International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

An Advanced Relevance Feedback Method to Improve Performance of CBIR using Convolutional Neural Network and Comprehensive Values

Varkala Satheesh Kumar, T. Vijaya Saradhi

Abstract: ***Content-Based Image Retrieval (CBIR) is extensively used technique for image retrieval from large image databases. However, users are not satisfied with the conventional image retrieval techniques. In addition, the advent of web development and transmission networks, the number of images available to users continues to increase. Therefore, a permanent and considerable digital image production in many areas takes place. Quick access to the similar images of a given query image from this extensive collection of images pose great challenges and require proficient techniques. From query by image to retrieval of relevant images, CBIR has key phases such as feature extraction, similarity measurement, and retrieval of relevant images. However, extracting the features of the images is one of the important steps. Recently Convolutional Neural Network (CNN) shows good results in the field of computer vision due to the ability of feature extraction from the images. Alex Net is a classical Deep CNN for image feature extraction. We have modified the Alex Net Architecture with a few changes and proposed a novel framework to improve its ability for feature extraction and for similarity measurement. The proposal approach optimizes Alex Net in the aspect of pooling layer. In particular, average pooling is replaced by max-avg pooling and the non-linear activation function Maxout is used after every Convolution layer for better feature extraction. This paper introduces CNN for features extraction from images in CBIR system and also presents Euclidean distance along with the Comprehensive Values for better results. The proposed framework goes beyond image retrieval, including the large-scale database. The performance of the proposed work is evaluated using precision. The proposed work show better results than existing works.***

Keywords- CBIR, CNN, Alex Net, Feature Extraction, Similarity Distance, Comprehensive Values and Image Retrieval.

I.INTRODUCTION

The recovery of images based on content, also known as image query based image retrieval (QBIR) and content-based visual information retrieval (CBVIR), is the application of artificial vision techniques to the problem of image retrieval, or the problem search for digital images in large databases. Content-based image retrieval is opposed to traditional concepts-based approaches.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Varkala Satheesh Kumar, Ph.D Scholar (KL University) Assistant Professor, Dept of CSE SNIST, Hyderabad satteesh3@gmail.com
T. Vijaya Saradhi, Professor, Dept of CSE SNIST, Hyderabad saradhit@sreenidhi.edu.in

"Content-based" means that the search analyzes the image content rather than the metadata, such as keywords, tags or descriptions associated with the image. The term "content" in this context could refer to colors, shapes, textures or any other information that may be derived from the image itself. CBIR is desirable because searches based solely on metadata depend on the quality and integrity of the annotation. Making humans manually annotate images by entering keywords or metadata into a large database can take a long time and not capture the keywords you want to describe the image. The evaluation of the effectiveness of the search for images of keywords is subjective and has not been well defined. Similarly, CBIR systems have similar challenges to define success: "Keywords also limit the scope of the queries to the set of predetermined criteria". and "having been configured" are less reliable than using the content itself. The most common method of retrieving multimedia content from a large collection of images is to use metadata associated with images, such as date and time, geolocation, keywords, tags, labels or short descriptions and perform the task recovery through a text. Based on Text-Based Image Retrieval (TBIR). Although TBIR requires expensive time and labor, it is often not as effective. This is due to the subjectivity of the activity with respect to the meaning of its semantic content. Furthermore, the use of descriptive text to label and tag images does not always clearly and definitively reflect what an image represents because, for users, the image often contains more meaning and represents realities in a direct and clearer way. Those long sentences that can get confusing. Not only that, sometimes even the same word can have different meanings in different contexts. Furthermore, the content of an image can be described on two different levels: an image digitally contains pixels of colors from which it is possible to extract descriptors of colors, textures and shapes. At the semantic level, an image can be interpreted and can have at least one meaning. Unfortunately, in today's information systems, images are digitally described, while users are interested in their semantic content, while it is currently difficult to find correspondences between the semantic and digital levels.

Recent successes in deep learning techniques, in particular Convolutional Neural Networks (CNN) to solve the problem of computer vision applications, have inspired us to address this problem to improve CBIR performance. We must not overlook the fact that the flowering investigations in recent years have proposed the CNN in a primary position before extraction, classification and representation of CBIR characteristics.

In CNN there are several architectures available for feature extraction. They are:

a. The LeNet Architecture (1990s)
b. AlexNet (2012) – In 2012, Alex Krizhevsky (and others) released AlexNet which was a deeper and much wider version of the LeNet and won by a large margin the difficult ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. It was a significant breakthrough with respect to the previous approaches and the current widespread application of CNNs can be attributed to this work.
c. ZF Net (2013) – The ILSVRC 2013 winner was a Convolutional Network from Matthew Zeiler and Rob Fergus. It became known as the ZFNet (short for Zeiler& Fergus Net). It was an improvement on AlexNet by tweaking the architecture hyper parameters.
d. GoogLeNet (2014) – The ILSVRC 2014 winner was a Convolutional Network from Szegedy et al. from Google. Its main contribution was the development of an *Inception Module* that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M).
e. VGGNet (2014) – The runner-up in ILSVRC 2014 was the network that became known as the VGGNet. Its main contribution was in showing that the depth of the network (number of layers) is a critical component for good performance.
f. ResNets (2015) – Residual Network developed by Kaiming He (and others) was the winner of ILSVRC 2015. ResNets are currently by far state of the art CNNmodels and are the default choice for using ConvNets in practice (as of May 2016).
g. DenseNet (August 2016) – Recently published by Gao Huang (and others), the Densely Connected Convolutional Network has each layer directly connected to every other layer in a feed-forward fashion. The DenseNet has been shown to obtain significant improvements over previous state-of-the-art architectures on five highly competitive object recognition benchmark tasks.

CNN's architectures work directly on image pixels. Conventional neural network architectures always believe that an image contains three channels. These channels are channels R, G and B. As we know that each pixel is represented by the values R, G, B, CNNs take the atomic image as a three-tier structure.

Now we are ready to describe the Alexnet architecture for CNN. As shown in Figure 1, the grid contains eight layers. The first five are convoluted and the other three are completely connected. The output of the last fully connected layer is fed to a 1000-line softmax that generates a distribution over 1000 class labels. The kernels of the second, fourth, and fifth convolutional layers are connected only to the kernel maps in the previous layer located on the same GPU (see Figure 1). The kernels of the third convolution layer are connected to all the central maps in the second layer. The neurons in the fully connected layers are connected to all the neurons in the previous layer. The response normalization layers follow the first and second convergence layers. Maximum clustering levels follow both response normalization levels and the fifth convolutional layer.

The non-linearity of ReLU is applied to the output of each convolutional layer and fully-connected layer. The first convolutional layer filters the input image 227 × 227 × 3 with 96 cores of size 11 × 11 × 3 with a pitch of 4 pixels (this is the distance between the receptive field centers of neighboring neurons on a kernel map) . The second convolutional layer takes as input the output (normalized and grouped response) of the first convolutional layer and filters it with 256 cores of 5 × 5 × 48 size. The third, fourth and fifth convolutional layer are connected together without grouping intermediate or normalization layers. The third convolutional layer has 384 cores of 3 × 3 × 256 dimensions connected to the outputs (normalized, grouped) of the second convolutional layer. The fourth convolutional layer has 384 nuclei of size 3 × 3 × 192 and the fifth convolutional layer has 256 nuclei of size 3 × 3 × 192. The fully connected layers have 4096 neurons each.

