Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

Advanced System for Identifying Ancient Photographs Using Contemporary Tagged Images

Prashant Ghulappanavar[1] , Hameem Shanavas[2]
[1*]*Research Scholar, MVJCE Bengaluru, Karnataka, India*
[2]*Professor, Dept of ECE, MVJCE* Bengaluru, *Karnataka, India*

Abstract: - Huge collections of imagery on the Internet have inspired a wave of approaches to location recognition, the problem of determining where a photo was taken by comparing it to a database of images of previously seen locations from the past few years. Due to this, as excitement in these areas increases, a world-scale location recognition engine from all the geo-tagged pictures from online photo collections, such as Flicker, Instagram, and street view databases from Microsoft and Google. Matching modern historical images to old ones requires a special effort when dealing with historical photos. The performance of many algorithms is good on modern pictures but is not very efficient on old historical photos. A novel approach to place identification, taking pictures taken in different weather conditions, illuminations, and positions, is proposed in this paper, expecting more accuracy than other existing systems.

Keywords*: rootSIFT (rSIFT), Convolutional Neural Networks (CNNs), Hessine Affine (HA), Fisher Vector (FV).*

I. Introduction

The matching feature has been extensively researched and applied to computer vision, pattern recognition, image processing, and object detection. One such approach is the matching of modern to historical photos. Many travelers, historians, historical campaigns, and researchers are interested in matching images that depict a timeline view of specific locations or landmarks. Capturing a photo of a location of an existing place by a photographer in the past was traditionally done using photographic films. These photos are usually only shared with a few friends or family, so the photographer's picture usually has limited quality. With the exponential growth of the World Wide Web and the growth of photo-sharing websites like Flicker, a massive change in the size of photo processing and photo Collections has happened with digital photography's invention.

We now have many photos of cities, monuments, and places worldwide. Therefore, dealing with these images and extracting their features is a challenging task as a great set of pictures of specific benchmarks captured countless times by different photographers with different cameras from different points of view and in various weather conditions is difficult.

Developmental stages of landmarks or detecting deformation or damage that has occurred can be discovered during processing steps using these features. There are different feature extraction approaches, such as vector-based and binary-based. In our paper, we made use of vector-based approaches. A photo set of famous landmarks in the world has been gathered. Different photographers captured these photos countless times, from various perspectives, using different cameras and different weather conditions. Figure 2 shows pictures of 6 landmarks with historical and modern images. Compared to contemporary digital images, you can see old photographs with different textures, colors, and contrast characteristics. All these images can be used to create a novel place recognition system during implementation.

II. Existing System

Soonmin Bae et al. [04] presented a real-time visualization and estimation approach for rephotograph that helps users reach a desired location during capture. Reference images taken from the chosen viewpoint are the input to this approach using SIFT Detector, ANN matching, RANSAC, and a 5-point algorithm. Nikolas Hesse et al. [10] presented the guidelines for optimizing the performance and exploring how well a standard place recognition system is suited to handle IR data. The system will increase enormously if SIFT descriptors computed on Hessian Affine regions are used instead of SURF features taking three datasets. Bolei Zhou et al. [11] used Convolutional Neural Networks (CNNs) to learn bottom-line features for scene recognition tasks and establish new state-of-the-art results using many scene-centric datasets. Working on the CNN layer's responses shows differences in the internal representations of object-centric and Scene-centric networks.

Basura Fernando et al. [12] presented a dataset spanning over 25 locations and more than one century. They analyzed several representations, looking for the most novel approach to the variability induced by color degradation and different image acquisition processes. Experimental evolution has depicted that the Hessian Affine detector, root-SIFT, and fisher vector are more suitable for the task at hand than other detector and descriptor pairs. Niko Sunderhauf et al. [14] comprehensively compared the behavior of three state-of-the-art ConvNets on the problems of particular relevance to navigation for robots. An extensive experiment presentation took four real-world datasets cultivated to evaluate each specific challenge in place recognition. Networks trained for semantic categorization networks also place better in the recognition site when faced with extreme changes in appearance and provide a reference for the networks. The layers are optimal for various aspects of the place recognition problem.

III. Proposed system

The methodology presented here is a novel approach to recognizing historical images. It consists of two phases: training and testing.

