

CHAPTER 1

INTRODUCTION

In the recent years we have observed a rapid rise in the size of digital image compilations. Giiga bytes of test images are created daily by both civilian as well as military equipment. However to manage such incredible amount of data pouring in everyday from so many sources we need to have an efficient storage and retrieval system for images. Since the 1970s work on retrieval of images is being actively carried on. Image retrieval can be looked upon from two angles- one his text based and the other being visual based.

Since ancient times inscriptions have been used as a medium of storing and passing the information from one generation to the next. These inscriptions give us an insight into their worlds and carry answers to many of our questions. For ages we have tried to decipher their meanings and determine their origins but with the increasing amount of data and constant digitization of the world it is only natural that this treasure of knowledge is archived as well. Once the inscription images are archived we move to the next step which is the retrieval process. With a database having innumerable images it is necessary to ensure that – i) we have the fastest possible retrieval solution. ii) Next, we need to ensure that the retrieval technique is language independent considering the inscriptions from various cultures are inscribed in separate languages. iii) It should be able to perceive images that have undergone damages such as blurring and darkening over time.

Different methods have been suggested for efficient retrieval, word spotting is popular for recovering the relevant images without recognition. Various word spotting techniques have been used, such as the Dynamic Time Warping (DTW) algorithm [3], Hidden Markov Model (HMM) [2] etc. DTW compares a series of feature vector for image retrieval, though successful in many cases it has a drawback when it comes to large databases, having a computation time of 1 second for comparing two word images hence it loses its practicality. In [5] Zagoris et al. define a segmentation based word spotting technique using document specific local features that was tested on two historical datasets. In [1] Rajakumar and Bharathii suggest a contour let transform to

recognize Tamil characters from stone inscriptions. Sankar and Mamantha [4] have suggested automatic word annotation for document retrieval where the digital image is assigned with a metadata by the system automatically.

The Bag of Visual Words has come across as one of the most efficient techniques for document image retrieval [6] [7] [8] [9] [10]. Augereau et al. [6] talk of combining visual and textural features for document classification and retrieval using BoW and BoVW. Shekhar and Jawahar [13] use the BoVW for image retrieval using SIFT as the feature vector. Aldavert et al. [11] work on the George Washington data set for keyword spotting using BoVW. In [13] it has been observed that SIFT descriptor is not very robust for all possible image degradations and hence it calls out for the need of a more robust descriptor. Nabeel et al. [12] have summarized the performance of SIFT and SURF on several datasets showing SURF to be outperforming SIFT in case of blurred images and darkened images. SIFT though known to work well with images from different datasets is replaced by SURF here which has proved to be more efficient when it comes to the highly blurred inscription images.

With the inscriptions dating back to medieval times the effect of blurring tends to become very prominent in the inscribed texts or pictorial representations. The biggest problem that then arises is – how to retrieve them accurately in the shortest time span possible? To overcome this issue we have come up with an image retrieval procedure that employs the bag of visual words (BoVW) technique using the SURF (Speeded Up Robust Feature) descriptor. SURF has been known to perform exceptionally well with the shortest execution time and hence we have replaced the originally used SIFT (Scale Invariant Feature Transform) with the former technique. SURF works well with blurred and darkened images. The BoVWs technique ensures that the correct output is obtained with its help.

The major advantage of our methodology is that it works well with images that have undergone degradation of different forms like darkening or blurring over time and its language independence. Inscriptions from all over the world belonging to separate timelines in any language can be retrieved using this technique making it highly robust.

1.2 PROBLEM OVERVIEW

The techniques available for document retrieval are not as feasible for document image collections. The techniques used at present are derived from image matching methods which are rather complex and don't give satisfactory results for large databases of images present. The real issue lies in the time required for processing and the word image matching techniques.

A lot of different methods have been proposed in the literature to give access at content level. Mostly these solutions are query. A sample image is taken as the query and images similar to the query are retrieved. The end user however will demand textual query support as is the case with popular search engines like chrome or mozella.

We need a better representation in order to index a huge collection of documents. The methods of word matching are expensive ones and become costlier still as the words are being represented by higher dimension features. Not only does the high dimension vector increase the complexity but also the time taken for indexing. Hence it is critical when the question of large datasets arises. Representation of similar words which may vary with the changes in image size, noise degradations etc. Should be same so that better results are produced while matching.

In our work we assume that pre-processing of document images and segmentation of text images into words, that are required for the recognition as well for the retrieval tasks are available. The study of preprocessing and text block segmentation are beyond the scope of the present work.

1.3 Contribution

In the research work, we focus on image retrieval techniques while working on inscription images as well as while detecting images containing text and their retrieval. Some of our contributions are as follows :

- Proposal of an efficient retrieval technique of word image matching that is based on Bag of Visual Words to compare inscribed images. Our method is a language independent method and can retrieve image written in any language, it is scalable and faster than the existing methods.

- The efficiency of the method has been shown in images in different languages including English and tamil.
- The proposed SVM and MSER based text detection and retrieval method has proved to give very accurate results on images containing text.

1.4 Thesis Organization

This thesis has been organized into five chapters. The first chapter is the introduction of the methods used and the need for image retrieval. It will cover the background and the problem overview. In the second chapter we give a literature survey of the various techniques present out there that have been used for the purpose of image retrieval.

The experimental work done on databases has been elaborated in chapter three. The results and conclusions have been defined in chapter four and five.

CHAPTER 2

BACKGROUND AND PREVIOUS WORK

A large database of handwritten documents, inscription images, manuscript images and printed documents are available electronically online and in digital libraries. For example the catholic university of America holds records of Greek and Latin inscriptions, the Digital Library of India holds scanned documents of different languages. Searching such images is a fastidious task and not practically feasible without a reference or ground truth available. These issues together make a fast access to the images in the database a challenging problem. In early times search was done manually i.e. a person was assigned to the task of indexing of documents. This had high error rate since a man is subject to make a mistake while handling large databases. Optical character recognition (OCR) based systems were introduced later.

OCR converts the document image into text and then the text is used for the purpose of searching. This method has proven to be robust for European languages and works well for them, however for other languages such as Hindi, Urdu, Tamil and Telugu etc. It is not as robust. Even in the case of European languages the OCR for highly degraded images does not work well. Next, the direct image domain matching has been proposed. Here, features are used to represent word images and comparison of these features results in retrieval. In this regard work has been done and image retrieval has been made possible by Dynamic Time Warping (DTW).

2.1 OPTICAL CHARACTER RECOGNITION

The OCR process includes binarization of a scanned image which is then sent for pre-processing steps that consists of removal of noise, separation of text graphics and correction of skew etc. Once this step is over the image is segmented into smaller and smaller units, i.e. into lines, followed by words and characters. Once the character is segmented the feature extraction step follows after which a suitable classifier is used for recognition. The basic architecture of OCR is shown in fig. 2.1

The OCR faces problems when it comes to Indian languages. A few of the issues faced are – the presence of “shirorekh” in Hindi which makes segmentation of characters difficult, the number of characters (unique classes) are high making the designing task harder, vowel modifiers in conjunction with consonants make the character segmentation difficult. A detailed study of challenges in developing OCR is given in [10].

For the OCR retrieval technique indexing is done based on the text gained from the OCR itself. Due to the complexity in the scripts, the error rate is high for Indian and non European script OCR. Therefore the OCR will index incorrect words. Hence, even when the user will provide the right query the OCR will be incapable of retrieving the right word due to error.

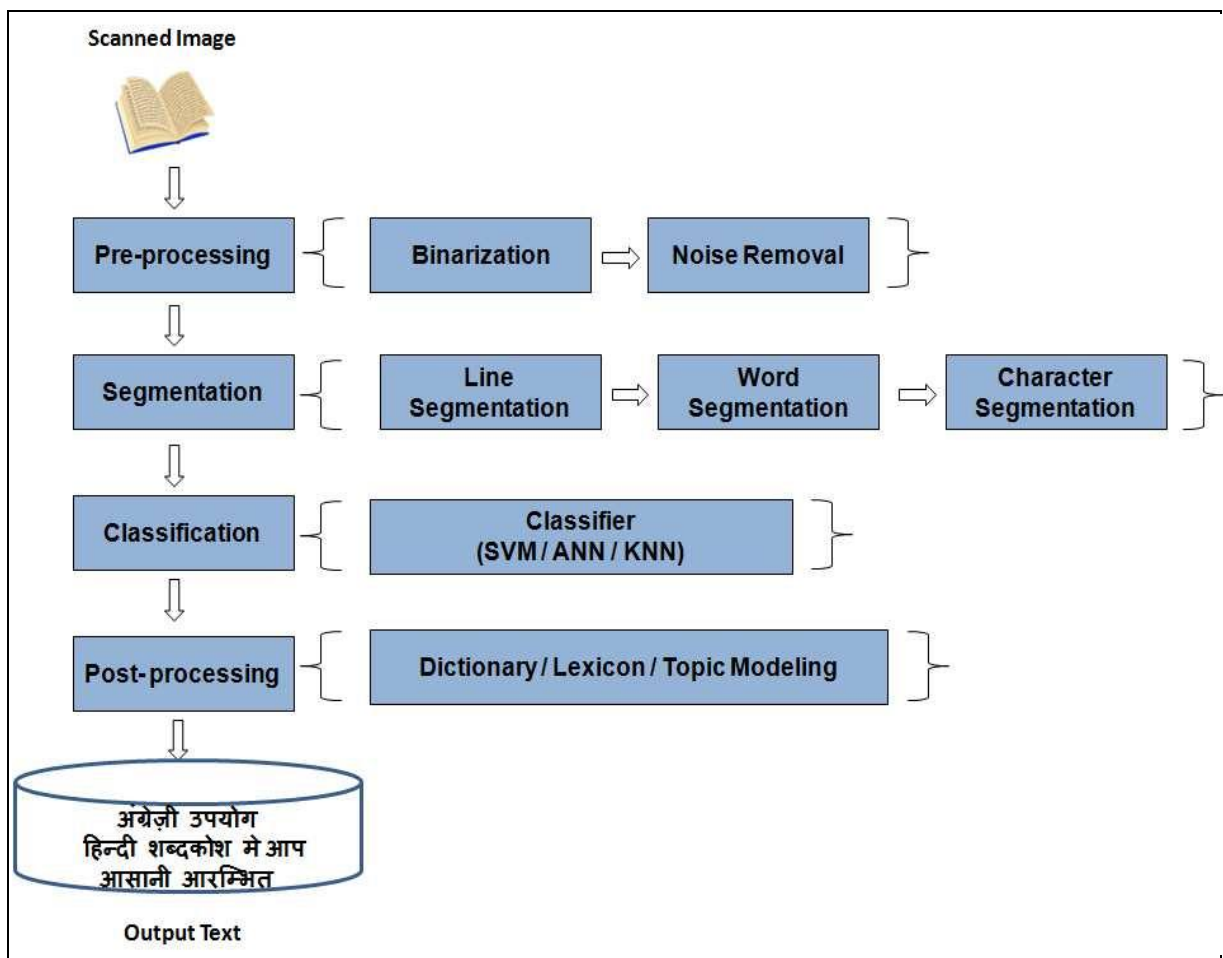


Figure 2.1 Basic OCR Architecture

The properties of OCR make it a good search engine for documents i.e. quick access, efficient, less cost of computation, good accuracy. It is mostly automated and less expensive, giving it a fast access and improved accuracy. So, OCR can be said to

be used for recognition and retrieval of document images, OCR gets document images as input and automatically outputs their digital format.

