Latika Pinjarkar*, Manisha Sharma and Smita Selot

Deep CNN Combined With Relevance Feedback for Trademark Image Retrieval

https://doi.org/10.1515/jisys-2018-0083
Received February 10, 2018; previously published online September 15, 2018.

Abstract: Trademark recognition and retrieval is a vital appliance component of content-based image retrieval (CBIR). Reduction in the semantic gap, attaining more accuracy, reduction in computation complexity, and hence in execution time, are the major challenges in designing and developing a trademark retrieval system. The direction of the proposed work takes into account these challenges by implementing trademark image retrieval through deep convolutional neural networks (DCNNs) integrated with a relevant feedback mechanism. The dataset features are optimized through particle swarm optimization (PSO), reducing the search space. These best/optimized features are given to the self-organizing map (SOM) for clustering at the preprocessing stage. The CNN model is trained on feature representations of relevant and irrelevant images, using the feedback information from the user bringing the marked relevant images closer to the query. Experimentation proved a significant performance when evaluated using FlickrLogos-27, FlickrLogos-32, and FlickrLogos-32 PLUS datasets, as illustrated in the performance results section.

Keywords: Content-based image retrieval (CBIR), deep convolutional neural networks (DCNNs), particle swarm optimization (PSO), self-organizing map (SOM), relevance feedback, trademark.

1 Introduction

The automatic trademark retrieval system evaluates the query trademark among the database of trademark images for similarity, in approving the trademark of the corporation/organization. The majority of the content-based image retrieval (CBIR) applications based upon retrieval, require the image to search, to match, and to retrieve. Conversion of the image's high level semantics inferred by individuals to low-level semantics' illustration is the key challenge of the work, giving rise to the possibility of research in this area.

A typical CBIR system involves feature extraction, similarity computation, and retrieval of output images based on similarity. The feature vector's dataset is constructed by implementing feature extraction to deduce shape, color, and texture representation for images of a trademark. This database of features may include a few irrelevant and superfluous features, unnecessary for retrieval process. Hence, the particle swarm optimization (PSO) technique is employed to optimize this feature set, narrowing the search space of the retrieval process.

The optimization practice, PSO, exploits a parallel exploration of various points, which are adjusted in the search room. The PSO has the advantage of fast convergence while being judged against other optimization practices, for example, global optimization algorithms and genetic algorithms [13]. The PSO is a potential technique for optimization but does not do assurance optimization [1]. Therefore, clustering is employed upon this optimized feature set through the self-organizing map (SOM) prior to the search process implementation. The SOM is trained upon the best dataset features obtained. This training entails the input vectors being plotted next to one another in the adjacent plot elements [16]. The SOM arranges the optimized feature set in the form of clusters. After giving a query to the system, the images are searched in some top relevant clusters,

***Corresponding author: Latika Pinjarkar,** Shri Shankaracharya Technical Campus, Bhilai (C.G.) 490020, India, e-mail: latikabhorkar@gmail.com
Manisha Sharma: Bhilai Institute of Technology, Bhilai (C.G.) 490020, India
Smita Selot: Shri Shankaracharya Technical Campus, Bhilai (C.G.) 490020, India

and retrieval is done based on similarity. User feedback is taken on these retrieved images regarding relevance. The information obtained through the user's feedback, i.e. relevance feedback (RF), is used in the query modification process [20, 26].

The proposed work is intended for the improvement of the RF performance by utilizing the convolutional neural networks (DCNNs) for trademark image recognition and retrieval. The proposed scheme is to employ the capability of a DCNN to revise its inner organization to provide the improved representations of the images applied for the improvement of retrieval procedures through the user's feedback. The feature extraction is executed through the deepest neural layers of the convolutional neural network, in order for the feature illustrations of the relevant images to come nearer to the query and for the not relevant images to deviate from the query image.

The suggested framework is a combination of the following two approaches applied for trademark image retrieval: (1) the novel RF approach based on query modification and mining navigation patterns and (2) the influence of DCNN in the RF approach.

The rest of the paper is organized as follows: Section 2 explains the novelty and contributions of the suggested framework. Literature regarding the related contributions in the area is described in Section 3. The proposed framework is depicted in Section 4. The experimentation results are presented in Section 5, and in Section 6, conclusions are drawn.

2 Novelty and Contributions of the Suggested Framework

Successful RF methodologies suggested earlier are combined with the learning approaches like decision trees, support vector machines, and boosting techniques. The proposed work utilizes the power of DCNN in improving the performance of the RF implemented through the following two phases:

1. Preprocessing phase: optimization of the feature dataset using PSO and clustering using SOM.
2. Retrieval phase: query modification and navigation pattern mining-based RF is combined with DCNN for trademark recognition and retrieval.

Feedback information records are retained to deduce the patterns of navigation for the subsequent query stages. This feedback information from the user is used to train the DCNN model regarding relevant and irrelevant images. The preprocessing phase results in the reduction of the computation complexity of the suggested framework. The combination of deep CNN with RF proves improved retrieval results as shown in the results section.

