JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 09 Issue 01
2025

Research Article
https://doi.org/10.56979/901/2025

A Deep Instance-Based Learning Model for Content-Based Image Retrieval

Ayesha Bibi[1], Humayun Salahuddin[2*], Khawaja Tehseen Ahmed[1], Sayyam Zahra[1], and Mamoona Shafique[3]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan.
[2]Department of Computing and Innovation, Riphah International University, Sahiwal, Pakistan.
[3]Department of Computer Science, NCBA&E (Sub-Campus Multan), Multan, Pakistan.
[*]Corresponding Author: Humayun Salahuddin. Email: humayun.salahuddin@riphahsahiwal.edu.pk

Received: March 26, 2025 **Accepted:** May 16, 2025

Abstract: Over a decade, content-based image retrieval has been an active field. It is not possible to compare the performance of two of these systems using objective means. Consequently, finding successful or hopeful ways forward is very challenging which delays the progress of the field. Finding out if a CBIR application is of good quality is tough which influences how well such systems can be commercialized. A severe application cannot be developed or grow commercially unless its reliability can be proved. The TREC metric is frequently used for operations within a text document and TPC is usually used for database processing. Because of the framework in place, systems can now be checked against little, open-to-the-public test databases. This work sets out to build an image retrieval system that uses deep learning to understand the similarity in images belonging to certain classes because of how learnable features and a similarity measure are used, all supported by inception-v3 CNN technology. To achieve simplicity, good retrieval and efficiency, the CNN features with a Siamese design are put to work.

Keywords: CBIR; Image and video Retrieval System; Deep Learning; Deep Neural Networks and Deep Features; Retrieval Performance Measures; Scale-Invariant Features Transform; Bag-of-Words; CNN Features; Siamese Convolutional Neural Network; Corel A,B; ZB Building Data Set; Simulation Tool

1. Introduction

1.1. Background and Motivation:

Computer, communication, and multimedia technologies have advanced significantly in the last two decades. As a result of this advancement, massive video/image data is being produced and archived, and large image databases and repositories are being established. Daily, we find more collections of images such as from medical scans, security trails, road traffic videos and so on. Because so much data is being collected, it is now necessary to build new image retrieval systems designed for large-scale use. In order to achieve its main aim, the CBIR is focused on developing an efficient system that supports tasks such as creating, handling and searching image databases in an accurate and speedy way. Processes that use internal features of an image—for example, shape, texture, statistics and color—to automatically store or find images are known as content-based image retrieval (CBIR). It is also possible to handle low-level image information in ways useful for object recognition, 3-D reconstruction, image registration and additional applications [4].

With Easy matching or basic algorithms like Support Vector Machine, Naive Bayes, K Nearest Neighbor, Random Forest or Euclid, you can find out how similar two images are. Even though data and images are now more elaborate, researchers from a number of research areas are interested in CBIR. It is equipped to manage

collections of images and collections of videos, too. A given query image is sent to the CBIR system which returns all the pictures that are similar. An image database gathers pictures from a wide range of classes and a CBIR system is supplied with a query image. Each of the video images is placed in one place in a video file. More and more, we use machine learning, an AI technology, to try and predict upcoming events. The Bag of Visual Words based on Speeded-Up-Robust Features (SURF) was created for handling video summary, CBIR and bio-metrics systems. SURF is developed for features that do not change when the size, rotation, position or light setting changes.

Previously, CBIR relied on images and the techniques used did not guarantee the same results for different image formats. Getting every detail from the image with this method often leads to memory errors and the process is very slow. SURF, unlike the older method, is both faster and more efficient. As soon as SURF features are extracted from the video frames, an unsupervised machine learning method will be used to make the double format into fewer quantized features to improve system performance. Data will be quantized by using the k-means clustering algorithm. A centroid or mean is the outcome of putting together all the visual words to create a bigger collection of images and words.

To find the number of occurrences for every word, a histogram can be used to uncover the meaning of visual words. A distance function is used to compare the video frame features with the query images' features. It will find out how far apart the features are in the query and the video frames. The distance between the feature images is always from 0 to 1 pixel; those that are most similar due to having the smallest distance are put into the search results. A CBIR system that can be used automatically and works well at scale is what we need. CBIR needs to examine images using only image features, not any text or metadata, to sort and find digital collections according to the type of content described above.

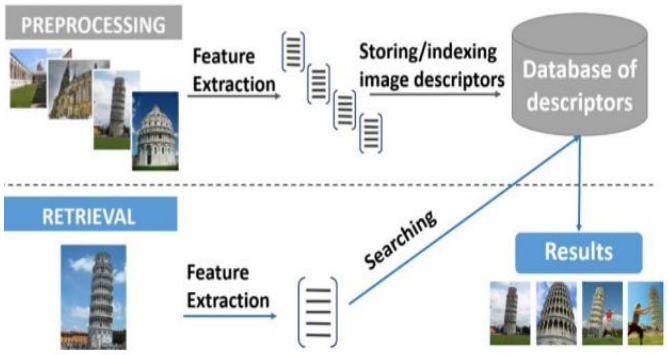

Figure 1. Overview of a generic visual information system.

The latest report on Content-based image retrieval (CBIR) outlines that there are three main parts in the system: processing the images, extracting features from them and measuring the distance between images [9]. All these three key factors are given the most attention. Initially, we need image descriptors that highlight unique features of each image. Normally, these descriptors are found in metric space, allowing the use of a distance function to compare images. The image features are put into a database so that if a query image is given, the system looks for the images in the database that match the query image features the most. What we have outlined here is just the outline of a visual information retrieval system. Extracting images from large collections of data often causes two important and repeated issues.

1- Efficiency:

People should get their search results almost instantly, with minimal use of resources and memory.

2- Effectiveness:

For users to be happy, search results should match the meaning of their search images.

A- Content-Based Image Retrieval: Any platform called a content-based image retrieval (CBIR) system helps systems organize image repositories so images can be searched and retrieved based on their contents [10]. In content-based search, images are analyzed directly, without needing tags, text or image metadata. CBIR systems use a process to take pieces of information from the pictures, these are called features. The images are

arranged using these important characteristics. Even so, getting systems to automatically analyze a colored pixel grid is not easy, mainly because of the semantic [11] and intentional [12] gaps. When there is a semantic gap, the strength and skill of image features differ greatly from the clear semantic concepts a user can recognize in an image. You can't notice the intention gap easily, because the meaning of an image is multi-layered and users have a hard time turning their search intent into a query.

