Journal of Engineering and Applied Sciences 12 (Special Issue 12): 9483-9489, 2017
ISSN: 1816-949X
© Medwell Journals, 2017

News Ticker Recognition: Performance Analysis SIFT and M-SIFT

Pooja and Renu Dhir
Department of Computer Science and Engineering, NIT Jalandhar, Punjab, India

Abstract: This study explores the use of Scale Invariant Feature Transform (SIFT), Modified-Scale Invariant Feature Transform (M-SIFT) for feature extraction of text, typically the punjabi text, however, this area is mature enough for languages like English, Chinese and Arabic. This study presents an ongoing effort to build fusion based text extraction approach using SIFT also variations in SIFT are experimented and it is denoted as M-SIFT. Fused image is obtained using principal component analysis based fusion method. From within the fused image, region of interest is perceived to find the part of frame which contains text in it. Region, thus, supposed is cropped for feature extraction. Feature extraction is carried out by using SIFT and M-SIFT based algorithm separately. This study addresses the problems in the various stages of the development of a recognition system for Punjabi text in news ticker present in Punjabi news videos.

Key words: Video image, OCR, text recognition, PCA, SIFT, modified SIFT

INTRODUCTION

With the increase of digital media resources, content-based image and video retrieval will be a long-term research. Such a text extraction system can be instrumental in video categorization, retrieval, automatic content-based video indexing, content analysis, annotation or in providing an additional source of information to a retrieve and rank system, it can provide useful semantic information for semantic analytics. More generally, text in natural images provides a rich source of information about the underlying image or scene. Text extracted from a video sequence provides natural, meaningful keywords that reflect the video's content. Text which includes plentiful semantic information can do some contributions to the retrieval tasks (Ye *et al*., 2003). Text in news videos is of great importance for the semantic analytics and for the analysis of political events. As well as text in videos can be a clues for decoding the video's structure and for classification. Video text is important in applications such as navigation, surveillance, video classification or analysis of sporting events. Text extracted from video clips can provide meaningful keywords which can reflect the rough content of video. These keywords can be used for indexing and summarizing the content of video clip (Ghorpade *et al*., 2011). Also when it comes to making documentaries may be for students and adding news clippings in such documentaries such text recognition system is of great use. A system that can locate and recognize news text in video images of a news channel has many practical applications. For instance, such a system can be instrumental in video categorization, retrieval, automatic content-based video indexing, content analysis, annotation or in providing an additional source of information to a retrieve and rank system, it can provide useful semantic information for semantic analytics. More generally, text in natural images provides a rich source of information about the underlying image or scene. Text extracted from a video sequence provides natural, meaningful keywords that reflect the video's content. At the same time, however, text recognition in video frames/images has its own set of difficulties. While state-of-the-art methods generally achieve nearly perfect performance on Object Character Recognition (OCR) for scanned documents, the more general problem of recognizing text in video images is far from solved. A study of the literature as done in our study (Dhir, 2016) reveals that no complete video text extraction system has been developed. Additionally, it is seen that no single algorithm is robust for detection of an unconstrained variety of text appearing in the video. Most methods have been developed to extract text from complex color images and have been extended for application to video data. However, these methods do not take advantage of the temporal redundancy in video. The foremost challenge in extracting news ticker from the video is to separate out frames from the video. A small video clip of say 4-5 MB size may have more than 3000 frames in it. All the frames may not be containing the news ticker as for a while the video screens of news channels do not have news ticker. Extracting the relevant frames is a big issue. Extracting text from the frames is also a challenge because of background contrast and color bleeding. Text quality is decreased due

Corresponding Author: Pooja, Department of Computer Science and Engineering, NIT Jalandhar, Punjab, India

to the presence of noise and image encoding and decoding procedures. Usually, the news tickers are scrolling on the screen due to which texts on different frames may not be same there are chances that the text at the sides of the frames will be broken or skewed. The resolution of the text on the screen is low is also an issue as OCRs accepts 300 dpi resolution text for getting good text detection results.