3 Related Contributions in the Area

Kameyama et al. [13] proposed a method by employing PSO for the optimization of the factors incorporated in the CBIR system's evaluation algorithm for relevance, by optimization in accordance to the results' suitability. Laaksonen et al. [16] used PicSOM, SOMs like a relevance method for CBIR. Suganthan et al. [23] executed indexing on a shape scheme through an SOM. The geometric shape structural information was extracted through pairwise relational attribute vectors. Two SOMs, SOM1 and SOM2, were implemented in the work. The relational attribute vector's global histograms present in SOM1 were given as an input to SOM2. The shape properties of the objects were defined by SOM1, whereas SOM2 created a topology-preserving mapping for the structural shapes. Vesanto et al. [28] employed the SOM toolbox as a quantization of vectors by arranging the example vectors on top of a standard framework in an arranged manner. Ma et al. [19] designed a mixed scheme for image clustering and retrieval, utilizing a support vector machine (SVM) and PSO. Xue et al. [30] presented two PSO-based multi-objective algorithms for selecting the features. The first was based upon the notion of noncontrolled arrangement in PSO to tackle troubles while selecting the features. The proposals for mutation, crowding, and domination to PSO in searching the Pareto front solutions were included within

the second one. Revaud et al. [21] learned a statistical form for the wrong detection distribution produced through an image matching algorithm. The datasets used for testing were the BelgaLogos and FlickrLogos datasets.

Chu and Lin [6] proposed the logo recognition and localization method by expressing the spatial relationships among the feature points. The outliers were filtered after searching the candidate regions through the mean-sift algorithm. The comparison of every region with the logo was done based upon visual patterns and visual word histograms. The visual pattern approach proved to be efficient for the description of the logos.

Boia et al. [3] implemented a logo recognition system invariant to scaling, colors, and illuminations coming from source lights of diverse intensities. The approach was based upon the BoW (bag-of-words) structure and scale invariant feature transform (SIFT) features. The improved accuracy in performance was achieved through the complete rank transform feature when tested using the FlickrLogos-32 database.

DCNNs, encompass subsampling and convolutional layers with nonlinear neural activations, subsequently following fully connected layers. The neural network is fed with an input image in the form of a three-dimensional tensor having equivalent dimensions with the input image and three color channels, i.e. RGB. These three dimensional learned filters are employed in every layer for convolution. Its output is transferred to the subsequent layer neurons to carry out nonlinear transformation through suitable functions for activation. The organization of the deep architecture transforms to single-dimensional signals and fully connected layers after subsampling and various convolution layers. These activations can be utilized as deep representations for clustering, classification, and retrieval purposes. Krizhevsky et al. [15] suggested AlexNet, a DCNN architecture which has five convolutional layers and three fully connected layers. Simonyan and Zisserman [24] developed ImageNet DCNNs into very deep networks which has 19 layers as VGG-19. GoogLeNet [27] with increased depth consisted of "inception" meta-layers including several convolution filter resolutions.

Donahue et al. [7] assessed how the features extracted through the deep convolutional network activations can be remodeled to new generic tasks, when given training in a fully supervised manner on a large preset of object recognition tasks. They examined and visualized the deep convolutional feature semantic clustering, concerning the various tasks such as domain adaptation, scene recognition, and challenges in fine-grained recognition. Erhan et al. [9] designed a scalable method for object detection using deep neural networks. The model was able to predict class-agnostic bounding boxes set with a single score for every box, matching its possibility of enclosing some object of concern and was also able to handle various instances for every class with the permissibility of the cross class generalization at the uppermost levels of the network. Zeiler and Fergus [31] introduced the visualization technique giving a view about the intermediate feature layer function and the classifier's operation. The contribution of different model layers in the performance was studied. The test datasets used were Caltech-101 and Caltech-256 datasets proving the significant results with the retraining of the softmax classifier.

Babenko et al. [4] and Wan et al. [29] examined the usage of the descriptors (neural codes) as an application to the image retrieval area. On the basis of the observations, the activations solicited by an image through the convolutional neural network's top layers give the high level descriptors for the image's visual content.

The purpose of DCNN application in the image retrieval area is to get the representations for features through a pretrained model. The images are input to the model, and the last layer activation values having a high-level semantic information are obtained. Babenko et al. [4] retrained the convolutional neural network on relevant image datasets and classes. Wan et al. [29] proposed the parameter refinement of a pretrained model with information on a class for improving the retrieval performance. Razavian et al. [22] utilized the CNN features with a spatial search for the image retrieval application.

Recent works surveyed in the area employed the deep learning approach for trademark retrieval. Bianco et al. and Eggert et al. [5, 8] evaluated the structure for retrieval of logos through pretrained CNNs and the data that was synthetically produced. Hoi et al. [11] utilized the deep learning mechanism for the recognition of the logos. The object-detecting method was experimented on through deep region-based convolutional networks. Iandola et al. [12] experimented on logo recognition through DCNNs. Various DCNN architectures were presented and examined for results using the FlickrLogos-32 database. Bao et al. [2] explored the appropriate

R-CNN's plan and situation for logo image recognition after examining the system using the FlickrLogos-32 database. Maria and Anastasios [18] utilized the DCNNs with RF for the retrieval of images. They utilized the feedback information from the user in RF to refine the deeper layer's feature representation of CNN. The pre-trained model was retrained for the relevant and irrelevant features from the feedback, resulting in improved retrieval results by taking into account the requirement of the user.

The effective techniques found for optimization and clustering are PSO and SOM, respectively, employed at the preprocessing phase. As discussed already, when only the DCNN technique is employed for image retrieval, it provides good retrieval results. The influence of DCNN is combined with query modification-based RF, at the retrieval phase in the proposed framework yielding good retrieval results.

4 Proposed Framework

The proposed framework is implemented by the subsequent steps:

- Feature vector database generation
- Feature set optimization
- Cluster creation through SOM
- Input query image
- Retrieving the relevant images based on similarity
- Taking the user's feedback (RF) for relevant/irrelevant images
- Using this feedback information to train DCNN
- Retrieving the refined results as final relevant images. The proposed framework is depicted in Figure 1.

