Word-Level Thirteen Official Indic Languages Database for Script Identification in Multi-script Documents

Sk Md Obaidullah[1(✉)], K.C. Santosh[2], Chayan Halder[3], Nibaran Das[4], and Kaushik Roy[3]

[1] Department of Computer Science and Engineering, Aliah University Kolkata, West Bengal, India
sk.obaidullah@gmail.com
[2] Department of Computer Science, The University of South Dakota, Vermillion, SD 57069, USA
santosh.kc@usd.edu
[3] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
chayan.halderz@gmail.com, kaushik.mrg@gmail.com
[4] Department of Computer Science, West Bengal State University, Kolkata, India
nibaran@gmail.com

Abstract. Without a publicly available database, we cannot advance research nor can we make a fair comparison with the state-of-the-art methods. To bridge this gap, we present a database of eleven Indic scripts from thirteen official languages for the purpose of script identification in multi-script document images. Our database is composed of 39K words that are equally distributed (i.e., 3K words per language). At the same time, we also study three different pertinent features: spatial energy (SE), wavelet energy (WE) and the Radon transform (RT), including their possible combinations, by using three different classifiers: multilayer perceptron (MLP), fuzzy unordered rule induction algorithm (FURIA) and random forest (RF). In our test, using all features, MLP is found to be the best performer showing the bi-script accuracy of 99.24% (keeping Roman common), 98.38% (keeping Devanagari common) and tri-script accuracy of 98.19% (keeping both Devanagari and Roman common).

Keywords: Multi-script documents · Official indic script database · Script identification

1 Introduction

Researches on multi-script document processing have real impact for a country like India, where 23 different languages (including English) and 13 different scripts (including Roman) exist. In general, OCRs are script specific, and processing documents having more than one scripts is not easy. Therefore,

K.C. Santosh et al. (Eds.): RTIP2R 2016, CCIS 709, pp. 16–27, 2017.
DOI: 10.1007/978-981-10-4859-3_2

one of the common/suggested solutions is to develop a script identification system (SIS), so that we can take it as a precursor to the specific OCR. To highlight this issue, in this paper, we present a database that is composed of 13 different languages under 11 different scripts (having fairly large amount words in it) for an automatic script identification in multi-script documents.

Until today, few works have been reported on Indic script identification. Pati et al. [1] reported 11 different scripts in their study, which is found to be the maximum number of scripts in the literature. They used a database from 11 different languages, where two languages: Kashmiri and Dogri originating from Northern part of India were not considered. To represent the scripts, Gabor filter and directional cosine transform (DCT) based frequency domain techniques were used. Based on these features, their reported performances are 98% for bi-Script and tri-Script, and 89% for eleven-scripts by using three different classifiers: nearest neighbor, linear discriminative and support vector machine (SVM). Since then, this can be considered as a benchmark work on printed script identification (PSI) at word level. Among other available popular PSI works on Indic and Non-Indic scripts, Hochberg et al. [2] proposed a technique to identify six different scripts: Arabic, Armenian, Devanagari, Chinese, Cyrillic, and Burmese, using some textual features. Pal et al. [3] proposed a line level script identification technique considering five different scripts: Bangla, Devnagari, Chinese, Arabic and Roman. Jahawar et al. [4] proposed a headline and contextual information based technique to identify Devnagari and Telugu scripts using principal component analysis (PCA) and SVM. Chanda et al. [5] proposed a word level script identification technique considering six different scripts: Bangla, Devnagari, Roman, Malayalam, Gujarati and Telugu. Joshi et al. [6] proposed a Gabor energy based paragraph level technique to identify ten different Indic scripts using k-nearest neighbor (k-NN) classifier. Dhanya et al. [7] proposed a script identification technique using Gabor filter based directional feature and SVM classifier considering Tamil and Roman scripts. Chaudhury et al. [8] proposed script identification techniques by combining trainable classifiers for six different scripts: Devnagari, Telugu, Roman, Malayalam, Bangla and Urdu. In the script identification review paper [9], authors pointed out the unavailability issue of benchmark works by considering all official Indic scripts. Following this review, we are, indeed, motivated to publish a benchmark database and results considering all 13 official Indic scripts.

