https://doi.org/10.1007/s42979-021-00529-4

ORIGINAL RESEARCH

Hybrid Approach for Content-Based Image Retrieval using VGG16 Layered Architecture and SVM: An Application of Deep Learning

Padmashree Desai[2] · Jagadeesh Pujari[1] · C. Sujatha[2] · Arinjay Kamble[2] · Anusha Kambli[2]

Received: 25 January 2021 / Accepted: 16 February 2021 / Published online: 25 March 2021

Abstract

Advancements in the sector of computer and multimedia technology and introduction of the World Wide Web have increased the volume of image databases and collections, for example medical imageries, digital libraries, art galleries which in total contain millions of images. The retrieval process of images from such huge database by traditional methods such as Text Based Image Retrieval, Color Histogram and Chi Square Distance may take a lot of time to get the desired images. It is necessity to develop an effective image retrieval system which can handle these huge amounts of images at once. The main purpose is to build a robust system that builds, executes and responds to data in an efficient manner. A Content-Based Image Retrieval (CBIR) system has been developed as an efficient image retrieval tool where user can provide their query to the system to allow it to retrieve user's desired image from the image collection. Moreover, the emergence of web development and transmission networks and also the number of images which are available to users continue to grow. We propose an effective deep learning framework based on Convolution Neural Networks (CNN) and Support Vector Machine (SVM) for fast image retrieval. Proposed architecture extracts features using CNN and classification using SVM. The results demonstrate the robustness of the system.

Keywords Content-Based Image Retrieval · Deep Convolution Neural Network · Support Vector Machine · Features · Extraction

Introduction

Content-Based Image Retrieval is a process of auto-indexing of images by extraction of their low-level features like color and shape; these features are responsible only for image retrieval. Feature representation and similarity measurement are the two most important instances for the implementation of the CBIR program and various researchers have done on the same research for more than a decade. Many different approaches have been proposed but to date it remains as one of the most problematic one in the ongoing CBIR research, and a major reason is the difference between low-resolution image pixels captured by machines and high-level human-sensing. This problem poses as the fundamental challenge for Artificial Intelligence with a high-level view which is the way to build and train intelligent machines as humans to perform real-world tasks. Machine learning is a promising alternative that attempts to address this challenge. Machine learning techniques have shown progress in recent years. Deep Learning is an important form of abstraction. It involves a family of machine learning algorithms that try to demonstrate high-quality data extraction techniques through deep design techniques that are made up of many offline variables.

Deep learning enables complex design architectures to be programmed as a human brain and process information in many stages of transformation and representation, in contrast to the conventional machine learning methods that often use fixed structures. By exploring deep design architectures which learn features at multiple levels of creativity from data, deep learning techniques makes it possible for the system to learn complex functions to map input data to the output directly, without relying on features present in the man-made domain.

This article is part of the topical collection "Data Science and Communication" guest edited by Kamesh Namudri, Naveen Chilamkurti, Sushma S J and S. Padmashree.

Padmashree Desai
padmashri@kletech.ac.in

[1] S. D. M College of Engineering and Technology, Dharwad, Karnataka, India

[2] KLE Technological University, Hubballi, Karnataka, India

The success of deep learning inspired us to explore deep learning techniques with application to CBIR tasks for images. Not enough focus is prioritized for CBIR applications although research has been conducted for application of deep learning techniques for classification of image and recognition in computer vision.

Proposed method involves application of deep learning techniques to solve the CBIR function of human-reduced images. We will be training large-scale neural network to learn representations of functional features. We have tested against established categories of the Corel dataset (African tribes, Beaches, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains and Food). The results of the proposed methods are compared with existing techniques and performance analysis is done.

Literature Survey

In [1], the authors have proposed a deep learning framework for CBIR by training large-scale Deep Belief Networks for learning effective feature representations of images. The authors have discussed three kinds of feature generalization schemes. In scheme I, images of dataset are fed into the pre-trained CNN model and then take the activation values from the last three layers. In scheme II, instead of directly using the features extracted by the pre-trained deep model, they explored similarity learning algorithms to refine the features obtained in scheme I. In scheme III, the deep convolutional neural network is retrained on image dataset for different CBIR tasks by initializing the CNN model with the parameters of the Image Net-trained models. Scheme III showed best performance out of the three schemes.

In [2], the authors have proposed an efficient pre-trained Convolution neural network model. The authors have used LeNet-5 model of CNN and experiments are conducted on Corel 1 K dataset. The proposed model has an average precision of 0.79 with Corel 1 K dataset [3–5]. The results are compared with three traditional visual features, the hue saturation value (HSV) color feature, gray level Co-occurrence matrix (GLCM) features and scale-invariant feature transform (SIFT) in which the proposed model has the highest average precision value- 86% compared to other methods.

In [6], the authors have proposed the Deep Belief Network (DBN) method of deep learning for feature extraction and classification. The DBN has several layers which include restricted Boltzmann machine (RBM) stacked into multi-stages, which consists of only single hidden layers each to make the learning process faster. The experimental results show that for the small dataset with 1000 images, the accuracy rate would be 98.6% but with a large data set (> 10,000 images) the accuracy would be 96% without losing the time complexity requirement.

In [7], the authors have proposed a CNN—SVM model, where CNN is for feature extraction, and SVM performs as a recognizer. The first part of a CNN is the convolutional phase. It works as an extractor of image features. In the end, the convolution maps are flattened and concatenated into a feature vector, called a CNN code. The SVM takes this CNN code at the output of the convolution phase as a new feature vector for training. The precision obtained using pre-trained CNN with the Caltech256 database is 90% for 1000 images. Padmashree Desai et al. in [8–13] discusses different methods of feature extraction using wavelets, edge operators, morphological operators ad moment invariants. Performance analysis is done using different distance measures. The videos summarization [14, 15] can be used in image/video retrieval by searching the query image in the summarized video dataset rather than in the original dataset. This can improve the retrieval time.

