Multilingual Artificial Text Extraction and Script Identification from Video Images

Akhtar Jamil*, Azra Batool†, Zumra Malik†, Ali Mirza† and Imran Siddiqi†
*Yildiz Technical University, Istanbul, Turkey
†Bahria University, Islamabad, Pakistan

***Abstract*—This work presents a system for extraction and script identification of multilingual artificial text appearing in video images. As opposed to most of the existing text extraction systems which target textual occurrences in a particular script or language, we have proposed a generic multilingual text extraction system that relies on a combination of unsupervised and supervised techniques. The unsupervised approach is based on application of image analysis techniques which exploit the contrast, alignment and geometrical properties of text and identify candidate text regions in an image. Potential text regions are then validated by an Artificial Neural Network (ANN) using a set of features computed from Gray Level Co-occurrence Matrices (GLCM). The script of the extracted text is finally identified using texture features based on Local Binary Patterns (LBP). The proposed system was evaluated on video images containing textual occurrences in five different languages including English, Urdu, Hindi, Chinese and Arabic. The promising results of the experimental evaluations validate the effectiveness of the proposed system for text extraction and script identification.**

Keywords*—*Multilingual Text Detection; Video Images; Script Recognition; Artificial Neural Networks; Local Binary Patterns.

I. Introduction

Over the recent years, there has been a remarkable growth in the amount of multimedia data in the form of images, videos and audios. With the advancements in image/video capture hardware and the increase in the number of online image and video databases, digital multimedia content is likely to increase manifolds in the days to come. With this has increased the need to have efficient indexing and retrieval mechanisms allowing users rapid access to the content they are interested in. Among different types of multimedia data, the focus of our research interest lies on videos.

In addition to the visual content, videos comprise audio, text and other objects. The audio and visual information in the video could be effectively employed for development of semantic indexing and retrieval systems [1] and has been an attractive research area for over two decades now [2], [3]. In some cases, especially on the video sharing portals, users manually assign tags to videos allowing their retrieval. This retrieval, however, does not take into account the actual content of the video and is based on matching of tags only. In addition to the content of the video, a very powerful component, which could serve as an effective index, is the textual information in the video.

Text embedded in videos provides important, short and relevant information about the visual content. Examples of text occurrences include names of persons, sports scores, important dates, scene locations, movie credits, and stock rates etc. These embedded instances of text can be extracted and used as an effective index for retrieval from large video archival systems. As a result, development of automatic systems which could extract text from videos or images has been an attractive area of research in image analysis and pattern classification. Despite significant research on this problem, detection of textual information remains a challenging problem due to complex backgrounds, different font sizes and orientations and low contrast and resolution.

It is interesting to note that most of the research on this subject has focused on detecting text in a particular script. Properties of text in a particular script are exploited to detect its occurrences. Recently, there has been the trend of having multilingual text in videos especially the news channels where news tickers are flashed in multiple (generally two different) languages. It would be interesting to develop a generic system that could extract textual occurrences in videos or images irrespective of any language or script and this, in fact, is the subject of our study. The text detection module is generally integrated with text recognition (OCR) module to convert the occurrences of text in the image into text. For a detection system that works on a single script, the output of detector can directly be fed to the OCR module. In case of a multilingual detection system, however, the script of the detected text also needs to be identification so that it could be fed to the respective OCR system. This script identification has also been addressed in our work.

This work extends our previous contributions on text detection and extraction from video images [4], [5], [6].The main contribution of this research includes development of a generic text detection system in a multi-script environment which is not tuned to detect text in a particular language. The proposed approach is a combination of unsupervised and supervised techniques. In the first step, an unsupervised approach exploits the visual properties of text to segment candidate text regions using image analysis techniques. These candidate textual regions are validated by an Artificial Neural Network which is trained to differentiate between text and non-text blocks on the basis of a set of features extracted using the Gray Level Co-occurrence Matrices (GLCM). The developed system also identifies the script of the detected text using texture based features computed from the Local

Binary Patterns (LBP). The system evaluated on images with textual occurrences in five different languages (Urdu, English, Arabic, Chinese and Hindi) reports promising results on text detection as well as script recognition.

We first discuss the recent advancements in video text detection and extraction followed by the proposed methodology in Section III. Section IV describes the experimental evaluations conducted to validate the proposed methodology along with an analysis of the results realized. Finally, we conclude the paper with some ending remarks.

II. Background

Considering the applications it offers, detection of textual content from images and videos has been a highly researched area over the last decade. Text appearing in videos/images is generally classified into two categories, artificial text and scene text. Artificial text, also known as caption or superimposed text, is the text embedded and laid over the videos during the editing process to provide additional information related to its content such as news captions, sports scores, stock rates, etc. Scene text, on the contrary, is the text which appears naturally in the scene and is captured by the camera as a part of scene. Examples of scene text include text appearing on sign boards, billboards, names on shirts and vehicles etc. [7]. Detection and recognition of each category of text offers different types of applications. Scene text generally finds applications in robot navigation, license plate recognition and navigation of intelligent vehicles etc. Artificial text, which in general, is correlated with the content, is preferred for semantic indexing and retrieval of videos. Sample images containing occurrences of scene and artificial text are illustrated in Figure 1.

Fig. 1: Examples of (a) Scene text (b) Artificial text

In general, textual content based indexing and retrieval systems rely on four major steps namely text detection, localizaion, extraction and recognition. Text detection includes classification of a given region of interest as text or non-text region. Candidate text regions are fed to the localizer which finds the boundaries of text at character, word or line level depending upon the application. The localized text regions are then segmented from the background by the text extraction module. Finally, the extracted text regions could be fed to a recognition engine for conversion to text and subsequent indexing. Our research is aimed at extraction of text and subsequent identification of its script hence recognition of text is beyond the scope of our discussion.