Characteristics of news ticker and challenges: A news ticker is also named as "crawler" or "slide". This term mainly referred to the screen space on television news networks where the news headlines or other text content is represented and displayed. News ticker resides usually in the lower third of the television screen space. It appears on the screen at some particular position, e.g., horizontally at the lower part of the screen and usually in news videos it appears at the lower ¼ th part of the screen. Sometimes thin scoreboard-style display seen outside some offices or public buildings is also referred as news ticker. Most of the news tickers used in India are scrolling ones. Some television channels also display news ticker with different effects like flipping effect as well. The text in the news ticker is added artificially into a video and is the artificial text. Caption text is artificially superimposed on the video at the time of editing. It is also referred to as superimposed text or caption text.

Extraction of text from video presents distinctive challenges over OCR of document images. Such challenges comprise:

- Lower resolution: video frames are typically captured at resolutions of 320×240 or 640×480 pixels while document images are typically digitized at resolutions of 300 dpi or more
- Unspecified text color: text can have random and non-uniform color
- Unknown text size, position, orientation and layout: captions lack the structure usually associated with documents
- Unconstrained background: the background can have colors similar to the text color. The background may include streaks that like that of character strokes
- Color bleeding: lossy video compression may cause colors to run together
- Low contrast: low bit-rate video compression can result into undistinguishable contrast between character strokes and background

Framework for recognition: In the study, Kaur *et al*. (2016) various text recognition techniques, methods and its applications are presented while in study of Leng and Jinhua (2014), we presented a text region extraction approach based on feature analysis. This scheme consists of 5 main phases that are frame extraction, edge detection and binarization, fusion, normalization and feature extraction. Full text recognition system is shown in Fig. 1. For frame extraction, MATLAB simulator is used where video reader is used to extract frames from the video according to the length of video.

Frame extraction: Set of frames together running at particular speed become a video and to know more about the video these frames are required to be extracted out of video as individual images, so that, further image processing can be done for information retrieval. Internet is considered for obtaining news videos for experimenting with the methodology proposed and Punjabi news channel videos are downloaded (freely available) (Choudhary and Renu, 2017).

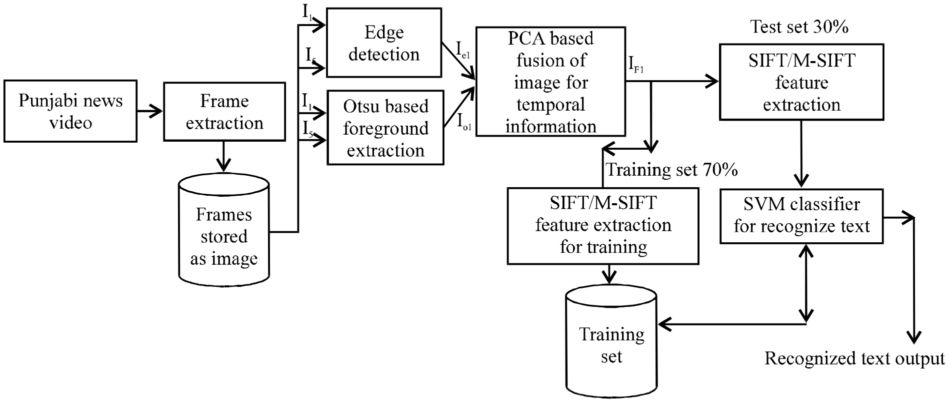
Punjabi news video
Frame extraction
Frames stored as image
I1
I5
I1
I5
Edge detection
Otsu based foreground extraction
Ie1
Io1
PCA based fusion of image for temporal information
IF1
Test set 30%
SIFT/M-SIFT feature extraction
Training set 70%
SIFT/M-SIFT feature extraction for training
Training set
SVM classifier for recognize text
Recognized text output

Fig. 1: Framework for news ticker (Punjabi) text recognition

Algorithm for frame extraction: Following is the algorithm using which frames are extracted from collected videos:

```
Step 1:Initialize count = 0 and input the video
Step 2: Find properties viz. frame rate, duration, number of frames,
height, width of the video selected
Step 3: Calculating the start (startF) and end(endF) frame indices and
the number of frames to skip each time(stepF)
Step 4: for k = startF:stepF:endF
        Read frame one at a time
                  count = count+1
                   Save image as 'jpg/png'
        end
Step 5: Exit
```

As only text is to be considered, here, we experimented with sobel edge detection as sobel edge detection is a very powerful filter to extract horizontal and vertical edges as that are in text. And the other is Otsu algorithm, another best algorithm based on threshold to detect even the squeeze and broken edges. Otsu extracts foreground from background, so, the remaining curve like edges are taken cared by it. Blending the outcomes of both these algorithms gives a perfect image for detecting text lines from the frame. Fusion of these two images is done using principal component analysis (Choudhary and Renu, 2016 a, b).