Figure 2. Image polysemy and intention gap.

Furthermore, CBIR has become a popular technique for web image search, allowing images to be found by similar visual features, as well as on mobile apps, by using a live image query [13, 14]. To use CBIR technology well, the user must also have knowledge of object detection and recognition.

B- Images Retrieval System: A large database of images containing thousands of pictures is used in an image retrieval system. Both the source data and the end result in the system are pictures. Features are computed using the database images and the extracted features are saved in vectors called feature vectors. A single image is sometimes referred to as a query image for purposes of finding and extracting similar ones. Features from the query image are identified and then, by using a classifier, the most similar images in the database are found.

C- Video Retrieval System: A video retrieval system uses a database filled with many thousands of videos. Both the input and output of the system are video data. The data in the videos is transformed into video frames. Image features are calculated and taken from the video frames data and placed inside a feature vector. Similar videos are found by using a query video. Information from the query videos is taken and then, with a classifier or other method, the closest matches from the database are found.

D- Image-Based Video Frames Retrievals System: Videos and query images are both used in a video frame retrieval system from a video file. A video and a query image are given as input to the system and its output are similar frames. The video is turned into video frames as the first step. Features are computed and then taken from every frame in a video and stored in a vector. A similar image search needs to be performed. The features of the query image are also found and extracted and by using a classifier or comparison method, the closest frames from the video are identified.

E- Image Representations: Representing images in a particular features space forms part of Content-based image retrieval (CBIR) systems. A feature in image recognition is a characteristic that helps describe and represent an image in the best way. Generally, image features are expressed as numbers in a space and their similarity is found using a (di) similarity measure that takes into account their appearances. At the start, CBIR researchers used global features to describe the appearance of images using color, shape and texture. Global features help you work fast because they can be analyzed and compared in no time, yet they are too strict to represent images as they may ignore important details. The observation that every part of an image transmits a unique amount of information has inspired the use of more local features such as SURF [15] and SIFT [16]. Promising results were obtained for object recognition and images retrieval using deep features.

F- Deep Learning: The last few years have seen deep learning become increasingly popular in machine learning. Deep learning is defined as "a group of techniques that use several levels of representation and create these levels by joining together simple but non-linear modules that convert one level of representation into another" [17]. The origins of deep learning include the time frames from the 1940s to the 1960s [18] and from the 1980s to the 1990s [19]. For a long time, it wasn't popular, but that changed in 2006. But it was only in 2012 that deep learning began to quickly gain popularity, as you can see in figure 3. Three important reasons have made it possible for deep learning approaches to gain popularity again.

i) Availability of large datasets to train the models.

ii) Cheap and fast GPUs (to train and run large models.

iii) Algorithm improvement concerning the early works.

In addition, various open-source libraries (Caffe, Theano, Torch, Tensor-Flow, PyLearn2, MX-Net) and cloud providers (Amazon Web Services and Microsoft Azure) have been useful for many deep learning research or industrial projects. CNNs are just a type of deep learning model. Is normally focused on image handling. Other ways deep models can be used are with Recurrent Neural Network (RNN)

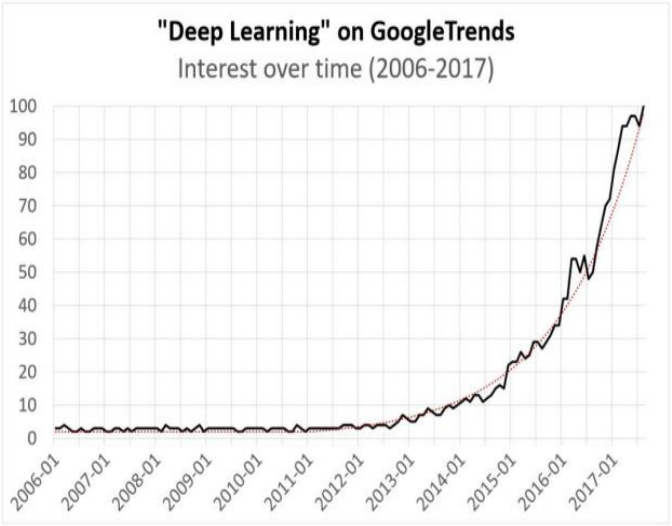

Figure 3. Interest over the time of the term "Deep Learning" estimated using Google Trends.

G- Deep Neural Networks and Deep Features: Deep neural networks which are common artificial neural networks, include an input layer, several hidden layers and an output layer. Each hidden layer applies certain math equations to its input in order to supply an output. The word "deep" is used because the network has more than one layer that is not visible. Having nonlinear operations in the hidden layers is necessary to avoid calling our network "shallow." Each deep neural network creates its output by using deep features from every hidden layer.

H- Main Contribution *of this Research*

• A new, efficient and adaptable framework is offered for the Content-based image retrieval (CBIR) system. Features descriptor can be used to extract image features from a database and query and another, more efficient, technique can also be used instead.

• The system applies instance-based learning to measure the distance between the query image and all the images in the database and the distance is sorted. Sorting the locations and distances of each point in the image is done by using this method.

• Inception-v3 is made up of 315 layers and the 313 layers before the last are responsible for getting 1000 features from a 299x299x3-pixel image.

• The new Content-based image retrieval (CBIR) framework is tested using three standard datasets. To assess and compare our results with others, we use the Mean Average precision as a performance evaluation metric.

• The proposed approach achieves 89.18%, 85.73% and 88.37% for Mean Average Precision on three different benchmarks, clearly showing that it is among the top CBIR approaches.

2. Literature Review

2.1. Related work:

We use benchmark datasets to assess the performance of our proposed features extraction method. CBIR accuracy is improved through the use and evaluation of several training and matching algorithms. This research is designed to boost the accuracy of the CBIR system.

