# Robust Off-line Text Independent Writer Identification Using Bagged Discrete Cosine Transform Features

Faraz Ahmad Khan [1*], Muhammad Atif Tahir [1,2], Fouad Khelifi [1], Ahmed Bouridane [1], Resheed Almotaeryi [1]

[1] *Department of Computer Science and Digital Technologies, Engineering and Environment, Northumbria University at Newcastle*

[2] *Systems Research Laboratory Computer Science, FAST- National University of Computer and Emerging Sciences, Karachi, Pakistan*

**Abstract**

Efficient writer identification systems identify the authorship of an unknown sample of text with high confidence. This has made automatic writer identification a very important topic of research for forensic document analysis. In this paper, we propose a robust system for offline text independent writer identification using bagged discrete cosine transform (BDCT) descriptors. Universal codebooks are first used to generate multiple predictor models. A final decision is then obtained by using the majority voting rule from these predictor models. The BDCT approach allows for DCT features to be effectively exploited for robust hand writer identification. The proposed system has first been assessed on the original version of hand written documents of various datasets and results have shown comparable performance with state-of-the-art systems. Next, blurry and noisy documents of two different datasets have been considered through intensive experiments where the system has been shown to perform significantly better than its competitors. To the best of our knowledge this is the first work that addresses the robustness aspect in automatic hand writer identification. This is particularly suitable in digital forensics as the documents acquired by the analyst may not be in ideal conditions.

*Keywords:* Writer identification, handwritten offline documents, text independent, DCT, bagging, multiple classifiers, robust writer identification.

## 1. Introduction

Handwriting has been shown to be a very strong identifying characteristic of a person and can be a useful behavioural biometric trait. This makes handwriting an important tool for use by forensic document experts to determine the author of an unknown sample. Due to the sheer size of handwriting databases it will take a forensic expert a long time to manually compare the unknown sample with all of the known samples within the database. Therefore, automated handwriting identification algorithms can be very useful by making the identification of an unknown sample of text from a large database of known writers quite fast with high confidence, (Al-Maadeed, 2012; Fiel and Sablatnig, 2012; Franke and Köppen, 2001). This greatly reduces the work of the forensic analyst when comparing an unknown sample to tens of thousands of documents.

Writer identification is the process of determining the author of an unknown sample of handwritten text. The system must be made familiar with a set of documents from known writers before it can assign an unknown sample to one of the writers already known to it. The writer with the highest similarities to

*Corresponding author. Tel: +44 7883531748

*Email addresses:* faraz.khan@northumbria.ac.uk (Faraz Ahmad Khan [1]), muhammad.tahir@northumbria.ac.uk (Muhammad Atif Tahir [1,2]), fouad.khelifi@northumbria.ac.uk (Fouad Khelifi [1]), ahmed.bouridane@northumbria.ac.uk (Ahmed Bouridane [1]), resheed.almotaeryi@northumbria.ac.uk (Resheed Almotaeryi [1])

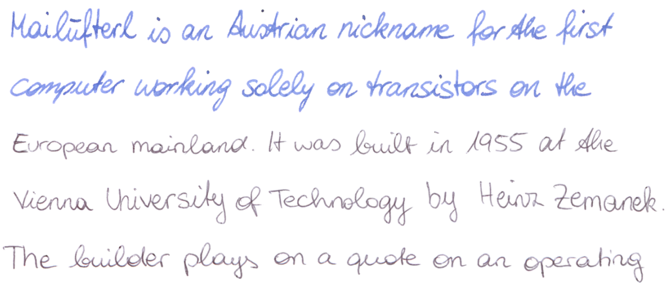



Figure 1: Sample from the CVL database showing effect of pen change on writing style.

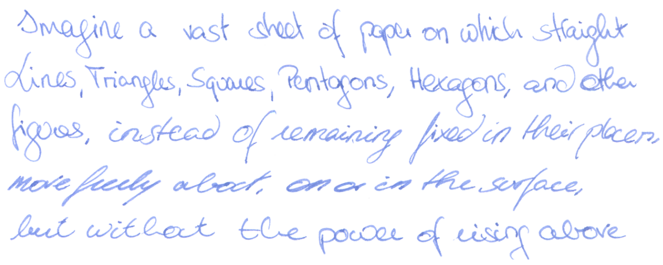



Figure 2: Sample from CVL database showing effect of writing conditions on writing style.

known writers are selected to be the most likely candidates for the unknown sample of text (Bulacu and Schomaker, 2005). Writer identification has a very wide application field such as verifying the authenticity of financial documents, wills etc and in criminal investigation where a piece of handwriting is the only piece of evidence available to the police, such as in case of ransom notes. This has made hand writer identification an active area of research which has resulted in significant progress being made recently. However, despite this achievement some challenges still exist that need to be overcome.

The writing style is unique to every person and no two people write in the exact same way. Furthermore, a person will not write a text in exactly the same way twice. These two points form the basis for writer identification research as well as the main reasons for the difficulties caused in this field, (Fiel and Sablatnig, 2013). To complicate matters even further the pen used for writing also affects the identification process as shown in Figure 1 which is a sample from the CVL (Kleber et al., 2013) database. It can be seen that the writer changed their pen mid-document and with it the writing style also got affected. Similarly, the conditions in which the text is written also affects the individual's writing style for example text written in a rush, was the writer seated or standing etc. This effect can be observed in Figure 2 (another writer from the CVL database) where the writing style has changed mid-document, it appears that the changed text was written in a rushed condition.

Various approaches for hand writer identification have been proposed in the literature including methods for the segmentation of text (Biadsy et al., 2011), feature extraction techniques (Fiel and Sablatnig, 2012; Jain and Doermann, 2014; Al-Maadeed et al., 2014), local descriptor computation such as Graphemes (Khalifa et al., 2015; Schomaker et al., 2004; Schomaker and Bulacu, 2004) and SIFT with code book generation (Wu et al., 2014). A significant effort has been devoted to develop much improved solutions in each of these topics. However, the main concern with these approaches is their inability to perform well under noisy conditions. In this paper, a robust Bagged Discrete Cosine Transform (BDCT) technique is proposed for writer identification. DCT has been used because of its robustness to noise and blurring and also because the representation of an image in the DCT domain has been shown to be effective for the purpose of image matching (Mitrea et al., 2004). The main components of our proposed system include local descriptor computation using Discrete Cosine Transform, multiple vector quantisation using bagging and clustering, structured writer representation via localised histograms of vector codes, dimensionality reduction using kernel discriminant analysis and classification using nearest centre rule. The proposed system has been evaluated on four hand written datasets including IAM (Marti and Bunke, 2002), CVL (Kleber et al., 2013), AHTID/MW (Mezghani et al., 2012) and IFN/ENIT (Pechwitz et al., 2002). The results achieved show that the system delivers comparable performance with state-of-the-art systems in the case where query documents are presented in ideal conditions on one hand. On the other hand, the system exhibits robustness against noise and blur unlike existing systems. The main contributions of this work are twofold:

1. A new BDCT approach is proposed for writer identification which avoids the problems associated with traditional DCT-based feature extrac-

tion techniques, i.e., memory limitations due to the abundance of features and undesirable similarity of local features among various writers.

2. Unlike previous automatic writer identification systems, the proposed system is only marginally affected by distortion and noise; this is mainly due to the DCT feature extractor which is known to be very robust to noise and blurring distortions. The robustness of writer identification is vital in forensic applications where the query data is collected under severe imaging conditions.

This paper is organised as follows. Section 2 reviews the current state of the art research of writer identification. Section 3 describes the proposed system of writer identification where the BDCT concept is explained including the codebook generation. Section 3 also discusses the dimensionality reduction and classification using nearest centre rule. Section 4 provides an experimental evaluation of the proposed system while Section 5 analyses it from different perspectives and discusses ways for improvement. Finally, section 6 is dedicated to the conclusions drawn from our work.

## 2. Related Work

Over the last decade's considerable advancements have been made in the field of writer identification. A detailed survey in the field of writer identification can be found in the works done by (Chen, 2012) and (Sreeraj and Idicula, 2011) for the interested readers. Due to recent increased interest in the scientific research community regarding writer identification we will present a survey of notable advancements achieved in this field which relates to our proposed system.