Algorithm for fusion of images:

```
Step 1: Input two image
       im1 = sobel edge detected image
       im2 = Otsu extracted image
 Step 2: Calculate covariance (covar) of two images for computing PCA
       C = convar([im1(:) im2(:)])
 Step 3: Compute Eigen values and Eigen vectors of the covariance matrix
  [V, D] = eig(C)
        (D)-Matrices of Eigenvalues
       (V)-Eigenvectors of matrix C
 Step 4: Calculate PCA
 if D(1,1) > = D(2, 2)
 PCA = V(:,1)./sum(V(:,1))
 else
         PCA = V(:,2)./sum(V(:,2))
 end
 Step 5: Fuse images using PCA
 Fused image = Sum(PCA(1)*im1+PCA(2)*im2)
```

After fusion for feature extraction, the line of text detected is segmented into words using connected components. Both SIFT and M-SIFT features are dig out for each segmented word. Next section explains more on SIFT and what all parameters are modified for M-SIFT.

MATERIALS AND METHODS

Feature extraction SIFT and M-SIFT techniques: In 2004, Lowe presented SIFT for extracting invariant features from images that can be invariant to image scale and rotation that are distinctive. It was widely used in image recognition, retrieval. Bay in 2006, introduced speeded up robust features technique and used integral images for the convolutions of images and also Fast-Hessian detector. Both approaches do not only detection of interest points or so called features but also solve the propose for the creation of invariant descriptor. This descriptor can be used to identify the found interest points and match under a variety of disturbing conditions, like changes in scaling, rotation, changes in illumination as well as viewpoints. There are also many other feature of detection methods as edge detection, corner detection, etc. Different methods have their own advantages.

SIFT features: SIFT algorithm obtained from the theory of scale-space is a local feature extraction algorithm. Looking at the outermost point, extract the location, scale and rotation invariant in the scale space. The principle diagram of the algorithm cab be divided into following steps:

- Scale-space extrema detection
- Keypoint localization
- Orientation assignment
- Keypoint descriptor

The SIFT algorithm locates the points in a picture which are invariant to shift and scale. These points are spoken to by orientation invariant component vector. An effective algorithm can separate a substantial number of elements from the typical pictures. These components are exceedingly unmistakable subsequently, a solitary element is accurately coordinated with high likelihood against a vast database of elements. Filter components are normally utilized for the protest acknowledgment and have scarcely been utilized for face acknowledgment. Filter elements are invariant to scale, pivots, interpretations and light changes. The SIFT calculation has four stages: extrema detection, removal of key points with low differentiation, orientation task and descriptor calculation (Leng and Jinhua, 2014; Neeru and Kaur, 2016).

Scale-space extrema detection: To recognize blob structures in an image, a scale space is constructed where the points of interest which are known as key points in the SIFT framework are detected. The function of scale space is produced from the convolution of a variable-scale Gaussian, G(x, y, σ^2) with an input image, I(x, y):

$$L(x, y, \sigma^2) = G(x, y, \sigma^2)*I(x, y) \tag{1}$$

As studied by Lindeberg, the Laplacian normalization, σ^2 σ^2G with the factor σ^2 is needed for true scale invariance. In this manner automatic scale selection is done in the image after convolution with the Laplacin function which is normalized:

$$O(x, y, \sigma^2) = \sigma^2\Delta^2G*I(x, y) = \sigma^2\Delta^2L(x, y, \sigma^2) \quad (2)$$

As demonstrated in equation above, if the image structure scale is very near to the normalized Laplacian function value, the output $O(x, y, \sigma^2)$ calculated from convolution of the image with $\sigma^2\Delta^2G$ will be extremum. In this way to recognize blob structures and represent them at the most ideal scale, the points which are extrema in both spatial and scale spaces are chosen.