A. Training phase

In the training phase, as in Figure 1, input modern images are pre-processed based on the dataset chosen and are passed to the detector block. Where the detection is done using Hessine Affine (HA)[01] and Dense detectors [08], this output is passed to the descriptor block, which is used to do a feature description using the rootSIFT (rSIFT) descriptor. Representation is done using Fisher Vector (FV). The ESA domain adoption technique is also used to avoid problems caused by data variability. The extracted feature is stored in the Knowledge base after training the Convolutional Neural Network (CNN).

Fisher Vector

Let $X = \{xt, t = 1 \dots T\}$ be the set of T local descriptors extracted from an input image. By probability density function u_λ with parameters λ [4] we assume that the generation process of X can be modeled. By the gradient, vector X can be described.

$$G_\lambda^X = \frac{1}{T} \nabla_\lambda \log u_\lambda(X) \tag{01}$$

The log-likelihood gradient describes the contribution of parameters for the generation process. This vector‘s dimensionality depends only on the number of parameters in λ but not on the number of patches T. These gradients' natural kernel is represented as,

$$K(X, Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y \tag{02}$$

where F_λ is the Fisher information matrix of u_λ:

$$F_\lambda = E_{x \sim u_\lambda}[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)'] \tag{03}$$

As F_λ is symmetric and positive definite, it has Cholesky decomposition $F_\lambda = L'_\lambda L_\lambda$ and $K(X, Y)$ can be rewritten as a dot product between normalized vectors G_λ with,

$$g_\lambda^X = L_\lambda G_\lambda^X \tag{04}$$

As the Fisher vector of X refers to G_λ^X. We choose u_λ to be the Gaussian mixture model (GMM),

$$u_\lambda(x) = \sum_{i=1}^{K} w_i u_i(x) \tag{05}$$

λ Is denoted as $\lambda = \{w_i, \Sigma_i, i = 1 \ldots K\}$, where w_i, μ_i and Σ_i are mixture weight, mean vector, and covariance matrix of Gaussian u_i respectively. Assuming covariance matrices to be diagonal, we denote by the variance vectorσ_i^2. Using Maximum Likelihood (ML) estimation, the GMM u_λ is trained on a large number of images. Assuming that the x_t 's are generated by u_λ independently and, therefore,

$$G_\lambda^X = \frac{1}{T} \sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t) \tag{06}$$

Concerning the standard deviation and mean parameters (the gradient concerning the weight parameters brings little additional information), we consider the gradient. Diagonal closed form approximation, in which case the normalization of the slope by $L_\lambda = F_\lambda^{-1/2}$ is simply a Whitening of the dimensions used. Consider $\gamma_t(i)$ be the soft assignment of x_t descriptor to Gaussian i,

$$\gamma_t(i) = \frac{w_i u_i(x_t)}{\sum_{i=1}^{K} w_i u_i(x_t)} \tag{07}$$

The dimensionality of the descriptors x_t is denoted by D. $\mathcal{G}_{\mu,i}^X$ be the D-dimensional gradient concerning the mean u_i of Gaussian i. The mathematical derivations are given by,

$$\mathcal{G}_{\mu,i}^X = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - u_i}{\sigma_i} \right) \tag{08}$$

$$\mathcal{G}_{\sigma,i}^X = \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - u_i}{\sigma_i^2} - 1 \right) \tag{09}$$

where the division operation between the vectors is a term by term. The final gradient vector obtained G_λ^X is the concatenation of the $\mathcal{G}_{x,i}^X$ and $\mathcal{G}_{\sigma,i}^X$ for i=1. . . K, and therefore it is 2K D dimensional [03].