Detection of text from images and videos has received notable research attention in the recent years. Traditionally, these methods are categorized into two broad classes, unsupervised and supervised techniques. The un-supervised approaches are based on image analysis techniques and use segmentation methods to differentiate text from other parts of the image. Supervised approaches for text detection employ machine learning algorithms to find text regions in an image. Traditionally, the supervised methods consist of two steps, training and classification. During training, features extracted from text and non-text regions are fed to a classifier to make it learn to differentiate between the two classes. During the classification phase, features extracted from the region in question are evaluated on the trained classifier which outputs the likelihood of the region as being text or non-text.

The unsupervised approaches for text detection mostly exploit the statistical and temporal features of text and, in general, work well in relatively less complex images. However, these methods may produce more false positive in complex scenes. The techniques used in this class of methods are further classified into gradient, connected component, texture and color clustering based methods.

Gradient-based methods [4], [8], [9], [10], [11], [12], [13] use edge information to segment the video images. They assume that there is high contrast between text and its background. Generally, an edge filter (e.g. Sobel or Canny operator) is applied for text detection, which is usually followed by some morphological processing to merge the desired edges to determine text lines [10], [14].

Texture based methods [15], [16], [17], [18] assume that text appearing in video frames has a unique texture that differentiates it from other objects in the image. Since the textural properties vary with font style and size, a generic texture filter for varying scenarios is hard to devise [1]. In addition, the computational complexity of these methods is also high as they require an exhaustive scan of whole image for text detection and localization.

Connected component based methods [19], [20], [21] either use region growing or splitting approach in order to group text pixels into clusters until all regions in the input image are identified. These methods are widely used for text localization due to their simple implementation. However, since these methods mainly rely on the contrast between text and background, they produce false alarms in case of low resolution images.

Color based methods [22], [23], [24], [25] use color information to cluster image content into text and non-text regions. These methods perform well for images with high

resolution and simple backgrounds. However, these assumption may not be true in many real world scenarios where text may appear in various colors and can be superimposed on complex backgrounds. In addition, due to compression, images may suffer from color bleedings affecting the performance of color based methods.

In supervised approaches for text detection, a learning machine is first trained on a set of features extracted from both text and non-text samples. Generally, these features are extracted by scanning the image with a small window which are then fed to the classifier. Calssifiers like support vector machine (SVM) and artificial neural networks (ANN) have been extensively applied for this purpose [26], [27], [28], [29], [30], [31]. In some cases, coarse-to-fine algorithms have also been evaluated where the candidate text pixels are first identified and then valiated by a classifer [32], [33].

With few exceptions, most of the text detection methods reported in the literature target text in a particular script. The literature is very rich when it comes to detect text in any of the languages based on the Latin alphabet (English, French, and German etc.). Detection of caption text in Chinese has also witnessed a significant research attention. For most of the other scripts, the research is either in its early days or is non-existent. In our proposed system, we aim to develop a generic text detection system that is not tuned to detect text in any particular script and works on multilingual text as detailed in the following section.

III. Proposed Methodology

This section presents in detail the proposed methodology for text detection and script identification. As discussed earlier, the target application of such text detection systems is indexing and retrieval of videos. The general architecture of such a system is illustrated in Figure 2. Textual information extracted from videos is fed to an Optical Character Recognition (OCR) system to convert it into text. The focus of our research, however, is on the first part, i.e. detection and extraction of text and identification of the script of the detected text.

The proposed system can be divided into three main modules. An unsupervised approach is first used to detect potential text regions. These text regions are validated through a supervised approach that employs an artificial neural network as classifier. Finally, the script of the extracted text is recognized using texture based features. Each of these modules is discussed in the following sub-sections.

A. Text Detection

For detection of potential text regions in the image, the image is first converted to grayscale [5]. A sequence of image analysis techniques is then applied to the image as discussed in the following.

1) Gradient Computation: Edges are a common feature of text in all scripts. Different scripts have different proportions of horizontal, vertical and diagonal edges corresponding to text strokes in each of these directions. In our study, we consider text in Urdu, English, Chinese, Arabic and Hindi, an example

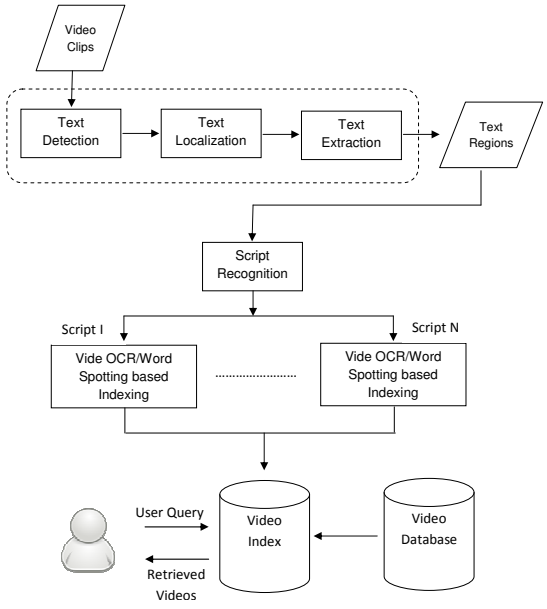


Fig. 2: General framework of a video indexing and retrieval system

of each being shown in Figure 3. It can be seen that in all of these texts, a reasonable proportion of strokes are vertical.

Fig. 3: Samples of text in (a) Urdu (b) Arabic (c) Chinese (d) Hindi (e) English

In our implementation, vertical edges are computed using the first derivative (gradient) by convolution of the image with the respective Sobel mask.

Figure 4 illustrates two images and their respective (vertical) gradient images. It should be noted that objects other than text may also respond to the gradient operator. Hence, the gradient image, in addition to text strokes may also contain many unwanted edges which are removed in the subsequent steps.