OCR procedure can be summarized as below:

- Input: Document images
- Preprocessing: It involves binarization, skew detection and normalization. Image enhancement is done to remove noise and increase contrast.
- Segmentation: It involves line segmentation, word segmentation and then character segmentation consecutively.
- Feature extraction : feature extraction makes the document invariant to rotation, translation and scaling etc. Either global or local features can be used.
- Classification: once features are extracted they are utilized in training of classifiers. The classifiers work is to identify the input document and classify it based on the training it has received. Some common classifiers are – Support Vector Machines (SVM), Neural Networks, Hidden Markov Model (HMM), Bayesian classifier and Template matching.
- Post-processing: it is used to improve the results of the document recognition and increase the accuracy.
- Output: text documents.

There are many commercial OCRs available, for example, OmniPage pro from Caere, FineReader from ABBYY, Text Bridge from Xerox and Capture from Adobe, Tesseract from HP labs etc. For some scripts, OCR is efficient system for recognition [28]. OCRs promise high accuracy in case of some Latin languages [16] and some non-Latin languages [17]. But all OCRs perform well for non-degraded high quality document images.

In [26] a survey on indexing and retrieval methods has been given. In this Doermann et al. Has shown retrieval of documents using OCR. It is mostly focussed on English language. OCRs for non Latin and Indian languages are not available leading to high error rate, causing poor accuracy. [64, 43] have discussed approaches to overcome errors in OCR. In the next paragraph we shall discuss the development of OCR in Indian as well as some other languages.

[46] presents a survey of Indian script recognition. In this, pal et al. Maximum development in OCR is concentrated in two languages Bangla and Hindi. For Hindi KNN based OCR and NN based OCR are shown in [14] and [19] respectively. In [19], this work has been extended for Bangla language. Chaudhuinary et al. [20] prosed a complete OCR system for printed bangle language. Their method uses tree classifier and template matching. In [33] Jawahar et al. Used this method for Telugu and Hindi documents. That method is based on PCA followed by SVM. Jitesh et al. [35] developed Malayalam language OCR in which level segmentation, classification and feature extraction, decision tree is used. For post processing spell checkers are used. In [5], Negi et al. Presented an OCR for Telugu using connected component approach.

Cowell et al. [23] Amharic character with an algorithm that uses fast sign. Here the major concern is the developemet of better feature extraction script identification.

In [71] yaregal et al. Gave a method using direction field tensor to realize Ethiopic characters. Primitive features and the spatial relationships are used. Tree structures are used to obtain spatial relationships. Edward et al. [27] has used gHMM for Latin scripts. Here, they convert the script into text using the above mentioned generalised hidden makaov model and retrieve on that basis. 75% of accuracy was reported.

In [64] Taghva made a search engine which functioned on the similarity amongst the query word and the words that are in the database. Ishitani [29] proposed a method which was tolerant of OCR errors. Keyword matching was utilized for search of a string like pattern from the results of OCR. In [18], Chan et al. Proposed method for search of Arabic document images. The segmenting step and recognition process are carried out together. Use of gHMM along with bi-gram transition and Kernel Principle Component Analysis (KPCA) has been suggested for character discrimination. Zhidong et al. [72], presented Bbn Byblos OCR system, in which HMM is used.

For Indian languages [17, 46], and for many non-Latin languages the recognition method are not available. The present methods are neither robust nor

reliable for recognition of those languages. Hence, the need for an alternate approach arises.

Lee et al. [40] proposed an adaptive language image prototype to improve book OCR. Document centric models have been used in both. The model will adapt itself to vocabularies and shapes for correction within the given book. Each of the models exploits the redundancy in font and vocabularies for inconsistency detection. 25% improvement over Tesseract OCR.

Kluzner et al [37] proposed hierarchical optical flow along a second order term for comparison of every input character with a pre existing set of super-symbols using their distance maps.

2.2 Word Spotting

In OCR we first need to recognize each character and then word formulation takes place based on it. So, a major drawback arises since, even if one character is not recognized correctly and entire word can be in error irrespective of the rest of the word. To face this drawback a new approach was proposed that was recognition free. Word spotting is one such approach. In word spotting words are recognized based on their features. Here we treat a document as a collection of images. Therefore, we first segment the document into lines and then the corresponding words. Each of the document will then be indexed based on the visual image feature of the words contained in it. Matching is done for printed [21] as well as online [30] and offline [47] handwritten documents. This is useful when we need to find similar images to the query word.

Rath et al. [47] proposed the location of a specific word in a text written by hand, matching features of image containing this query word. This technique has been furthered [48, 47] for word search in printed document images such as newspapers and magazines. In [30] a Dynamic Time Warping (DTW) based algorithm for word spotting has also been used. DTW is a dynamic technique which is used for programming to align sequences and hence, used for feature matching of the word images. Word images as proposed in [47,49] are matched and then clustered. Then a person annotates each cluster. Jawahar et al. [12, 34] has demonstrated that in the scenario of a printed book the query image can be synthesised from a textual query to

make the system more useful. For simplification purposes we generate a word image corresponding to each query image and then the cluster which corresponds to it will be identified. Efficiency is thus obtained by online computation.

Work in handwritten documents in Ottoman language has been done [11] by Ataer and Duygulu. Gatos [38] used word spotting technique for Greek typewritten manuscripts dating back in time, OCR did not work for these. One of the advantages that word spotting has over the traditional technique of OCR is that the probability of word images to be similar is high in printed books. Traditional OCR do not boast of such advantage. Also, techniques that work on character level of word image [18], are quite sensitive to segmentation. Indian language segmentation poses a challenge on its own being very complex to be segmented.

In scripts like Arabic, Devanagari and Latin word spotting using shapes and features has been demonstrated.

Word similarities are found using Global word shape features during search. Word image features like gradient, structural and concavity which will measure the image characters at local, middle and large scale and therefore get a heterogeneous paradigm to feature extraction. Each image is divided into 4x8 rectangles for feature extraction and each rectangle contains 384 bits of structural features and 256 bits concavity features delivering a binary vector of 1024 length. A normalised correlation similarity measure computes the distance between all the words and the word we need to spot.

In [51] Rusnol et al. have suggested a method free of segmentation for word spotting in heterogeneous documents collections. A patch framework is used for retrieval.

2.3 SVM CLASSIFIER

The support vector machine (SVM) can be defined as a supervised learning technique that produces input/output mapping functions from a set of given training data. We can use either a regression function or a classification function which is the classes of the input function as our mapping function. For the purpose of classification, nonlinear kernel functions are frequently used to change the data to a

high-dimensional feature space in which the input turn out to be more distinguishable or separable contrasted with the first input space. Most extreme margin hyperplanes are then made. The model in this manner created relies on just a subset of the training set close to the class limits. Thus, the model delivered by Support Vector Regression overlooks any training data that is adequately near the model expectation. SVMs are additionally said to have a place with kernel methods.

SVMs are regulated learning models with related learning algorithms that investigate information and perceive patterns, utilized for regression and classification examination. When we give SVM a set of training data each of which has been marked to belong to one of the two classes, a SVM algo fabricates a model that allocates new example into one class or the other, making it a non-probabilistic binary linear classifier .

An SVM model is a representation of the case as points in space, mapped so that the case of the different classes are separated by a distinguishable margin that is as wide as could be allowed. New examples are then mapped into that same space and anticipated to have a place with a category in view of which side of the crevice they fall on.

Support Vector Machine (SVM) finds an ideal arrangement. It boosts the separation between the hyperplane and the "troublesome points" near decision boundary.

SVMs amplify the margin around the isolating hyperplane, and so it is also known as large margin classifiers.

The decision function is completely indicated by a subset of training sets, the support vectors. Explaining SVMs is a quadratic programming issue Seen by numerous as the best current text classification strategy.

SUPPORT VECTORS

Those data points that are found closest to the boundary or the desicion surface are known as the support vectors. Their proximity to the boundary makes them the hardest to get classified and so they have a direct impact on the optimum location for the decision boundary.

Formalization of maximum margin :

- \mathbf{w} : decision hyperplane normal vector
- \mathbf{x}_j : data point j
- y_j : class of data point j (+1 or -1) NB: Not 1/0
- Classifier is: $f(\mathbf{x}_j) = \text{sign}(\mathbf{w}^T \mathbf{x}_j + b)$
- Functional margin of \mathbf{x}_j is: $y_j (\mathbf{w}^T \mathbf{x}_j + b)$
- Functional margin of dataset is twice the minimum functional margin for any point
- On measuring the whole width of the margin the factor of 2 comes.

