Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 718–731

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

A Multi-modal CBIR Framework with Image Segregation using Autoencoders and Deep Learning-based Pseudo-labeling

Manu John[a,*], Terry Jacob Mathew[a,b], Bindu V R[a]

[a]*School of Computer Sciences, Mahatma Gandhi University, Kottayam, Kerala, India*
[b]*MACFAST, Tiruvalla, India*

Abstract

Various modulated techniques of Content-Based Image Retrieval (CBIR) using deep learning provide better search outputs even though they are computationally challenging. These methods can be enhanced further, if the search key can be tagged effectively and directed towards target images. In this paper, we have developed a new multilevel aggregation technique along with autoencoders, to be implemented on image features for precise feature selection and accurate search output. Locally significant datasets and generic ones are dealt separately for obtaining better hit rates along with the process of query expansion. The concept of pseudo-labelling by deep learning is introduced to classify images into positive and negative classes, where positive class consists of the images that are similar to the query image. The feature spaces of query images are compared with the images in the search pool based on their assigned weights. The target images are finally ranked and selected based on an adaptive threshold level. The methodology for this enhanced CBIR technique is tested on public datasets and its results are collated with recent approaches proposed in the literature. The proposed method has attained significant improvements in precision, recall and computation and thereby the use of images from connected devices has been proven to be significant. Given the importance of more accurate image retrieval in CBIR, this method can be experimented for other similar applications as well.

Keywords: Deep learning; CBIR; review; image retrieval

1. Introduction

Rapid proliferation of digital devices and social media has given way to an unprecedented growth in storage systems. It has also become necessary to gather stored content from storage systems based on demand. This phenomenal growth of cyberspace has triggered a series of research activities for managing digital data profiles. Researchers are inclined to explore effective solutions for image retrieval without the aid of any textual annotation. This has resulted in

* Corresponding author.
E-mail address: manumjohn@gmail.com

1877-0509
10.1016/j.procs.2023.01.052

the development of a popular mechanism, known as Content Based Image Retrieval (CBIR). CBIR is the application of computer vision methods that retrieves digital images relevant to a query image on the basis of its image contents [16]. This technique does not depend on tags or text attributes associated with images, and are selected by similarity.

Text-based image retrieval (TBIR) is a primitive method, which compares the similarity of textual information present in images for retrieval. For this reason, TBIR search cannot be applied for images without textual descriptors. CBIR, on the other hand considers generic image features and is far superior to TBIR systems. CBIR techniques are used in search engines, medical applications, tourism industry, etc. [11, 17]. The input query in CBIR can be classified into query by text, query by image, query by sketch and query by concept. A pictorial representation of the different CBIR methods is shown in Fig. 1.

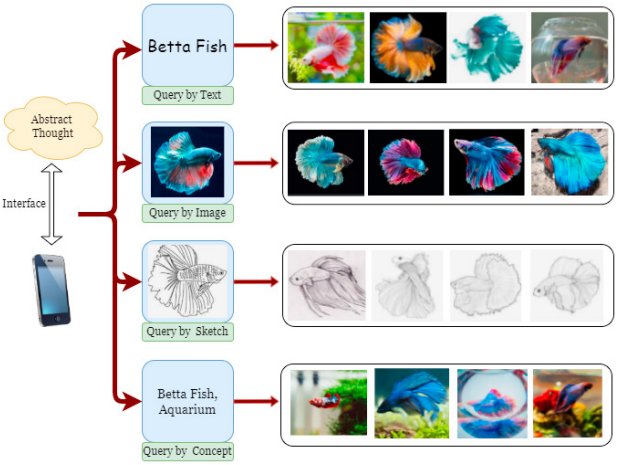

Fig. 1: The representation of different methods of image retrieval.

1.1. Framework of CBIR

A typical CBIR system has two basic modules - feature extraction and similarity measurement. In feature extraction, the image features are extracted and consequently stored as a feature vector for future reference. In similarity measurement, the query image is compared with other images in the repository based on image features. Similarity measurement techniques are utilized in most cases to select similar images and are finally presented to the end user. A two-stage process is required for any CBIR framework to obtain quick and accurate image retrieval [29]. A typical CBIR methodology is shown in Fig. 2.

Feature extraction is the method of generating value from the region of interest by summarizing pixel information and thereby the important aspects of an image can be extracted and represented by fewer, but significant information. This activity is crucial to the system development as its efficacy is directly dependent on the extracted features. The most common markers used in CBIR applications are colour, texture, and shape [13]. It is customary to use multiple features for developing efficient CBIR systems. The global features of an image include image descriptors such as color, shape, and texture components. Researchers have primarily focused on the color feature as it is the most appealing among all other features. Colour is related to the spectral information present in an image, while texture refers to the variation of intensity in the image. The shape features are capable of describing the shape of a selected target area in an image [10]. The process of feature extraction performed at the level of an image produces global features, while the extraction at pixel level generates local features [30]. Thus, it is obvious that these low level features that are prominently evident from an image are categorized as global and local.

1.2. Challenges of CBIR

A picture is worth more than a thousand words as images can convey its meaning more effectively than a mere verbal description. As humans are capable of analysing and inferring subtle information from images, it is better

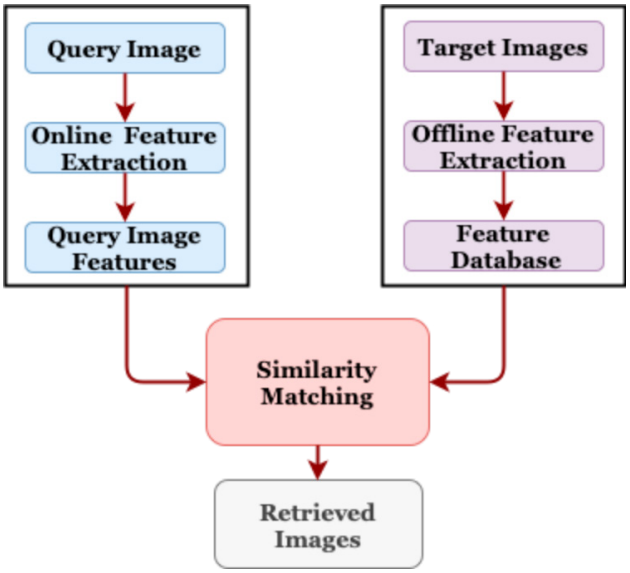

Fig. 2: The framework of a typical CBIR system.