The proposed approach optimizes AlexNet in the aspect of pooling level. In particular, the average group is replaced by the max-avg group for a better local extraction of the characteristics and the non-linear Maxout activation function is used after each convolution level for a better global information extraction. And we also did an operation on the image channels by dividing it with the smallest value of three corresponding pixels of three channels. This operation is mainly done to reduce the values in order to reduce the calculation time and will also help us to focus on the values with greater importance. When we divide the three corresponding values of three different channels with the minimum value of them, then the position in which the minimum value is found will finally be equal to 1 and the other values will not be equal to 1.

2. RELATED WORK

According to different classic techniques developed in recent years, each of which has many disadvantages, for example the histogram; First, this representation leads to the loss of spatial information, which is important for correctly representing the content of the image. Secondly, the use of the histogram poses the problem of quantifying the characteristic spaces.

CNN, specifically designed to manage the variability of 2D shapes, has proven to have superior performance to

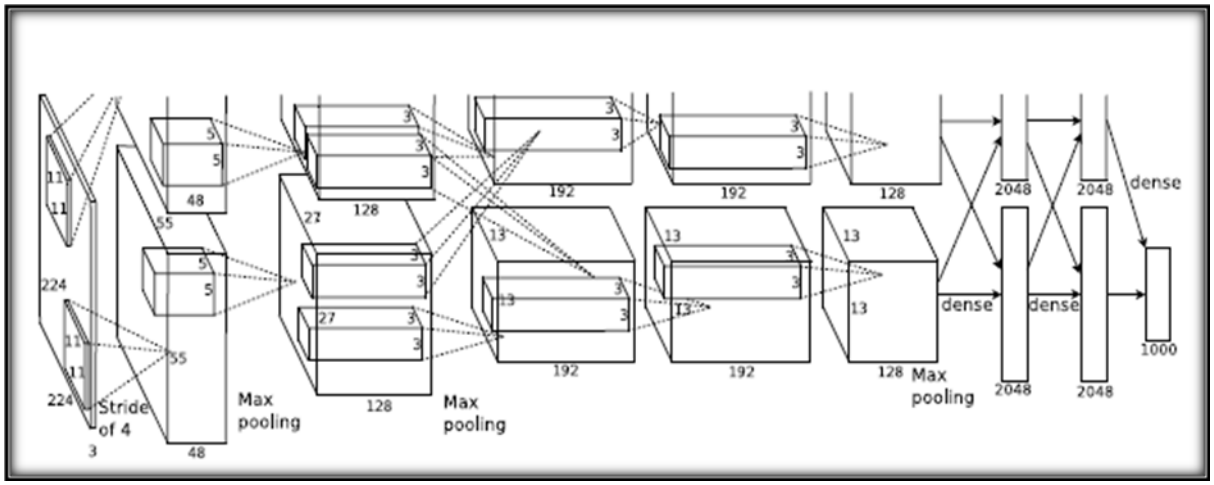

Figure 1 An Alexnet architecture of CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers.

all other techniques. The recognition systems are composed of several modules that include the extraction of features, the classification and learning of the paradigm. They enable these multimodal systems to be globally trained using gradient-based methods to optimize a measure of overall performance.

In 1998 [1], it achieved positive results for the first time by adopting supervised back-propagation networks for the recognition of figures. With the advancement of machine learning and recovery of image processing and a significant reduction in computer hardware costs, such as the GPU (Graphics Processing Unit), researchers have exploited and developed different models and techniques. [2, 3] studied CNN.

[2] He studied a deep learning framework with CBIR applications by testing an advanced method of deep learning (convolutional neural networks) for CBIR tasks in various environments. It [3] offers a survey, through the fusion of the main information: ingredients that every recipe for the design of a CBIR system must include to meet the demanding needs of users.

He presented[4] a parameterization workflow of the CNN model developed in the cloud computing platform of Microsoft Azure Machine Learning Studio (MAMLS), which is able to learn from the feature maps and classify multimodal images with different variability using a common flow Unlike the previous methods, the binarization approaches [5, 6] require pair wise entries for learning the binary code, the representation of the features has the best performance provided by CNN, the ability to generalize the extracted characteristics, the relationship between dimensional reduction and loss of precision in CBIR, the best distance measurement technique in CBIR and the advantage of coding techniques to improve the efficiency of CBIR, [5] have proposed to present a binarization approach of the characteristics for a better CBIR efficiency. More precisely, the binarization has reduced the use of the 31/32 space of the original data. [6] He proposed a deep learning framework to generate binary hash codes for a quick image restoration. The idea is that when data tags are available, the binary codes can be learned using a hidden layer to represent the latent concepts that dominate the class tags. The use of CNN also allows representations of "learning" images. His method passes hash codes and image representations in a timely manner, which makes it suitable for large-scale data sets.

They applied an active learning algorithm [7] of the support vector machine to make effective relevance feedback comments for image retrieval. The proposed algorithm chooses the most informative images to consult a user and quickly learns a limit that separates images that satisfy the concept of user queries from the rest of the data set. To exploit the PCA classification methods, the authors combined CNN with PCA [8-10]. Combined CNN and the victor support machine (PCA) [8] are applied in the CBIR and use (PCA) to train a hyperplane able to largely separate pairs of similar images and pairs of different images. PCA inputs are features of pairs that are assembled from image pairs: the image of the query and each test image in the image data set. The test images are classified according to the distance between the characteristic vectors of the pair and the trained hyperplane. Proposed CNN-SVM model [9], CNN is the extraction of features and SVM functions as recognition, while [10] combined CNN with linear SVM.

A new content-based medical image recovery (CBMIR) [11] framework was applied using CNN and a hash code is proposed. The new framework adopts a Siamese network in which image pairs are used as inputs and a model is learned to make the images belonging to the same class have similar characteristics through the use of weight sharing and a loss function. In each branch of the network, CNN adapts to extract the features, followed by the hash mapping, which is used to reduce the dimensionality of the feature vectors. In the training process, a new loss function is designed to make the feature vectors more distinguishable and a regularization term is added to encourage real value outputs to approximate the desired binary values. In the recovery phase, the compact binary hash of the query image is obtained from the trained network and subsequently compared with the hash codes of the database images. We experimented with two sets of medical imaging data: computed tomography of the cancer imaging file (TCIA-CT) and the vision and image analysis group / international program for early action against lung cancer (VIA / I- ELCAP). The results indicate that our method is superior to the existing hash and CNN methods.

Compared to the traditional hashing method, CNN-based feature extraction has advantages. The proposed algorithm that combines a Siamese network with the hash method is superior to the classic CNN-based methods. Applying a new loss function can effectively improve recovery accuracy.

Shikui Wei et al [12], sought to explicitly discover the essential effect of visual prominence in CBIR through qualitative and quantitative experiments. To this end, we first generate the image-fixing density maps of a CBIR data set widely used through the use of an ocular tracking device. These fundamental truth prominence maps are used to measure the influence of visual prominence on the CBIR task by exploring different probable ways to incorporate such prominence signals into the recovery process. We found that visual prominence is really advantageous for CBIR activity and the best pattern of participation in prominence is probably different for different models of image recovery. Inspired by the results, this document presents careful convolution neural networks (CNN) of two currents with salient features incorporated within CBIR. The proposed network has two flows that simultaneously manage two activities. The main sequence focuses on the extraction of discriminatory visual characteristics that are closely related to semantic attributes. Meanwhile, the auxiliary flow aims to facilitate the main flow by redirecting the extraction of features to the content of the highlighted image to which a human being can pay attention. By combining these two currents in the main and auxiliary CNNs (MACs), the similarity of the image can be calculated as human beings do when they reserve surprising contents and suppress irrelevant regions. Numerous experiments show that the proposed model achieves impressive performance in image recovery in four public data sets.

