A Simple and Effective Solution for Script Identification in the Wild

Ajeet Kumar Singh, Anand Mishra, Pranav Dabral and C. V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India.

***Abstract*—We present an approach for automatically identifying the script of the text localized in the scene images. Our approach is inspired by the advancements in mid-level features. We represent the text images using mid-level features which are pooled from densely computed local features. Once text images are represented using the proposed mid-level feature representation, we use an off-the-shelf classifier to identify the script of the text image. Our approach is efficient and requires very less labeled data. We evaluate the performance of our method on a recently introduced CVSI dataset, demonstrating that the proposed approach can correctly identify script of 96.70% of the text images. In addition, we also introduce and benchmark a more challenging Indian Language Scene Text (ILST) dataset for evaluating the performance of our method.**

***Keywords*—Mid Level Features, Script Identification, Scene Text**

I. Introduction

Reading text in scene images can provide useful information about the content of the image. In multilingual country like India, sign boards often contain text of regional languages along with English and Hindi. The first step of reading text in such images is to answer "what script is this?". The goal of this paper is to answer this question (see Figure 1). To this end we use off-the-shelf text localization method and propose a novel mid-level feature based representation, for robust script identification. The proposed method achieves an accuracy of 96.70% on a recently introduced Video Script Identification (CVSI) dataset [1]. For comprehensive evaluation of our method we also introduce and benchmark a more challenging Indian Language Scene Text (ILST) dataset in this work. This dataset can be viewed and downloaded from our project website[1].

Script identification in printed and handwritten document images is a highly researched problem [2, 3, 4]. Contrary to the scanned and handwritten images, script identification in scene images poses many additional challenges such as, (i) lack of context. Scene text often appears as a single word or a group of words, and applying larger sentence or paragraph level context is hard. (ii) stylish fonts. Scene text images often contains stylish fonts to attract the viewers and do not easily generalize to the test data, (iii) complex background. Scene text come with highly complex natural scene background, on the other hand document images often contain, predominantly, text. In case of scene images, text localization and dealing with the false detection are few additional challenges. In this work we demonstrate the cropped word script identification as well as end-to-end script identification on scene images.

[1]http://cvit.iiit.ac.in/projects/SceneTextUnderstanding/

Fig. 1. A typical example of a street scene image captured in a multilingual country, e.g. India. Our goal in this paper is to localize the text and answer "what script is this?" to facilitate the reading in scene images.

There are many methods in the literature for script identification [1, 2, 5, 6, 7, 8]. Texture based features such as Gabor filter [7], LBP [9] have been used for script identification. Joshi *et al.* [6] proposed multi-channel *log*-Gabor filter bank and hierarchical classification scheme for script identification in Indic language documents. Reader is encouraged to refer [2] for detailed survey of classical methods in this area. These classical methods, though achieve high performance on printed documents, are not very successful in our case (see Section IV). In ICDAR 2015, a competition for script identification on video text was organized [1]. We compare our method with the entries for this competition and show that our method is comparable to the top performing methods in this competition. There have been few contemporary methods based on deep learning [1, 8] and RNN [3]. These methods achieve noticeable success on some of the selected benchmarks. However, these methods often require huge training data and computation resources.

In this paper, we propose a simple and effective solution for script identification in the wild. Our method is inspired by recent advancements made in mid-level features [10, 11, 12]. First, we densely compute the local features on the given image and then pool these local features to encode the larger context about the given image. In our case, these larger context encode the mid-level representation of the scripts. We represent each training image using bag of these mid-level features and learn a classifier to identify the script of a test image. The advantages of our method are two fold, firstly, it is robust to variations and noise commonly present in the scene text and secondly, the method is easily trainable and computationally efficient.

(a) (b)

Fig. 2. Few example images from the Indian Language Scene Text dataset (ILST) datset we introduce. (a) we provide ground truth text bounding box, script and text for the images. (b) Few cropped word images of our dataset.