The remainder of the paper is organized as follows. In Sect. 2, we explain our database. We then describe our method in Sect. 3. It includes pre-processing, feature extraction, and script identification. In Sect. 4, we provide experimental test results and analysis. We conclude the paper in Sect. 5.

2 Database

As shown in Fig. 1 our database of thirteen different official Indic languages: (1) Bangla (BEN), (2) Devnagari (DEV), (3) Dogri (DOG), (4) Gujarati (GUJ), (5) Gurumukhi (GUR), (6) Kannada (KAN), (7) Kashmiri (KAS), (8) Malayalam (MAL), (9) Oriya (ORY), (10) Roman (ROM), (11) Tamil (TAM), (12)

Fig. 1. Sample word images of 13 different official Indic languages, i.e. 11 different scripts

Telugu (TEL), (13) Urdu (URD) with 3K words per languages. Altogether, we have collected 39K words. The sources of data collection were newspaper, articles and books. For example, Bangla words were collected from scanned copy of different Tagores books, novels, poems and newspaper. As a consequence, the collected samples vary writing style, thickness of the characters and resolution. Document image scanning was carried out using HP flatbed scanner, resolution 300 dpi and stored at 8-bit gray level jpeg format. The word dimension is found in the range of 150×50 pixels. Note that the word images are extracted by an automated process, as explained in [10,11].

The database is created for public use but, limited to research purpose. A part of the database is available on-line, and will be provided upon the request.

3 Script Identification

Our study is not an exception, we start with pre-processing, and then extract features for script identification purpose. In our study, we study three different features: (1) spatial energy (SE), (2) wavelet energy (WE) and (3) the Radon transform (RT), including their possible combinations, by using three different classifiers: (1) multilayer perceptron (MLP), (2) fuzzy unordered rule induction algorithm (FURIA) and (3) random forest (RF). Again, Our idea is not only to check what features but also to check what classifiers can consistently provide optimal performance.

3.1 Pre-processing

The word images are binarized by using the following steps. (1) In grayscale word image, region-of-interests (ROIs) are generated using a local window-based algorithm. Run length smoothing algorithm is applied to overcome the presence of stray/hollow regions generated due to window size. Connected component labelling is applied and the ROIs are mapped to the original grayscale images. (2) A global thresholding technique is then applied on ROIs.

3.2 Features

As said before, we propose to study three different features: spatial energy (SE), wavelet energy (WE) and the Radon transform (RT).

Spatial Energy (SE). SE distribution varies in accordance with the change in textural information, and therefore, it is important in our study. SE distribution was observed by computing entropy on the grayscale images. It can be represented by

$$Entropy = -\sum p(i,j)log(p(i,j)).$$

In general, entropy is complement of energy. Therefore, for any non-uniform or aperiodic gray level distribution, there exists high entropy.

Another measure is the standard deviation of binary images of different scripts. Standard deviation is a measure of the variability of the image pixels. It can be represented by

$$\sigma_x = \sqrt{\frac{1}{n}\left\{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2\right\}},$$

where, $x_1, x_2, ..., x_n$ be n observations of a random variable X, which is representation of an arbitrary image pixel.

Wavelet Energy (WE). For present work, wavelet packets has been generated using DWT or discrete wavelet transform which uses sub-band coding on images with respect to spatial and frequency components and allows analysis the images from coarse to fine level [15]. Here Daubechies wavelets *dbN* where N = 1, 2, 3 are chosen to generate sub-band images with approximation coefficients *cA*, *cH*, *cV* and *cD*. Their advantage includes computational ease with minimum resource and time requirements. These orthogonal wavelets are characterized by maximum number of vanishing moments for some given support. Here a signal (for present work it is an word image) is decomposed into different frequencies with different resolutions for further analysis. In general the family of Daubechies wavelet is denoted as *dbN*, where the family is denoted by the term *db* and the number of vanishing moments is represented by *N*.