to convey complex and multifaceted ideas through images than text. But, dependence of machines on numerical comparisons for analysis often leads to a major semantic gap between human and mechanical perception of an image. As a result, human analysis produces superior results than machine analysis. This understanding prompted researchers to incorporate semantic image retrieval with the aid of machine learning [33]. Retrieval of images from a large pool is challenging due to several reasons. As images of today possess high pixel resolutions, it is quite natural to consider low dimensional features for comparison, rather than the direct images. Traditional methods work well with features such as shape and colour, but fail miserably in identifying semantic meanings within an image. The failure is due to the absence of techniques to handle the perceptions of human mind [33]. Another reason for the failure is the non-adaptive nature of techniques, which reduces the extraction of generic image characteristics. These drawbacks affect the semantic processing of images, even with the mapping of other features.

The gap that exists between the retrieved data and the real life meaning of the image is considered as the semantic gap. The nature of database is a determining factor for reducing the semantic gap. If the elements of the database are strictly restricted to a particular domain, then the semantic gap can be minimized. Limited domain databases contain images of lesser detail and granularity, and are detected with ease. However, generic databases attribute to many dimensions and hence relevant images are not easily detectable. In addition to this, the presence of high resolution images under different themes, background, context etc. poses a challenge for image retrieval. Smeulders et al. [27] suggested the inclusion of details regarding the image from external sources for better results. Most of the relevant applications of CBIR use generic databases and hence there is a continuous effort by researchers to reduce the semantic gap [33].

1.3. ML and CBIR

The performance of new vision based algorithms has enabled the representation of low dimensional images with more precision. CBIR systems have benefited from machine learning by adopting the recent developments for noise removal, feature vector extraction etc. An optimal feature vector should be capable of representing the properties of images such that it can be used for improving the efficiency of image retrieval. This quality of feature vector being crucial to the success of CBIR systems, utmost care and attention is taken for developing reliable feature descriptors. Some of the popular feature descriptors are HOG, BRISK, MSER, SIFT, SURF etc. [7, 19, 3, 21]. With a host of reasons for poor performance [26], new avenues of decision making [35, 34] can boost the overall performance.

1.4. Introduction to DL

The best way to extract relevant images by machine learning is to provide source data that are represented by lower dimensional feature vectors. But these shallow retrieval models often fail to meet the expected retrieval rates due to fewer number of training samples, lack of crispness in feature space, and class imbalance problems. But, studies have shown that deep learning algorithms [18] perform better than the shallow models in generic image retrieval [31, 20, 32, 22]. Deep learning approaches work with large volume datasets on GPU's having phenomenal computational power and affordability. The adaptability of the algorithms and the ability to represent generic features are the main reasons for the success of deep learning approaches. The architecture of Deep Convolutional Neural Networks (CNN) [15] consists of a host of convolutional and sub-sampling layers supported by activation functions and fully connected layers. In reality, the input images are fed into CNN as a three dimensional vector with matching image dimensions viz., height, width and depth. After processing, the CNN architecture changes to a fully connected network with a single dimension. The final architecture is commonly employed for CBIR systems. The CNN models can be customised by input and the intervals of classification can be adjusted for better performance when training is done with massive volumes of data. For better image retrieval in CBIR systems, autoencoders are also being introduced in recent studies along with the machine learning approach.

As there is a need to improve the efficiency of the CBIR search techniques, this paper proposes a novel blend of traditional and advanced methods. The proposed CBIR system utilises traditional unsupervised machine learning techniques such as clustering, autoencoders and supervised CNN approaches to explore and filter the image contexts from query image and database repository in multiple levels. The remaining sections of the paper are organized as follows. Section 2 gives a detailed review of the state-of-the-art approaches that use techniques for enhanced retrieval rates. The proposed methodology is explained in Section 3. The results and discussions are detailed in Section 4. Finally, in Section 5 we conclude the paper.

2. Literature Survey

Query by Image Content (QBIC) analysis [9] is one of the best first generation CBIR approaches that uses image textures, color palates, local objects, etc., for retrieving images and video frames. The simplicity of the system is advantageous but it has inherent limitations due to the use of direct pixel based similarity comparison. This model may not work efficiently on slightly translated or rotated images. Due to these drawbacks research has later shifted towards feature descriptor-based analysis, which is evaluated by hand-crafted low-dimensional descriptors. As a result, a set of robust feature extraction schemes and machine learning techniques were developed to aid CBIR techniques. The main advantages of such low dimension feature comparisons include less error rate, less computation cost, and robustness against geometrical distortion such as rotation and translation.

In [38], Yuan et al. proposed a local descriptor-based CBIR technique using Local Binary Pattern [12] and Scale Invariant Feature Transform (SIFT). This method extracted key points, which was used as the feature space for comparison. The local features are compared with the target images and are clustered together for image grouping. The similarity search is then applied over the clustered feature vectors to identify similar images. In a modified approach [37] to this model, the feature space is expanded by adding Histogram of Oriented Gradients (HOG) features. This inclusion helps to improve the performance while processing with geometrically distorted image samples. The main disadvantage of these models is their higher sensitivity to noise and higher computation time while processing large feature descriptors such as HOG.

