Chapter 1: Introduction

1.1 Overview

The use of inscription in the ancient world has importance in helping preserve information on those ancient cultures for which we can find such artifacts. The use of inscription occurred in a variety of different ancient cultures. These ancient inscriptions provide insights into the history of the cultures that used them and they also act as a means of transmission of poetic and literary writings from the ancient to the modern world. Inscription refers to writing done on durable material, versus those on more commonly used material of later times such as paper and papyrus. When art is so engraved into durable material then that is known as epigraphy. Earliest writing most likely was done everywhere on such durable material, including ivory, bone, stones and clay. In modern times, society still uses inscription, typically when the intent is to provide for a permanent record.

Inscriptions at the historical monuments are a common site that gives a rare sneak-peek into the history attached with it. The text or graphics in inscriptions may symbolize a sequence of events from the past or may represent a story or culture of that era. Lot of work has been done to conserve the monuments which are precious from heritage point of view. Reading of inscriptions from these monuments is very important so that the history can be digitally preserved and stored, translated in other languages and the culture and lifestyle of the ancient era can be understood.

Several methods exist for detection, localization and extraction of text but the problem intensifies when the background foreground is similar or textured or there exists very minimal difference between them. Such is the case of camera held images of inscriptions where the difference between background and foreground creates problem in reading thus leading to the problem of text extraction. The images of inscriptions have problems like illumination, wrapping, minimal difference of background foreground, multi-lingual text, complex backgrounds and perspective projection distortion. Due to these challenges extraction of text becomes very difficult.

When such images are passed to commercial OCR's (Optical Character Recognition), the recognition rate is 0% in maximum cases. Recognition of text is possible only when the images have distinctive backgrounds and foregrounds. Thus, there arises need to enhance these images so that the background and foreground appear distinctive and clear.

The proposed work deals with this problem of enhancing the image to make recognition possible and achievable over such images. The work deals with enhancing the multi-lingual inscriptions using NGFICA technique. Multi-lingual here refers to images of inscriptions collected from different sites which have texts in multiple languages. Two such examples are shown in fig1.

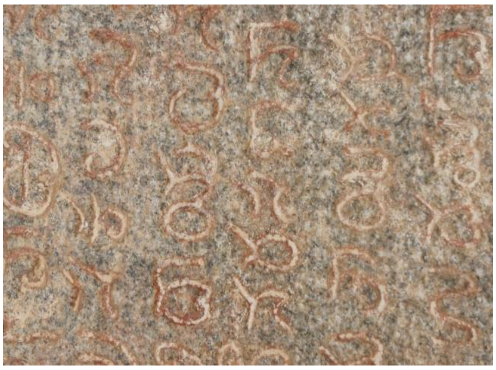


Figure 1 a) inscription from Hampi- heritage site b) from an ancient temple in South

Sculptures are another important pillar of unveiling past. The sculptures found at historical sites are usually of god/goddesses which the people of that era worshipped. For e.g., Hampi has sculptures of Lord Ram, Lord Hanuman, Lord Krishna etc. The images of these sculptures also have same problems as possessed by the images of inscriptions on the walls of these monuments. For training a classifier for such images of the sculptures, it becomes mandatory to preprocess the images such that they don’t have any of the problems as discussed above.

Currently, there exist no methods or techniques to classify the images Indian gods/goddesses. While thinking of such a 2-class classifier, we need images with similarity. Figure 2 a,b) shows examples of sculptures from Hampi. Fig 2a) is image of god Krishna and 2b) is image of god Ram.

In our work, we have proposed a 2-class classifier that takes a 2-class data set (for e.g, either Ram and hanuman or hanuman and *shivling* or hampi images of god Ram and Hanuman). Then the HOG features are calculated over these images after resizing them. After feature extraction, the classifier is trained and it classifies the query images as either belonging to class A or class B. We have used svm_light for classification. This gives an accuracy of 60% to 100%, depending on the dataset.

Fig 2.a) Image of Lord Krishna at Hampi b) Image of Lord Ram at Hampi

1.2 Related work

Text Extraction from document images has been of interest for the research community over a decade, but there has been very little work done in digitizing inscription images of historical monuments.

1.2.1 Text Region Extraction

Inscriptions are nothing but text written/engraved on walls. Work done so far in this field mostly includes document images and extraction of text from them. The extraction is based on the following factors as discussed below [24].

1.2.1.1Extraction using color continuity

Color is the most common feature to differentiate between objects in an image. The color helps to visually segment the regions or objects at human level. And at machine level, this feature can be used to classify and separately label the objects based on their color. This method as described in [24] uses color reduction at first step to reduce number of colors and processing time. With a simple bit-dropping method, lower six bits of each R, G, B components of each pixel are removed. Result image can be expressed up to 64 colors. Color clustering is performed in order to reduce the number of colors and to merge similar colors into a same color. Euclidean distance is used for clustering. Colors with a very small number of histograms are removed. This leads to further reduction of colors. The pixels with removed colors are changed into the similar color with a large number of color histogram. After this, color that exists in a corner in RGB space with the largest histogram, or that exists near a corner with the largest histogram is selected. The clustering color is selected and clustered. When the clustering stops, another clustering begins in the RGB space where the distance is the farthest from the previous clustered color. This process is performed repetitively until the color totals to only 2.

1.2.1.2Extraction using intensity variation

As discussed in [24], most of text regions in the natural scene images have edges densely populated, thus this approach finds the regions based on the edge density. Also, the regions are usually surrounded by the rectangular boxes, and we remove the box lines first. These lines can also be used for estimating skew and perspective corrections of the text regions. A median filter is applied to the gray-level images, and edges are found by canny edge detection method. The long line components include horizontal or vertical lines, quadrilaterals, and broken lines in the edge image. 8-directional edge following is performed and histograms of each direction are obtained. Since the 8-directional histogram bins can be different according to the starting point of the edge following, 4-directional bins by adding the opposite directional bins together are considered. Directional variation and maximum and minimum X and Y coordinate of the component are calculated. Horizontal or vertical long lines have dominant histograms only in one direction with small variations of the direction. Quadrilaterals have dominant histograms usually in two directions. And, the broken lines also have the dominant histograms in two or three directions. Then the long line components will become the connected components with the dominant values of the histograms in either one or two directions both in 8 and 4-directional bins. After the lines are removed, the text regions will appear in the dense regions of edges. Structuring elements with sizes 2x5 and 3x2 are used for the morphological dilation and erosion, respectively.

1.2.1.3 Extraction based on color variance

There are many cases that text regions are not clearly distinguished by background regions due to light or illumination variations. If color variance is used as a feature, we can expect better separation between text regions and backgrounds. Using color variance, one can find most of the regions even with small variations in gray-level values. The method based on variance by M. Babu et al [8] makes use of the variance in the text and non-text regions. The variance is high at the text edges and vice versa. Variance method to extract the text as in [8] did not prove successful as the edges belonging to the text were not sharp and the distinction between text and non-text regions that was supposed to be there in order to get the desired results, was not present.

1.2.2 Edges as text extractors

High contrast edges between text and background is obtained using the red color component in the approach by Agnihotri et al. [5]. In [6], the "uniform color" blocks within the high contrast video frames are selected to correctly extract text regions. Kim et al. [7] used 64 clustered color channels for text detection where cluster colors are based on Euclidean distance in the RGB space. The text in inscription images does not consist of a uniform color and there is low contrast between text and background thus making the use of [6] unsuitable. Simple edge-based approaches are also considered useful to identify regions with high edge density and strength. This method performs well if there is no complex background but the inscription images have complex background thus these methods cannot be used directly.

1.2.3 Contrast enhancement and ICA

Garain et al [3] describe how to enhance image using FastICA algorithm which results in three independent components or layers which correspond to the contribution of text in them. The method is an enhancement method, which however is unable to enhance inscription dataset being dealt with in our method as FastICA is found inefficient in case of weak or highly spatially correlated sources[10]. More recently, its convergence has been shown to slow down or even fail in the presence of saddle points, particularly for short block sizes [11]. We have used natural gradient based independent component analysis learning algorithm with flexible nonlinearity as described in [12] which gives better results than other algorithms on the inscription images in our dataset. The above said methods were based on binarization, text extraction using variance or edge detection based methods. These methods depend upon pixel's threshold value based on difference between foreground and background part.

But in case of unclear and complex archaeological inscription images, there is no sharp distinction between foreground and background. Hence we propose an efficient multi-lingual inscription image enhancement method.

In our method we used NGFICA for minimizing the dependency between foreground and background of such inscription images and further the distinction enhancement technique based on global average is applied for enhancing the difference between foreground and background part of images.

1.2.4 Sculpture classification

Regarding the sculptures' classification, there is not much work done. According to [18], a recognition system called Thai Buddhist Sculpture Recognition System (TBuSRS) is built which is trained and tested on around 50 types of different Buddhist sculptures with around 500 images in total. The features used for training are 14 in number namely, edge detection feature, eye, nose detection feature, bottom neck feature etc. In our case this method [18] is not suited because it considered only the face and recognizing it as a positive or a negative example of Buddhist sculpture. But, ours is a classification problem where we have two different kinds of datasets with variation in the poses in each dataset. Thus, the features as used in [18] were not feasible to be used in our case.

1.2.5 Features for object recognition

Features have always played an essential role in object detection and classification problems. Haar wavelet [19] is an effective feature for general object detection, especially when combined with the boosting-based learning framework. The SIFT feature [20] has become one of the most popular features for object recognition and image retrieval/matching due to scale/rotation invariant property. HOG was used as feature descriptor Dalal and Triggs [15] for person detection. Talking of other features, the original SIFT descriptor [20] was computed from the image intensities around interesting locations in the image domain which can be referred to as interest points, alternatively key points. These interest points are obtained from scale-space extrema of differences-of-Gaussians (DoG) within a difference-of-Gaussians pyramid. HOG descriptor [15] is popular for object recognition for many reasons. In [21], Feng et al used two-stage approach to detect people and vehicles in static images using extended histogram of oriented gradient (HOG) and SVM for classification. In the first stage, people and vehicle locations are hypothesized. This step takes the prior knowledge about what people and vehicle may look like in the depth map of the whole scene. The second stage is the verification of hypothesis using extended HoG and SVM.

1.2.6 HOG in our method

We have used HOG features in our proposed method to extract the features of sculptures. HOG uses vector-space model and the Euclidean distance between two HoG vectors is used to calculate the perceptual similarity. This means that many off-the-shelf learning and database algorithms can work directly on HoG representations. Another reason is that it uses intensity gradients rather than intensity directly which means that the responses of edges are localized.

HoG is similar to sift as it is also scale and rotation invariant but better and different because HoG computes descriptors without features as it computes them on a dense and isotropic grid.

For our purpose and the dataset, calculating SIFT features was not efficient as the aim of our problem was to classify the images as either class A or class B. For this HoG proved successful as the images of god/goddesses were human poses and in upright positions. HoG was better because the vector-space model of HoG made it easy to approximate the perceptual similarity between two HoG vectors using Euclidean (or cosine) distance. HoG focuses on local shape by capturing edge or gradient structure of that shape. In HOG the responses of edges are localized as HoG uses intensity gradients rather than using intensity directly. The response of the edges is sensitive to local but not global contrast due to normalization scheme. HoG is relatively invariant to local geometric and photometric transformations. Within cell rotations and translations do not affect the HOG values. Another important reason was time. The time to calculate HoG features over an image is less than 2 seconds. This made the use of HoG more suitable.

1.3 Motivation

The work proposed here is a step towards preservation of our history. Digitization helps in a better and efficient way of working. Taking the e.g. of inscriptions on the walls of monuments, digitalization can help in shaping the work of preservation in a better way. If the digital form of data is with us, then we can preprocess it, apply number of techniques for preserving them, extracting information out of them etc.

The main reason for tourism at ancient monuments is exploring past, learning the methods and techniques used by ancient people. After digitization, authorities looking after these monuments can build up their own databases where they can store the images, provide translation for the inscriptions in the user-desired language, provide an abstract about each sculpture and many such useful things that can help the past reach maximum people in the form convenient to them. For e.g. a blind person might be able to read inscriptions using Braille if there are some hand-held pads in front of these inscriptions, another tourist might like to know the history of a sculpture using ear phones or head phones.

All this can also be available online. That is, a user might just select his choice of sculpture or inscription from the dataset and he can translate it or read about it.

These are few practical applications and a major motivation behind this work.

1.4 Scope of work

As mentioned above, the monuments which bear such precious inscriptions on them are very old. So, it is very likely that the walls and stones of these historical monuments are decayed or broken or not in a condition that they can be read easily. Thus, the task of enhancing the text on them becomes more challenging due to bad illumination and decayed condition.