A new automatic detection method for bio-reparative vascular scaffolds (BVSs) [13] has been applied through a convolutional U-shaped neural network. The method consists of three phases: data preparation, network formation and network testing. First of all, in the data preparation phase, we complete the task of labeling the related samples based on the experience of the experts, so these labeled OCT images are divided into original and masked OCT images (corresponding to X and Y in supervised learning, respectively). Thus, we form our data in a convolutional U-shaped neural network, which consists of five down-sampling modules and four up-sampling modules. A related training model can be obtained, which can be used to predict related samples. In the testing phase, we can easily use the trained model to predict the OCT input data so that we can get the relevant information about a VHL in an OCT image. Obviously, this method can help doctors diagnose the disease and make important decisions. Finally, some experiments are performed to validate our proposed method and the IOU criterion is used to measure the superiority of the proposed method. The results show that the method is completely feasible and superior.

To accelerate the recovery of images in the mobile cloud service [14], an efficient content-based image recovery system (CBIR) has been designed based on Hadoop. First of all, the characteristics of the image are extracted through a CNN and the characteristics of the image library are divided into different databases of secondary functions. Secondly, the results of image recovery are obtained based on the corresponding results of the distributed Hadoop platform. Simulation experiments verify the efficiency of the proposed system.

Avascular Necrosis [15] (AN) is a cause of musculoskeletal disability. As is common among young people, early intervention and rapid diagnosis are needed. This disease usually affects the bones of the femur, so the shape of the bones is changed due to the fracture. Other common sites include knees, shoulders, jaws and ankles. The recovery of bone images affected by AN is a challenge due to its different fracture positions. This work proposes an effective methodology for the recovery of AN images using Deep Belief CNN Feature Representation. Initially, the set of input data is subject to preprocessing. Image noise is eliminated with the median filter (MF) and resized in the pre-processing phase. The characteristics are represented using the CNN of deep belief (DB-CNN). Now, the representations of the characteristics of the image are transmuted into binary codes. Thus, the similarity measurement is calculated using the modified Hamming distance. Finally, images are retrieved centered on similarity values. The test results showed that the proposed work is better than the other existing techniques.

Unmanned aerial vehicles (UAVs) [16] have been widely applied in different fields, facing enormous image data, the detection of objects in UAV images is the subject of extensive research for their important status in both theoretical studies and practical applications. In order to obtain the precise detection of objects in UAV images based on real-time processing, a method is proposed to detect thick or fine objects for UAV images using a light CNN and salience of deep motion. The proposed method comprises three phases:

1) The extraction of the key frame using the image similarity measurement is performed on the UAV images to accelerate the procedure for detecting successive objects;
2) PeleeNet, a light CNN, extracts deep features to obtain the detection of thick objects in key frames;
3) LiteFlowNet and preliminary knowledge of objects are used to analyze the salience ma p of deep movements, which helps to refine the results of the survey. Detection results in key frames are temporarily propagated to the nearest non-key frames for accurate detection. Five experiments are performed to verify the effectiveness of the proposed method in the Stanford drone dataset (SDD).

The experimental results show that the proposed method can reach a comparable detection speed but accuracy higher than six state of art methods.

Alex Krizhevsky et al [17], trained a vast and deep CNN to classify 1.2 million high-resolution images in the ImageNet LSVRC-2010 competition in 1000 different classes. In the test data, they achieved first 1 and first 5 error rates of 37.5% and 17.0%, which is considerably better than the previous technique. The neural network, which has 60 million parameters and 650,000 neurons, is composed of five convolutional layers, some of which are followed by maximum grouping layers and three layers completely connected with a final 1000-path softmax.

To speed up training, we use unsaturated neurons and a very efficient GPU implementation of the convolution operation. To reduce excessive regulation in the fully connected layers, they used a recently developed regularization method called "abandonment" which proved to be very effective. They also included a variant of this model in the ILSVRC-2012 competition and reached a winning 5% error rate in the first 5 tests, compared to 26.2% achieved by the second best entry.

In [18], presented a CNN for the extraction of features in CBIR. The proposed CNN aims to reduce the semantic gap between low and high level features. Therefore, improve recovery results. His CNN is the result of a learning transfer technique that uses Alex Net's preformed network. Find out how to extract representative features from a learning database and then use this knowledge to extract query functionality. The experiments conducted on the Wang database show a significant improvement in terms of accuracy compared to the more modern classical approaches.

Large-scale multimedia information processing has attracted great attention in recent years due to the rapid growth of data, such as images and videos. Among them CBIR is one of the important problems for many applications, such as medical image analysis, video surveillance, remote sensing, etc. [19, 20]. CBIR aims to recover images that have the most relevant visual content from large databases. To reach the "content", the analysis of the image content is essential. Therefore, the representation of characteristics and the measurement of similarity are fundamental for CBIR. There is a famous stimulating problem called "semantic gap" in the CBIR [21]. The reason for this gap is the different way of seeing the image between man and the computer. Humans are used to using high-level concepts to describe visual content and measure similarities. Unlike humans, computers extract low-level characteristics of the pixels in the image. As there are no direct links between high-level concepts and low-level characteristics, there is a "semantic gap." To reduce the semantic gap mentioned above, many researchers have been led. Most approaches use different craft features to represent the visual contents of the images and try to measure the appropriate similarities to make the similarity of the low level characteristics closer to the similarity of the high level concepts [22 -23]. Some complete surveys can be found in [24-25].

Artificial intelligence (AI), particularly machine learning technology, has attracted great attention in recent years [26-27]. The goal of artificial intelligence is to allow computers to simulate human intelligence and manage activities in the real world. It is essentially similar to the reduction of semantic gaps in the CBIR. There are some efforts that seek to reduce the semantic gap with machine learning technologies. In particular, deep learning has made great progress in recent years [28], such as the deep neural network [29], the Boltzmann deep machine [30], the short deep network [31], etc. Among these, the deep convolutional neural network (DCNN) has already achieved many significant results in artificial vision, such as image classification [32], image segmentation [33] and object recognition [34]. The use of deep learning technology to reduce the semantic gap in the CBIR has begun in recent years [35-36]. In this document, we are committed to designing a new DCNN framework for the recovery of semantic images in large-scale databases. The proposed framework optimizes the internal structure of Alex Net's classic deep convolutional network. Improves the ability to represent network features. Numerous experiments were conducted in four representative databases and the results showed that the proposal outperformed AlexNet and other cutting-edge methods.

The contributions of deep learning technology to solve the problems of se- mantic image recovery can be divided into two categories: use them in the phase of extraction and representation of the characteristics and in the phase of measurement of similarity. It has been shown that the use of DCNN to extract image functionalities could obtain better semantic information than artisanal functionality. The basic idea is that the images are entered directly into the DCNN. The functions generated by the convolutional levels and the grouping levels are used as low level functions and the functions extracted from fully linked levels contain complete semantic information [37]. These functions could be used directly in image recovery [35].

The distance of the cosine or Euclidean distance could be used to measure the similarity to complete image recovery [38]. Meanwhile, the use of compact global descriptors learned from image classification [39] or the use of aggregate local descriptors [40] as a representation of features show better performance in image recovery. Furthermore, it is proposed to learn the information on the structure and color of the DCNN model previously trained in Image Net to retrieve images [41]. However, these methods require large-scale training data with labels. The training time will increase along with the deepest part of the network. To solve this problem, we propose an unsupervised way to extract functionality in DCNN [42]. In the aspect of similarity measurement, the central idea is to learn a metric of adequate distance that can minimize the distance of similar images and the maximum distance of dissimilarity. For example, Wu et al. [43] proposed a multimodal online framework of deep similarity learning to learn optimal metrics and the optimal combination of multiple modes. While Yan et al. [44], Norouzi et al. [45] and Lu et al. [46] used the DCNN to learn the hashing functions to measure the similarity. The common idea of these proposals is to learn the hash functions to map the vectors of high-dimensional features into binary codes. The Hamming distance is used to measure the similarity.