2) Mean gradient: The textual content in images occurs in clusters hence a number of studies consider enhancing the magnitude of image gradients in the text regions while suppressing it in the non-text areas. Generally this is achieved by scanning the gradient image with a small window and performing some operations [10], [8]. Authors in [10] exploit this idea using accumulated gradients where the gradient values in a predefined sliding window are accumulated. Shivakumara [8] employed the difference of the maximum and minimum values of pixels in a fixed neighborhood to calculate the value of central pixel in each window. In our study, we slide a horizontal

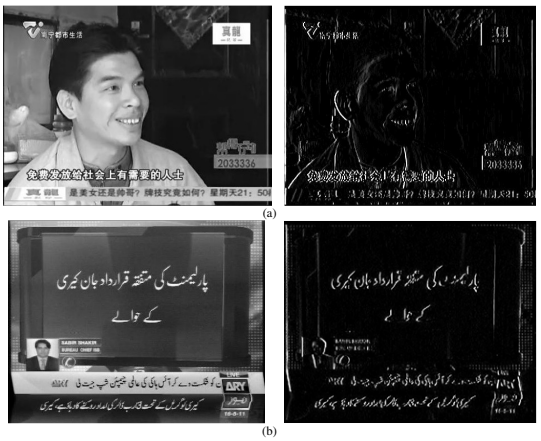


Fig. 4: Vertical gradient images (a) Chinese text (b) Urdu text

window of size $1 \times s$ on the gradient image and replace each pixel with the average of the gradient magnitude in the window [4]. The motivation behind this operation is that edges in text regions appear in clusters. Hence, computing the average gradient in windows over text regions is likely to maintain high values. On the other hand, isolated gradients in the non-text regions, when replaced by the mean of neighboring pixels, are suppressed [4]. Equation 1 summarizes the average gradient operation, s being the size of averaging window which is empirically fixed to 31 in our study.

$$Avg(x,y) = \frac{1}{s}\left[\sum_{j=-s/2}^{s/2} G(x+j,y)\right] \tag{1}$$

The averaged gradient image is binarized to have text or text-like regions as white pixels on black background. Binarization threshold is computed using Otsu's global thresholding algorithm. As a result of binarization, gradients with weak magnitude are removed (become a part of background) and text-like regions are retrained which are merged together by applying morphological operations on the binarized image.

3) Morphological Processing: In order to combine the binarized gradients into larger components, we apply horizontal run-length smoothing algorithm (RLSA). As a result of this, components in the proximity of one another are merged together while the isolated components remain separated. It can be seen from Figure 5 that most of the textual content is merged into large components which correspond to words or groups of words.

4) Foreground Density Filter: Applying the horizontal RLSA to the binarized averaged gradients joins most of the textual elements into larger components. The image, however, still contains non-text components which need to be addressed. Exploiting the same idea that text components appear in clusters, we next employ a density filter on the image using a rectangular sliding window. The window is moved in the top-bottom, left-right fashion and for each position of window the density of foreground (likely text) pixels is computed as.

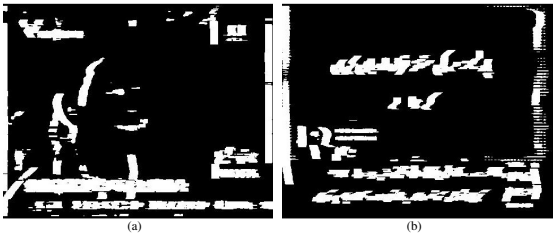


Fig. 5: Application of RLSA to averaged-gradient images (a) Chinese (b) Urdu

$$\text{Foreground Density} = \frac{\text{Number of white pixels}}{\text{Total pixels}} \tag{2}$$

The foreground pixel density is compared to a pre-defined density threshold. If the pixel density at a given window position is greater than the threshold, the central pixel is assigned a value 1, else it is considered a non-text pixel and is assigned a 0.

$$h(x,y) = \left\{ \begin{array}{ll} 1 & \text{if density(x,y)} > t \\ 0 & \text{otherwise} \end{array} \right\} \tag{3}$$

Where t is the density threshold set to 0.8 while the window size is fixed to 10×10 pixels. As evident from Figure 6. The density filter, although effective, does not suppress all

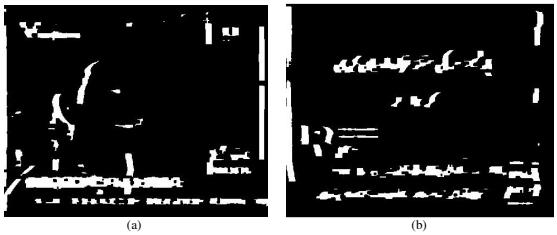


Fig. 6: Images after application of foreground density filter (a) Chinese (b) Urdu

the unwanted non-text regions. We, therefore, apply some geometrical constraints on the detected components to further reduce the false alarms.

5) Geometrical Constraints : With the realistic assumptions that size of the text on the image is large enough to be read by the audience, traditional geometrical constraints are applied to the localized bounding boxes. Another important property, as discussed earlier, is that text components are likely to occur in groups and not in isolation. Similarly, since we target horizontally aligned text, constraints can be applied to the aspect ratio of such text. Components satisfying the empirically determiend thresholds on aspect ratio, minimum height and minimum width are kept as potential text regions while the remaining components are discarded. Figure 7 illustrates the components retained as text after application of geometrical constraints on the two example images used as reference in our description.

Fig. 7: Images after application of geometrical constraints (a) Chinese (b) Urdu

After having discussed the detection of potential text regions using an unsupervised approach, we present the validation mechanism of these detected text rectangles in the next section.

B. Text Validation

The output of the text detector mostly comprises valid text regions. However, some other objects, which exhibit text like properties, are also falsely detected as text regions. The objective of validation step is to take as input each text block localized by the detector and validate it using a supervised approach. This module comprises two phases, training and validation, each of these is discussed in the following.

