FLIP-INVARIANT VIDEO COPY DETECTION USING SPARSE-CODED FEATURES

Aysha Mol K S[1], Silpa Joseph[2]

[1]*M.Tech Student, Dept. of Computer Science and Engineering, VJCET, Kerala, India*
[2]*Asst. Professor, Dept. of Computer Science and Engineering, VJCET, Kerala, India*

Abstract

Now a days, a number of videos are available in video databases, social networking sites and other web servers. Large size of these video database make it difficult to trace the video content. To ensure the copy-right of the videos in video database, a video copy detection system is needed. A Video copy detection system stores the video features that characterize a video along with the video in the database. Existing copy detection systems store the video features as simple codewords. A simple and compact representation of video features makes the system more efficient. Moreover, the memory constraint problem can also be solved. This paper propose a sparse-coding technique that can represent the video features as sparse-codes. Proposed video copy detection system using sparse-codes works as follows: keyframes of the videos in the database are extracted using abrupt-transition detection algorithm. Salient regions of keyframes are detected by Harris-Laplacian detector and its local features are described by Flip-Invariant SIFT(F-SIFT) descriptor. F-SIFT enriches SIFT with flip invariance property by preserving its feature distinctiveness. F-SIFT is invariant to operations like flip, rotation, scale etc. A 128-Dimensional F-SIFT descriptor is extracted from each salient region. Extracted descriptors are converted to sparse-codes by the proposed sparse-coding technique. Each keyframe is represented by the sparse feature vector. Sparse vectors of all the keyframes of a video forms the sparse code of the video. Sparse-codes of the input video are compared with the sparse-codes stored in video database to identify the near duplicate videos. Experimental results demonstrate that proposed sparse-coding technique reduces the memory constraint problem. It also improves the detection accuracy.

Keywords: *Keyframes, F-SIFT descriptor, Sparse-Coding, Video Processing, Copy-Right Infringement, Video Copy Detection, Salient-Regions, Harris-Laplacian Detector.*

----------***----------

1. INTRODUCTION

Many of the videos stored in the video databases are near-duplicate copies of an original video. Near-duplicate videos[9] are approximately identical videos with similar appearance, but varies in terms of rotation, scale, photometric variation etc. On original videos operations like text insertion, combining scenes from 2 videos, performing flip, adding noise, rotation, scaling etc are performed to make the videos look different. The massive capacity of the video database makes the tracing of video content a difficult task. Also, controlling the copyright of huge number of videos uploaded everyday is a critical challenge. Hence a video copy detection system is needed to protect the copyright of the videos. A video copy detection system identifies illegal copies of a video by analyzing and comparing them to the original content.

Main challenge in the video copy detection system is that the video feature representation that is used along with this system requires a huge amount of memory for storage. Hence in a video copy detection system[11], a compact feature representation that can address this memory constraint problem is needed. Many of the existing image retrieval systems [7], [11] make use of sparse-coding to represent an image feature more efficiently. Also a video copy detection system need a feature descriptor that is invariant to operations like scale, rotation, light changes etc. Local feature descriptors[8] perform better than global feature descriptors. In addition the descriptor must be invariant to flip operation which is a common infringement technique. Flip is difficult to detect using widely used SIFT descriptor[12]. By using Flip-Invariant version of SIFT proposed in [12] a near duplicate copy of a video can be identified. By using the F-SIFT descriptor the computational complexity to identify flip operation can also be reduced.

This paper propose a sparse-coding technique that can represent a video feature using sparse-codes instead of simple codewords.. The sparse-codes reduces the number of bits required to store the video feature. Hence it reduces the memory constraint problem in copy detection system. Proposed sparse-coding technique also improves the accuracy of video copy detection system by computing the median of the feature descriptor histogram. The False Rejection Rate(FRR) of the video copy detection is also reduced by the proposed sparse –coding technique.

This paper is organized as follows: section 2 describes the related works. Section 3 explains the system overview. Each module is explained in detail in the sub-sections of this section. Section 4 contains the results of performance evaluation. Section 5 concludes the paper.