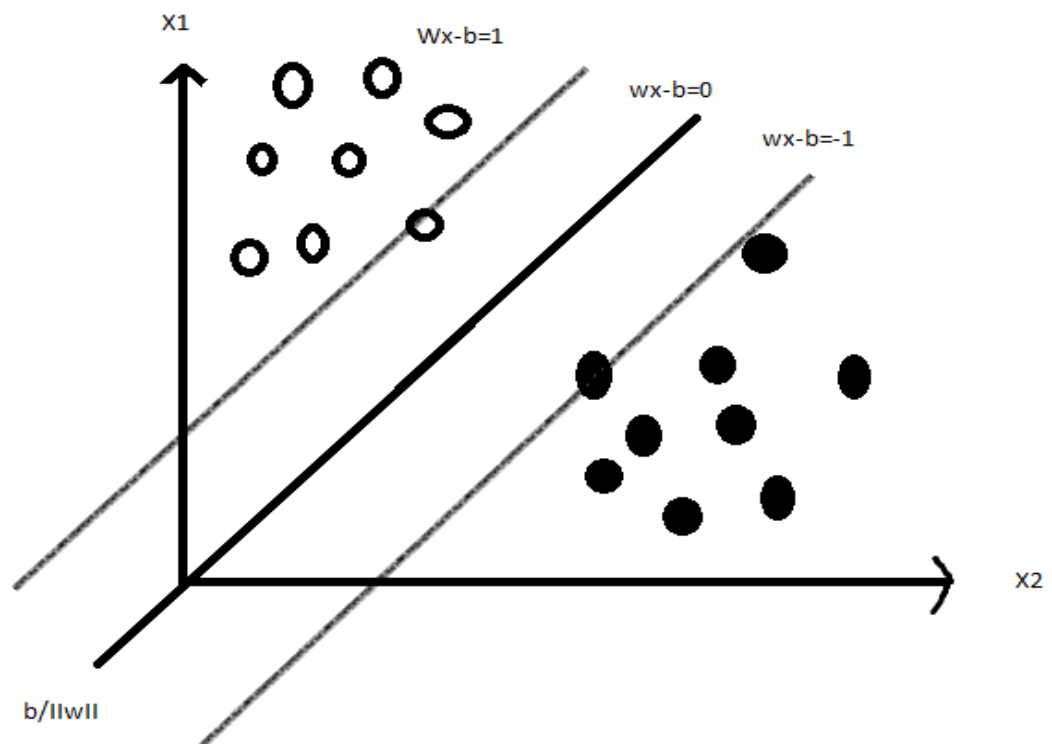


Figure 2.2. Sample from 2 classes on either side of the boundary line flanked by the margins. The samples lying on the margins are called the support vectors.

LINEAR SVM MATHEMATICALLY

Assume that all data is at least distance 1 from the hyperplane, then the following two constraints follow for a training set $\{(\mathbf{x}_i, y_i)\}$

- $\mathbf{w}^T \mathbf{x}_i + b \geq 1$ if $y_i = 1$
- $\mathbf{w}^T \mathbf{x}_i + b \leq -1$ if $y_i = -1$

For support vectors, the inequality becomes an equality, then since each example's distance from the hyperplane is

The margin is:
$$r = y \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|}$$

$$\rho = \frac{2}{\|\mathbf{w}\|}$$

Advantages of SVM:

- Effective in high dimensional spaces.
- Effective even in situations where number of dimensions is larger than the quantity of samples.
- A subset of training points is used in the decision function (known as support vectors), so it is likewise memory effective.
- Versatile: diverse Kernel functions can be determined for the choice capacity. Common kernels are given, however it is likewise conceivable to determine custom kernels.

2.4. MSER (MAXIMALLY STABLE EXTREMAL REGIONS)

MSER is a strategy for blob detection in pictures. The MSER also extricates from a picture various co-variation regions, called MSERs: an MSER is a stable associated part of some grey level sets of the picture.

MSER depends on taking locales which stay almost the same through an extensive variety of thresholds.

- All the pixels underneath a given limit are white and each one of those above or equivalent are black.
- If we are demonstrated an arrangement of thresholded images It with frame t relating to limit t , we would see initially a black image, then white spots comparing to local intensity minima will appear then become bigger.
- These white spots will in the long run converge, until the entire picture is white.

- The set of every single associated segment in the arrangement is the arrangement of all extremal locales i.e. all the connected component set in the image will form the set of all extremal regions.

Alternatively, elliptical frames are connected to the MSERs by fitting ellipses to the areas. Those districts descriptors are kept as key points. The word extremal alludes to the property that all pixels inside the MSER have either higher (bright extremal areas) or lower (dark extremal areas) intensity than every one of the pixels on its external boundary.

This task can be carried out by first sorting all pixels by gray value and after that incrementally adding pixel to each connected component as the threshold is varied. The region is monitored. The areas such that their variation with respect to the threshold is insignificant are characterized maximally stable: –

Making every one of the pixels underneath a certain threshold white, the others are taken black

- Considering an arrangement of thresholded pictures with expanding thresholds moving from black to white we go from a dark picture to pictures where white blobs show up and become bigger by converging, up to the last picture.
- Over a vast scope of limits the nearby binarization is steady and demonstrates some invariance to affine transforms of picture intensities and scaling.



Figure 2.3. MSER processed word image

MSER processing

The MSER extraction is executed with the accompanying strides:

- The threshold intensity is swept from black to white, playing out a straightforward luminance thresholding of the picture
- The connected components are then extracted (Extremal Regions)
- Identifying the threshold for which the extremal region is maximally stable, i.e. local minima of the relative development of its square. Because of the discrete way of the picture, the locale underneath/above might be correspondent with the real region, in which case the area is still considered maximal.
- Approximate an area with a ellipse (this progression is discretionary)
- Keep those areas descriptors as features

In any case, regardless of the possibility that an extremal district is maximally steady, it may be rejected if

- it is too huge (there is a parameter MaxArea)
- it is too little (there is a parameter MinArea)
- it is excessively not stable (MaxVariation)
- it is excessively comparable, to its parent MSER

Extremal locales have the essential properties, that the set is closed under continuous (and in this manner projective) change of picture coordinates, i.e. it is affine invariant and it doesn't make a difference if the picture is distorted or skewed.

Because the locales are characterized only by the intensity function in the area and the external fringe, this prompts numerous key attributes of the regions which make them valuable. Over an extensive scope of thresholds, the local binarization is steady in specific regions, and have the properties mentioned beneath:

- Invariance to relative change of picture intensities
- Stability: just areas whose support is about the same over a range of thresholds is chosen.
- Multi-scale discovery both fine and vast structure is detected.
- The methodology is rather delicate to regular lighting impacts as change of sunlight or moving shadows.

Properties of MSER

- MSER performs well on pictures containing homogeneous areas with distinct boundaries.

- MSER functions admirably for little locales
- MSER doesn't function admirably with pictures with any movement or blur
- Multi-resolution MSER gives better robustness to extensive scale changes and obscured pictures and enhances coordinating execution over huge scale changes and for obscured pictures
- Good repeatability
- Affine invariant
- A shrewd execution makes it one of the speediest locale detectors.

USE IN TEXT DETECTION

The MSER algorithm has been utilized as a part of content detection by consolidating MSER with Canny edges. Canny edges are utilized to adapt to the shortcoming of MSER to obscurity or blurring. MSER is initially connected to the picture being referred to decide the character areas. To augment the MSER regions any pixels outside the limits framed by Canny edges are left out. The partition of the latter given by the edges enormously builds the ease of use of MSER in the extraction of obscured text.

An option utilization of MSER in detection of text is the work by Shi utilizing a graph model. This technique again applies MSER to the picture to produce preliminary locales. These are then used to build a graph model in view of the position separation and colour separation between each MSER, which is taken as a node. Next the nodes are isolated into frontal area and background area utilizing cost capacities. One cost capacity is to relate the separation from the node to the frontal area and foundation. Alternate penalizes nodes for being altogether not the same as its neighbor. At the point when these are minimized the graph is then sliced to isolate the text nodes from the non-text nodes. To empower content discovery in a general scene, Neumann utilizes the MSER algorithm as a part of an assortment of projections. Notwithstanding the greyscale power projection, he utilizes the red, blue, and green shading channels to identify content locales that are color distinct however not as a matter of course particular in greyscale intensity. This technique takes into consideration discovery of more text than exclusively utilizing the MSER+ and MSER-capacities examined previously.

2.4.1. CANNY EDGE DETECTION

Edges portray limits and are along these lines an issue of major significance in image processing. Edges in pictures are regions with solid intensity contrasts – a bounce in intensity starting with one pixel then onto the next. Edge identifying a picture essentially lessens the measure of information and channels out futile data, while saving the vital basic properties in an image.

The Canny edge identification algorithm is referred to numerous as the ideal edge finder. Canny's goals were to upgrade the numerous edge detectors officially out at the time he began his work. In his paper, he took after a rundown of criteria to enhance current strategies for edge identification.

- The first and most evident is low rate of error. It is critical that edges happening in images ought not be missed and that there be no reactions to non-edges.

- The second basis is that the edge points be all around confined. At the end of the day, the distance between the edge pixels as found by the locator and the genuine edge is to be at the very least.

- A third foundation is to have a single reaction to a single edge. This was actualized on the grounds that the initial 2 were not sufficiently enough to totally wipe out the likelihood of various reactions to an edge.

Taking into account these criteria, the canny edge detector first smoothes the picture to dispose of the noise. It then finds the picture gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and smothers any pixel that is not at the most extreme (non greatest concealment). The inclination exhibit is currently further diminished by hysteresis. Hysteresis is utilized to track along the remaining pixels that have not been stifled. Hysteresis utilizes two limits and if the extent is underneath the principal edge, it is set to zero (made a non edge). In the event that the size is over the high threshold, it is made an edge. Also, if the extent is between the 2 limits, then it is set to zero unless there is a way from this pixel to a pixel with a gradient above T_2 .

2.5 EXTRACTION AND REPRESENTATION OF FEATURES

Feature extraction deals with the problem of collecting information from raw inputs. Over the times a number of feature detection techniques have been introduced in image processing and pattern recognition, these represent the document images [25]. Features can either be “global” or “local”. In global features a single vector is used to represent the image, in case of local features a keypoint is selected, and based on the surroundings of this keypoint the feature is designed.

Some common features are: SIFT, Gradient based binary feature (GSC), SURF, profile features, shape context and moments etc. We have defined some of the features in word image retrieval.