The enhancement of images in our work is done by first dividing the image into its independent components using ICA based NGFICA algorithm. Three independent components are referred as text layer, non-text layer and mixed layer. These three components have some content of text in them in an increasing order that is, the text layer has the highest text content and non-text layer has lowest text content. Then over these three components a consolidated average which is average of all the individual averages (of three components) is calculated. This consolidated average is referred as global average here. It is experimentally seen that the independent component whose individual average is farthest from the consolidated or global average, have a clearer distinction between background and foreground than the other components. This component is selected for further processing. Then various morphological and noise removal techniques are used to further enhance the image. The enhanced images are then passed to OCR for recognition. The results are discussed later.

Also, our work demonstrates the results of NGFICA algorithm and Fast-ICA algorithm, which are two variants of ICA algorithm. Fast ICA based enhancement proved efficient in the case of inscription images with a reasonable distinction between the background and foreground regions. But, in our case, most of the images do not have this distinctive property, making it difficult to enhance images (in order to perform text extraction and later text recognition). FICA seemed failing in the case of enhancement of inscription images of our dataset thus, we used NGFICA algorithm.

For the classification of sculptures, we have considered the images from INTERNET and few images from Hampi temples too. The images were preprocessed by cropping and resizing them and converting them to binary images. Then using HOG, the features were extracted. We have used the freely available SVMLight software package in conjunction with their HOG descriptors to train the classifier and classify the sculptures as belonging to either of the two classes (on which the classifier has been trained). The classification results in the testing images being classified as either of the classes used for training. The advantages of classification of sculptures are listed below:

- After classification the information retrieval becomes easy.
- One might use this to know more about the god/goddesses; about the era they belonged to, the people who worshipped them.
- It helps in better categorization based on the temples they are worshipped at and other relevant details about them

These advantages are few areas of applications. There are many more such advantages.

Chapter 2: NGFICA based inscription enhancement

Performing enhancement of the noisy inscription images is a difficult task because of the challenges stated earlier in chapter 1. Contrast plays a major role in deciding the clarity of an image. Contrast is the difference in luminance and/or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. Because the human visual system is more sensitive to contrast than absolute luminance, we can perceive the world similarly regardless of the huge changes in illumination over the day or from place to place. The maximum contrast of an image is the contrast ratio or dynamic range. Contrast is also the difference between the color or shading of the printed material on a document and the background on which it is printed, for example in optical character recognition. The images of inscriptions fail on the OCR software because of the lack of contrast in those images. By our method we therefore try to enhance such images.

The proposed method works on the images captured at Hampi (a heritage site in South), images from India Gate (New Delhi), images of inscriptions in multiple languages from INTERNET (namely, European inscriptions, Islamic inscriptions, Telgu inscriptions from temples etc). These images lack contrast and distinction between foreground and background. Our work deals with the enhancement of images so that they can be recognized by their respective language OCR's. These images of inscriptions bear many problems due to bad illumination, decaying or breaking of stones or walls etc as stated above. When such images are tried over their respective OCR's, then the recognition rate is nearly 0%. This is obvious because of the bad quality of the images and amount of information it contains. A human readable form of such inscriptions is not clear enough to be processed or OCR-ed. There is not much work done in this field and the work that is somehow related, does not give the desired results. The proposed work deals with using NGFICA that is; Natural Gradient based Fast ICA (Independent Component Analysis). We have used this method to separate out the independent components and use the best out of them which contains minimal noise and whose average is farthest from the average of averages of all ICs. This is explained in detail in later section. The ICA algorithm mainly belongs to signal processing domain but, we have used it in our work of image processing. We have used NGFICA algorithm as the other variant of ICA called FICA failed in our case.

Before we discuss our methodology a brief about contrast enhancement, text detection, segmentation is given below with few other details of the techniques and algorithms that are related.

2.1 CONTRAST ENHANCEMENT

Contrast enhancements improve the perceptibility of objects in the scene by enhancing the brightness difference between objects and their backgrounds. Contrast enhancements are typically performed as a contrast stretch followed by a tonal enhancement, although these could both be performed in one step. A contrast stretch improves the brightness differences uniformly across the dynamic range of the image, whereas tonal enhancements improve the brightness differences in the shadow (dark), midtone (grays), or highlight (bright) regions at the expense of the brightness differences in the other regions.

2.1.1 Gray-level histogram

Most contrast enhancement methods make use of the gray-level histogram, created by counting the number of times each gray-level value occurs in the image, then dividing by the total number of pixels in the image to create a distribution of the percentage of each gray level in the image. The gray-level histogram describes the statistical distribution of the gray levels in the image but contains no spatial information about the image. Setting the exposure of the camera to span the full dynamic range would optimize the contrast, but this runs the risk of saturating the detector with any radiance value that would exceed 255 counts, thus clipping these values into 255 counts and losing any scene information above this radiance level. Exposures are, therefore, usually set to collect lower-contrast images that do not span the dynamic range because the images can be processed later to enhance the contrast while maintaining control over the amount of clipping that occurs.

2.1.2 Contrast stretch

A high-contrast image spans the full range of gray-level values; therefore, a low-contrast image can be transformed into a high-contrast image by remapping or stretching the gray-level values such that the histogram spans the full range. The contrast stretch is often referred to as the dynamic range adjustment (DRA). The simplest contrast stretch is a linear transform that maps the lowest gray level GLmin in the image to zero and the highest value GLmax in the image to 255 (for an eight-bit image), with all other gray levels remapped linearly between zero and 255, to produce a high-contrast image that spans the full range of gray levels.

2.2 Segmentation

Segmentation is the process of separating something into parts or dividing it. Image segmentation may use statistical classification, thresholding, edge detection, region detection, or any combination of these techniques. The output of the segmentation step is usually a set of classified elements. Most segmentation techniques are either region-based or edge-based. Region-based techniques rely on common patterns in intensity values within a cluster of neighboring pixels. The cluster is referred to as the region, and the goal of the segmentation algorithm is to group regions according to their anatomical or functional roles. Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of :

- Intensity values from original images, or computed values based on an image operator.
- Textures or patterns that are unique to each type of region
- Spectral profiles that provide multidimensional image data.

Textures or patterns that is unique to each type of region. Edge-based techniques rely on discontinuities in image values between distinct regions, and the goal of the segmentation algorithm is to accurately demarcate the boundary separating these regions. If something is to be divided or clustered then there is a criteria of division attached to it. That is, we need to allot or decide a criterion as to how this division or segmentation is to be made. Setting the criteria depends on the problem or object that we are working on. For e.g. in the images of inscriptions we are segmenting the words (explained later). So, the criterion chosen is with respect to that. Text segmentation is the process of dividing written text into meaningful units, such as words, sentences, or topics. Word segmentation is the method of dividing a string of written language into its component words. In English and many other languages using some form of the Latin alphabet, the space is a good approximation of a word delimiter. Word splitting is the process of parsing concatenated text (i.e. text that contains no spaces or other word separators) to infer where word breaks exist. Word splitting may also refer to the process of hyphenation.

2.3 Text segmentation

It is well known text extraction, including text detection, localization, segmentation and recognition is very important. For scanning images for recognition (in OCR) the images needs to be clean and distinction between background and foreground is compulsory.

The work discussed here is about enhancing the not so good inscription images (in terms of distinction of non-text and text part). Because of the illumination problems and minimal intensity variation between foreground background in the images captured, we work on enhancing the images such that the text in them can be segmented and they can be run on commercially available OCR's in order to preserve them forever.

2.4 ICA

“Independent component analysis (ICA) is a method for finding underlying factors or components from multivariate (multi-dimensional) statistical data. What distinguishes ICA from other methods is that it looks for components that are both *statistically independent*, and *nonGaussian*.”

-A.Hyvarinen, A.Karhunen, E.Oja

Nowadays, performing statistical analysis is only a few clicks away. However, before anyone carries out the desired analysis, some assumptions must be met. Of all the assumptions required, one of the most frequently encountered is about the normality of the distribution (Gaussianity). However, there are many situations in which Gaussianity does not hold. Human speech (amplitude by time), electrical signals from different brain areas and natural images are all examples not normally distributed. The well-known "cocktail party effect" illustrates this concept well. Let us imagine two people standing in a room and speaking simultaneously. If two microphones are placed in two different places in the room, they will each record a particular linear combination of the two voices. Using only the recordings, would it then be possible to identify the voice of each speaker.

If Gaussianity was assumed, one could perform a Principal Component Analysis (PCA) or a Factorial Analysis (FA). The resulting components would be two new orderly voice combinations. Therefore, such a technique fails to isolate each speaker's voice. On the other hand, if non-Gaussianity is assumed, then Independent Component Analysis (ICA) could be applied to the same problem and the result would be quite different. ICA is able to distinguish the voice of each speaker from the linear combination of their voices. The pictorial representation of the same is shown in fig 3.

This reasoning can be applied to many biological recording involving multiple source signals (e.g. EEG). However, the readers must bear in mind that there are two main differences in the interpretation of extracted components using ICA instead of PCA. First, in ICA, there is no order of magnitude associated with each component. In other words, there is no better or worst components (unless the user decides to order them following his own criteria). Second, the extracted components are invariant to the sign of the sources. For example, in image processing, a white letter on a black background is the same as a black letter on a white background.

In other words, ICA (Independent Component Analysis) is a signal processing algorithm that separates a signal into its independent components. It is based on an assumption that a signal has few independent components that are mutually independent. ICA is transformations that rely on statistics of the given data set. ICA is based on the information given by high order statistics. Therefore the result obtained by ICA is assumed to be more meaningful. ICA is often perceived as an extension of PCA. ICA has recently become popular tool in various fields, e.g. blind source separation, feature extraction, telecommunication, finance, text document analysis, seismic monitoring and many others. All successive ICA experiments were designed in MATLAB environment using FastICA package proposed by Aapo Hyv¨rinen et al.

Generally, ICA assumes a finite number of components or so called source signals which is represented by s(t) where s(t) is further represented as:

$$s(t) = [\ s_1(t)\ , s_2(t), \dots\dots\dots\ , s_m(t)]^T \qquad 2.1$$

and each component is mutually independent. Here, t is a discrete time index, m is the number of components and $[.....]^T$ means transpose of row vector. These components are linearly mixed through unknown m x n matrix A and n sensors observe and record the mixed signals

$$x(t) = As(t). \qquad 2.2$$

ICA algorithm finds a separating n x n matrix W that extracts independent components from the observed signals:

$$y(t) = Wx(t) \qquad 2.3$$

It would be ideal if y(t) equals s(t). However, the estimated signals come out in a random fashion, and their amplitudes are not the same, because we have or few information of sources.

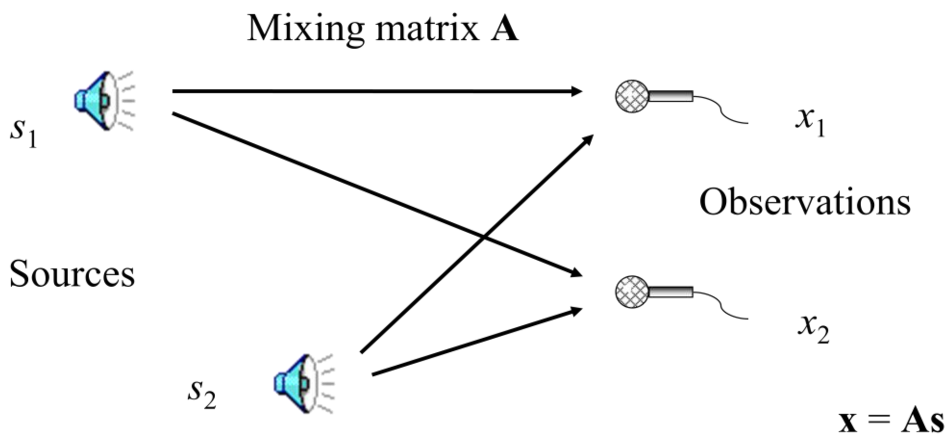


Fig 3. Pictorial representation of how ICA works, speech signal separation problem

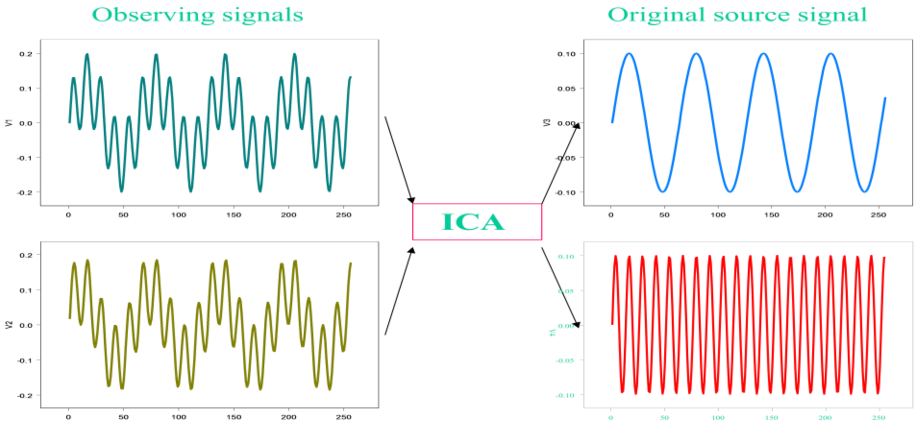


Fig 4. The speech signals separated by ICA (in signal processing field)

General strategy for implementing ICA

The general strategy underlying ICA is given in various literatures and it is summarized as:

- It is assumed that different physical processes give rise to unrelated source signals. Source signals are then assumed to be statistically independent.