Root SIFT

This approach is well known for areas such as image categorization and texture classification, which uses Euclidean distance to compare histograms, which often yields inferior performance compared to using measures such as $\mathcal{X}^2$ or Hellinger. SIFT [09] was initially designed to be used with Euclidean distance. But here, we used the Hellinger kernel to bring a more significant benefit. Let us consider that x and y is n vectors with unit Euclidean norm($\|X\|_2 = 1$), then $d_E(X, Y)$ the Euclidian distance between them is related to their similarity $S_E(X, Y)$ as

$$d_E(X, Y)^2 = \|X - Y\|_2^2 = \|X\|_2^2 + \|y\|_2^2 - 2X^T Y \qquad = 2 - 2S_E(X, Y) \tag{10}$$

where $S_E(X, Y) = X^T Y$ and the last step follows from$\|X\|_2^2 + \|y\|_2^2 = 1$. We want to replace the Euclidean similarity/kernel with the Hellinger kernel. This kernel is also known as the Bhattacharyya's coefficient for two L1 normalized histograms, X and Y i,e $\sum_i^n x_i = 1$and $x_i \geq 0$), is defined as,

$$H(X, Y) = \sum_{i=1}^{n} \sqrt{x_i y_i} \tag{11}$$

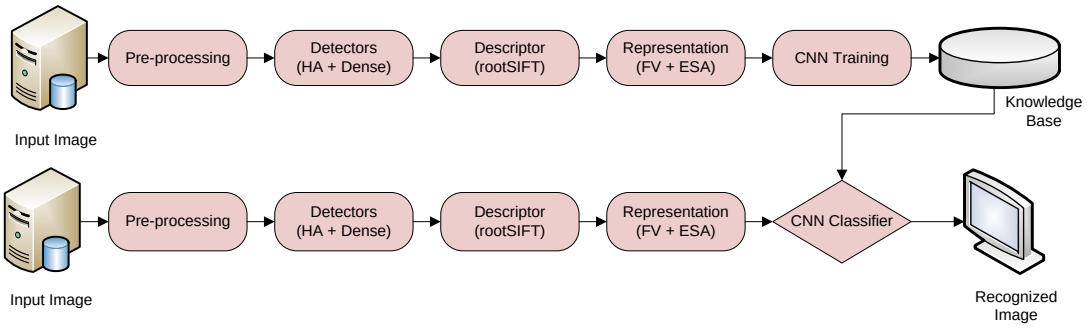

Figure 1: Block Diagram of Place Recognition System

Using a simple algebraic manipulation in two steps: 1) L1 normalize the SIFT vector (initially, it has unit L2 norm);2) square root each element. It is then followed by $S_E(\sqrt{X}, \sqrt{Y}) = \sqrt{X}^T\sqrt{Y} = H(X, Y)$, and the resulting vectors are L2 normalized since $S_E(\sqrt{X}, \sqrt{X}) = \sum_{i=1}^{n} x_i = 1$. Hence, a new descriptor is described, which is called RootSIFT. This is an element-wise square root of the normalized L1 SIFT vectors. Comparing RootSIFT descriptors using Euclidean distance is equivalent to using the Hellinger kernel to compare the original vectors of SIFT,

$$d_E(\sqrt{X}, \sqrt{Y})^2 = 2 - 2H(X, Y) \tag{12}$$

RootSIFT is used in the specific object retrieval pipeline by simply replacing SIFT with RootSIFT at every point [05].

Extended subspace Alignment(ESA)

The Subspace Alignment (SA) method learns a linear transformation matrix $M \in .R^{d_s \times d_T}$ Which aligns the target and source coordinate systems by mining the below given Bregman divergence,

$$F(M) = \|X_s M - X_T\|_F^2 \tag{13}$$

where $\|.\|_F^2$ denotes the Frobenius norm. It is shown that the optimal matrix is $M = X_S' X_T$, and $X_a = X_s X_S' X_T$. The similarity between these two samples is given by,

$$Sim(X_s, X_T) = (x_S, X_a)(x_T X_T)' \tag{14}$$

The Demonstration of deviation between two successive Eigenvalues to be bounded can be shown. We make use of bounds to determine the maximum size of the subspaces d_{max} that to get a non-overfitting and stable matrix M. Subspace dimensionality d can then be done by minimizing the classification error through twofold cross-validation over the labeled source data and finally setting $d_s = d_T = d$. For more information, refer to [06].

