

MACHINE LEARNING OF HANDWRITTEN NANDINAGARI CHARACTERS USING VLAD VECTORS

Prathima Guruprasad¹ and Jharna Majumdar²

Department of Computer Science and Engineering, Nitte Meenakshi Institute of Technology, India

Abstract

This paper provides an early attempt to train and retrieve handwritten Nandinagari characters using one of the latest techniques in visual feature detection. The data set consists of over 1600 handwritten Nandinagari characters of different fonts, size, rotation, translation and image formats. In the Learning phase, we subject them to an approach where their recognition is effective by first extracting their key interest points on the images which are invariant to Scale, rotation, translation, illumination and occlusion. The technique used for this phase is Scale Invariant Feature Transform (SIFT). These features are represented in quantized form as visual words in code book generation step. Then the Vector of Locally Aggregated Descriptors (VLAD) is used for encoding each of the Image descriptors in the database. In the recognition phase, for query image, SIFT features are extracted and represented as query vector. Then these features are compared against the visual vocabulary generated by code book to retrieve similar images from the database. The performance is analysed by computing mean average precision. This is a novel scalable approach for recognition of rare handwritten Nandinagari characters with about 98% search accuracy with a good efficiency and relatively low memory usage requirements.

Keywords:

Handwritten Nandinagari Characters, Invariant Features, Scale Invariant Feature Transform, Image Vectorization, Indexing and Retrieval

1. INTRODUCTION

The cognition of the technique of writing might be one of the greatest wisdom towards the human civilization. Humans used writing as visual representation of a language. It is based on a script that is distinctively developed for the language in question. Scripts undergo modifications due to various linguistic, political and socio-cultural factors with the passage of time. A prevalent script may even be discarded as a whole and could be replaced by a new one. Nandinagari handwritten script has many unknown inscriptions which has knowledge as a treasure house of human wisdom [1]. This is the earlier version of Devanagari scripts which is widely used currently in India and in some parts of the world.

Written manuscripts are the most authentic source of information for understanding the history and culture of any society. Many Indian philosophers indicate that the source of wisdom comes from practicing Sages like Maharishi Ved Vyas and not fully agree with the version of Max Muller and Burnell who indicate that the technique of writing was unknown to Indians until a few centuries prior to the Christian era. This is also proven from the fact that ancient libraries in Mysore and Chennai have physically stored a wealth of these manuscripts and their origin is judged by Archeologists and Paleontologists to almost 5000 years.

Sri Anandathirtha, also named as Acharya Madhwa, who founded the Tatvavaada school of thought seven centuries back

has written many manuscripts in Nandinagari. These cover ancient research on different disciplines like Ayurveda, Mantra, Tantra, Vedanta, Jyotsna (Astrology), Nyaya, Vyakarana, Itihasa, Kavya, Natyasastra and Shiplashastra to name a few. There is a wealth of invaluable information in these manuscripts and hence need to be preserved and interpreted for passing across generations. Hence we have a compelling reason to convert them to a standard format which can then be indexed and searched.

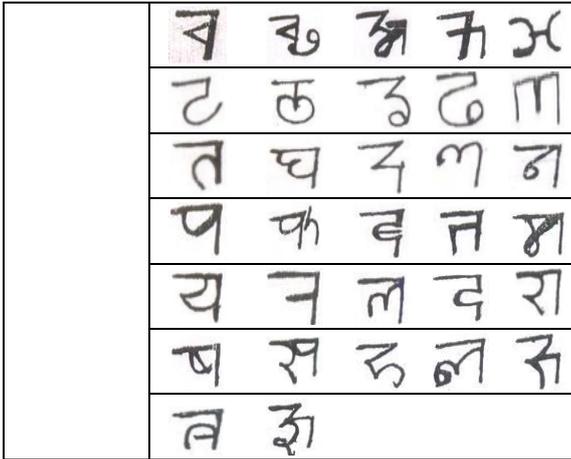
As shown in Table.1, Nandinagari character set has 52 characters out of which 15 are vowels and 37 are consonants. The Handwritten characters on manuscripts pose significant challenges and sufficient work need to be done in each step for their identification. To perform reliable recognition, it is important that for every character, the features are extracted from the image even under changes in image scale, noise and illumination. Typically, such points lie on high-contrast regions of the image.