2. RELATED WORKS

In the existing video copy detection system, keyframes are extracted from the videos that are stored in video databases. They are the representative frames/shot of a video. salient regions of the keyframes are identified and the descriptors of these regions are extracted. The descriptors are then vector quantized to form a visual vocabulary(dictionary) by k-means clustering. Each keyframe in the video is represented as a Bag-of-Word(BoW) representation.

BoW model quantizes each of the extracted features from a keyframe, to one of the codewords in the dictionary using some distance or similarity measure. Finally, the keyframe is represented as the counts of the features quantized to each codeword. This form the descriptor histogram of the keyframe. The descriptor histogram of all the keyframes of a single video will form the codeword of the video . In the video database, the video name together with the codeword of that video is stored for copy detection.

Though, BOW model[7] provides good accuracy in the retrieval scenario, it is not practical for large video/image databases, as it is intensive both in memory and computations. Moreover, the resulting BOW vectors will also be sparse. This paper proposes a sparse-coding method which can be used to convert the codeword of the video into sparse-codes. Proposed Sparse-coding technique reduces the codeword size and hence reduces the memory requirements. It also improves the accuracy of the video copy detection system.

3. SYSTEM OVERVIEW

Proposed video copy detection system using sparse-coding framework works as follows: From the videos stored in video database, keyframes are extracted using abrupt-transition detection algorithm. salient-regions of the keyframes are identified using Harris-Laplacian detector[12] and the descriptors describing the salient region features are extracted using F-SIFT. A number of other local feature descriptors[1], [2], [3], [4], [5] [6] [10] are available in literature for feature extraction. Among them, the Scale-Invariant Feature Transform(SIFT)[1] descriptor is the most appealing descriptor for practical use and also the most widely used descriptor since it is invariant to transformations like rotation, scaling, light changes etc.But SIFT is not invariant to flip operation[12], which is a commonly used infringement technique.

Flip is a common operation used in creating near-duplicate videos. Flip produces the mirror of an image. Flip operation are of two types: horizontal and vertical (Fig. 1). Horizontal flip performs flipping around vertical axis and vertical flip performs flipping around horizontal axis. The main advantage of this operation is that it will not cause a change in the video content, only the direction of information flow will get changed. Hence it is easy to create the copy of a video without much change in content. Hence to identify flip, the feature-invariant descriptor used in a video copy detection system must be invariant to flip transformation.

F-SIFT[12] is the flip-invariant version of SIFT. For transformation involving no flip, F-SIFT shows similar performance as SIFT. When flip happens, F-SIFT[12] performs better than SIFT. F-SIFT descriptors from the video copy detection system are vector quantized to form a visual vocabulary by k-means clustering. For each keyframe, a descriptor histogram is formed that contain the count of features quantized to each code-word. This descriptor histogram will act as the BoW for each keyframe. Median value of the Bag-of-Words of all the keyframes in the video database is computed. BoW positions having value greater than this median value will be set to one and others are set to zero. Thus the Bag-of-Word of each keyframe is converted to sparse code. The Bag-of-Words of the keyframes of a particular video forms the code-word of that video. In the video database the video name together with the sparse-code of the video and the median value are stored.

Fig. 1 Flipped images of Lena (a) Original Image (b) Horizontal Flip (c) Vertical Flip

When an input video arrives, first keyframes are extracted using abrupt-transition detection algorithm. From the keyframes, F-SIFT keypoints are detected and feature descriptors are extracted. A descriptor histogram is generated for each keyframe of the video. It forms the BoW for the keyframe. The BoW is converted to sparse codes by using sparse coding technique described above. i.e. BoW positions having value greater than the median value will be set to one and others are set to zero. The sparse-code of each keyframe are combined to form the sparse-code for the input video. The sparse-code of the input video is then compared with the sparse-code of database videos. The one with maximum similarity is considered as a match. System Overview is shown in figure 2.Various steps in the proposed video copy detection system is explained below:

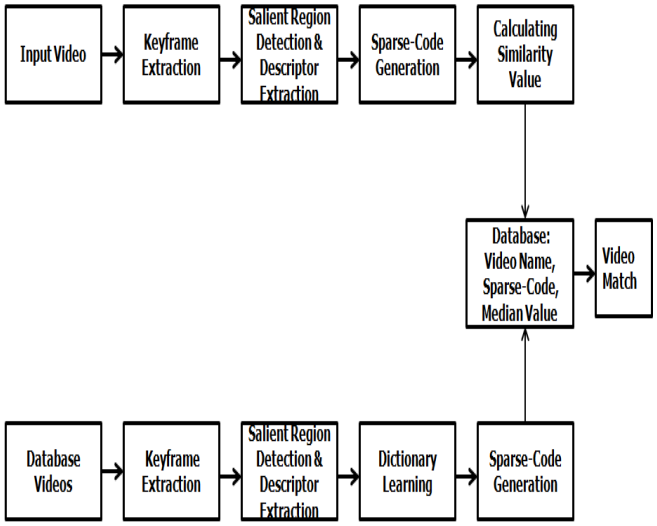

Fig 2 System Overview

3.1 Keyframe Extraction

A Keyframe is a representative frame per shot in a video. Keyframe extraction is a fundamental technique for video processing. Keyframe extraction make use of abrupt transition detection algorithm. During abrupt transition, there is normally a big difference between the two transition frames. Abrupt-transition detection algorithm detects this big difference. Abrupt-transition is detected by computing intensity histogram difference. The intensity for an RGB frame can be calculated as,

$$I = 0.299R + 0.587G + 0.114B \qquad (1)$$

For each frame in a video above equation is computed where R, G, B are R=red G=green and B=blue channels of the pixels. For the intensity histogram difference eqn. 2 is used.

$$SD_i = \sum_{j=1}^{G} | H_i(j) - H_{i+1}(j) \qquad (2)$$

where, $H_i(j)$ = histogram value for i^{th} frame at level j, G=total number of bins in the histogram.

In a continuous video frame sequence, the histogram difference is small. But for abrupt transition detection, the intensity histogram difference spikes. Therefore, the difference of intensity histogram with a proper threshold is effective in detecting abrupt transitions. The threshold value to determine whether the intensity histogram difference indicates an abrupt transition can be set to,

$$T_b = \mu + \alpha\sigma \qquad (3)$$

Where μ=mean, σ=standard deviation and α=3- 6. All the frames in the video having intensity histogram difference greater than T_b is considered as keyframes

3.2 Salient Region Detection and Descriptor Extraction

There are a number of local feature detectors available in literature. All of them are flip invariant[12]. These detectors differ in their choice of saliency function. Harris-Laplacian detector[12] is based on the second moment matrix which is defined in Eqn. 4 for a point X.

$$\mu(X,\sigma_1,\sigma_D) = \sigma_D^2 g(\sigma_1) * \begin{bmatrix} L_x^2(X,\sigma_D) & L_x L_y(X,\sigma_D) \\ L_x L_y(X,\sigma_D) & L_y^2(X,\sigma_D) \end{bmatrix} \qquad (4)$$

where σ_1 is the integration scale, σ_D is the differential scale and L_g is to compute the derivative of X in g (*x* or *y*) direction. Local derivatives are computed with Gaussian kernels of scale σ_D. The derivatives are averaged in the neighbourhood of X by smoothing with integration scale σ_1. Based on Eqn. (4), the Harris function at pixel *X* is

$$Harris(X) = | \mu(X,\sigma_1,\sigma_D) | - \alpha * trace^2(\mu(X,\sigma_1,\sigma_D)) \qquad (5)$$

where α is a constant. Scale invariance is further achieved by scale-space processing computed by Laplacian-of-Gaussian matrix.

$$LoG(X,\sigma_1) = \sigma_1^2 | L_{xx}(X,\sigma_1) + L_{yy}(X,\sigma_1) | \qquad (6)$$

where L_{gg} denotes the second order derivative in direction *g*. The local maxima value of X, with respect to integration scale σ_1 , is determined based on the structure around *P*. Harris-Laplacian detector considers a pixel *X* as salient, if it attains local maxima in *Harris(X)* and $LoG(X,\sigma_1)$ simultaneously.