The equation in (14) operates in the original R^D space. Any problem can be formulated in the R^{d_T} target subspace after the domain transformation. To reduce the computational complexity, ESA proposes a new approach to evaluate the similarity between target subspaces projected data and target-aligned source samples by using their Euclidean distance[07] directly,

$$\Theta(x_S, x_T) = \|x_S X_a - x_T X_T\|_2 \tag{15}$$

When working with data represented by high-dimensional features, the cross-validation procedure described to define the best d for SA becomes very slow and tedious. In cases where some kinds of origin have minimal recorded samples, reliable results are unlikely to be provided. The two domains are considered separately, which implies $d_s \neq d_T$. For more information, refer to [07]. The flow for FV is as in Figure 2.

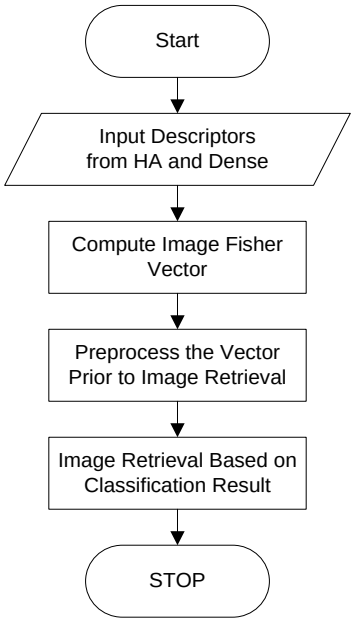

Figure 2: Pre and Post steps for Fisher Vector

B. Testing phase

In the testing phase, the query image is pre-processed based on the chosen dataset and passed to the detector block. Where the detection is done using Hessine Affine (HA) [02] and Dense detectors, this output is passed to the descriptor block, which is used to do a feature description using the rootSIFT (rSIFT) descriptor. Representation is done using Fisher Vector (FV). The output of this is passed to ESA in the training phase. The extracted feature is compared with the features stored in the Knowledge base using a CNN classifier. Classified output is then displayed in Figure 3.

Test Image	Training Image					Locations
						Taj Mahal
						Great wall of china
						Chiche Nitza
						Statue of Liberty

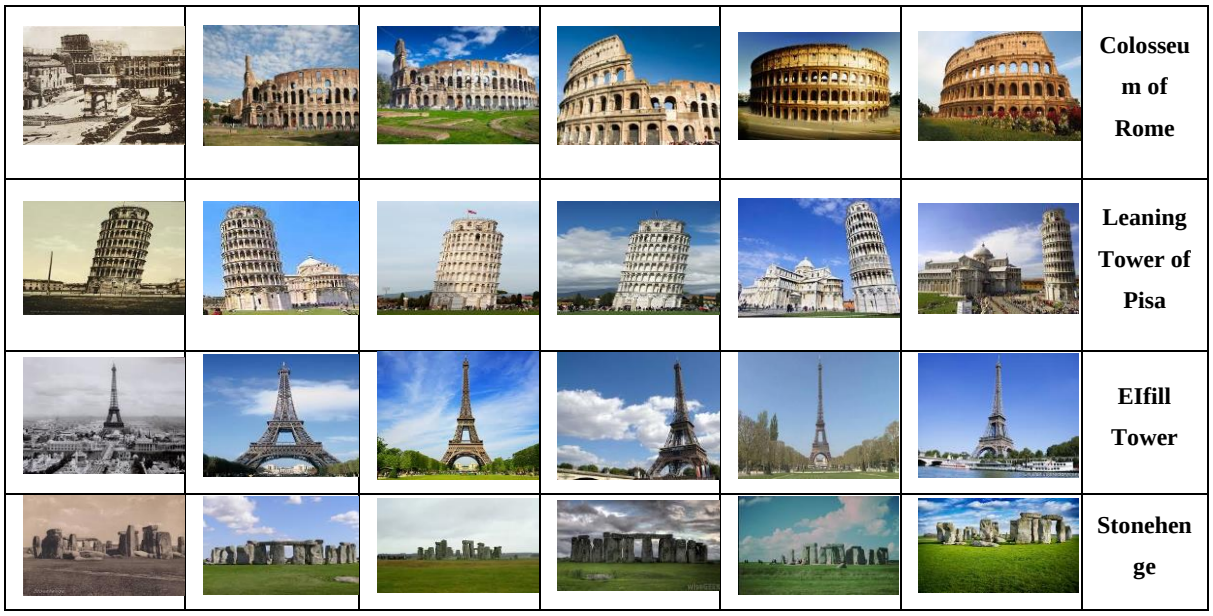

Figure 3: Pictures of Seven Locations Over Large Time Lags Showing an Evident Change in Appearances