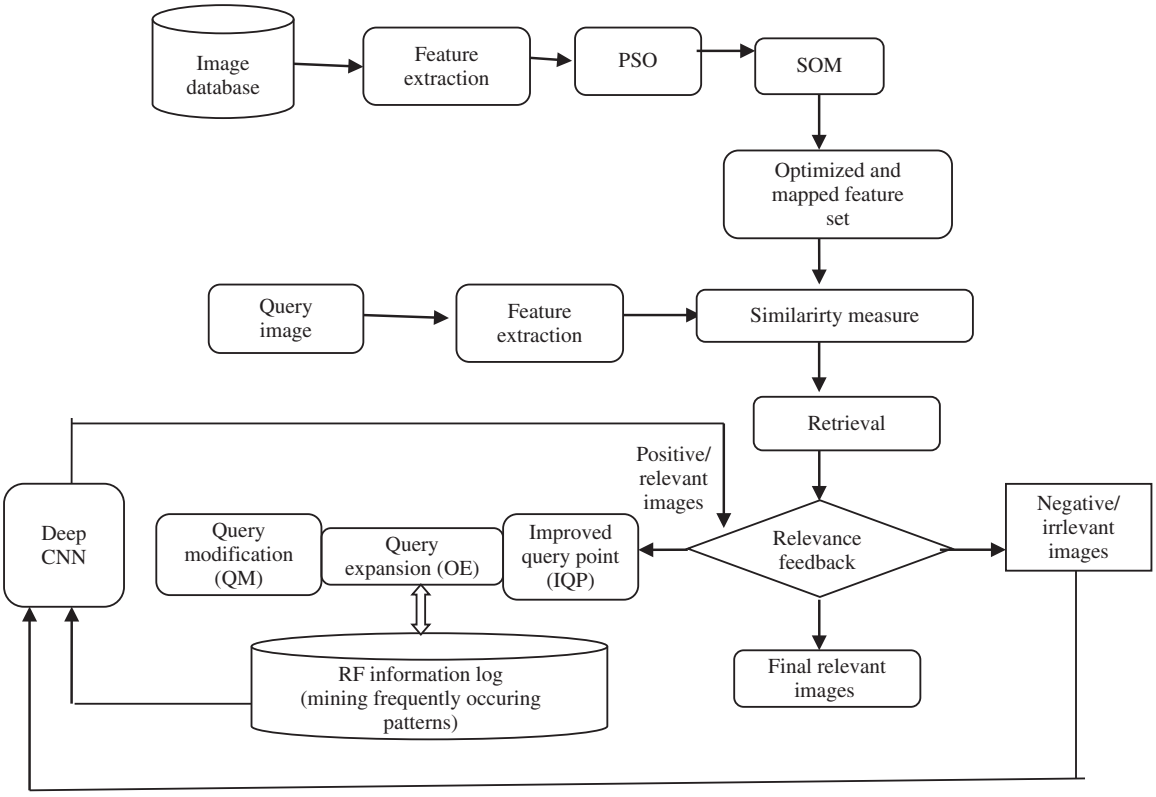

Figure 1: The Proposed Framework.

4.1 Feature Vector Database Generation

4.1.1 Datasets

FlickrLogos-27[1], FlickrLogos-32[1], and FlickrLogos-32 PLUS Datasets are used for evaluating the experiments. The details are depicted in Table 1. The openly accessible FlickrLogos-32 database includes images that are nontrademark images. The FlickrLogos-32 PLUS dataset is generated by eliminating these nontrademark images and including a number of trademark images.

Figure 2 shows example images of trademarks utilized in the implementation commencing with the FlickrLogos-32 PLUS database. The low-level features of an image such as shape, color, and texture, express the visual content of the image [10]. Shape, color, and texture feature extraction are implemented to retain the database feature's vector of the trademark images. Circularity and Fourier descriptor methods are executed for shape, color moments, color histogram, and color correlogram practices that are employed to express color, and the Haar wavelet and Gabor wavelet are used for texture.

4.1.2 Positive to Negative (P to N) Ratio

The percentage of example distribution within a class is called a class distribution. It may vary between 0% and 100%. Class distribution can be depicted through positive and negative rates. The percentage of positive examples within the dataset defines the positive rate, and the percentage of the negative examples defines the negative rate. The P to N ratio is the ratio of positive examples to negative examples present within the dataset [17]. This ratio can be calculated using the following equation:

$$\frac{\mathrm{P}}{\mathrm{N}} = \frac{\text{Number of positive class examples}}{\text{Number of negative class examples}} \tag{1}$$

The proposed FlickrLogos-27, FlickrLogos-32, and FlickrLogos-32 PLUS datasets have P to N ratios of 61:39, 58:42, and 70:30, respectively.

Table 1: Trademark Databases Used for Experimentation.

Database	Total no. of images	Trademark category
Publicly accessible: FlickrLogos-27 database[a]	1080	27
Publicly accessible: FlickrLogos-32 database[a]	8240	32
Generated: FlickrLogos-32 PLUS database	8550	38

[a]http://www.multimedia-computing.de/flickrlogos/data/

Figure 2: Example Trademarks From FlickrLogos-32 PLUS Database.