Proposed Method and Implementation

The proposed architecture as shown in Fig. 1. consists of two layers, first layer uses CNN for training and feature extraction and second layer uses SVM for classification and image retrieval. Feature vector is obtained from CNN model is fed as input to the SVM.

Basic flow starts with a query image submitted by user as input to the system. Features of query image are extracted from CNN model and features are stored in a vector. This query image feature vector will be passed to the SVM, which has already been trained using the Corel dataset. The pre-trained SVM module will calculate the distance between the features of the query image and the feature of the entire dataset. Retrieved images are displayed based on similarity index that is, distance values with respect to query image. Top 10, top 20 and so on are displayed as a part of retrieved process.

Feature Extraction

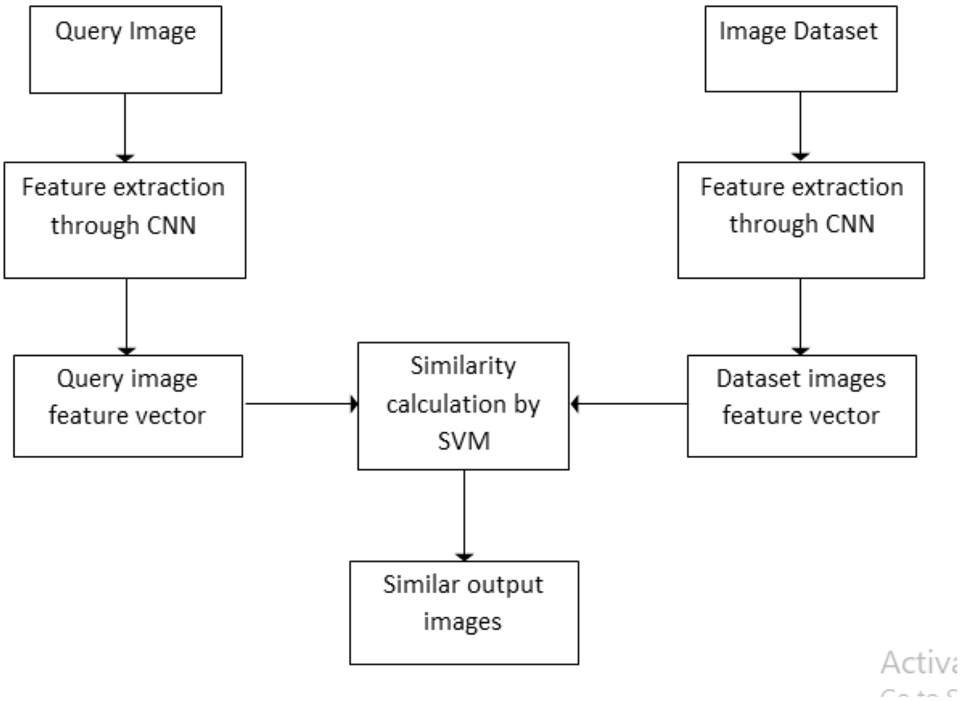
VGG16 layered CNN model is used for extraction of features of data set. Query image submitted will be extracted when user submits to a system. Figure 2 represents theVGG16 layered architecture. It consists of twelve convolutional layers, followed by maximum pooling layers and then four fully connected layers and finally a softmax classifier.

The original images are of size 384 × 256 or 256 × 384 which will be converted to 224 × 224 pixels and then fed to the CNN model. The data goes through the following layers in CNN module and image is transformed into different sizes leading to extraction of features.

- First and second layers

  VGG16 CNN model takes 224 × 224 × 3 RGB image as input. The image is passed through 1st and 2nd convolu-

Fig. 1 Architecture of CBIR System


Query Image
Image Dataset
Feature extraction through CNN
Feature extraction through CNN
Query image feature vector
Similarity calculation by SVM
Dataset images feature vector
Similar output images
Activ