The remainder of the paper is organized as follows. We discuss about datasets in Section II. Here, we introduce Indian language scene text dataset for the problem. In Section III, mid-level features and novel mid-level features based feature representation for text images is introduced. Section IV gives details of the evaluation protocols, and performance measures used in this work. Experimental settings, results, discussions, and comparisons with various techniques are also provided in this section, followed by conclusions.

TABLE I. THE ILST DATASET: WE INTRODUCE A ILST DATASET WHICH CONTAINS 578 SCENE IMAGES AND 3486 CROPPED IMAGES FROM 5 MAJOR INDIAN LANGUAGES.

Languages	# scene images	# word images	Mode of collection
Hindi	76	514	Authors, Google Images
Malayalam	121	515	Authors, Google Images
Kannada	115	534	Char74K [16]
Tamil	59	563	Authors
Telugu	79	510	Authors
English	128	850	Authors
total	578	3486	-

II. DATASETS

A. The ILST dataset

Scene text understanding has gained huge attention in last decade, and several benchmark datasets has been introduced [13, 14]. Most of these datasets are used for scene text localization and recognition in English. There are also few datasets [8, 1] of multiple scripts e.g., east Asian languages or Indian language video text. In this work we introduce Indian Language Scene Text (ILST) dataset which is a comprehensive dataset for Indian language scene text containing six scripts commonly used in India, namely Telugu, Tamil, Malayalam, Kannada, Hindi and English. The dataset contains 500 scene images with more than 3000 words. It can be used for following tasks- text localization, recognition, script identification. In this work we use this dataset for two tasks- cropped word script identification and text localization with script identification (i.e., end-to-end pipeline).

Comparison with other related datasets. To our knowledge ILST dataset is the largest scene text dataset for the Indian languages. Other related datasets such as CVSI [1], SIW [8] are only meant for script identification on cropped words whereas ours can be used for many other related tasks e.g., recognition and text localization. Also, the dataset is collected in a realistic setting and has wide variations in scale, font style, background and illumination.

Mode of collection. We have collected the images for this dataset by either capturing pictures in streets of various cities in India, harvesting images from Google image search or importing and providing annotation for few images from other existing datasets. These images contain signboards, billboards, posters mainly from urban part of the country. We have collected these images in an unconstrained manner, i.e., without considering much of the view angle and camera setting. These are intentionally done to create a realistic dataset.

Annotations. For annotations of the scene images, we use a publicly available web based tool LabelMe [15]. All the annotations are provided in XML for each image separately describing global image features, bounding boxes of text and its special characteristics. The XML-Schema of LabelMe has been adapted and extended by tags for additional metadata.

Each text field (word) in the image is annotated with following attributes: (i) word bounding box. These bounding boxes are rectangular and parallel to the axes. (ii) the inherent script of the text, and additional information such as, (iii) illumination, (iv) blur, (v) occlusion, and (vi) 3D Effects.

Train-Test splits. We provide a standard train and test splits for this dataset. We use randomly chosen 30% images of the dataset for the training and the rest for testing, while making sure the adequate representation of each script in train and test sets. We will make this dataset publicly available on our project website. The details of dataset is provided in Table I. We also show few example images of our dataset in Figure 2.

B. CVSI 2015 [1]

To show the genarality of our method, we also evaluate our method on a dataset which has been introduced in Video Script Identification Competition held at ICDAR 2015. The dataset is composed of images from news videos of various Indian languages. It contains 6412 training text images and 3207 testing text images from 10 different scripts namely,

English, Hindi, Bengali, Oriya, Gujarati, Punjabi, Kannada, Tamil, Telugu and Arabic, commonly used in India.

III. Methodology

In this section we introduce our mid-level features based representation of text images for script identification task. First, we briefly give motivation and overview of our method, compare it with closely related works, and then give the details of how we obtain mid-level features representation by pooling local low level features. We finally summarize the full pipeline of our approach.

A. Motivation and overview

Mid-level feature representation have gained huge attention in last few years. These features are potentially more distinctive than the traditional low-level local features constructed in a purely bottom-up fashion [10]. Mid-level features have achieved noticeable success in image classification and retrieval tasks [11, 12, 10]. Our method is inspired from these methods as we present a mid-level feature based representation which are robust for the task of our interest, i.e., script identification in the wild.