In [23], a CBIR algorithm is proposed along with color fusion and texture descriptors. The introduction of Laplacian score reduced the dimensions of the feature vector. Another CBIR approach proposed by Bibi et al. [4] used a set of sparse complementary features for robust representation and selection. Their classification approach is based on locality-preserving projection, fuzzy c-means clustering, and soft-labelled support vector machines. They introduced complementary features on a larger size codebook generated using miniature ones, thereby improving both the overall recall and precision.

ElAlami et al. [8] proposed a solution to reduce the computation cost while working with larger feature representations by adopting a feature optimization technique along with a genetic algorithm. This approach helps to improve the retrieval precision by eliminating irrelevant features during similarity assessment. Another image retrieval approach

by Chum et al. [6] uses query image expansion and a feature known as *bag of words* from suitable locations in the image. This method supports searching with salient image features and improves the retrieval performance over the whole image analysis.

A modified query expansion approach is reported in [5], which relies on a feature filtering correlation process for image retrieval. This approach is beneficial for reducing ambiguity in image search and retrieval. An advanced mobile image retrieval scheme using SIFT features and a query expansion technique is presented in Yang et al. [36], which examines similar images from local device memory. The query object selection is dependent on the details of locations and timestamps of the images saved in the local memory. The whole process is dependent on key point extraction, but such a similarity assessment often leads to wrong outputs with noisy images. Saliency preservation is another challenge while attempting such a key point-based similarity analysis.

In [10], Garg et al. presented a CBIR technique that highlights generation and compression across multiple features. This approach used a multi-level image decomposition by extracting approximation and correct coefficients. This is done after applying discrete wavelet transformation to individual color channels. The structural formation is generated from local binary patterns and its magnitude is extracted for extra discriminating power. Further, GLCM method is utilized by the available dominant patterns to obtain statistical inputs for texture classification. Similar multilevel pipelines [25, 28] harness the power of CNN architecture to improve accuracy in medical applications.

An efficient CBIR model that utilizes the associated system memory was proposed by Aiswarya et al. [2]. Their retrieval approach used several layers of optimizations to reduce the inherent drawbacks of the key point descriptor-based search process. PSO-SIFT key points are used as the feature descriptor, while query expansion is used for maintaining image saliency. The feature space is further optimized using dimensionality reduction to eliminate outliers. However, the use of feature descriptors does not produce reliable performance in all types of images. On a general note, most of the performance flaws are due to the use of static feature extraction algorithms for image comparison. It is also found that feature representation cannot provide stable performance for all scenarios. A dynamic method that extracts features in accordance with the type and nature of image samples can be used to overcome this limitation. Another concern is the computation time required for the similarity check in a big dataset, especially while working in a mobile device with limited hardware specification. This computational overhead can be relieved by limiting the number of reference images used for similarity searching. Another updated CBIR approach was presented in [1] that uses an autoencoder stage for feature extraction and adaptive selection of relevant features from the target dataset. This algorithm also makes use of a query expansion method to preserve visual saliency. Even though this model delivers decent retrieval accuracy, the processing time and cost is relatively high. To improve upon the existing techniques, we propose a multi-level CBIR technique.

The proposed scheme takes into account the existing issues present in the aforementioned approaches and proposes a similarity search that uses fast retrieval based on the features that are extracted from a deep learning image classification model. The model uses clustering over the target space for reducing the number of reference images and employs a CNN based image classification scheme to select appropriate clusters for similarity assessment. The detailed explanation of the proposed algorithm follows in Section 3.

3. Methodology

The proposed process of image retrieval consists of splitting the input data into a global, local and query image dataset. Global dataset refers to the real world image data from which the most relevant and similar ones have to be filtered. This includes images in cloud storages, internet repositories, memory cards, external hard drives etc. Local dataset consists of the image resources which are available locally in the immediate proximity or within the existing system itself, from which a set of significant and relevant images can be shortlisted. Data augmentation can be done on the local dataset, if it is small in size. Query image is the input data that is used as the key for searching similar images from the master dataset.

In order to perform an ideal classification, a labelled data set is required. But in CBIR, the lack of such labelled data attributes poses a challenge for the image retrieval process. Hence, CBIR image search cannot make use of any pre-trained image recognition applications. In a theoretical setting of a machine learning model, CBIR acknowledges the need to have a class of dissimilarity and similarity mapped objects to carry on with the process of training the model. To overcome this drawback and to improve the accuracy of the image search, we developed a method known

as pseudolabelling, wherein the images are categorically labelled as positive or negative according to their similarity in a better way.

The proposed methodology is simulated with global, local and query datasets. After preparing these datasets, the proposed methodology initiates the process of pseudolabelling the local dataset. We rely upon autoencoders to derive precise image features as they are proven from experiments. The autoencoder features of both query image and local data set are extracted for the pseudolabelling process. The proposed CBIR method follows an adaptive approach for

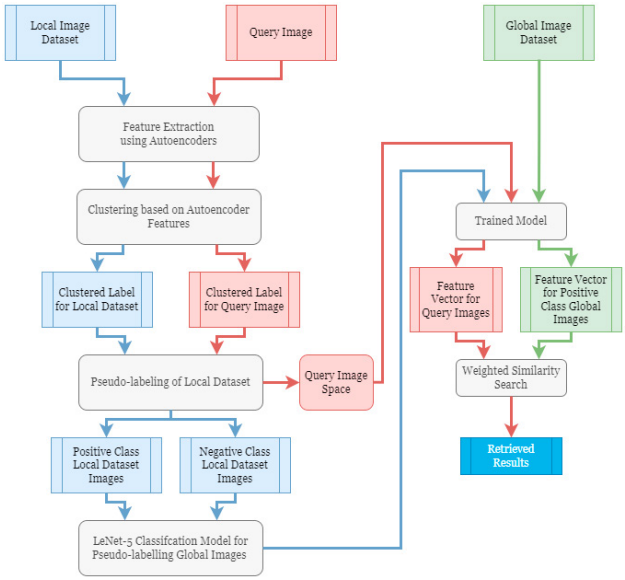

Fig. 3: Block diagram of the proposed CBIR system.

deriving feature elements from the image search pool. A descriptive block diagram of the proposed CBIR scheme is shown in Fig. 3. The stage-wise details are given in the following sections.