3. PROPOSED METHOD

The following Figure 2 shows the general strategy of the proposed system using CNN approach in CBIR system. The basic idea of relevance feedback is to make the system more

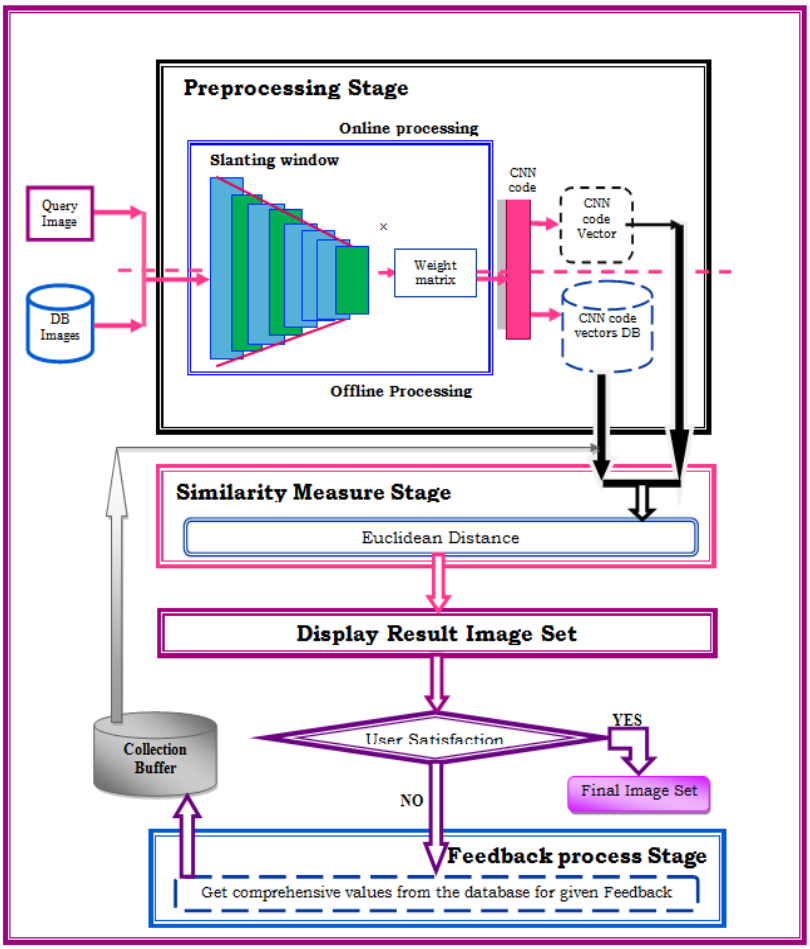

Figure 2 Proposed System Architecture

efficient so that it will display the user interested images and it also bridges the semantic gap between the low level features and high level concepts of the images. As we know that the low level features of an image are the color, shape, and texture. And high level concepts can be clearly explained with an example. Suppose a user gives a sun image as query image to the system, then the system may give the orange images to the user. Because system does not know that the retrieved is not related to the user's context. So here high level concepts are what type of images the user is expecting to retrieve in the result set.

Algorithm for proposed system is as follows

The proposed system is divided into four stages.

I. Preprocessing Stage

Step1: User will give the input query image to the proposed system.

Step2: Feature Representation step is used to find CNN code of an image for both given input image and database images. Construct the CNN code vectors for input image and CNN code vectors DB for image database. And also find out the $Average$, $Max\_Value\_Locations$, and $Min\_Value\_Locations$ for the query image and the database images. After successful completion of this send information to next stage.

II. Similarity Measure Stage

Step3: Finds the similarities between both images (query images and database images) using Euclidean distance and comprehensive values ($Max\_Value\_Locations$, $Min\_Value\_Locations$, $Average$). The similarity between I and Q is given by the following Eq. (1).

$$d(I,Q) = \sqrt{\sum_{i=1}^{n}(I_i - Q_i)^2} \quad (1)$$

where I_i *and* Q_i are CNN code values of *I(Db image)* and of *Q(Query image)* respectively.

III. Display Result Image Set

Step4: Displays the set of images which is having less value when comparing the similarities between input image and database images. Highest ranked images will be displayed to the user; the number of images displayed will depend on the size of the display window or threshold value given in the code of experiment.

Step 5: Check user satisfaction, if user satisfies with result stop the process. Otherwise ask user to give feedback on result image set by marking them as relevant images.

Step 6: After feedback the system will learn the user preferences.

IV. Feedback Process Stage

Step7: Identify relevant and sets from retrieved image set.

Step8: find aggregate Comprehensive values for the relevant set in the given feedback.

Step 9: collect any one of the comprehensive values into collection buffer based on the nth iteration.

Step 10: Repeat the procedure once again from stage 2 onwards using collection buffer.

Step 11: Repeat the same procedure until user gets satisfied with the displayed result set.
The following sections are described in detail about main parts of above proposed system. They are as follows.

3.1 The Preprocessing Stage

A. Slanting Window

The following Figure 3 describes about how the given query image is processed in various layers. We named it as a slanting window because in various stages the length and breadth dimensions of the feature maps will be reduced. We can find the 96 feature maps in convolution layer 1, and convolution layer 2. We can also find the 256 feature maps in Max-Avg1 pooling layer. In the same way we can find 384,256, 256,384 feature maps in CN3, CN4, CN5, Max-Avg3 pooling layer respectively.

B. Convolution Layer 1 (CN 1)

In convolution layer 1, we take 96 kernels of dimension 11×11×3 and we convolve with the query image of dimension 227×227×3. A kernel will consist the values of -1, 0, 1. The three channels in the kernel will superimpose on the three channels of the given query image and then a convolution operation will be taken place. A convolution operation is nothing but multiplying the values of the R, G, B channels with the values of the kernel. The first channel in the kernel will superimpose on the first channel of the image that is the R channel. The second channel of the kernel will superimpose on the second channel of the image that is the G channel. The third channel of the kernel will superimpose on the third channel of the query image. We used a stride of 4 in convolution layer 1. Finally we got 96 feature maps with the dimensions of 55×55.

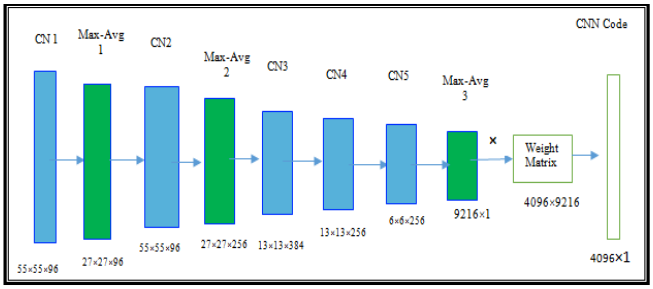

Figure 3 Slanting Window

Just after the completion of CN1, we applied this $ReLU$ activation function on the elements of each channel. By using this function we can make the negative elements of a channel to zero. Deep CNNs with $ReLUs$ train several times faster than their equivalents with $tanh$ units. Alex's team noticed that the number of iterations required to reach 25% training error on the CIFAR-10 dataset for a particular four-layer convolutional network. We have shown the plot below. In Figure 4 the plot shows how much fastly we can train the model, when we use the $ReLU$ function.