2.5.1 Profile Feature

This feature gives a coarse way of representation of word images for matching. The features taken into account for this purpose are upper profile, projections, lower profile and transitions. The projection transition profile works on the ink distribution along one dimension of a given word image. The outline of the word is captured by the two profiling.

2.5.2 Shape Feature

Propose by Belongie et al. [52], the descriptor measures the similarity in shapes using point correspondence recovery between the two shapes being analysed. First, we select a set of points of interest. Then, the point distribution in a plane respective of points of shape are captured. A histogram is used to count the number of interest points inside each of the bin.

2.5.3 Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [42] is one of the most popular descriptor proposed in literature. SIFT is widely used in different computer vision applications like image retrieval [60] and image classification [68] etc. Recently, in document community also many application of SIFT can be found as word image retrieval [51, 69], logo retrieval [31] and page retrieval [61] etc. SIFT, as described in [42], consists of four major stages: (1) scale-space peak selection; (2) keypoint

localization; (3) orientation assignment; (4) keypoint descriptor. In the first stage, potential interest points are identified by scanning the image over location and scale. This is implemented efficiently by constructing a Gaussian pyramid and searching for local peaks (termed keypoints) in a series of difference-of-Gaussian (DoG) images. In the second stage, candidate keypoints are localized to sub-pixel accuracy and eliminated if found to be unstable. The third identifies the dominant orientations for each keypoint based on its local image patch.

The assigned orientation(s), scale and location for each keypoint enables SIFT to construct a canonical view for the keypoint that is invariant to similarity transforms. The final stage builds a local image descriptor for each keypoint, based upon the image gradients in its local neighborhood (discussed below in greater detail). The final (keypoint descriptor) stage of the SIFT algorithm builds a representation for each keypoint based on a patch of pixels in its local neighborhood (see Figure 2.2).

SIFT involves detecting salient locations in an image and extracting descriptors that are distinctive yet invariant to changes in viewpoint and illumination etc.

The SIFT feature computation can be summarized by the following steps:

1. Gradually Gaussian-blur the input-image to construct a Gaussian-pyramid.
2. Construct the Difference of Gaussian (DOG) pyramid by computing the difference of any two consecutive Gaussian-blurred images in the Gaussian pyramid.
3. Find local maximums and local minimums in the DOG space and use the locations and scales of these maximums and minimums as key-point locations in the DOG space.
4. Compute gradients around each key-point (at least a 16×16 region) at the key-point scale and assign an orientation to each key-point based on nearby gradients.
5. Compute histogram of 8-direction Gaussian weighted gradients in 16 sub-blocks (minimum size is 4×4).
6. Concatenate the 16 histograms from 16 sub-blocks to form a 128 dimensional vector as a feat descriptor.

2.5.4 Speeded Up Robust Features

Speeded up robust features (SURF) [15] is proposed as an alternative to SIFT which is of less dimension and can be computed faster than SIFT. SURF descriptor for a given patch is calculated by first equally subdividing a given patch into a 4 grid. For each subsection, the Haar wavelet response D_x and D_y are computed in the x and y directions respectively. SURF descriptor calculates the 4 attributes (SD_{yx} , SD_y , $S|D_x|$, $S|D_y|$) per interest point. In document retrieval, SURF is used in logo retrieval [31] and sign board detection [56] etc.

SURF utilizes integral images that are intermediate presentation for the image and comprises the addition of gray scale pixel values of the image, reducing computation time. The detector has been based on Hessian matrix since, it has good performance where accuracy and computation time is concerned.

The Haar wavelet responses are used by the descriptor where the response lie within the interest points neighborhood. The working of SURF descriptor can be described as follows first we have to identify an orientation that is reproducible this can be derived from the information around the point of interest radius. Next, a square region is built which will be aligned along the selected orientation and SURF descriptors are extracted.

i) Assignment of orientation: we first calculate the response of the Haar-wavelet along the x and y axis in a neighborhood of 6s radius around interest point. These responses are shown as vectors. We will sum all the responses that lie at an angle of 60 degree within a sliding orientation window. We sum up both the horizontal as well as vertical responses yielding a new vector response. The longest of which is the dominant vector.

ii) Description: in this step we construct a square area that will be centered around the interest points and oriented accordingly. We then split the interest point in a 4x4 sq sub region with 5x5 regularly spaced sample points. Haar wavelet response in the horizontal and vertical directions are determined. They are then weighed using a Gaussian Kernel centered at the keypoints.

2.6 BAG OF WORDS

Recovering documents in text domain has been thoroughly studied over a period of time. The Bag of Word (BoW) approach has been one of the earliest and mainstream techniques to seek in text domain, see Figure 3.1. BoW approach is an instrument for representing text documents. BoW is by and large isolated into followings: pre-processing of text, Stemming and Vocabulary Selection.

During the pre-processing step, the document is divided into a series of words by eliminating all accentuation marks and by supplanting tabs and other non-content characters by single white spaces. This tokenized representation is then utilized for further steps. A dictionary or vocabulary is created by using the numerous words derived from all the text documents in a given archive. Once we have the vocabulary we resize it by removing unwanted words which pose no utility to the vocabulary, such words could be the prepositions, articles, conjunctions etc. removal of these is going to have no relevance on the content or document representation. Next we remove words whose frequency of occurrence is very low and hence could bear no significant role in the representation of documents. removal of s from plurals like apple(s), of prefixes such as ing like fly(ing) and other affixes and prefixes and assigning the same stem to similar words such as plays, playing are assigned the common stem word play. A stem can be defined as the root of words originating from a single unit and having similar or near similar meanings. Hence all the words are represented by their stems.

Indexing helps in further lessening the number of words, many algorithms have been designed for the same purpose. For this situation, just the chosen keywords are utilized to portray the archives. A straightforward strategy for keyword determination is to extricate keywords taking into account their entropy. Entropy gives a measure how well a word is suited to documents by keyword search. As vocabulary words a number of words that have high entropy in respect to their general recurrence can be picked, i.e. of words happening similarly frequently those with the higher entropy can be favored. Keeping in mind the end goal to acquire a fixed number of vocabulary terms that properly cover the records, a straightforward insatiable technique is connected: From the first archive in the accumulation select the term with the most astounding relative entropy as a vocabulary.

The arrangement of various words got by combining all content reports of a gathering is known as the word reference or vocabulary of a record accumulation. After getting vocabulary, vocabulary size is lessened by filtering and stemming. Introductory filtering is done in term of evacuating stop words like articles, conjunctions, prepositions, and so on. Expelling stop words bear little on the other hand no substance data on report representation. Besides, words that happen to a great degree frequently can be said to be of little data substance to recognize reports, furthermore words that happen sometimes are liable to be of no specific factual significance and can be expelled from the vocabulary. Next is to attempt to guide verb structures to the infinite strained and things to the solitary form. In Stemming, attempt to construct the essential types of words, i.e. strip the plural s from things, the ing from verbs, or different affixes , for instance 'read', "perusing" and so on is doled out same stem 'read'. A stem is a characteristic gathering of words with equivalent (or undamentally the same as) which means. After the stemming procedure, each word is spoken to by its stem.

To further lessen the quantity of words that ought to be utilized additionally indexing or feature choice algorithms can be utilized. For this situation, just the chosen feature are utilized to depict the documents. A straightforward technique for keyword determination is to remove keywords in view of their entropy. Entropy gives a measure how well a word is suited to separate archives by keyword appearance. As vocabulary words a number of words that have a high entropy in respect to their general recurrence can be picked, i.e. of words happening similarly frequently those with the higher entropy can be favored.

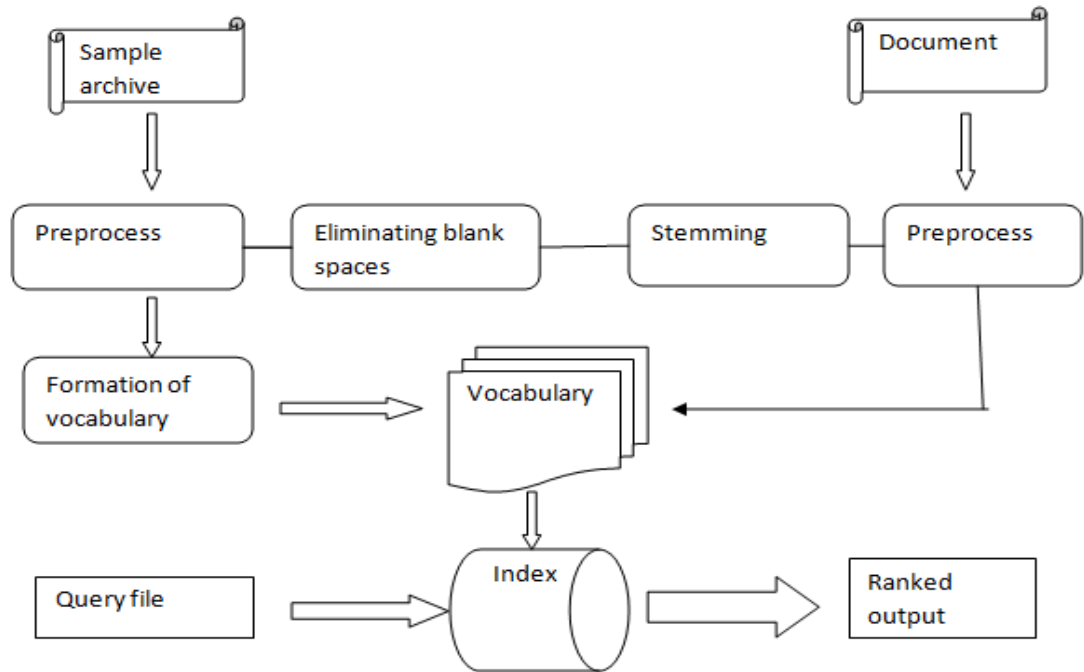


Figure2.4 Bag of Words for document image retrieval

With a specific end goal to get a fixed number of vocabulary terms that properly cover the documents, a straightforward ravenous technique is used: From the first document in the accumulation select the term with the most elevated relative entropy as a vocabulary.

