs implementations on GPU

To date, there are four implementations of SVM for the GPU: Catanzaro's gpuSVM [20], Herrero's multiSV [21], Carpenter's cuSVM and Lin's sparse SVM [22]. All of these are written in CUDA for NVIDIA's GPUs.

This work follows the first implementation of an SVM binary classifier using programmable GPUs, named "gpuSVM". As in Catanzaro's it explores both the costly compute bound step of the SMO algorithm and the update of the Karush-Kuhn-Tucker (KKT) conditions.

An implementation supporting multiple classes was put forward by Herrero"multiSVM" [21]. Their work is very similar to Catanzaro's, but it executes different binary classifiers at the same time on the GPU. Additionally, it uses NVIDIA's CUBLAS algebra library to help calculating the kernel matrix. Another one which is also largely based upon "gpuSVM" is "cuSVM" by Carpenter. The major improvement over Catanzaro's work is the usage of mixed precision floating point arithmetic. In "cuSVM" most computations are in 32-bit precision (float) but some computations like the sum of dot products are done and stored in double precision floating point (64-bit double). According to the author, this can be of extreme importance for some data set problems. Likewise [22] used sparse matrices for cache the kernel matrix. Their work is also very similar to Catanzaro's SVM. They claim an speedup over "gpuSVM" of $1.9\times$ to $2.41\times$.

TABLE II: NVIDIA GeForce 570 GTX characteristics.

Characteristic	Value
Number of scalar processors (cores)	480
IEEE single precision (float) peak performance	748.8 GFlops
Number of streaming multiprocessors	15
Shading clock speed	1.56 GHz
Memory size	1.25 GB
Memory bandwidth	152.0 GB/s
Shared memory per block	48 KB

C. Kernels for GPU Implementation

As stated before, our implementation features the UKF kernel. Listing 1 shows a CUDA device (GPU) function that computes the dot product for the UKF kernel.

Our implementation requires a total of five CUDA kernels (InitializeSMO, UpdateAlphas, UpdateKKTConditions, FirstOrderHeuristic1stPass and FirstOrderHeuristicFinalPass) which are called in sequence, as depicted in Figure 2, until the SMO algorithm converges.

V. Experimental setup

A. System Configuration

The system's configuration for running the experiments is an Intel Core i5 at 3.33 GHz with 12 GB of RAM and a NVIDIA Geforce GTX 570 whose characteristics are described in Table II.

Our GPU-based SVM classifier is an SVM binary solver, improved by methods that represent well the data by means of enriched kernel functions. Hence, the multi-class functionality is not supported. Therefore the identification is performed by comparing one individual against another, "one against one" or "coupling pairwise" also known as "round robin" strategy. An interesting study is held in [23] where the posterior probability of the test error is compared in both strategies "one against one" and the "one against all" to handle the multi-class SVM in an handwriting recognition problem; the results on handwritten recognition databases conclude that the approach to be used depends on the characteristics of the problem at hand, although in skewed datasets the "one against one" performs better. We also developed an external driver program which controls the GPU SVM allowing for fine tune of the grid search algorithm, K-Fold cross-validation or a custom validation scheme as well as supporting multiple parallel executions of either the Multi-Threaded CPU or the GPU-based SVMs. This is useful as in most experiments the training process is faster than the dataset loading time. Therefore it is possible to run multiple training instances at the same time in order to minimize the total execution time.

B. Signatures Data Sampling

The first experience (A) comprises the identification of original and forged signatures. For that purpose, we used all