To study the applicability of wavelet analysis in our work, we studied that an image can be represented by the combinations of different coefficients i.e. constant, linear, quadratic etc. Daubechies *db1* represent the constant coefficient of the image component, *db2* represent the linear and *db3* can represent quadratic coefficients. So here, wavelet decomposition at level 1 has been made using *db1*, *db2* and *db3* which are capable enough to capture the constant, linear and quadratic coefficients of an image component. Four coefficients namely approximation coefficients (cA), horizontal coefficients (cH), vertical coefficients (cV), and diagonal coefficients (cD) has been generated.

To measure the WE or wavelet energy feature we have computed wavelet entropy on these approximation coefficients for each of the sub-band images.

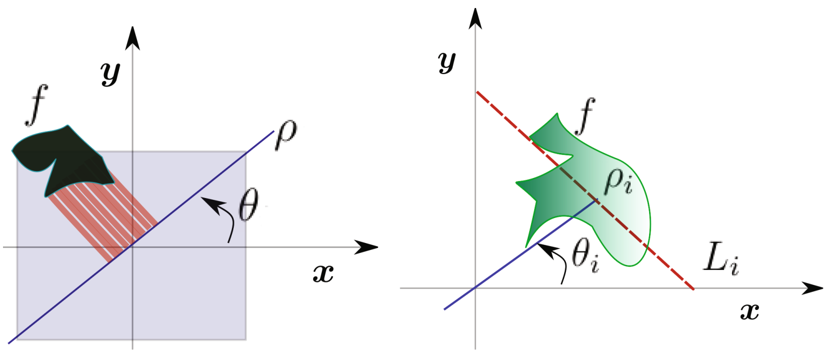
Suppose ws is the word level image signal and $(ws_i)_i$ the coefficients of ws in an orthonormal basis. Then the normalized shanon entropy is defined as:

$$SE(ws_i) = (ws_i^2)log(ws_i^2).$$

$$\text{So, } SE(ws_i) = -\sum (ws_i^2)log(ws_i^2).$$

It produces a feature vector of dimension of size 15.

The Radon Transform (RT). Motivated by the presence of the strokes at different orientations in the word images, we propose to use the RT. The RT consists of a collection of projections of a pattern at different angles [16], as illustrated in Fig. 2. In other words, the radon transform of a pattern $f(x, y)$ and for a given set of angles can be thought of as the projection of all non-zero points. This resulting projection is the sum of the non-zero points for the pattern in each direction, thus forming a matrix. The matrix elements are related to the



(a) projection at angle θ

(b) definition

Fig. 2. Illustrating the Radon transform theory (source: Ref. [17]).

integral of f over a line $L(\rho, \theta)$ defined by $\rho = x \cos\theta + y \sin\theta$ and can formally be expressed as,

$$R(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(x \cos\theta + y \sin\theta - \rho) dx dy$$

where $\delta(.)$ is the Dirac delta function, $\delta(x) = 1$, if $x = 0$ and 0 otherwise. Also, $\theta \in [0, \pi)$ and $\rho \in]-\infty, \infty[$. For the RT, L_i be in normal form (ρ_i, θ_i).

Such a description is useful for scripts such as Bangla and Devanagari, where there exists horizontal line, known by the name 'matra' or 'shirorekha'. These clear lines can be exploited by computing $0°$ projection. Similarly, scripts like Tamil and Roman have many vertical lines which can be represented by $90°$. However, to exploit meaningful information, we do not require all possible orientations, and therefore, we study the RT at an interval of $15°$.