For the last two decades, researchers have found the use of content in image retrieval from large databases to be an exciting topic. CBIR is an important use of computer vision and it uses Rank let Transform and RGB color features. At the preprocessing level, Rank let Transform helps make the image unaware of rotation. The purpose behind the similarity indexing method was to tackle the big scale error that arises with large-scale invariant feature transform (SIFT). CBIR systems are built with the help of two different features: texture and color. Since the texture feature LPB (Local Binary Pattern) changes with scaling, rotating and translation, it must be replaced with another LBP descriptor. Various feature extraction approaches are examined when using a K-NN classifier. RGB/HSV, shape/geometric and texture features are taken from a large image collection.

A new GPU based index structure for graphics processing is presented and results show that the approach works very quickly with only a slight reduction in data accuracy. Experimental testing demonstrates that the GPU retrieval system is faster in terms of time for CBIR systems [20].

A different approach for computing and extracting image features is proposed. To develop an image retrieval system, the research proposes a CNN model that effectively extracts good features from images. To use human vision, the model is structured in a hierarchy and labeled as a neural system response. Every image/frame and electrical charge is taken into account by the electromagnetism-based optimization technique. Here, CBIR systems use encrypted cloud data for similar images; first, a binary clustering technique is joined as a classification method. The extracted features are obtained using the HSV histogram and DCT histogram. The binary clustering technique is used to classify the matching between image database features and query image features. According to both the analysis and experimental work, the results are more accurate. Initially, image data is enhanced using a nonlinear method. The classification/matching.

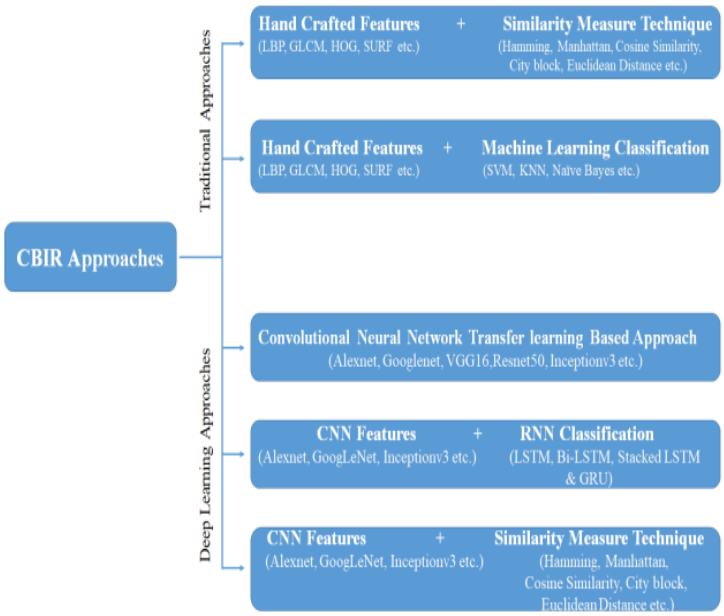

Figure 4. Various Traditional and Deep Learning Approaches for CBIR system Development.

2.2. Retrieval Performance Measures:

Quality of results and computation time are both used to assess the performance of an image retrieval system. The best option is to have accurate results, a fast reply and low memory usage. The main priority is for the

system to run efficiently. Which metric is used to evaluate an algorithm's performance depends on the reason the algorithm was created. Ordinarily, it takes from the moment the user sends a query to the time the results are shown for an image retrieval system. The most important thing is how long it takes to get image features and how much extra memory it requires during this part of the process. The attention of an index algorithm may be on how quickly it can run and how many computations and disk accesses it takes. The two most important methods to measure system performance are recall and precision. Precision measures how many of the objects returned by an algorithm are related to the query. How many relevant objects are brought back by the algorithm; this number explains how far the algorithm's reach covers the query.

Formally,

$$precision = \frac{|retrived \cap relevant|}{|retrived|}$$

$$recall = \frac{|retrived \cap relevant|}{|relevant|}$$

Where |. | represents set size. Such notions are frequently expressed using the below table of contingency.

It is to be noted that recall and precision are set-based measures. The rank positions in the result set of relevant

$$Precision = \frac{TP}{TP + FP}$$

And

$$Recall = \frac{TP}{TP + FN}$$

objects are considered by these measures. Another evaluation measure used for assessing unranked result-set is the mean of recall and precision, known as F-score. Other alternative measures are the accuracy= TP+TN/TP+TN+FP+FN, the specificity= TN/TN+FP, and the false-negative rate= FN/FN+TP; [21]

These metrics are valuable because they focus on getting the correct answers (relevant data). The position rank of the relevant objects in the results should be taken into account during evaluation in ranked retrieval. If we average the precision values every time a matching image is found to the query, we obtain the average precision.

In this case, we assess the results' quality using the probability p of finding an image of the same query object within the first r results. It is defined as:

$$\boldsymbol{p(r) = (R \leq r)}$$

Table 1. Measurements

	Relevant	**Non-Relevant**
Retrieved	True Positive (TP)	False Positive (TP)
Not Retrieved	False Negative (TP)	True Negative (TP)

Where *R* denotes random variable representing the position of the very first image to querying a ranked result list. For*r* = 1, and p is the classifier accuracy for recognition of most similar query objects.

$$\frac{1}{N} \sum_{i=1}^{N} [[r q_i \leq r]]$$

For r > 1, we estimate the probability p(r) as

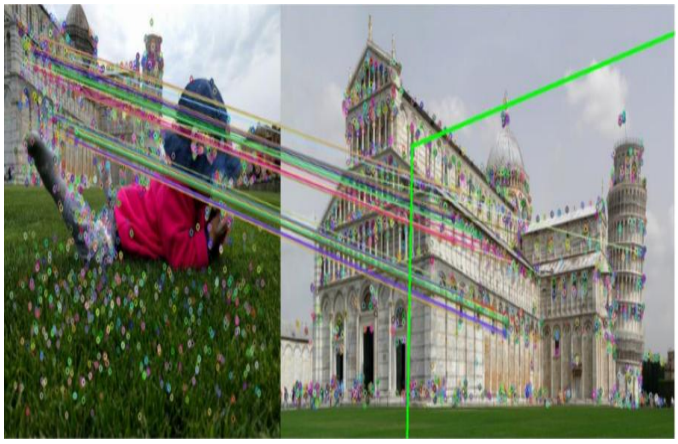

Figure 5. Image compared by matching their local feature and searching for a geometric transformation that can associates the region of both images.