3.1. Initialize input data

The query image and target database are the two basic units of any CBIR system. The query image is given as the search input to the target dataset from which similar images are retrieved. In this multilevel aggregation method, we use an expanded query set, a local image dataset (present within the device memory) and the repository of images known as global image dataset (online or offline target dataset). Inclusion of these different types of datasets can be crucial for retrieval systems that involve mobile devices. Accordingly, the process of image retrieval is made more precise by including the data present in the local memory and connected memory of devices. The offline memory consists of the storage found in the devices and accessories under consideration, while the online memory refers to any linked storage server. The local images are locally proximal to the query image whereas the global images are located in external offline or online datasets.

In order to simulate the retreival process, the complete set of available images are randomly divided into three groups namely, test, local and global. A total of 10% of the dataset is set aside for the test data, from where query images will be taken for the evaluation; 30% for the local dataset and the rest 60% is allocated to the master search pool. As the dataset consists of various classes of images, steps are taken to include even representation of images in all groups. Since the ground truth information is not available for local, global or query images, a pseudolabelling stage is included to group the images approximately in the input datasets. The immediate purpose of pseudolabelling is to bring out additional query images from the local dataset. This facilitates a swift search mechanism by considering a small subset of target images during the final image retrieval phase.

3.2. Initial clustering

To create the pseudolabels in the local dataset, we applied k-means clustering on the local dataset images and the query image based on features extracted from an autoencoder. An autoencoder is an iterative learning process to define a set of low dimensional latent space, which can reconstruct the images with minimal reconstruction error. Essentially it consists of an encoder (that creates a feature vector) and a decoder (that reconstructs the image based on this feature vector). The encoding and decoding functions are fine tuned in an iterative manner for a set of images. Using the trained autoencoder module, feature vectors for each image can be generated. Here, we used all the local images along with the query images to train the autoencoder. ie. If there are n images in the local data set, a total of $(n + 1)$ images will be present in the pool. A moderate feature vector size of 250 was selected to obtain a generalized feature representation. As the image features are computed, these autoencoder features are used for the clustering operation. Since the primary aim of this stage is to list query like images in the local set, we clustered the images into multiple cluster labels.

Following the clustering phase, the feature vector of the local images are classified into positive and negative classes. Assuming a set of n images, the feature space of the local data set (obtained from the autoencoder) is of size $n \times 250$. We also added the query image feature vector along with it to make it $(n + 1) \times 250$. The method proceeds by performing k-means clustering with $k = 2$. This clustering gives an output label corresponding to each $n + 1$ feature vector. Each feature vector will be tied up with a cluster label and this forms the output of the initial clustering stage. The $(n + 1)^{\text{th}}$ image represents the query image and the corresponding cluster obtained for this is very significant. The cluster obtained for $(n + 1)^{\text{th}}$ image is taken as the seed point for dividing the images into positive and negative datasets. The images which fall under the same cluster as the query image are set aside as positive class and the others into negative class. This defines the first level of aggregation. These positive and negative class labelling refers to the pseudolabelling of the local dataset images and all the subsequent stages are dependent on these pseudolabels.

3.3. Creation of query image space

A query image space is an extended set of images appended along with the original query. In this stage, we find out the most similar N images from the reduced positive class of local dataset images to create a 'query image space' from a maximum of N images. A similarity threshold is set in this comparison for choosing the similar images. The search query is augmented along with a few selected local data to create the query image space. This method creates a pseudo-labelled local dataset and a query image space apart from the global repository for CBIR search. The final query set Q is a set of images, $\{q_1, q_2, \ldots q_N\}$, where q_1 is the actual query image, q_2 is the most similar image from the local dataset, q_3 is the next most similar image etc. So, the similarity of query space images is of the order as $q_1 > q_2 > q_3 > q_4 > \ldots > q_N$.

3.4. Pseudo labelling of global dataset

In this stage, we estimate the pseudolabels for each target image from the global dataset. The detailed steps are as follows.

3.4.1. Training a CNN classification model

Since pre-defined labels are not available for global dataset images, a deep learning classification model is created for prediction in the global dataset. We use a CNN architecture for training with the help of the local dataset images and their already generated pseudo-labels. In the experiments, we used a customized LeNet-5 convolution neural network model for image classification. Since, the local images are less in number the local dataset is augmented to a sufficiently higher number to train the model without underfitting. The CNN network is deployed with three convolution layers (with kernel size 5×5), two max-pooling, and one fully convolutional layer. The first convolution layer uses six filters, the next one uses 16 each, and the final convolution layer uses 120 each to extract complex features. All images are resized to 64×48 color images for a better trade-off between accuracy and computational overhead. Hence the input layer are fed with images of $64 \times 48 \times 3$ dimension. The model uses batch normalization and dropouts along with ReLU activation. Addition of two max-pooling layers after the first and second convolutional layers restricts the

feature space. The model also uses Adam optimizer with a learning rate of 0.001, which is determined empirically after trials. The final classification layer uses SoftMax activation with the categorical cross-entropy loss function.

3.4.2. Predicting global image labels from the trained model

After obtaining the trained model, all the images in the global dataset are passed on to the trained CNN model to predict each global image into either positive (images that are similar to the query image) or negative class (images with different class labels as that of query image). This pseudolabelling helps to restrict the global images in the subsequent stages. Eg: if there are 1000 global images and if 400 are positive and 600 negative; we take only the 400 positive images for further analysis. By this aggregation the final search will be restricted to the positive global images and the query image space obtained from the previous stage.

3.4.3. Generating feature vectors for final retrieval process

Once the positive class global images are shortlisted, a refined set of feature vectors are generated from the final layer of the CNN model. If each of the 400 extracted images correspond to 120 neurons in the final layer, then a 400×120 feature space (global feature space) is generated. This feature space constitutes the final search space in the retrieval process. Assuming a maximum of 5 query images, they are processed through the trained CNN model to obtain a 5×120 feature space (query feature space).