4.2 Feature Set Optimization

The optimized feature set is obtained by executing the PSO method, which is given to the SOM for training. The retrieval of image schemes includes various features of database depiction comprising unnecessary features not functional for the retrieval process. Hence, optimization of features is needed to eliminate these unnecessary features from the database features. The search room contains the particle's present coordinate, in the structure of optimization search. The best solution is acknowledged by retaining the trace of every particle's coordinates within the crisis area. This value is referred to as P_{best}. The top value attained by any particle among the entire particles in the area is known as Q_{best}. The k^{th} particle can be represented as $C_k = (C_{k1}, C_{k2}, \ldots\ldots\ldots\ldots C_{kn})$, and the velocity, i.e. the particle's rate of progress, is designated as $T_k = (T_{k1}, T_{k2} \ldots\ldots\ldots\ldots T_{kn})$ particle and alters its positions and velocity through the subsequent equations:

$$T_k(v+1) = T_k(v) + d_1 * a_1(P_{\text{best}} - C_k(v)) + d_2 * a_2(Q_{best} - C_k(v)) \tag{2}$$

$$C_k(v+1) = C_k(v) + C_k(v) \tag{3}$$

where

$T_k\,(v)$ and $T_k\,(v+1)$ denote the velocity of the k^{th} particle on the iteration v and $v+1$, respectively;

$C_k\,(v)$ and $C_k\,(v+1)$ denote the k^{th} particle position on the iteration v and $v+1$, respectively;

d_1 and d_2 depict the cognitive acceleration coefficient and social acceleration coefficient; and a_1 and a_2 being random number ranges between (0, 1).

PSO is a transformative computational method showing swarm intelligence and optimizing feature set; however, it does not ensure optimization [1]. Consequently, the clustering of these features is executed utilizing SOM. The RF is implemented using some clusters near the query, identified by averaging their features.

4.3 Cluster Creation Through the Self-Organizing Map

The planning of a dataset as an input, S^n, on top of a typical two-dimensional node array is defined by SOM. The parametric reference vector $n_k \in S^n$ is linked with each node k in the plan.

The proposed work projected the node array onto a rectangular pattern. The input is mapped by evaluating every input vector v against the $n_k{:}P$, and the top correspondent is determined. Euclidean distance is used for comparing each input vector $v \in S^n$ with the $n_k{:}P$. The winning node e is computed as follows:

$$\|v - n_e\| = \min\{\|v - n_k\|\} \tag{4}$$

Here, v is mapped through e, comparative to the component values n_k. Topographically closer nodes within the array exhibit learning through a similar input. The updation formula is:

$$n_k(t+1) = n_k(t) + h_{e,k}(t)[v(t) - n_k(t)] \tag{5}$$

where the discrete-time coordinate is t, and the region-defining function is $h_{e,k}$. The random values are used for the $n_k{:}P$ in the early phase [14].

4.4 Relevance Feedback

The query modification-based RF is used in the proposed work and integrated with the DCNN. Query modification is a hybrid through the improved query point (IQP), query expansion (QE), and query modification (QM) methods. The testing is done by executing 10 RF iterations. The positive image query points of every iteration are clustered by employing the k-means clustering algorithm. Each cluster is allotted a number. The feedback information from the user is maintained as log records in the following tables, and the patterns found repeatedly are mined in the next query.

I. Initial log table – includes the query image, relevant images, query point, and iteration number.
II. Query point table – includes the query image and query point.
III. Navigation action table – includes the query image, cluster number, and iteration number.
This table is created as follows:
i) Using the results of retrieval for each iteration, the clustering of query points is done through the *k*-means algorithm and allocated a number.
ii) The cluster number and iteration number are stored in this table.
iii) A pattern for every query is constructed using these clusters. For instance, for a query image, the retrieval result images exist as 1st iterations: cluster 3, 2nd iterations: cluster 2, and 3rd iterations: cluster 1; then, the pattern is N31, N22, and N13. The mining is done using these sequential patterns through the Apriori algorithm for mining.
IV. Log division table – encloses the query point and relevant image

The initial log table holds the information related to every feedback after each RF iteration. The navigation patterns are mined using the navigation action table. The relevant images are searched through the query point table and log division table. The sequential patterns obtained are used for constructing the navigation pattern tree. The sequential pattern is depicted through every branch of this tree. The query for the entire chronological patterns is exploited as the seed of the tree called the query kernel.

The following steps describe the search process:

1. The modified query point is obtained by averaging the features of images marked as positive.
2. The nearest query kernels are obtained to acquire the similar chronological patterns.
3. These similar trees of navigation pattern are utilized to get the neighboring leaf nodes.
4. The query points denoted by "t", which are relevant, are obtained from the collection of the nearest leaf nodes.
5. The "n" significant images are obtained.

The suggested method implements step 1 for generating the IQP. Steps "2–5" are executed for QE. The QM process is implemented by obtaining new feature weights utilizing the positive image features taken from each feedback.

4.4.1 Improved Query Point (IQP)

The old_{qpt} denotes the query point of the images in the earlier feedback. The positive image features I are averaged to get the new query point new_{qpt}. If $I = \{i_1, i_2, \ldots, i_k\}$ describes the positive images and the j^{th} feature's dimensions $D_j = \{d_1{}^N, d_2{}^N \ldots\ldots d_m{}^N\}$ taken through the N^{th} positive image. Therefore, the modified query point new_{qpt} designated by I is described as [26]:

$$\text{new}_{\text{qpt}} = \{\overline{D_1}, \overline{D_2}, \ldots\ldots, \overline{D_c}\}, \text{ where } 1 <= j <= c \tag{6}$$

$$\overline{D_j} = \{\overline{d_1}, \overline{d_2}, \ldots\ldots, \overline{d_m}\} \text{ and } \overline{d_t} = \frac{\sum^1 \leq n \leq s,\ d_t^n \in D_j d_t^n}{s}$$

4.4.2 Query Modification (QM)

A set of positive images $I = \{i_1, i_2, \ldots, i_k\}$ was obtained through the previous query point old_{qpt} during the former feedback. The modified weight for the k^{th} feature D_k is calculated as [26]:

$$N_k = \frac{\frac{\sum_{x=1}^{a} \alpha_x}{\alpha_k}}{\sum_{y=1}^{a} \frac{\sum_{x=1}^{a} \alpha_x}{\alpha_y}} \tag{7}$$

where

$$\alpha = \sum_{z=1}^{m} \frac{\sqrt{\sum_{i=1}^{b} \left(d_i^z - d_i^{\text{old}_{\text{qpt}}}\right)^2}}{b} \tag{8}$$

and $1 <= k <= a$

4.4.3 Query Expansion (QE)

The QE method is used to perform the weighted KNN search. The nearest query kernel to each of I is deduced, named as the positive query kernel, and the nearest query kernel for every negative image (NI) is named as the negative query kernel. Few query kernels may be common in a query kernel for positive and negative collections. Each kernel is allotted with a token pn.chk to deal with such a condition. pn.chk $= 0$ if the negative example number is greater than the positive example, or else, it will be equal to 1.