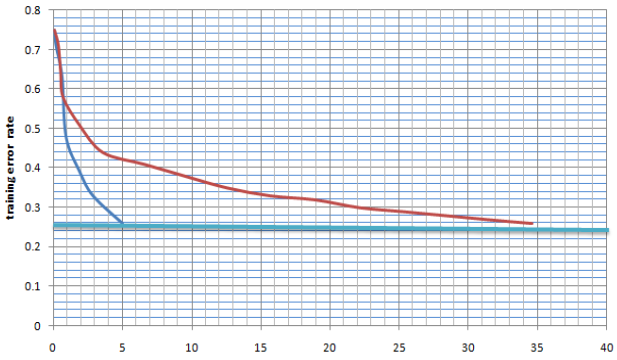

Figure 4 A four-layer convolutional neural network with $ReLUs$ (solid line) reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with $tanh$ neurons (dashed line). The learning rates for each network were chosen independently to make training as fast as possible.

Formula for ReLU activation function can be given by Eq. (2),

$$f(a) = \max(0, a) \tag{2}$$

After the completion of the ReLU activation function we applied a new method which was shown in Figure 5. This method will find out the minimum value in all the channels at corresponding locations. The minimum value that we get will be used to divide the corresponding values in all three channels. The main reason to apply this operation is to reduce the computational time that it requires. As we know that, as the values get lesser then the computational time will increase. This operation is shown in the figure given below. In the figure we can observe that we will divide the pixel values of R, G, B, channels with the minimum value 2. At the first location in the resultant matrix we will get the value of channel B will be equal to 1. So this operation reduces the intensity of the blue value at location I will be reduced. This will help us to concentrate on the values of the channels with maximum intensity.

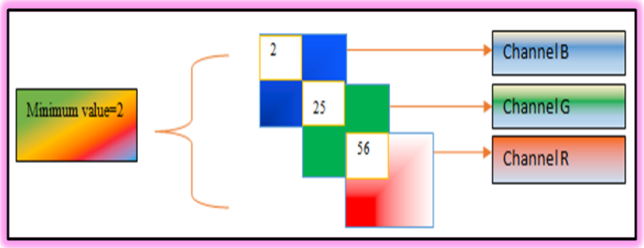

Figure 5 Obtaining minimum value from the R, G, B channels

We can get the minimum value by the following equation. In the equation the Min is a function that is used to get the minimum value (as in Eq. (3)). The i represents element's location. The location may be 1st or 2nd or 3rd whatever it may be. The i value range from zero to the number of elements in one channel. Suppose a channel of 3×3 will contain 9 elements locations. The i value for these 9 elements locations will range from zero to 8.

$$Min = min_i(R, G, B) \tag{3}$$

Note that the Min function will be used in all the layers.

C. Max-Avg1 Pooling Layer(Max-Avg 1)

In this layer we used the kernel of size 3×3×3 with the stride of 2. The Max-Avg pooling layer is divided into two parts. Firstly we will perform the max pooling operation and then followed by the avg pooling operation. Then we will get the difference of these Max pooling layer and Average pooling layer. So finally we represent this layer as Max-Avg1 pooling layer. The reason why we used the Max-Avg pooling layer is because we wanted to extract the values that are lying in between the max values and avg values. As these values will greatly represent the query image's content. So we concentrated on those features.

D. Max Pooling

The equation for the Max pooling operation is given below Eq. (4).

$$Z_{ij} = Max(x_{i,j}) \quad (4)$$

where x is 3×3 pooling window. It will superimpose on all the R, G, B channels, and it will extract one maximum value from all the R, G, and B channels.

E. Average Pooling

Taking all the elements averages as in Eq. (5)

$$A = \frac{1}{9}\sum x_{i,j} \quad (5)$$

where x is 3×3 is pooling window. It will superimpose on all the R, G, and B channels. We add all those 9 values of R, G, and B channels, and we divide them by the 9. Figure 5 is our modified Alexnet architecture. It contains 5 convolution layers, three Max-Avg pooling layers.

F. All other Layers

The operations in all the layers of query image are same. The only differences are the kernel size, the stride length followed and length of zero padding applied. Table-I shows the parameters and their dimensions used in our system. These are the details of the kernel size, stride length and the length of zero padding used.

Table-I. Parameters and their dimensions used in our system.

Layer Name	Kernel size	Pooling Window	No of kernels	Padding Length	Stride length	Input dimensions	Output dimensions
CN1	11×11×3	-	96	-	4	127×127×3	55×55×96
Max-Avg1	-	3×3×3	-	-	2	55×55×96	27×27×96
CN2	5×5×48	-	256	2	1	27×27×96	55×55×96
Max-Avg2	-	3×3×3	-	-	2	55×55×96	27×27×256
CN3	3×3×256	-	384	1	1	27×27×256	13×13×384
CN4	3×3×192	-	384	1	1	13×13×384	13×13×256
CN5	3×3×19	-	256	1	1	13×13×256	6×6×256
	2						
Max-Avg3	-	3×3×3	-	-	2	6×6×256	9216×1

Below we can find the algorithm (as shown in algorithm .1) for *finding* $Max\_Value\_Locations$ *and* $Min\_Value\_Locations$ *and Average* for all the images in the database. This is the offline process which is going to be triggered in preprocessing stage. As soon as we just open the application software, all the comprehensive values for the images in the database will be calculated. This will help us to readily keep all the comprehensive values for all images. So that, it will take less time when the user gives the feedback. It will directly get the comprehensive values from the database for the given feedback. So that the system need not to calculate the comprehensive values at that point of time. It can directly get those values and compare them.

Algorithm 1: *finding* $Max\_Value\_Locations$ *and* $Min\_Value\_Locations$ *and average for all the images in the database.*

Input : Given image

Output: $Max\_Value\_Locations$, $Min\_Value\_Locations$ *and* Avg

Begin

// for all the feature vectors from 0 to n-1,

$for(i = 0;\ i < n;\ i++)$

//for all the feature elements 0 to m-1.

$for(j = 0;\ j < m;\ j++)$

//finding average for the feature vector

Step1. $Avg[i]=\frac{1}{m}\sum_{j}^{m-1} f_m$

// finding $Max\_Value\_Locations$ to a feature vector i

Step2. $if\ (f[j] > Avg[i])$

Step3. $Max\_Value\_Locations[i] = j$

// finding $Min\_Value\_Locations$ to a feature vector i

Step4. $if\ (f[j] < Avg[i])$

Step5. $Max\_Value\_Locations[i] = j$

End

3.2 The Similarity Measure Stage

The Figure 6 given below shows the total process of what happens in the Similarity stage and Figure 7 Gives a rough sketch of Euclidean formula. As we can see in the figure, the system firstly compares the Euclidean distance. The query image's features are compared with the images in the database in terms of Euclidean distance. The formula for Euclidean distance is given in the equation 1. Now we will focus on Figure 6, in which now we are going to discuss about the 2nd iteration. Just after the first iteration, the relevant images will be displayed to the user based on the Euclidean distance. The user if satisfied with those images, then it's ok. Otherwise, the system asks the user for feedback by ticking the images of user interest. Just after the user gives his feedback by related images from images shown in the first iteration, the system collects the $Max\_Value\_Locations$ of the images which are given as feedback. Then, in the 2nd iteration the system will compare the $Max\_Locations$ of query images with the max value locations of the images retrieved in 1st iteration.

If the *Max_Value_Locations* of the query image matches with the *Max_Value_Locations* images retrieved in the first iteration then the system will display the matched images along with the query images. Now again the system will ask for the feedback. Just after the user gives his feedback, the system will find out the *Min_Value_Locations* of images and stores it in the collection buffer. The system will compare the *Min_Value_Locations* of the feedback images with images obtained in the second iteration. The matched images will be displayed. Again the system asks the user for feedback. Based on the feedback image's *Max_Avg* and *Min_Avg* the system will check the images obtained in the third iteration are lying in this *Max_Avg* and *Min_Avg*. Then after finding the satisfied images, the system will display them. Now at this time, the user will be definitely satisfied. And the system stops.