- A measured signal usually contains contributions from many different physical sources, and therefore consists of a mixture of unrelated source signals.

-It is assumed that if a set of signals with “maximum entropy” can be recovered from a set of mixtures then such signals are independent.

-In practice, independent signals are recovered from a sets of mixtures by adjusting the separating matrix W until the entropy of the fixed function (say, g) of the signals recovered by W is maximized.[“g”is assumed to be the cumulative density function (cdf) of the source signals.] Hence, the independence of a signals recovered by W is achieved indirectly, by adjusting W in order to maximize the entropy of a function g of signals recovered by W (as maximum entropy signals are independent).

2.5 Applications of ICA

In this section we review some applications of ICA. The most classical application of ICA, the cocktail-party problem, was already explained above.

2.5.1 Separation of Artifacts in MEG Data

Magnetoencephalography (MEG) is a noninvasive technique by which the activity or the cortical neurons can be measured with very good temporal resolution and moderate spatial resolution. When using a MEG record, as a research or clinical tool, the investigator may face a problem of extracting the essential features of the neuromagnetic signals in the presence of artifacts. The amplitude of the disturbances may be higher than that of the brain signals, and the artifacts may resemble pathological signals in shape.

In (Vigário et al., 1998), the authors introduced a new method to separate brain activity from artifacts using ICA. The approach is based on the assumption that the brain activity and the artifacts, e.g. eye movements or blinks, or sensor malfunctions, are anatomically and physiologically separate processes, and this separation is reflected in the statistical independence between the magnetic signals generated by those processes.

The results show that using the ICA technique and the FastICA algorithm, it is possible to isolate both eye movement and eye blinking artifacts, as well as cardiac, myographic, and other artifacts from MEG signals. The FastICA algorithm is an especially suitable tool, because artifact removal is an interactive technique and the investigator may freely choose how many of the IC's he or she wants.

2.5.2 Finding Hidden Factors in Financial Data

It is a tempting alternative to try ICA on financial data. There are many situations in that application domain in which parallel time series are available, such as currency exchange rates or daily returns of stocks, that may have some common underlying factors. ICA might reveal some driving mechanisms that otherwise remain hidden. In a recent study of a stock portfolio (Back and Weigend, 1997), it was found that ICA is a complementary tool to PCA, allowing the underlying structure of the data to be more readily observed.

2.5.3 Reducing Noise in Natural Images

The third example deals with finding ICA filters for natural images and, based on the ICA decomposition, removing noise from images corrupted with additive Gaussian noise. ICA makes use of the statistical data of the image unlike other methods.

2.5.4 Telecommunications

Another emerging application area of great potential: telecommunications. An example of a real-world communications application where blind separation techniques are useful is the separation of the user's own signal from the interfering other users' signals in CDMA (Code-Division Multiple Access) mobile communications. This problem is semi-blind in the sense that certain additional prior information is available on the CDMA data model. But the number of parameters to be estimated is often so high that suitable blind source separation techniques taking into account the available prior knowledge provide a clear performance improvement over more traditional estimation techniques.

ICA is a very general-purpose statistical technique in which observed random data are linearly transformed into components that are maximally independent from each other, and simultaneously have "interesting" distributions.

ICA can be formulated as the estimation of a latent variable model. The intuitive notion of maximum nongaussianity can be used to derive different objective functions whose optimization enables the estimation of the ICA model. Alternatively, one may use more classical notions like maximum likelihood estimation or minimization of mutual information to estimate ICA; somewhat surprisingly, these approaches are (approximately) equivalent. Applications of ICA can be found in many different areas such as audio processing, biomedical signal processing, image processing, telecommunications, and econometrics.

Below are two major variants of ICA algorithm namely: FICA(Fast Independent Component Analysis) and NG-FICA(Natural Gradient FICA). As FICA was not able to solve the problem of enhancement of inscriptions, we used the NGFICA method for the proposed work.

Fast ICA algorithm: Fast ICA algorithm is based on the maximum principle of non-Gaussian character, uses fixed-point iterative theory to look for non-Gaussian character maximum of WT_x, this algorithm adopts Newton iterative algorithm and carries out batch to amount of sampling points of observed variables *x*, isolates a independent component from observation signal every times. In order to reduce the estimate parameters of the algorithm and simplify the calculation of algorithm, before running Fast ICA algorithm, we need carry out data pretreatment, that is removing mean value and bleaching process. The solving process of the Fast ICA algorithm is shown as below: (Jutten and Herault, 1996):

- Randomly selecting chosen initialized weights vector W_0 and k = 0
- Using formula $w^{k+1} = w^{k+1} - \sum_{j=1}^{k} w_{k+1}^T w^j w^j$, 2.4

  $w^{k+1} = w^{k+1} / (w_{k+1}^T w_{k+1})^{1/2}$ to update weights vector w_{k+1} : 2.5
- Normalized w_{K+1} and $w_{k+1} = w_{k+1}/||w_{k+1}||$ 2.6
- If $|w_{k+1}-w_K|>^{\varepsilon}$ then the algorithm is not convergence, return, or Fast ICA algorithm estimate a independent component and the algorithm is over.

2.6 NGFICA

NGFICA (Natural Gradient based flexible ICA) has been extensively used in separating highly correlated signals as it minimizes dependency among the different signals present in the source signal. Mathematical formulation of the same is explained later. The algorithm adopted by the proposed work is NG-FICA (Natural Gradient - Flexible ICA) [5]. NG-FICA uses kurtosis as independency criterion and uses natural gradient for the learning algorithm. This algorithm is implemented as a part of the package, ”ICALAB for signal processing” [7]. In NG-FICA, input data of vector x is applied sphering (prewhitening) as a linear transformation.

z = Q .(x – x'),   Q ={E[(x – x')(x – x')^T]}$^{-1/2}$ 2.7

where vector x' is the mean of x. Q is calculated by Principal Component Analysis (PCA).

The presumption method uses vector z as the input data. The update nonlinear functions are based on the following expressions.

$$\Delta W = \eta\,\Delta W = \eta\,(I - E\,[yy^T - (\varphi y^T + y\varphi^T)])W \quad 2.8$$
$$\varphi_i = |y_i|^{\alpha_i - 1} sgn(y_i)\ \ (i = 1,2,\ldots\ldots,n)$$

where η is the appropriate learning rate (constant number), y is the temporary estimated signal (= Wz), and sgn(y_i) is the signum function of y_i. Gaussian exponent α_i is decided based on the kurtosis κ_i $\left(= \frac{E|y_i^4|}{\{E|y_i^2|\}^2} - 3\right)$ of y_i; α_i is decided near 0 if κ_i is big, but α_i is decided 4 if κ_i is small. Finally, the independent component(s) y is estimated as:

$$y = WQx. \quad 2.9$$

2.7 Methodology

Our work deals with the enhancement of images so that they can be recognized by their respective language OCR's. The need to enhance the images of inscription arises because of the non-recognition or nearly 0% recognition of inscription-text on their respective language OCRs. Thus, we have tried to apply independent components separation over the images.

The enhancement method requires the image to be separated into three components. This is achieved using NGFICA based method. This results in three mutually independent components. From these components, a global average is calculated. This global average is used to select the best component out of three for further processing. It is experimentally seen that the component with maximum difference between the consolidated (or global) average and its individual average, gives best results. A set of morphological operations and noise removal techniques are applied to this image which results in distinctive and clear text and non-text regions. The image is then recognized after being passed to OCR.

2.7.1 Finding independent components

Images of inscriptions were noisy, complex because of the earlier stated set of problems. The illumination, shadow etc added to the problem of clarity in images. So, we performed Gaussian smoothing using a 5x5 kernel. This removed small scale noise and other irrelevant details from the image. R, G, B components (red, green, blue) of the smooth image were extracted. On these RGB components NGFICA algorithm was used to separate mutually independent components of the image considered. For reference these independent components were named text layer, non-text layer and mixed layer on the basis of the text part present in them.

The dependency among the sources are minimized by minimizing the partial differential function L(W) as explained above. The three independent components are shown in fig. 13.

2.7.2 Distinction enhancement

As visible in fig 13, the text part cannot be separated from NGFICA output images as range of pixel values of the text region is still distributed over the three output images. Hence a global average (X) of pixel values of three images can be used to determine the threshold of the text region. It was observed that this threshold was not numerical but a criterion based on the maximum difference between X (global average) and individual averages. Image with maximum difference was selected. The rationale behind calculating average of averages was if the object pixels are brighter than the background, they should also be brighter than the average.

The average pixel value of each component is calculated as:

$$x_i = \sum_{x,y=0}^{x=M,\ y=N} (Ii(x,y)/M*N) \qquad 2.7$$

where $I_i(x,y)$ is the pixel value of the component image at location (x,y) of the i^{th} image with size MxN.

Then a global average X using individual averages is calculated as:

$$X = (x_i)/n \qquad 2.8$$

where n is the number of components and i varies from 1 to n. The image selected for further processing and with maximum difference is shown in fig 14. This image is further processed using noise filtering by applying median filtering and then applying morphological operations for further enhancement. Sobel edge detection is applied too for edge detection.

The resultant image is shown in fig 14.

The final algorithm appears as:

- Take the image to be considered
- Extract the three components of that image. That is, the red, green, blue components.
- Pass the components to ICALAB image processing toolbox as input.
- Apply ICA variant called the NGFICA algorithm.
- This results in three independent components of the image
- Store these three components and use the individual averages of the three ICs to calculate a global average.
- Now, the best suitable component out of the three is one which has the individual or local average farthest from the global average. *
- The best component (image) is selected and converted to binary image.
- It is then processed morphologically using operations like dilation, erosion etc. Connected components are sorted based on their sizes and unwanted noisy particles are removed.
- The image after processing is passed to OCR software which now recognizes it as the image has clear distinction between foreground and background and the noise is also removed.

The * represents that it was observed experimentally.

For English language inscriptions taken from India Gate (Delhi), we also calculated word accuracy from our dataset. For this, the image of the blocks (as in fig 7) on which text was inscripted was processed to extract individual words using word segmentation method described below.

- Take the image for e.g. fig 7
- Calculate a threshold, h, using the average height of words in the image.
- Use histogram equalization for increasing the contrast between the text and non-text (background and foreground).
- Convert the image to binary.
- Then using *h*, calculated above, fill the words as shown in fig 8 .
- Use morphological operation such as dilation and erosion to neatly mark the boundaries of words as shown in fig 10.
- Use connected components and its properties to draw bounding boxes around these words.
- Extract them individually and save them as separate images.
- Apply the NGFICA method to these individual word images now to perform enhancement. The results are shown in Chapter 4.

This word segmentation method is only suitably used for English language inscription images of India Gate. The other languages' inscription images had the challenges (as discussed above) which made the word segmentation unessential in those cases.

The morphological operations in the algorithm were chosen according to the images. It included operations like erosion, dilation, dilation after erosion and erosion after dilation. The median filter was applied to smoothen the edges. The OCR software that was used in our work was ABBYY FINEREADER.

Below is the pictorial representation of the NGFICA based algorithm.