In this paper we focus on extracting robust scale invariant features for the identification of handwritten Nandinagari characters. For processing historic handwritten Nandinagari characters we use Scale Invariant Feature Transform (SIFT) [2][3]. The detection and description of local image features will help in character recognition. The SIFT features are local and based on the appearance of the object at particular interest points. They are invariant to different sizes and orientations. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In addition to these properties, they are highly distinctive, relatively easy to extract and allow for correct object identification with low probability of mismatch [4][5].

Here after extracting invariant features using SIFT, Visual word representation of feature space in the form of code book is generated using a standard clustering approach like K Means. Then each image feature descriptors are vectorized using Vector of Locally Aggregated Descriptors (VLAD) approach. Next these amalgamated residual features are indexed and stored as a visual dictionary. Similar handwritten characters are retrieved from the vocabulary by comparing with the vector generated for the query character. The retrieval performance and recognition accuracy is computed using mean average precision method.

Table.1. Nandinagari Vowels and Consonants

Vowels	अ आ इ ई उ
	उ ए ऋ ॠ ऌ ॡ
	ॢ ओ औ आः आः
Consonants	क ख ग घ ङ



2. PROPOSED METHODOLOGY OF FRAMEWORK

The proposed architecture framework is as shown in Fig.1 consists of two important phases:

2.1 LEARNING PHASE

- *Scan distinct handwritten Nandinagari character and its variants:* Characters from two different writers are collected in the image form. Variants for these characters are manually prepared with different size, orientation, and occlusion. For handwritten Nandinagari character set, with an average of 5 different variations over the format of representation (jpg or png), size, degree of rotation and translation, we prepare a database of 1049 characters from 1st writer. The characters from the second writer consists of different categories of images such as normal images, convolved images, thinned images, and blurred images, noise images, images rotated by 15° and translated by 5 pixels. This forms 578 images in the data set. So from two different writers, a total 1627 images are taken for experimental analysis.
- In this approach, the preprocessing is kept to the minimal by just converting the images to grayscale. We need not apply a threshold on this image although it increases the recognition accuracy since the corner detection method is a very robust technique and we save the overall time of processing by avoiding the thresholding step.
- Candidate points from the input image are extracted using the SIFT technique. From each candidate point 128 feature descriptors are extracted which are invariant to scale, rotation and illumination.
- We then perform the training. Here visual words are generated by clustering the SIFT feature descriptor space using K Means clustering approach [6]. This is called as the SIFT descriptor code book which is the quantized representation of image.
- Represent the features by encoding them using VLAD vectorization technique. These features are subsequently indexed using Linear Indexing method and stored as visual vocabulary for faster retrieval. Berkeley Database (BDB) indexing environment is used for this purpose.

- In this step, we are aggregating a set of local descriptors into a VLAD vector [7]-[10]. We obtain the VLAD vector computed from a set of SIFT features as follows:
 - a) Each local descriptor x is associated to its nearest visual word $c_i = NN(x)$.
 - b) The idea of the VLAD descriptor is to accumulate, for each visual word c_i , the differences $x - c_i$ of the vectors \underline{x} assigned to c_i .

$$v_{ij} = \sum_{x \in NN(x)=c_i} x_j - c_{ij} \tag{1}$$

where x_j and c_{ij} respectively denote the j^{th} component of the descriptor x considered of its corresponding visual word c_i and component of v is obtained as a sum over all the image descriptors

2.2 IDENTIFICATION PHASE

- Take an unknown character as query image, extract SIFT key points and represent them in 128 descriptor format.
- Generate the query vector using VLAD vectorization technique based on generated visual code book in learning phase.
- Compare the query vector with visual vocabulary and retrieve the top N results of the images that match the query image.