To compute RT based feature vector we applied the Radon transform on each of the binary word images. Additionally RT spectrum of each of the sub band images is also obtained from Daubechies multi-resolution analysis using db1, db2 and db3 at level 1. Then statistical textural features are computed from the generated Radon spectrum. This step results a sixty five dimensional RT feature vector.

3.3 Script Classification

In our study, three different classifiers are used to train and to identify the words. They are MLP, FURIA and RF, which are briefly explained in the following.

Multilayer Perceptron (MLP). It consists of multiple layers with number of neurons in each layer represented as a directed graph [12]. MLP uses back propagation algorithm to train the network. In our experiment, we choose the configuration of the NN as 84-*hl*-13 (i.e., 84 number of attributes while taking SE+WE+RT and 13 output classes). We empirically designed the number of neurons in the hidden layer, *hl*.

Fuzzy Unordered Rule Induction Algorithm (FURIA). It is a fuzzy-rule-based classifier which learns from fuzzy rules and unordered rule sets [13]. It is an extension of the well-known rule learner RIPPER algorithm which is a state-of-the-art rule learner technique. Its preserves its advantages, such as simple and comprehensible rule sets for the learning. Along with that, RIPPER also includes a number of positive modifications and extensions. In particular, FURIA learns fuzzy rules instead of conventional rules and unordered rule sets instead of lists of rules. Moreover, to deal with uncovered examples, it makes use of an efficient rule stretching method. Experimental results show that FURIA significantly outperforms the original RIPPER in terms of classification accuracy.

Random Forest (RF). It is an ensemble learning method for classification, regression and other tasks. RF operates by constructing a multitude of decision trees at training time and providing the output class, which is the mode of the classes (classification) or mean prediction (regression) of the individual trees. RF corrects for decision trees' habit of over fitting to their training set [14].

4 Experiments

4.1 Evaluation Metrics

To measure the performance of the system, we use the following metric,

$$Identification\_rate = \frac{\#correctly\_classified\_words}{\#total\_words} \times 100\%.$$

Very specifically, we have computed the features (*cf.* Sect. 3.2), their possible combinations, and classifiers (*cf.* Sect. 3.3) separately.

4.2 Set up

In our study, from 13 different languages from 11 different scripts, we have considered two different test categories: (1) bi-script and (2) tri-script. In general, there are ${}^{13}C_2$ and ${}^{13}C_3$ possible combinations of bi-script and tri-script categories. But, considering the nature of the multi-script documents, these straightforward combinations may not hold true in the real-world (e.g. postal documents and application forms). We have also observed that, Devanagari and Roman exist in most of the documents. This means that any bi-script or tri-script document in general contains either or both Devanagari and/or Roman in addition to their local script. Considering such a context, we have formed two different script sub-categories for bi-script: case 1 and case 2. Bi-script case 1 contains twelve script combinations with Devanagari common. Bi-script case 2 contains Roman as common script, for all remaining 12 scripts. For tri-script category, we have a total number of 11 combinations where both Devanagari and Roman are kept as common with other local scripts.

Also, note that, we have divided the database into training and test sets as 2:1 ratio.

4.3 Results and Analysis

Again, our experimental test framework can be summarized as follows. As said before, in this work, our idea is not only to check what features but also to check what classifiers can consistently provide optimal performance. Therefore, we have seven different tests in accordance with the use of individual features and their possible combinations: SE, WE, RT, SE+WE, SE+RT, WE+RT and SE+WE+RT. These are tested by using three different classifiers: MLP, FURIA and RF.

Table 1. Bi-Script case 1 (Devnagari common): average performance scores (in %) for different feature combinations.

Feature type (dimension)	Classifier		
	MLP	FURIA	RF
SE (4)	81.93	80.30	86.28
WE (15)	91.10	89.30	92.35
RT (65)	96.93	95.31	95.84
SE+WE (19)	94.83	93	94.98
SE+RT (69)	97.86	97.09	97.03
WE+RT (80)	97.80	96.36	96.48
SE+WE+RT (84)	**98.38**	97.42	97.35

Table 2. Bi-Script case 1 (Devnagari common): average performance (in %) scores for 12 different combinations when all features (SE, WE, RT) are combined.