At that point mark this record and all different records containing this term. From the first of the remaining unmarked reports select again the term with the most noteworthy relative entropy as a list term. At that point check again this report and all different records containing this term. This procedure is rehashed until all documents are checked, then unmark all of them and begin once more. The procedure is ended when the sought number of vocabulary term has been chosen. Despite the fact that the vocabulary is a set, a discretionary requesting fix for it so we can allude to word 1 through word V where V is the extent of the vocabulary.

On fixing the vocabulary we can represent each of the documents as a vector of length V. let the vector be X then jth component of X signifies how many times the word j has been spotted in the document.

Next, we will index these records by the use of the vocabulary created. Inverted file index is used for this purpose; it is a very common and popular method for indexing these vectors. The inverted file index comprises section for all words and every word is connected with all the archives where the word is available and additionally its occurrence is stored. While is recovery stage, for a given inquiry relating word in examined in index and all archives which are having given query words is recovered.

2.7 BAG OF VISUAL WORDS

The bag of visual words technique has evolved from the bag of words technique where we move from document classification in BOWs to image classification in BOVWs. The Bag of Words technique is used for document classification where the documents are represented based on the frequency of word occurrence. The document is considered as a stream of words where punctuation marks and blank spaces are ignored. A vocabulary is created and its size reduced by removing words that occur too often (as they might not hold much significance) or words that occur too less say once or twice. After filtering the vocabulary to the desired size we proceed further by indexing the documents. The desired document is retrieved from the indexed vocabulary using their histogram (frequency of word occurrence) by comparing the query image with the indexed files. Similar to the BoWs we use the Bag of Visual Words (BoVWs) technique for image classification. Here instead of text words we use visual words to build our vocabulary. An image of an inscription can be represented as a set of non distinctive discrete visual features, where the visual features corresponds to the vocabulary, and the image is defined as a histogram of visual word occurrence.

The BoVW model is roused by the achievement of utilizing BoW as a part of text classification and recovery. In BoW display, every record is spoken to by an unordered arrangement of non-distinct words present in the archive, paying little heed to the sentence structure and word order. Archive is formally spoken to with the assistance of recurrence of (histogram) the words in the vocabulary. These histograms are then used to perform archive classification and recovery. Similarly, a picture is presented by an unordered set of non-distinct discrete visual components. The

arrangement of these discrete visual elements is called vocabulary. On account of a document image, one can think about the glyphs as the vocabulary and a word can be defined as a set of these glyphs. By taking an image as a histogram of visual words, one can acquire certain level of invariance to the spatial area of object in the image. In any case, this makes certain issues in document image representation.. For instance, "PIN" and "NIP" are same for this representation because of the absence of order in the representation. This lessens the accuracy in a recovery. We address this issue, while exploiting the computational points of interest of the BoVW representation as clarified in the following area.

We set off by defining the interest points (either corners or blobs) and then determining the features of those interest points. We have used SURF (speeded up robust features) as our key feature detector. SURF uses an integer approximation of the determinant of Hessian blob detector to find its interest points. After extracting the features we segment them using K mean clustering. This is done to get distinct clusters and then eventually to derive a code book where each cluster is represented by its centroid.

The k mean clustering is an unsupervised learning algorithm which forms k clusters from M observations. It is a cyclic process where the cluster centroids keep on rearranging their location until they reach a point beyond which they remain in a fixed positions and the clustering process ends. First k centroids are defined and each point of the data set is grouped into clusters based on the nearest centroids. Once this is done we recalculate k new centroids as barycentre of previous clusters and the points are regrouped according to the new centroids. This process is repeated to get k fixed centroids. And these are then used in the codebook generation process. Once the clustering is done the extracted features are given labels of the closest centroid and thus the SURF features are quantized. Figure1. Depicts how from a database of inscription images each image is taken and its SURF extracted followed by generation of a codebook and histogram computation of the quantized SURF features.

SURF was first introduced by Herbert Bay et al. [12] as a feature detector and descriptor that is both scale and rotation invariant. The SURF algorithm makes use of the Hessian matrix which acts by defining interest points. The Hessian matrix uses the Difference of Gaussian (DoG) which is a basic laplacian based descriptor. The SURF works in four steps, first the Hessian matrix is used to determine interest points, then

the major interest point in scale space are found, next feature direction is taken into account as it should be rotation invariant and finally the feature vectors are generated. Haar transforms are used to assess the direction of the features.

CHAPTER 3

PROPOSED METHOD

OVERVIEW

In this section we shall discuss the proposed methodology. The purpose and need for image retrieval has been explained in previous chapters. We have employed two techniques for retrieval of image, these are- the Bag of Visual Words (BoVWs) and an amalgamation Of Support Vector Machines (SVM) and Maximally Stable Extremal Regions (MSER).

The BoVWs technique is a word image retrieval technique that can be employed to any image for retrieval purpose. We have focussed on retrieval of inscription images. Inscriptions are a form of gateway that allow us into a world that has been lost for many years, the early cavemen used to inscribe on stones and inside their caves, many civilizations that followed have used inscription as a medium of passing their information from one generation to the next. Hence, we can say it is very important that we safeguard these inscriptions. The best way to do so is to create a digital library where these inscriptions can be safeguarded in the form of images. Work on inscription images has not been explored much and so have defined an efficient, robust and fast inscription image retrieval system.

3.1 Inscription Image Retrieval

As mentioned above, the inscription image retrieval has been done using the Bag of Visual Words technique. The BoVW is similar to BOW, where we have a document image retrieval.

3.1.1 Bag of Visual Words

The bag of visual words technique has evolved from the bag of words technique where we move from document classification in BOWs to image classification in BOVWs. The Bag of Words technique is used for document classification where the documents are represented based on the frequency of word

occurrence. The document is considered as a stream of words where punctuation marks and blank spaces are ignored.

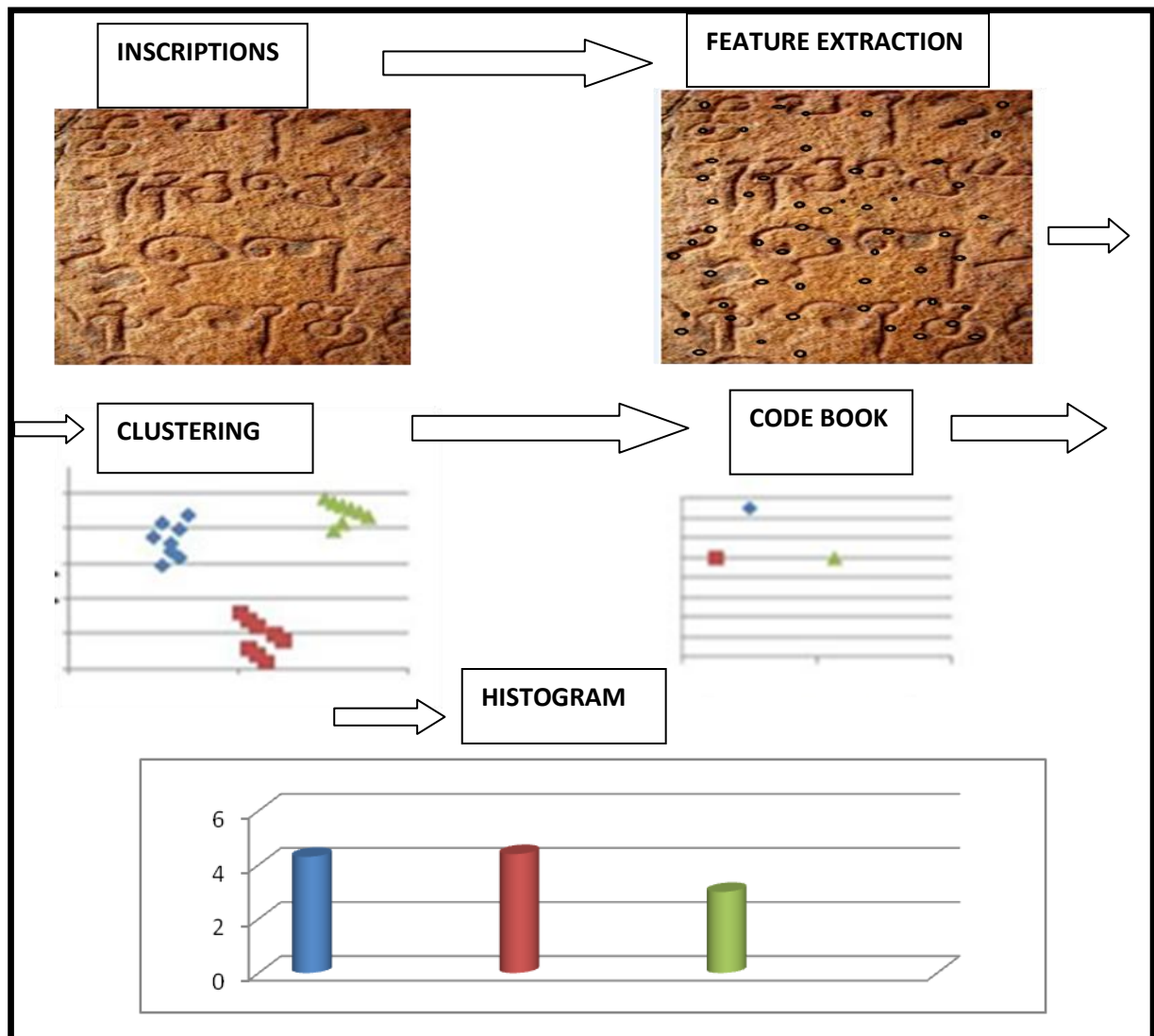


Figure3.1.The Bag of Visual Words feature extraction, code book generation and histogram formation.

A vocabulary is created and its size reduced by removing words that occur too often (as they might not hold much significance) or words that occur too less say once or twice. After filtering the vocabulary to the desired size we proceed further by indexing the documents. The desired document is retrieved from the indexed vocabulary using their histogram (frequency of word occurrence) by comparing the query image with the indexed files. Similar to the BoWs we use the Bag of Visual Words (BoVWs) technique for image classification. Here instead of text words we

use visual words to build our vocabulary. An image of an inscription can be represented as a set of non distinctive discrete visual features, where the visual features corresponds to the vocabulary, and the image is defined as a histogram of visual word occurrence.