C. Convolutional Neural Networks(CNN)

CNNs are multi-layer supervised networks that can learn features automatically from datasets. CNNs have achieved state-of-the-art performance in almost all essential classification tasks for the last few years. Place recognition is an important task of image similarity matching. According to the benchmark demonstration result, in-depth features from different layers of CNNs consistently perform better than other matching techniques; mid-level features of CNNs are evaluated for implementing image retrieval and achieving comparable performance characteristics using other prior art. The best performance is obtained using mid-network features rather than those learned at the final layers [23]. The three types of layers present in a convolutional neural network are:

Convolutional:

Layers consisting of a rectangular grid of neurons are called convolutional layers. The requirement is that the previous layer is a rectangular grid of neurons. Each neuron takes its input from a rectangular section of the previous layer. The weights for the rectangular section are the same for each neuron in the convolutional layer. Therefore, this layer is not more than the convolution image of the previous layer, and weights indicate that the filter is convolutional. In addition to it, there may be many grids in each convolutional layer. Using potentially different filters, each grid takes inputs from all the grids in the previous layer.

Consider we have some $N \times N$ square neuron layer followed by our convolutional layer. If we use an $m \times m$ filter ω, then our convolutional layer output of size$(N - m + 1) \times (N - m + 1)$. For computing the pre-nonlinearity input to some unit $x_{i,j}^{l}$in this layer, it is necessary to sum up the contributions (weighted by the filter components) from the cells of the previous layer. Figure 4 depicts the different layers of CNN.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{ab}\, y_{(i+a)(j+b)}^{l-1} \tag{15}$$

This approach is taken during forward propagation. Then, this layer applies its nonlinearity, and the relation is given by,

$$y_{i,j}^{l} = \sigma\left(x_{ij}^{l}\right) \tag{16}$$

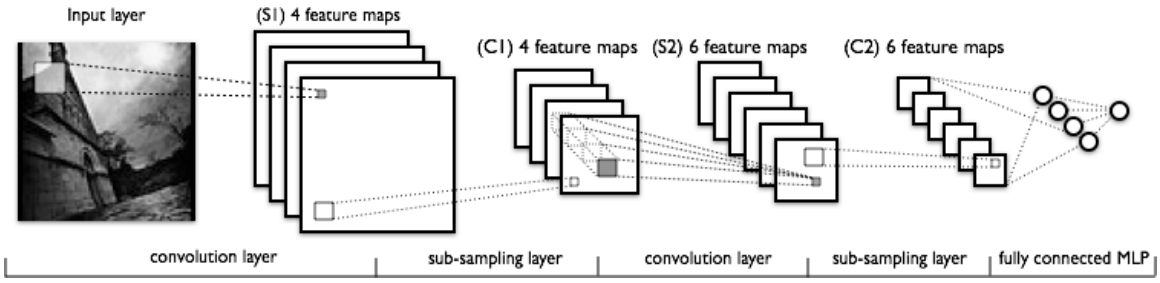

Figure 4: Different Layers of Convolutional Neural Network (CNN)

Max-Pooling:

There may be a pooling layer after every convolutional layer. This layer takes small rectangular blocks from the convolutional layer and produces a single output from that block subsamples it. There are several approaches to do this poling, such as taking the average or the maximum and learning the linear combination of the neurons in the block. These layers are always max-pooling layers, i.e., they take the maximum amount of the block they are pooling. Comparatively, max-pooling layers are superficial, and they take $k \times k$ region and give a single value as the output, which is the maximum in that region. Suppose their input layer is a $N \times N$ layer, then the output is a $\frac{N}{k} \times \frac{N}{k}$ Layer, as each $k \times k$ block is reduced to a single value via the maximum function.