Bi-script combinations case 1	Classifier		
	MLP	FURIA	RF
DEV-BEN	94.70	95.00	94.20
DEV-DOG	99.70	99.00	98.30
DEV-GUJ	99.40	98.70	98.30
DEV-GUR	90.90	89.50	91.60
DEV-KAN	99.20	97.90	97.90
DEV-KAS	99.90	99.30	99.00
DEV-MAL	99.70	98.80	98.30
DEV-ORY	99.90	99.50	99.60
DEV-ROM	99.30	97.60	97.50
DEV-TAM	98.40	95.90	96.10
DEV-TEL	99.90	98.80	98.60
DEV-URD	99.60	99.00	98.80
Average	**98.38**	97.42	97.35

In Table 1, average performance scores for different feature combinations are provided. The results are provided for bi-script case 1. One of the scores in this table is computed by making 12 number of runs as shown in Table 2. Altogether, we have 12(bi-script combinations) $\times$ 3(classifers) $=$ 36 runs, for just a single feature type. In Table 1, MLP provides the best performance (i.e., 98.38%) when all features are combined, which, however, does not provide a significant difference other classifiers. In a similar fashion, bi-script case 2 has been tested, where Roman is common. Results are provided in Tables 3 and 4 for bi-script case 2 (follow Tables 1 and 2). In the latter case (i.e., Table 3),

Table 3. Bi-Script case 2 (Roman common): average performance scores (in %) for different feature combinations.

Feature type (dimension)	Classifier		
	MLP	FURIA	RF
SE (4)	80.94	79.93	81.90
WE (15)	92.56	91.2	93.50
RT (65)	98.01	96.67	96.66
SE+WE (19)	95.06	94.07	95.60
SE+RT (69)	98.96	97.68	97.68
WE+RT (80)	99.07	97.58	97.62
SE+WE+RT (84)	**99.24**	97.91	98.11

Table 4. Bi-Script case 2 (Roman common): average performance (in %) scores for 12 different combinations when all features (SE, WE, RT) are combined.

Bi-script combinations case 2	Classifier		
	MLP	FURIA	RF
ROM-BEN	99.00	96.60	97.50
ROM-DEV	99.30	97.60	97.50
ROM-DOG	99.30	97.80	98.00
ROM-GUJ	98.20	94.90	95.40
ROM-GUR	99.30	99.20	98.80
ROM-KAN	99.30	99.00	98.40
ROM-KAS	99.50	99.20	99.40
ROM-MAL	99.10	97.40	98.60
ROM-ORY	99.70	98.90	99.20
ROM-TAM	99.30	97.40	97.10
ROM-TEL	99.50	98.60	99.00
ROM-URD	99.40	98.30	98.4.0
Average	**99.24**	97.91	98.11

the highest possible accuracy is 99.24%. Like before, MLP provides better results when all features are combined – even for tri-script combinations. In Table 5, average performance scores are provided for tri-script combinations, where the highest possible identification rate is 98.19%. In this test, we have submitted 11(tri-script combinations) $\times$ 3(classifers) = 33 runs, for just a single feature type. Again, for a comparison (between the classifiers) purpose, their average scores are provided in Table 6, where we found MLP$>$ RF$\geq$FURIA, even though there exists no significant difference between them. In this comparison table, one can also note that higher the script combination, lower the performance of classifiers – which is obvious because it increases number of classes to be classified.

Table 5. Tri-Script (Devnagari & Roman common): average performance (in %) scores for 11 different combinations when all features (SE, WE, RT) are combined.