1) Training : A unique property of text in any script is its texture which can be exploited to distinguish it from other objects or complex backgrounds. Texture information can be captured using a variety of measures. In our implementaiton, we compute a set of features from the Gray Level Co-occurrence Matrices (GLCM) of text and non-text blocks to represent the texture. These features are then used to train a classifier, an artificial neural network in our case, to learn to discriminate text and non-text regions.

Training of the classifier requires samples of text and non-text blocks. We have used a training data set which comprises video images containing textual occurrences; 30 images for each script making a total of 150 images. The text rectangles in each image are manually extracted while rest of the image is considered as non-text region. For each text and non-text rectangle, we divide it into small blocks of 30×50. This gives a large number of text and non-text blocks which constitutes our training data. Some examples of text and non-text blocks can be seen in Figure 8.

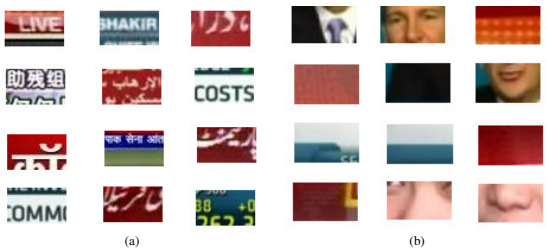


Fig. 8: Blocks used to train the neural network (a) Text blocks (b) Non-text blocks

Each block (text or non-text) is converted to grayscale and a GLCM is computed for each block. The GLCM considers the relationship among two neighboring pixels and determines how frequently different combinations of gray levels co-occur for a given direction and distance. The size of GLCM matrix is the same as the number of gray levels in the image. It is therefore a common practice to quantize the gray levels to have a smaller GLCM. In our implementation, we quantize each block to 64 gray levels and compute the GLCMs using four displacement vectors (offsets). These offsets include (0,1), (1,-1), (0,-1) and (-1,-1) and correspond to four directions $0°$, $45°$, $90°$, $135°$.

Once the GLCMs are computed, several statistics can be computed from each GLCM and could serve as features to characterize the underlying texture of the input image (block). In our study, we compute the contrast, correlation, homogeneity, entropy and energy of each GLCM and use them as features to characterize each block. These statistics are summarized in Table I. These five statistics are computed for each of the four GLCMs $(0°, 45°, 90°, 135°)$ for each training block. Finally, the average of each feature for the four directions is computed giving a 5 dimensional feature vector [34]. These features are fed to a feed forward artificial neural network. In our implementation, we use a neural network with 5 neurons in the input layer (corresponding to five features), 20 neurons in the hidden layer (chosen experimentally) and two neurons in the output layer, each neuron with a sigmoid activation function. The network is trained on 396 text blocks and 938 non-text blocks using back propagation algorithm.

SNo.	Feature	Computational Details
1.	Contrast	$\sum_{i,j=0}^{N-1} P_{i,j} = (i,j)^2$
2.	Correlation	$\sum_{i,j=0}^{N-1} P_{i,j} \left[\frac{(i-\mu_i)(j-\mu_j)}{(\sqrt{(\sigma_i^2)(\sigma_j^2)})} \right]$
3.	Homogenity	$\sum_{i,j=0}^{N-1} P_{i,j} = 0 \frac{P_{i,j}}{i+(i-j)^2}$
4.	Entropy	$\sum_{i,j=0}^{N-1} P_{i,j}(-InP_{i,j})$
5.	Energy	$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [P(i,j)]^2$

TABLE I: Summary of GLCM based features

2) Validation of Text regions: The trained neural network is employed to validate the candidate text regions produced by the detection module. Each detected rectangle is divided into blocks which are fed to the network for classification. If more than 60% of the blocks in a detected rectangle are classified as text, the rectangle is retained as a valid text region. Otherwise, it is considered a false positive and is discarded. This validation step is intended to remove the false alarms and improve the overall precision of the system. A relaxed threshold of 60% is used so that valid text regions are not eliminated during this step and recall of the system is not compromised. The final text rectangles are then separated from the background using the text extraction module discussed in the following.

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

Fig. 9: Text detection and extraction examples in five different languages

C. Text Extraction

Text extraction is the step where the text components are segmented from the background. This step is straight forward if the background is homogenous but can pose difficulties on complex backgrounds. A number of global and local thresholding algorithms have been propsoed to segment text from the background both in scanned document images and video frames [35], [36], [37], [38], [14]. In our implementaiton, we employ the Wolf's algorithm [14] which has been specifically developed for segmentation of video text from the background and is known to work better than many of the binarization algorithms. Examples of text extracted using Wolf's binarization [14] can be seen in Figure 9.

This concludes our discussion on text detection which comprised detection of potential text regions, validation of these regions and segmentation of text from the background. We now present the script identification in the next section.

D. Script Identification

Script identification is aimed at identifying the script of the text detected by the detection module. Literature on script identification of video text is relatively limited as most of the text detection systems have been designed to operate on text in a known language. The existing literature on this subject is mostly on document images only and script identification from text in videos has been a less investigated area. In case of printed and handwritten document images, features at page, paragraph, line and word level have been explored for identification of script [39], [40], [41]. Among recent video text script identification methods, supervised [42] as well as unsupervised [43] techniques have been employed.

For detection of multi-script text, the objective is to find the common properties of text in different scripts and exploit these properties to allow its detection. In script recognition, the objective is to exploit the variations between different scripts. In our study, we consider text in each script as a different texture and employ Local Binary Patterns (LBP) to capture the texture information. The histograms of LBPs computed from texts in different scripts are used to train a neural network which then classifies a given text as being one of the script classes.