Unreliable key point's removal: In this step, the values of $|O(x, y, \sigma^2)|$ at each candidate key point is evaluated. If this value is below some threshold which means that the structure has low contrast (and is therefore, sensitive to noise), the key point will be removed. For poorly denned peaks in the scale-normalized Laplacian of Gaussian operators, the ratio of principal curvatures of each candidate key point is evaluated. If the ratio is below some threshold, the key point is kept.

Orientation assignment: In this step, each key point is assigned one or more orientations based on local image gradient directions.

Key point descriptor: The image gradient magnitudes and orientations are sampled around the key point location using the scale of the key point to select the level of Gaussian blur for the image. The feature descriptor is computed as a set of orientation histograms on 16×16 pixel neighbourhoods around the key point. Each histogram contains 8 bins and each descriptor contains a 4×4 array of histograms around the key point. Therefore, the feature vector for each key point is 4×4×8 = 128 dimension (Leng and Jinhua, 2014; Gupta and Garg, 2014).

Modified SIFT features

Assessments used in original SIFT: Number of octaves can be calculated by formula:

- O = floor (log2 (min (M, N)))-omin-3
- Scale space (S) = 3
- sigma0 = 1.6*2^ (1/S) (App. 2.0) % smooth lev. -1 at 1.6
- sigman = 0.5

Alterations in SIFT as used in this research: Number of octaves can be calculated by formula:

- O = floor (log2(min(M,N)))-omin-4
- Scale space(S) = 5
- sigma0 = 1.6*2^(1/S) (App. 1.8)
- sigman = 1.414

In this research, we have used sigma value that is greater than the previous used values because greater sigma value, greater blurr and greater blurr generate more key points and it gives better results.

Experimental outcomes: In this study, step by step outcomes are represented for the proposed system and then comparisons are made based on recognition rate between two feature extraction approaches that is SIFT and M-SIFT.

Step 1; Extracting frames from video results: Format .mp4, size: 6.03 MB, length: 1 min 10 sec, duration 101.953 sec, frame width 480, frame height 360, frame rate: 29 fps, duration considered: 0-101 sec, total frames calculated: 3027.

Trial 1st: Time duration taken in between two frames to be extracted: 2.68 sec (picked randomly), i.e., frames extracted are: every 81st frames and total frames extracted: 38 (480×360).

Trial 2nd: Unitary method is applied to find the value of time slot in between two frames to be extracted. Time duration taken in between two frames to be extracted: 0.344 sec, i.e., frames extracted are: every 11th frames and total frames extracted: 276 (480×360).

Trial 3rd: Time duration taken in between two frames to be extracted: 0.172 sec, i.e., frames extracted are: every 6th frames and total frames extracted: 505 (480×360).

Trial 4th: Time duration taken in between two frames to be extracted: 0.086 sec, i.e., frames extracted are: every 4th frames and total frames extracted: 757.

After analyzing these extracted frames, it is concluded that news stays on the screen for hundreds of frames. So, considering every 6th frame lead to good enough results and speeding up the task. This is represented in 'bold' in Table 1.

Step 2; Edge detection and Otsu results: This step is implemented in two sub-steps. Firstly, sobel filter is used for edge detection as it is a powerful filter to extract horizontal and vertical edges and Otsu is used for foreground detection, another best algorithm based on threshold to detect even the squeeze and broken edges. Further, the outcomes of those two algorithms are fussed into one image by using principal component analysis technique. This fusion based extraction of text is used

Table 1: Tabular representation of frame extraction instances from the video

Frame extraction instances	Values
Video	Video 1 (ABP Sanjha)
Duration of video (sec)	101.953
Duration considered for extraction	0-101
Frame rate (fps)	29
Total number of frames	3027
No. of frames (each 81st)	38
No. of frames (each 11th)	256
No. of frames (each 6th)	505*
Number of frames (each 4th)	757
Frame size	480×360