The task of writer identification has been tackled using various approaches. (Schlapbach and Bunke, 2004) relied on an HMM (Hidden Markov Model) based recognizer to identify the unknown images. For each writer, the authors built a single HMM recognizer using the features extracted from a shifting pixel-wide window, the sliding window which extracts 9 features in total with three global and six local. The global features included the number of black pixels in the window, the second order moment and the centre of gravity, whereas the local features extracted were the positions of the top most and lowest pixel, the fraction of black pixels between these two limits and the number of black to white transitions. Using this 9 dimensional feature vector the corresponding HMM is trained for every writer and the authors were able to achieve a 96.5% accuracy using 100 writers from the IAM database. The identification was achieved using a log-likelihood score to rank the writers. The same authors proposed a further improvement to their previous work in (Schlapbach and Bunke, 2006) where they replaced the Hidden Markov Model with a Gaussian Mixture Models (GMM). At the time GMM was used mainly in the speech recognition community but by applying the same concept to writer identification, the authors were able to achieve an improved result, when compared against their previous work an identification rate of 98.4% was achieved using 100 writers from the IAM database. Furthermore, GMM was conceptually simpler and faster to train than the HMM models. A drawback of both these systems was that they were highly dependent on perfect line segmentation to achieve the desired results as poor segmentation would greatly affect the performance of the system.

(Bulacu and Schomaker, 2007) identified the writers based on two sets of features. The first set of features were extracted at the texture level and a probability distribution function (PDF) was used to represent the features such as the slant, roundness of the writing style and the curvature. The second set of features operated at the character level and focused on information at the allograph level where the writers were characterized by a stochastic pattern generator of graphemes. The graphemes extracted from each writer are characteristic to that writer. The PDF's of these graphemes were computed using a codebook obtained by grapheme clustering. This combination of features achieved attractive results as their proposed system showed an accuracy of 89% using the full IAM database of 650 writers. (Siddiqi and Vincent, 2007) used the codebook concept and improved it by extracting the graphemes at a much smaller scale i.e. at a sub-grapheme level. The authors achieved this by using a modified form of component by component extraction for the purpose of the codebook generation. A fixed window of size

$n \times n$ ($13 \times 13$ achieved best results) was moved over the text from left to right while keeping the vertical origin fixed. Due to the small scale of the grapheme extraction the authors mentioned that to get an effective accuracy rate each writer would require a large amount of training data to familiarize the identifier. Their proposed system achieved an identification rate of 94% using 50 writers from the IAM database.

(Fiel and Sablatnig, 2012) proposed a writer identification and retrieval system based on the codebook approach but rather than generating the codebook using graphemes or textural based identifiers, the authors proposed to extract the Scale Invariant Feature Transform (SIFT) features from all writers. These features were clustered and a codebook was generated. By using SIFT, the authors avoided the binarization step hence eliminating the problems associated with binarization such as poor binarization of faded text or low contrast documents which can lead to a loss of important identifying features. Their proposed system achieved an accuracy of 91% using 650 writers of the IAM database. Later (Fiel and Sablatnig, 2013) used local SIFT features for identification where SIFT features were used to create a visual vocabulary by a clustering process using Gaussian Mixture Models (GMM). This enabled the authors to calculate the Fisher vector for each image. Finally, the classification was performed using the least distance rule. Their proposed system was applied on the CVL and ICDAR 2011 databases in which they showed top 1 results of 97.8% and 91.3% respectively.

(Schomaker and Bulacu, 2004) proposed a new feature in upper-case western handwriting called Connected-Component Contours (CO3). This feature was used to construct a universal code-book with a self-organising map. By using the codebook, a descriptor could be computed for each text image based on the occurrence histogram of its corresponding CO3. In order to enhance the identification performance, the authors have also combined CO3 with another edge-based feature describing the angle of edges in a histogram. A variant of CO3, called Fragmented Connected Component Contours (FCO3 ) was also proposed by (Schomaker et al., 2004). Interestingly, (Bulacu and Schomaker, 2005) have shown that the K-means clustering technique can be used for generating the code-book as the performance offered was very close to the one obtained with the self organising map. Recently, FCO3 (also referred to as graphemes) have also been adopted by (Khalifa et al., 2015) with a multiple codebook approach where the codebook for every writer was divided into 12 sub-codebooks. These multiple codebooks were then used to represent every writer. It was demonstrated that using multiple codebooks to represent every writer produced better results than by using a single codebook approach.

(Bertolini et al., 2013) considered the handwriting text as a texture and used Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) to extract textural features for writer verification and identification. They built upon a previously reported work using the dissimilarity framework approach by (Hanusiak et al., 2012) and extended the idea to writer identification. The concept underlying the dissimilarity framework is based on the mapping of texture vectors into dissimilarity vectors where two classes only could be constructed, i.e. positive population and negative population. The samples from both classes are then used to train a binary Support Vector Machine (SVM) classifier. Given a reference text image and a query text image, the system calculates the corresponding dissimilarity vector and uses the trained SVM to classify it (this is to verify whether the query image is from the same class as the reference image). In writer identification, the query image is compared with all images in the database to extract the corresponding dissimilarity vectors. Each dissimilarity vector is then classified with SVM. The hits found for each writer according to SVM are combined via a fusion function (sum, max, median, product, etc.) to determine the closest writer. The system was tested using two databases, the Brazilian Forensic Letter (BFL) and the IAM database (650 writers). An accuracy of 99.2% and 96.7% was achieved on the BFL and IAM database respectively.

## 3. Proposed System

This section describes our proposed system, where the aim is to develop a system which is capable of identifying an unknown handwritten image by providing a likely list of candidates from a known

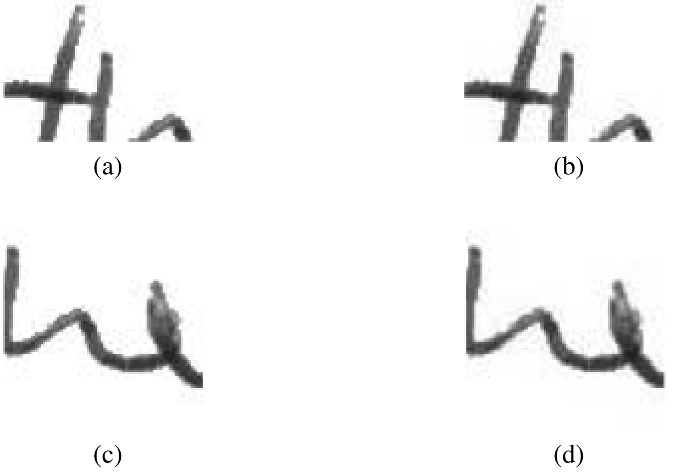



Figure 3: Reconstructed blocks with fewer DCT coefficients. (a) original block. (b) reconstruction of (a) with 2500 DCT coefficients. (c) original block. (d) reconstruction of (c) with 2500 coefficients.

database of writers with the highest degree of confidence. The feature extraction is accomplished in the DCT domain with the help of an overlapping sliding window. The DCT has been adopted in this paper for two main reasons: (i) because of its compressive nature as the DCT can represent a block of handwritten text with fewer coefficients while maintaining most of the handwriting information and (ii) the DCT coefficients are normally robust to distortions that may occur during the writing or scanning process (noise, blurring, change in contrast, etc.). Figure 3 shows two reconstructed blocks of size $64 \times 64$ with only the first 2500 coefficients in a zigzag order. As can be seen, the handwriting information can be perfectly recovered.

The DCT can be viewed as a projection of the signal onto an orthogonal basis composed of cosine functions. In addition to being a de-correlating transform, the DCT has been widely used in image compression due to its compressive nature (Sayood, 2012). The DCT transforms a block of pixels $b$ of size $N_1 \times N_2$ into a matrix of real numbers as

$$B(u,v) = \frac{2}{N_1 N_2} C(u)C(v) \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \cos\left(\frac{u\pi}{N_1}(i+0.5)\right) \cos\left(\frac{v\pi}{N_2}(j+0.5)\right) b(i,j) \quad (1)$$

where $0 \leq u \leq N_1 - 1$ and $0 \leq v \leq N_2 - 1$. $C(0) = \frac{1}{\sqrt{2}}$ and $C(\delta) = 1$ for $\delta \neq 0$.