The relevant query pits are deduced. A set of similar leaf nodes is obtained through the navigation pattern tree. The relevant images are searched using the modified weights of the features as computed in equation (7). The search procedure is categorized into two stages: 1) generation of the relevant query points and 2) determination of images that are relevant.

The related query points are obtained through the query point table. Step 1 determines the "t" similar query points. These "t" query points are searched in the log division table to acquire the relevant images. The KNN search method is employed to retrieve the "n" images, which are nearer to $\text{new}_{\text{qpt.}}$

4.5 DCNN in Relevance Feedback

This paper proposes a CBIR framework for trademark images by integrating DCNN with an RF mechanism. The relevant images from the log records maintained in the above tables are given as input to the DCNN for training. The CaffeNet model [4] is utilized; an execution of the AlexNet model [15] is used for experimentation. The model includes eight neural network layers, which are trained; the initial five are convolutional, and the remaining three are fully connected. The max-pooling layers trail the 1st, 2nd, and 5th convolutional layers, whereas the ReLU nonlinearity [f(y) $=$ max(0; y)] is employed to each fully connected and convolutional layer, excluding the last one indicated by FC8. The seventh neural network layer representations indicated by FC7 are extracted through the CaffeNet model. The CaffeNet model parameters are used to initialize the entire layers of the latest model, excluding the FC8, substituted by a modified classification layer depicting the given dataset labels. There are no changes up to the convolutional layers, which are initially a fully connected layer indicated by FC6; FC7 and FC8 layers are updated using error backpropagation. The feature representations are extracted through the FC7 layer's activations having a feature dimension of 4096.

In the first RF iteration, the user gives feedback as relevant or irrelevant images to the system. This feedback information is utilized to train the DCNN model regarding the feature representation of relevant and irrelevant images. Subsequently, the feedback information in the next RF iterations is utilized in the framework, and the CNN model weights are updated, with the intention that relevant images move toward the query depiction, whereas the irrelevant ones deviates from the same. The representations acquired through a CNN model for a collection of images as input are modifiable by changing the model weights. The FC7 parameters are trained on positive/relevant and negative/ irrelevant images. The regression task during this training is executed through Euclidean loss. The query modification in RF is, thus, combined with DCNN for training upon relevant and irrelevant images.

Let the feature illustration for the query image that appeared in the FC7 layer is be $s \in P^{4096\times 1}$.

$Y^+ = \left\{ Y_p \in P^{4096\times 1}, p = 1, \ldots, M \right\}$ is the feature representation set of M images marked as relevant by the user in the feedback. $Y^- = \left\{ Y_q \in P^{4096\times 1}, q = 1, \ldots, N \right\}$ denotes the feature representation set of N irrelevant images marked by the user.

These relevant and irrelevant illustrations of the images are modified using retraining capabilities of the neural network. The following equations are utilized to deduce the modified target representations of the positive/relevant and negative/irrelevant images.

$$\min_{y_p \in y^+} \tau^+ = \min_{y_p \in y^+} \sum_{p=1}^{M} \|y_p - s\|_2^2 \tag{9}$$

$$\max_{y_q \in y^-} \tau^- = \max_{y_q \in y^-} \sum_{q=1}^{N} \|y_q - s\|_2^2 \tag{10}$$

The above optimization problems are solved through gradient descent. The objective function's first order gradients τ^+ and τ^- are calculated as follows:

$$\begin{aligned} \frac{\partial \tau^+}{\partial y_p} &= \frac{\partial}{\partial y_p}\left(\sum_{p=1}^{M} \|y_p - s\|_2^2\right) \\ &= \frac{\partial}{\partial y_p}\left((y_p - s)^\tau (y_p - s)\right) \\ &= 2(y_p - s) \end{aligned} \tag{11}$$

and

$$\begin{aligned} \frac{\partial \tau^-}{\partial y_q} &= \frac{\partial}{\partial y_q}\left(\sum_{q=1}^{N} \|y_q - s\|_2^2\right) \\ &= \frac{\partial}{\partial y_q}\left((y_q - s)^\tau (y_q - s)\right) \\ &= 2(y_q - s) \end{aligned} \tag{12}$$

As a result, the update rules for the n^{th} iteration are devised as:

$$y_p^{(m+1)} = y_p^{(m)} - 2\beta\left(y_p^{(m)} - s\right),\ y_p \in y^+ \tag{13}$$

$$y_q^{(m+1)} = y_q^{(m)} + 2\beta\left(y_q^{(m)} - s\right),\ y_q \in y^- \tag{14}$$

The preferred distance from the query representation is monitored by the parameter $\beta \in [0, 0.5]$. The above feature representations are used for the $n + 1$ iteration as targets within the FC7 layer. The neural network and the DCNN regression task is devised and given further training through backpropagation. Resulting in the generation of positive/relevant and negative/irrelevant images representations nearer to the targets $y^{(m+1)}$. Hence, the positive/relevant images come nearer to the query image within the FC7 layer, and the negative/irrelevant ones move far from the query. Accordingly, the RF feedback is incorporated by inputting the query and the feedback information of the images to the CNN input layer and getting the modified representations for FC7. This procedure is executed for every RF iteration; the CNN model is initialized with the preceding iteration's parameters and re-trained for the modified set of positive/relevant and negative/irrelevant images [18]. The above process is executed by the system from the start for every new query.