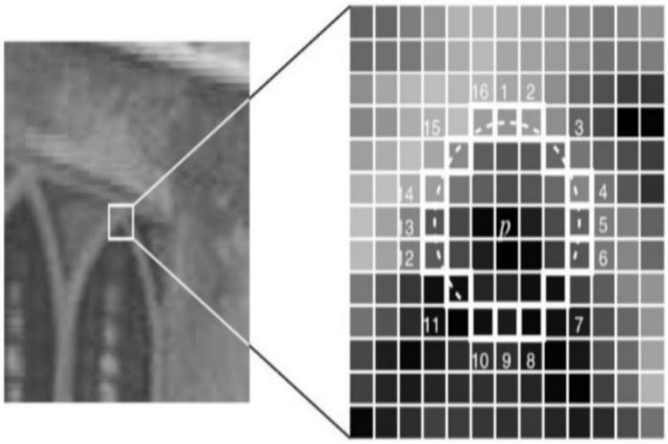

Figure 6. Visualization of the FAST key point detector reported

N is total the number of tested queries, rqi is the position of the first relevant image when querying qi, and [[·]] indicates the Iverson bracket argument is valid; it will be equal to one. Otherwise, it will be equal to zero. It is worth noting that, even if not used in this thesis, there are many other measures for evaluating ranked retrieval results, such as Position Error and Cumulative Gain [22].

2.3. Local Features:

A local feature is an image area that is distinct from its closest surroundings. The approach being used together with different features in an image is what allows us to find and describe local patterns. The idea behind these techniques is to identify certain main points (interesting locations) and identify the areas around each one. During the first phase called feature detection, key points in an image are automatically found and in the second phase, feature description, the patch around them is numerically described. [23-24].

2.4. Scale-Invariant Features Transform:

It is used widely as a local feature for recognition tasks because it is distinctive and robust to several image transformations and oft cited Scale Invariant Feature Transform (SIFT) is one such local feature. A Gaussian pyramid algorithm is used to establish a scale-space representation. Scale space extrema in the Difference of Gaussian function with the image are searched to localize candidate key points in space (2D position in the image) and scale (a level of the pyramid). The traditional choice is the highest peaks in a histogram of 36 bins over the range of 360 degrees, i.e. orientations. The result of a normalized histogram from the orientations of local image gradients (in the region around the selected point) is considered the feature descriptor. In our

original SIFT, 4x4 arrays of histograms with eight orientation bins in each were used. Thus, the final feature vector is of 128 dimensions: 4x4x8.

2.5. Binary Local Features:

In order to address this need, binary local features were recently proposed to compute local features fast efficiently and to compare quickly. It is also known (because of its name), that binary descriptors take only little memory (e.g. 512 bits, 256 bits and 128 bits). We match binary features based on XOR and with Hamming Distance. These methods just don't rely on a 'binary representation' notion alone. What follows are almost every binary description I have seen in the literature:

1. Taking a sample of pixel or pixel patches around a key point in a region.
2. Then we find region's key point using a method and rotate the region.
3. Next I chose a set of pixel or patch tuples (e.g., pairs or triples) For each tuple I calculated the bit value 0–1 as the result by comparing objects in the tuple. For this reason, the only real variation between differing methods is the comparison rule employed, selection of orientation and the selection of points or patches [25].

2.6. Bag-of-Words:

Researchers suggested at first that objects that can be found in a video database should initially be compared using the Bag-of Words (BoW) [26]. It has been popular lately for execution of classification and CBIR [27] afterwards. Image local descriptors are taken by BoW and each image is assigned to a set of visual words. For the purpose of learning visual vocabulary, large set local descriptors are grouped from training images. People tend to just use k-means or a variant of k-means. The primary visual concepts in the vocabulary are these centroids. Each local feature in the image is assigned to its nearest cluster centroid and the image is shown as a histogram of visual word occurrences.. (Figure 7).

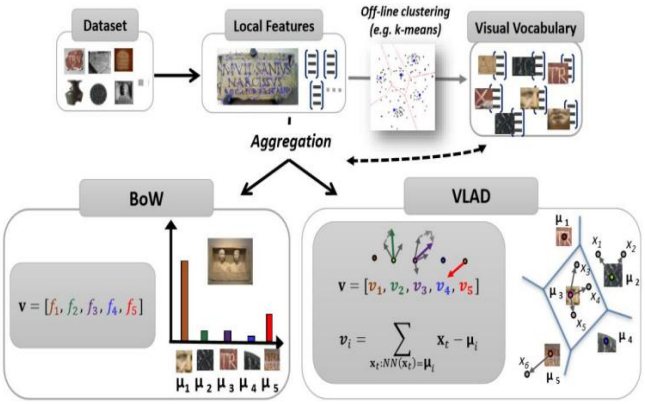

Figure 7. We present a simplified illustration of BoW and VLAD encodings. With a given visual vocabulary and the extraction of local feature from a given image, the bow encodes the number of descriptors assigned to each visual word and the VLAD encodes the accumulated difference bet.

To address the issue of quantization loss, we first used alternative encoding methods that could produce a correct representation of the original descriptors (e.g., Hamming Embedding [27], soft assignment, multiple assignments [27], locality constrained linear coding, sparse coding and the use of spatial pyramids).