3.5. Creating bags of query images

In this stage, the search query has to be compared with positively mapped global repository. Instead of a blind image differentiation, a weighted similarity measure is applied to compare the query feature space and global feature space. This is done for grouping similar images in the global feature space according to each of the query space feature vector. The feature vectors are compared using similarity checks, and Chi-square similarity measure is preferred over Root Mean Square (RMS) due to its benefits. The query images and the reduced images are compared based on a threshold similarity. The threshold is fixed adaptively as the algorithm progresses its run. This results in the creation of a bag of indices which are ordered by similarity. Assuming a total of q images in the query image space having 120 feature vectors per image, it accounts for a total size of $q \times 120$ elements. This results in creating q bags with relevant search results. If the value of q is 5, then q will range from q_1 to q_5. Thus q_1 will be compared with the 400 global images and the similar image indices will be set aside as Bag 1. Similarly, q_2 will be compared with the 400 global images and the similar image indices will be set aside as Bag 2, and so on.

As an ideal case, if all query images are similar to each other, then all bags will contain similar indices. However, in practical application, the contents of the bag may differ as the query space images are not identical to each other. To obtain refined results, we proceed to find the most frequently occurring item from the bag based on a weighted priority. The assignment of weights to bags is done as: the weight of Bag 1 is 1, weight of Bag 2 is $1/2$, weight of Bag 3 is $1/3$, weight of Bag 4 is $1/4$, and weight of Bag 5 is $1/5$. If an image is present in all bags, then its score will be calculated as $1 + 1/2 + 1/3 + 1/4 + 1/5$. A unique union of all images are taken from the set and the priority is calculated as follows.

Bags	Score calculation	Final Score
1	1+1/3+1/4+1/5	1.783
2	1+1/2+1/4	1.75
3	1+1/2+1/3	1.83
4	1/2+1/5	0.7
5	1/3+1/4+1/5	0.78

Table 1: Score calculation for setting the priority.

Let us assume that Bag 1 has global image indexes 1, 2 and 3; Bag 2 has global image indexes 2, 3 and 4; Bag 3 has global image indexes 1, 3 and 5; Bag 4 has global image indexes 1, 2 and 5; and Bag 5 has global image indexes 1, 4

and 5. The unique union of image indexes in this example is 1, 2, 3, 4 and 5. Hence, the score for the global image 1 is calculated as $1 + 1/3 + 1/4 + 1/5$, giving a value of 1.783. The score for image 2 is calculated as $1 + 1/2 + 1/4$, giving a score of 1.75. Similarly, the score for image 3, 4 and 5 are calculated as 1.83, 0.7 and 0.78 respectively. The priority calculation is shown in Table 1. Once again, a similarity threshold is applied adaptively on the score. If 1 is given as the threshold, then according to the proposed method, we select the images corresponding to the global image index represented by 1, 2 and 3. This will be finally reported as the retrieved result.

4. Results & Discussion

Quantitative and qualitative evaluations were conducted on the proposed model to prove its efficiency. For evaluation purposes, we used Oxford Buildings Dataset, [24, 14] which contains nearly 5000 color images and has 17 categories of images; each of these categories are focused on different landmarks at Oxford. For the experiments, we selected 50 images each from 15 classes. Some sample images are represented in Fig. 4. For an unbiased evaluation, we randomly created a query image set, local dataset, and global dataset by selecting 10%, 30%, and 60% of the overall data respectively. All experiments were conducted on MATLAB®2021*A* on a PC with Intel®Core i7-7700HQ CPU @ 2.80GHz processor, 16 GB RAM, and an NVIDIA® GTX 1050 graphics card with 4 GB memory.

Fig. 4: Some sample images from Oxford dataset used in the proposed CBIR system.

The evaluation of the algorithm using various standard image retrieval benchmarks are explained below.

1. *Average Precision (P_{avg})*
   Average precision gives the mean precision obtained while evaluating multiple random query images. If N_q is the number of query images then, the mean precision can be computed from the formula:

$$P_{avg} = \frac{1}{N_q} \sum_{i=1}^{N_q} \left(\frac{R_i}{N}\right)$$

   where R_i is the count of the i^{th} class images that are retrieved from the top N retrievals.
2. *Average True Positive Rate (TPR_{avg})*
   Average True Positive Rate (TPR_{avg}) or Recall is the average percentage of correctly retrieved samples from the available data in the global dataset. The formula is given as:

$$TPR_{avg} = \frac{1}{N_q} \sum_{i=1}^{N_q} \left(\frac{R_i}{min(N, M_i)} \right)$$

where R_i is the count of properly retrieved samples in i^{th} class, M_i is the total count of images in i^{th} class over the first N results.

3. *Average Error Rate (Err_{avg})*
   The average Error rate denotes the mean error rate of overall queries. It is calculated as:

$$Err_{avg} = \frac{1}{N_q} \sum_{i=1}^{N_q} \left(\frac{E_i}{N} \right)$$

where E_i is the erroneous detection in i^{th} class in the top N retrievals.

4. *Average False Positive Rate (FPR_{avg})*
   False Positive Rate is the average measure of the wrong retrievals from among false groups. It is calculated as:

$$FPR_{avg} = \frac{1}{N_q} \sum_{i=1}^{N_q} \left(\frac{FP}{Neg_i} \right)$$

where FP is false positives, and Neg_i is the number of negative samples from the i^{th} class.

In the proposed approach, the decisive factors that control the efficiency of the retrieval are the number of clusters used for finding the pseudo-labels, the feature vector used for assessing the image similarity and the query space. Experiments were conducted to analyze the performance variations with respect to the changes in those steering factors.