*After analyzing these extracted frames, it is concluded that news stays on the screen for hundreds of frames. So, considering every 6th frame lead to good enough results and speeding up the task

Table 2: Actual dataset created from the frames collected considering every 6th frame

Video channe l	Training setframes	Test set frames	Total frames
Abpsanjha	339	120	459

Table 3: Depiction of segmented words and comparing the number of features collected using SIFT and M-SIFT: case 1

Word segments	Index	SIFT	M-SIFT
L1-word 1	1	128×2	128×14
L1-word 2	2	128×4	128×7
L1-word 3	3	128×8	128×11
L1-word 4	4	128×7	128×21
L1-word 5	5	128×5	128×3
L1-word 6	6	128×5	128×11
L1-word 7	7	128×1	128×2
L1-word 8	8	128×8	128×8

Table 4: Depiction of segmented words and comparing the number of features collected using SIFT and M-SIFT: case 2

Word segment	Index	SIFT	M-SIFT
L2-word 1	1	128×15	128×16
L2-word 2	2	128×2	128×4
L2-word 3	3	128×3	128×9
L2-word 4	4	128×4	128×6
L2-word 5	5	No. of feature collected	128×4
L2-word 6	6	128×4	128×11
L2-word 7	7	128×4	128×
L2-word 8	8	128×4	128×
L2-word 9	9	128×6	128×4

further for feature extraction. Figure 2 shows edge map of frame, Fig. 3a, b show Otsu applied image and fused image, respectively.

Step 3; Feature extraction and training results: Two sentences of Gurumukhi news ticker, L1 and L2, are considered for this study to demonstrate the results. Table 2-4 are displaying the SIFT and M-SIFT features extracted for words of these sentences. The sample notation used is as follows: L1 refers to Line 1, L2 refers to line 2, W refers to word, S refers to sample (Fig. 4).

Step 4; Recognition results: Table 5 and 6 represents the recognition output of SIFT for case 1 and 2, respectively. The sample notation used is as follows: L1 refers to Line 1, L2 refers to line 2, W refers to word, S refers to sample.

Fig. 2: Edge detection of considered frame

Fig. 3: a) Otsu output of considered frame and b) Fused image (principal component analysis based)

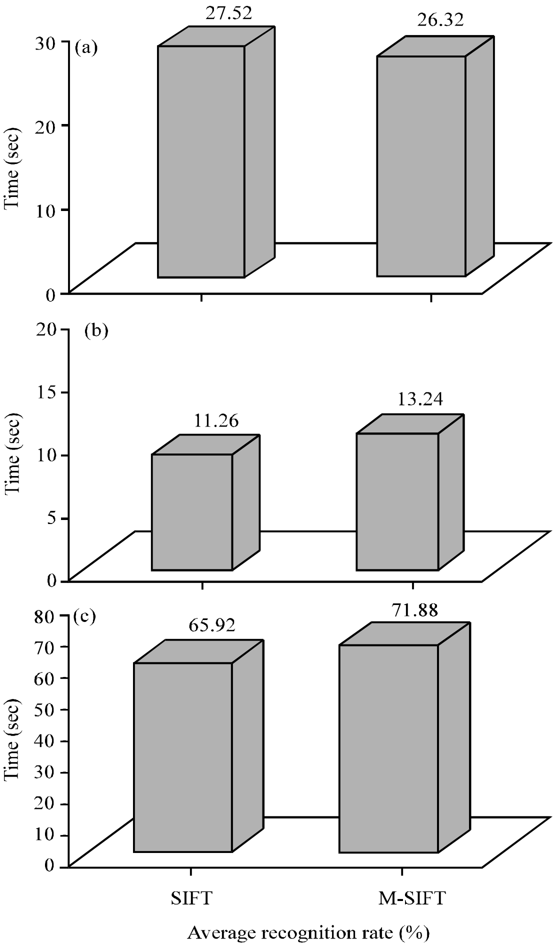

Fig. 4: a) Average training time comparison of SIFT and M-SIFT; b) Average testing time comparison of SIFT and M-SIFT and c) Graphical representation of average recognition rate of SIFT and M-SIFT

Table 7-9 represent the recognition output of M-SIFT for case 1 and 2, respectively. The sample notation used is as follows: L1 refers to Line 1, L2 refers to line 2, W refers to word, S refers to sample.