The algorithmic steps for the suggested framework can be given as follows:

Input: Database Images from FlickrLogos-32 PLUS dataset and Query Image

Output: The nine most relevant images against the given query are retrieved

Algorithm:
Begin
1. Implement the feature extraction techniques for the database images as follows and obtain the feature dataset.
  i) For color:
    a) color correlogram, b) color histogram, c) color moments
  ii) For texture:
    a) Haar wavelet, b) Gabor wavelet
  iii) For shape:
    a) circularity feature, b) Fourier descriptor
2. Apply PSO upon the dataset obtained in step 1 to get the optimized dataset of the features
3. Train SOM on these optimized dataset features to form the clusters
4. Give query image
5. Implement the feature extraction techniques for the query image similar to the database images as in step 1
6. Retrieve the top 20 relevant/similar images from the database based on the similarity/distance (e.g. Euclidean distance)
7. Now for every retrieved image in step 6, take feedback as relevant? Or not relevant image?
  If yes,
    Mark image as relevant/positive
  Else,
    Mark image as irrelevant/ negative
  End If
8. Obtain modified query point (new_{qpt}) and retain the feedback information in the log
9. Input the feedback information to DCNN for training upon relevant and not relevant images
10. The output of the algorithm is the top nine relevant images against the given query.
End

5 Experimentation Results

5.1 Implementation Setup

The dataset used for testing the framework are FlickrLogos-27, FlickrLogos-32, and FlickrLogos-32 PLUS database. The optimization of the database features from the overall 598 to the best features as 315,4615 to the selected number of best features as 2310 and 5034 number of features to the best 2519 number of features is done through the PSO algorithm for the FlickrLogos-27, FlickrLogos-32, and FlickrLogos-32 PLUS database, respectively. These optimized features are fed to the SOM for learning. The classes/clusters are formed by the SOM using these optimized features. The top three most relevant clusters are recognized by the system against a given query by averaging the image features within the cluster and the Euclidean distance between them. These three clusters are searched for relevant images.

First, 20 relevant/similar images are returned by the system based on similarity. For every query, 10 RF iterations are executed. For every RF iteration, user feedback is taken upon two images per feedback, as relevant or irrelevant. The CNN layers are initialized through the CaffeNet model parameters, except the FC8, which is replaced by a modified classification layer describing the dataset labels. The proposed work used 10 as relevant and 10 as irrelevant images to refine the model. The value of parameter β is set as 0.2, and the model training is done for one epoch for every RF iteration. The final output images retrieved are the most relevant nine images.

5.2 Analysis of Time Complexity

The PSO algorithm's time complexity is O (2nd $+ n^2 +$ 2nd); here, n represents the particle number, and the search space's dimension is d. For the SOM algorithm, it is given by O (s^2); where s denotes the input sample size. The RF algorithm possesses the time and memory cost linear to the dimension of the total features [25].

Therefore its time complexity is calculated as:

$$\sum_{a=1}^{m} O(\text{number of total images} * p_a) = O(\text{number of total images}) * P$$

Here, $P = \sum_{a=1}^{m} p_a$, where p_a represents the a$^{\text{th}}$ feature's dimension.

All the convolutional layers in the DCNN have time complexity as:

$$O\left(\sum_{i=1}^{n} m_{i-1}\, p_i^2 m_i n_i^2\right) \tag{15}$$

where i represents the convolutional layer's index, and n describes the depth, i.e. the convolutional layer number. The filter number is also recognized as the width in the i^{th} layer depicted as m_i. Whereas m_{i-1} represents the input channels number of the i^{th} layer. The spatial size, i.e. the filter's length is p_i. The spatial size of the output feature map is denoted by n_i. This time complexity is applicable at both times, i.e. testing and training, but through a different range. The training time required for every image is approximately 3 times (one for forward and two for backward propagation, respectively) of the time needed for testing per image. The above formula does not account the time cost pooling layers and fc layers; which acquires 5–10% of the computational time.

The overall time complexity of the suggested algorithm can be given as:

$$T = T1 + T2 + T3 + T4$$

where $T1$, $T2$, $T3$, and $T4$ are the time complexity of PSO, SOM, RF, and DCNN algorithm, respectively. The $T1$ and $T2$ cost are incurred at one time in the preprocessing stage, whereas the $T3$ and $T4$ costs are incurred at each query stage.

5.3 Performance Results in Terms of Retrieval

The final retrieved images against a given query using the influence of the DCNN are depicted in Tables 2–4, respectively, on the basis of the similarities of color, texture, and shape.

Table 2: Color Similarity-Based Output Retrieved Images.

Image given as query	Relevant output images				
					
					

Table 3: Texture Similarity-Based Output Retrieved Images.

Image given as query	Relevant output images				
					
					

Table 4: Shape Similarity-Based Output Retrieved Images.

Image given as query	Relevant output images				
					
					

Table 5: Confusion Matrix.

	Predicted positive	**Predicted negative**
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

Table 6: Retrieval Results for 10 Images Given as a Query.

S. no.	**Query image**	**Precision**	**Recall**	**Accuracy (%)**	**F_1-score**
1		0.975	0.922	93.58	0.947
2		0.974	0.948	93.82	0.960
3		0.967	0.918	93.86	0.941
4		0.980	0.940	94.59	0.959
5		0.964	0.946	94.77	0.954
6		0.980	0.942	93.71	0.96
7		0.940	0.917	92.97	0.928
8		0.979	0.932	94.81	0.954
9		0.982	0.948	94.76	0.964
10		0.988	0.979	94.54	0.983

Table 7: Similarity (%) of Color, Texture, and Shape for the Topmost Nine Relevant Output Images.