Fig. 5 represents the Average recall (TPR) using different sets of clusters and different number of images in the query space (Q). With differently sized query space, the best average recall performance was observed when Q is set as 3. This indicates that 3 images were present in the query space for the final comparison. In all cases, the performance degrades while retrieving more images per query space. More the number of cluster labels, higher will be the chance of finding similar images from the local dataset. But, the higher number of clusters reduces the images in the positive class leading to lower recall rates. So, the number of clusters was decided as a hyper- parameter, which was empirically selected as 2, as we obtained a reliable clustering output with this. The number of clusters used in the initial pseudolabelling also have a significant impact on the retrieval performance. Generally, the best performance is observed when there are less number of cluster labels. This is because, while increasing cluster labels, there is a chance to wrongly interpret many positive images into the rejection class.

In Fig. 6 the evaluation in terms of average error rate is represented. While retrieving less number of target images, the error is quite low (lesser than 5%), irrespective of the query objects in the query space. However, a small increase in the error rate is observed when there are four or more query objects in the query space. The error rate also increases while retrieving more number of target images per query object.

Receiver Operating Characteristic curve (ROC) is used to represent the effectiveness of a classifier system at various threshold levels. In the ROC curve, True Positive Rate (TPR), also known as Sensitivity, is plotted against False Positive Rate (FPR) and is shown in Fig. 7. Area Under the Curve (AUC) is the numerical value that indicates the

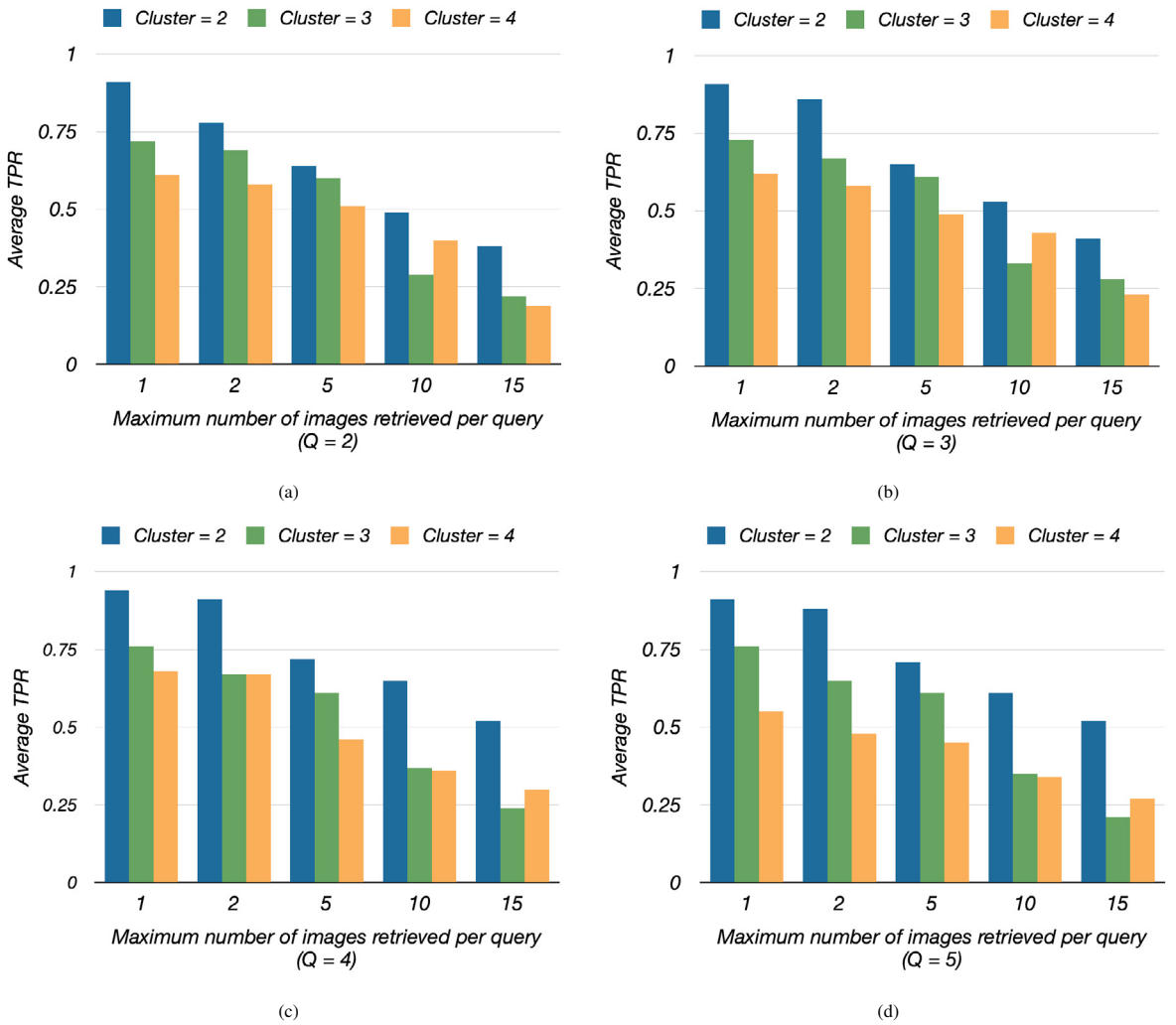

Fig. 5: Average True Positive Rate with different number of clusters and Q represents the number of images in the query space.

effectiveness of the classifier model. AUC value ranges from 0 to 1 where the value 1 indicates a perfect classifier model.

Methods	Average Error Rate	Average TPR	Computation time (Sec)
[36]	0.57	0.53	5.9
[2]	0.48	0.61	7.11
Proposed	0.19	0.64	7.3

Table 2: Performance comparison of the proposed method with state-of-the art approaches (while retrieving 10 images per query image).