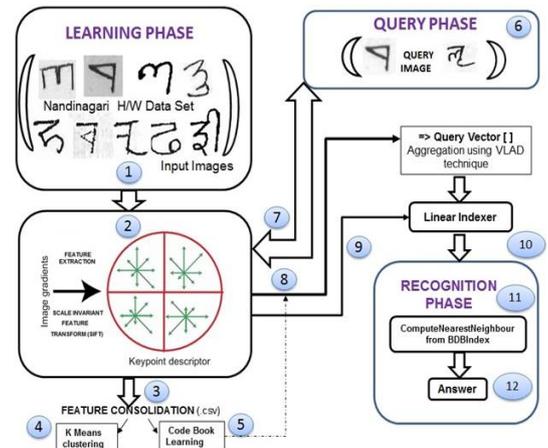


Fig.1. Proposed Architectural Framework for Handwritten Nandinagari Character Recognition

3. RESULTS AND DISCUSSION

The handwritten character set has 52 characters and for analysis two handwriting styles are used. The results are obtained for various stages of character recognition. The details of samples images for handwriting style 1 consists of different sizes 256×256, 384×384, 512×512, 640×640, different orientation angles 0°, 45°, 90°, 135°, 180°. This forms a 1049 characters of type 1 font in the database.

In handwriting style 2, we have different categories of images such as normal images of sizes 128×128 and 512×512. We also have convolved images, thinned images, blurred images, images with noise, images rotated by 15° and images translated by 5 pixels. All these images have various sizes including 92×92,

112×112 and 128×128. This forms 578 images in the data set. Thus a total of 1627 handwritten Nandinagari character images are taken for result analysis as shown in following steps.

3.1 SIFT FEATURE EXTRACTION

This step performs the SIFT feature extraction from a set of images (.jpg or .png files) contained in a folder and writes the extracted features in text files. For each image, this first key points of each image is identified and 128 feature descriptors are generated for each of these key points.

In this way a total of 1627 input images are fed to the feature extraction module and feature files are written in a directory with csv extension which represents different types and styles of images. Then all these feature files are merged to a single file and given as input to next stage. The feature analysis is as shown in Fig.2 and Fig.3. Average number of interest points for these images are ranging from 33 to 268 key points. And the average execution time is ranging from 32ms to 218ms. This is based on type of the image, image thickness and for various categories of image set the feature descriptor size is ranging from 4.954MB to 166.652MB. Overall for 1627 images in the data set the descriptor size is upto 438.695MB.

3.2 CODEBOOK GENERATION

This step generates the SIFT codebook using K means clustering approach. The code book file stores only cluster centers and the number of cluster centers depends on number of clusters formed based on the value of K . To analyze the performance we have generated the codebook with 16, 24, 32 and 52 cluster centroids. These are called visual words. These internally get mapped to all key points present in that particular cluster. The Table.2 shows the clustering time and code book size for different values of K . Even though clustering time is slightly high, this is acceptable for large robust feature set because code book is generated only once and same is used for further steps. The code book is the compact representation of SIFT feature descriptors.

3.3 IMAGE VECTORIZATION

In this step we aggregate a set of SIFT features into a VLAD vector and this is based on the algorithm mentioned in step 5 of section two. This reads the codebook of K centroids that was learned using 128 dimensional SIFT features. From this VLAD vector is generated by aggregating SIFT features mapped to nearest cluster centroid.