Once the salient region is detected, to make the descriptor flip-invariant, Curl[12] is computed at each salient region (keypoint) . Curl defines the direction of rotation of a vector field. Curl is positive when direction of rotation is anti-clockwise and curl is negative, when direction of rotation is clockwise. Curl of a keypoint is computed using the equation,

$$C = \sum_{(x,y)\in I} \sqrt{\frac{\partial I(x,y)^2}{\partial x} + \frac{\partial I(x,y)^2}{\partial y}} \qquad (7)$$

where

$$\frac{\partial I(x,y)}{\partial x} = I(x-1,y) - I(x+1,y)$$

$$\frac{\partial I(x,y)}{\partial y} = I(x,y-1) - I(x,y+1)$$

Θ is the angle from direction of the gradient vector to the tangent of the circle passing through (x, y). G is the Gaussian kernel of sigma=40. If we enforce that every local region should have a positive curl, for regions with negative curl flipping the regions along the horizontal (or vertical) axis as well as complementing their dominant orientations are explicitly performed to geometrically normalize the regions.128-D SIFT descriptors are then extracted from the normalized regions. This makes the descriptor flip-invariant and is called Flip-Invariant SIFT (F-SIFT) [12] descriptor.

For all the videos in the database, salient regions are detected and F-SIFT descriptors are extracted.

3.3 Dictionary Learning

During dictionary learning the descriptors extracted from all the keyframes are vector quantized by means of k-means clustering to form 16 clusters. k-means clustering is a method of vector quantization, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. Here the descriptor set from the entire video database are grouped into 16 clusters and their means are returned. This 16-cluster means will act as the dictionary (visual vocabulary). Dictionary will have size 16*128.

3.4 Sparse-Code Generation

During sparse-code generation, the descriptors from each keyframe are compared with the dictionary and the descriptor histogram of each keyframe is formed. It acts as the Bag-of-Word for a keyframe. Bow represents the counts of the features quantized to each codeword. The median of all the descriptor histograms in the system is computed. Sparse-Code of each keyframe is generated as follows: For each keyframe, the sparse-code value at position (i, j) is set to one if the descriptor histogram value at (i, j) is greater than median else the value at that position is set to zero. Equation for sparse coding is as follows:

$$SC(i,j)=\begin{cases}1 & if \quad DH(i,j)>median\\ 0 & otherwise\end{cases} \tag{8}$$

Sparse-Code of all the keyframes of a particular video forms the sparse code of that video. For a video with n keyframes, the sparse-code size is n*16, where 16 is the number of clusters. In video database video name together with its sparse code and median value are stored.

3.5 Video Matching

When an input video with some transformations applied on it arrives, its keyframes are extracted, salient regions are detected and 128-D F-SIFT descriptors are extracted. A descriptor histogram is formed for each keyframe. It act as the Bag-of-Word for each keyframe. Bag-of-Word of each keyframe is converted to sparse code using eqn. 8. Sparse-Code of all the keyframes of a particular video forms the sparse code of that video. If the sparse–code of the input video is the same as the sparse-code of the video in database, the videos are considered as matched. i.e. Input video is a near-duplicate copy of the database video. Similarity value is computed using eqn. 9.

$$Similarity value = mean(SC(inputvideo) == SC(Videodatabase)) \tag{9}$$

The video having the highest similarity value with the input video is considered as match.

4. EXPERIMENTAL EVALUATION

The proposed flip-invariant video-copy detection system using sparse-coded features is evaluated and compared with flip-invariant video-copy detection system using Bag-of-Words(codewords) in the same scenario. The objective is to evaluate the performance and the memory usage of the system. All the algorithms were implemented using MATLAB.

4.1 Comparison of Bits required to store Video Features

For a video with n keyframes and 16 cluster centers, sparse coding requires n*16 bits to store the video features. In general codeword representation, codeword size is n*16*p, where p=number of bits required to represent the maximum number in the codeword as binary. By using sparse-codes the memory needed to store video feature can be reduced.

4.2 Comparison of Accuracy

In order to check the accuracy of the system during various transformations, a number of transformations like text insertion, adding noise(Gaussian ,Salt and Pepper), combining 2 videos, converting to grayscale etc are performed in addition to flip. It is shown that during video combining and in Gaussian noise insertion, sparse codes works better than general codeword representation.

4.3 Computation of False Rejection Rate (FRR)

FRR is the ratio of unrecognized flip appearances to the total number of flip appearance.FRR value obtained is .29 for sparse-codes and .37 for Bag-of-Words representation. Reducing the FRR improves system efficiency. Hence Sparse-Codes works more efficiently than general codeword representation.