We set off by defining the interest points (either corners or blobs) and then determining the features of those interest points. We have used SURF (speeded up robust features) as our key feature detector. SURF uses an integer approximation of the determinant of Hessian blob detector to find its interest points. After extracting the features we segment them using K mean clustering. This is done to get distinct clusters and then eventually to derive a code book where each cluster is represented by its centroid.

3.1.2 K Mean Clustering

The k mean clustering is an unsupervised learning algorithm which forms k clusters from M observations. It is a cyclic process where the cluster centroids keep on rearranging their location until they reach a point beyond which they remain in a fixed positions and the clustering process ends. First k centroids are defined and each point of the data set is grouped into clusters based on the nearest centroids. Once this is done we recalculate k new centroids as barycentre of previous clusters and the points are regrouped according to the new centroids. This process is repeated to get k fixed centroids. And these are then used in the codebook generation process.

Once the clustering is done the extracted features are given labels of the closest centroid and thus the SURF features are quantized. Figure4.1. Depicts how from a database of inscription images each image is taken and its SURF extracted followed by generation of a codebook and histogram computation of the quantized SURF features.

3.1.3. SURF (speeded up robust features)

SURF was first introduced by Herbert Bay et al. [12] as a feature detector and descriptor that is both scale and rotation invariant. The SURF algorithm makes use of the Hessian matrix which acts by defining interest points. The Hessian matrix uses the Difference of Gaussian (DoG) which is a basic laplacian based descriptor. The SURF works in four steps, first the Hessian matrix is used to determine interest points, then the major interest point in scale space are found, next feature direction is taken into

account as it should be rotation invariant and finally the feature vectors are generated. Haar transforms are used to assess the direction of the features.

The length of the descriptor vector is of 64 floating point numbers however we can extend it to 128. SURF has been made available in the form of a precompiled library in which the core of the algorithm is a closed source.

SURF utilizes integral images that are intermediate presentation for the image and comprises the addition of gray scale pixel values of the image, reducing computation time. The detector has been based on Hessian matrix since, it has good performance where accuracy and computation time is concerned.

The Haar wavelet responses are used by the descriptor where the response lie within the interest points neighborhood. The working of SURF descriptor can be described as follows first we have to identify an orientation that is reproducible this can be derived from the information around the point of interest radius. Next, a square region is built which will be aligned along the selected orientation and SURF descriptors are extracted.

iii) Assignment of orientation: we first calculate the response of the Haar-wavelet along the x and y axis in a neighborhood of 6s radius around interest point. These responses are shown as vectors. We will sum all the responses that lie at an angle of 60 degree within a sliding orientation window. We sum up both the horizontal as well as vertical responses yielding a new vector response. The longest of which is the dominant vector.

iv) Description: in this step we construct a square area that will be centered around the interest points and oriented accordingly. We then split the interest point in a 4x4 sq sub region with 5x5 regularly spaced sample points. Haar wavelet response in the horizontal and vertical directions are determined. They are then weighed using a Gaussian Kernel centered at the keypoints.

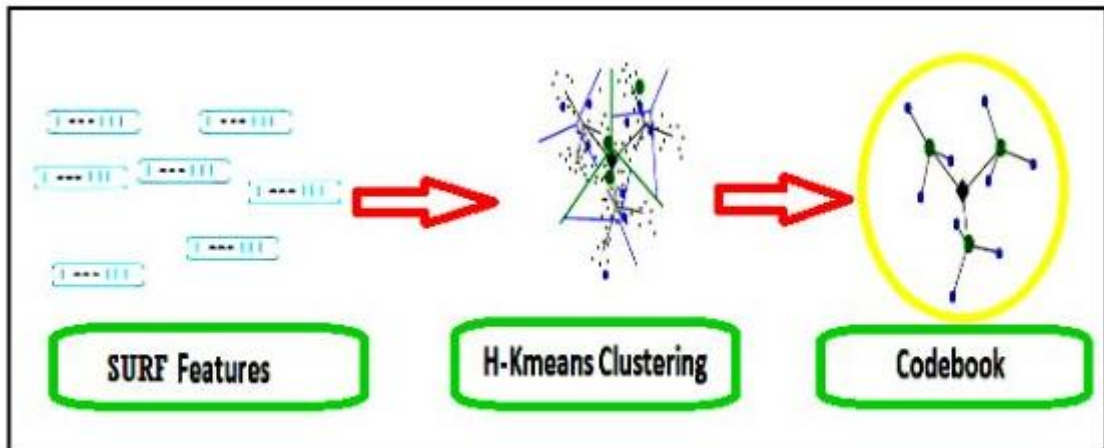


Figure3.2. Generation of code book.

3.1.4 The Retrieval Process

Once we have created the codebook and have a vocabulary the next step is the retrieval of images. For the retrieval purpose we first index the codebook using the inverted file indexing. Inverted file indexing is introduced as it has proven to enhance the systems efficiency by several orders. The images can be presented as a catalogue of characteristics or attributes. The indexed file then behaves as a sorted list of these attributes and each attribute has a link to the image comprising it. Hence whenever an attribute is recalled all the images from the catalogue comprising of it will be retrieved. The retrieval process as depicted in figure4.3 starts with inputting the query image. From our entire collection of images if we want to retrieve an inscription image we need to input a query image which will be used to compare and extract the correct output image. The initial processing on the query image is similar to those on the inscription image from the database. First the SURF features of the query are extracted. Once this is done we cluster them using the k means algorithm. From the vocabulary the closest cluster to the derived cluster of the query image is located and assigned to it.

This is represented in the codebook and from the indexed files all the images that are assigned to the given cluster are brought forward of which the best match is retrieved.

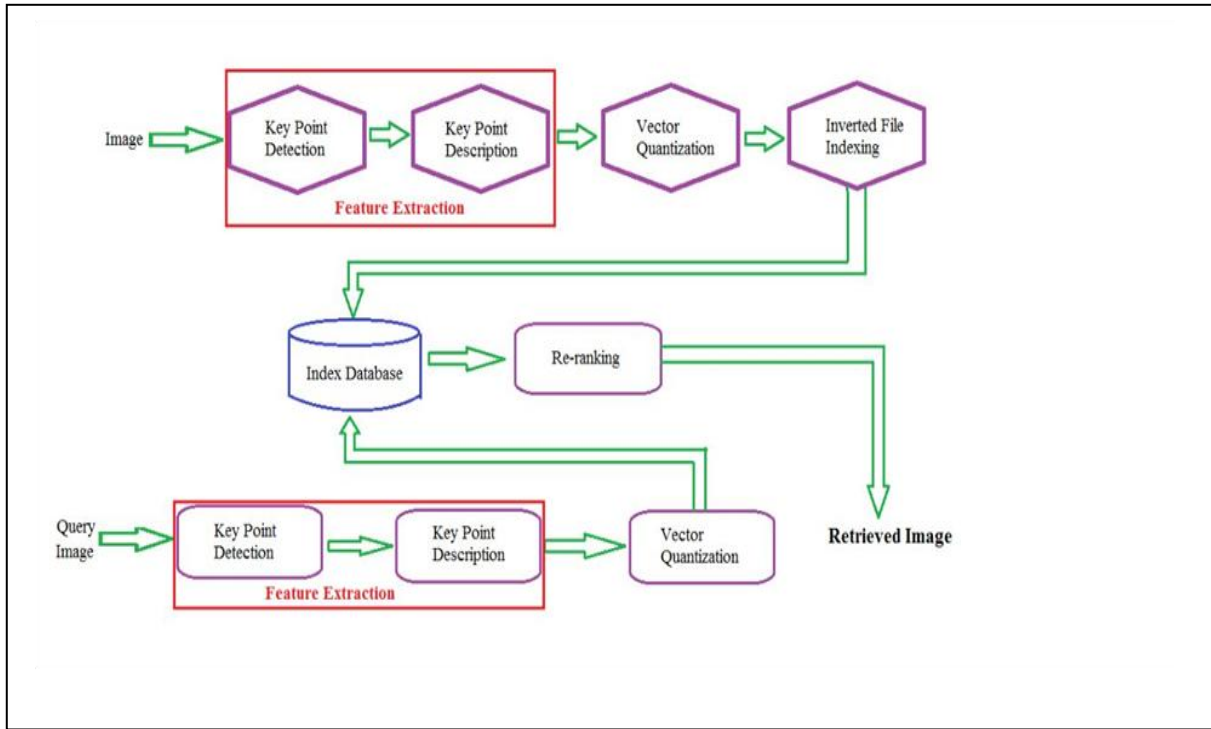


Figure 3.3 Retrieval System

The figure 4.3 is divided into two sections indexing and retrieval. Indexing follows three basic steps- we extract the features from the images, we perform vector quantization for histogram generation, and last is the creation of database for which word images are indexed using inverted file indexing.

The initial two steps of indexing are similar in retrieval process as well. The histogram is provided to the index structure and we get the output in a ranked form.

A pre-computed vocabulary is used for the computation of the histogram. In order to create a vocabulary we extract the features from a database of images. Clustering follows next on the extracted features of these images. Collection of the centroids from the clustering comprises the vocabulary.

CHAPTER 4

RESULT AND CONCLUSION

In this section we shall discuss the two methods that have been used for the retrieval of an image. We shall start with the SVM and MSER based technique that was used for retrieval of text in an image, then we will see the limitations faced in this method and move next to the second technique of BoVWs for image retrieval. We shall show how this technique is superior to the former.

4.1 SVM and MSER Based Image Retrieval

We have trained the SVM to recognise two classes, the first class is image with text and the other is image without any text in them. The SVM was trained using a dataset of 120 images each. The image below was taken up as small windows and text detection done in each window, retrieving those images that detect text in them.

Original Image :



Figure 4.1 image used for text detection using SVM.

Results of text detection after applying SVM : According to SVM, the below sections (sub images) of the original image contains text.