Each image generates tens of thousands of feature vectors since the image is divided into small overlapping blocks. This huge amount of data demands a lot of resources in terms of available memory and also may cause over fitting of data. Due to these issues, it is simply not possible to build a model by using traditional DCT. To overcome this problem, random unique features are selected from every image from all writers for the generation of every predictor model. This collection of random features is clustered using a clustering algorithm. Previously three main clustering algorithms have been used for codebook generation i.e. k-means, 1D SOM (Self Organizing Map) and 2D SOM. However it was demonstrated by (Bulacu and Schomaker, 2005) that the clustering method used to generate the codebook did not affect the end result since basically the same performance was observed for all three clustering methods. This paper uses k-means for clustering of features. The clustering of these select random features from all writers allows for the creation of a universal codebook, that is, a feature vector of each sample can be generated by producing an occurrence histogram whose bins (equal to the number of centroids used during clustering) correspond to the indices of each feature to its nearest centroid. This histogram of occurrences is then normalized to get the final feature vector. The process of universal codebook generation is shown in Figure 4.

Once the universal codebook for every model has been generated, the system can be trained by extracting the descriptor for every handwriting image with respect to its own codebook. These generated feature vectors are of high dimensionality, it is therefore sensible to reduce the dimension of the feature space. Furthermore, the universal codebook generated from the DCT features suffers from the same problem as with many other local feature extractors relating to

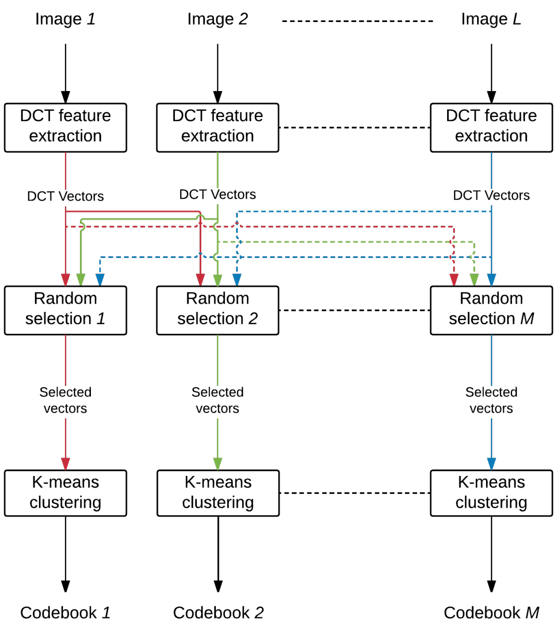



Figure 4: Multiple codebook generation with random feature selection. $L$ = number of images, $M$ = number of models, ($L >> M$)

a high intra-class variance with a long tail distribution. In order to solve this problem, kernel discriminant analysis with spectral regression(SR-KDA) is deployed for reducing the dimensionality of the feature space while decreasing the intra-class variance (Explained in Section 3.1). The SR-KDA method creates a predictor model which can be used to identify the writer from a query image. The training phase of the system is shown in Figure 5.

It is worth noting that each universal codebook is generated using a set of randomly selected feature vectors from all writers. However, a random selection of features may not always best represent a class and thus may lead to poor classification results. To overcome this problem bootstrap aggregation or bagging is used. Bagging is considered to be one of the most popular re-sampling ensemble methods. The concept of bootstrap aggregating was proposed by (Breiman, 1996) and is based on the assumption that by using multiple versions of a predictor rather than just one, a better result can be achieved by aggregating the results of those predictors.

Let's assume that a learning set $L$ consists of data $\{(y_n, x_n), n = 1 \ldots N\}$ , where $x$ is the data matrix, $y$ represents the class labels of that data matrix and $N$ represents the number of samples. From this data a predictor model, $\phi(x, L)$ can be generated and the label $y$ of an unknown image can be predicted using this model $\phi$. The same learning set $L$ can be divided into a sequence of learning sets $\{L_k\}$ each consisting of N independent observations from the same master learning set $L$. These $k$ learning sets can be used to generate $k$ predictors, $\{\phi(x, L_k)\}$. In this case each model will predict its own class label $y$ for an unknown image. The final prediction is achieved by aggregating all of the individual predictions by method of either majority voting (Kittler et al., 1998), mean or product (Tao et al., 2006).

The main concept of bagging is that the predictor models generated from the $k$ learning sets will disagree at times due to variance of the learning sets but this variance is compensated via aggregation. In the proposed approach, the variance of the learning set is obtained by the random selection of DCT features for universal codebook generation. This concept has shown to provide better results than just using a base model.

The testing phase of our system is shown in Figure 6. When an unknown image is presented to the system, its DCT features are extracted in the same manner as in the training phase. The vectors are then used to generate a descriptor (i.e. a histogram) for the test image via each codebook. That is, $M$ descriptors are obtained for each query image. Finally, each descriptor is classified by the corresponding SR-KDA predictor model. The predicted writers from all models are subjected to majority voting and the writer having the majority from all the predictor models is selected as the most probable writer for that unknown test image.

### *3.1. Kernel Discriminant Analysis with Spectral Regression (SR-KDA) for Dimensionality Reduction*

Kernel discriminant analysis is a non-linear technique of Linear Discriminant Analysis (LDA) (Cai et al., 2011; Fukunaga, 2013). In LDA, the projection vectors are acquired by decreasing the variation of the same class and at the same time, increasing the between class scatter. Equation 2 described the main goal of LDA

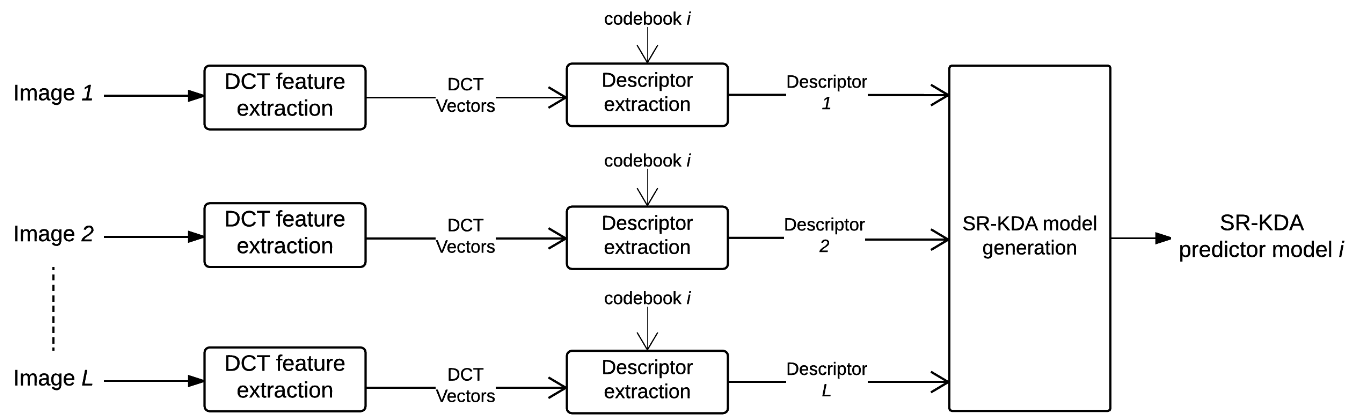



Figure 5: Training phase - SR-KDA predictor model *i* is generated for codebook *i*.

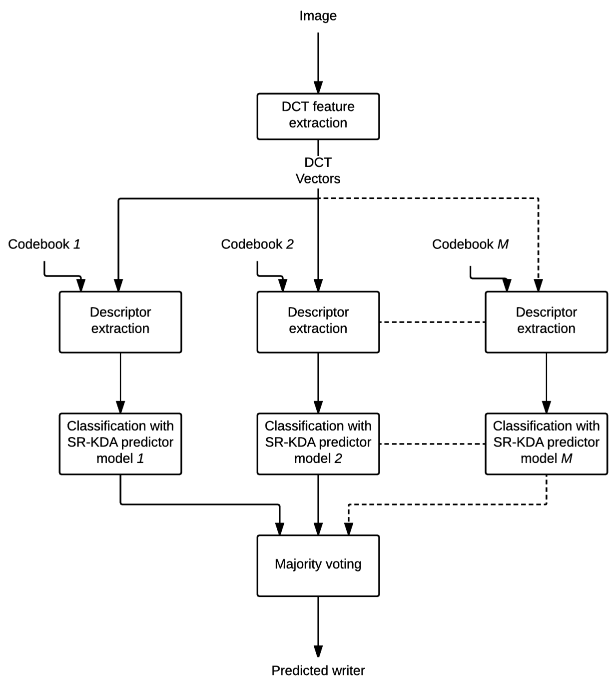



Figure 6: Testing phase of the proposed system.

$$e_{opt} = argmax \frac{e^T C_b e}{e_T C_t e} \tag{2}$$

where $C_t$ and $C_b$ indicate the within and between class scatter matrix respectively. The eigenvectors related to the non-zero eigenvalues of matrix $C_t^{-1} C_b$ are the optimal $e's$.