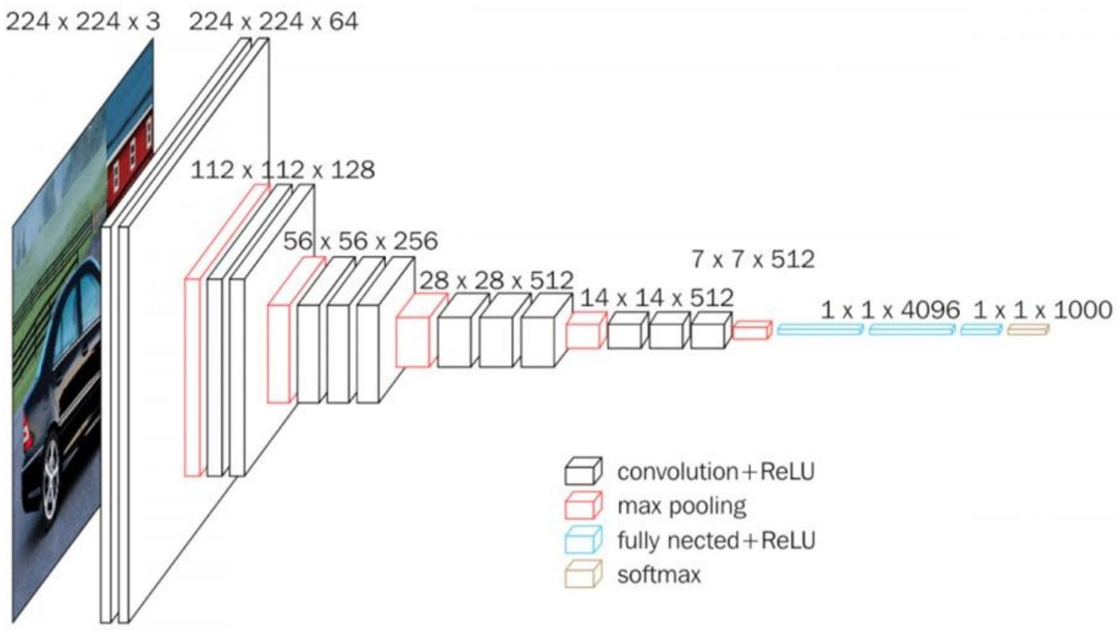


Fig. 2 VGG16 layered architecture

tional layers with 64 feature maps or filters of size 3×3 and same pooling with a stride of 14. The image dimensions change from 224×224 × 3 to 224×224 × 64.

Then the maximum pooling layer is applied with a filter size 3×3 and with a stride of 2. The resulting image dimensions will be changed from 224×224 × 64 to 112×112 × 64.

- Third and fourth layers

The image is passed through third and fourth convolution layers with 128 feature maps having filters of size 3×3 and stride of 1. Then maximum pooling layer is applied with filter size 3×3 and with a stride of 2. The output image is reduced to 56×56 × 128.

- Fifth and sixth layers

Then the image is passed through fifth and sixth convolution layers with 256 feature maps having a filter size of 3×3 and a stride of 1. Then maximum pooling layer is applied with filter size 3×3 and stride of 2 and has 256 feature maps.

- Seventh to twelfth layer

  Seventh to twelfth layers consist of 2 sets of three convolutional layers followed by a maximum pooling layer. All the three convolutional layers have 512 filters of size 3×3 and with a stride of 1. The final output image will be reduced to 7×7 × 512.

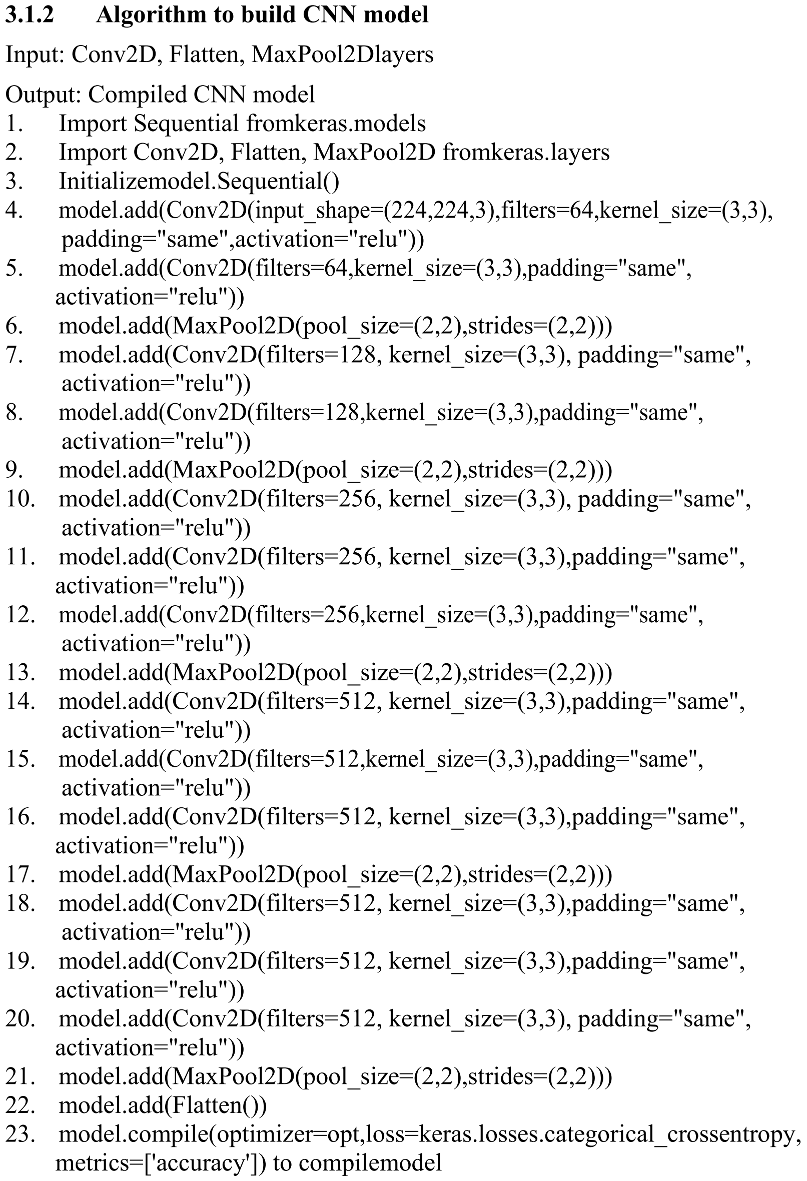
- Thirteenth Layer

  This final layer is a fully connected flatten layer to flatten the output with 25,088 feature maps each of size 1×1.