Tri-script combinations	Classifier		
	MLP	FURIA	RF
DEV-ROM-BEN	96.20	93.40	90.70
DEV-ROM-DOG	99.20	96.90	98.00
DEV-ROM-GUJ	97.80	95.60	94.00
DEV-ROM-GUR	94.00	89.00	84.70
DEV-ROM-KAN	98.90	96.70	96.50
DEV-ROM-KAS	99.60	97.00	96.70
DEV-ROM-MAL	98.70	96.00	95.50
DEV-ROM-ORY	99.50	97.60	98.00
DEV-ROM-TAM	97.90	94.40	94.00
DEV-ROM-TEL	99.30	97.70	97.90
DEV-ROM-URD	99.00	96.70	96.00
Average	**98.19**	95.55	94.73

Table 6. Comparison of classifiers when all features are combined. Average scores are reported.

Test category (combinations)	Classifier		
	MLP	FURIA	RF
Bi-script case 1 (12)	98.38	97.42	97.35
Bi-script case 2 (12)	99.24	97.91	98.11
Tri-script (11)	98.19	95.55	94.73

4.4 Previous Relevant Work – Analogy

Prior to this study, Pati et al. [1] proposed a word-level script identification by using 11 Indic languages, where Gabor and DCT based features are taken. They have compared their performances using three different classifiers namely neural network (NN), linear discriminant analysis (LDA) and support vector machine (SVM). Their performance scores are approximately 98% from both bi-script and tri-script combinations.

In contrast, our work is composed of all 13 official languages under 11 different scripts, with 39k dataset. Three type of features are used: spatial energy, wavelet energy and radon transform. Performances of three different classifiers namely MLP, FURIA, and RF have been compared, and MLP is found to be better performer. In our comprehensive tests, we have script identification rate of 98.38% (keeping Devanagari common) and 99.24% (keeping Roman common) for bi-script combination, and identification rate of 98.19% for tri-script combination. For better understanding a comparative chart is shown by Table 7.

Table 7. Analogy with the previous work.

Method	Database	Identification rate (in %)
Pati et al. [1]	11 languages	98.00 (bi-script)
		98.00 (tri-script)
Proposed work	13 languages	99.24 (bi-script)
		98.38 (bi-script)
		98.19 (tri-script)

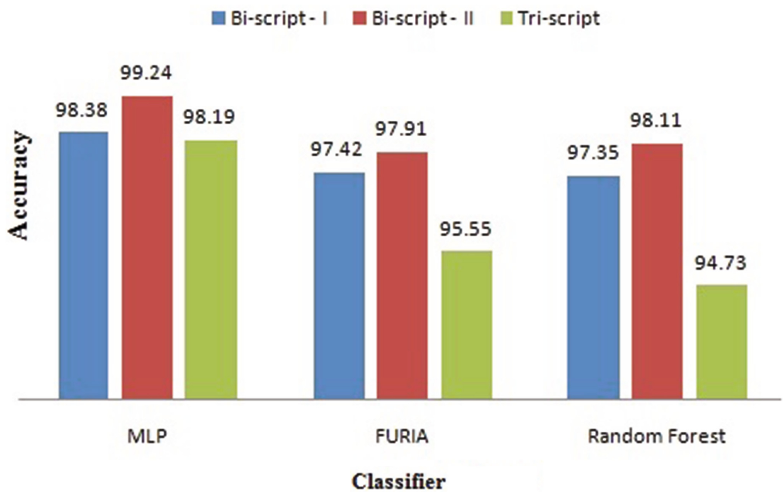


Fig. 3. Performance comparison of different classifiers.

The graphical representation of the performance comparison of different classifiers is illustrated in Fig. 3.

5 Conclusions and Plan

No doubt, script identification has been taken as the well studied problem since several years but, we do not have fairly large database for research, and therefore, one can make fair comparison. Motivated by this, in this paper, we have presented a script identification database, which is composed of 13 official Indic languages for research purpose. Our database is composed of 39K words that are equally distributed (i.e., 3K words per language). We have also studied MLP, FURIA and RF classifiers by using three different features that are derived from spatial energy, wavelet energy and the Radon transform. In our test, using all features, MLP is found to be the best performer showing the bi-script accuracy of 99.24% (keeping Roman common), 98.38% (keeping Devanagari common) and tri-script accuracy of 98.19% (keeping both Devanagari and Roman common).