Table 5: Training time comparison of SIFT and M-SIFT

Techniques	Case 1 (sec)	Case 2 (sec)
SIFT	29.7724	25.2773
M-SIFT	27.1844	25.4477

Table 6: Recognition output of SIFT for 31 samples of case 1

Test sample	Matches found	Matched index	Actual index	Time (sec)
L1_W1_S1	1, 2, 4, 5, 7	1	1	9.6913
L1_W2_S1	2, 4, 5, 7	2	2	9.9966
L1_W3_S1	2, 3, 4, 6, 8	3	3	10.7434
L1_W4_S1	2, 3, 4, 6, 8	4	4	14.3151
L1_W5_S1	2, 4, 5, 6, 7	5	5	10.0446
L1_W6_S1	2, 4, 5, 6, 7	6	6	9.0432
L1_W7_S1	1, 2, 4, 5, 7	7	7	9.9443
L1_W7_S2	1, 2, 4, 5, 7	7	7	11.5352
L1_W8_S1	2, 3, 4, 6, 8	8	8	10.0979
L1_W1_S2	1, 2, 4, 5, 7	1	1	11.1356
L1_W1_S3	1, 2, 4, 5, 7	1	1	10.7175
L1_W1_S4	1, 2, 4, 5, 7	2	1	14.2278
L1_W2_S2	1, 2, 4, 5, 7	2	2	10.3245
L1_W2_S3	No feature collected for this sample			
L1_W8_S2	1, 3, 4, 6, 8	8	8	9.8879
L1_W8_S3	2, 3, 4, 6, 8	3	8	13.9389
L1_W2_S4	2, 4, 5, 6, 7	2	2	18.3418
L1_W3_S2	2, 4, 5, 6, 8	4	3	9.4451
L1_W3_S3	2, 4, 5, 6, 8	5	3	10.2461
L1_W3_S4	2, 3, 4, 6, 8	3	3	12.4536
L1_W3_S5	1, 3, 4, 6, 8	3	3	13.3319
L1_W3_S6	2, 3, 4, 6, 8	3	3	11.8549
L1_W4_S2	2, 4, 5, 6, 8	4	4	12.2607
L1_W4_S3	2, 4, 5, 6, 7	4	4	12.0581
L1_W4_S4	2, 4, 5, 6, 8	4	4	14.1898
L1_W4_S5	2, 4, 5, 6, 7	4	4	11.3737
L1_W5_S2	2, 4, 5, 6, 8	6	5	20.2454
L1_W6_S2	2, 3, 4, 6, 8	3	6	15.6558
L1_W6_S3	1, 2, 4, 5, 7	2	6	13.2154
L1_W7_S3	No feature collected for this sample			
L1_W8_S4	2, 4, 5, 6, 8	5	8	11.2724

*L1: Line 1; W; Word; S: Sample

Table 7: Recognition output of SIFT for 23 samples of case 2

Test sample	Matches found	Matched index	Actual index	Time (sec)
L2_W1_S1	1, 3, 4, 6, 8, 9	1	1	10.5775
L2_W2_S1	2, 3, 5, 7, 8	2	2	10.9266
L2_W3_S1	2, 3, 4, 6, 7, 8	3	3	10.1314
L2_W4_S1	2, 3, 4, 6, 7, 8	4	4	10.4763
L2_W5_S1	No feature collected for thissample			
L2_W6_S1	2, 3, 4, 6, 8	6	6	11.4859
L2_W7_S1	2, 3, 4, 6, 7, 8	7	7	10.3234
L2_W8_S1	2, 3, 4, 6, 8	8	8	11.4096
L2_W9_S1	2, 3, 4, 6, 8, 9	9	9	11.7933
L2_W1_S2	2, 3, 4, 6, 8	2	1	13.0629
L2_W1_S3	1, 3, 4, 6, 8, 9	1	1	12.7755
L2_W1_S4	2, 3, 4, 6, 7, 8	7	1	11.9580
L2_W2_S2	2, 3, 5, 7, 8	2	2	10.3758
L2_W2_S3	2, 3, 4, 6, 7, 8	3	2	11.4978
L2_W6_S2	2, 3, 4, 6, 8	6	6	10.8839
L2_W6_S3	2, 3, 4, 6, 8	6	6	14.2653
L2_W7_S3	No feature collected			
L2_W7_S3	2, 3, 5, 7, 8	2	7	14.9669
L2_W8_S2	2, 3, 4, 6, 7, 8	3	8	12.9709
L2_W8_S3	No feature collected			
L2_W9_S2	2, 3, 4, 6, 8, 9	9	9	10.7461
L2_W9_S3	1, 3, 4, 6, 8, 9	1	9	11.6461
L2_W9_S4	2, 3, 4, 6, 8, 9	9	9	16.9709