Table.2. Code book size and Clustering time different number of clusters

No. of Clusters	Code Book Size (KB)	Clustering Time (ms)
16	42	69777
24	62	97743
32	83	118895
52	134	178367

3.4 INDEXING AND RETRIEVAL OF QUERY IMAGES

This is used to generate an index for all database images and we use BDB (Berkeley Database) library for indexing purpose. This is based on persistent storage of the linear indexing approach and assigns the index for the images stored in the image folder. The index size varies from 25.8MB to 84MB for different values of K and so does the indexing time as shown in Table.3.

Table.3. Index size and Indexing time different number of clusters

No. of Clusters	Index Size (MB)	Indexing Time (ms)
16	25.8	67455
24	39.3	68961
32	51.75	69552
52	84	76179

The retrieval of similar images is obtained by comparing the query vector against the vectorized images using VLAD aggregator. Thus we compute the answer using the index of the images folder and the query image vector with nearest neighbor method. Here, top N similar images are retrieved from visual vocabulary. From the image set a representative of 8 query samples as shown in Table.4 are taken for analyzing the performance of our framework.

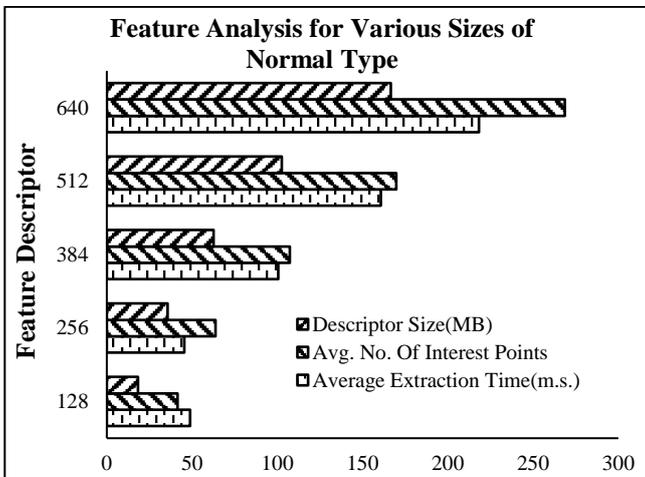


Fig.2. Average number of interest points, Average execution time and Feature Descriptor size for type 1 images

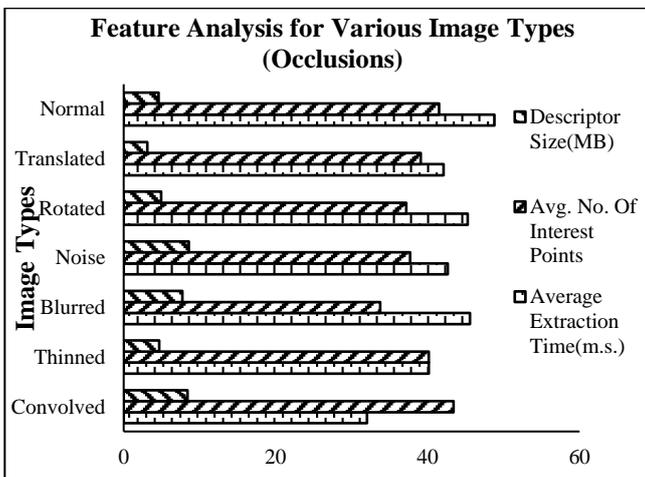


Fig.3. Average number of interest points, Average execution time and Feature Descriptor size for type 2 images

The retrieval performance is measured by precision at 10, precision at 20, precision at 30, precision at 40 and Average Precision (AP). The precision at N shows how accurate top N retrieved results are which is indicated in Fig.4 to Fig.8. Then Mean Average Precision (mAP) and retrieval time for all query images is computed as shown in Table.5.