2.7. Vector of Locally Aggregation Descriptors:

The Vector of Locally Aggregation Descriptors (VLAD) is an encoding scheme. It adopts the k-means to build a visual Also known as codebook, {μ1..., μK} is called 'Vocabulary'. Indeed, as is the case with BoW, we map each local feature (xt) from an image to its nearest visual word (N (xt)) in the codebook. Having computed these residual vectors, VLAD accumulates the residual vectors for each visual word, each is a vector of difference xt−μi between the centroid μi and the local feature xt that is assigned to it (Figure. 7). The accumulated residual vector is defined formally for the centroid μi as:

$$\boldsymbol{v_i} = \boldsymbol{\Sigma_{x_t:N\ N\ (x_t)=\mu_i} x_t - \mu_i}$$

Third, the vectors vi are combined into a single descriptor expressed as $v = [v1, \cdots, vK]$ called VLAD. The size D of the local features is exactly the same as the size of all residual subsectors. This means the dimensionality of entire vector V is fixed, i.e. D.K. A small number of centroids (K = 64−256) is used. The normalization is usually set as power law (v → |v|β sign (v)) and '2 normalization (v → v/kvk2'), after which two VLADs can be contrasted via the Euclidean distance or, equivalently, the inner product [28]. With high dimensional VLAD descriptors, PCA may be used to obtain better representation. Proposed several changed to the basic VLAD [29].

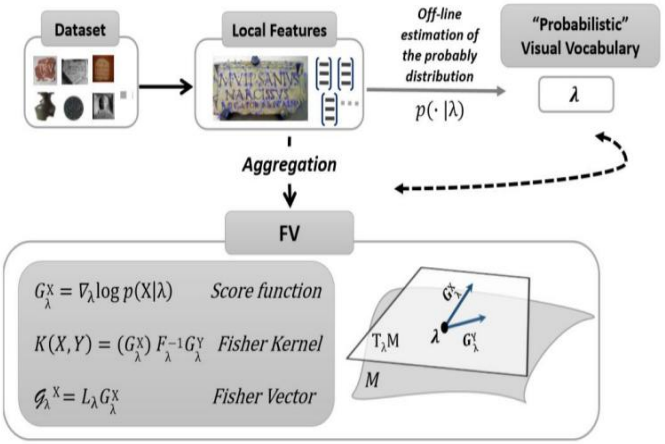

Figure 8. Simplified illustration of the F.V. encoding.

We consider parameterized family of distribution M = {p (·|λ)} λ indexed by a parameter vector λ as a Riemannian manifold with local metric induced by the FIM (F λ) as a local metric. Having a set X of local descriptors of an image, we compute the score function G.X. λ living in tangent space T λ M giving us a direction in such a parameter λ should change to fit X best. The induced metric F λ in Euclidean distance between the corresponding F.V.s gives this distance between two score functions.

2.8. CNN Features:

We argue that by using intermediate outputs (activations) of feed forward deep neural networks con motivated as data features of any (generic) task, one is able to achieve high performance on that new task even if it is clearly unrelated to the original task that the deep network was trained for. This work is the first work in this direction using CNN and the first to apply outputs of the sixth and seventh layers, fc6 and fc7 of pre trained Alex Net for the object recognition, scene recognition, domain adaptation and fine-grained recognition problems. Additionally other works started to view features as local features in a convolutional

We aggregate a layer to be using VLAD or F.V. In the end, we have to say that certain layers extract the features of an image.

A CNN of a CNN capture the image characteristics in various levels of abstraction. [30-31].

2.8.1. Hamming Distance:

The Hamming Distance is an often-used distance metric in term of comparing binary strings. Given two binary strings of the same length, it measures the number of bits positions that those two strings differ on. That real vector space gives rise to the Hamming distance of two binary strings which is indeed the 1 distance when we consider those two as points of that space. Implementation is trivial now and the most efficient way to calculate the Hamming distance is basically a bitwise XOR gate and a bit count. [32].

2.8.2. Euclidian Distance Based Similarity

For feature comparison and similar images retrieval the standard Euclidean distance similarity measure given in the equation below is extensively used in the state-of-the-art CBIR research work.

$$d(\theta_f, \Psi_f) = \sqrt{\sum_{i=1}^{v} [\theta_f(i) - \Psi_f(i)]^2}$$

The Questions are when to extract features from query image.

We determine that an appropriate time to extract features from the database image is and, if the dimension of a feature vector in Euclidian space is V and if d is the distance between two feature vectors in the same Euclidian space. A distance d between an image in the database and the query image is assigned to denote the difference between each image in the database and the query image and the distance d to each image in the database is calculated. Images that are most similar to the query image and database image the more likely the image appear higher in the results and the smaller the value indicates this is the case.

Figure 9. Example photos from the INRIA Holidays dataset.

2.8.3. Cosine Similarity

The word "cosine similarity "does not have any specific meaning and neither does this method have any clear meaning in literature either, so it worth to explain. Below is given the general equation for cosine

$$S_{cos}(x, y) = \frac{x \,.\, y}{\|x\|\|y\|}$$

similarity:

Referred to as the cosine distance, gives an angle of similarity between two images or other data points, which is a convenient estimate of their dimensional correlation. However, cosine similarity is not a proper metric because the equation's triangle of equality is missing. A method that can compute the distance and is also considered a proper method is to convert the values of two different vectors into angles. The output is ranged between 0 and 1. By:

The cosine distance is the term used to refer to an angle of similarity between two images or other data points — a nice estimate of how correlated the dimension is. Yet, cosine similarity

isn't a good metric because it lacks the triangle of equality of the equation. One way is a converted method which computes the distance thought considered a proper method is to convert the value of two different vectors into angles. This is ranged between 0 and 1.x/||x|| and y/||y||: so, it coincides with a rescaled Euclidean distance whenever the vectors are L2-normalized. In this thesis, we always use dCos as cosine distance unless stated specifically otherwise.

$$d_{cos}(x, y) = d_{cos}\left(\frac{x}{\|x\|}, \frac{y}{\|y\|}\right) = \frac{1}{\sqrt{2}}\left\|\frac{x}{\|x\|} - \frac{y}{\|y\|}\right\|$$

$$\tilde{\mathrm{d}}_{\cos}(\mathrm{x}, \mathrm{y}) = \cos^{-1}(\mathrm{S}_{\cos}(\mathrm{x}, \mathrm{y}))/\pi$$

$$d_{cos}\,(x, y) = \sqrt{1 - S_{cos}\,(x\,, y)}\ ,$$

We started with us building port image collections for which we provide a ground truth or built it automatically. Retrieval performance of different tested approaches is measured using the ground truth..