Script identification in the wild is a challenging problem. The traditional low-level local features are not competent for this task. This is primarily due to the imaging, variation in scale, ambiguity, sharing of mid-level representation in the scene text images. On the other hand strokes are the atomic units of scripts and collection of mid-level features are discriminative enough for the task of identifying the scripts. Our method is build on these intuitions. In this work, given a text image we first densely compute local visual features, and then pool these local features into frequently occurring larger patterns (or mid-level representation) and each text image is represented using histogram of these larger patterns (or mid-level representation).

Comparison with other related approaches: The mid-level features have outperformed the naïve bag-of-visual-words based features for image classification [11], because of their robustness and better discriminating power. Usually these mid-level features capture the larger context in the image as compared to the local or semi-local features. One alternative to use larger context is to simply cluster larger patches, as done for local feature computation. However such method are not effective in our case due to the large variability in scene text images. In context of supervision methods, mid-level features can be grouped into three categories: supervised [11], weakly-supervised [10] and unsupervised [17]. Our method falls in weakly supervised category where we only need the class information.

B. Mid-level Features based representation

We compute the mid-level feature representation of words from a labeled data. The overview of our method is illustrated in Figure 4. Given training text image I_i and its script s_j where $s_j \in \mathcal{S}$ (set of scripts), we follow the following steps.

- We densely compute the local features and represent each training image I_i as a set of descriptors (see Section III-B1 for details of feature computation).
- All the descriptors are then clustered to obtain visual words. Let $C = \{c_1, c_2, ..., c_m\}$ be the set of visual words with vocabulary size = m.
- We obtain assignment for every feature ϕ_k, i.e., obtain the feature-visual word pair (ϕ_k, c_l)
- In a $p \times q$ rectangular neighborhood around feature ϕ_k we obtain a local histogram of visual words H_k. These H_ks capture a larger context and are more discriminative patterns.
- We again cluster local histograms H_k to obtain larger patterns which encode the mid-level representation. Let $\psi = \{\psi_1, \psi_2, \cdots, \psi_n\}$ be the set of such clusters with mid-level features' vocabulary size = n.
- Once H_k and ψ in hand, we assign every local histogram H_k to one of cluster from ψ. In other words, each image is represented as bag of mid-level features. We name this representation χ.

Fig. 3. We show some representative mid-level representation of following scripts (top to bottom): Hindi, Kannada, Malayalam, Tamil and Telugu. Our method yields the mid-level representation which are representative and discriminative enough for a cropped image.

At the end of this process each word image in the training data is represented with mid-level features χ. To only use the best mid-level representation we prune few less informative mid-level representation by using the method described in Section III-B2.

1) Feature computation: Given a text image we compute the SIFT descriptors at points on a regular grid with spacing of M pixels. At each point the descriptors are computed over four circular support patches with different radii, consequently each point is represented by four SIFT descriptors. We also learn the multiple descriptors to allow the scale variation between images. At each grid point the descriptors are computed over circular support patches with radii $r = 4, 6, 8$ and 10.

2) Finding the best mid-level representation for the task: We wish to use the mid-level features χ as a novel set of mid-level features to describe the text image. But not all mid-level features are are relevant for the task of script identification, e.g., a representation commonly occurring in all the scripts