Table 1 Performance Comparison

Transformations	Accuracy of Code-words (%)	Accuracy of Sparse-Codes(%)
Flip	80	80
Flip+ Text	75	75
Flip+Gray	80	80
Flip+Noise(Gaussian)	60	70
Flip+Noise(Salt & Pepper)	40	40
Combine Videos	40	80

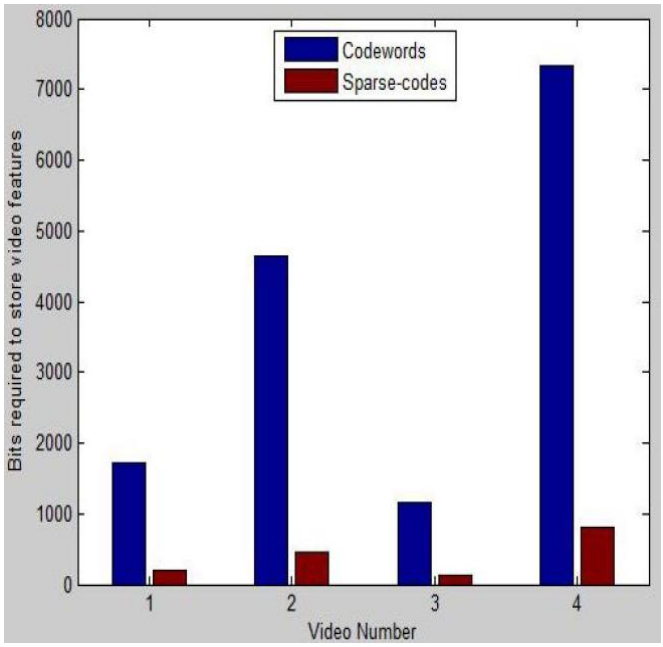

Fig. 3 Comparison of bits required to store video feature

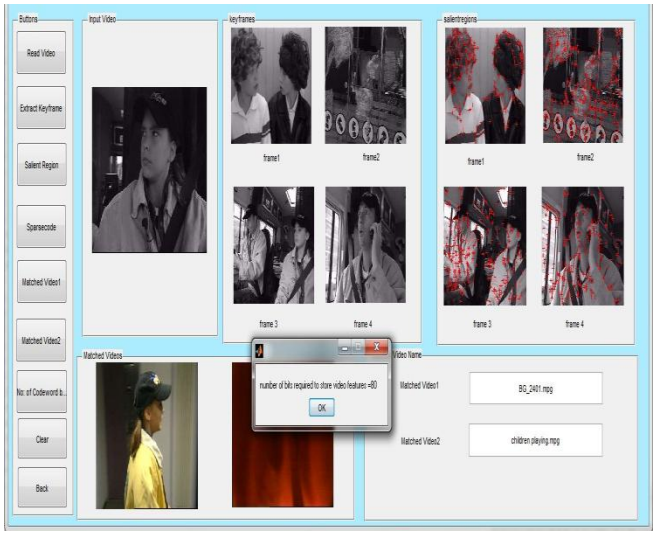

Fig 4 Performance result of flip-invariant video copy detection using sparse-codes