*L2: Line 2, W: Word, S: Sample

Table 8: Recognition output of M-SIFT for 31 samples of case 1

Test sample	Matches found	Matched index	Actual index	Time (sec)
L1_W1_S1	1, 3, 5, 6, 8	1	1	12.2199
L1_W2_S1	2, 3, 5, 6, 8	2	2	11.0868
L1_W3_S1	2, 3, 5, 6, 8	3	3	18.3323
L1_W4_S1	2, 3, 4, 6, 8	4	4	11.4603
L1_W5_S1	1, 2, 3, 5, 7	5	5	12.7129
L1_W6_S1	2, 3, 5, 6, 8	6	6	11.0480
L1_W7_S1	1, 2, 3, 5, 7	7	7	11.4346
L1_W7_S2	1, 2, 3, 5, 7	7	7	11.4846
L1_W8_S1	2, 3, 5, 6, 8	8	8	11.0496
L1_W1_S2	2, 3, 5, 6, 8	2	1	15.7142
L1_W1_S3	1, 3, 5, 6, 8	1	1	13.7311
L1_W1_S4	2, 3, 5, 7	2	1	12.2578
L1_W2_S2	2, 3, 4, 6, 8	2	2	16.3819
L1_W2_S3	2, 3, 5, 7	2	2	16.1598
L1_W2_S4	2, 3, 4, 6, 8	2	2	14.4040
L1_W3_S2	2, 3, 4, 6, 8	2	3	21.7222
L1_W3_S3	2, 3, 5, 6, 8	6	3	12.5051
L1_W3_S4	2, 3, 5, 6, 8	3	3	20.5605
L1_W3_S5	2, 3, 5, 6, 8	3	3	12.0124
L1_W3_S6	2, 3, 5, 6, 8	3	3	11.3430
L1_W4_S2	2, 3, 4, 6, 8	4	4	10.5765
L1_W4_S3	2, 3, 4, 6, 8	4	4	14.7620
L1_W4_S4	2, 3, 5, 6, 8	6	4	12.7746
L1_W4_S5	2, 3, 4, 6, 8	4	4	12.7620
L1_W5_S2	2, 3, 5, 6, 8	6	5	27.4298
L1_W6_S2	2, 3, 5, 6, 8	6	6	11.9786
L1_W6_S3	2, 3, 5, 6, 8	8	6	16.5878
L1_W7_S3	1, 2, 3, 5, 7	7	7	13.5227
L1_W8_S2	1, 2, 4, 6, 8	4	8	12.8610
L1_W8_S3	2, 3, 5, 6, 8	8	8	12.8320
L1_W8_S4	2, 3, 4, 6, 8	4	8	18.8619