We used a collection of 1,491 personal holiday photos as the dataset. The dataset images are high quality and is composed of several scenes (water, human-made, fire effects, etc.). The dataset consists of 500 queries, all of which describe an object or scene. For every query they give us a list of positive results. The last kind of example images are shown as Figure,9.

Figure 10. Example images from EDR collection.

Oxford5k dataset is a collection of 5,062 Flickr images. It is consisting of 11 various Oxford buildings along with distractors. The dataset contains 55 query images: 5 images for each building. The dataset is provided with an extensive ground-based reality. Each query contains a set of four images: Junk, Bad, OK, and Good [33]. Some examples of the collection are shown in Figure 10:

Figure 11. Example of photos from the Oxford5k collection.

2.9. Epigraphic Database Roma:

EDR is a member of the International Federation of Epigraphic Databases (EAGLE), whose members also include the Electronic Archive of Greek and Latin Epigraphy (EAGLE) to which the Epigraphic Database Roma (EDR) contributes. But his counterpart working at the EDR has been charged with assembling all known Greek and Latin inscriptions published since the time of antiquity through the seventh century AD. In Figure 11: we present several applications.

2.10. Pisa Dataset

Pisa Dataset consists of 1,227 images of 12 various landmarks and monuments at Pisa Italy. Flickr 11 crawled the dataset during the VISITO Tuscany project. The complete dataset comprises of 226 (20% of the complete dataset) images as training set and 921 (80% of the complete dataset) images as test set. Some example photos are shown in Figure 12:

3. Research Methodology

The proposed work comprises two parts, learning based features extraction using Convolutional neural network and similar image retrieval by Euclidean distance similarity measure. An image matching technique which finds the similarity between two images to know to how large extents are they similar is called similarity measure. In data mining and computer vision, a similarity matrix is a distance of dimensions of object features.

Figure 12. Example photos from the Pisa dataset, uploaded to Flickr by the following users (left to right): Livorno Queen, eddip51, allylic, and Bunbury shire.

The closer we get to one, the more similar and the more distant the distance, the lower the similarity. What we are talking about here are similarity measures that are subjective and depend in the application and domain. For example, colors or edges difference makes two frames identical. As best a single feature shouldn't dominate distance computation nor can the relative values be normalized per element.

3.1. Siamese Convolutional Neural Network:

If two images or signals need to be worked on together in sequence, a Siamese neural network must be used. In image or signal processing, artificial neural networks apply the same weights to both aspects. To compare or evaluate output vectors, useful features must be extracted as features vectors. In most cases, the system computes one of the output vectors ahead of time to act as a reference for comparing the other output vector before the latter is calculated. A distance function for Locality-sensitive hashing works similarly to when comparing fingerprints; however, it is usually considered a distance function for fingerprints.

$$s(t) = \int_{-\infty}^{\infty} x(t-\tau)w(\tau)d\tau$$

When referring to identical twins, they are called "Siamese" twins. However, although they both appear as Convolutional Neural Networks above in figure 13, they are really two identical networks that share learning.

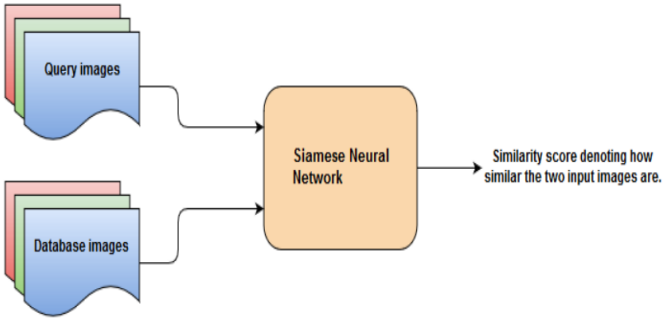

Figure 13. Siamese Neural Network Architecture.

When talking about parameters, they are almost the same. Expression x1 and x2 are fed as inputs into Conv Net. For every image, the Conv Net creates two feature vectors of fixed length (h(x1) and h(x2)) and then only outputs one feature vector (sum(h(x1)) + sum(h(x2))). Based on our assumption the neural network has been

properly trained; we may test this hypothesis. Feature vectors should be the same for both images if they are from the same person or object. If the images come from different sets of characters, their feature vectors must coincide. As a result, the element-by-element difference between the two representation vectors has to be quite different in both cases. We suggest applying Siamese Networks, because they form the basis for the Siamese Networks we designed.

3.2. Convolutional Neural Networks:

When the data looks like images or time series, feed-forward CNNs are the neural networks that perform best. The CNN processes a tensor, also known as a multidimensional array and delivers a high-dimensional structured result such as class probabilities in classification or real results in regression. For image classification, the network processes a three-dimensional color image as input, producing a vector of scores in response (0.70 for the cat, 0.10 for tiger, 0.05 for the dog and so on). A CNN referred to as deep uses a convolution in at least one of its layers. The overwhelming majority of neural network libraries and books specify the cross-correlation if instead you say "convolution," in machine learning, input (or receptive field) is defined by the first argument x, the kernel or filter as the second argument, w and the product, called the feature map, is given by the result s. Under these conditions, the input and the kernel are both tensors with a finite number of index. values (e.g., x(t) indicates what x is at the index t and t can only take on a few discrete values). That is, in regular practice, the convolution involves summing up a finite number of terms. An example is x of dimension nx ×mx and w of dimension nw ×mw, making the feature map a tensor (nx − nw + 1) × (mx − mw + 1) holding an element at position (t1, t2).

where the indexes range according to domains of definition of the considered tensors. Figure 14 gives an example of a 2D convolution.