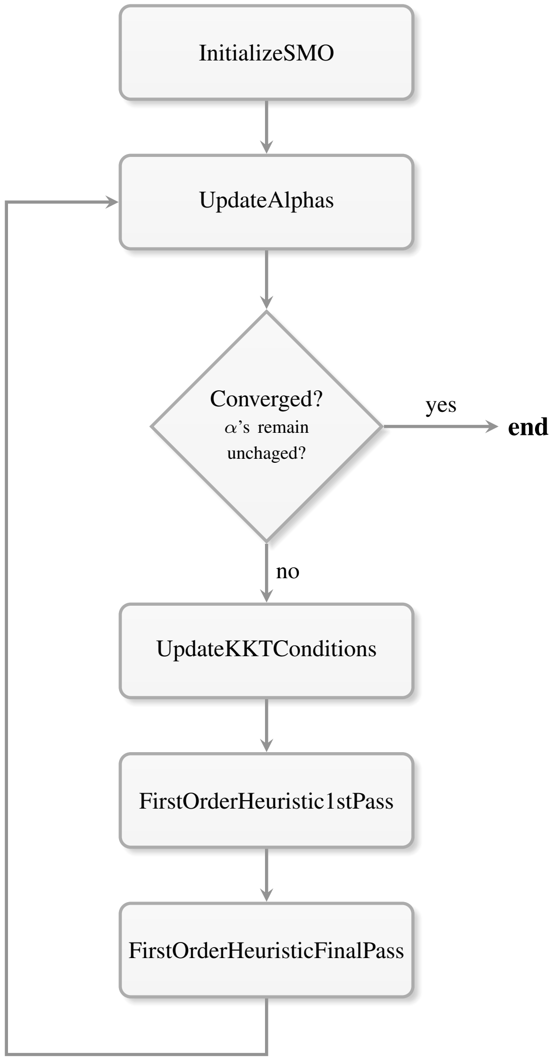

Fig. 2: Sequence of kernel calls (SMO algorithm).

the 300 individuals and studied each group of features. We run two experimental tests. In the first, we used the RBF kernel with the configuration setup (obtained by grid search) illustrated in Table III. In the second, the feature combination Discrete Cosine Transform (DCT) + Modified Direction Feature (MDF) was tested for all the supported kernel functions by our GPU implementation, with the configuration setup (once again obtained by grid search) depicted in table IV. No other feature combinations were used, in order to reduce the computational cost of the grid-search algorithm. Moreover, the results were obtained using 5-Fold cross-validation.

In the second experience (B), instead of using all the 300 groups of signatures we have exploited several groups or combinations of features for each individual. Therefore, this experience consisted of identifying, for each person, if a signature was either original or forged. Only the RBF kernel was employed.

The third experience (C) consisted on identifying a signature according to the related individual, using both the original and forged signatures. As we described before, since our classifier is currently binary we simulated a One-Against-One multi-class classifier, that is, we trained and tested each individual class against one of the others. Thus, for a dataset with c classes, $(c \times (c-1)/2)$ training tasks are required. In this third experience we performed $(300 \times 299)/2 = 44850$ runs, excluding the validation procedure for each experience. As the cost involved in the training process is high, we only used the RBF kernel and 5 K-Fold cross-validation procedure. Note that GPU implementation was crucial in performing such large number of runs.

TABLE III: RBF kernel configuration used in the first experiment, the generic identification of original and forged signatures.

Features	C	γ
polar	0.08	0.04
cartesian	0.64	0.08
mdf	10.24	0.04
wavelet	0.01	64.00
bestfit	0.01	8.19
dct	0.08	2.56
gravitycenter	0.01	0.00
hist	0.01	8.19
kmeans	0.04	0.00
maxint	0.01	0.01
sixfold	0.02	3.78
tri-fold-surface	1.28	0.02
dct+mdf	11.71	0.02
dct+mdf+cart	8.00	0.01

TABLE IV: The configuration of all kernel functions used in the experience A with combination of features DCT + MDF.

Kernel	C	γ	L	σ	α	b	q
RBF	11.710	0.02					
UKF	10.240		1.00	2.56	0.25		
Linear	0.001						
Polynomial	0.100				0.99	13.40	0.04
Sigmoid	0.100				0.08	10.24	

C. Performance Metrics

In order to evaluate the binary decision task of the identification models, we defined several measures based on the possible outcomes of the classification, such as, False Positive Rate ($FPR = \frac{FP}{FP+TN}$), and False Discovery Rate ($FPR = \frac{FP}{FP+TP}$), as well as combined measures, such as, the van Rijsbergen F_β measure, which combines recall ($R = \frac{TP}{TP+FN}$) and precision ($P = \frac{TP}{TP+FP}$) in a single score ($F1 = \frac{2PR}{P+R}$), an harmonic average between precision and recall.