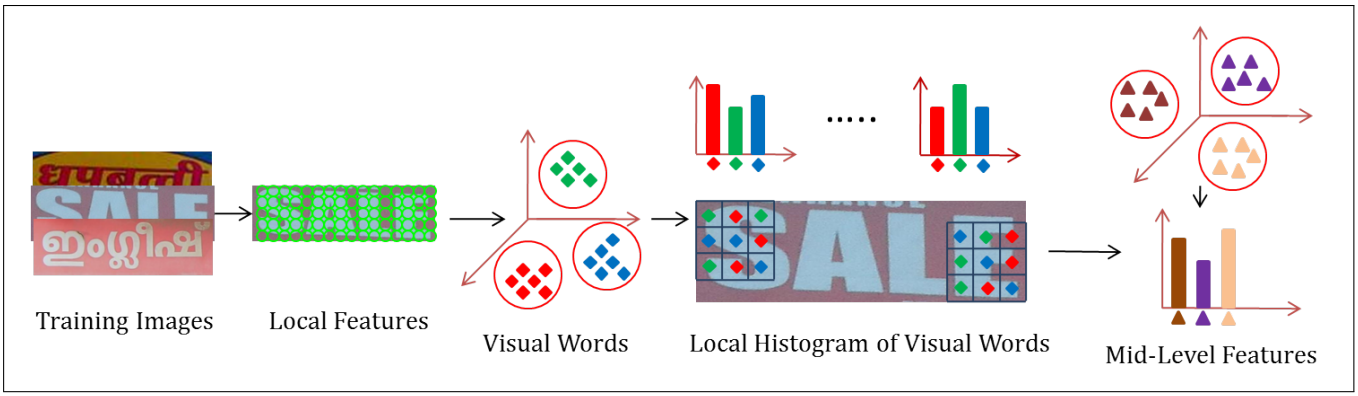

Fig. 4. Method Overview: The figure depicts the feature computation process where, first we find the local features from the images, we cluster these feature to get the local histogram of visual words. Then we cluster the histogram of visual words to get the representation of words in form of mid-level representation.

TABLE II. RESULTS ON ILST (CROPPED WORDS SCRIPT IDENTIFICATION)

Method	***Accuracy** (%)*
Baseline Methods	
Gabor features [7]	59.25
Gradient features	47.74
Profile Features [18]	49.24
LBP [9]	78.08
Ours	**88.67**

TABLE III. RESULTS ON ILST (END-TO-END PIPELINE). WE USE [19] AND TESSERACT [20] FOR TEXT LOCALIZATION AND EVALUATE OUR PROPOSED METHOD OF SCRIPT IDENTIFICATION BASED ON MEASURE PRESENTED IN SECTION IV-B

Script	*Precision*	*recall*	*f-score*
Telugu	0.47	0.54	0.51
Tamil	0.41	0.44	0.42
Malayalam	0.49	0.45	0.47
Kannada	0.39	0.47	0.42
Hindi	0.42	0.48	0.45
English	0.46	0.56	0.50

may not carry useful information (not *discriminative*). Moreover, there are few mid-level features which can occur in most of the images in a script and are very good *representative* of that script. To measure discriminativity and representativity of a representation $str \in \chi$ we compute following relevance score:

$$rel(str) = D(str) \times R(str), \quad (1)$$

where $D(str)$ and $R(str)$ are discriminativity and representativity scores respectively. To compute $D(str)$ and $R(str)$ we follow entropy based formula. We compute the entropy of representation by considering (i) scripts as class (script specific entropy) (ii) individual images in scripts as class (image specific entropy). We use these entropies to define $D(str)$ and $R(str)$ such that lower value for script specific entropy and higher value for image specific entropy results in higher values of $D(str)$ and $R(str)$. This ensures that those mid-level representation which are found in certain script and almost all the images of that script are more relevant. We prune the bottom 20% less relevant mid-level representation. Figure 3 shows some of the relevant representation of the scripts used in this task.

C. Script identification: Full pipeline

Given a scene image our goal is to localize text and then identify its script. For this, we first obtain text localization using a method proposed in [19] and an open source OCR [20]. While the text localization technique we apply is rather standard, we adapt this for the multi-script dataset we use. Once the text is localized we represent it using mid-level features representation which is learned from the training data (discussed in Section III-B). Each localized text is now fed to a linear SVM classifier which is trained for the task to obtain the text script.

IV. EXPERIMENTS

Given a scene image containing text our goal is to localize the text and identify it's script. We show results in two settings, (i) end-to-end pipeline, and (ii) cropped word script identification on the datasets presented in Section II. In this section we provide details of implementation, evaluation protocols and baseline methods, and evaluate the performance of our method and compare it with previously published works.