ReLU Activation Function

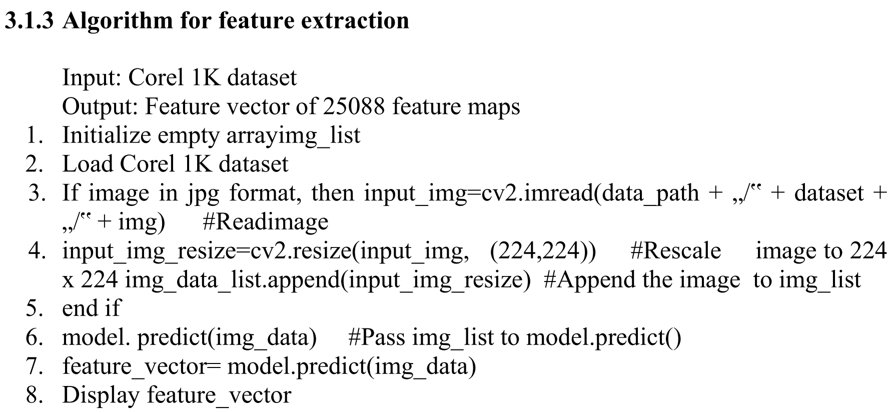
ReLU's purpose is to introduce non-linearity in the CNN model. ReLU is linear for all the positive values, and zero for all negative values. It only activates a node if the input is above a certain quantity, while the input is below zero, the output is zero that is node value is determined as A(x) = max (0, x).

The CNN model which consists of above layers is compiled using model. compile(). Then the image array is passed to model. Predict() to get the feature vector output of the flattening layer.



3.1.2 Algorithm to build CNN model

```
Input: Conv2D, Flatten, MaxPool2Dlayers
Output: Compiled CNN model
1.  Import Sequential fromkeras.models
2.  Import Conv2D, Flatten, MaxPool2D fromkeras.layers
3.  Initializemodel.Sequential()
4.  model.add(Conv2D(input_shape=(224,224,3),filters=64,kernel_size=(3,3),
    padding="same",activation="relu"))
5.  model.add(Conv2D(filters=64,kernel_size=(3,3),padding="same",
    activation="relu"))
6.  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
7.  model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same",
    activation="relu"))
8.  model.add(Conv2D(filters=128,kernel_size=(3,3),padding="same",
    activation="relu"))
9.  model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
10. model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same",
    activation="relu"))
11. model.add(Conv2D(filters=256, kernel_size=(3,3),padding="same",
    activation="relu"))
12. model.add(Conv2D(filters=256,kernel_size=(3,3),padding="same",
    activation="relu"))
13. model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
14. model.add(Conv2D(filters=512, kernel_size=(3,3),padding="same",
    activation="relu"))
15. model.add(Conv2D(filters=512,kernel_size=(3,3),padding="same",
    activation="relu"))
16. model.add(Conv2D(filters=512, kernel_size=(3,3),padding="same",
    activation="relu"))
17. model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
18. model.add(Conv2D(filters=512, kernel_size=(3,3),padding="same",
    activation="relu"))
19. model.add(Conv2D(filters=512, kernel_size=(3,3),padding="same",
    activation="relu"))
20. model.add(Conv2D(filters=512, kernel_size=(3,3), padding="same",
    activation="relu"))
21. model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))
22. model.add(Flatten())
23. model.compile(optimizer=opt,loss=keras.losses.categorical_crossentropy,
    metrics=['accuracy']) to compilemodel
```



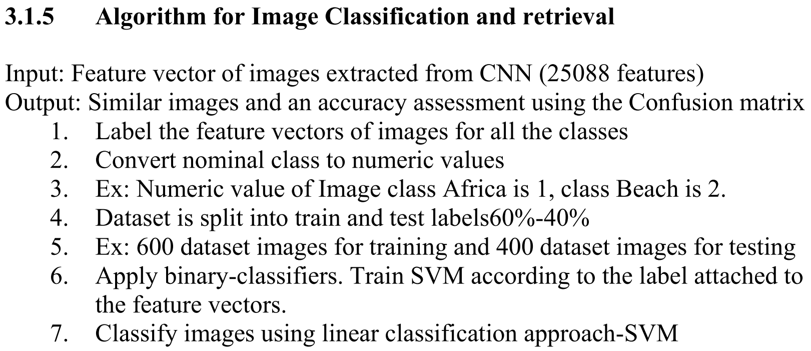
3.1.3 Algorithm for feature extraction

```
    Input: Corel 1K dataset
    Output: Feature vector of 25088 feature maps
1.  Initialize empty arrayimg_list
2.  Load Corel 1K dataset
3.  If image in jpg format, then input_img=cv2.imread(data_path + „/" + dataset +
    „/" + img)    #Readimage
4.  input_img_resize=cv2.resize(input_img,  (224,224))   #Rescale   image to 224
    x 224 img_data_list.append(input_img_resize)  #Append the image  to img_list
5.  end if
6.  model. predict(img_data)    #Pass img_list to model.predict()
7.  feature_vector= model.predict(img_data)
8.  Display feature_vector
```

Image Classification and Retrieval

Support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The obtained feature vector is given as input to the SVM, which calculates the distance between each image of the dataset and the query image. Binary SVM classifiers are also called as one-versus-one method. Multi-classification with this method can be described as the task of constructing n (n–1)/2 binary SVMs, one classifier Cij for every pair of distinct classes, i.e. the ith class and the jth class, where i ε j, $i=1, \ldots, n$ $j=1, \ldots, n$. Each classifier Cij is trained with the samples in the ith class with positive labels, and the samples in the jth class with negative labels. Train SVM according to the label attached to the feature vectors.

Classification of images is done using linear Regression approach. Consider via set T of t training feature vectors $x_i \in R$^D, $i=1\ldots$t and the corresponding class labels $\in\{1\ldots t\}$, y_i (wT * x_i+b)-1 $y_i\in\{1\ldots t\}$.Where w is the hyper plane normal vector, b is the perpendicular distance between the hyper plane and the origin. In this case, we are training the shape and color features.