VI. Results

In this section we present the results for the three experiences mentioned above.

1) Experience A: Tables V and VI present the obtained results for the first experiment where the single objective was to identify if a given signature is original or forged. Table V

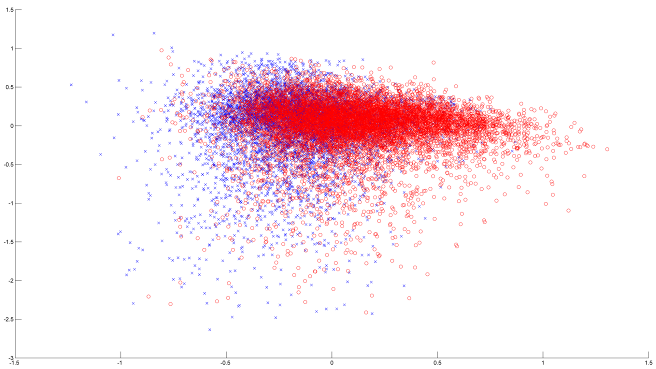

Fig. 3: Two dimensional projection of LDA for the experiment 1 using features MDF and DCT.

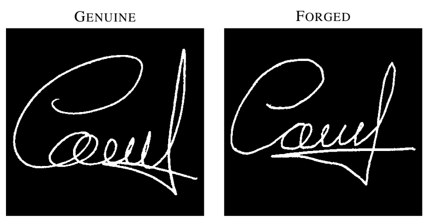

Fig. 4: Examples of genuine and forged signatures for the individual 23/300 from the GPDS database.

shows the results achieved using only the RBF kernel while Table VI shows the performance using all the available kernels in our SVM implementation. Since, in the first experiment, the best F-Score was obtained using the DCT+MDF features (see Table V), in the last experiment only those features were used. Incidentally, on average, using the DCT+MDF features, the proposed SVM-GPU yielded a speedup of 11.22 times over the well-known Library for Support Vector Machines (LIBSVM) software [24] and a speedup of 3.95 times over our previous Multi Threaded SVM implementation [6].

We also look at LDA as a supervised feature reduction algorithm which aims to reduce the input space dimensionality while maximizing the classes separation. In Figure 3 an LDA projection using both features DCT and MDF is shown, in order to visualize the level of the task's complexity at hand.

2) Experience B: With respect to the second experience, in Table VII we present the average results (and standard deviations) for the performance metrics described above and for several runs of groups of features. For model selection of best kernel parameters, we take as an example the individual of the Folder nr. 23 (out of 300) with signatures samples represented in Figure 4) for the genuine author (left) and forged one (right). In Figures 5 and 6 we show the grid-search contour map for the individual 23 using, respectively, DCT features and MDF features.

3) Experience C: The third experiment's results are shown in Table VIII, which illustrate the average and standard deviations of the performance metrics as indicated in the above experiments.

TABLE V: Performance for the first identification experiment in Experience A using the RBF kernel.

Features	Accuracy	F-Score	FPR	FDR
mdf	77.46±0.72	72.87±1.04	**15.08±1.08**	21.65±1.23
dct	67.18±0.92	69.52±0.87	46.48±1.91	40.80±0.98
cart	66.71±0.65	69.07±0.77	46.85±1.19	41.18±0.99
polar	57.77±1.21	62.71±0.81	59.93±3.70	48.35±1.42
six-fold-surface	52.80±1.16	61.55±0.42	72.98±3.75	51.74±0.68
tri-fold-surface	44.50±0.07	61.54±0.04	99.83±0.16	55.53±0.03
gravity	44.44±0.68	61.54±0.65	100.00±0.00	55.56±0.68
kmeans	44.45±0.89	61.53±0.86	100.00±0.00	55.55±0.89
maxint	44.54±0.81	61.45±0.83	99.44±2.97	55.53±0.68
wavelet	45.08±0.87	61.10±0.82	96.52±0.63	55.42±0.85
bestfit	53.95±0.86	59.18±0.62	63.01±3.02	51.16±0.88
hist	55.97±0.89	54.68±1.04	47.08±1.69	49.60±1.16
dct+mdf	79.42±1.13	**79.03±0.92**	26.83±2.89	27.69±1.85
dct+mdf+cart	**80.53±0.73**	77.82±1.07	16.63±3.90	**21.01±2.86**

TABLE VI: Performance for the second experiment in Experience A. using the combination of features MDF + DCT for all kernels.