A. Implementation details and design choice

The proposed method is implemented on a system with 16 GB RAM and Intel® Core™ *i3*-2105 CPU @ 3.10GHz system. The proposed system takes approximately 0.4 *ms* to identify the script of a cropped word. The two important parameters visual word vocabulary size m and representation vocabulary size n (refer Section III-B) were empirically chosen as 4K and 3K respectively. The parameter C in SVM is obtained using grid search on independent validation set. We keep these parameters fixed for all our experiments.

B. Evaluation Protocols

End-to-end script identification. We evaluate our method on end-to-end pipeline of script identification. For this we first localize the text in scene images. We use a standard available text localization scheme for localizing the text. Obviously, this step misses some text regions and produces few false bounding boxes. We fed all the text candidate bounding boxes to script identifier.

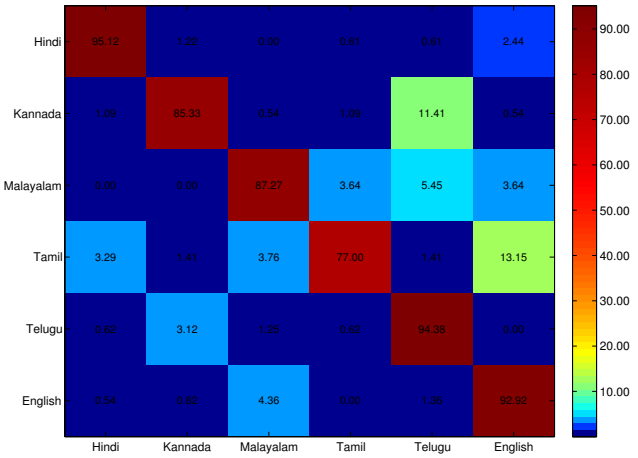

Fig. 5. Confusion matrix on ILST cropped words. Our method achieve a 88.67% accuracy of script identification on the introduced dataset.

It should be noted that end-to-end script identification is far more challenging than script identification in cropped words or document images. Since the final score of this reflects the error accumulated due to text localization and incorrect script identification.

To evaluate the end-to-end script identification we use standard measures, precision ($prec$), recall (rec) and $fscore$ computed for every script. For every script s we compute following terms: (i) number of correctly identified words (TP_s). A detected word is called correctly identified if the intersection by union overlap with the ground truth bounding box is more than 60% and it has the same script identified as the ground truth, (ii) total number of identified words (TI_s), and (iii) total number of ground truth words (TG_s)

Once these in hand we compute *precision*, *recall* and *f-score* for every script and every image. We then report mean of these score over all the images in the dataset.

$$prec_s = \frac{TP_s}{TI_s} \tag{2}$$

$$rec_s = \frac{TP_s}{TG_s} \tag{3}$$

$$fscore_s = 2\,\frac{prec \times rec}{prec + rec} \tag{4}$$

The ideal script identifier should achieve 1 for these measures for all the scripts.

Cropped word script identification. We also evaluate our method on cropped words. For this we compute accuracy which defined as follows:

$$Accuracy = \frac{correctly\ identified\ words}{total\ number\ of\ words} \times 100. \tag{5}$$

Here a word is called correctly identified if the method identifies script same as the ground truth.

C. Baseline Methods

We compare our methods with popular features used for script identifications in document images namely LBP [9], Gabor features [7]. We also evaluate gradient based features and profile features [18] for script identification task and compare with our method. For comparison in CVSI dataset we compare our method with the best performing methods reported in [1].

Fig. 6. Success and Failure Cases. Despite high variations in the dataset, our method correctly identifies the script of scene text images. The "Success" columns depicts the correctly classified word images, and wrongly classified words are shown in "Failure" column along with recognized script in red color alongside them.

TABLE IV. TASK SPECIFIC EVALUATION ON CVSI [1]. HERE A: ARABIC, B: BENGALI. E: ENGLISH, H: HINDI,G: GUJRATI, K: KANNADA, O: ORIYA, P: PUNJABI, TA: TAMIL, TE: TELUGU. HENCE AEH MEANS WHERE SCRIPT IDENTIFICATION OF THREE CLASS NAMELY, ARABIC, ENGLISH AND HINDI, IS PERFORMED AND SO ON. FURTHER, TASK-1, TASK-2, TASK-3 AND TASK-4 INDICATES SCRIPT TRIPLETS, NORTH INDIAN SCRIPT, SOUTH INDIAN SCRIPT, ALL SCRIPT IDENTIFICATION, RESPECTIVELY.