3.1.5 Algorithm for Image Classification and retrieval

```
Input: Feature vector of images extracted from CNN (25088 features)
Output: Similar images and an accuracy assessment using the Confusion matrix
    1.  Label the feature vectors of images for all the classes
    2.  Convert nominal class to numeric values
    3.  Ex: Numeric value of Image class Africa is 1, class Beach is 2.
    4.  Dataset is split into train and test labels60%-40%
    5.  Ex: 600 dataset images for training and 400 dataset images for testing
    6.  Apply binary-classifiers. Train SVM according to the label attached to
        the feature vectors.
    7.  Classify images using linear classification approach-SVM
```

Testing

1. Select query Image = ×.jpg, from dataset
2. Find the query image class label in which it belongs.
3. Perform classifications of the images.
4. Perform binary predictions for the selected query image.
5. Retrieve the first n similar images to the input image from class x using voting.

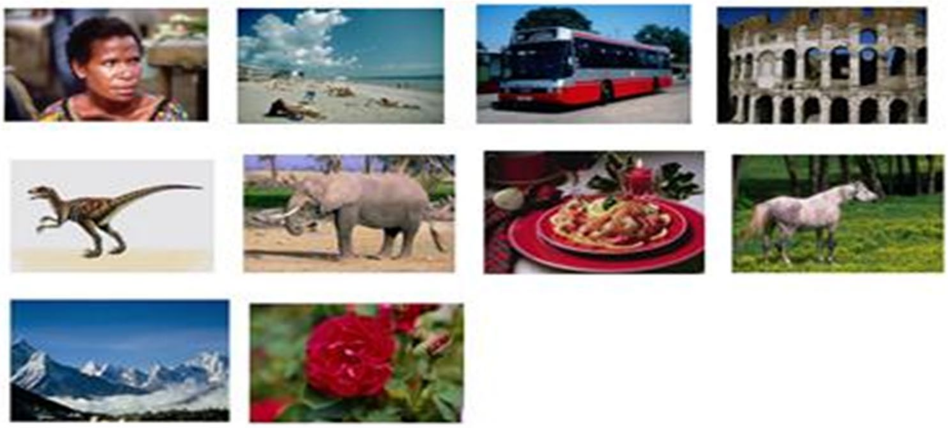


Fig. 3 Query set of proposed work

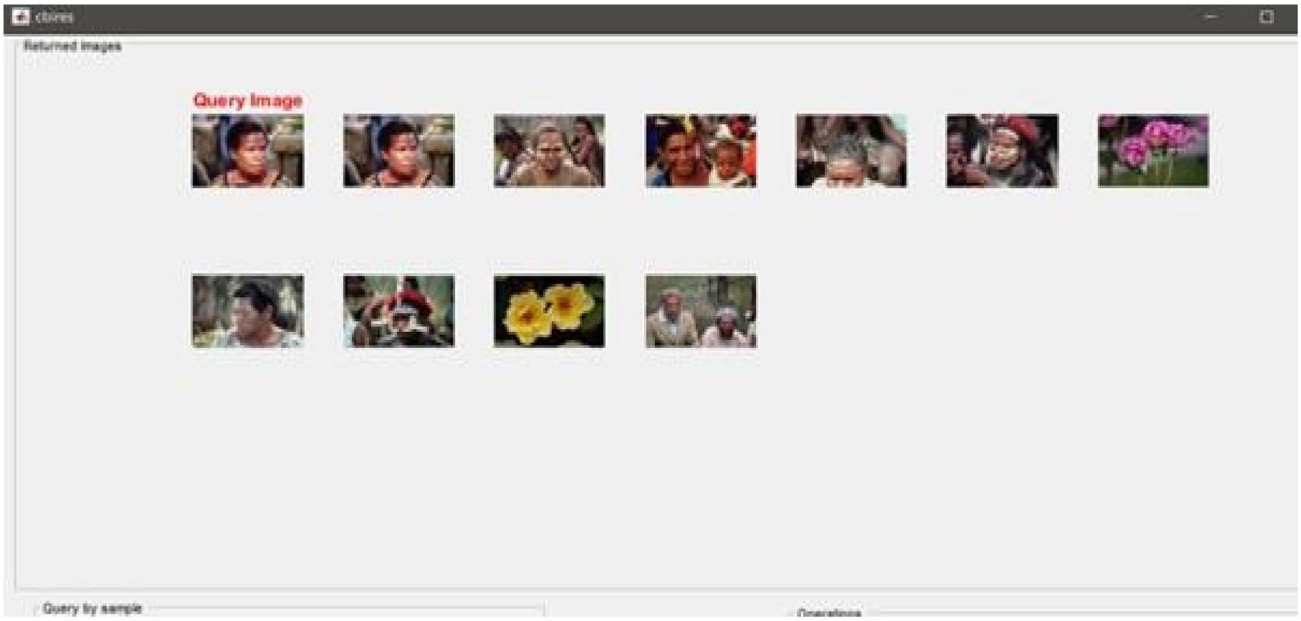
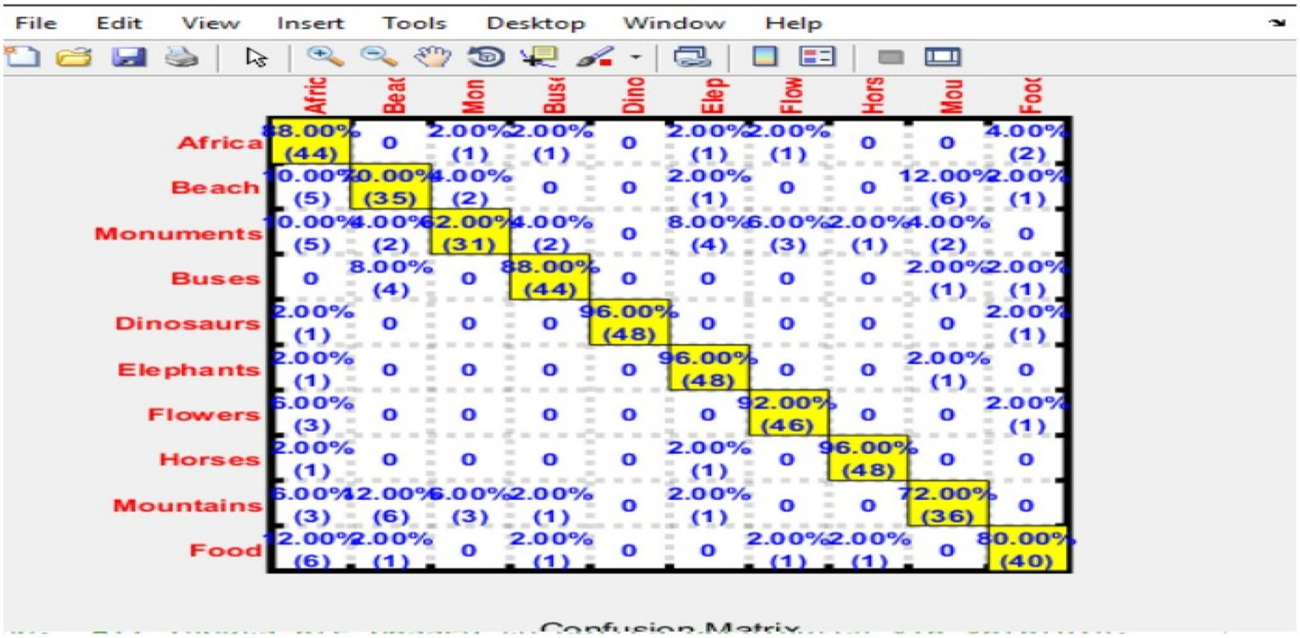