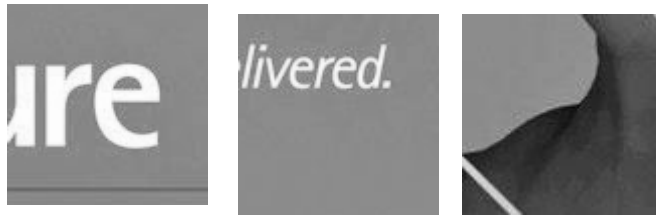


Figure 4.2 output images of SVM

From the above we can see that SVM detects text but also there are a lot of frames that do not have text in them. In order to improve the retrieved result we apply MSER to the received output images.

Results after applying MSER : Each individual output image subsection above from SVM is passed on to MSER for text recognition. MSER detects individual characters also in images classified by SVM as containing text correctly, and filters out the incorrect classification results

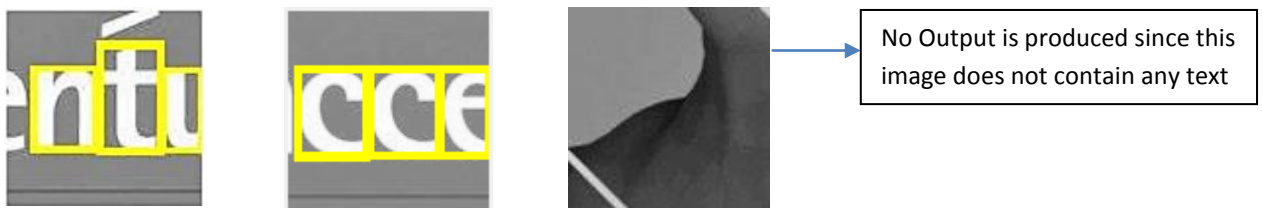


Figure 4.3 Result after applying SVM and MSER to the image.

Figure 4.4 shows the final output obtained using this technique.

4.1.1 CONCLUSION

In this work, a novel text detection algorithm is proposed, which employs Maximally Stable Extremal regions as basic letter candidates. To overcome the sensitivity of MSER with respect to image blur and to detect even very small letters, we developed an edge-enhanced MSER which exploits the complementary properties of MSER and Canny edges. Further, we present a novel image operator to accurately determine the stroke width of binary CCs. Our proposed method has demonstrated state-of-the-art performance for localizing text in natural images. The detected text are binarized letter patches, which can be directly used for text recognition purposes.

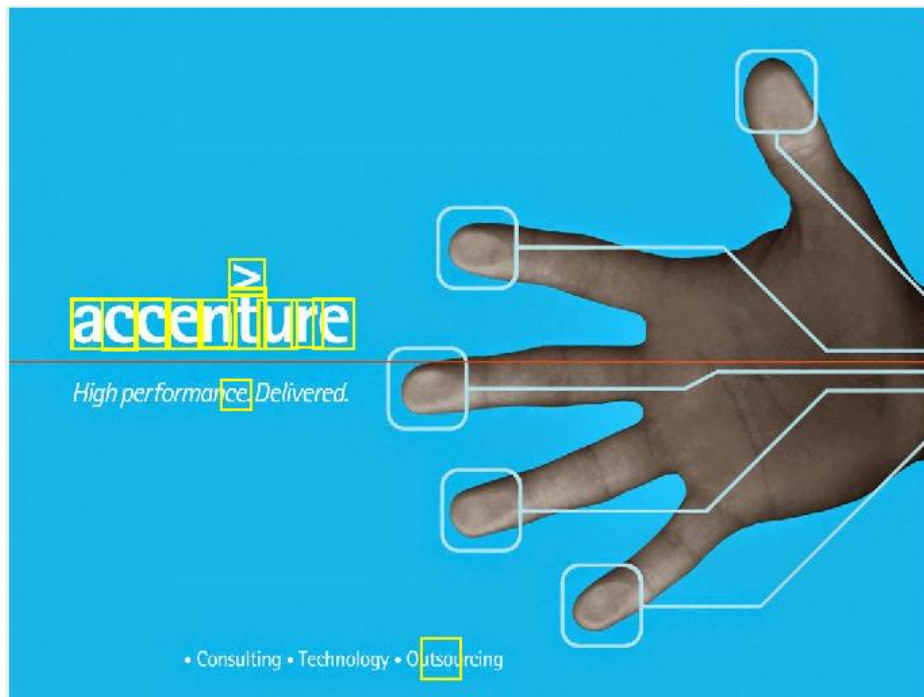


Figure 4.4 text retrieval in an image using SVM and MSER.

4.1.2 Drawback

The above technique though produces good results but we face a number of drawbacks which are as follows:

- i. The method though successful in recognising most of the text still has the possibility of missing out on text written in smaller fonts.
- ii. It is not language independent. We need to train SVM for different languages and this loses the practicality.
- iii. It is not scale invariant
- iv. It can be susceptible to noise.

After going through all these drawbacks we worked towards a method that could overcome all these shortcomings and hence, we proposed the Bag of Visual Words.

4.2 Bag of Visual Words

In this section we shall discuss the utility and scalability of the proposed system. The proposed method of indexing and retrieval has been tested on a collection of inscription images. These images have been selected from a series of inscribed languages, English has been used for common understanding, apart from that Tamil and symbolic representations have also been used. We have measured the quantitative performance using the precision which is given by

$$P = \frac{\text{true_positive}}{\text{true_positive} + \text{false_positive}} \quad (1)$$

Where true_positive is a hit and false_positive is a miss. Figure 4.5 demonstrates some of the inscription images used in the retrieval process.



Figure 4.5 Inscription images

PERFORMANCE STATISTICS

TABLE 4.1 PRECISION PERFORMANCE

LANGUAGE	NO. OF IMAGES	QUERY IMAGES	PRECISION
English	55	10	0.772
Tamil	73	25	0.78
Others	212	42	0.763

Table 4.1 gives the performance statistics of the system, we have created a database of around 300 inscription images out of which 72 are in English, 56 in Tamil and the remaining in other languages. We get an average precision of 0.755. As mentioned in the previous section we have used a number of query images giving the corresponding outputs.

QUERY	OUTPUT	QUERY	OUTPUT

Figure 4.6 query image with corresponding output

Figure 4.6 depicts several query images that were given to the system, there SURF features were extracted and clustered and matched to with the codebook based on which the output is derived.

The system also works better for degraded images, for example in case of a blurred image we are able to retrieve the output to a large extent of blurring in both the query image and the original image in the database. This has been depicted in figure 4.7 and 4.8 below.

Query image	Original image	Blurred output retrieved
		
		

Figure 4.7 blurred image retrieval

SURF is also a scale invariant detector therefore we can get an output even when the query is out of phase with the original image. Figure 4.9 depicts the scale invariance property of the system.


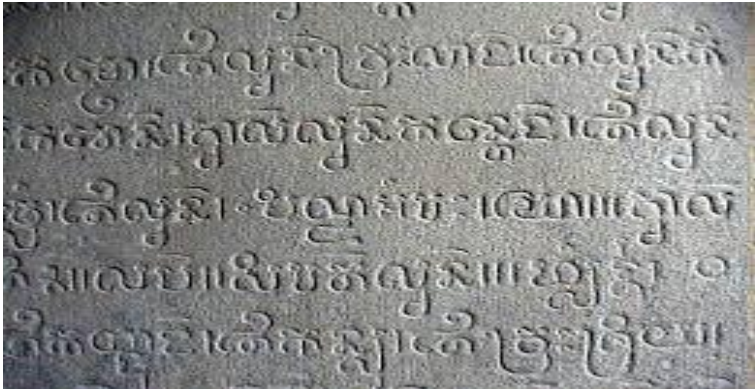
Blurred query	Output
	

Figure 4.8 Blurred query retrieving original image.



Figure 4.9 scale invariant retrieval.

		
Query image	Original image	Darkened image

Figure 4.10 darkened image being retrieved as output.

Figure 4.10 depicts how the method can work with noise affected images, such as a darkened image.

4.2.1 CONCLUSION

The propose method is an image retrieval system that is based on Bag of Visual Words. It is highly language independent and scalable, which was an issue with the method used previously. We have shown the efficiency of the proposed method to be high and accurate. English, Tamil, symbolic languages and other inscription images have been used to train and test the proposed method.

CHAPTER 5

CONCLUSION AND FUTURE WORK

As the world is becoming digitized all the data and information are being stored online or in digitized libraries, in order to have an easy access to those millions and millions of images, documents and archives we need to have an efficient retrieval system that is not only accurate but also very fast. We have therefore proposed in this report a bag of visual words based image retrieval system and worked on inscription images as our data set.

Inscription images were chosen as work in that field is very limited and they pose certain challenges which any other data set could not, these are as follows:

- i. Inscriptions not only comprise of words or text but can also be present in the form of pictorial representation.
- ii. Inscriptions over the period of time have undergone degradation.
- iii. Noise such as blur or darkening is very likely to be present while dealing with inscription images.

Therefore we needed a system that could overcome all these challenges and give us a retrieval system that is not only language independent but also scale invariant and has tolerance to noise.

Initially a method using SVM and MSER was proposed that could detect the text in the image and retrieve it for us, however it proved to have shortcomings and could not fulfil our list of objectives.

Next an approach based on vocabulary creation was suggested that works by extracting the features of a word image and creating a dictionary of those features known as the codebook. The codebook is then used to get the desired output by matching the features of the query image with the codebook. This technique was called the Bag of Visual Words.

SURF was the key feature extractor and it is SURF that enabled us to overcome the drawbacks of the previous methods. SURF is a scale invariant key

feature detector that works well for noisy images suffering from degradation in the form of darkening or blurring.

The performance of the method was tested on a database of 300 images, where English, Tamil and other sample inscriptions (55, 70, 212 respectively) were used.

Performance was measured based on precision of the output. The average precision coming out to be 7.77 was shown. Results on blurred, darkened and rotated images were also shown in the results section.

The proposed method has been tested on 300 images, we can extend it to a larger dataset for further work. We can explore similar methods for handwritten and camera based documents, the speed of execution can be improved, other feature detectors can be explored in place of SURF.