S. no.	Relevant output image	Similarity (%) of color	Similarity (%) of texture	Similarity (%) of shape
1		89.91	87.98	92.11
2		92.284	91.033	93.654
3		94.605	93.126	95.12
4		98.769	97.012	98.561
5		95.721	96.117	87.274
6		78.214	85.813	88.891
7		87.661	82.929	85.721
8		78.939	84.862	74.632
9		75.127	84.321	68.519

Table 8: Suggested Framework's Comparison with Su et al. [26].

S. no.	Methodology	Database	Parameter	
			Precision	Recall
1	Su et al. [26]	Seven classes of different categories of 200 images each (1400 images)	0.910	–
2	Framework by employing just RF	FlickrLogos-27 database	0.905	0.721
		FlickrLogos-32 database	0.921	0.768
		FlickrLogos-32 PLUS database	0.935	0.898
3	Suggested framework (by employing PSO + SOM + DCNN in RF)	FlickrLogos-27 database	0.967	0.865
		FlickrLogos-32 database	0.958	0.887
		FlickrLogos-32 PLUS database	0.973	0.939

Table 9: Comparison of the Approaches of Iandola et al. [12] and Bao et al. [2] with the Suggested Framework.

S. no.	Methodology	Database	Parameter	
			Mean average precision	Accuracy
1	Iandola et al. [12] a) FRCN + AlexNet b) FRCN + VGG16	FlickrLogos-32 database	a) 73.5% b) 74.4%	89.6% (maximum with GoogLeNet-GP architecture)
2	Bao et al. [2]	FlickrLogos-32 database	84.2%	—
3	Suggested framework (employing PSO + SOM + DCNN in RF)	FlickrLogos-27 database FlickrLogos-32 database FlickrLogos-32 PLUS database	96.71 % 95.84 % 97.36 %	91.04 % 91.23 % 94.14 %

The metrics used for the evaluation of the framework are accuracy, precision, recall, and F_1-score calculated as below:

1. Accuracy $= (TP + TN)/(TP + TN + FP + FN)$
2. Precision $= TP/(TP + FP)$
3. Recall $= TP/(TP + FN)$
4. $F_1 - \text{Score} = 2\frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$

Here,

TP = True positive: positive example and prediction positive
TN = True negative: negative example and prediction negative
FP = False positive: negative example but prediction positive
FN = False negative: positive example but prediction negative

The confusion matrix, as shown in Table 5, maintains the examples from every class identified as correct and incorrect.

For example, considering every RF iteration, with output retrieval of 20 images, the TP, TN, FP, and FN calculation gives the count as 8, 8, 2, and 2 correspondingly.

Accuracy calculation is given by:

$$\begin{aligned}\text{Accuracy} &= TP + TN/(TP + TN + FP + FN)\\ &= (8 + 8)/(8 + 8 + 2 + 2)\\ &= 16/20 = 0.8 = 80\%\end{aligned}$$

The retrieval results for 10 example images as queries are reported in Table 6. Table 7 describes the similarity (%) for color, texture, and shape for the topmost output images retrieved. In Tables 8 and 9, the suggested framework is compared with that of Su et al. [26] and the recent deep learning-based approaches proposed by Iandola et al. [12] and Bao et al. [2]. Su et al. [26] proposed the relevance feedback, based upon the mining of the navigation patterns and a query modification scheme. Iandola et al. [12] and Bao et al. [2] experimented on the DCNN using the FlickrLogos-32 database for the retrieval of the logo images. The proposed scheme integrated preprocessing (PSO and SOM) with the query modification and mining-based RF approach and incorporated the DCNN within RF, experimented using the FlickrLogos-27, FlickrLogos-32, and FlickrLogos-32 PLUS databases. The results are promising as shown in the comparison in Tables 8 and 9.

Table 8 depicts the improved retrieval results of the method when the DCNN is integrated with the RF. The optimization (PSO) and clustering techniques (SOM) applied earlier in the preprocessing phase reduced the space for searching and, therefore, the complexity.

The comparison in Table 9 proves the improved retrieval results with the integration of the DCNN in the RF when compared with the other two only deep learning-based models.

6 Conclusion

In the proposed trademark retrieval system, the DCNN model is combined with the query modification-based RF technique at the retrieval phase. In the first iteration of the RF, the feedback information from the user is given as input to the deep CNN and trained upon the relevant and irrelevant feature representations. In the successive RF iterations, these feature representations are refined in the CNN model. This improved the retrieval performance because of the knowledge of the user's information need and bringing the relevant images nearer to the query and moving the irrelevant images far from it. The preprocessing phase reduced the search space, thereby, the time needed for execution significantly due to the optimization (PSO) and clustering (SOM) techniques employed. Future research work is directed on replacing the SOM algorithm used for clustering at the preprocessing phase by DCNN.