$$s(t) = (x * w)(t) = \int_{-\infty}^{\infty} x(t-\tau)w(\tau)d\tau$$

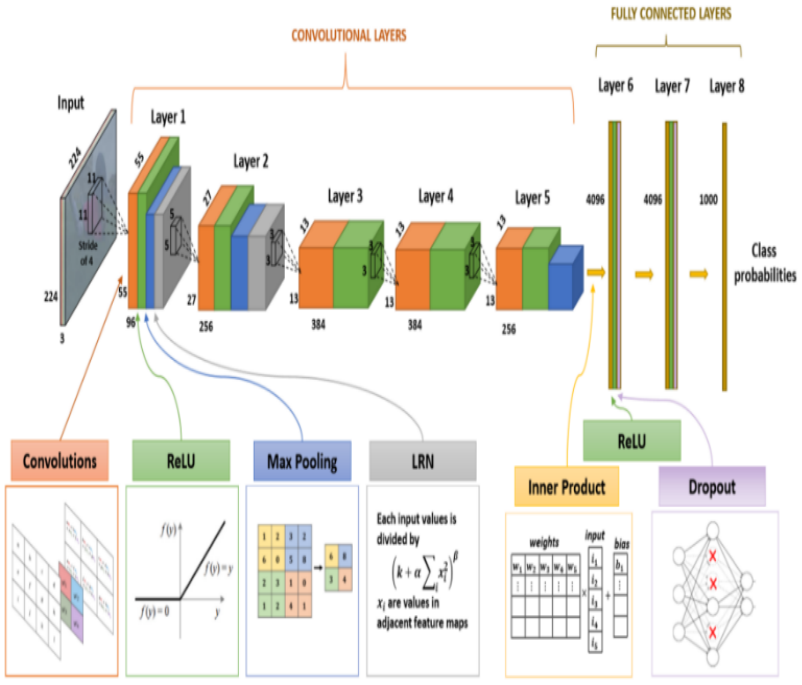

Figure 14. Example of a CNN model: the BVLC Reference Caffe Net.

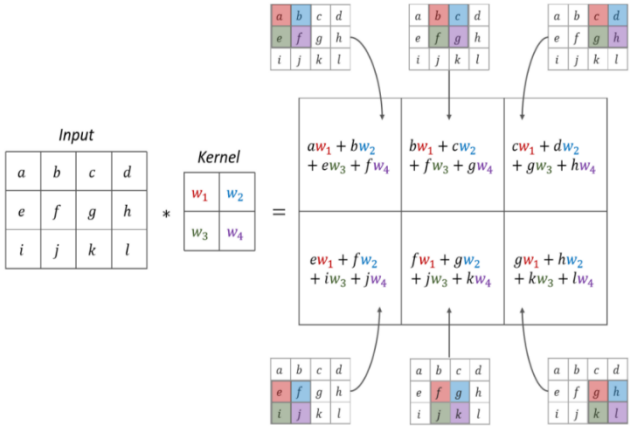

Figure 15. Example of convolution of 2D tensors.

Most of the time, CNNs use two hidden layers, known as convolutional and fully connected. Usually, the convolutional layers are built with multiple stages which make them notable including:

• Convolutions: Simultaneously, different convolutions help to produce the feature maps. Convolution kernels extract features from the input, so that's why we also call the output a feature map.

• Non-linear function: A non-linear activation function is applied to each element of the map. An example is the Rectified Linear Unit transform (ReLU) that replaces all the negative values by 0 (ReLU(x) = max (0, x)).

• Pooling: An element in a certain spot of the feature map is updated with an overall summary of the elements nearby. Most pooling functions let you choose between max-pooling (for maximum value in a rectangular patch) and Euclidean normalization of the same patch.

In principle, convolutional layers benefit from not having dense connections (using a kernel smaller than the input), sharing parameters among different parts of the kernel (each parameter is used everywhere in the input, aside from the boundaries) and keeping the same output pattern when the input image is translated. The second common type of layer is called the fully connected layer, since all of its units get information from the previous layer. Typically, the model goes through a "product with weights" step and then one or more nonlinear steps are applied. Dropout operates by randomly omitting units and their connections from the networks which keeps units from adapting and co-adapting too much on those data points.

The dataset uses RGB images that measure 224x224 pixels in size. Convolutions, ReLU, max-pooling and Local Response Normalization (LRN) outputs are orange, green, blue and grey, respectively and fully connected layers give yellows outputs; purple blocks highlight where dropout regularizations are used. Finally, the upcoming layer is a soft max module. Figure 15 contains a simplified diagram of the BVLC Reference.

4. Experimental Work

For the experimentation, we chose the Corel-1k, Corel-1.5K, ZB building coil and ZB building image datasets. We used these datasets to evaluate performance because they have been studied in recent research. I check how fast feature extraction from images is compared to the latest in CBIR research.

4.1. Dataset:

Three sets of challenging benchmarks CBIR datasets are used to perform the experiments presented in this thesis. The better performance in this experiment was given by a more robust CBIR system as the dataset being used had a unique number of classes, multiple images in each class and distinct image resolutions. These data are commonly used by researchers today. The results produced by our model are matched with the results seen in similar studies.

4.2. Corel A:

All images in the data set are from 100 semantic categories, with sizes of 256 x 384 or 384 x 256 pixels. All the 1000 images included in the Corel-1K image repository have been organized into ten groups based on their

themes. The images found in Figure 16 have been chosen as a representative sample from every semantic category in the Corel-A database. 700 random images are selected from the Corel-1K image repository to form the training set and another 300 random images from the Corel-1K repository are used to evaluate the approach.

4.3. Corel B:

The authors have collected and arranged approximately 1500 images into 15 semantic categories. All images in each semantic category are in a resolution of either 256x384 pixels or 3648x256, depending on the images available in that category. Corel-1.5K includes over 20,000 images and we show 15 categories and some images in Figure 16. The classifiers are trained with 750 randomly chosen images from the Corel-B image repository. A different set of 750 images from the Corel-B repository is also used for each of the two testing stages. The Corel-B image repository randomly selects the 750 images included in each dataset.

Figure 16. Random images from 15 semantic categories of the coral-B images database.

4.4. ZB Building Dataset

Many Contents based-image-retrieval operations rely on the Z.B. building dataset. The main reason this dataset is regarded as the most difficult is because many of the building photos are similar because of shared doors, windows and walls. The dataset includes two types of images: queries and database samples and the query image retrieves a match from the database. This collection includes 1005 images and there are 100 images per category.

4.5. Performance Measures on the Corel-A Image Database:

The Corel-A dataset is benchmarked, forming part of the Corel CBIR Databases. The results from using this dataset are used to assess the performance of different CBIR systems [36-37]. There are 10 different classes in the dataset and the images in each class number 100. Some samples from each of the ten classes make up the images shown in Figure 17. A total of 20 images were used to produce the MAP seen in Figure 17.