REFERENCES

1. S.Rajakumar Dr.V.Subbiah Bharathi, “ Century Identification and Recognition of Ancient Tamil Character Recognition “ Research scholar Sathyabama University Department of ECE Chennai, India., India International Journal of Computer Applications (0975 – 8887) Volume 26– No.4, July 2011.
2. Jose A Rodriguez Serrano, “Handwritten word-spotting using hidden Markov models and universal vocabularies” September 2009, Pages 2106–2116 elsevier.
3. M.Brown L.Rabiner, “Dynamic time warping for isolated word recognition based on ordered graph searching techniques” Acoustics, speech and signal processing, IEEE international conference on ICASSP’82(Volume : 7).
4. K.pramod Sankar, R. manmatha, C.V Jawahar, “Large scale document image retrieval by automatic word annotation” International Journal on Document Analysis and Recognition archive springer-verlag Berlin, Heildelberg volume 17 Issue 1, March 2014.
5. Konstantinos Zagoris Ioannis PratikakisBasilis Gatos,” Segmentation-based Historical Handwritten Word Spotting using Document-Specific Local Features “14th International Conference on Frontiers in Handwriting Recognition 2014.
6. Olivier Augereau, Nicholas Journet, Anne Vialard, Jean-Philippe Domenger,” Improving classification of an industrial document image database by combining visual and textual features”.
7. Rusi~nol, M., Aldavert, D., Toledo, R., Llad_os, J,” Browsing heterogeneous document collections by a segmentation-free word spotting method” In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 63{67 (2011).
8. Ataer,E., Duygulu, P,” Matching ottoman words: an image retrieval approach to historical document indexing”. In: Proceedings of the International Conference on Image and Video Retrieval, pp. 341{347 (2007).
9. Sankar, P., Jawahar, C., Manmatha, R, “ Nearest neighbour based collection ocr” In: Proceedings of the IAPR Workshop on Document Analysis Systems, pp. 207{214 (2010).
10. Rothacker, L., Rusi~nol, M., Fink, G,” Bag-of-features hmms for segmentation-free word spotting in handwritten documents”In: Proceedings of the International Conference on Document Analysis and Recognition, pp. 1305{1309 (2013).

11. David Aldavert · Marçal Rusiñol · Ricardo Toledo · Josep Lladós “A Study of Bag-of-Visual-Words Representations for Handwritten Keyword Spotting”.
12. Herbert Bay, Tinne Tuytelaars, Luc Van Gool, "Speeded-Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp.346-359, EECV, 2008.
13. Ravi Shekhar and C.V. Jawahar “Word Image Retrieval using Bag of Visual Words” 2012 10th IAPR International Workshop on Document Analysis Systems
14. V. Bansal and R. M. K. Sinha. A complete OCR for printed hindi text in devanagari script. In *International Conference on Document Analysis and Recognition*, 2001.
15. H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.48
16. A. Bhaskarhatla, S. Madhavanath, M. P. Kumar, A. Balasubramanian, and C. V. Jawahar. Representation and annotation of online handwritten data. In *International Workshop on Frontiers in Handwriting Recognition*, 2004.
17. S. H. K. C. Y. Suen, S. Mori and C. H. Leung. Analysis and recognition of asian scripts - the state of the art. In *International Conference on Document Analysis and Recognition*, 2003.
18. J. Chan, C. Ziftci, and D. A. Forsyth. Searching off-line arabic documents. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1455–1462, 2006.
19. B. B. Chaudhuri and U. Pal. An OCR system to read two indian language scripts: Bangla and devanagari (hindi). In *International Conference on Document Analysis and Recognition*, 1997.
20. B. B. Chaudhuri and U. Pal. A complete printed bangla ocr system. *Pattern Recognition*, 31(5):1997,531-549.
21. S. Chaudhury, G. Sethi, A. Vyas, and G. Harit. Devising interactive access techniques for indian language document images. In *International Conference on Document Analysis and Recognition*, pages 885–889, 2003.
22. O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE Conference on Computer Vision*, pages 1–8, 2007.
23. J. Cowell and F. Hussain. Amharic character recognition using a fast signature based algorithm. In *International Conference on Information Visualization*, 2003.
24. S. Deorowicz. Solving longest common subsequence and related problems on graphical processing units. *Software - Practice and Experience*, 40(8):673–700, 2010.

25. R. O. Duda, P. Hart, and D. Sttork. *Patter Classification*. JohnWiley & Sons, 2001.
26. A. Dutta and S. Chaudhury. Bengali alpha-numeric character-recognition using curvature features. *Pattern Recognition*, 26(12):1757–1770, 1993.
27. J. Edwards, Y. W. Teh, D. A. Forsyth, R. Bock, M. Maire, and G. Vesom. Making latin manuscripts searchable using ghmmms. In *Neural Information Processing Systems*, 2004.
28. R. T. Hartley and K. Crumpton. Quality of ocr for degraded text images. In *ACM Conference on Digital libraries*, 1999.
29. Y. Ishitani. Model-based information extraction method tolerant of OCR errors for document images. In *International Conference on Document Analysis and Recognition*, 2001.
30. A. K. Jain and A.M. Namboodiri. Indexing and retrieval of on-line handwritten documents. In *Internationa Conference on Document Analysis and Recognition*, pages 655–659, 2003.
31. R. Jain and D. S. Doermann. Logo retrieval in document images. In *IAPR International Workshop on Document Analysis Systems*, pages 135–139, 2012.
32. V. Jawahar and A. Kumar. Content-level annotation of large collection of printed document images. In *International Conference on Document Analysis and Recognition*, pages 799–803, 2007.49
33. C. V. Jawahar, M. N. S. S. K. P. Kumar, and S. S. R. Kiran. A bilingual ocr for hindi-telugu documents and ts applications. In *International Conference on Document Analysis and Recognition*, 2003.
34. C. V. Jawahar, M. Meshesha, and A. Balasubramanian. Searching in document images. In *Indian Conference on Vision, Graphics and Image Processing*, pages 622–627, 2004.
35. K. Jithesh, K. G. Sulochana, and R. R. Kumar. Optical character recognition (OCR) system for malayalam language. In *National Workshop on Application of Language Technology in Indian Languages*, 2003.
36. A. L. Kesidis and B. Gatos. Efficient cut-off threshold estimation for word spotting applications. In *International Conference on Document Analysis and Recognition*, pages 279–283, 2011.
37. V. Kluzner, A. Tzadok, D. Chevion, and E. Walach. Hybrid approach to adaptive ocr for historical books. In *International Conference on Document Analysis and Recognition*, pages 900–904, 2011.

38. T. Konidakis, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *International Journal on Document Analysis and Recognition*, 9(2-4):167–177, 2007.
39. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.
40. D. S. Lee and R. Smith. Improving book OCR by adaptive language and image models. In *IAPR International Workshop on Document Analysis Systems*, pages 115–119, 2012.
41. X. Lin. DRR research beyond COTS OCR software: A survey. In *In SPIE Conference on Document Recognition and Retrieval XII*, 2005.
42. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
43. K. Marukawa, T. Hu, H. Fujisawa, , and Y. Shima. Document retrieval tolerating character recognition errorsevaluation and application. *Pattern Recognition*, 30(7):1997, 1361-1371.
44. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, 2002.
45. D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.
46. U. Pal and B. B. Chaudhuri. Indian script character recognition: A survey. *Pattern Recognition*, 37(9):1887–1899, 2004.
47. T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *International Conference on Document Analysis and Recognition*, pages 218–222, 2003.
48. T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 521–527, 2003.
49. T. M. Rath and R. Manmatha. Word spotting for historical documents. *International Journal on Document Analysis and Recognition*, pages 139–152, 2007.50
50. P. P. Roy, J.-Y. Ramel, and N. Ragot. Word retrieval in historical document using character-primitives. In *International Conference on Document Analysis and Recognition*, pages 678–682, 2011.

51. M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *International Conference on Document Analysis and Recognition*, pages 63–67, 2011.
52. J. M. S. Belongie and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 509–522, 2002.
53. R. Saabni and A. Bronstein. Fast key-word searching via embedding and active-dtw. In *International Conference on Document Analysis and Recognition*, pages 68–72, 2011.
54. K. P. Sankar, V. Ambati, L. Pratha, and C. V. Jawahar. Digitizing a million books: Challenges for document analysis. In *IAPR International Workshop on Document Analysis Systems*, pages 425–436, 2006.
55. K. P. Sankar, C. V. Jawahar, and R. Manmatha. Nearest neighbor based collection ocr. In *IAPR International Workshop on Document Analysis Systems*, pages 207–214, 2010.
56. G. Schroth, S. Hilsenbeck, R. Huitl, F. Schweiger, and E. G. Steinbach. Exploiting text-related features for content-based image retrieval. In *ISM*, pages 77–84, 2011.
57. S. Setlur and V. Govindaraju, editors. *Guide to OCR for Indic Scripts*. 2009.
58. S. Setlur and V. Govindaraju (editors). Guide to OCR for indic scripts. In *Springer*, 2009.
59. R. Shekhar and C. V. Jawahar. Word image retrieval using bag of visual words. In *IAPR International Workshop on Document Analysis Systems*, pages 297–301, 2012.
60. J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *IEEE Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.
61. D. Smith and R. Harvey. Document retrieval using SIFT image features. *Journal of Universal Computer Science*, 17(1):3–15, 2011.
62. M. Sridha, D. Mandalapu, and M. Patel. Active-DTW: A generative classifier that combines elastic matching with active shape modeling for online handwritten character recognition. In *International Conference on Frontiers in Handwriting Recognition*, 1999.
63. S. N. Srihari, H. Srinivasan, C. Huang, and S. Shetty. Spotting words in latin, devanagari and arabic scripts. *Vivek: Indian Journal of Artificial Intelligence*, 16(3):2006, 2-9.

64. K. Taghva, J. Borsack, and A. Condit. Evaluation of model-based retrieval effectiveness with OCR text. *ACM Trans. Inf. Syst.*, 14(1):1996, 64-93.
65. K. Takeda, K. Kise, and M. Iwamura. Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved Iah. In *International Conference on Document Analysis and Recognition*, 2011.
66. J. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *European Conference on Computer Vision*, pages 696–709, 2008.51
67. A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. <http://www.vlfeat.org/>.
68. J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3360–3367, 2010.
69. I. Z. Yalniz and R. Manmatha. An efficient framework for searching text in noisy document images. In *IAPR International Workshop on Document Analysis Systems*, pages 48–52, 2012.
70. J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794–1801, 2009.