In our plan, we are in the process to investigate those few misclassification samples (i.e., from Kashmiri-Urdu, Devnagari-Gurumukhi combinations) so that we can come up with new features to achieve the expected performance. Also, integrating classifiers in an immediate step.

References

1. Pati, P.B., Ramakrishnan, A.G.: Word-level multi-script identification. Pattern Recog. Lett. **29**(9), 1218–1229 (2008)
2. Hochberg, J., Kelly, P., Thomas, T., Kerns, L.: Automatic script identification from document images using cluster-based templates. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 176–181 (1997)
3. Pal, U., Chaudhuri, B.B.: Identification of different script lines from multi-script documents. Image Vis. Comput. **20**(13/14), 945–954 (2002)
4. Jawahar, C.V., Kumar, M., Kiran, S.S.R.: A bilingual OCR for Hindi-Telugu documents and its applications. In: Proceedings of International Conference Document Analysis and Recognition, pp. 408–412 (2003)
5. Chanda, S., Sinha, S., Pal, U.: Word-wise English Devnagari and Oriya script identification. In: Speech and Language Systems for Human Communication, pp. 244–248 (2004)
6. Joshi, G.D., Garg, S., Sivaswamy, J.: Script identification from Indian documents. In: 7th International Association of Pattern Recognition Workshop on Document Analysis Systems, pp. 255–267 (2006)
7. Dhanya, D., Ramakrishna, A.G., Pati, P.B.: Script identification in printed bilingual documents. Sadhana **27**(1), 73–82 (2002)
8. Chaudhury, S., Harit, G., Madnani, S., Shet, R.B.: Identification of scripts of Indian languages by combining trainable classifiers. In: Indian Conference on Computer Vision Graphics and Image Processing (2000)
9. Ghosh, D., Dube, T., Shivprasad, S.P.: Script recognition-a review. IEEE Trans. Pattern Anal. Mach. Intell. **32**(12), 2142–2161 (2010)
10. Hangarge, M., Santosh, K.C., Pardeshi, R.: Directional discrete cosine transform for handwritten script identification. In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 344–348 (2013)
11. Pardeshi, R., Chaudhuri, B.B., Hangarge, M., Santosh, K.C.: Automatic handwritten Indian scripts identification. In: 14th International Conference on Frontiers in Handwriting Recognition, pp. 375–380 (2014)
12. Obaidullah, S.M., Mondal, A., Das, N., Roy, K.: Script identification from printed Indian document images and performance evaluation using different classifiers. Appl. Comput. Intell. Soft Comput. **22**, 12 (2014)
13. Huhn, J., Hullermeier, E.: FURIA: an algorithm for unordered fuzzy rule induction. Data Min. Knowl. Discov. **19**(3), 293–319 (2009)
14. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
15. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 674–693 (1989)
16. Deans, S.R.: Applications of the Radon Transform. Wiley Interscience Publications, New York (1983)
17. Santosh, K.C., Lamiroy, B., Wendling, B.: DTW for matching radon features: a pattern recognition and retrieval method. In: 13th International Conference on Advances Concepts for Intelligent Vision Systems, pp. 249–260 (2011)

Springer

http://www.springer.com/978-981-10-4858-6

Recent Trends in Image Processing and Pattern Recognition
First International Conference, RTIP2R 2016, Bidar, India, December 16–17, 2016, Revised Selected Papers
Santosh, K.C.; Hangarge, M.; Bevilacqua, V.; Negi, A. (Eds.)
2017, XIV, 452 p. 223 illus., Softcover
ISBN: 978-981-10-4858-6