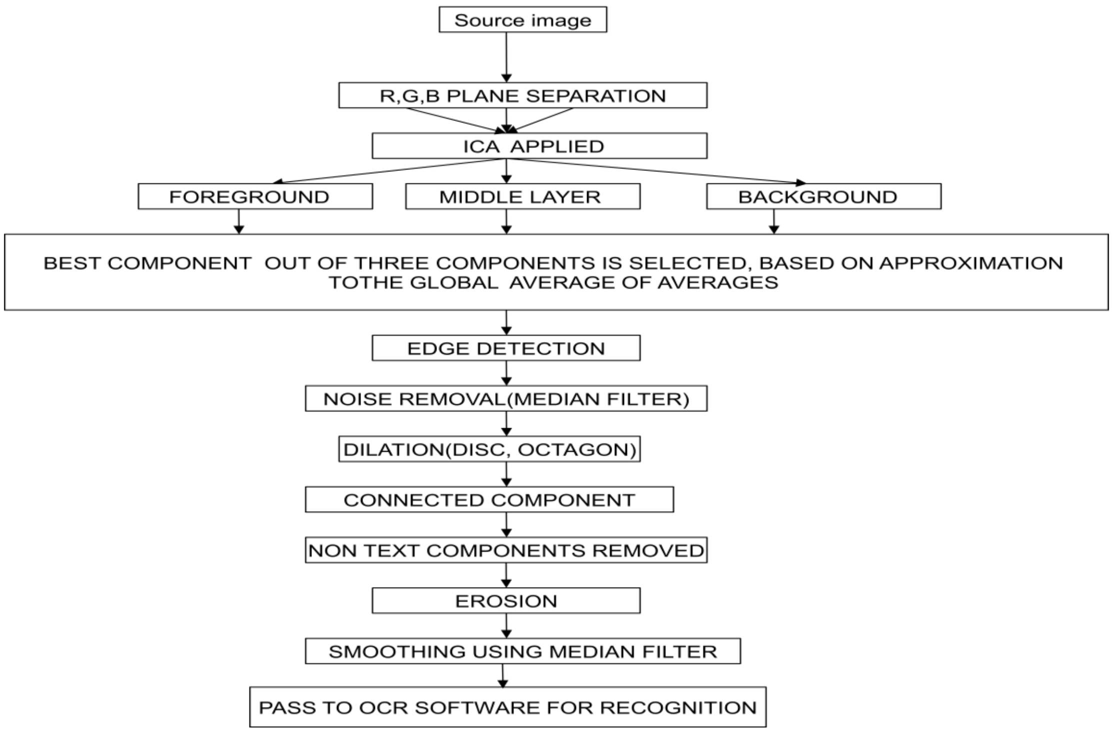


Fig 5:Pictorial representation of the NGFICA based algorithm

Chapter 3: HOG based classification

As stated above, India's rich past and need to preserve this priceless heritage has been a major motivation behind this work. With the help of the proposed work, we have tried to do our best to contribute to the preservation and digitization. Though there remain many unsolved problems but still we have come up with a solution to many of them through our work. The proposed work deals with two aspects. The first one, as discussed in previous chapter, involves text in the form of inscriptions and the second aspect involves graphics in the form of sculptures, which will be dealt in this chapter.

Classification involves recognition also as there are certain particular traits that belong to a particular god. For e.g. an image of lord Hanuman is mainly characterized by the *gada* or the tail in the image. Another example is of lord Krishna, where he holds a flute in his hands.

Thus, we can say that recognizing objects in images is one of the most important problems in computer vision. A common approach is to first extract the feature descriptions of the objects to be recognized from reference images, and store such descriptions in a database. When there is a new image, its feature descriptions are extracted and compared to the object descriptions in the database to see if the image contains any object we are looking for. In real-life applications, the objects in the images to be processed can differ from the reference images in many ways:

- Scale, i.e. size of the object in the image
- Orientation
- Viewpoint
- Illumination
- Partially covered

Sculptures also have similar problems as discussed for inscription images. Though the proposed work does not deal with OCR-ing the sculptures' images but some amount of preprocessing in case of sculptures is still necessary as they are noisy too.

The HOG based classification of sculptures deals with building a classifier that takes up the query image of a sculpture of a god and tells whether it belongs to class A or class B. Here class A and B means sculptures of God A or God B. In our work, we have taken 3 types of Gods namely, Ram, Hanuman and Shiva (*Shivling*). The dataset for the sculptures was created mostly by the images from INTERNET. These were the 2D colored images with mostly colorful backgrounds. This again was a problem in calculating features. So, the dataset was made such that it had mostly those images with clear and uniform backgrounds.

After preparing the dataset, it was divided into train data and test data in the ratio 6:4. HOG features were used and training was done using svm_light on windows 7 32-bit machine. The testing was done later and the accuracy varied between 60-100% on different datasets.

Below is explanation of few terms and algorithms related to the work.

3.1 SIFT

Many features, interesting points or keypoints on the object can be extracted to provide a feature description of the object. This description can then be used when attempting to locate the object in an image containing many other objects. There are many considerations when extracting these features and how to record them. Scale-invariant feature transform (SIFT) is an algorithm for extracting stable feature description of objects call keypoints that are robust to changes in scale, orientation, shear, position, and illumination.

SIFT image features provide a set of features of an object that are not affected by many of the complications experienced in other methods, such as object scaling and rotation.

While allowing for an object to be recognised in a larger image SIFT image features also allow for objects in multiple images of the same location, taken from different positions within the environment, to be recognised. SIFT features are also very resilient to the effects of "noise" in the image.

For calculating SIFT features of an image, the image is transformed to "large collection of local feature vectors" (From "Object Recognition from Local Scale-Invariant Features", David G. Lowe). These feature vectors are invariant to any scaling, rotation or translation of the image. SIFT algorithm applies a 4 stage filtering approach to find the features of an image:

- Scale-Space Extrema Detection
- Keypoint Localistaion
- Orientation Assignment
- Keypoint Descriptor

According to [20], SIFT descriptor is a coarse description of the edge found in the frame. Due to canonization, descriptors are invariant to translations, rotations and scalings and are designed to be robust to residual small distortions.

Scale Space is $L(x, y,\sigma) = G(x, y, \sigma) *I (x, y)$ 3.1

1.Scale space extrema detection: A sequence of coarser pictures are generated then DOG is used to identify potential interest points that are invariant to scale and orientation.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \approx \nabla^2 G \qquad 3.2$$

Find keypoint from maximum and minimum of D over neighboring pixels and scales above and below.

2. Keypoint Localization: Reject low contrast points and eliminate edge response.

$$D(x) = D + \frac{\partial D^T}{\partial x} + \frac{1}{2}x^T \frac{\partial^2 D}{\partial x^2} x \quad \text{at } (x, y, \sigma) \qquad 3.3$$

set derivative equal to zero, gives extremum point

$$\ddot{x} = -\left(\frac{\partial^2 D}{\partial x^2}\right)^{-1} \frac{\partial D}{\partial x} \quad D(\ddot{x}) = D + \frac{1}{2}\left(\frac{\partial D}{\partial x}\right)^{-1} \ddot{x} \qquad 3.4$$

Derivatives are approximated by finite differences. if $D(\ddot{x})$ is below threshold eliminate this keypoint. Hessian matrix is used to compute curvature and eliminate keypoints that have a large principal curvature across the edge but small curvature perpendicular.

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \quad \frac{D_{xx} + D_{yy}}{D_{xx}D_{yy} - D_{xy}^2} < \frac{(r+1)^2}{r} \qquad 3.5$$

Where $r = \frac{largest\_eigenvalue}{smallest\_eigen\_value}$. If inequality fails remove keypoints.

3. Orientation Assignment: An Orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint in order to get an orientation assignment.

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \qquad 3.6$$

$$\tan(\Theta) = \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \qquad 3.7$$

gradient magnitude and orientation of scale L(x,y)

4. Keypoint Descriptor: gradient histogram is formed from gradient orientations around keypoint at every 10 ie 36 directions. Each sample in the histogram is weighted by its gradient magnitude and by a Gaussian window with σ 1.5 times scale of the input. Find highest peak in histogram and other local peaks with orientation of dominant orientation of the key point. So

there can be keypoints with the same location and scale but several orientations.

Each keypoint is described in a 16x16 region. In each 4x4 subregion calculate the histograms with 8 orientations bins. Every seed point is a 8 dimensional vector. So we have a 4x4 array of histograms with 8 orientation bins in each so a descriptor has 4x4x8=128 dimensions.

To compare images we compare their descriptors using Euclidean distance. We use a sunset of keypoints that agree on location, scale and orientation of the new image using a generalized Hough transform.

3.2 SURF

The feature finding process is usually composed of 2 steps; first, find the interest points in the image which might contain meaningful structures; this is usually done by comparing the Difference of Gaussian (DoG) in each location in the image under different scales. A major orientation is also calculated when a point is considered a feature point. The second step is to construct the scale invariant descriptor on each interest point found in the previous step. To achieve rotation invariant, we align a rectangle to the major orientation. The size of the rectangle is proportional to the scale where the interest point is detected. The rectangle is then cropped into a 4 by 4 grid. Different information’s such as gradient or absolute value of gradient are then subtracted from each of these sub square and composed into the interest point descriptor.

The SURF feature is a speed up version of SIFT, which uses an approximated DoG and the integral image trick. The integral image method is very similar to the method used in the famous Viola and Jones’ adaboost face detector. An integral image, despite its pretty name, is just an image which its each pixel value is the sum of all the original pixel values left and above it in a gray scale image. It is three times faster than SIFT. The advantage of integral image is that after an image is computed into an integral image, it can compute block subtraction between any 2 blocks with just 6 calculations. With this advantage, finding SURF features could be several order faster than the traditional SIFT features. But, it is not as good as SIFT as it does not deal with invariance to illumination changes.

3.3 HOG

The essential idea behind the Histogram of Oriented Gradient descriptors is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The implementation of these descriptors can be achieved by dividing the image into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. The combination of these histograms then represents the descriptor. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block.

The HOG features are widely used for object detection. HOG decomposes an image into small squared cells, computes a histogram of oriented gradients in each cell, normalizes the result using a block-wise pattern, and returns a descriptor for each cell. Stacking the cells into a squared image region can be used as an image window descriptor for object detection, for example by means of an SVM.

Based on the above explanation, HOG feature extraction consists of many histograms of orientated gradients in localized areas of an image. A detailed theoretical overview of the algorithm is as follows:

Compute a Histogram of Oriented Gradients (HOG) by

1. (optional) global image normalization
2. computing the gradient image in x and y
3. computing gradient histograms
4. normalizing across blocks
5. flattening into a feature vector

The first stage reduces the influence of light (background or foreground) using an optional global image normalization equalization that is designed specifically for this purpose. Gamma compression or power law is used by either computing the square root or the log of each color channel. Image texture strength is typically proportional to the local surface illumination so this compression helps to reduce the effects of local shadowing and illumination variations.

The second stage computes first order image gradients. These capture contour, silhouette and some texture information, while providing further resistance to illumination variations. The locally dominant color channel is used, which provides color invariance to a large extent. Variant methods may also include second order image derivatives, which act as primitive bar detectors - a useful feature for capturing, e.g. bar like structures in bicycles and limbs in humans.

The third stage aims to produce an encoding that is sensitive to local image content while remaining resistant to small changes in poses or appearance. The adopted method pools gradient orientation information locally in the same way as the SIFT [20] feature. The image window is divided into small spatial regions, called “cells”. For each cell we accumulate a local 1-D histogram of gradient or edge orientations over all the pixels in the cell. This combined cell-level 1-D histogram forms the basic “orientation histogram” representation. Each orientation histogram divides the gradient angle range into a fixed number of predetermined bins. The gradient magnitudes of the pixels in the cell are used to vote into the orientation histogram.

The fourth stage computes normalization, which takes local groups of cells and contrast normalizes their overall responses before passing to next stage. Normalization introduces better invariance to illumination, shadowing, and edge contrast. It is performed by accumulating a measure of local histogram “energy” over local groups of cells that we call “blocks”. The result is used to normalize each cell in the block. Typically each individual cell is shared between several blocks, but its normalizations are block dependent and thus different. The cell thus appears several times in the final output vector with different normalizations. This may seem redundant but it improves the performance. We refer to the normalized block descriptors as Histogram of Oriented Gradient (HOG) descriptors.

The final step collects the HOG descriptors from all blocks of a dense overlapping grid of blocks covering the detection window into a combined feature vector for use in the window classifier.

In the above explanation, computation of HOG feature extraction is divided into the following three major mathematical steps. They are as follows:

Gradient computation

In HOG feature extraction, first of all, 1st order differential coefficients, $Gx(i,j)$ and $Gy(i,j)$, are computed by the following equations.

$$G_x(i,j) = f(i+1,j) - f(i-1,j) \quad 3.8$$

$$G_y(i,j) = f(i,j+1) - f(i,j-1) \quad 3.9$$

Where f(i,j) means luminance at (i,j)

Then magnitude m and direction Θ of the computed gradients are computed by the following expressions, respectively.

$$m(i,j) = \sqrt{G_x(i,j)^2 + G_y(i,j)^2} \quad 3.10$$

$$\Theta(i,j) = \arctan\left(\frac{G_x(i,j)}{G_{y(i,j)}}\right) \quad 3.11$$

Histogram generation

After obtaining the values of m and θ , histograms are generated as follows:

1) Determine the class whichθ (i, j) belongs to

2) Increase the value of the class determined by step 1)

3) Repeat above operations for all gradients belong to the cell.

In order to reduce the effect of aliasing, the values of two neighboring classes are increased. The increment make n indicates a class number whichθ (i, j) belongs to, and n+1 would be the class which is the nearest one to class n. The increased values mn and mn+1 are computed as follows:

$$\begin{cases} n = \lfloor \frac{b\Theta(i,j)}{\Pi} \rfloor \\ m_n = (1-a)m(i,j) \\ m_{n+1} = am(i,j) \end{cases} \qquad 3.12$$

Where b indicates the total number of classes, α is a parameter for proportional distribution of magnitude $m(i,j)$ which is defined as the distance fromθ (i, j) to class n and n+1,

$$a = \frac{b}{\pi}\left(\Theta(i,j) mod \frac{\pi}{b}\right) \qquad 3.13$$

Histogram normalization

Finally, a large histogram is created by combining all generated histograms belonging to a block consists of some cells.