*L1: Line 1; W; Word; S: Sample

Table 9: Recognition output of M-SIFT for 23 samples of case 2

Test sample	Matches found	Matched index	Actual index	Time (sec)
L2_W1_S1	1, 3, 4, 6, 8, 9	1	1	11.5413
L2_W2_S1	2, 4, 5, 7, 8	2	2	12.9016
L2_W3_S1	2, 3, 4, 6, 8	3	3	11.3166
L2_W4_S1	2, 4, 5, 7, 8	4	4	10.7923
L2_W5_S1	2, 4, 5, 7, 8	5	5	10.1722
L2_W6_S1	2, 3, 4, 6, 8	6	6	13.6928
L2_W7_S1	2, 3, 4, 7, 8	7	7	11.5332
L2_W8_S1	2, 3, 5, 7, 8	8	8	9.07580
L2_W9_S1	1, 3, 4, 6, 8, 9	9	9	11.9213
L2_W1_S2	1, 3, 4, 6, 8, 9	1	1	13.6481
L2_W1_S3	1, 3, 4, 6, 8, 9	1	1	13.4432
L2_W1_S4	1, 3, 4, 6, 8, 9	1	1	13.3159
L2_W2_S2	2, 4, 5, 7, 8	2	2	10.4955
L2_W2_S3	2, 4, 5, 7, 8	4	2	11.2381
L2_W6_S2	2, 3, 4, 6, 8	6	6	12.2193
L2_W6_S3	2, 3, 4, 6, 8	6	6	14.2653
L2_W7_S2	2, 3, 5, 7, 8	4	7	10.7786
L2_W7_S3	2, 3, 5, 7, 8	2	7	11.6104
L2_W8_S2	2, 4, 5, 7, 8	8	8	10.0645
L2_W8_S3	2, 4, 5, 7, 8	2	8	12.0147
L2_W9_S2	2, 3, 4, 6, 8, 9	1	9	11.7238
L2_W9_S3	1, 3, 4, 6, 8, 9	1	9	12.6461
L2_W9_S4	2, 3, 4, 6, 8	3	9	14.1262

*L2: Line 2; W: Word; S: Sample

CONCLUSION

In this study, we have analyzed SIFT and M-SIFT based text recognition of Punjabi news ticker visible on

Punjabi news videos. It is evident from the study that modified SIFT has captured more number of features (keypoints) that the original version of SIFT. Moreover, it went ahead and has captured keypoints of words for which SIFT was not able to extract even a single feature. Study represents time consumption comparison. Study exhibitsthe recognition of various word samples tested for both SIFT and M-SIFT and it can be drawn that average recognition rate from both SIFT and M-SIFT is obtained at good percentage while M-SIFT overrides SIFT with a hike of almost 06%. In addition to this, average training time taken by M-SIFT is less as compared to that of SIFT. It's worth mentioning that average testing time per word is slightly more than that of SIFT but 2% increase of average testing time can be compensated for better (6%) recognition rate of modified scale invariant feature transform.

REFERENCES

Choudhary, P. and D. Renu, 2016b. Fusion based video text detection approach. Intl. J. Eng. Appl. Sci. Technol., 1: 94-98.

Choudhary, P. and D. Renu, 2016a. Text region extraction from Punjabi news videos using feature analysis. Res. Cell Intl. J. Eng. Sci., 18: 23-39.

Choudhary, P. and D. Renu, 2017. Taking frames out of (news) videos. Intl. J. Comput. Sci. Inf. Secur., 15: 391-395.

Dhir, R., 2016. Video text extraction and recognition: A survey. Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), March 23-25, 2016, IEEE, Chennai, India, ISBN:978-1-4673-9339-3, pp: 1366-1373.

Ghorpade, J., P. Raviraj, P. Ajinkya and R. Snehal, 2011. Extracting text from the video. Signal Image Process. Intl. J., 2: 103-112.

Gupta, T. and L. Garg, 2014. Face recognition using SIFT. Intl. J. Emerging Technol. Adv. Eng., 4: 358-363.

Kaur, A., B. Manju and Pooja, 2016. Text recognition past, present and future. Intl. J. Recent Innovation Trends Comput. Commun., 4: 506-516.

Leng, X. and Y. Jinhua, 2014. Research on improved SIFT algorithm. J. Chem. Pharm. Res., 6: 2589-2595.

Neeru, N. and L. Kaur, 2016. Modified SIFT descriptors for face recognition under different emotions. J. Eng., 2016: 1-12.

Ye, Q., W. Gao, W. Wang and W. Zeng, 2003. A robust text detection algorithm in images and video frames. Proceedings of the 2003 Joint Conference and 4th International Conference on Information, Communications and Signal Processing and Fourth Pacific Rim Conference on Multimedia Vol. 2, December 15-18, 2003, IEEE, Singapore, ISBN:0-7803-8185-8, pp: 802-806.