To extend LDA as non linear, consider kernel matrix $K$ of size $n \times n$ which is computed from the training vectors obtained using codebook generation. Let $x_j \in \mathcal{R}^d, j = 1, \cdots, k$ are the vectors of training data and $K(x_a, x_b) = \langle \wp(x_a), \wp(x_b) \rangle$. Here, $\wp(x_a)$ $\wp(x_b)$ are the embeddings of $x_a$ and $x_b$. Let us represent the projection function as $\rho$ into the kernel space. Equation 3 described the objective function of KDA

$$\max_{\rho} D(\rho) = \frac{\rho^T S_b \rho}{\rho^T S_t \rho} \tag{3}$$

where $S_t$ and $S_b$ represent the total and between class scatter matrices respectively in the feature space. Eigen-problem $C_b \rho = \lambda C_t \rho$ which is equivalent to Equation 4 as proved by (Baudat and Anouar, 2000) is then used to solve Equation 3.

$$\max_{\sigma} D(\sigma) = \frac{\sigma^T KBK\sigma}{\sigma^T KK\sigma} \tag{4}$$

where $\sigma = [\sigma_1, \sigma_2, ....., \sigma_n]^T$ is an eigenvector conform to $KBK\sigma = \lambda KK\sigma$. Every eigenvector $\sigma$ assigns a projection function $\rho$ into the feature space. $B = (B_j)_{j=1,....n}$ is a $(n \times n)$ block diagonal matrix of writers or classes.

It is shown in (Cai et al., 2011; Tahir et al., 2015) that the following two linear equations can be used to obtain the KDA projections

$$\begin{aligned} B\psi &= \lambda\psi \\ (K + \delta I)\sigma &= \psi \end{aligned} \tag{5}$$

where $\delta > 0$ is a regularization parameter, $I$ is the identity matrix, and $\psi$ is an eigenvector of $B$. Gram-Schmidt method is utilized to obtain Eigen-vectors $\psi$. Since $(K + \delta I)$ is positive definite, linear equations in 5 are solved using Cholesky Decomposition as follows

$$K^{*}\sigma = \psi \Leftrightarrow \begin{cases} R^{T}\beta = \psi \\ R\sigma = \beta \end{cases} \tag{6}$$

which initially involves finding vector $\beta$ and then solving for vector $\sigma$. Briefly,

- SR-KDA prevents the computation of eigenvector by solving regularized regression problems.
- The main advantage is its capability to handle large kernel matrices and in the significant reduction of the computational cost. The two main steps when computing SR-KDA are the response generation using Gram-Schmidt method and the use of Cholesky decomposition to solve $(c - 1)$ linear equation where $c$ is the number of writers or classes in the training data. Let "flam" be an operation consisting of one multiplication and one addition (Stewart, 1998). $(mc^2 - \frac{1}{3}c^3)$ flams is the total cost of Gram-Schmidt method (Cai et al., 2011) and $m^2c$ flams are required to solve $c - 1$ linear equations. The Cholesky decomposition needs $\frac{1}{6}m^3$ flams. Thus, the total cost of SR-KDA is $\frac{1}{6}m^3 + m^2c + mc^2 - \frac{1}{3}c^3$. This cost can easily be approximated as $\frac{1}{6}m^3 + m^2c$. If we compare this cost with ordinary KDA ($\frac{9}{2}m^3 + m^2c$), SR-KDA considerably reduces the most expensive eigenvector computation. It achieves 27 times speed-up over traditional KDA.
- After obtaining $\sigma$, test data samples are calculated from : $f(x) = \sum_{i=1}^{n} \sigma_i K(x, x_i)$ where $K(x, x_i) = \langle \wp(x), \wp(x_i) \rangle$ and the projected data can be used for prediction. In this study, nearest centroid classifier (NCC) to get the final decision from each model of SR-KDA (Cai et al., 2011).

## 4. Experiments and Results

The proposed BDCT approach was applied to four challenging databases (two English and two Arabic) to evaluate its performance. The details of these databases are summarized below.

*IAM Database*

The IAM handwriting database (Marti and Bunke, 2002) can be considered as one of the most popular English databases for the purpose of writer identification and verification. The database contains handwritten samples from 657 writers scanned at 300 DPI and saved in PNG format at 256 gray levels. Of these 301 writers produced two or more handwritten samples while the remaining 356 writers only produced a single sample. For comparable test conditions we arranged the database as described in (Bulacu and Schomaker, 2007); two samples are retained from the writers that contributed two or more than two documents, and the writers that contributed only a single image, have that image divided roughly in half. By this arrangement the database contains two handwritten samples from each of the 657 writers, one of which is used for training while the other is used for testing.

*CVL Database*

The CVL database (Kleber et al., 2013) consists of handwritten documents from 310 writers. Each writer contributed five cursively written documents, four of which are in English and one in German. The CVL database is publicly available and can be used for writer retrieval, writer identification and word spotting. Only the English documents are used in our experiments. Three documents per writer are used for training while the fourth English document is used for testing.

*AHTID/MW Database*

The Arabic Handwritten Text Images Database written by Multiple Writers (AHTID/MW) (Mezghani et al., 2012) consists of 3710 text lines and 22,896 words written by 53 native Arabic writers of different ages and educational backgrounds. The text samples are scanned in grayscale format at a resolution of 300 dpi. The writers were not restricted to the use of any one type of pen. The handwritten samples are divided into 4 sets for the purpose of training and testing. For our experiments we used 3 sets for training and the last set was used for the purpose of testing.

*IFN/ENIT Database*

The IFN/ENIT database (Pechwitz et al., 2002) is the most popular Arabic handwritten database. It consists of 26,000 handwritten Tunisian village names written by 411 writers. All the documents are scanned at 300 dpi and are saved in binary image format. The database is arranged similar to as explained in (Hannad et al., 2016) where a reduced set of samples are used per writer in order to simulate real world conditions. 30 randomly selected words per writer are used for training while 20 words per writer are used for testing.

### *4.1. Extraction of DCT features*

Since the DCT is a frequency transform that characterises the significance of changes across adjacent pixels, it is more sensible to use digital documents in the form of grey level images rather than binary images in order to capture as much frequency information as possible. Moreover, to ensure that we do not get an overwhelming return of blocks containing only white spaces the images are first segmented to return all the connected components of that image with respect to a set threshold. The thresholding ensures that the object such as diacritics and accidental ink traces can be ignored. These connected components are then divided into overlapping sliding blocks of size $n \times n$. The block size should be large enough to contain an acceptable amount of information about the writer and also small enough to ensure acceptable identification (Séropian, 2003). The optimum block size was determined empirically and the results achieved with different block sizes are shown in Figure 7.

It can be seen that 32×32 block size produced the best results for the CVL, IFN/ENIT and AHTID/MW databases whereas a block size

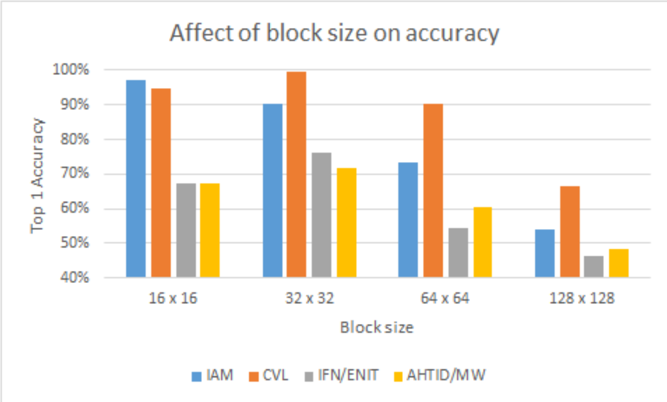



Figure 7: Comparison of Top 1 accuracy based on block size.

of 16×16 produced the best results for the IAM database. For each block, DCT features were extracted and saved using a zig-zag scan pattern as described by (Robinson and Kecman, 2003). This zig-zag extraction (shown in Figure 8) allows for converting the 2-D DCT matrix into a 1-D vector of size 1024 (for our 32 x 32 block size). The magnitude of the coefficients decreases as we travel down the vector.