In the experiments based on ROC curve, we observed an improvement in the AUC performance with an increase in the number of query space images. However, the performance came down while using very large number of images (greater than 4) in the query space due to the increased false positives during the retrieval process. Table 2 gives a quantitative performance comparison of the model with recent similar systems. The proposed method shows superior

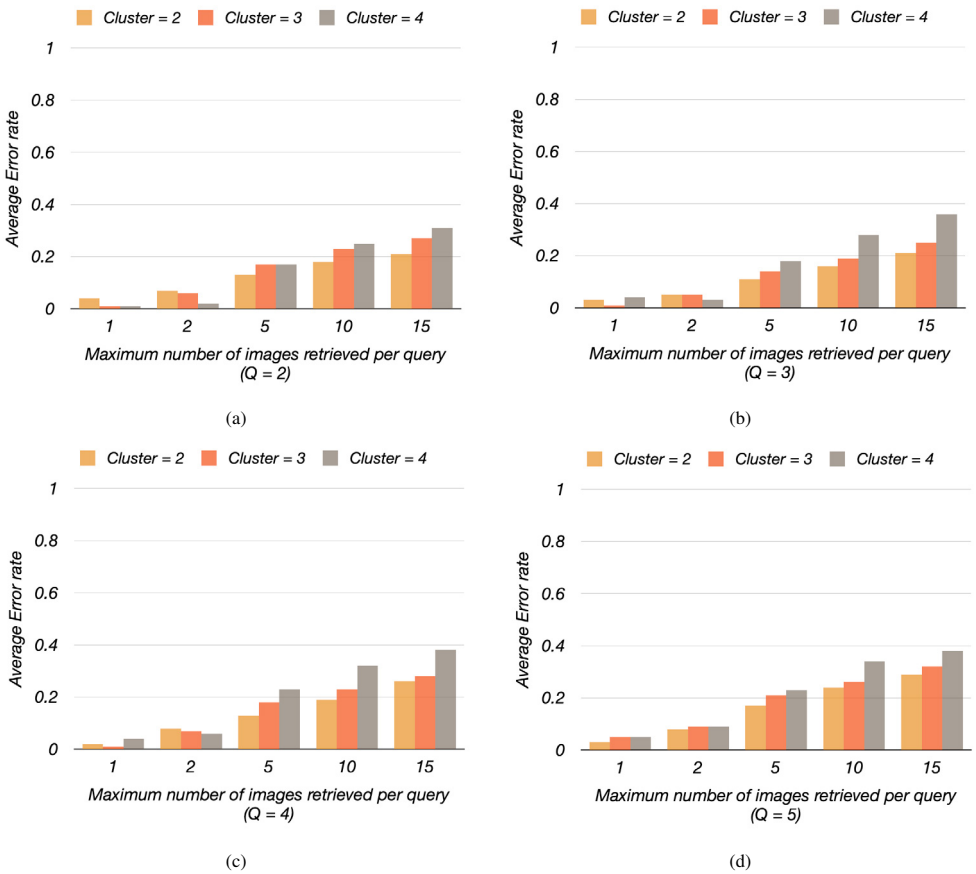

Fig. 6: Average Error rate with different number of clusters. Q represents the number of images in the query space.

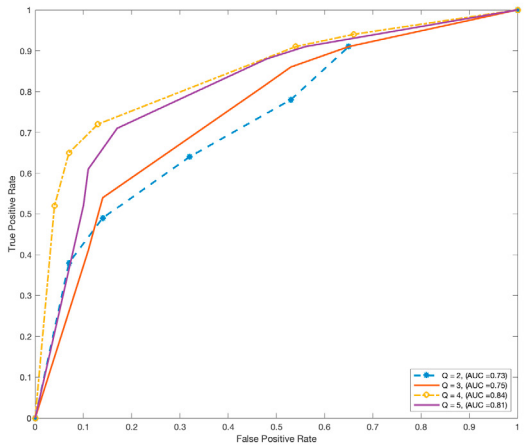

Fig. 7: TPR vs FPR ROC curve with 2 clusters for the pseudo labelling. Q represents the number of images considered in the query space.

performance over the related approaches in terms of error rate and average TPR. Nevertheless, the method consumes slightly more execution time as it involves stages that require very high computational overhead.

5. Conclusion

The proposed model for CBIR is targeted for electronic devices to search images from sources such as the internet, or other memory drives. The generation of image feature descriptors using autoencoders helps in dealing with images with lower dimensions. The use of k-means clustering with auto-encoder based features helps to pseudo-label the images with reduced computation cost so as to analyze its features. The approach also used a query expansion scheme that uses multiple query objects with weighted priority to make use of available images in the local memory to explore the visual saliency of the query images. K-means clustering and query image expansion help to restrict the false samples in the retrieved results and eventually improve the retrieval performance. The deep learning-based image classification model helps to restrict the target images for the accurate mapping of query objects onto relevant clusters. This cluster selection limits the image search, improves the retrieval precision and reduces the computation overhead. The final phase with weighted similarity assessment on the query images helps to maintain the priority of query space objects and thus reduces the retrieval error. The proposed model also shows better results over traditional CBIR techniques in terms of different quantitative assessments. Improvement in recall rate can be achieved through further research by integrating more robust image labelling and without compromising the average precision performance.

Acknowledgements

The second author is thankful to the School of Computer Sciences, Mahatma Gandhi University, Kottayam for his post doctoral fellowship.