Kernel	Accuracy	F-Score	FPR	FDR
Linear	60.33±7.21	53.34±22.30	41.45±25.65	40.40±12.98
Polynomial	54.33±9.22	62.45± 3.42	70.06±25.19	49.45± 6.14
RBF	79.42±1.13	79.03± 0.92	26.83± 2.89	27.69± 1.85
Sigmoid	44.41±0.14	61.40± 0.20	99.65± 0.91	55.60± 0.05
UKF	**79.97±1.63**	**79.29± 1.16**	**25.05± 6.04**	**26.25± 3.76**

VII. Discussion

The analysis of results is conducted for the research design indicated in the previous sections. Regarding Experience A, from Table V and Table VI a generic analysis indicates that the results clearly show the global original/forged signature identification task is quite complex. Most features are (probably) redundant and/or correlated. Furthermore, for all groups considered False Positive Rate and False Discovery Rate are very high, which indicates that the model with RBF kernel is unable to verify correctly the whole set of signatures. Intuitively this can suggest that identifying forged signatures, regardless of the individual they represent, is problematic. Thus the performance of the resulting classifier is poor with low mean values of accuracy and F-score with exception for the DCT and MDF features. In fact, MDF identifies the line direction of the image signature and extracts somehow its structure while DCT gives the information of intensity of frequency in the transformed image. Thus under the above conditions a generic classifier is out of sight.

In a more detailed analysis, we observe that the feature MDF alone yielded promising results and when combined with either the DCT or the Cartesian coordinates resulted in high F-Scores (79.03%) and low FPR (21.01%).

To inspect how the selected group of features MDF + DCT would lead to an improved model under better ways to represent the similarity among data points, a further study varying the kernel functions is presented in Table VI. The results show that the performance was slightly improved when

TABLE VII: Performance for the Experience B (Forged/original signature identification per individual).

Features	Accuracy	F-Score	FPR	FDR
cartesian	77.23±11.39	67.54±16.92	10.64±12.24	15.98±18.38
mdf	76.74±11.56	66.77±18.05	10.62±12.20	15.82±17.96
polar	71.13±12.84	59.45±18.78	13.84±14.39	21.44±21.88
wavelet	61.63±13.09	42.06±19.90	16.36±16.66	28.96±27.52
kmeans	54.00±13.19	31.32±19.58	22.71±20.11	38.84±30.81
six-fold-surface	67.19±12.58	50.33±18.91	13.10±14.32	22.83±23.70
dct	79.16±10.55	69.06±16.00	**7.97± 9.50**	13.06±15.54
histogram	64.33±12.68	45.16±19.64	14.44±14.57	26.92±25.40
bestfit	66.18±12.43	48.54±19.08	14.08±14.60	25.30±24.72
tri-fold-surface	65.73±11.96	57.87±16.45	29.02±18.19	35.76±21.37
gravity center	59.34±12.17	52.23±16.19	35.92±19.97	42.92±21.96
max intensity angle	55.93±12.30	46.86±16.99	37.66±20.30	46.93±22.38
mdf + dct	78.76±11.01	69.74±16.67	8.60±10.47	12.87±15.59
polar + cartesian	73.71±12.65	61.02±18.95	10.36±12.59	15.55±18.57
mdf + cartesian	78.33±11.07	69.28±16.58	9.29±11.08	13.45±16.30
dct + cartesian	78.70±11.55	69.81±17.46	9.45±10.98	14.00±16.44
dct + polar + mdf	75.75±12.02	64.87±18.35	10.24±12.12	15.02±17.58
dct + cart + mdf	79.45±11.50	71.16±16.71	8.78±10.80	12.96±15.74
dct + cart + mdf + polar	**79.92±11.07**	**71.62±16.59**	8.18± 9.78	**12.33±14.93**

using the UKF kernel while the FPR decreased by 1.78% and 1.44%, respectively. With this kernel more Support Vectors (SVs) than with the RBF kernel were found, allowing a better matching of the decision boundary. For the remaining kernels the results yield worst values for the performance metrics (e.g. F-Score $< 63\%$)).