Fig. 4 Top 10 Images retrieved for class Africa

Results and Discussion

The query set consists of randomly chosen 10 images, first image from each category. Figure 3 shows the query set of different categories for proposed work. Figure 4 represents the top 10 images retrieved for class Africa. In the below Fig. 5, confusion matrix, the class with highest accuracy percentage is Horses with 96% and its true positive value is 48 and the class with lowest accuracy percentage is Mountains with 62% and true positive value 31.

The experimental results Fig. 6, shows top 10 images retrieved. Experiments conducted on 10 classes of the Corel 1 K dataset are shown below in Table 1. It shows the precision values of each class when we retrieve 10 images, 15 images, and 20 images from the database respectively.



	Afric	Beac	Mon	Bus	Dino	Elep	Flow	Hors	Mou	Foo
Africa	88.00% (44)	0	2.00% (1)	2.00% (1)	0	2.00% (1)	2.00% (1)	0	0	4.00% (2)
Beach	10.00% (5)	70.00% (35)	4.00% (2)	0	0	2.00% (1)	0	0	12.00% (6)	2.00% (1)
Monuments	10.00% (5)	4.00% (2)	62.00% (31)	4.00% (2)	0	8.00% (4)	6.00% (3)	2.00% (1)	4.00% (2)	0
Buses	0	8.00% (4)	0	88.00% (44)	0	0	0	0	2.00% (1)	2.00% (1)
Dinosaurs	2.00% (1)	0	0	0	96.00% (48)	0	0	0	0	2.00% (1)
Elephants	2.00% (1)	0	0	0	0	96.00% (48)	0	0	2.00% (1)	0
Flowers	6.00% (3)	0	0	0	0	0	92.00% (46)	0	0	2.00% (1)
Horses	2.00% (1)	0	0	0	0	2.00% (1)	0	96.00% (48)	0	0
Mountains	6.00% (3)	12.00% (6)	6.00% (3)	2.00% (1)	0	2.00% (1)	0	0	72.00% (36)	0
Food	12.00% (6)	2.00% (1)	0	2.00% (1)	0	0	2.00% (1)	2.00% (1)	0	80.00% (40)