Figure 17. Some random images from Coral-A image database.

4.6. Simulation Tool:

The research work is done using MATLAB 2018b which includes toolboxes for image processing, computer vision, machine learning and deep learning, helping us to develop and test different methods rapidly. Since images are organized as matrices in MATLAB, it is excellent for matrix manipulation and is therefore often used in image processing, signal processing and related work. Data science researchers are drawn to MATLAB because of its data mining and Text analytics toolbox. The parallel processing and parallel computing toolbox

help us complete experiments efficiently on datasets that are larger than a gigabyte. You can use CUDA with MATLAB when operating on NVidia GPUs. The performance of MATLAB on a GPU with NVidia is impressive.

5. Conclusion

We have built a Content-Based-Image Retrieval system using Siamese networks in this work. Features that are important for learning are found in the query and database images by using a trained deep CNN model. With the Inception-v3 model, feature extraction is done; it is a model with 316 layers, including both extraction and classification parts. For our work, we depend on the 1st to 314th layer to draw out the image features. Euclidean distance is an accurate and quick method to determine resemblance between images for matching and getting similar images. Using three benchmark datasets—Corel-A, Corel-B and Z.B. building—the Mean Average Precision for the proposed system is respectively 89.18, 85.73 and 88.37. The model we suggest gives high accuracy and efficiency in retrieving similar images, excelling over some of the best methods being used today in CBIR systems.

References

1. Wu, T., Luo, T., & Wunsch, D. (2022). Learning Deep Representations via Contrastive Learning for Instance Retrieval. arXiv preprint arXiv:2209.13832.
2. Zhang, Z., & Peng, H. (2021). Instance-weighted Central Similarity for Multi-label Image Retrieval. arXiv preprint arXiv:2108.05274.
3. Zhang, Z., Wang, L., Wang, Y., Zhou, L., Zhang, J., Wang, P., & Chen, F. (2022). Instance Image Retrieval by Learning Purely From Within the Dataset. arXiv preprint arXiv:2208.06119.
4. Jagadale, A., Saif, M. A. N., Ghaleb, O. A. M., Ahmed, A. A. Q., Aqlan, H. A. A., & Al-Ariki, H. D. E. (2024). Efficient Artificial Intelligence Approaches for Medical Image Analysis. Artificial Intelligence Review, 57, 5953–5980.
5. Ahmed, K.T., Jaffar, S., Hussain, M.G., Fareed, S., Mehmood, A. and Choi, G.S., 2021. Maximum response deep learning using Markov, retinal & primitive patch binding with GoogLeNet & VGG-19 for large image retrieval. Ieee Access, 9, pp.41934-41957.
6. Chen, W. (2021). Exploring Deep Learning for Intelligent Image Retrieval. Doctoral Thesis, Leiden University.
7. Zhang, Z., & Peng, H. (2021). Instance-weighted Central Similarity for Multi-label Image Retrieval. arXiv preprint arXiv:2108.05274.
8. Wu, T., Luo, T., & Wunsch, D. (2022). Learning Deep Representations via Contrastive Learning for Instance Retrieval. arXiv preprint arXiv:2209.13832.
9. Ahmed, K.T., Shahid, N., Shabir, A., Khan, M.Y. and Hameed, M., 2024. Signature Elevation Using Parametric Fusion for Large Convolutional Network for Image Extraction. VFAST Transactions on Software Engineering, 12(2), pp.174-191.
10. Ahmed, K.T., Afzal, H., Mufti, M.R., Mehmood, A. and Choi, G.S., 2020. Deep image sensing and retrieval using suppression, scale spacing and division, interpolation and spatial color coordinates with bag of words for large and complex datasets. IEEE Access, 8, pp.90351-90379.
11. Sablatnig, R., Schurischuster, S., Loaiciga, J. M., & Kurtic, A. (2020). In-Time 3D Reconstruction and Instance Segmentation from Monocular Sensor Data. In 2020 17th Conference on Computer and Robot Vision (CRV) (pp. 27–34). IEEE.
12. Ahmed, K.T., Aslam, S., Afzal, H., Iqbal, S., Mehmood, A. and Choi, G.S., 2021. Symmetric image contents analysis and retrieval using decimation, pattern analysis, orientation, and features fusion. IEEE Access, 9, pp.57215-57242.
13. Delic, A., Neidhardt, J., Nguyen, T. N., & Ricci, F. (2016). Research Methods for Group Recommender Systems. In Proceedings of the Workshop on Recommenders in Tourism (RecTour 2016) (pp. 30–37). CEUR-WS.org.
14. Palotti, J., Goeuriot, L., Zuccon, G., & Hanbury, A. (2016). Ranking Health Web Pages with Relevance and Understandability. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 633–636). ACM.
15. Lipani, A., Lupu, M., Kanoulas, E., & Hanbury, A. (2016). The Solitude of Relevant Documents in the Pool. In Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval (pp. 1989–1992). ACM.
16. Rekabsaz, N., Sabetghadam, S., Andersson, L., Lupu, M., & Hanbury, A. (2016). Standard Test Collection for English-Persian Cross-Lingual Word Sense Disambiguation. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA).
17. Fesenmaier, D. R., Kuflik, T., & Neidhardt, J. (2016). RecTour 2016. In Proceedings of the 10th ACM Conference on Recommender Systems (pp. 417–418). ACM.
18. Bauer, W., & Seiringer, W. (2016). Improving PSS Costing based on Customer Integration. In Product-Service Systems across Life Cycle (pp. 123–128). Elsevier B.V.
19. Zuccon, G., Palotti, J., & Hanbury, A. (2016). Query Variations and their Effect on Comparing Information Retrieval Systems. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 729–738). ACM.
20. Lipani, A., Lupu, M., Aizawa, A., & Hanbury, A. (2015). An Initial Analytical Exploration of Retrievability. In Proceedings of the 2015 International Conference on The Theory of Information Retrieval (pp. 11–20).
21. Chen, Y., et al. (2023). Image classification with deep feature fusion: A survey. IEEE Access.