Similar conclusions can be taken for the Experience B where the goal was the identification of original vs. forged signatures for each person. Again, we found that the K-Means, Histogram, Best-Fit, among others, are not good feature indicators for the image signature recognition problem. In spite of smaller False Positive Rate (FPR) than in the previous experience, we observed that Recall (used in the F-score measure) attained a high value of False Negatives harnessing the task of correct positives finding. Regarding MDF and DCT features the results yielded by the model were reasonably good. Moreover, by combining them with Cartesian and Polar coordinates we obtained the best original/forged identification, corresponding to a F-score of 71.62% and a FDR of 12.33%.

Bearing in mind that the best features so far are the MDF and DCT we used a grid search for finding the RBF kernel model parameters. Interestingly, the results depicted in Figure 5 and Figure 6 clearly demonstrate contradictory results regarding the parameter choice. When using the DCT feature, the best results are obtained both with increased γ and penalization constant C. On the other hand, using the MDF feature requires a smaller γ and a higher C that should be used in order to achieve higher performance in terms of the best trade-off between FPR and FDR. Thus, it is hard to conciliate both features in a unique model unless some transformation is applied to one of the features, in order to make both compatible with the same parameters space. Another observation is that for both features MDF and DCT a simpler solution could be the use of a specialized SVM to separate the kernel parameters for each feature in a multi-kernel learning task. We think multi-kernel would be adequate for further explore this feature combination.

Regarding the experience C, the One-Against-One person

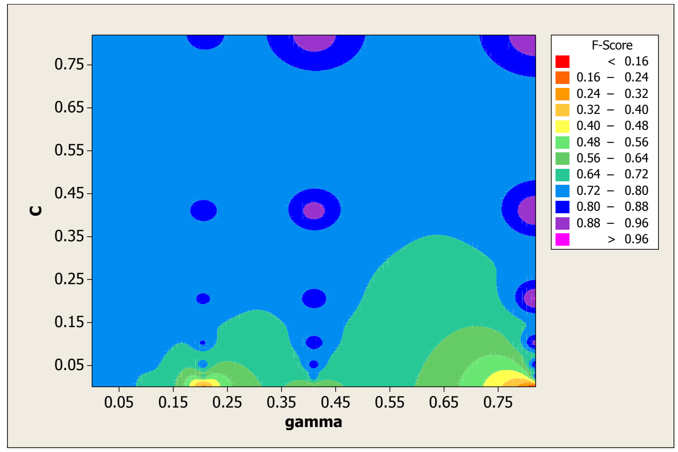

Fig. 5: F-Score RBF grid search using the DCT features for the detection of forged/original signature identification, author number 23.

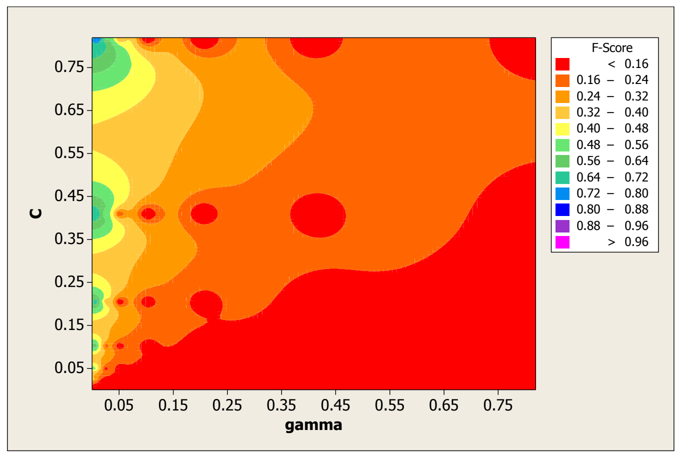

Fig. 6: F-Score RBF grid search using the MDF features for the detection of forged/original signature identification, author number 23.

TABLE VIII: Results in Experience C with One-Against-One signature author identification.