Confusion Matrix

Fig. 5 Confusion matrix for class Africa

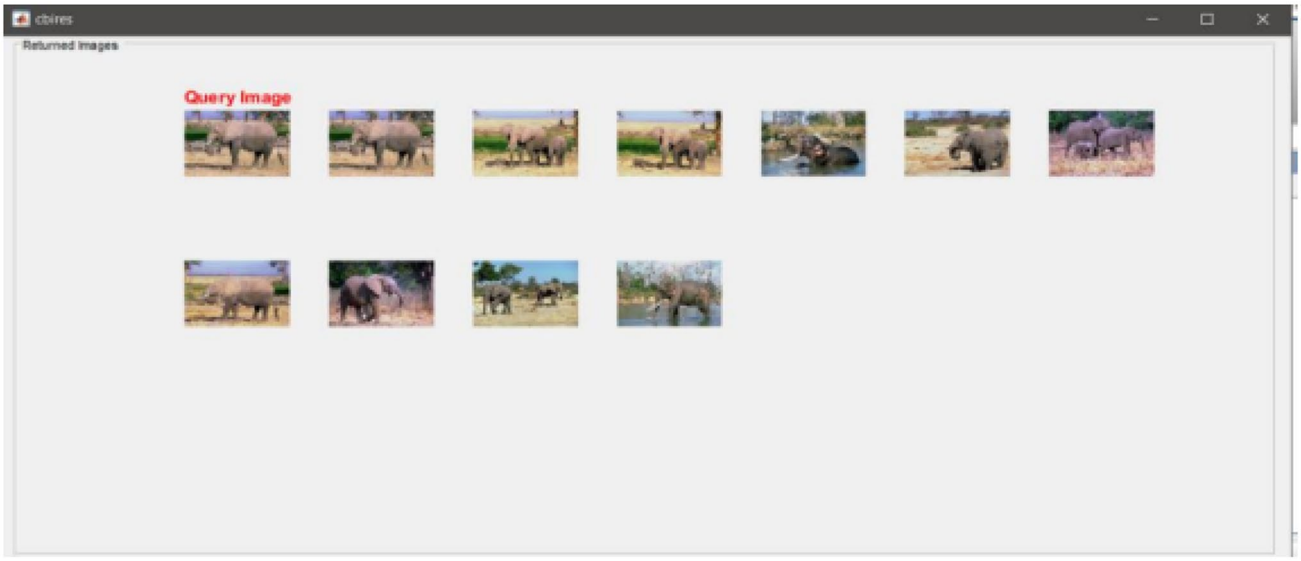


Fig. 6 Top 10 Images retrieved for class Elephant

The precision of class Elephant obtained by retrieving 10 similar images is 83.86%. Figure 7 shows the confusion matrix and it shows that class with highest accuracy percentage is Dinosaurs with 100% and its true positive value is 50 and the class with lowest accuracy percentage is Mountains with 52% and true positive value 26.

Table2 shows the comparison of average precision values of various previously implemented CBIR models like HSV (hue, saturation, value), GLCM (gray-level co- occurrence matrix), SIFT (scale-invariant feature transform) and CNN with CNN- SVM model. The average precision of the proposed CNN-SVM model is higher than the average precision of HSV, GLCM, SIFT, CNN models respectively.

Table 1 Results of different classes obtained with the Corel dataset

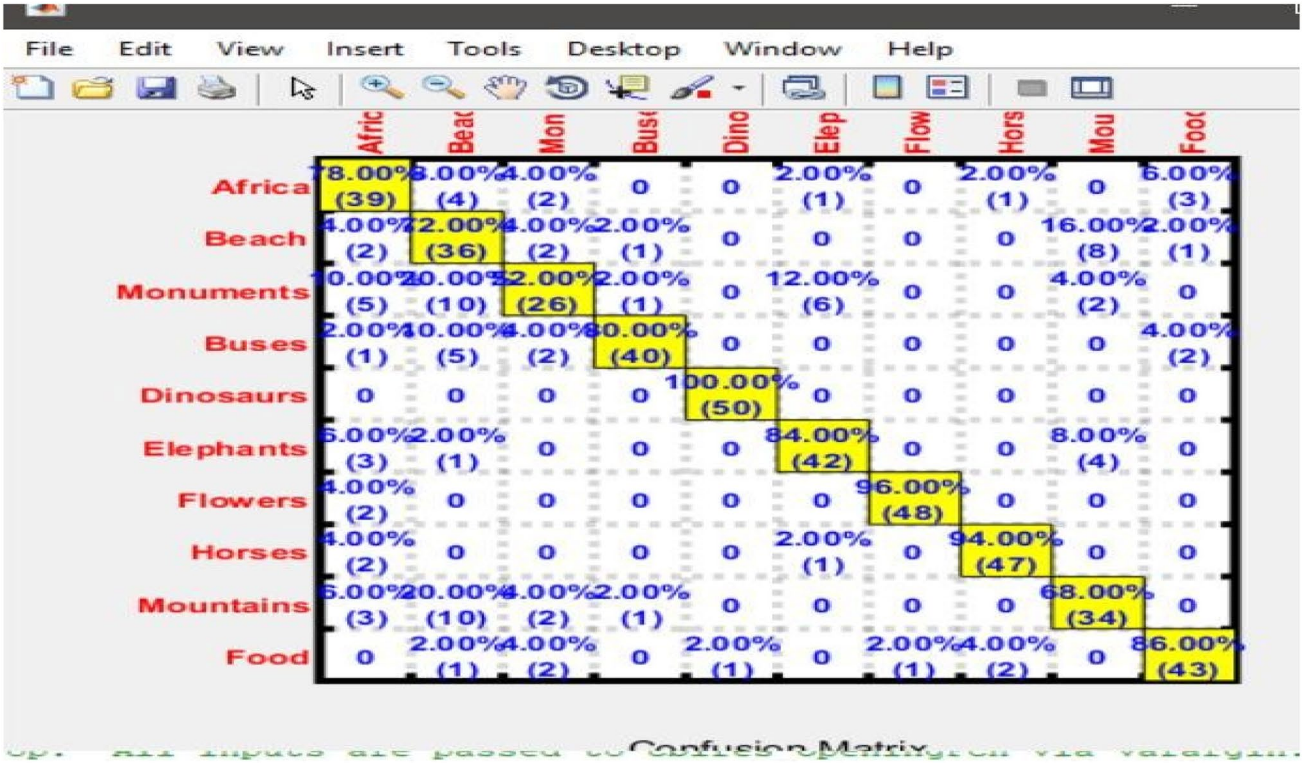
Category	Obtained precision with number of similar images retrieved (in %)		
	10 images	15 images	20 images
Africa	84.00	83.40	84.00
Beaches	84.06	85.40	82.86
Buildings	83.53	82.53	85.26
Buses	82.73	83.26	83.06
Dinosaurs	83.2	83.86	83.4
Elephants	83.86	84.46	83.53
Flowers	83.08	83.93	83.6
Horses	84.13	83.4	83.4
Mountains	83.8	83.73	82.33
Food	83.73	82.73	82.26
Total	83.61	83.67	83.37

Table 2 Comparison of CNN- SVM model with existing models

Model	HSV [2]	GLCM [2]	SIFT [2]	CNN [2]	CNN-SVM
Precision	0.43	0.39	0.53	0.79	0.84

Conclusion

The proposed method of Content-Based Image Retrieval system using CNN for feature extraction and SVM for classification provided an average efficiency of 83.5%. The use of SVM helped to reduce the time required to retrieve the results. The experimental results were compared with other previously proposed models like HSV (hue, saturation, value), GLCM (gray-level co-occurrence matrix), SIFT (scale-invariant feature transform), and CNN. The average precision of our proposed system is higher than the existing proposed models, which is effective and promising.