1) Local Binary Patterns: Local Binary Patterns, introduced by Ojala [44], [45] for texture classification, have been effectively applied to wide variety of texture classification problems [46], [47], [48], [49]. The original LBP feature [44], [45] considers for each pixel V_0 a set of neighboring pixels. The pixel values of all the neighbors are compared with the value at central pixel. If the value of a neighboring pixel is less than the central pixel, the neighbor is assigned a value of 0, otherwise, it is assigned a 1. The resulting string of 0s and 1s is considered a binary number. The computation of LBP for a reference pixel is illustrated in Figure 10.

In a later study [50], the authors proposed extensions to the original LBP operator to take into account neighborhoods of different sizes. The generalized LBP is represented using the notation (P, R), where P represents the number of neighboring pixels while R is the distance of the neighboring pixels from the central pixel. In addition, based on the number of transitions between 0s and 1s, uniform and non-uniform binary patterns were introduced. LBP codes for which the number of transitions is less than or equal to 2 are considered uniform while those with more than 2 transitions are considered non-uniform [50].

To generate an LBP based descriptor of texture, the LBP is computed for each pixel in the image and the histogram of LBP is used as feature to characterize texture. In our implementation, we compute the $(16, 2)$ LBP from the grayscale images of text blocks with dark text on bright background. For 16 neighboring points, this gives a 243 dimensional feature vector characterizing the texture of each script.

2) Training and Classification: An artificial neural network is used as classifier to recognize the script. The neural network is trained using the same training set that was used to train the network for text validation. Text rectangles from a total of 150 images, with 30 images per script are used as training data. The LBP histogram is computed from each image and the extracted histograms are fed to the network for training. The network comprises 243 neurons in the input layer (same as dimension of the feature vector/histogram), 200 neurons in the hidden layer and 5 neurons in the output layer (corresponding to 5 scripts). For recognition, the LBP histogram is determined from the detected text rectangle and is fed to the network which classifies it as being English, Arabic, Urdu, Hindi or Chinese text.

IV. Experiments and Results

All experiments are carried out on the multi-lingual artificial text database developed at Image Processing Center (IPC) - a research facility at National University of Sciences and Technology (NUST), Pakistan. The database comprises a total of 500 video frames extracted from different news channels, sports videos, talk shows etc. These images contain occurrences of artificial text in five different languages namely English, Arabic, Urdu, Chinese and Hindi with 100 images of each category as the major text of the image. A subset of this data set (images with Urdu text) has been published as [51]. The resolution of the images varies from a minimum of 320x240 to a maximum of 720x576 pixels. Out of the 100 images of each category, 30 images are used as training data (for training the ANN for text region validation and script identification) while 70 are used for testing. The ground truth data for the images was generated by labeling the text occurrences and storing the coordinates of each text rectangle.

Several evaluation metrics have been proposed to evaluate the performance of text localization systems [52], [53]. In our system, we have employed the area based precision and recall measures. Let A_E be the estimated text area given by the system and A_T be the ground truth text area, then the precision P and recall R are defined as:

$$P = \frac{A_E \cap A_T}{A_E} \tag{4}$$

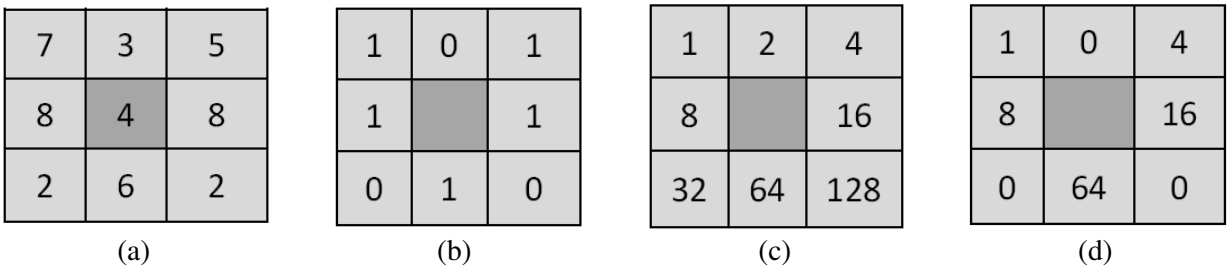


Fig. 10: Calculation of LBP (a): Pixel values (b): Binary codes (c): Weight assignment (d): Decimal number

$$R = \frac{A_E \cap A_T}{A_T} \tag{5}$$

The same idea can be extended to N images to compute the overall precision and recall values. For script recognition experiments, we report the confusion matrix and the overall correct classification rate of the system.

A. Text Detection Results

The text detection module first identifies potential text regions using an unsupervised approach. These candidate text rectangles are then validated by a supervised approach to find the final set of text regions. Detection results, in terms of precision and recall, for both of these are summarized in Table II and Table III respectively. Using the unsupervised detection scheme, an overall precision of 59% and a recall of 89% is achieved. It is interesting to note that the results are consistent across text in different languages demonstrating the generality of the system.

Language	Precision	Recall	F-measure
Urdu	0.58	0.84	0.69
English	0.61	0.92	0.73
Arabic	0.59	0.89	0.71
Chinese	0.60	0.87	0.71
Hindi	0.58	0.94	0.72
Total	**0.59**	**0.89**	**0.71**

TABLE II: Precision and recall of text detection (unsupervised)

It can be seen from Table II that precision values are lower than that of recall values. There are mainly two reasons for this. The first reason is that the system parameters are tuned to achieve high recall and, low values of precision at the detection step are acceptable. The next step of text validation is aimed to reject the false alarms and improve the precision of the system. Since validation cannot detect the text regions which are missed by detection, the recall cannot be improved by the validation step and hence high values of recall are desired at the detection step. The second reason is that we are using an area based metric to compute precision and recall where area represents the number of pixels. Figure 11 illustrates an example of the ground truth text region and the text region detected by the system. Although the system has detected the text but since all three text regions are merged in one big rectangle (having background pixels in the detected region), this results in a low precision.