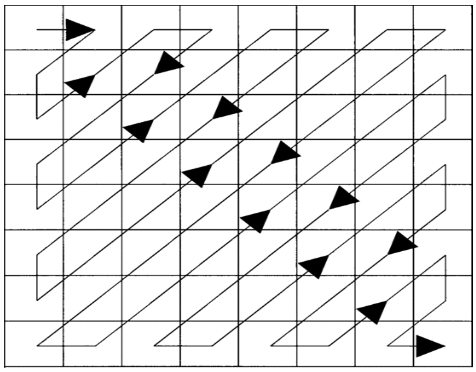


Figure 8: Zig-zag extraction of the DCT coefficients.

This process of dividing an input image into connected components and then further dividing these components into blocks of size $n \times n$ is shown in Figure 9.

Recall from Section 3 that a random selection of DCT feature vectors is performed to create a number of codebooks. The codebook size (i.e. the number of centroids used in clustering) affects the accuracy achieved and the optimum codebook size for our system was determined through experimentation. The codebook size should be sufficiently large to represent the variability in the feature space but

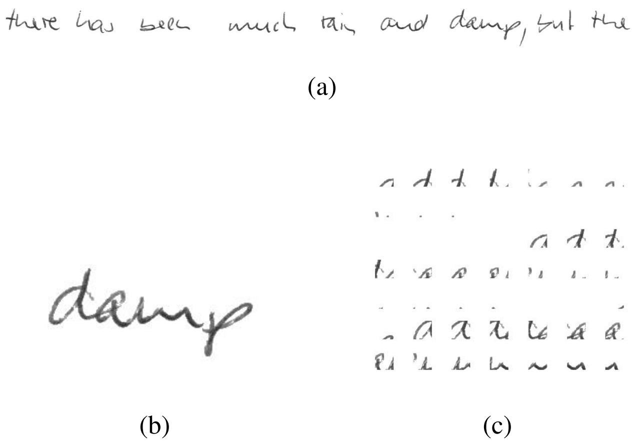



Figure 9: (a)An image sample from the IAM database (b) One of the words extracted from the line (c) The word divided into overlapping blocks of size 32 x 32.

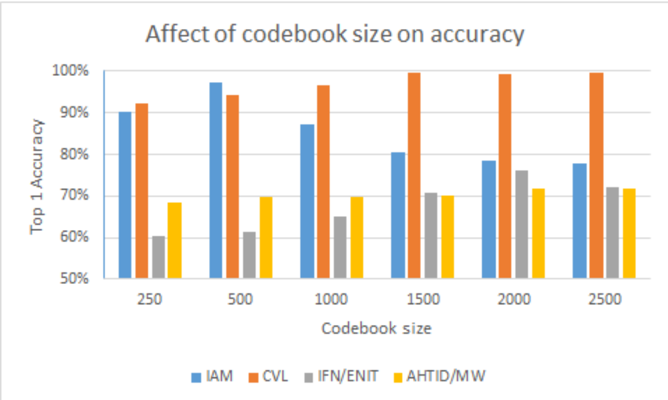



Figure 10: Comparison of the Top 1 accuracy based on codebook size.

on the other hand it should not cause over fitting. The different codebook sizes tested and their accuracy achieved is shown in Figure 10. The codebook size of 1500 proved to be the best for the CVL, for the Arabic AHTID/MW and IFN/ENIT databases a codebook size of 2000 worked best whereas for the IAM database best results were achieved with a codebook size of 500. IAM database performing better with a smaller codebook size can be related to the small amount of data available per writer, as by using the modified version of the database each writer is left with a very limited amount of text. Note that the descriptors (i.e. histogram vectors) of all writers obtained using a codebook are subjected to dimensionality reduction via SR-KDA.

The main improvement in our proposed system lies in the generation of multiple SR-KDA predictor models for every writer which in turn were generated from universal codebooks of random selection of DCT features. Since our system relies on a majority voting rule from all predictor models to generate the final result it would only make sense that the higher the number of models the more consistent the result would be. Since every predictor depends on randomly selected features, there exist models which are generated using features that may not completely represent the writer. Therefore, although more models would theoretically produce better results there must exist a point beyond which using more predictors do not bring any significant gain. This level needed to be determined as generating a model is computationally expensive. Our proposed system was tested with models, $\phi = 5, 7, 10, 12, 14, 16, 18, 20, 22, 25, 30$ to determine the optimal number of models needed. Figures 11, 12, 13 and 14 show the accuracy achieved from the majority voting of the various models generated for the IAM, CVL, AHTID/MW and the IFN/ENIT databases respectively. A steady increase in accuracy can be observed up until 20 models. After which the results show that further increasing the models after 20 has no significant effect on the performance of the overall system.

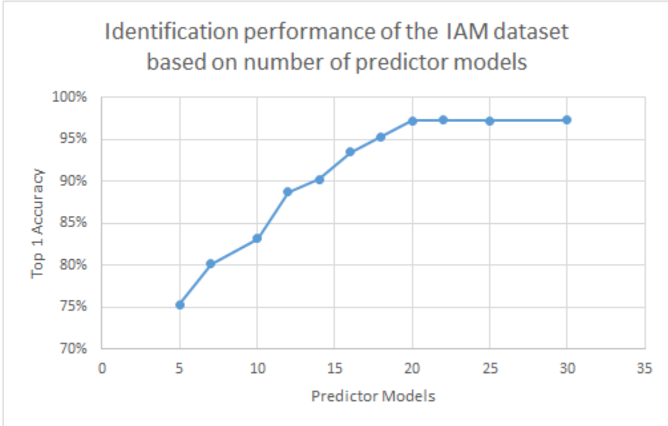



Figure 11: Comparison of the Top 1 accuracy of the IAM database based on number of predictor models.

### 4.2. *Comparison of our proposed system with existing work*

A comparative study was performed in order to compare the performance of the proposed system with the state of the art techniques already published

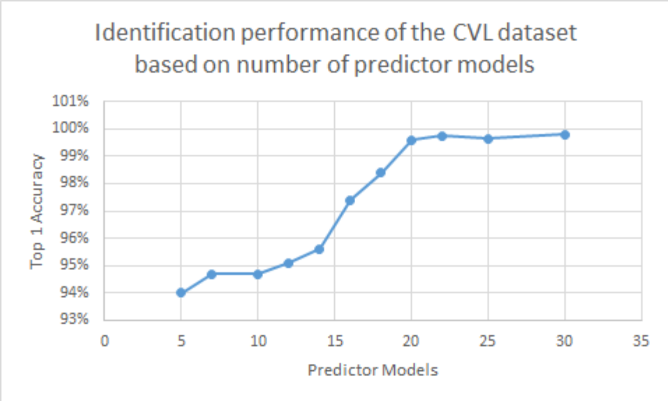



Figure 12: Comparison of the Top 1 accuracy of the CVL database based on number of predictor models.

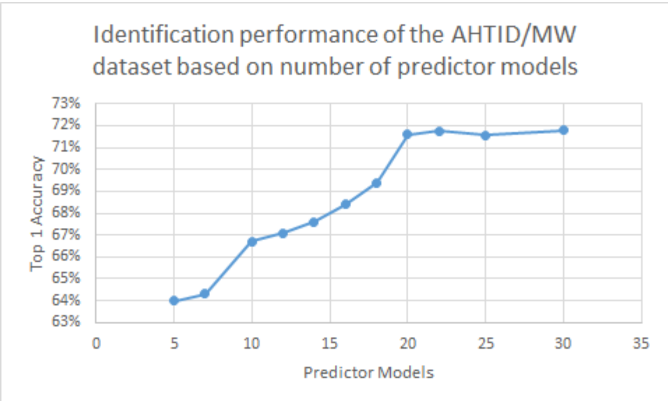



Figure 13: Comparison of the Top 1 accuracy of the AHTID/MW database based on number of predictor models.