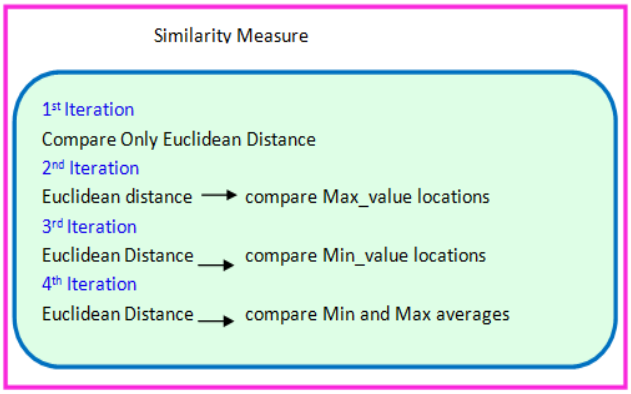

Figure 6 Iterations allowed in similarity Measure

A rough sketch of how the similarity measure will be taken place is given in the Figure 7 shown below. The similarity measure will be performed based on the two methods. The first method is the similarity measure by finding the Euclidian distance and second is by using the aggregate functions like min max and average. The formula for the Euclidian distance can be given by the formula shown below. And the formula for the average is also given below. We have also algorithmetically shown the process to Calculate *Max_Value_Locations for images* and also Matching the DB image *Max_Value_Locations* with query image *Max_Value Locations*. Now we will see the formula for Euclidian distance. This is the first method to find out similarity between two images.

$$d(I,Q) = \sqrt{\sum_{i=1}^{n}(I_i - Q_i)^2} \qquad (6)$$

where I_i *and* Q_i are CNN code values of *I* and of *Q* respectively.

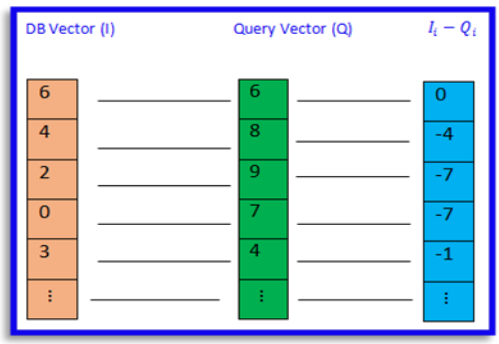

Figure 7 A rough sketch for Euclidean formula

A. Average

As we know that an average can be given by the sum of the elements by the number of elements, we performed the same here. We added the elements in the feature vector and then divided that sum of those values by the no of elements added. The formula for that is shown below.

$$avg = \frac{1}{n}\sum_{i=1}^{n} f_i \qquad (7)$$

where n implies no of feature elements of query image. f_i implies feature element at location i.

Here now we are going to show how we can find out the max value locations and *Min_value* locations. We have given an algorithm to calculate the *Max_value* locations and *Min_Value_Locations* for images. The images may be query image or may be database images. We have separate database for storing the *Max_Value_Locations* and *Min_Value_Locations* and *Average* values for each vector of an image. Figure 8 shows the Comparison of Max_locations, Min_locations and *Average* values of query images with the database images

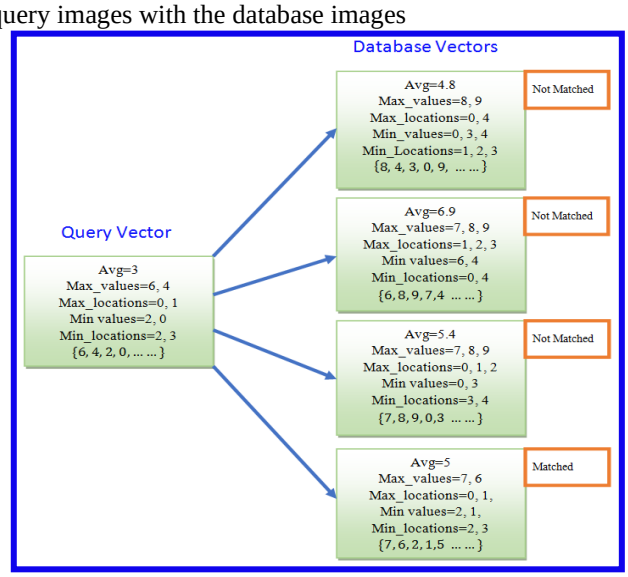

Figure 8 Comparison of *Max_Value_Locations*, *Min_Value_Locations* and *Average* values of query images with the database images.

***Algorithm2: Matching the query image $Max\_Value\_Locations$, $Min\_Value\_Locations$ and Average with DB image $Max\_Value\_Locations$, $Min\_Value\_Locations$ and* Averages.**

Input : Query image

Output: Similar Images

Begin

// To find *Average*,$Max\_Value\_Locations$, and $Min\_Value\_Locations$ *of query image.*

$for(i = 0;\ i < n;\ i++)$

// finding Average of a query vector

Step 1. Query_Avg= $\frac{1}{m}\sum_{i}^{n-1} f_m$ (n is no of elements in query vector)

// finding $Max\_Value\_Locations$ to a query vector

Step 2. $if\ (f[i] >$ Query_Avg)

Step 3. $Query\_Max\_Value\_Locations[i] = i$

// finding $Min\_Value\_Locations$ of a query vector

Step 4. $if\ (f[j] <$ Query_Avg))

Step 5. $Query\_Min\_Value\_Locations[i] = i$

$for(i = 0;\ i < n;\ i++)$

$for(j = 0;\ j < m;\ j++)$

// To find similarity between the Max_value locations of DB Image and given query image

Step 6. $if(Query\_Max\_Value\_Locations[i] == Database\_Max\_Value\_Locations[i][j])$

Step 7. Display_Images()

// To find similarity between the $Min\_Value\_Locations$ of DB Image and given query image

Step 8. $if(Query\_Min\_Value\_Locations[i] == Database\_Min\_Value\_Locations[i][j])$

Step 9. Display_Images()

//To find similarity between the Averages of DB Image and given query image

Step 10. $if(Query\_Avg \leq Max\_Avg \&\& Query\_Avg \geq Min\_avg)$

End

3.3 Feedback Process Stage

At this stage the user gives the feedback for three times. The first time when the user gives the feedback the system will find out the $Max\_Value\_Locations$ for feedback images and stores the values in the collection buffer for comparing with the values of the images obtained in the 1st iteration. The second time when the user gives feedback, the system will find out $Min\_Value\_Locations$ of the feedback images and store them in the collection buffer for comparing the values of the images obtained in 2nd iteration. The third time when the user gives the feedback, the system will find out the Max_Avg and Min_Avg from the feedback vectors of images and stores them in the Collection Buffer. Now the System will check the images obtained in the 3rd iteration. It will see whether the Avg values of the images displayed in the 3rd iteration are lying between Max_Avg and Min_Avg or not. If the Avg values of the images lies between the Max_Avg and Min_avg, then the system will display those images. At this stage the user will be completely satisfied.

4. PERFORMANCE EVALUATION

Precision is the ratio of the number of relevant images retrieved to the total number of images retrieved.

$$Precision = R/B \quad (8)$$

where B is the total number of images retrieved.

Figure 9 shows the 9 sample images of the COREL Database for different categories like people , beach, monuments, buses, dinosaur, flowers, horse, food. Figure 10 shows the Query images and desired set of retrieved images from COREL Database. Table-II shows the experimental results for the proposed work and the existing work using precision. Figure 11 shows graphical representation of precision percentage for each image category.