In order to reduce the influence of variations in illumination and contrast, L1-norm is adopted in this paper. After obtaining the large combined histogram, it can be normalized as follows:

$$v = \frac{V_k}{||V_k|| + \varepsilon} \qquad 3.14$$

where V_k is the vector corresponding to a combined histogram for the block, ε is a small constant, and v is the normalized vector, which is a final HOG feature.

3.4 SVM^{light}

SVM^{light} is an implementation of Support Vector Machines in C for the problem of pattern recognition, regression and learning a ranking function. It is a fast optimization algorithm which solves classification, regression and ranking problems. It can efficiently handle several hundred-thousands of training examples.

SVM^{light} is easy to use than libsvm. The main advantage is the compact and light code of the classifier. It contains two executables called svm_learn.exe and svm_classify.exe which handles the training and testing part. SVM^{light} is appropriate for a 2-class problem of classification. It is easy to train and use.

3.5 Methodology

Sculpture images were collected from the images present over the INTERNET. The images with less background details were preferred. Few example images have been shown in fig. 8. The HOG based classification of sculptures consists of five main components namely:

1) image acquisition
2) image preprocessing
3) feature extraction
4) classification
5) display result

Each component has the following details:

- **Image acquisition:** This component takes the images of God Ram, Hanuman, Shivling and Goddess Durga for the training and testing purpose. The images with minimal background information and noise are selected. The images have similar upright poses. Datasets are prepared using these images. The images are sorted by grouping images belonging to one God and so on. We now have four sets of images out of which we use two at a time.

- **Image preprocessing:** This component prepares the images in the proper characteristics for processing method. This component works in four sub-modules namely: 1) image cropping 2) image resizing 3) image grooming (optional) and 4) binary conversion. The image is cropped to remove the unwanted and irrelevant part from the image. This is done manually using image editor. Image resizing brings all the images in a standard size to be worked upon. The standard size is chosen to be 250x250. The image is then groomed to remove the unwanted details in it for example, text at bottom, highly illuminated background which creates problem in distinguishing the boundary of object clearly etc. This is an optional step as most of the images selected were clear and didn't have those components. Then the image is converted to binary image for applying feature extraction.

- **Feature Extraction:** This module detects the HOG features of the image. The binary image is used to calculate the HOG features. The HOG features, widely used for object detection, decomposes an image into small squared cells, computes a histogram of oriented gradients in each cell, normalizes the result using a block-wise pattern, and returns a descriptor for each cell. The feature vector is calculated by this module and is a 1-D vector which is stored in the SVM format for testing and training.

- **Classification:** This module uses the features extracted by the previous module. The training and testing files are prepared in the SVM format and the SVM^{light} is trained using that file. The testing is then performed. The accuracies over four datasets are discussed in the next chapter.

- **Display results:** The final module deals with displaying the result using Matlab code. The result is displayed in the form of multiple windows, each with a test image in it. The result is displayed inside the window too by tagging the image as either 'Image classified as God A' or 'Image classified as God B'. The results are also shown in the next chapter.

Below is a pictorial representation of the steps.

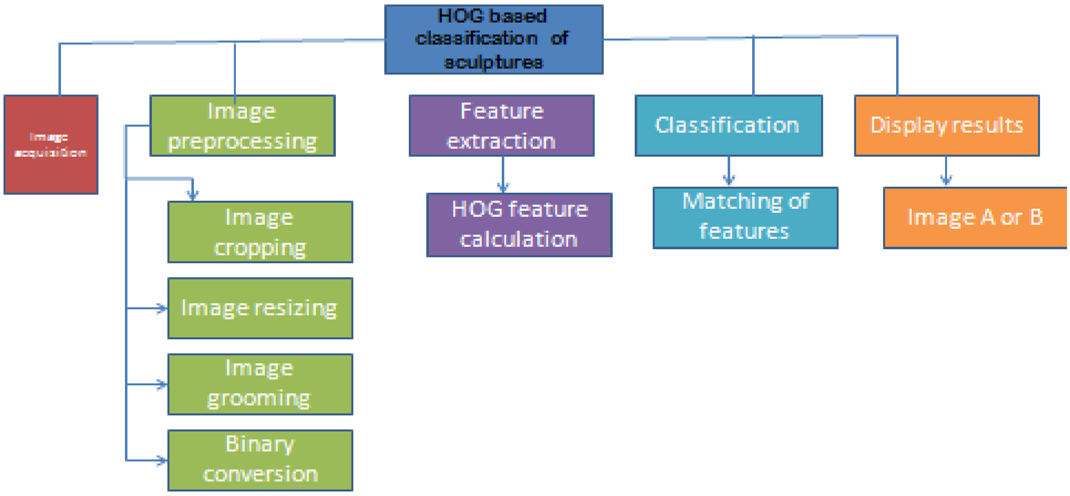
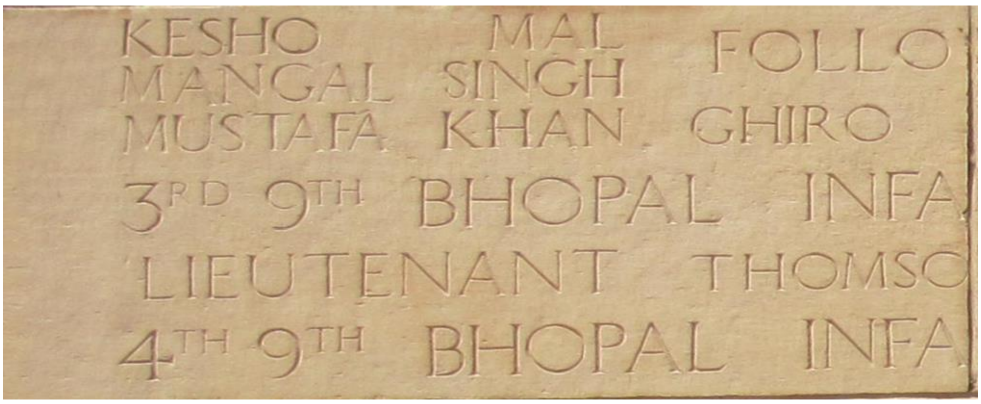


Fig. 6: Pictorial representation of steps for feature based classification of sculptures

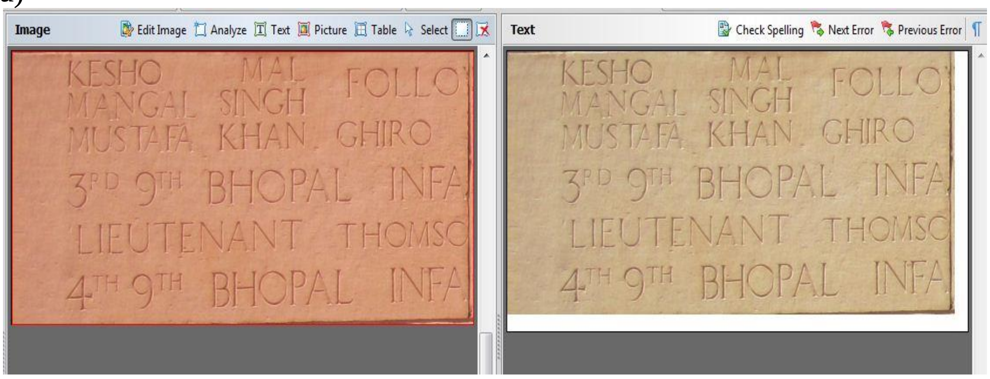
Chapter 4: Results and Conclusions

The dataset for validating the proposed method was prepared by gathering images of inscriptions belonging to historical monuments (India Gate, Delhi), heritage sites (Hampi, Karnataka), ancient temples (Vishnu temple, Tamil Nadu) etc. Such inscriptions are found engraved into or projected out from stones or other durable materials. Some of the images were manually clicked using a 10 mega pixel camera and few were taken from the Internet. Problems like uneven illumination, wrapping, perspective distortion, multi-lingual text etc existed in the images. Images of India Gate(English) without enhancement were tested on web based OCR and the results are shown in table 1 whereas these images after enhancement using proposed method is shown in table 2. The enhanced outputs of India Gate images using the proposed method is shown in fig 11 and 12. We have also compared the proposed method with Fast ICA based enhancement [3] and results are shown in table 3 in fig 16.

A novel method for enhancement of complex and unclear archaeological inscription images has been proposed and validating using 650 word images. This method establishes the important role of NGFICA in digitizing inscription images which has been extensively used for signal processing till now. The method improved character recognition accuracies from 10.1% to 75.4% and from 32.4% to 86.7% respectively. The method enhanced multi-lingual inscription images efficiently. This method can be further extended for digitization of ancient coins, manuscripts and archaeological sculptures.



a)



b)

Fig 7: a) Image of inscription from India Gate and (b) its recognition from a commercially available OCR, showing 0% recognition before enhancement

Fig 8 :Words marked using threshold, h (as discussed in chapter 2)

Fig 9: Connected component applied on the image containing marked words

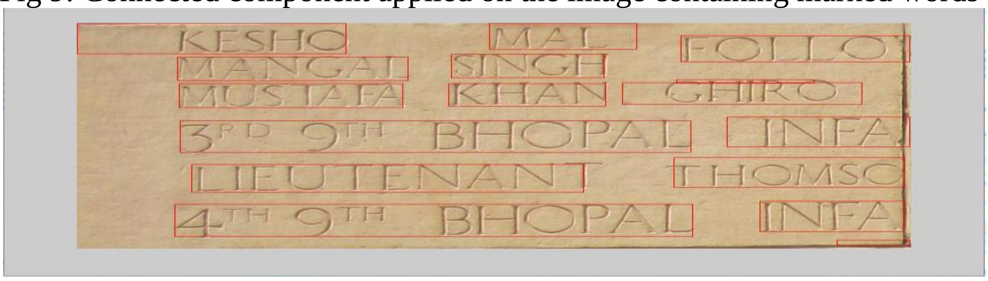
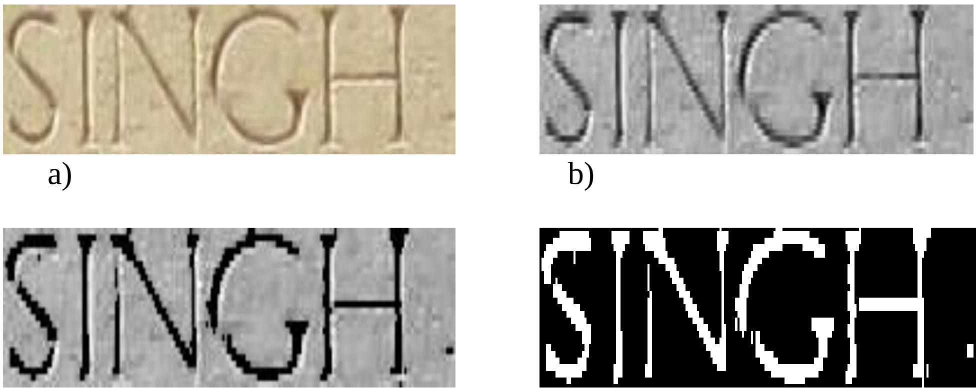


Fig 10: Bounding boxes drawn using connected component property



a) b)

c) d)