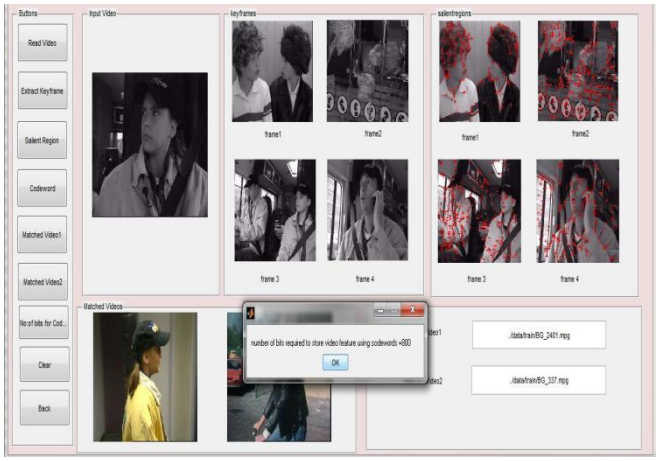

Fig 5 Performance result of flip-invariant video copy detection using codewords

5. CONCLUSIONS

This paper propose a video copy detection framework that identifies near-duplicate copies of videos by analysing them and comparing them to the original content. Near-duplicate videos are identical or approximately identical videos with similar appearance, but varies from the original one because of the transformations applied on it. Various transformations applied to create near-duplicate videos are flip, inserting text, rotation, scaling, grayscale conversion, combining 2 videos etc. Proposed method first extracts keyframes of a video. The salient regions of keyframes are detected using Harris-Laplacian detector and F-SIFT descriptors are extracted. The descriptors are then vector quantized to form a visual vocabulary. Each keyframe is represented as a Bag-Of-Visual Word representation. Bag-Of-Word representation is converted to sparse-codes by the proposed sparse-coding technique. Sparse-Codes of the keyframes in a video forms the sparse-code of that video. Video matching is performed to identify the original versions of the input video. F-SIFT descriptor identifies almost every copy-right infringement techniques. Proposed Sparse-coding technique reduces the memory requirement for the storage of video features. It also improves the matching accuracy. FRR of the proposed video copy detection system is also reduced.

ACKNOWLEDGMENTS

Authors would like to thank the Department Head and Group Tutor for their constructive comments and informative suggestions that have helped them to improve this paper.

REFERENCES

[1] D. Lowe, “Distinctive image features from scale-invariant keypoints,” in *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] S. Lazebnik, C. Schmid, and J. Ponce, “A sparse texture representation using local affine regions,” in *IEEE Trans. Pattern Anal. Mach. Intell.* ,vol. 27, no. 8, pp. 1265–1278, Aug. 2005

[3] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” in *IEEE Trans.*

Pattern Anal. Mach. Intell. , vol. 27, no. 10,pp. 1615–1630, Oct. 2005.

[4] R. Ma, J. Chen, and Z. Su, "MI-SIFT: Mirror and inversion invariant generalization for SIFT descriptor," in *Proc. Int. Conf. Image Video Retr.*,2010, pp. 228–236.

[5] X. Guo and X. Cao, "FIND: A neat flip invariant descriptor," *in Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 515–518.

[6] W.-L. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning," in *IEEE. Trans. Multimedia*, vol. 9, no. 5, pp. 1037–1048, Aug.2007.

[7] M. K Reddy, J Talur, Venkatesh Babu R, "sparse coding based VLAD for efficient image retrieval," in *IEEE International Conference on Electronics, Computing and Communication Technologies*, 6-7 Jan 2014.

[8] Julien Law-To, Li Chen, Alexis Joly, Ivan Laptev, Olivier Buisson Brunet, Boujemaa, Stentiford, "Video Copy Detection: a Comparative Study," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007, Pages 371-378 .

[9] W.-L. Zhao, X. Wu, and C.-W. Ngo, "On the annotation of web videos by efficient near-duplicate search," in *IEEE Trans. Multimedia*, vol. 12,no. 5, pp. 448–461, Aug. 2010.

[10] X. Guo, X. Cao, J. Zhang, and X. Li, "MIFT: A Mirror Reflection Invariant Feature Descriptor", *Springer, ACCV 2009, Part II, LNCS 5995*, pp. 536–545, 2010.

[11] Ge,Ke and Sun, "Sparse-Coded features for image retrieval," in *Proc.microsoft research publication*, 2013.

[12] Zhao and Ngo "Flip-invariant sift for copy and object detection", in *IEEE trans.image processing*, March 2013.

BIOGRAPHIES

AYSHA MOL K S is currently doing **M.Tech** in computer science and Engineering at Viswa Jyothi College of Engineering and Technology under M G University, Kottayam. She completed **B.Tech** from Rajiv Gandhi Institute of Technology, Kottayam. She has published a paper "A Survey on State-of-Art Flip-Invariant Descriptors" in International Journal for Research in Applied Science and Engineering Technology. Her areas of interest are Image and Video Processing.

SILPA JOSEPH is currently working as **Asst. Professor** in computer science and engineering at Viswa Jyothi College of Engineering and Technology from 2007 onwards. She has about 7 years of teaching experience. She has completed **M.Tech** from Karunya university in 2007 and **B.Tech** from Madras university in 2004. She has published papers in various national and international conferences/journals. Her areas of interest are image processing and network security.