	Afric	Beac	Mon	Buse	Dino	Elep	Flow	Hors	Mou	Food
Africa	78.00% (39)	8.00% (4)	4.00% (2)	0	0	2.00% (1)	0	2.00% (1)	0	6.00% (3)
Beach	4.00% (2)	72.00% (36)	4.00% (2)	2.00% (1)	0	0	0	0	16.00% (8)	2.00% (1)
Monuments	10.00% (5)	20.00% (10)	52.00% (26)	2.00% (1)	0	12.00% (6)	0	0	4.00% (2)	0
Buses	2.00% (1)	10.00% (5)	4.00% (2)	80.00% (40)	0	0	0	0	0	4.00% (2)
Dinosaurs	0	0	0	0	100.00% (50)	0	0	0	0	0
Elephants	6.00% (3)	2.00% (1)	0	0	0	84.00% (42)	0	0	8.00% (4)	0
Flowers	4.00% (2)	0	0	0	0	0	96.00% (48)	0	0	0
Horses	4.00% (2)	0	0	0	0	2.00% (1)	0	94.00% (47)	0	0
Mountains	6.00% (3)	20.00% (10)	4.00% (2)	2.00% (1)	0	0	0	0	68.00% (34)	0
Food	0	2.00% (1)	4.00% (2)	0	2.00% (1)	0	2.00% (1)	4.00% (2)	0	86.00% (43)

Fig. 7 Confusion matrix for class Elephant

Compliance with Ethical Standards

Conflict of Interest There is no conflict of interest.

References

1. Wan J, Wang D, Hoi SCH, Wu P, Zhu J, Zhang Y, Li J. Deep learning for content-based image retrieval: a comprehensive study. In: ACM international conference on multimedia, November 3–7. 2014.
2. Huang W, Qiang W. Image retrieval algorithm based on convolutional neural network. In: Selected paper from Common Sense Media Awards. 2017.
3. Wang J, Li J, Wiederhold G. Simplicity: semantics-sensitive integrated matching for picture libraries. IEEE Trans Pattern Anal Mach Intell. 2001;23(9):947–63.
4. Chen Y, Wang JZ, Li J. FIRM: fuzzily integrated region matching for content-based image retrieval. In: Proceedings of the ninth ACM international conference on multimedia. ACM. 2001. p. 543–545.
5. http://wang.ist.psu.edu/IMAGE.
6. Saritha RR, Paul V, Ganesh Kumar P. Content based image retrieval using deep learning process. Cluster Comput 2019;22(2):4187–200.
7. Mohamed O, El Asnaoui K, Mohammed O, Brahim A. Content-based image retrieval using convolutional neural networks. Original paper in Lecture Notes in Real-Time Intelligent Systems book. 2019. http://wang.ist.psu.edu/IMAGE. Accessed Jan 2001.
8. Desai P, Pujari J, Goudar RH. Image retrieval using wavelet based shape features. J Inform Syst Commun (JISC) 2012;3:1162–166.http://www.bioinfo.in/contents.php?id=45.
9. Desai P, Pujari J, Parwatikar S (2011) Image retrieval using shape feature: a study. In: International conference on computaional intelligence and information technology (CIIT 2011), ACEEE, CIIT 2011, CCIS 250. Berlin: Springer; 2011. p. 817–21.
10. Desai P, Pujari J,Ayachit NH, Kamakshi Prasad V. Content based image retrieval using hexagonal resampling and detection of ailments in MRI scans of Brain. In: Third international conference on computational intelligence and information technology, CIIT 2013 ACEEE. Elsevier. 2013.
11. Desai P, Pujari J, Kinnikar A. Performance evaluation of image retrieval systems using shape feature based on wavelet transform. In: IEEE second international conference on cognitive computing and information processing CCIP 2016, India. IEEE. 2016. p. 1–5. https://doi.org/10.1109/CCIP.2016.7802876.
12. Desai P, Pujari J, Kinnikar A. An image retrieval using combined approach wavelets and local binary pattern. In: International conference on informatics and analytics (ICIA-16), Aug 25th and 26th 2016, Department of computer science and engineering, Pondicherry engineering college, India. ACM digital library within its international conference proceedings series. 2016. https://doi.org/10.1145/2980258.2980404.
13. Desai P, Pujari J, Ayachit NH, Kamakshi Prasad V (2013) Classification of archaeological monuments for different art forms with an application to CBIR IEEE. In: International conference on advances in computing, communications and informatics (ICACCI-2013). 2013. p. 1108–1112. https://doi.org/10.1109/ICACCI.2013.6637332.
14. Sujatha C, Chivate AR, Tabib RA, Mudenagudi U. Multilevel framework for summarization of surveillance videos. In: International conference on signal and image processing (ICSIP). 2014. p. 265–270.
15. Sujatha C, Mudenagudi U. Gaussian mixture model for summarization of surveillance videos. In: National conference on computer vision, pattern recognition, image processing and graphics (NCVPRIPG). 2015. p. 1–4.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.