Bibliography

[1] Q. Bai, Analysis of particle swarm optimization algorithm, *Comput. Inf. Sci.* **3** (2010), 180–184.
[2] Y. Bao, H. Li, X. Fan, R. Liu and Q. Jia, Region-based CNN for logo detection, in: *ICIMCS*, August 19–21, 2016, Xian, China, ACM. ISBN 978-1-4503-4850-8/16/08, DOI: http://dx.doi.org/10.1145/3007669.3007728.
[3] R. Boia, A. Bandrabur and C. Florea, Local description using multi-scale complete rank transform for improved logo recognition, in: *IEEE International Conference on Communications*, Sydney, Australia, 2014, pp. 1–4.
[4] A. Babenko, A. Slesarev, A. Chigorin and V. Lempitsky, Neural codes for image retrieval, in: *Computer Vision*, ECCV Springer, Zurich, Switzerland, September 6–7 and 12, 2014, pp. 584–599.
[5] S. Bianco, M. Buzzelli, D. Mazzini and R. Schettini, Logo recognition using CNN features, in: *Image Analysis and Processing ICIAP 2015*, Springer, Via Garibaldi n. 5, 16124 Genova, 2015, pp. 438–448.
[6] W.-T. Chu and T.-C. Lin, Logo recognition and localization in real-world images by using visual patterns, in: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, 2012, pp. 973–976.
[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, (2013), arXiv preprint arXiv: 1310.1531.
[8] C. Eggert, A. Winschel and R. Lienhart, On the benefit of synthetic data for company logo detection, in: *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, ACM, Brisbane, Queensland, Australia from 26–30 October 2015. pp. 1283–1286.
[9] D. Erhan, C. Szegedy, A. Toshev and D. Anguelov, Scalable object detection using deep neural networks, in: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, June 24–27, 2014, pp. 1–8.
[10] L. Fuhui, H. Zhang and D. D. Feng, *Multimedia information retrieval and management*. Chapter-1 Fundamentals of Content-Based Image Retrieval, pp. 1–26, Springer, Berlin Heidelberg, 2003.
[11] S. C. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue and Q. Wu, Logo-net; Large-scale deep logo detection and brand recognition with deep region-based convolutional networks, (2015), arXiv preprint arXiv: 1511.02462.
[12] F. N. Iandola, A. Shen, P. Gao and K. Keutzer, DeepLogo: hitting logo recognition with the deep neural network hammer, (2015), arXiv preprint arXiv: 1510.02131.
[13] K. Kameyama, N. Oka and K. Toraichi, Optimal parameter selection in image similarity evaluation algorithms using particle swarm optimization evolutionary computation, IEEE International Conference on Evolutionary Computation, 16–21 July 2006, Vancouver, BC, Canada, ISBN: 0-7803-9487-9.
[14] T. Kohonen, *Self organizing maps*, 2nd ed, Springer-Verlag, New York, 2014.
[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: *Advances in Neural Information Processing Systems*, Tahoe Nevada, December 03–08, 2012, pp. 1097–1105.
[16] J. Laaksonen, M. Koskela, S. Laakso and E. Oja, Self-organising maps as a relevance feedback technique in content-based image retrieval, *Pattern Anal. Appl.* **2** (2001), 140–152.
[17] N. Limsetto and K. Waiyamai K, Integrating weight with ensemble to handle changes in class distribution, in: *10th International Conference on machine learning and data mining in pattern recognition*, MLDM 2014, Russia, (July 21–24, 2014), 2014, pp. 91–106.
[18] T. Maria and T. Anastasios, Relevance feedback in deep convolutional neural networks for content based image retrieval. in: *Proceedings of the 9th Hellenic Conference on Artificial Intelligence*, SETN, May 18–20, 2016, Thessaloniki, Greece, ACM, ISBN 978-1-4503-3734-2/16/05, DOI: http://dx.doi.org/10.1145/2903220.2903240.
[19] L. Ma, L. Lin and M. Gen. A PSO-SVM approach for image retrieval and clustering, in: *41st International Conference on Computers and Industrial Engineering*, Los Angeles, California, USA, 23–25 October, 2011, ISBN: 978-1-62748-683-5.

[20] L. Pinjarkar, M. Sharma and K. Mehta, Comparative evaluation of image retrieval algorithms using relevance feedback and its applications, *Int. J. Comput. Appl. (IJCA)* **48** (2012), 12–16.
[21] J. Revaud, M. Douze and C. Schmid, Correlation-based burstiness for logo retrieval, in: *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12 ACM Multimedia Conference Nara, Japan, October 29–November 2, 2012. pp. 965–968.
[22] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson S. CNN features of-the-shelf: an astounding baseline for recognition, in: *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2014, pp. 512–519.
[23] P. N. Suganthan, Shape indexing using self- organizing maps, *IEEE Trans. Neural Netw.* **13** (2002), 835–840.
[24] K. Simonyan and A. Zisserman, Very deep convolutional networks for large scale image recognition, (2014), arXiv:1409.1556, abs/1409.1556.
[25] Z. Su, H. Zhang, S. Li and S. Ma, Relevance feedback in content-based image retrieval: bayesian framework, feature subspaces, and progressive learning, *IEEE Trans. Image Process.* **12** (2003), 924–937.
[26] J.-H. Su, W.-J. Huang, P. S. Yu and V. S. Tseng, Efficient relevance feedback for content-based image retrieval by mining user navigation patterns, *IEEE Trans. Knowl. Data Eng.* **23** (2011), 360–372.
[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, (2014), arXiv:1409.4842.
[28] J. Vesanto, J. Himberg, E. Alhoniemi and J. Parhankangas, Self-organizing map in Matlab: the SOM Toolbox, in: *Proceedings of the MATLAB DSP Conference*, Espoo, Finland, pp. 35–40, Nov 16–17, 1999.
[29] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang and J. Li, Deep learning for content-based image retrieval: a comprehensive study, in: *Proceedings of the ACM International Conference on Multimedia*, MM '14 2014 ACM Multimedia Conference Orlando, FL, USA, November 3–7, 2014, pp. 157–166.
[30] B. Xue, M. Zhang and W. N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, *IEEE Trans. Cybern.* **43** (2012), 1–16.
[31] M. D. Zeiler and R. Fergus, Visualizing and understanding convolutional networks, in: *European Conference on Computer Vision (ECCV) Zurich*, Switzerland, September 6–12, 2014, pp. 818–833.