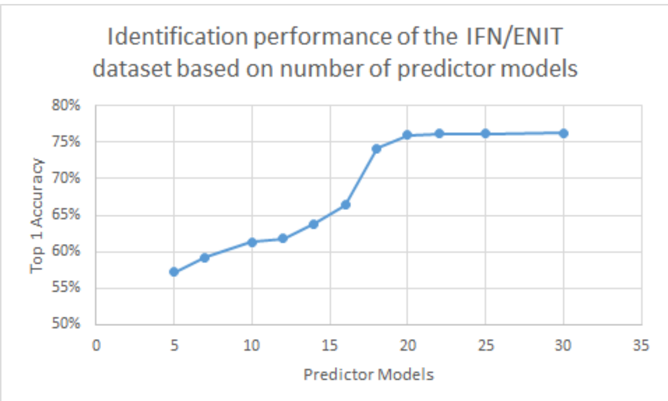



Figure 14: Comparison of the Top 1 accuracy of the IFN/ENIT database based on number of predictor models.

in the field of writer identification. As discussed earlier, experiments were performed on the IAM[1], CVL[2], AHTID/MW[3] and IFN/ENIT[4] databases. The arrangement of these databases have been explained in Section 4. Using our proposed BDCT approach, a Top 1 accuracy of 97.2% on the IAM database has been achieved, which outperforms the nearest best performing system of (Bertolini et al., 2013) by 0.5%. For the CVL database, 99.6% of Top 1 accuracy has been reached by our system. This outperforms by 0.2% the nearest best system developed by (Jain and Doermann, 2014). For the AHTID/MW database, 71.6% of Top 1 accuracy has been obtained with the proposed system which is still comparable to the state of the art, outperformed only by (Hannad et al., 2016). For the IFN/ENIT database, however, the system shows a clear drop in performance. Note that the images of this dataset are given in binary form and the system seems to be severely affected by this type of images when compared to existing techniques. This was expected since the DCT features describe the frequency content of images (see Section 4.1). In fact, because binary images carry extremely little frequency information, the documents written by different writers would have similar frequency contents if they were represented in binary form, i.e., the inter-class similarity increases drastically due to binarization.

### 4.3. *Robustness of the proposed system*

In practice, the handwritten samples under investigation are not always presented to the forensic analyst in ideal conditions. The samples could be noisy or blurry due to the imaging conditions under which they have been collected. It is imperative that identification algorithms are robust enough to ignore such distortions.

To demonstrate the robustness of the proposed system, the AHTID/MW database and 100 randomly selected writers from the IAM database were subjected to two types of distortion; blurring with a low pass

[1]http://www.iam.unibe.ch/fki/databases/iam-handwriting-database

[2]http://www.caa.tuwien.ac.at/cvl/category/research/cvl-databases/

[3]http://ieeexplore.ieee.org/document/6424426/

[4]http://www.ifnenit.com/

| System | Number of writers | Top 1 Accuracy |
|---|---|---|
| Bulacu and Schomaker (2007) | 650 | 89.0% |
| Siddiqi and Vincent (2010) | 650 | 91.0% |
| Kumar et al. (2014) | 650 | 88.4% |
| Ghiasi and Safabakhsh (2013) | 650 | 93.7% |
| Bertolini et al. (2013) | 650 | 96.7% |
| Khalifa et al. (2015) | 650 | 92.0% |
| Jain and Doermann (2014) | 657 | 94.7% |
| Hannad et al. (2016) | 657 | 89.5% |
| Schomaker and Bulacu (2004) | 657 | 82.5 % |
| **Proposed system** | **650** | **97.2%** |

Table 1: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the IAM database.

| System | Number of writers | Top 1 Accuracy |
|---|---|---|
| Fiel and Sablatnig (2013) | 309 | 97.8% |
| Jain and Doermann (2014) | 310 | 99.4% |
| Christlein et al. (2014) | 310 | 99.2% |
| Fiel and Sablatnig (2015) | 309 | 98.9% |
| Schomaker and Bulacu (2004) | 310 | 81.8% |
| Hannad et al. (2016) | 310 | 96.2% |
| **Proposed system** | **310** | **99.6%** |

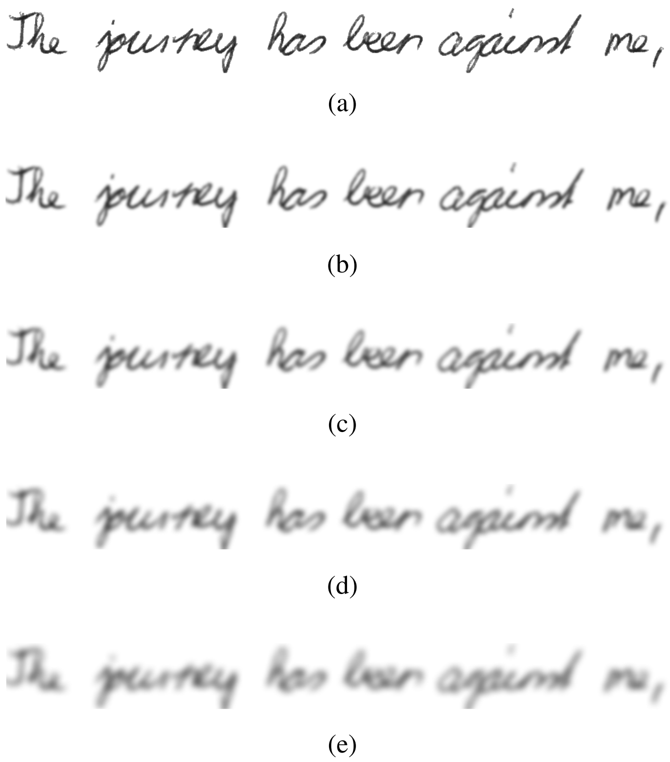
Table 2: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the CVL database.

| System | Number of writers | Top 1 Accuracy |
|---|---|---|
| Slimane and Margner (2014) | 53 | 69.4% |
| Schomaker and Bulacu (2004) | 53 | 66.4% |
| **Hannad et al. (2016)** | **53** | **77.3%** |
| Proposed system | 53 | 71.6% |

Table 3: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the AHTID/MW database.

| System | Number of writers | Top 1 Accuracy |
|---|---|---|
| Bulacu and Schomaker (2007) | 350 | 88.0% |
| Abdi and Khemakhem (2015) | 411 | 90.0% |
| **Hannad et al. (2016)** | **411** | **94.9%** |
| Proposed system | 411 | 76.0% |

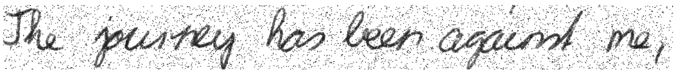
Table 4: Accuracy comparison of the proposed system with the state of the art systems in writer identification for the IFN/ENIT database.



(a) (b) (c) (d) (e)

Figure 15: Gaussian blurring applied to a sample of text from the IAM database. (a) Original image. (b) Gaussian filter with standard deviation of 2. (c) Gaussian filter with standard deviation of 3. (d) Gaussian filter with standard deviation of 4. (e) Gaussian filter with standard deviation of 5.

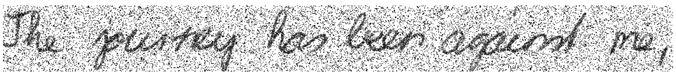
(a)



(b)

(c)

(d)



(e)

Figure 16: Salt & pepper noise applied to a sample of text from the IAM database. (a) Original image. (b) Salt & pepper with a noise density of 0.1. (c) Salt & pepper with a noise density of 0.2. (d) Salt & pepper with a noise density of 0.25. (e) Salt & pepper with a noise density of 0.3.

Gaussian filter and "salt & pepper" noise. This noise was applied at incrementally increasing levels. The application of blurring and "salt & pepper" noise to samples of the IAM database can be seen in Figure 15 and Figure 16 respectively. These noisy versions of the database were used to record the Top 1 accuracy of the proposed system along with two other systems previously published in literature i.e. the systems proposed by (Schomaker and Bulacu, 2004) and (Hannad et al., 2016). Furthermore, a variation of our proposed system was also applied on the noisy databases, where SIFT was used for feature extraction in place of DCT. SIFT is the preferred feature extractor for purposes related to object detection in images which has also been widely used for the purpose of writer identification (Fiel and Sablatnig, 2012, 2013; Wu et al., 2014; Xiong et al., 2015). To verify our implementation of the systems used in our experimental comparison, i.e. (Schomaker and Bulacu, 2004) and (Hannad et al., 2016), the same datasets, adopted in the original papers, have been used with similar settings. That is, (Schomaker and Bulacu,

| System | Database Used | Accuracy Reported | Our Implementation |
|---|---|---|---|
| Schomaker and Bulacu (2004) | Firemaker | 94.0% | 92.3% |
| Hannad et al. (2016) | IAM | 89.5% | 88.7% |

Table 5: Comparison of reported accuracy of published works against our implementation of the same.

2004) applied their system on the Firemaker database (Schomaker and Vuurpijl, 2000). Only the uppercase handwriting samples from 150 different writers were considered in their study. A codebook was generated from the samples of 100 writers while the samples of another set of 150 writers were used for evaluation by splitting each document in half. The top half of each full document was used as the reference document whereas the bottom half was used as the query document. In (Hannad et al., 2016), the authors applied their system on the full IAM database while retaining a maximum of 14 text lines per writer. For each writer 60% of the text lines were used as a reference while the remaining 40% were used for evaluation. The results are depicted in Table 5.