Language	Precision	Recall	F-measure
Urdu	0.65	0.80	0.72
English	0.68	0.88	0.77
Arabic	0.66	0.85	0.74
Chinese	0.66	0.83	0.73
Hindi	0.60	0.87	0.71
Total	**0.65**	**0.85**	**0.74**

TABLE III: Precision and recall after text validation

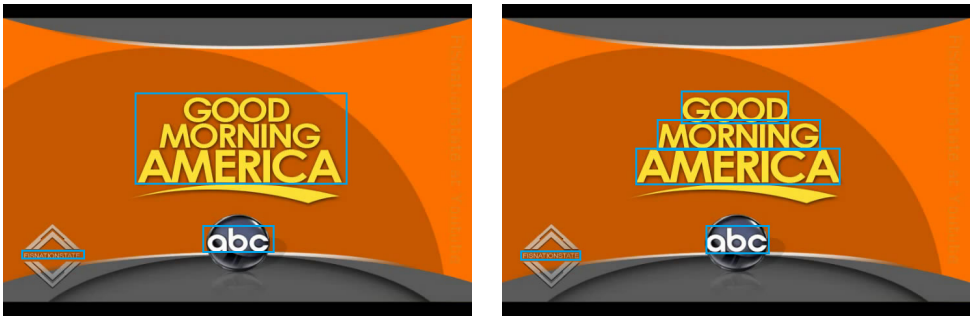
It should be noted that the idea of having a validation step after detection is to enhance the precision of the system by rejecting the regions falsely detected as text. Although precision values in Table III are better than those in Table II, there is a slight decrease in the recall values. This is because while false alarms are reduced by the validation step, some text regions are also eliminated. Overall, however, increased values of F-measure reflect the usefulness of this validation step.

B. Script Identification Results

Script identification is aimed at identifying the script of the text extracted from the images. From the view point of application, script identification module should be fed the output of text detector. However, since the text detection does not extract all the text rectangles, script recognition experiments are carried out on manually extracted text blocks. This allows evaluation of script recognition on all the text blocks in our dataset. Out of a total of 1,448 text blocks, the script of 1,291 blocks was correctly recognized making it a classification rate of 89%. The detailed confusion matrix is illustrated in Table IV where it can be observed that the performance of script identification is more or less consistent across text in different scripts.

	Arabic	**English**	**Urdu**	**Hindi**	**Chinese**	**Total**
Arabic	192	9	11	3	12	227
English	4	349	3	4	20	380
Urdu	2	14	202	2	5	225
Hindi	1	11	2	266	16	296
Chinese	0	20	7	11	282	320

TABLE IV: Script Recognition - Confusion matrix



(a) (b)

Fig. 11: (a) Detected text region (b)Ground truth text region

For script identification, we have used the histogram of local binary patterns using $(16, 2)$ neighborhood ($LBP_{(16,2)}$). By varying the neighborhood size, we study the variation in the classification rate as illustrated in Figure 12. Neighborhoods of (8,1), (8,2), (8,3), (16,1), (16,2) and (16,3) have been considered in out experiments. It can be observed from Figure 12 that the script recognition rates are not very sensitive to the neighborhood size with neighborhoods of 16 pixels naturally performing better than those of 8 pixels.

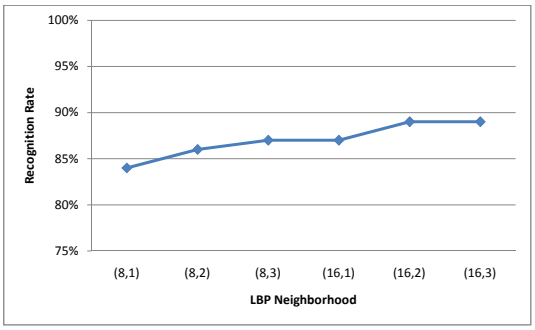


Fig. 12: Script identification rates as a function of different neighborhoods of LBP

We also performed a comparative analysis of the proposed system with well-known existing systems in the literature. The comparison can be carried out for text detection as well as script recognition. Text detection, however, has been evaluated by different metrics in different studies hence a meaningful comparison may not possible. We, therefore, present a comparison of the performance of different script recognition systems in Table V. It can be seen from the table that the database employed, the number of scripts and the number of images in each study is different making it difficult to perform a direct comparison of recognition rates. A maximum of 10 different scripts have been considered in [54] realizing a recognition rate of 91%. The system, however, has been evaluated on 100 test images only. The recognition system in [43] reports a correct classification rate of around 96% on 770 test images which indeed is very promising. Our proposed LBP based technique realizes a recognition rate of 89% on 500 test images in 5 different scripts. These results are comparable with most of the studies and we look to improve them further by introduction of other texture based features to complement the LBP features.

V. Conclusion

This work presented a system for detection of multilingual artificial textual content from video images, an important component for text based indexing and retrieval of videos. Script recognition was also considered in our study. Most of the state-of-the-art approaches for text detection target a single script/language. We have presented a generic text detection system that is not tuned on one particular type of text. The detection is implemented using a combination of unsupervised and supervised techniques. The unsupervised approach relies on image analysis techniques including edge information, morphological processing and geometrical heuristics to detect potential text regions in an image. These candidate text regions are then validated by an artificial neural network that is trained on text and non-text blocks using a set of texture features computed from Gray Level Co-occurrence Matrices (GLCMs). The proposed methodology evaluated on images containing textual occurrences in five different languages (Urdu, Arabic, Hindi, English and Chinese) realized promising results.