References

[1] Aiswarya, K., Santhi, N., Ramar, K., 2020a. Content-based image retrieval for mobile devices using multi-stage autoencoders. Journal of Critical Reviews 7, 63–69.
[2] Aiswarya, K., Santhi, N., Ramar, K., 2020b. Retrieving mobile based scalable images using position scale orientation-scale invariant feature transform algorithm. Journal of Engineering Science and Technology 15, 524–540.
[3] Bay, H., Tuytelaars, T., Van Gool, L., 2006. Surf: Speeded up robust features, in: European conference on computer vision, Springer. pp. 404–417.
[4] Bibi, R., Mehmood, Z., Yousaf, R.M., Saba, T., Sardaraz, M., Rehman, A., 2020. Query-by-visual-search: multimodal framework for content-based image retrieval. Journal of Ambient Intelligence and Humanized Computing 11, 5629–5648.
[5] Chum, O., Mikulik, A., Perdoch, M., Matas, J., 2011. Total recall ii: Query expansion revisited, in: CVPR 2011, IEEE. pp. 889–896.
[6] Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A., 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE. pp. 1–8.
[7] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), IEEE. pp. 886–893.
[8] ElAlami, M.E., 2014. A new matching strategy for content based image retrieval system. Applied Soft Computing 14, 407–418.
[9] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., et al., 1995. Query by image and video content: The qbic system. Computer 28, 23–32.
[10] Garg, M., Dhiman, G., 2021. A novel content-based image retrieval approach for classification using glcm features and texture fused lbp variants. Neural Computing and Applications 33, 1311–1328.
[11] Giglio, S., Bertacchini, F., Bilotta, E., Pantano, P., 2020. Machine learning and points of interest: Typical tourist italian cities. Current Issues in Tourism 23, 1646–1658.
[12] Guo, Z., Zhang, L., Zhang, D., 2010. A completed modeling of local binary pattern operator for texture classification. IEEE transactions on image processing 19, 1657–1663.
[13] Hung, K.W., Aw-Yong, M., 2000. A content-based image retrieval system integrating color, shape and spatial analysis, in: Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0, IEEE. pp. 1484–1488.
[14] James Philbin, Relja Arandjelović and Andrew Zisserman, 2017. The oxford buildings dataset. https://www.robots.ox.ac.uk/~vgg/data/oxbuildings. Online; accessed 04 April 2022.
[15] Khan, A., Sohail, A., Zahoora, U., Qureshi, A.S., 2020. A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review 53, 5455–5516.
[16] Khan, U.A., Javed, A., Ashraf, R., 2021. An effective hybrid framework for content based image retrieval (cbir). Multimedia Tools and Applications 80, 26911–26937.
[17] Latif, A., Rasheed, A., Sajid, U., Ahmed, J., Ali, N., Ratyal, N.I., Zafar, B., Dar, S.H., Sajid, M., Khalil, T., 2019. Content-based image retrieval and feature extraction: a comprehensive review. Mathematical Problems in Engineering 2019.

[18] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436–444.
[19] Leutenegger, S., Chli, M., Siegwart, R.Y., 2011. Brisk: Binary robust invariant scalable keypoints, in: 2011 International conference on computer vision, IEEE. pp. 2548–2555.
[20] Li, P., Han, L., Tao, X., Zhang, X., Grecos, C., Plaza, A., Ren, P., 2020. Hashing nets for hashing: A quantized deep learning to hash framework for remote sensing image retrieval. IEEE Transactions on Geoscience and Remote Sensing 58, 7331–7345.
[21] Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60, 91–110.
[22] Maji, S., Bose, S., 2021. Cbir using features derived by deep learning. ACM/IMS Transactions on Data Science (TDS) 2, 1–24.
[23] Mistry, Y.D., 2020. Textural and color descriptor fusion for efficient content-based image retrieval algorithm. Iran Journal of Computer Science 3, 169–183.
[24] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A., 2007. Object retrieval with large vocabularies and fast spatial matching, in: 2007 IEEE conference on computer vision and pattern recognition, IEEE. pp. 1–8.
[25] Rastogi, P., Khanna, K., Singh, V., 2022. Gland segmentation in colorectal cancer histopathological images using u-net inspired convolutional network. Neural Computing and Applications 34, 5383–5395.
[26] Saritha, R.R., Paul, V., Kumar, P.G., 2019. Content based image retrieval using deep learning process. Cluster Computing 22, 4187–4200.
[27] Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. IEEE Transactions on pattern analysis and machine intelligence 22, 1349–1380.
[28] Sreedevi, S., Mathew, T.J., 2019. A modified approach for the removal of impulse noise from mammogram images, in: International Symposium on Signal Processing and Intelligent Recognition Systems, Springer. pp. 291–305.
[29] Thilagam, M., Arunish, K., 2018. Content-based image retrieval techniques: A review, in: 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW), IEEE. pp. 106–110.
[30] Tyagi, V., 2017. Content-based image retrieval techniques: a review. Content-Based Image Retrieval , 29–48.
[31] Tzelepi, M., Tefas, A., 2018. Deep convolutional learning for content based image retrieval. Neurocomputing 275, 2467–2478.
[32] Wan, J., Wang, D., Hoi, S.C.H., Wu, P., Zhu, J., Zhang, Y., Li, J., 2014. Deep learning for content-based image retrieval: A comprehensive study, in: Proceedings of the 22nd ACM international conference on Multimedia, pp. 157–166.
[33] Wang, H.H., Mohamad, D., Ismail, N.A., 2009. Image retrieval: techniques, challenge, and trend. World Academy of Science, Engineering and Technology 60, 716–718.
[34] Warrier, S.C., Mathew, T.J., Alcantud, J.C.R., 2020. Fuzzy soft matrices on fuzzy soft multiset and its applications in optimization problems. Journal of Intelligent & Fuzzy Systems 38, 2311–2322.
[35] Warrier, S.C., Mathew, T.J., Varadarajan, V., 2022. Parametrised hesitant fuzzy soft multiset for decision making. Data Science and Security: Proceedings of IDSCS 2022 462, 103.
[36] Yang, X., Qian, X., Xue, Y., 2015. Scalable mobile image retrieval by exploring contextual saliency. IEEE Transactions on Image Processing 24, 1709–1721.
[37] Yu, J., Qin, Z., Wan, T., Zhang, X., 2013. Feature integration analysis of bag-of-features model for image retrieval. Neurocomputing 120, 355–364.
[38] Yuan, X., Yu, J., Qin, Z., Wan, T., 2011. A sift-lbp image retrieval model based on bag of features, in: IEEE international conference on image processing, pp. 1061–1064.