As can be seen, the obtained identification results are very close to those reported in the original works. The minor difference, which is less than 2%, is probably due to some tiny variations in experiments which are beyond our control such as differences in segmented lines/connected components and the writers and/or paragraphs used for training and testing purposes.

For a system which is robust against image distortions, the identification results achieved on the distorted query documents must not differ significantly from those achieved on original documents. For this reason, noise was applied at incrementally increasing levels to the IAM and AHTID/MW databases and the drop in performance was observed. Table 6 and Figure 17 show the drop in performance on the IAM database for blurring when compared to the noiseless results. Table 7 and Figure 18 show the drop in performance on the IAM database for salt & pepper noise when compared to the noiseless results. Likewise, Table 8 and Figure 19 show the decrease in performance on the AHTID/MW dataset, where the blurring operation is considered on query documents,

| System | Standard Deviation | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| **Proposed system** | **1.2%** | **2.4%** | **2.5%** | **2.7%** | **3.1%** |
| Our implementation of Hannad et al. (2016) | 1.1% | 2.2% | 5.4% | 6.5% | 7.6% |
| Our implementation of Schomaker and Bulacu (2004) | 15.3% | 23.3% | 37.8% | 56.4% | 57.6% |
| Proposed system with SIFT | 1.9% | 3.6% | 6.2% | 9.6% | 13.7% |

Table 6: Drop in Top 1 accuracy observed for the IAM database subjected to Gaussian blurring when compared to the results achieved with the noiseless version of the database.

| System | Noise Density | | | | |
|---|---|---|---|---|---|
| | 0.05 | 0.1 | 0.2 | 0.25 | 0.3 |
| **Proposed system** | **0.2%** | **0.7%** | **2.7%** | **4.3%** | **10.1%** |
| Our implementation of Hannad et al. (2016) | 38.1% | 62.0% | 75.2% | 84.3% | 89.0% |
| Our implementation of Schomaker and Bulacu (2004) | 45.8% | 62.9% | 86.8% | 89.7% | 90.0% |
| Proposed system with SIFT | 4.5% | 5.8% | 10.5% | 17.1% | 21.1% |

Table 7: Drop in Top 1 accuracy observed for the IAM database subjected to salt & pepper noise when compared to the results achieved with the noiseless version of the database.

when compared to the results obtained on original documents. The drop in performance is also illustrated by Table 9 and Figure 20 on the AHTID/MW dataset where the query documents are affected by the salt & pepper noise. As can be seen, the proposed system shows a slight decrease in performance for all the tested noisy and blurry documents, whereas the competing systems suffer from massive performance drops and can no longer operate effectively in such conditions.

It can be seen that that the proposed system exhibits robustness against both types of distortion of various intensities and outperforms the competing systems including the SIFT-based variation of our system. This also illustrates the efficiency and suitability of DCT features in our system for forensic applications. The “salt & pepper” noise proved to be a more challenging task for all the systems but at ev-

| System | Standard Deviation | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| **Proposed system** | **1.8%** | **4.2%** | **6.1%** | **7.5%** | **9.5%** |
| Our implementation of Hannad et al. (2016) | 2.5% | 4.9% | 9.7% | 14.6% | 17.1% |
| Our implementation of Schomaker and Bulacu (2004) | 3.5% | 6.3% | 12.0% | 43.2% | 48.9% |
| Proposed system with SIFT | 2.5% | 5.2% | 8.3% | 16.5% | 22.0% |

Table 8: Drop in Top 1 accuracy observed for the AHTID/MW database subjected to Gaussian blurring when compared to the results achieved with the noiseless version of the database.

| System | Noise Density | | | | |
|---|---|---|---|---|---|
| | 0.05 | 0.1 | 0.2 | 0.25 | 0.3 |
| **Proposed system** | **7.7%** | **10.5%** | **17.6%** | **22.3%** | **25.7%** |
| Our implementation of Hannad et al. (2016) | 47.2% | 51.5% | 63.3% | 74.4% | 91.7% |
| Our implementation of Schomaker and Bulacu (2004) | 43.2% | 67.9% | 77.3% | 85.2% | 94.7% |
| Proposed system with SIFT | 4.1% | 11.8% | 31.1% | 40.4% | 55.3% |

Table 9: Drop in Top 1 accuracy observed for the AHTID/MW database subjected to salt & pepper noise when compared to the results achieved with the noiseless version of the database.

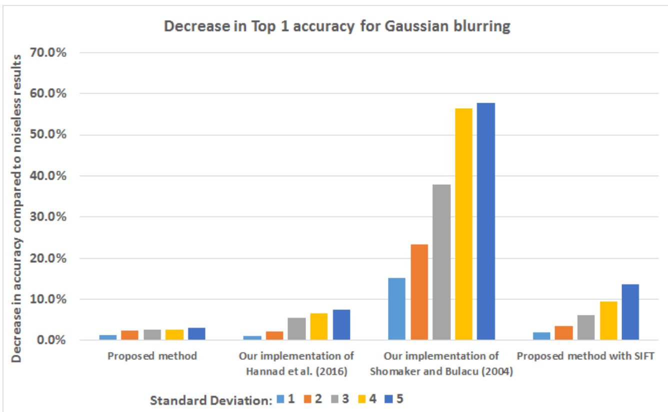



Figure 17: Comparison of the drop in accuracy observed for the IAM database subjected to Gaussian blurring with incrementally increasing standard deviation.

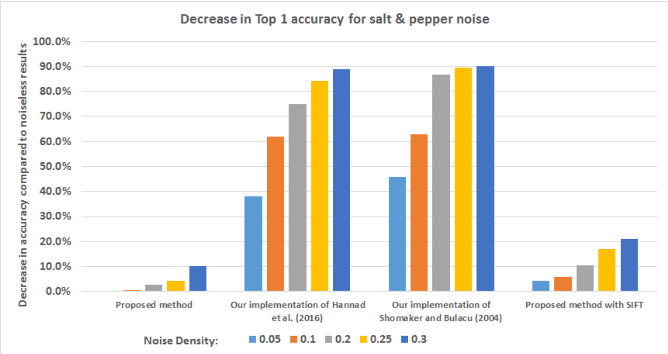



Figure 18: Comparison of the drop in accuracy observed for the IAM database subjected to salt & pepper noise with incrementally increasing noise density.

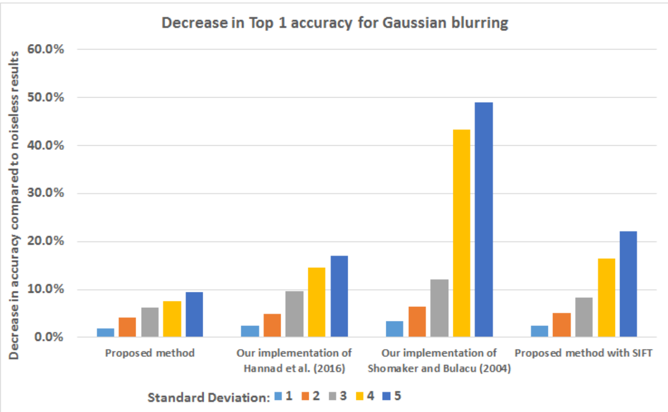



Figure 19: Comparison of the drop in accuracy observed for the AHTID/MW database subjected to Gaussian blurring with incrementally increasing standard deviation.

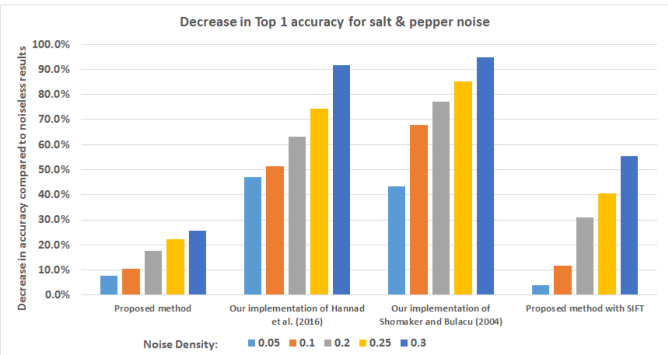



Figure 20: Comparison of the drop in accuracy observed for the AHTID/MW database subjected to salt & pepper noise with incrementally increasing noise density.

ery level of intensity the proposed system achieved more than acceptable results.