Fig 11: a) Individual word b) best IC c) enhanced and d) OCR output

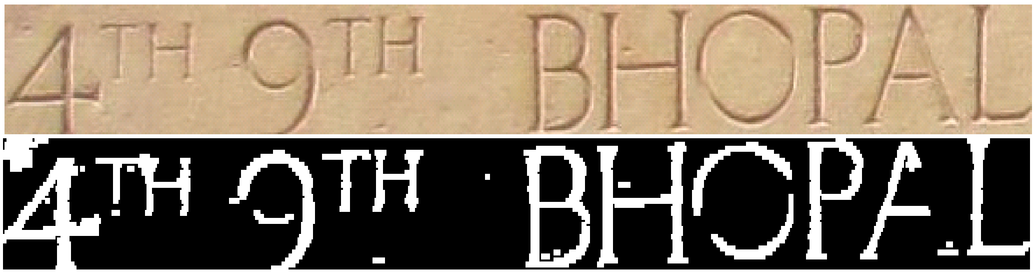
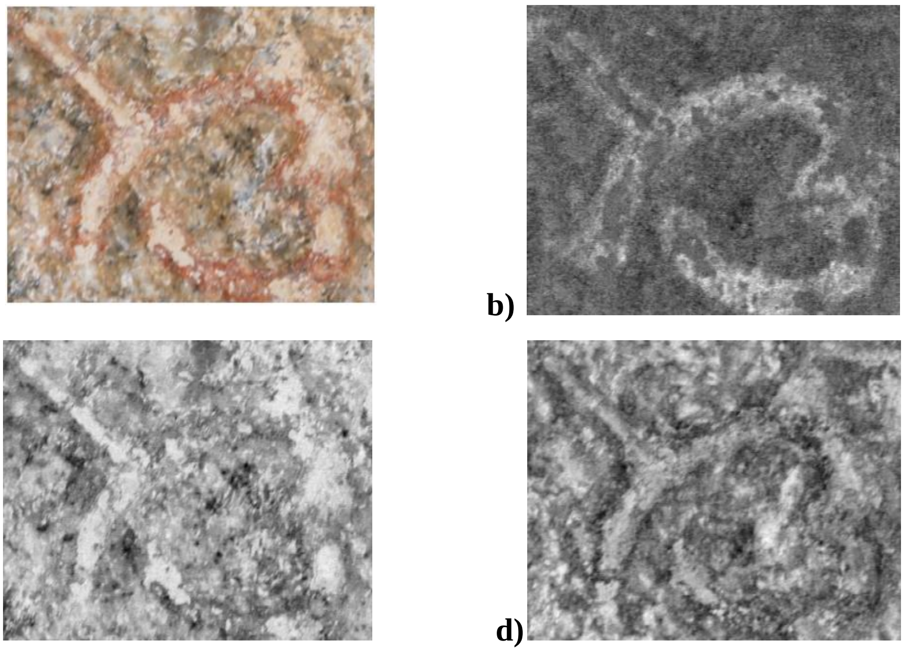


Fig. 12. Image of inscriptions and corresponding OCR output after proposed method of enhancement

Method	Accuracy of words	Accuracy of characters
ICA	**0.9%**	**1%**
Proposed Method	**75.4%**	**86.7%**

Table I Comparison of accuracies before and after the proposed method



a) **b)** **c)** **d)**

Fig. 13. a) Source image (b) (c) and (d) NGFICA output images

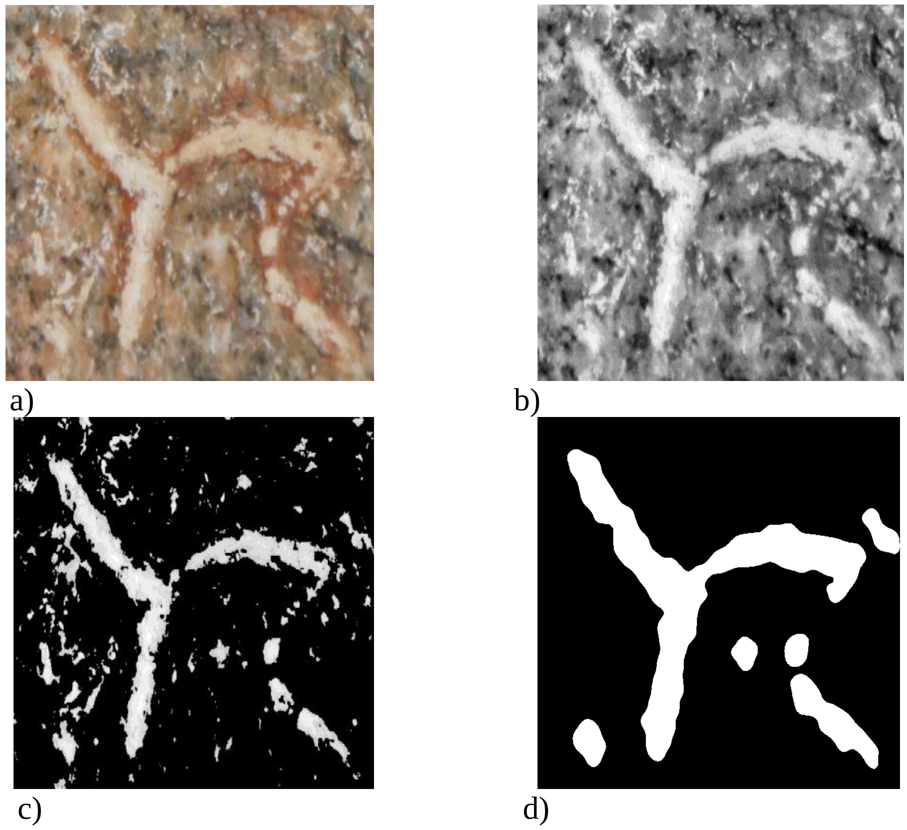


Fig 14: a) Source Image b) best IC selected c,d) during and after processing

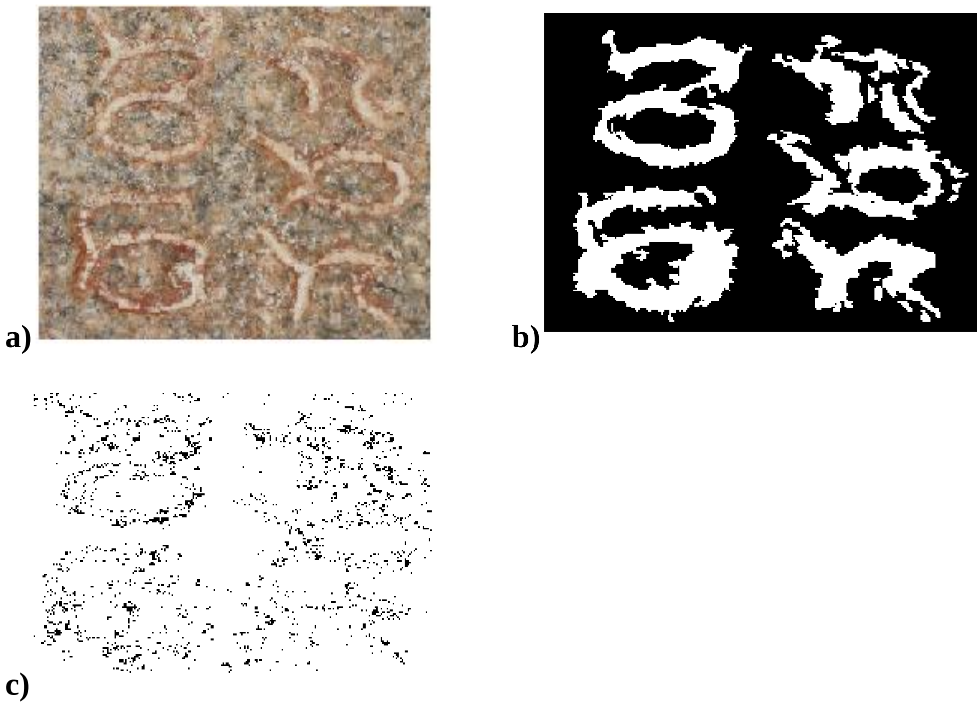
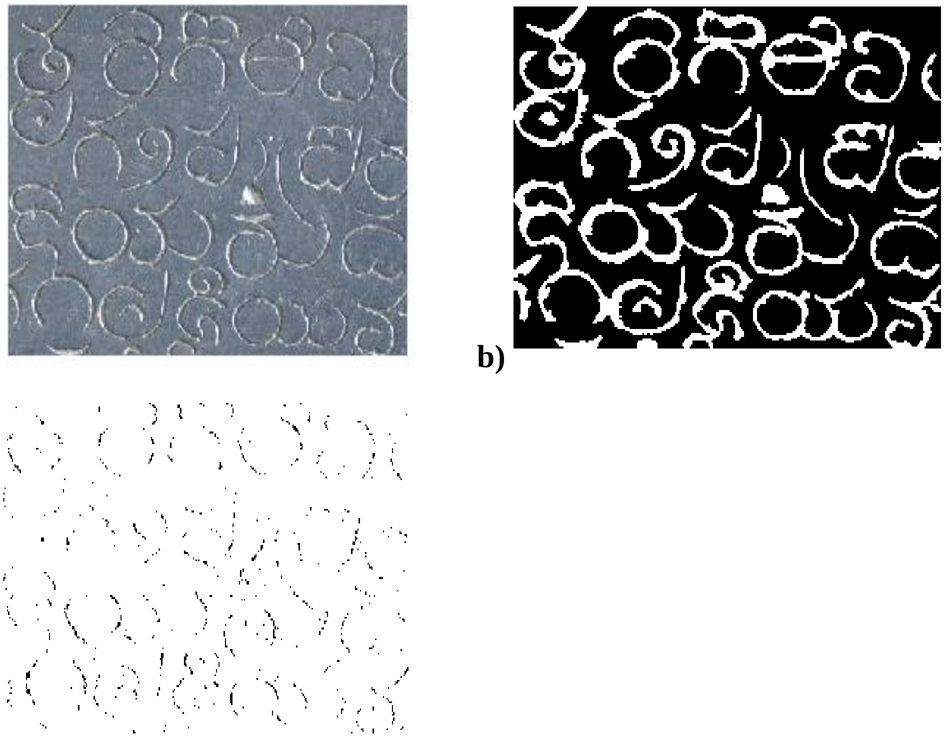


Fig. 15: (a) Original image (b) Output image of proposed method and (c) Output after Fast ICA based enhancement

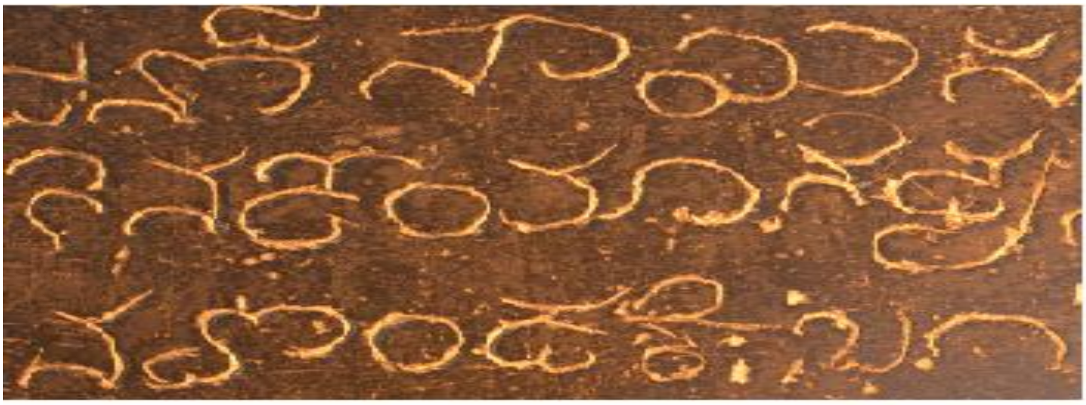


a) b) c)

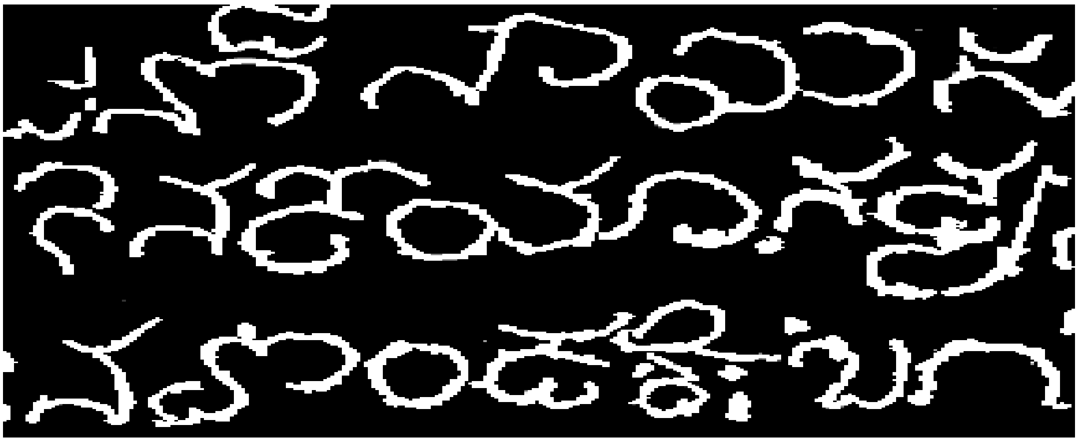
Fig. 16. (a) Original image (b) Output image of proposed method and (c) Output after Fast ICA based enhancement

Input Image	**Number**	**Rec. by OCR before enhancement**	**Accuracy in % before enhancement**	**Rec. by OCR after enhancement**	**Accuracy in % after enhancement**
Words	550	56	10.1%	415	75.4
Characters	2578	835	32.4%	2235	86.7

TABLE II OCR's word and character accuracy before and after enhancement



a)



b)

Fig. 17: (a) Source (b) result

Fig. 18. Other language results

Sculptures are another area of our problem. The HOG based classification of sculptures classifies the sculpture image of God as God A or God B. For e.g. we have images of God Ram and God Hanuman and we want to classify them as God Ram or God Hanuman so that we know which image is of God Ram and which is of Hanuman. Here, we have considered the dataset which comprises of four different types of images of four different Gods. We have worked on the images of God Hanuman, God Ram, Goddess *Durga* and the images of *Shivling*. The dataset was made of images from INTERNET. It was difficult to get the sculpture images of all the Gods so the 2D colored images were considered for preparing the dataset.