Task	Methods							
	C-DAC	CUK	HUST	CVC-1	CVC-2	Google	Shi *et al.* [21]	**Ours**
Task-1								
AEH	95.46	-	99.79	97.32	96.80	100.00	-	100.00
BEH	91.40	-	98.36	94.68	94.27	99.49	-	98.61
GEH	88.33	-	99.09	96.38	95.98	99.4	-	99.41
KEH	91.44	-	99.80	95.51	95.92	99.59	-	99.19
OEH	95.87	-	99.50	96.88	96.37	99.19	-	99.49
PEH	84.94	-	98.68	94.20	95.32	99.49	-	99.37
TaEH	92.71	-	99.39	95.95	96.66	99.70	-	99.61
TeEH	93.84	-	97.98	96.46	95.96	99.19	-	97.06
Task-2	96.79	79.50	97.69	95.73	95.91	99.19	93.80	97.99
Task-3	86.95	79.14	97.53	95.38	95.75	98.95	96.70	96.11
Task-4	84.66	74.06	96.69	95.88	96.00	98.91	94.30	96.70

D. Results on the ILST dataset

1) End-to-end script identification: We evaluate end-to-end script identification on ILST dataset. To this end we first use public implementation of [19] for text extraction and then fed it to an open source OCR [20] to obtain text boundaries. Once we get bounding boxes we perform script identification using our method and evaluate performance based on performance measures presented in Section IV-B. We summarize results of full pipeline in Table III. We observe that our method achieves reasonably high *fscore* for this challenging task. The robustness of mid-level features we use can be attributed as factor for this success. It should be noted that text localization is still an open problem and its performance affects the overall score of end-to-end script identification.

2) Cropped word Script Identification: We also show results on cropped words on ILST dataset. These results are summarized in Table II. Despite many challenges in this dataset (see figure 2) our method achieves script identification accuracy of 88.67% which is significantly better than methods used in document image script identification domain such as [7, 9]. To study script wise confusion we illustrate confusion matrix of our method for ILST dataset in Figure 5.

E. Results on CVSI dataset

Following the protocols of ICDAR competition on video script identification [1] we evaluate our method following for four tasks: (i) Task-1: script identification on script triplets (ii) Task-2: north Indian script identification (iii) Task-3: south Indian script identification, and (iv) Task-4: script identification in all the ten scripts.

We compare our method with top performing methods in this competition. These results are summarized in Table IV. Our method achieves 96.70% for Task-4, i.e., script identification in all the ten scripts and clearly outperform two methods in the competition namely, C-DAC and CUK. Moreover, our results are marginally superior to HUST, CVC-1, CVC-2 and comparable to the deep learning based best performing method by Google. We also compare our method with a recently published work [21] where our method achieves superior performance for Task 2 and Task 4 and comparable results on Task 1 and Task 3. For the Task 1, average accuracy of our method is 99.09% which is superior to the average accuracy of 96.1% as reported in [21].

F. Qualitative evaluation

We qualitatively evaluated our method in Figure 6 and Figure 7. We show results on end-to-end as well as cropped word script identification. We observe that despite high variations in images such as complex background, illumination change, low resolution our method is successful. Success and failure cases on the cropped image for six script are shown In Figure 6. In failure section, Kannada text is wrongly classified as Telugu due to similarity in inherent scripts of both the languages. Similarly, Malayalam text is wrongly classified as Tamil and vice versa. These scripts are visually very similar and often challenges script identifier. It is also very interesting that, an English word is classified as Kannada due to the writing style. Adding location information (i.e., where the image is captured), context (i.e., scripts of neighboring text) can help mitigating such errors. We plan to add such features in our method in future.