## 5. Discussion

The proposed BDCT approach compares well with state-of-the-art hand writing identification systems when the query documents are presented at a reasonably good visual quality. However, while existing systems fail to maintain acceptable performance when the query documents are subjected to noise and blurring, the proposed BDCT system shows significant improvements. Robustness is one of the main strengths of the proposed system. However, as mentioned earlier, the nature of the features used, i.e., DCT-based, suggest that the system cannot perform well on documents presented in binary form. This is a weakness that can be addressed in future by combining the DCT features with other local features that capture shape rather than the frequency content. Also, because SR-KDA uses all training samples to optimize the parameters of the feature mapping function, adding a new entry (writer) to the database in practice would require a new estimation of the parameters and this may be computationally expensive, especially, when the number of existing writers in the database is significantly large. Furthermore, by analysing the results obtained for all four databases, it is clear that the Arabic script results are not as good as that of the Latin script. Therefore it appears that automatic handwriter identification in Arabic script is more challenging than Latin script.

## 6. Conclusion

In this paper, a robust system for offline text independent writer identification has been proposed. The concept of universal codebooks has been used with bagged DCT features. Multiple SR-KDA predictor models have been generated for each writer and a majority voting rule is used to make the final decision on an unknown query document. Our proposed BDCT approach allows DCT features, that have been extracted from overlapping blocks, to be effectively used for automatic hand writer identification. It also allows us to avoid the problems associated with DCT features extracted at such a small scale i.e. memory limitations due to abundance of features and similar local features among various writers. The proposed system exploits the robustness property of the DCT features in hand writer identification. Experiments performed on noisy and blurry versions of query documents taken from two different datasets demonstrate a clear superiority of the proposed system over state-of-the-art techniques in noisy and blurry conditions.


### Acknowledgement

This publication was made possible by NPRP grant number NPRP NPRP7 442 1 082 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.


### References


Abdi, M. N. and Khemakhem, M. (2015). A model-based approach to offline text-independent arabic writer identification and verification. *Pattern Recognition*, 48(5):1890–1903.

Al-Maadeed, S. (2012). Text-dependent writer identification for arabic handwriting. *Journal of Electrical and Computer Engineering*, 2012:13.

Al-Maadeed, S., Hassaine, A., Bouridane, A., and Tahir, M. A. (2014). Novel geometric features for off-line writer identification. *Pattern Analysis and Applications*, pages 1–10.

Baudat, G. and Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404.

Bertolini, D., Oliveira, L. S., Justino, E., and Sabourin, R. (2013). Texture-based descriptors for writer identification and verification. *Expert Systems with Applications*, 40(6):2069–2080.

Biadsy, F., Saabni, R., and El-Sana, J. (2011). Segmentation-free online arabic handwriting recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(07):1009–1033.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Bulacu, M. and Schomaker, L. (2005). A comparison of clustering methods for writer identification and verification. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 1275–1279. IEEE.

Bulacu, M. and Schomaker, L. (2007). Text-independent writer identification and verification using textural and allographic features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(4):701–717.

Cai, D., He, X., and Han, J. (2011). Speed up kernel discriminant analysis. *The VLDB Journal*, 20(1):21–33.

Chen, J. (2012). Handwritten biometric systems and their robustness evaluation: a survey. Technical report.

Christlein, V., Bernecker, D., Honig, F., and Angelopoulou, E. (2014). Writer identification and verification using gmm supervectors. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 998–1005. IEEE.

Fiel, S. and Sablatnig, R. (2012). Writer retrieval and writer identification using local features. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 145–149. IEEE.

Fiel, S. and Sablatnig, R. (2013). Writer identification and writer retrieval using the fisher vector on visual vocabularies. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 545–549. IEEE.

Fiel, S. and Sablatnig, R. (2015). Writer identification and retrieval using a convolutional neural network. In *Computer Analysis of Images and Patterns*, pages 26–37. Springer.

Franke, K. and Köppen, M. (2001). A computer-based system to support forensic studies on handwritten documents. *International Journal on Document Analysis and Recognition*, 3(4):218–231.

Fukunaga, K. (2013). *Introduction to statistical pattern recognition.* Academic press.

Ghiasi, G. and Safabakhsh, R. (2013). Offline text-independent writer identification using codebook and efficient code extraction methods. *Image and Vision Computing*, 31(5):379–391.

Hannad, Y., Siddiqi, I., and El Kettani, M. E. Y. (2016). Writer identification using texture descriptors of handwritten fragments. *Expert Systems with Applications*, 47:14–22.

Hanusiak, R. K., Oliveira, L. S., Justino, E., and Sabourin, R. (2012). Writer verification using texture-based features. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(3):213–226.

Jain, R. and Doermann, D. (2014). Combining local features for offline writer identification. In *2014 14th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE*, pages 583–588.

Khalifa, E., Al-maadeed, S., Tahir, M., Bouridane, A., and Jamshed, A. (2015). Off-line writer identification using an ensemble of grapheme codebook features. *Pattern Recognition Letters*, 59:18–25.

Kittler, J., Hatef, M., Duin, R. P., and Matas, J. (1998). On combining classifiers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(3):226–239.

Kleber, F., Fiel, S., Diem, M., and Sablatnig, R. (2013). Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 560–564. IEEE.

Kumar, R., Chanda, B., and Sharma, J. (2014). A novel sparse model based forensic writer identification. *Pattern Recognition Letters*, 35:105–112.

Marti, U.-V. and Bunke, H. (2002). The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46.

Mezghani, A., Kanoun, S., Khemakhem, M., and Abed, H. E. (2012). A database for arabic handwritten text image recognition and writer identification. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 399–402. IEEE.

Mitrea, M., Prêteux, F., Vlad, A., and Fetita, C. (2004). The 2d-dct coefficient statistical behaviour: a comparative analysis on different types of image sequences. *Journal of Optoelectronics and Advanced Materials*, 6(1):95–102.

Pechwitz, M., Maddouri, S. S., Märgner, V., Ellouze, N., Amiri, H., et al. (2002). Ifn/enit-database of handwritten arabic words. In *Proc. of CIFED*, volume 2, pages 127–136. Citeseer.

Robinson, J. and Kecman, V. (2003). Combining support vector machine learning with the discrete cosine transform in image compression. *Neural Networks, IEEE Transactions on*, 14(4):950–958.

Sayood, K. (2012). *Introduction to Data Compression*. Morgan Kaufmann.

Schlapbach, A. and Bunke (2006). Off-linewriter identification using gaussian mixture models. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 992–995. IEEE.

Schlapbach, A. and Bunke, H. (2004). Using hmm based recognizers for writer identification and verification. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 167–172. IEEE.

Schomaker, L. and Bulacu, M. (2004). Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):787–798.

Schomaker, L., Bulacu, M., and Franke, K. (2004). Automatic writer identification using fragmented connected-component contours. In *Frontiers in Handwriting Recognition, 2004. IWFHR-9 2004. Ninth International Workshop on*, pages 185–190. IEEE.

Schomaker, L. and Vuurpijl, L. (2000). Forensic writer identification: A benchmark data set and a comparison of two systems [internal report for the Netherlands Forensic Institute]. pages –.

Séropian, A. (2003). *Analyse de document et identification de scripteurs*. PhD thesis, Université de Toulon.

Siddiqi, I. and Vincent, N. (2010). Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. *Pattern Recognition*, 43(11):3853–3865.

Siddiqi, I. A. and Vincent, N. (2007). Writer identification in handwritten documents. In *Document*

*Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 108–112. IEEE.

Slimane, F. and Margner, V. (2014). A new text-independent gmm writer identification system applied to arabic handwriting. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 708–713. IEEE.

Sreeraj, M. and Idicula, S. M. (2011). A survey on writer identification schemes. *International Journal of Computer Applications*, 26(2):23–33.

Stewart, G. (1998). Matrix algorithms. volume i: Basic decompositions. society for industrial and applied mathematics, philadelphia, pa, usa, 1998. Technical report.

Tahir, M. A., Kittler, J., and Bouridane, A. (2015). Multi-label classification using stacked spectral kernel discriminant analysis. *Neurocomputing*, 171:127–137.

Tao, D., Tang, X., Li, X., and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(7):1088–1099.

Wu, X., Tang, Y., and Bu, W. (2014). Offline text-independent writer identification based on scale invariant feature transform. *Information Forensics and Security, IEEE Transactions on*, 9(3):526–536.

Xiong, Y.-J., Wen, Y., Wang, P. S., and Lu, Y. (2015). Text-independent writer identification using sift descriptor and contour-directional feature. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 91–95. IEEE.