We also presented a script recognition module that takes text blocks as input and recognizes the script of the text. Each script is viewed as a different texture and the texture information is captured by computing the histogram of Local Binary Patterns. Recognition is carried out by an artificial neural network trained on text blocks from the five scripts considered in our study. The main idea of this module is to identify the script of the text rectangles detected in the images so that these rectangles can be further processed by their respective recognition engines.

The proposed system which presently targets extraction of text from images and recognition of the script of detected text can be extended to a complete video indexing and retrieval system. This will require either integration of recognition engines (for each of the scripts) or a word spotting based technique allowing indexing of videos on the extracted textual content. The video OCR itself is a challenging problem due to low resolution and complex backgrounds as opposed to document OCRs. Another interesting aspect which could be exploited is the temporal redundancy of text in videos. The

Study	Scripts	Languages	Data set	Overall Recognition Rate
[43]	6	English, Chinese, Japanese, Korean, Arabic and Tamil	770 images	95.71%
[40]	4	Chinese, Japanese, Korean and Roman	3200-3500 characters each	96.95% at character level and 99.85% at block level
[39]	4	English, Urdu, Hindi and Kannada	400 images	97%
[54]	10	Arabic, Cyrillic, Greek, Hebrew, Japanese, Roman, Bengali, Thai, Korean and Chinese	100 images	91%
[55]	3	English, Tamil and Chinese	500 images	51.6%
[42]	3	English, Hindi and Bengali	896 images	87.5%
Proposed method	**5**	**English, Urdu, Hindi, Chinese and Arabic**	**500 images**	**89%**

TABLE V: A comparison of script recognition systems

present system works on static images and does not take into account the redundancy that exists across multiple frames in a video. Integrating the detection results of multiple frames could serve to enhance to overall accuracy of the system. It is expected that the ideas put forward in this research would be helpful to researchers working on video retrieval systems in general and text extraction in particular.

REFERENCES

[1] K. Jung, K. In Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern recognition*, vol. 37, no. 5, pp. 977–997, 2004.

[2] J. Wu, A. Narasimhalu, B. Mehtre, C. Lam, and Y. Gao, "Core: a content-based retrieval engine for multimedia information systems," *Multimedia Systems*, vol. 3, no. 1, pp. 25–41, 1995.

[3] S.-F. Chang and H. Zhang, "Content-processing for video browsing, retrieval, and editing," *Multimedia Systems*, vol. 7, no. 4, pp. 255–255, 1999.

[4] A. Jamil, I. Siddiqi, F. Arif, and A. Raza, "Edge-based features for localization of artificial urdu text in video images," in *Proc. of International Conference on Document Analysis and Recognition*, 2011, pp. 1120–1124.

[5] A. Raza, A. Abidi, and I. Siddiqi, "Multilingual artificial text detection and extraction from still images," in *Proc. of Document Recognition and Retrieval, IS&T/SPIE Electronic Imaging*, 2013, pp. 86 580V–86 580V.

[6] A. Raza, I. Siddiqi, C. Djeddi, and A. Ennaji, "Multilingual artificial text detection using a cascade of transforms," in *Proc. of the 2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 309–313.

[7] S. Antani, D. Crandall, A. Narasimhamurthy, V. Mariano, and R. Kasturi, "Evaluation of methods for detection and localization of text in video," *Proc. of the IAPR workshop on Document Analysis Systems*, pp. 506–514, 2000.

[8] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A laplacian approach to multi-oriented text detection in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 412–419, 2011.

[9] T. B. Chen, D. Ghosh, and S. Ranganath, "Video-text extraction and recognition," in *Proc. of IEEE Region 10 Conference (TENCON)*, vol. 1, 2004, pp. 319–322.

[10] C. Wolf, J.-M. Jolion, and F. Chassaing, "Text localization, enhancement and binarization in multimedia documents," in *Proc. of the International Conference on Pattern Recognition*, 2002, pp. 1037–1040.

[11] L. Minhua and B. Meng, "A mixed edge based text detection method by applying image complexity analysis," in *Porc. of the 10th World Conference on Intelligent Control and Automation*, 2012, pp. 4809–814.

[12] P. Dubey, "Edge based text detection for multi-purpose application," in *Porc. of the 8th International Conference on Signal Processing*, 2006.

[13] A. Ikica and P. Peer, "An improved edge profile based method for text detection in images of natural scenes," in *Proc. of International Conference on Computer as a Tool (EUROCON)*, 2011.

[14] C. Wolf and J.-M. Jolion, "Extraction and recognition of artificial text in multimedia documents," *Pattern Analysis and Applications*, vol. 6, no. 4, pp. 309–326, 2004.

[15] C. Zhu, W. Wang, and Q. Ning, "Text detection in images using texture feature from strokes," in *Advances in Multimedia Information Processing*, ser. Lecture Notes in Computer Science, 2006, pp. 295–301.

[16] V. Wu, R. Manmatha, and E. Riseman, "Textfinder: an automatic system to detect and recognize text in images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1224–1229, 1999.

[17] Z. Li, G. Liu, X. Qian, D. Guo, and H. Jiang, "Effective and efficient video text extraction using key text points," *IET Image Processing*, vol. 5, no. 8, pp. 671–683, 2011.

[18] X. Qian and G. Liu, "Text detection, localization and segmentation in compressed videos," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, pp. 385–388.

[19] W. Fan, J. Sun, Y. Katsuyama, Y. Hotta, and S. Naoi, "Text detection in images based on grayscale decomposition and stroke extraction," in *Proc. of the Chinese Conference on Pattern Recognition*, 2009.

[20] A. Srivastav and J. Kumar, "Text detection in scene images using stroke width and nearest-neighbor constraints," in *Proc. of the IEEE Region 10 Conference (TENCON)*, 2008.

[21] M. Kumar, Y. C. Kim, and G.-S. Lee, "Text detection using multilayer separation in real scene images," in *Proc. of the 10th International Conference on Computer and Information Technology*, 2010, pp. 1413–1417.