Table.4. Query Image samples

Query Name	Query Image	Size and Orientations
AE		256x256, 45°
UU		384x384, 0°
LRU		512x512, 0°
GA		256x256, 0°
NNA		384x384, 0°
PHA		384x384, 0°
SA		256x256, 0°
HA		512x512, 0°

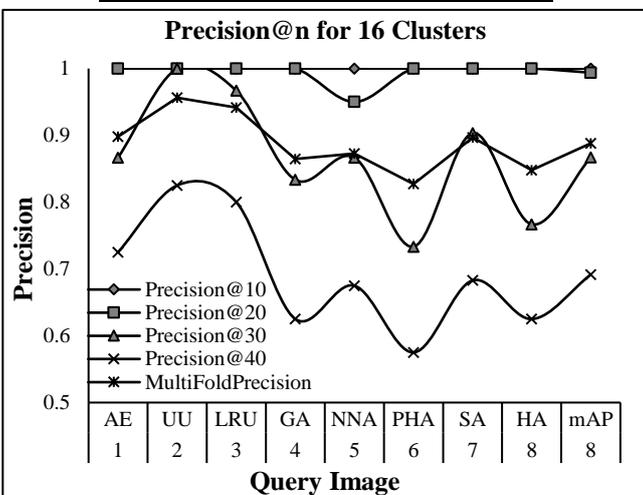


Fig.4. Precision at 10, 20, 30 and 40 for 16 clusters

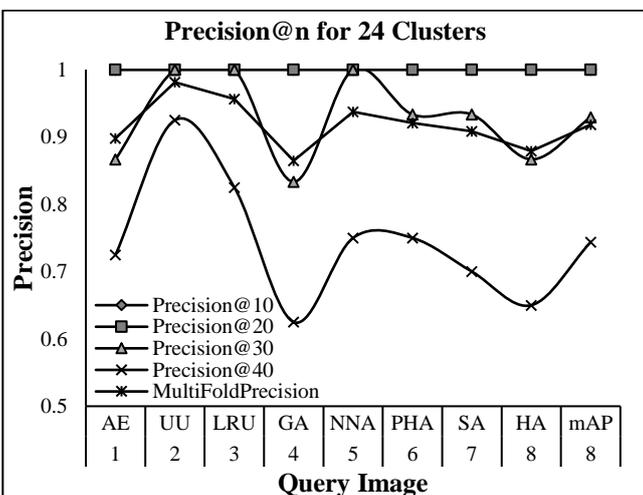


Fig.5. Precision at 10, 20, 30 and 40 for 24 clusters

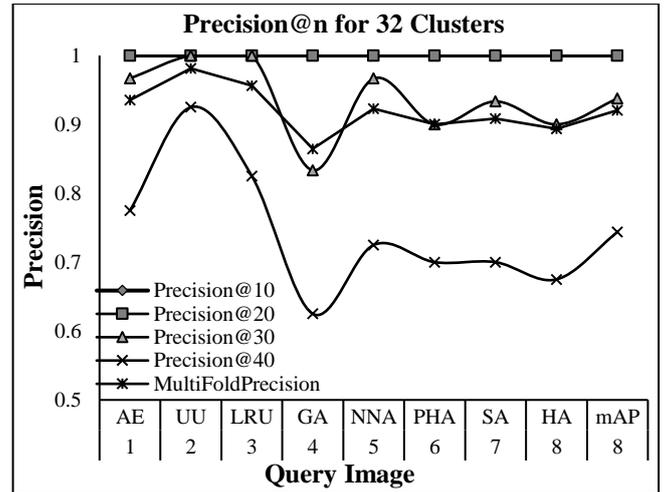


Fig.6. Precision at 10, 20, 30 and 40 for 32 clusters

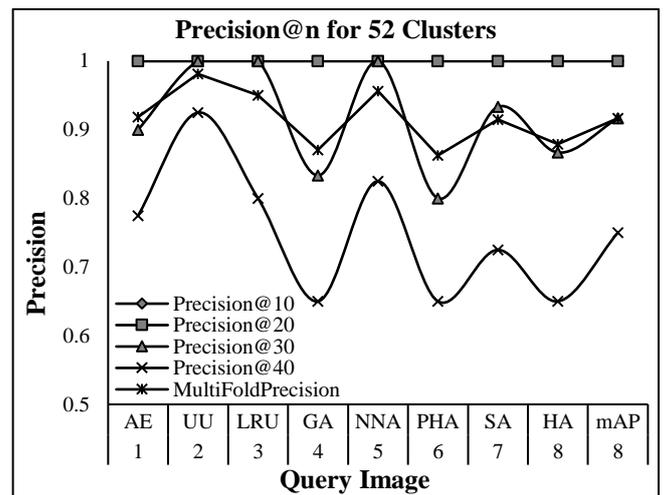


Fig.7. Precision at 10, 20, 30 and 40 for 52 clusters

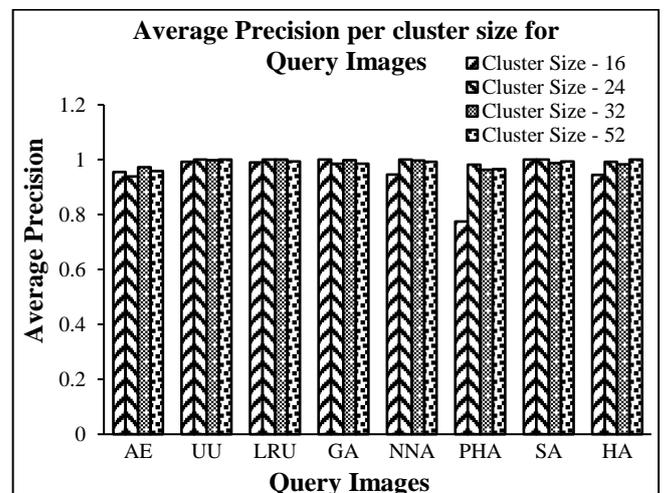


Fig.8. Average Precision for of query Images for different cluster sizes

Table.5. The mean average precision and retrieval time of different number of clusters

No. of Clusters	mAP	Retrieval Time (ms)
16	0.9504	39.25
24	0.9874	47.5
32	0.9876	56.625
52	0.9862	74.25

From Table.5, we conclude that mAP is maximum for a specific number of clusters where the recognition accuracy is also more. The Table.6 shows the top 10 retrieved results for the input query image. Thus for the proposed system on handwritten Nandinagari character recognition, the optimal clusters to be chosen is 32. The recognition accuracy has improved significantly in VLAD vectorization as compared to Bag of Visual words approach as discussed in [11].