Fully-Connected:

The high-level reasoning in the neural network is done via fully connected layers after several convolutional and max pooling layers. Fully connected layers are not spatially located anymore, so there can be no convolutional layers after a fully connected layer, as fully connected layers are not spatially located anymore. For more details, refer to [23]. The max-pooling layers do not do any learning themselves. Instead, they reduce the size of the problem by introducing sparseness. $k \times k$ blocks are reduced to a single value in forward propagation. Then, from the previous layer, this single value acquires an error computed from backward propagation. This error is then just forwarded to the place where it originated. The back-propagated errors from max-pooling layers are relatively sparse since they only came from one place in the $k \times k$ block.

IV. Experimental Result

As we know, modern to-historical image matching is a severe issue in many internet Media, so an efficient method to achieve this is necessary. In this methodology, we propose a new job of naming the places given in an old photo utilizing modern images. A set of contemporary images where trained and historical old images are shown as the test images for matching purposes.

The Matlab2012a tool is utilized to find the results of the approached method. Figure 5 explains the input images considered for testing the system. Figure 5 (a) shows the input image, which is present in a grayscale image; the intensity of the image is very poor; to enhance the image's brightness, pre-processing contrast stretching is used. Figures 5(b), (c), and (d) show the color image that was captured recently.

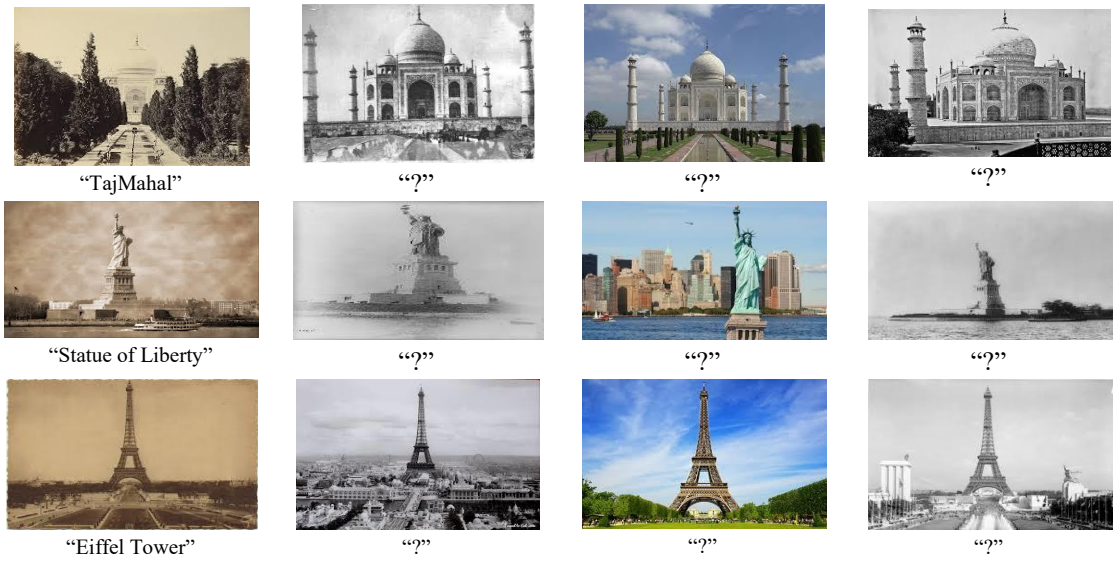

Figure 5: Pictures of Three Locations over Large Time Lags Showing an Evident Change in Visual Appearance

Figure 3 depicts the testing and training images. 8 historical image and their corresponding modern images with similarity matching are given. Proper matches for all the images are found with reasonable accuracy. A total of 8 classes are taken, each having 5 images. 40 images are trained, and the testing folder consists of 8 old photos. Figure 6 shows that the existing system without domain adoption has a recognition rate of 48.5%, and a recognition rate of 49% is obtained using our proposed method. Figure 7 depicts the comparison graph between Existing and proposed domain adoption methods. The comparison says the proposed system gives better retrieval results than existing systems.