[22] J. Yi, Y. Peng, and J. Xiao, "Color-based clustering for text detection and extraction in image," in *Proc. of the 15th International Conference on Multimedia*, 2007, pp. 847–850.

[23] D. Lopresti and J. Zhoum, "Extracting text from www images," in *Proc. of the 4th International Conference of Document Analysis and Recognition*, 1997, pp. 248–252.

[24] C. Thillou and B. Gosselin, "Combination of binarization and character segmentation using colour information," in *Proc. of the 4th IEEE International Symposium on Signal Processing and Information Technology*, 2004, pp. 107–110.

[25] Y. Liu, S. Goto, and T. Ikenaga, "A robust algorithm for text detection in color images," in *Proc. of 8th International Conference on Document Analysis and Recognition*, 2005, pp. 339–403.

[26] R. Wang, W. Jin, and L. Wu, "A novel video caption detection approach using multi-frame integration," in *Proc. of International Conference on Pattern Recognition*, 2004, pp. 449–452.

[27] K. I. Kim, K. Jung, and J.-H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously

adaptive mean shift algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1631–1639, 2003.

[28] X. Liu, H. Fu, and Y. Jia, "Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images," *Pattern Recognition*, vol. 41, no. 2, pp. 484–493, 2008.

[29] J. Ye, L.-L. Huang, and X. Hao, "Neural network based text detection in videos using local binary patterns," in *Proc. of the Chinese Conference on Pattern Recognition*, 2009.

[30] J. Yu and Y. Wang, "Apply som to video artificial text area detection," in *Proc. of the 4th International Conference on Internet Computing for Science and Engineering (ICICSE)*, 2009, pp. 137–141.

[31] C. Shin, K. Kim, M. Park, and H. J. Kim, "Support vector machine-based text detection in digital video," in *Proc. of IEEE Signal Processing Society Workshop Neural Networks for Signal Processing*, vol. 2, 2000, pp. 634–641.

[32] M. Anthimopoulos, B. Gatos, and I. Pratikakis, "A hybrid system for text detection in video frames," in *Proc. of the 8th IAPR International Workshop on Document Analysis Systems*, 2008, pp. 286–292.

[33] G. Miao, Q. Huang, S. Jiang, and W. Gao, "Coarse-to-fine video text detection," in *Proc. of IEEE International Conference on Multimedia and Expo*, 2008, pp. 569–572.

[34] T. A. Pham, "Optimization of texture feature extraction algorithm," Ph.D. dissertation, MSc Thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2010.

[35] W. Niblack, *An introduction to digital image processing*. Strandberg Publishing Company, 1985.

[36] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern recognition*, vol. 33, no. 2, pp. 225–236, 2000.

[37] C. Y. Graham Leedham, K. Takru, J. H. N. Tan, and L. Mian, "Comparison of some thresholding algorithms for text/background segmentation in difficult document images," in *Proc. of the 7th International conference on document analysis and recognition*, vol. 2, 2003, pp. 859–864.

[38] M. Feng and Y.-P. Tan, "Contrast adaptive binarization of low quality document images." *IEICE Electronic Express*, vol. 1, no. 16, pp. 501–506, 2004.

[39] B. V. Dhandra, P. Nagabhushan, M. Hangarge, R. Hegadi, and V. S. Malemath, "Script identification based on morphological reconstruction in document images," in *Proc. of 18th International Conference on Pattern Recognition*, vol. 2, 2006, pp. 950–953.

[40] S. Chanda, U. Pal, K. Franke, and F. Kimura, "Script identification: A han and roman script perspective," in *Proc. of the 20th International Conference on Pattern Recognition*, 2010, pp. 2708–2711.

[41] D. Ghosh, T. Dube, and A. P. Shivaprasad, "Script recognition:a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2142–2161, 2010.

[42] N. Sharma, S. Chanda, U. Pal, and M. Blumenstein, "Word-wise script identification from video frames," in *Proc. of the 12th International Conference on Document Analysis and Recognition*, 2013, pp. 867–871.

[43] D. Zhao, P. Shivakumara, S. Lu, and C. Tan, "New spatial-gradient-features for video script identification," in *Proc. of the 10th IAPR International Workshop on Document Analysis Systems*, 2012, pp. 38–42.

[44] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proc. of the 12th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision and Image Processing*, 1994, pp. 582–585.

[45] ——, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[46] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 6, pp. 765–781, 2011.

[47] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Texture-based descriptors for writer identification and verification," *Expert Systems with Applications*, vol. 40, no. 6, pp. 2069–2080, 2013.

[48] X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *Proc. of the IEEE 12th International Conference on Computer Vision*, 2009, pp. 32–39.

[49] I. Siddiqi, C. Djeddi, A. Raza, and L. Souici-meslati, "Automatic analysis of handwriting for gender classification," *Pattern Analysis and Applications*, 2014.

[50] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[51] I. Siddiqi and A. Raza, "A database of artificial urdu text in video images with semi-automatic text line labeling scheme," in *Proc. of the 4th International Conference on Advances in Multimedia*, 2012, pp. 75–81.

[52] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, and X. Lin, "Icdar 2003 robust reading competitions: entries, results, and future directions," *International Journal of Document Analysis and Recognition*, vol. 7, pp. 105–122, 2005.

[53] C. Wolf and J.-M. Jolion, "Object count/area graphs for the evaluation of object detection and segmentation algorithms," *International Journal on Document Analysis and Recognition*, vol. 8, no. 4, pp. 280–296, 2006.

[54] L. Li and C. L. Tan, "Script identification of camera-based images," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[55] T. Q. Phan, P. Shivakumara, Z. Ding, S. Lu, and C. L. Tan, "Video script identification based on text lines," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 1240–1244.