Table.6. Results of retrieval, where first column shows the query images and the retrieved images are shown in decreasing order, from left-to-right

Query Image	Retrieved top 10 Images with different scale, orientation and occlusions									
										
AE, 256×256, 45°	256×256, 0°	256×256, 90°	256×256, 135°	256×256, 180°	384×384, 135°	512×512, 135°	512×512, 45°	384×384, 180°	384×384, 90°	384×384, 0°
										
LRU, 512×512, 0°	256×256, 0°	256×256, 45°	256×256, 90°	256×256, 135°	256×256, 180°	384×384, 0°	384×384, 45°	384×384, 90°	512×512, 45°	128×128, trans5
										
GA, 384×384, 0°	256×256, 0°	256×256, 45°	256×256, 90°	256×256, 135°	256×256, 180°	128×128, 0°	128×128, 5°	128×128, Thin	128×128, Noise	384×384, 45°
										
HA, 512×512, 0°	256×256, 0°	256×256, 45°	256×256, 90°	256×256, 135°	256×256, 180°	128×128, 0°	384×384, 0°	384×384, 180°	128×128, Noise	384×384, 45°

Our approach which supports compact representation of large set of features is compared with traditional classification algorithms and performance is analyzed. The VLAD vectorization gives better result as compared to other approaches, shown in Table.7.