Features	Accuracy	F-Score	FPR	FDR
cartesian	97.62± 2.12	97.58± 2.16	2.62± 3.05	2.43± 2.91
mdf	**97.94± 1.90**	**97.80± 2.17**	**2.28± 2.77**	2.11± 2.67
polar	91.97± 5.25	91.56± 5.99	8.71± 8.62	7.73± 7.79
wavelet	82.87± 6.26	82.06± 8.53	17.79±13.72	14.05±10.40
kmeans	69.50± 8.92	68.47±12.51	31.87±21.25	25.72±15.48
six-fold-surface	89.60± 4.63	89.31± 5.53	10.62± 8.91	8.69± 7.31
dct	81.54± 6.62	80.65± 7.64	15.69±12.38	14.55±10.41
histogram	74.55± 7.59	73.12±10.23	24.25±17.48	20.82±13.69
bestfit	90.42± 4.22	89.88± 4.82	8.83± 7.18	8.09± 6.33
tri-fold-surface	85.51± 5.20	85.08± 5.97	14.47± 8.05	14.30± 7.69
gravity center	74.92± 7.07	73.92± 8.38	23.29±10.87	23.43±10.55
max intensity angle	66.99± 7.99	66.16± 9.50	33.12±12.44	32.28±11.93
mdf + dct	98.13± 1.75	98.14± 1.83	2.36± 2.70	**2.05± 2.48**
polar + cartesian	90.46± 6.27	90.08± 7.29	11.81± 9.57	9.81± 8.46
mdf + cartesian	96.53± 2.65	96.28± 3.05	4.35± 4.52	3.76± 4.03
dct + cartesian	97.50± 2.29	97.30± 2.60	2.81± 3.43	2.56± 3.15
dct + polar + mdf	91.72± 5.97	91.68± 6.43	10.78± 9.67	8.81± 8.03

identification task, we obtained in general excellent results. This was made possible by our GPU-based SVM classifier since the computational cost was handled in a expedite way. Excluding some features such as K-Means and Max-Intensity-Angle, the majority of features yielded in Accuracy and F-Score values above 80%. Again, the best features found so far like MDF, Cartesian and Polar Coordinates yielded F-Scores above 90%. Specifically, the MDF attained an F-Score of 97.80%. Combining both MDF and DCT features Accuracy was of 98.13%, F-Score of 98.14% corresponding to the highest Precision and Recall, 97.88% and 98.83%, respectively. It is worth of mentioning that with MDF feature alone FPR was of 2.28% and MDF + DCT the FDR was of 2.05% indeed remarkable values for the problem of signature identification. An excellent trade-off between performance and computational cost turned out possible due to the excellent capabilities of both steps involved in setting the solution of this problem.

VIII. Conclusion

Signature identification is a very important problem in authentication in several areas such as personal identification, security, bank transactions, etc.. Many efforts have been put to tackle the verification of signatures which contain biometric information. Often the databases are very large and such big data appears difficult to handle. Additionally, in offline settings, the lack of the dynamic characteristics makes the problem hard to solve. Fast machine learning algorithms that are able to extract relevant information from large repositories play an important role. To this end, in this paper we proposed a two-step process which examines the best group of features extracted form the biometric images and a GPU-based SVM classifier with characteristics to well-feature this problem. The component integrates the GPUMLib, by extending our previous work on multi-threaded parallel MT-SVM which parallelizes the SMO algorithm. Our implementation uses the power available on multi-core GPUs and efficiently learns (and classifies) the signatures exposing good properties in scaling data. Additionally, the UKF kernel which has good generalization properties in the high-dimensional feature space has been included. When faced with the problem of identifying a person's signature, our study demonstrated that this task is accessible and provides excellent results, even with a single feature (MDF). In further work, we intend to generalize the solver to other types of SVM problems such as (latent) structural SVMs. In another line of work, further exploration on multi-kernel learning particularly when the best groups of features (MDF and DCT) are at stake.

References

[1] V. Vapnik, *The nature of statistical learning theory*. Springer-Verlag, 1995.