Figure 9 Example images from each category of COREL Database

Category	Query image	Retrieved images			
People					
Beach					
Monuments					
Buses					

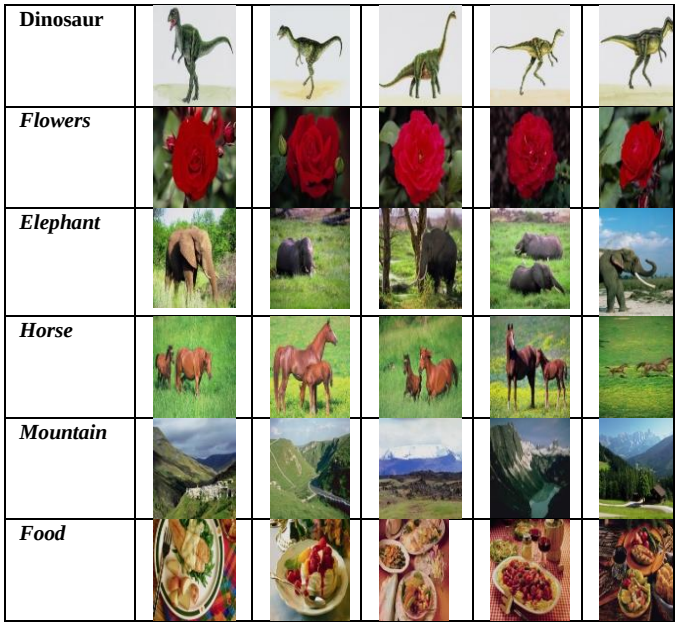

Figure 10 Query images and desired set of retrieved images from COREL Dataset

Table-II Experimental results for the proposed work and the existing work using precision

Method/ Cat	Peo	Bea	Mon ume	Bus es	Din osa	Average Value of
Yu [47]	0.8	0.3	0.616	0.81	0.99	
Lin [48]	0.6	0.5	0.562	0.88	0.99	
Guo [49]	0.8	0.4	0.682	0.88	1.00	
Anandh[50]	0.7		0.836	0.75	1.00	
Chiang [51]	0.0	0.9	0.260	0.07	0.87	
Amjad [52]	0.8	0.8	0.959	1.00	1.00	
Proposed	**0.9**	**0.9**	**0.989**	**0.95**	**1.00**	
Method/ Cat	**Flow**	**Elep**	**Horse**	**Mount**	**Food**	
Yu [47]	0.9	0.5	0.928	0.40	0.68	0.717
Lin [48]	0.8	0.6	0.803	0.52	0.73	0.727
Guo [49]	0.7	0.9	0.939	0.47	0.80	0.779
Anandh[50]	0.9	0.7	0.843	0.64	0.63	0.803
Chiang [51]	0.8	0.6	0.260	0.26	0.93	0.533
Amjad [52]	1.0	1.0	1.000	1.00	1.00	0.970
Proposed	**0.9**	**0.9**	**0.979**	**0.99**	**0.95**	**0.9768**

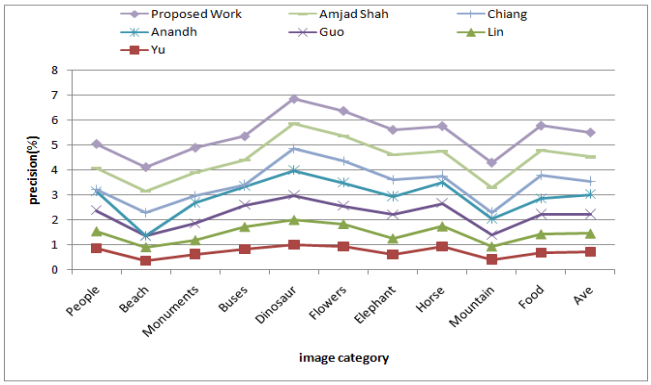

Figure 11 Graphical representation of precision percentage for each image category

CONCLUSION

This paper presents an enhanced CBIR system using CNN Alex Net architecture for features extraction. CNN extracts 4096 features per image. The experiments conducted on dissimilar image datasets. Proposed CBIR system uses Comprehensive Values for image retrieval and it results in higher accuracy in terms of precision i.e., 97% for Corel dataset. We accept as true that this study initiates studies using CNN for feature extraction in CBIR system and can be a foundation to extend to advanced CBIR approaches in future.

The experimental results are evidence that it is possible to build CBIR systems that can dynamically learn the target from very limited user feedback. The average duration of search iteration with our pipeline is very fast, and can be further reduced with more fine tuning of the system and improved hardware. In the future, we are planning to run more extensive simulation experiments as well as conduct extensive user studies to test the system for its applicability in various search scenarios.

REFERENCES

1. Lecun, Y., Bottou, L., Bengio, Y., et al.: Gradient-based learning applied to documentrecognition. Proc. IEEE 86(11), 2278–2324 (1998)
2. Wan,j.Wang,D.,Hoi,S.,Steven, C.H., et al.: learning for content based image retrieval.
3. Piras, L., Giacinto, G.: Information fusion in content based image retrieval: a comprehensiveoverview. Inf. Fusion 37, 50–60 (2017)
4. Ren, J.: Investigation of convolutional neural network architectures for image-based featurelearning and classification. Thesis (2017)
5. Wang, H., Cai, Y., Zhang, Y., Pan, H., Lv, W., Han, H.: Deep learning for image retrieval:what works and what doesn't. In: IEEE International Conference on Data Mining Workshop(ICDMW), pp. 1576–1583 (2015)
6. Lin, K., Yang, H.F., Hsiao, J.H., Chen, C.S.: Deep learning of binary hash codes for fastimage retrieval. In: Proceedings of the IEEE Conference on Computer Vision and PatternRecognition Workshops, pp. 27–35 (2015)
7. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In:Proceedings of the Ninth ACM International Conference on Multimedia, pp. 107–118. ACM(2001)
8. Fu, R., Li, B., Gao, Y., Wang, P.: Content-based image retrieval based on CNN and SVM.In: 2nd IEEE International Conference on Computer and Communications (ICCC), pp. 638–642. IEEE (2016)
9. Niu, X.X., Suen, C.Y.: A novel hybrid CNN–SVM classifier for recognizing handwrittendigits. Pattern Recognit. 45(4), 1318–1325 (2012)
10. Tang, Y.: Deep learning using linear support vector machines. arXiv preprint arXiv:1306.0239
11. YihengCai, Yuanyuan Li, ChangyanQiu, Jie Ma, XurongGao:Medical Image Retrieval Based on ConvolutionalNeural Network and Supervised Hashing
12. Shikui Wei, Lixin Liao, Jia Li, QinjieZheng, Fei Yang, Yao Zhao:Saliency Inside: Learning Attentive CNNs forContent-Based Image Retrieval.
13. W. Zhou *et al.*: Automatic Detection Approach for BVSs Using a U-Shaped Convolutional Neural Network
14. Caijuan Huang1, ZhuohuaLiu1(&), Hui Suo1, and Bin Yang2 mobile image retrieval system for cloud service based on convolutional neural network and hadoop.
15. Senthil Kumar Sundararajan1 & B. Sankaragomathi2 & D. Saravana Priya3: Deep Belief CNN Feature Representation Based Content Based ImageRetrieval for Medical Images
16. Jing Zhang a,*, Xi Liang a, Meng Wang a, b, Liheng Yang a, Li Zhuo a, c: Coarse-to-Fine Object Detection in Unmanned Aerial VehicleImagery Using Lightweight Convolutional Neural Network andDeep Motion Saliency
17. Alex Krizhevsky et al: ImageNet Classification with Deep ConvolutionalNeural Networks
18. SafaHamrera et all: Content Based Image Retrieval byConvolutional Neural Networks
19. J. Zhang , J. Shang , G. Zhang , Verification for different contrail parameteriza- tions based on integrated satellite observation and ECMWF reanalysis data, Adv. Meteorol. 2017 (1) (2017) 1–11 .
20. T. Mei , Y. Rui , S. Li , Q. Tian , Multimedia search reranking: a literature survey, ACM Comput. Surv. 46 (3) (2014) 1–38
21. R. Hong , Y. Yang , M. Wang , X.S. Hua , Learning visual semantic relationships for efficient visual retrieval, IEEE Trans. Big Data 1 (4) (2015) 152–161 .