V. Conclusions

In this paper, we have addressed the problem of script identification in the wild. To this end we made following two important contributions: (i) we introduced a comprehensive dataset for Indian language scene text. This dataset will be useful for the community for many scene text related tasks in multilingual environment in the future. (ii) We have established a baseline for the end-to-end script identification pipeline for scene text and shown that simple mid-level features can achieve reasonably high performance for this task. As a future work we intend to extend our ILST dataset to 10 popular scripts used in India and explore the usage of multiple cues as aid to our script identifier, such as location of the image where it is captured and script of neighboring texts.

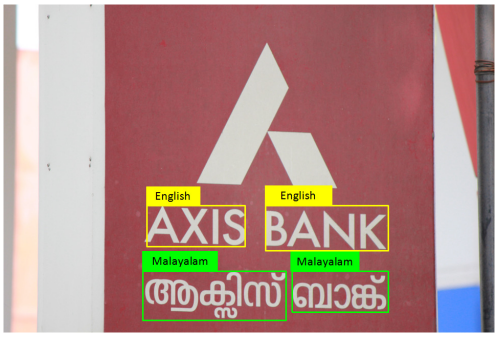

Fig. 7. An example result of End-to-end script identification of our method. We localize the text boxes in images using method using [19] and [20]. Then we apply our method to find the inherent script in the text boxes.

Acknowledgements. This work is supported by Ministry of Communication and Information Technology, Government of India, New Delhi.

References

[1] N. Sharma, R. Mandal, R. Sharma, U. Pal, and M. Blumenstein, "ICDAR2015 Competition on Video Script Identification(CVSI 2015)," in *ICDAR*, 2015.
[2] D. Ghosh, T. Dube, and A. P. Shivaprasad, "Script Recognition - A Review." *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.
[3] A. K. Singh and C. V. Jawahar, "Can RNNs Reliably Separate Script and Language at Word and Line Level?" in *ICDAR*, 2015.
[4] S. Chanda, S. Pal, K. Franke, and U. Pal, "Two-Stage Approach for Word-wise Script Identification," in *ICDAR*, 2009.
[5] T. Q. Phan, P. Shivakumara, Z. Ding, S. Lu, and C. L. Tan, "Video Script Identification Based on Text Lines," in *ICDAR*, 2011.
[6] G. D. Joshi, S. Garg, and J. Sivaswamy, "A Generalised Framework for Script Identification," *IJDAR*, 2007.
[7] P. B. Pati and A. G. Ramakrishnan, "Word Level Multi-script Identification," *PR Letters*, 2008.
[8] B. Shi, C. Yao, C. Zhang, X. Guo, F. Huang, and X. Bai, "Automatic Script Identification in the Wild," in *ICDAR*, 2015.
[9] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *TPAMI*, 2002.
[10] B. Fernando, É. Fromont, and T. Tuytelaars, "Mining Mid-level Features for Image Classification," *IJCV*, 2014.
[11] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks That Shout: Distinctive Parts for Scene Classification," in *CVPR*, 2013.
[12] Y. Boureau, F. R. Bach, Y. LeCun, and J. Ponce, "Learning Mid-Level Features for Recognition," in *CVPR*, 2010.
[13] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 Robust Reading Competition Challenge 2: Reading Text in Scene Images," in *ICDAR*, 2011.
[14] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene Text Recognition using Higher Order Language Priors," in *BMVC*, 2012.
[15] " LabelMe - The Open Annotation Tool," http://labelme.csail.mit.edu/.
[16] T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *VISAPP*, 2009.
[17] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised Discovery of Mid-Level Discriminative Patches," in *ECCV*, 2012.
[18] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *PAMI*, 2012.
[19] D. K. Lluis Gomez, "A Fast Hierarchical Method for Multi-script and Arbitrary Oriented Scene Text Extraction," in *arXiv:1407.7504*, 2014.
[20] Tesseract OCR, http://code.google.com/p/tesseract-ocr/.
[21] B. Shi, X. Bai, and C. Yao, "Script Identification in the Wild via Discriminative Convolutional Neural Network," in *Pattern Recognition*, 2015.