Images were resized to a standard size of 250 x 250 after removing the undesired background details and irrelevant objects (like text at the bottom or at the background, some designer frames at the edges etc.). Then the images were converted to binary images. The HOG features were extracted and used for training svm_light as explained in chapter 3. Fig. 19 shows few images from the dataset. For the HOG based classifier, the total number of images was nearly 100. The images were standardized and then used in the proposed algorithm.

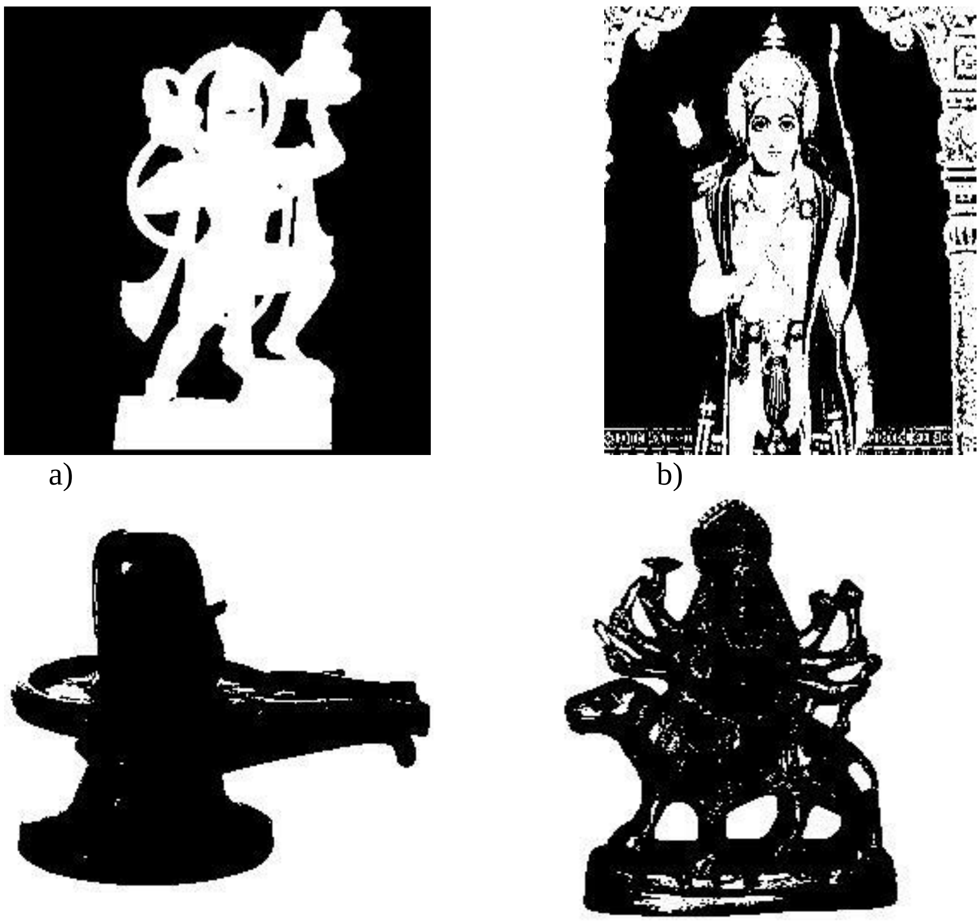
a) 

b) 


c


d)

Fig 19. Images of a) God Hanuman, b) God Ram, c) *Shivling* and d)Goddess Durga



a) b)

c) d)

Fig. 20: Binary images of a) Hanuman, b) Ram, c) Shivling and d) Durga

Fig. 18 shows the binary images of the source images in fig 17. Fig 21 shows one of the test images (of God Hanuman) being classified as God Hanuman. This is one example. The entire test dataset was classified one by one by the classifier. The result displayed windows opening one by one, each containing a test image with a line above it stating what exactly is it classified as? Fig 21 displays the glimpse of the final output of the classifier.

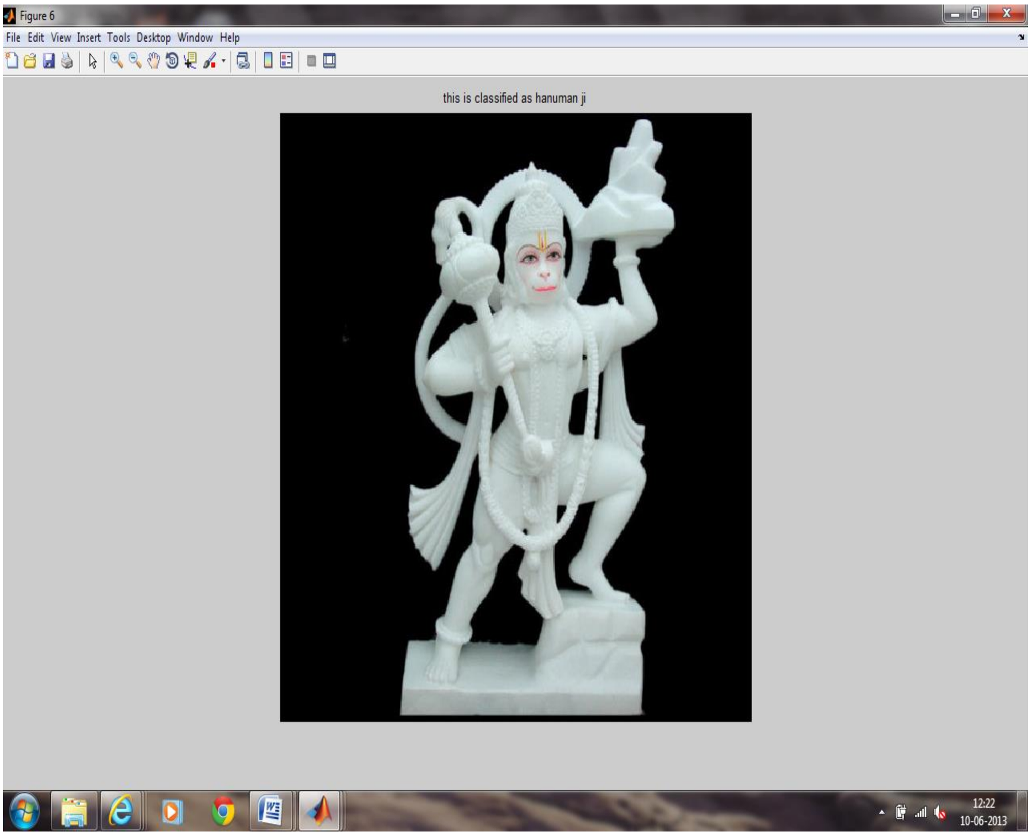


Fig. 21: Output of classifier on passing image of God Hanuman as test image.

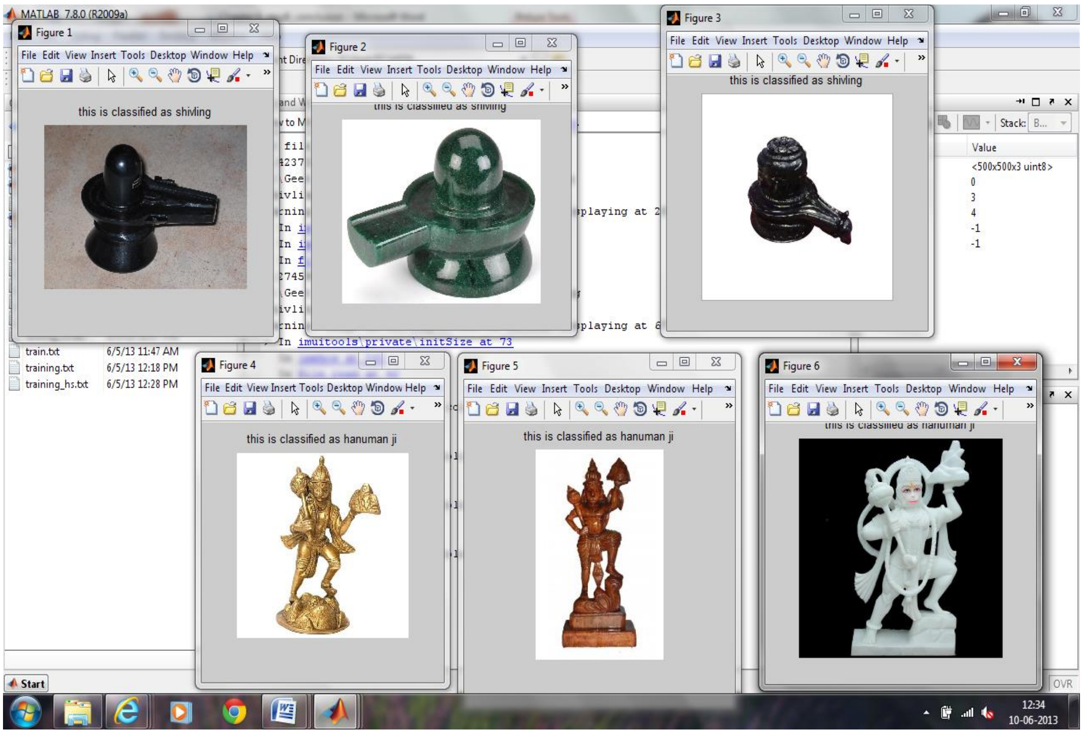


Fig. 22: Output of classifier on passing multiple images of *Shivling* and God Hanuman

Similar results are shown in Fig. 23, 24 and 25 also. The difference is that they are on other datasets.