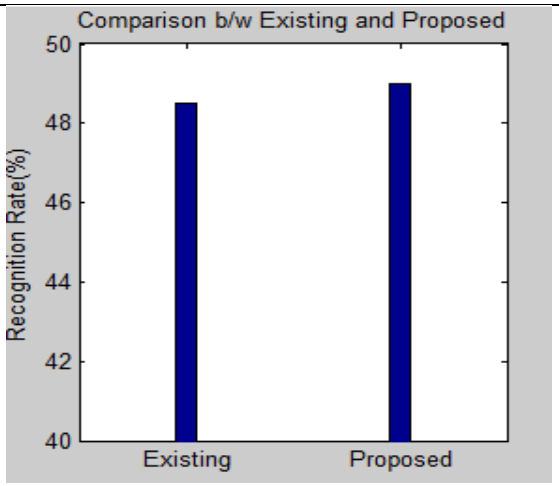

Figure 6: Comparison Graph for Existing and Proposed Method without Domain Adoption Method.

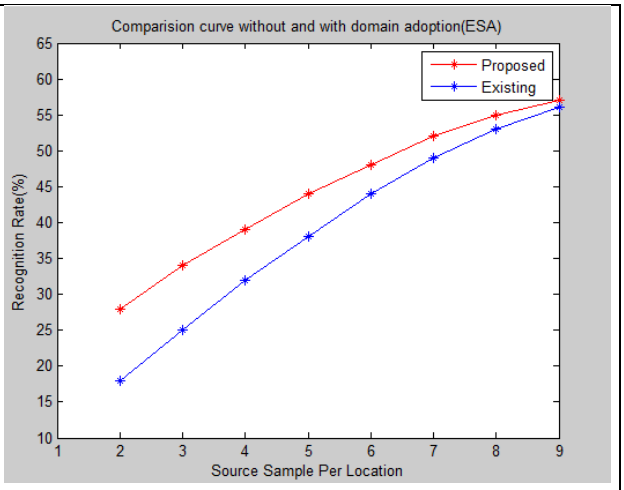

Figure 7: Comparison Graph for Existing and Proposed Method with Domain Adoption Method.

V. Conclusion

Modern to-historical image matching is a severe issue for many internet media, so an efficient method to achieve this is essential. In this paper, we proposed a new task of recognizing the places given in an old photograph using modern images. A set of contemporary images was trained, and historical old photos were provided as the test images for matching purposes. Our analysis depicted a robust approach involving a Hessian Affine detector and

dense detector with root-SIFT descriptor along with Fisher vector, ESA domain adoption method, and CNN classifier to get improved results compared to other place recognition systems.

Reference

[1] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. Van Gool, "A comparison of affine region detectors," Springer, Vol. 65, Issue 1, pp. 43-72, 2005.
[2] Michal Perdoch, Ondrej Chum and Jiri Matas, "Efficient Representation of Local Geometry for Large Scale Object Retrieval," pp. 9 – 16, IEEE, 2009.
[3] Florent Perronnin, Jorge Sanchez and Thomas Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," ACM, 2010.
[4] Soonmin Bae and Fredo Durand, "Computational Re-Photography," ACM, 2010.
[5] Relja Arandjelovic and Andrew Zisserman, "Three things everyone should know to improve object retrieval," IEEE, pp. 2911 – 2918, 2012.
[6] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, "Unsupervised Visual Domain Adaptation Using Subspace Alignment," IEEE, pp. 2960 – 2967, 2013.
[7] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars, "Subspace Alignment for Domain Adaptation," ACM, 2014.
[8] Basura Fernando, Tatiana Tommasi, and Tinne Tuytelaars, "Lost in the Past: Recognizing Locations Over Large Time Lags," ACM, 2014.
[9] Heider K. Ali and Anthony Whitehead, "Modern to Historic Image Matching: ORB/SURF an Effective Matching Technique," 2014.
[10] Nikolas Hesse, Christoph Bodensteiner, and Michael Arens, "Performance Evaluation of Image-Based Location Recognition Approaches Based on Large-scale UAV Imagery," 2014.
[11] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning Deep Features for Scene Recognition Using Places Database," 2014.
[12] Basura Fernando, Tatiana Tommasi, and Tinne Tuytelaars, "Location Recognition over Large Time Lags," ACM, 2015.
[13] Zetao Chen, Obadiah Lam, Adam Jacobson, and Michael Milford, "Convolutional Neural Network-based Place Recognition," 2015.
[14] Niko S¨underhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford, "On the Performance of ConvNet Features for Place Recognition", IEEE, 2015.