Table.7. Comparison of Recognition Accuracy using different classification approaches

Classification Algorithm	Retrieval Time (ms)	Classification Accuracy using 10 fold cross validation
Multilayer Perceptron	120.25	96.340%
Support Vector Machine	100.67	96.020%
Naive Bayes	96.09	95.667%
Logistic Regression	90.90	93.120%
Decision Tree	85.76	92.667%
Our Approach	56.625	98.760%

Currently we are focusing on applying deep learning concept for identifying these rare Nandinagari characters and words. For this we

need to generate large number of training and test datasets. Generating standard data set for this work is in progress. The concept is implemented on prototype model and giving better results. The processing is done with an Intel(R) code(TM) i5 - 5200 U CPU @ 2.20GHz, 12GB RAM, 64 bit operating system (Windows 8.1), x64 based Processor having Java 1.8 installed and the test is done in multithread with Java client.

4. CONCLUSION

The proposed smart system for handwritten Nandinagari character retrieval based on data visualization method is highly robust and scalable. The Scale-Invariant Feature Transform (SIFT) detects the interest points and derives feature descriptors. Our main aim is to provide an improved descriptor which can make the system compatible with a wide variety of document degradations. In this work we apply a K means clustering approach for codebook generation which is quantized representation of huge SIFT feature descriptors. This filters the characters that typically tend to get misclassified thereby improving the recognition accuracy. This can also be performed to compute rapidly against large databases. Focusing more on learning document-specific local descriptors gives mAP of over

98%. More handwritten images can be indexed on to the existing BDB thus providing a highly scalable model.

REFERENCES

- [1] P. Visalakshi, “*Nandinagari Script*”, 1st Edition, DLA Publication, 2003.
- [2] D.G. Lowe, “Distinctive Image Features from Scale-Invariant Key Points”, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.
- [3] E.N. Mortensen, H. Deng and L. Shapiro, “A SIFT Descriptor with Global Context”, *Proceedings of IEEE International Conference in Computer Vision and Pattern Recognition*, Vol. 1, pp. 184-190, 2005.
- [4] Ives Rey-Otero, Jean-Michel Morel and Mauricio Delbarcio, “An Analysis of Scale-Space Sampling in SIFT”, *Proceedings of IEEE International Conference on Image Processing*, pp. 15-19, 2014.
- [5] Ravi Shekhar and C.V. Jawahar, “Word Image Retrieval using Bag of Visual Words”, *Proceedings of 10th IAPR International Workshop on Document Analysis Systems*, pp. 1-6, 2012.
- [6] Akanksha Gaur and Sunita Yadav, “Handwritten Hindi Character Recognition using K means Clustering and SVM”, *Proceedings of 4th International Symposium on Emerging Trends and Technologies in Libraries and Information Services*, pp. 115-119, 2015.
- [7] Herve Jegou, Matthijs Douze, Cordelia Schmid and Patrick Perez, “Aggregating Local Descriptors into a Compact Image Representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence, Institute of Electrical and Electronics Engineers*, Vol. 34, No. 9, pp. 1704-1716, 2013.
- [8] Jonathan Delhumeau, Philippe-Henri Gosselin, Herve Jegou and Patrick Perez. “Revisiting the VLAD Image Representation”, Available at: <https://hal.inria.fr/hal-00840653v1/document>, Accessed on 2013.
- [9] Relja Arandjelovic and Andrew Zisserman, “All about VLAD”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1578-1585, 2013.
- [10] David Picard and Philippe-Henri Gosselin, “Improving Image Similarity with Vectors of Locally Aggregated Tensors”, *Proceedings of IEEE International Conference on Image Processing*, pp. 669-672, 2011.
- [11] Prathima Guruprasad and Jharna Majumdar, “Handwritten Nandinagari Image Retrieval System based on Machine Learning Approach using Bag of Visual Words”, *International Journal of Current Engineering and Scientific Research*, Vol. 4, No. 4, pp. 163-168, 2017.