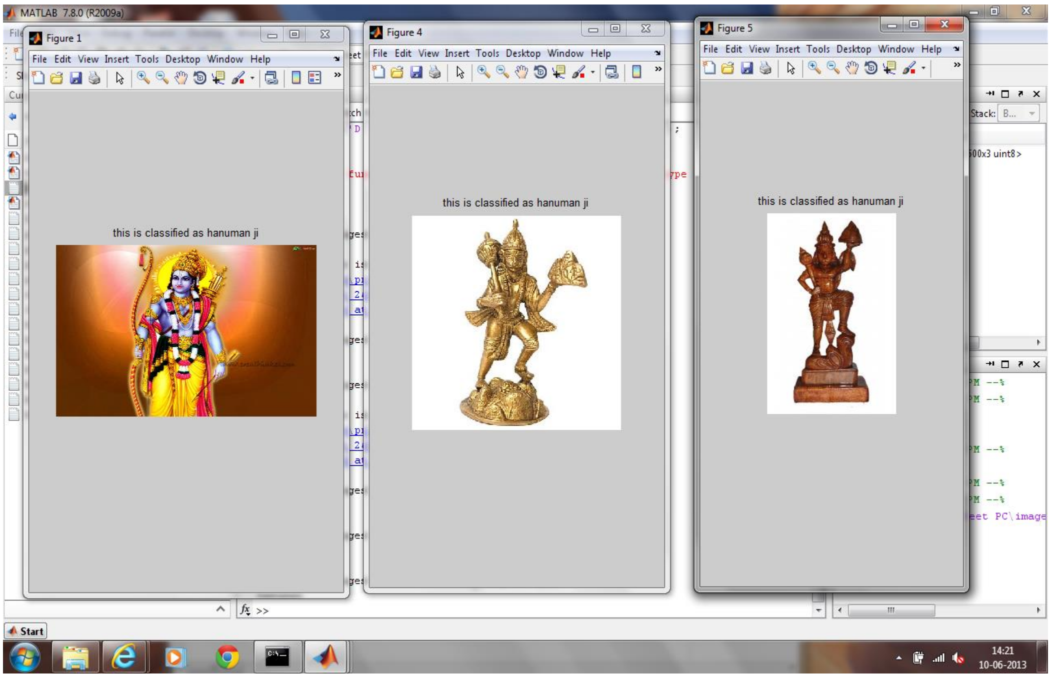


Fig. 23: Another dataset comprising of images of God Ram and Hanuman

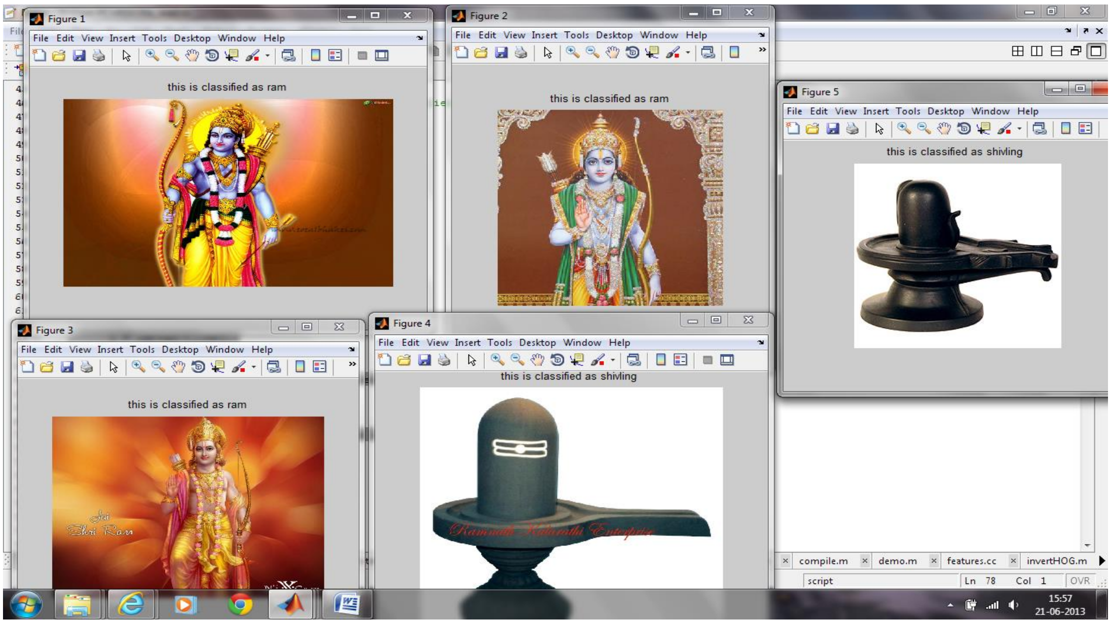


Fig. 24: Another dataset comprising of images of God Ram and *Shivling*

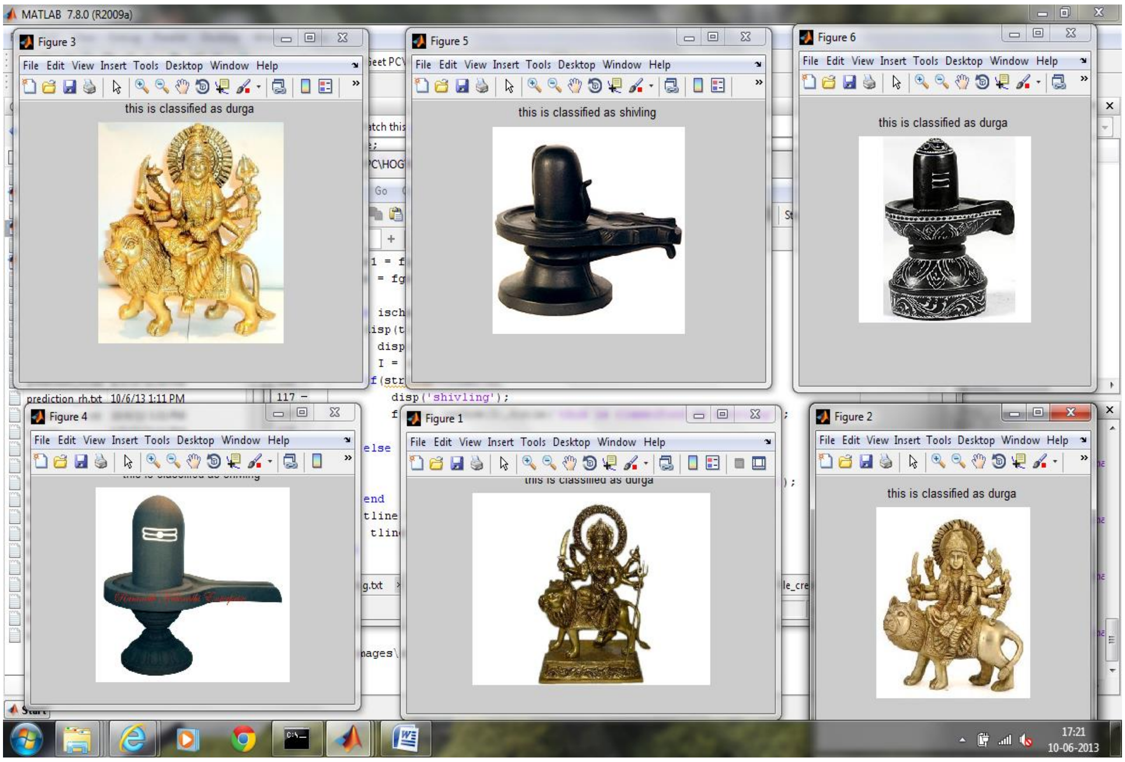


Fig.25:Another dataset comprising of images of Goddess Durga and*Shivling*

The accuracies on different dataset ranged from 50-100%. The variation in the accuracy was observed due to the backgrounds which created scope for non-required information. This created hindrance in calculating HoG features for the most appropriate parts. For e.g. in the image of Lord Ram, the background light was such that most of the features and edges got affected and the distinction between the boundary and the background became unclear when we converted the image to binary. This challenge can be met by applying a more advanced and improved level of image preprocessing.

Table III shows the accuracies of different datasets that the proposed method uses. They are as follows:

Dataset (Combinations of God A and B)	**Accuracy by svm_light(%)**
God Ram and Hanuman	50
God Hanuman and Shivling	100
Goddess Durga and Shivling	85.71
God Ram and Shivling	50

Table III representing the accuracies of the classifier on different types of datasets

It was observed that the results on datasets involving images of God Ram were least accurate. This was due to the reason explained above. Also, the best result was shown on the dataset of God Hanuman and *Shivling*. The second best result was that of dataset comprising of the images of Goddess *Durga* and *Shivling*.

HOG worked well on our dataset as most of the images were like human figures and in an upright position too. SIFT would not have solved the problem as it is restricted the scale, rotation invariant features. HOG covers up scale, rotation invariance factor together with being advantageous in terms of speed. HOG takes almost no time in calculating the features. This made the use of HOG more suitable.

The proposed work is useful in a way of understanding and learning about the Gods/Goddesses of India. The classifier can be used to identify/recognize many more sculptures from ancient temples and historical monuments.

Chapter 5: Future work

Historical monuments are a great source of information of past. The work described here deals with inscriptions and sculptures present on walls of monuments, temples etc. They need to be nurtured and preserved. Many monuments of historical importance have various inscriptions in different languages. So, it is an important step to work in this direction. Like the inscriptions dealt with in this work, many other inscriptions from ancient temples, monuments etc hold a precious place and value in India's history. These monuments hold an importance to the tourism departments of respective states. So, work can be done for increasing tourism and adding convenience to tourists visit to these monuments by either converting these texts to multiple languages for better understanding or designing Braille-aided readable text equipments for blinds.

A better system can be designed that can help tourists in reading the inscriptions on the walls in their preferred language. For e.g. a tourist not knowing Tamil might not take interest in visiting a temple which has Tamil inscriptions or where he may find problem in understanding the language. For this purpose, a system can be built that can translate the inscriptions. This will be an innovative step towards taking the Indian cultural heritage to places and make people aware of the past.

Sculptures are a great source of visualizing the past with open eyes. Many of the sculptures that we see in the temples or monuments, tell us about the people of that era, their habits, their way of living, the rulers of that time, the Gods they worshipped and many more things. So, a way of classifying images based on the era to which they belonged can help us join the series of broken events and form a full story. This will help relating to the past in a better way.

Work can be done in designing a multi-class classifier that identifies the sculptures correctly and then tells about them to the tourist. This system can even be used online with the respective site asking the user to upload the image of the sculpture and then similar images and other details will be displayed by the site.

Working in this direction is extremely helpful and useful for our future generations.

Author's Publications

Journal papers

Indu Sreedevi, Rishi Pandey, N. Jayanthi, Geetanjali Bhola, and Santanu Chaudhury, "NGFICA Based Digitization of Historic Inscription Images," ISRN Signal Processing, vol. 2013, Article ID 735857, 7 pages, 2013. doi:10.1155/2013/735857

Conference papers

Indu Sreedevi, Rishi Pandey, N. Jayanthi, Geetanjali Bhola, and Santanu Chaudhury, "Enhancement of inscription images" NCC'13, DOI

10.1109/NCC.2013.6488017

References

[1] S. Kuo and M. Ranganath, "Real time image enhancement for both text and color photo images," ICIP, vol. 1, pp. 159–162, 1995.

[2] C. Wolf, J.-M. Jolion, and F. Chassaing, "Text localization enhancement and binarization in multimedia documents," ICPR, vol. 4, pp.1037–1040, Sept. 2002.

[3] U. Garain, A. Jain, A. Maity, , and B. Chanda., “Machine reading of camera-held low quality text images an ica-based image enhancement approach for improving ocr accuracy,” ICPR, pp. 1–4, June 2008.

[4] X. Wen and D. Luo, “Performance comparison research of the fecgsignal separation based on the bss algorithm,” Research Journal of Applied Sciences Engineering and Technology, vol. 4, no. 16, pp.2800–2804, March 2012.

[5] L. Agnihotri and N. Dimitrova, “Text detection for video analysis,” Proc. IEEE Int. Workshop on Content-Based Access of Image and Video Libraries, pp. 109–113, June 1999.

[6] X. S. Hua, P. Yin, and H. J. Zhang, “Efficient video text recognition using multiple frame integration,” Proc. Int. Conf. Image Processing, vol. 2, pp. 22–25, Sept. 2004.

[7] K. C. K. Kim, “Scene text extraction in natural scene images using hierarchical feature combining and verification,” in Proceedings of International Conference Pattern Recognition, vol. 2, pp. 679–682, August 2004.

[8] G. R. M. Babu, P. Srimayee, and A. Srikrishna, “Heterogenous images using mathematical morphology,” Journal of Theoretical and Applied Information Technology., vol. 15, no. 5, pp. 795–825, November 2008.

[9] EhsanNadernejad, S. Sharifzadeh, and H. Hassanpour, “Edge detection techniques: evaluations and comparisons,” Applied Mathematical Sciences, vol. 2, no. 31, pp. 1507 – 1520, Sept. 2008.

[10] P.Chevalier, L. Albera, P. Comon, and Ferreol, “Comparative performance analysis of eight blind source separation methods on radio communications signals,” Proc. International Joint Conference on Neural Networks, vol. 8, no. 2, pp. 251–276, July 2004.

[11] P. Tichavsky, Z. Koldovsky, and E. Oja, “Performance analysis of the fast ica algorithm and cram rao bounds for linear independent component

analysis," IEEE Transactions on Signal Processing, vol. 54, no. 4, pp. 1189–1203, April 2006.

[12] S. Choi, A. Cichocki, and S. Amari, "Flexible independent component analysis," Journal of VLSI Signal Processing, vol. 26, no. 2, pp.25–38, August 2000.

[13] S. Choi, "Independent component analysis," 12th WSEAS International Conference on communications, pp. 159–162, July 2008.

[14] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," Advances in Neural Information Processing System, pp. 251–276, 1996.

[15] Dalal, Triggs "Histograms of Oriented Gradients for Human Detection" Coference on Computer Vision and Pattern Recognition (CVPR), 2005.

[16] David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

[17] O. Ludwig, D. Delgado, V. Goncalves, and U. Nunes, 'Trainable Classifier-Fusion Schemes: An Application To Pedestrian Detection,' In: 12th International IEEE Conference On Intelligent Transportation Systems, 2009, St. Louis, 2009. V. 1. P. 432-437.

[18] C.Pornpanomchai, V.Arpapong, P.Iamvisetchai and Nattida Pramanus, " Thai Buddhist

Sculpture Recognition System (TBuSRS), IACSIT International Journal of Engineering and Technology, Vol.3, No.4, August 2011.

[19] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. *CVPR*, 2006.

[20] D.G.Lowe, Object recognition from local scale-invariant features. International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157, 1999

[21] F. Han, Y. Shan, R. Cekander, H.S. Sawhney and R.Kumar. "A two-stage approach to people and vehicle detection with HOG-based SVM" In Performance Metrics for Intelligent Systems 2006 Workshop, 2006.

[22] L.Hai-bo, Y.Xin-rong, L.Hong-mei, D. Qing-hai, "A method for real-time implementation of HOG feature extraction.

[23] R.Hu, J.Collomossea, "A performance evaluation of gradient field HOG Descriptor for sketch based image retrieval", Computer vision and Image understanding, 2013.

[24] Likforman-Sulem, L., Zahour, A. and Taconet, B., "Text line Segmentation of Historical Documents: a Survey",International Journal on Document Analysis and Recognition, Springer, Vol. 9, Issue 2, pp.123-138, 2007.

[25] www.mathworks.com

[26] http://www.bsp.brain.riken.jp/ICALAB/ICALABImageProc/

[27] http://www.cvc.uab.cat/~aldavert/plor/

[28] http://chrisjmccormick.wordpress.com/2013/05/07/gradient-vectors/

[29] http://chrisjmccormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/

[30] http://scikit-image.org/docs/dev/auto_examples/plot_hog.html

[31] https://sites.google.com/site/5kk73gpu2011/assignments/sift