[2] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[3] B. Chen, S. Zhao, P. Zhu, and J. C. Prncipe, "Quantized kernel least mean square algorithm." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, 2012. [Online]. Available: http://dblp.uni-trier.de/db/journals/tnn/tnn23.html#ChenZZP12

[4] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Optimized fixed-size kernel models for large data sets," *Computational Statistics and Data Analysis*, vol. 54, no. 6, pp. 1484–1504, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2010.01.024

[5] L. Hoegaerts, J. A. K. Suykens, J. Vandewalle, and B. D. Moor, "Subset based least squares subspace regression in RKHS," *Neurocomputing*, vol. 63, pp. 293–323, 2005.

[6] J. Gonçalves, N. Lopes, and B. Ribeiro, "Multi-threaded support vector machines for pattern recognition," in *Proc. of International Conference on Neural Information Processing, Part II, LNCS 7664*, T. H. et al. (Eds.), Ed. Springer-Verlag Berlin Heidelberg, 2012, pp. 616–623.

[7] R. Zhang and W. Wang, "Facilitating the applications of support vector machine by using a new kernel," *Expert Systems with Applications*, vol. 38, pp. 14 225–14 230, 2011.

[8] S. Armand, M. Blumenstein, and V. Muthukkumarasamy, "Off-line signature verification using an enhanced modified direction feature with single and multi-classifier approaches," *Computational Intelligence Magazine, IEEE*, vol. 2, no. 2, pp. 18 –25, 2007.

[9] M. Blumenstein, X. Liu, and B. Verma, "A modified direction feature for cursive character recognition," in *IEEE International Joint Conference on Neural Networks*, vol. 4, 2004, pp. 2983–2987.

[10] M. Ferrer, J. Alonso, and C. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 6, pp. 993–997, 2005.

[11] M. Bluemenstein and X. Y. Liu, "A modified direction feature for cursive character recognition," in *IEEE-IJCNN*, 2004, pp. 2983–2987.

[12] P. D. Gader, M. Mohamed, and J.-H. Chiang, "Handwritten word recognition with character and inter-character neural networks," *IEEE Trans. on Sys., Man and Cyber. - Part B: Cybernetics*, vol. 27, no. 1, pp. 158–164, 1997.

[13] V. Nguyen, M. Blumenstein, V. M, and G. Leedham, "Off-line signature verification using enhanced modified direction features in conjunction with neural classifiers and support vector machines," in *IEEE-ICDAR*, 2009, pp. 1300–1304.

[14] B. Ribeiro, I. Gonçalves, S. Santos, and A. Kovacec, "Deep learning networks for off-line handwritten signature recognition," in *in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications Lecture Notes in Computer Science Volume (LNCS)*, vol. 7042. Springer-Verlag, 2011, pp. 523–532. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25085-9_62

[15] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[16] I. Gonçalves and S. Santos, "Off-line signatures verification system," University of Coimbra, Portugal, Tech. Rep. trp-#10/11, 2011.

[17] N. Lopes and B. Ribeiro, "GPUMLib: An efficient open-source GPU machine learning library," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 355–362, 2011.

[18] N. Lopes, B. Ribeiro, and R. Quintas, "GPUMLib: A new library to combine machine learning algorithms with graphics processing units," in *10th International Conference on Hybrid Intelligent Systems (HIS)*, 2010, pp. 229–232.

[19] N. Ayat, M. Cheriet, C. Suen, and M. C. C. Y. Suen, "KMOD - a two-parameter SVM kernel for pattern recognition," in *16th International Conference on Pattern Recognition*, vol. 3, 2002, pp. 331–334.

[20] B. Catanzaro, N. Sundaram, and K. Keutzer, "Fast support vector machine training and classification on graphics processors," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 104–111. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390170

[21] S. Herrero-Lopez, J. R. Williams, and A. Sanchez, "Parallel multiclass classification using SVMs on GPUs." in *GPGPU 10*, 2010, pp. 2–11.

[22] T.-K. Lin and S.-Y. Chien, "Support vector machines on GPU with sparse matrix format," in *Proceedings Ninth International Conference on Machine Learning and Applications (ICMLA 2010)*. IEEE Computer Society, 2010, pp. 313–318.

[23] J. Milgram, M. Cheriet, and R. Sabourin, ""one against one" or "one against all": Which one is better for handwriting recognition with SVMs?" in *10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006)*, 2006, pp. 1–6.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.