22. C. Bai , J. Zhang , Z. Liu , W.-L. Zhao , K-means based histogram using multiresolu- tion feature vectors for color texture database retrieval, Multimed. Tools Appl. 74 (4) (2015) 1469–1488 .
23. C. Bai , J. nan Chen , L. Huang , K. Kpalma , S. Chen , Saliency-based multi-feature modeling for semantic image retrieval, J. Visual Commun. Image Represent. 50 (2018) 199–204 .
24. Y. Rui , T.S. Huang , S.-F. Chang , Image Retrieval: current techniques, promising directions, open issues, J. Visual Commun. Image Represent. 10 (1) (1999) 39–62 .
25. O.A.B. Penatti , E. Valle , R.D.S. Torres , Comparative study of global color and tex- ture descriptors for web image retrieval, J. Visual Commun. Image Represent. 23 (2) (2012) 359–380
26. C. Yan , H. Xie , S. Liu , J. Yin , Y. Zhang , Q. Dai , Effective Uyghur language text detection in complex background images for traffic prompt identification, IEEE Trans. Intell. Trans. Syst. 19 (1) (2018) 220–229 .
27. C. Yan , Y. Zhang , J. Xu , F. Dai , J. Zhang , Q. Dai , F. Wu , Efficient parallel frame- work for HEVC motion estimation on many-core processors, IEEE Trans. Cir- cuits Syst. Video Technol. 24 (12) (2014) 2077–2089 .
28. D. Li , A tutorial survey of architectures, algorithms, and applications for deep learning, Apsipa Trans. Signal Inf. Process. (2014)
29. W. Liu , Z. Wang , X. Liu , N. Zeng , Y. Liu , F.E. Alsaadi , A survey of deep neu- ral network architectures and their applications, Neurocomputing 234 (2017) 11–26 .
30. M. Norouzi , M. Ranjbar , G. Mori , Stacks of convolutional restricted Boltz- mann machines for shift-invariant feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR, 2009, pp. 2735–2742 .
31. G.E. Hinton , S. Osindero , Y.W. Teh , A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2014) 1527–1554 .
32. C. Bai , L. Huang , J. nan Chen , X. Pan , S. Chen , Optimization of deep convolu- tional neural network for large scale image classification, RuanJianXueBao/J. Soft. 29 (4) (2018) 1029–1038 .
33. R. Girshick , J. Donahue , T. Darrell , J. Malik , Rich feature hierarchies for accu- rate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587 .
34. A.S. Razavian , H. Azizpour , J. Sullivan , S. Carlsson , Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of the Computer Vision and Pattern Recognition Workshops, 2014, pp. 512–519 .
35. J. Wan , D. Wang , S.C.H. Hoi , P. Wu , J. Zhu , Y. Zhang , J. Li , Deep learning for content-based image retrieval: a comprehensive study, in: Proceedings of the Twenty-Second ACM International Conference on Multimedia, MM, ACM, New York, NY, USA, 2014, pp. 157–166 .
36. M. Tzelepi , A. Tefas , Deep convolutional learning for content based image re- trieval, Neurocomputing 275 (2018) 2467–2478 .
37. J. Donahue , Y. Jia , O. Vinyals , J. Hoffman , N. Zhang , E. Tzeng , T. Darrell , Decaf: a deep convolutional activation feature for generic visual recognition, in: Pro- ceedings of the International Conference on International Conference on Ma- chine Learning, 2014, pp. I–647 .
38. H. Wang , Y. Cai , Y. Zhang , H. Pan , W. Lv , H. Han , Deep learning for image re- trieval: what works and what doesn't, in: Proceedings of the IEEE International Conference on Data Mining Workshop, 2016, pp. 1576–1583 .
39. A.B. Yandex , V. Lempitsky , Aggregating Deep Convolutional Features for Image Retrieval, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vol. 11-18-Dece, IEEE, 2015, pp. 1269–1277 .
40. H. Jegou , M. Douze , C. Schmid , P. Perez , Aggregating local descriptors into a compact image representation, in: Proceedings of the IEEE Computer So- ciety Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3304–3311 .
41. J. Wang , H. Lu , H. Guo , Multiple deep features learning for object retrieval in surveillance videos, IET Comput. Vis. 10 (4) (2016) 268–272 .
42. M. Paulin , J. Mairal , M. Douze , Z. Harchaoui , F. Perronnin , C. Schmid , Convo- lutional patch representations for image retrieval: an unsupervised approach, Int. J. Comput. Vis. 121 (1) (2017) 14 9–16 8 .
43. P. Wu , S.C. Hoi , H. Xia , P. Zhao , D. Wang , C. Miao , Online multimodal deep similarity learning with application to image retrieval, in: Proceedings of the 21st ACM international conference on Multimedia - MM, 2013, pp. 153–162 .
44. C. Yan , H. Xie , D. Yang , J. Yin , Y. Zhang , Q. Dai , Supervised hash coding with deep neural network for environment perception of intelligent vehicles, IEEE Trans. Intell. Trans. Syst. 19 (1) (2018) 284–295 .
45. M. Norouzi , D.J.D.D.J. Fleet , R. Salakhutdinov , D.M. Blei , Hamming distance metric learning, Adv. Neural Inf. Process. Syst. (2012) 1–9 .
46. J. Lu , V.E. Liong , J. Zhou , Deep Hashing for Scalable Image Search, IEEE Trans. Image Process. 26 (5) (2017) 2352–2367 .
47. http://wang.ist.psu.edu/docs/related/.
48. http://sites.stat.psu.edu/ jiali/index.download.html.
49. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, ser.CVPR '97. Washington, DC, USA: IEEE Computer Society, 1997, pp.762–.768.
50. A. A. Associate, "Content Based Image Retrieval System based on Semantic Information Using Color , Texture and Shape Features," in *Computing Technologies and Intelligent Data Engineering (ICCTIDE), International Conference on*. Kovilpatti, India: IEEE, 2016.
51. T.-W. T. T.W Chiang, "Content-base image retrieval using multi resolution color and texture feature," in *J infTechnolAppl (CVPR '97)*, vol. 1. IEEE Computer Society, 2006.
52. Amjad Shah et al: Improving CBIR Accuracy using Convolutional Neural Network for Feature Extraction(2017).

AUTHORS PROFILE

Dr .T.Vijaya Saradhi received his post graduation in 2000 from Andhra University and his doctoral degree in 2016 from KL University. Currently he is working as a professor in the department of CSE, SNIST, and Hyderabad, India. His Research Interests include Deep Learning, Convolutional Neural Networks,Fog computing and IOT .He published 20 papers in reputed journals. He is member in SCI

V.Sateesh Kumar, Working as Assistant Professor in SreeNidhi Institute of Science and Technology, Hyderabad in the Department of IT. He completed his Graduation in B.Tech (CSE) From JNTU Hyderabad in 2006 and Post Graduation in M.Tech (SE) From JNTU Hyderabad in the year 2011. He got More Than 11 years Teaching Experience in different Engineering Colleges. His Areas of interest includes Image Processing, Machine learning, Information Retrieval Systems, Computer Networks, Network security and Neural Networks.