Cbir system using integrated dwt and cnn architecture

K. Ramanjaneyulu¹, K. Veera Swamy², Ch. Srinivasa Rao³

¹Research Scholar, JNTU K, Associate Professor, Department of ECE, QISIT, Ongole.
²Professor, Department of ECE, Vasavi College of Engineering, Hyderabad
³Professor, Department of ECE, JNTUKUCEV, Vizianagaram
E-Mail:--chsrao.ece@jntukecv.ac.in

Abstract. Because of the rapid growth of multimedia technologies related to the Internet, a very large number of images are available; therefore, an efficient image retrieval method is required. In general, image-based retrieval is more effective than text-based retrieval because the features of an image are richer in information. In this article, we present an algorithm for CBIR using an elegant integration of DWT and deep learning neural network architecture, which yields the better outcomes in the area of machine learning and computer vision because of its excellent ability to extract features from an image. For calculating the resemblance between a query image and the database images, we used the Euclidean distance metric. The performance of the proposed method was evaluated in terms of precision and recall.

Key words: CBIR, Precision, Recall, Euclidean distance, DWT, CNN.

1. Introduction

A large number of images are required in many areas, such as medical treatment, satellite and video satellite data, and digital judicial systems, and these images have been made available by the development of multimedia technologies. This has led to an increasing demand for a system that can save and download multimedia data effectively. Thus far, many systems have been developed to meet these requirements of storing and downloading multimedia files. The extraction system (TBIR) generally extracts a text image using a search based on an automatic or manual address image. A simple TBIR system finds the database for analogous text around the image provided in the query string [2]. The most popular TBIR system is Google search. A text-based system is fast, because string matching consumes relatively less computational time. However, at times, it is difficult to display all the visible contents of an image using words, and thus, the TBIR system may produce inappropriate results. In addition, the annotations of the image do not always exist. The other way for searching images to overcome the drawback in the TBIR system is CBIR. The CBIR system uses image content described as low-level functions such as colours, textures, and spatial points to denote images in databases. The system takes an analogous image when a sample image or drawing is shown to it. Thus, querying eradicates the need to describe the visual content of the image in text and is close to the people’s perception of visual data.

In normal CBIR systems, low-level contents (shape, colour, and texture) are represented as a multi-dimensional vector. The feature database is constructed from the images in another database. The extraction process is initiated when a user queries the system using an example image or a drawing of the object. For a query image, the feature vector is also computed using the same process that is used to create an entry database. Furthermore, a similarity measure is used to calculate the distance between the query image and the feature vector of an image database [2]. Finally, the extraction is performed using an index scheme that makes it easier to find an
The rest of this paper is organised as follows: DWT and CNN are discussed in brief in Section II. In Section III, the proposed method is introduced. In Section IV, the experimental results are presented. Finally, the concluding remarks are presented in Section V.

2. DWT & CNN architecture:

DWT is commonly used in the domain of multi-scale image processing [5]; it decomposes the given image into four sub-images: 1. an approximated image, 2. a vertical (Dv) detailed image, 3. a horizontal (Dh) detailed image, and 4. a diagonal (Dd) detailed image. The detailed images calculate the changes along the rows (vertical edges), columns (horizontal edges), and diagonals (diagonal edges). To provide reduced and meaningful information related to the desired image, more than one decomposition level may be utilised. Then, the approximated image is decomposed again into wavelet sidebands. In general, two or more decomposition levels are used, and the obtained approximated image is used as a feature vector. Figures 2(a–d) show the original image, level-one decomposition, level-two decomposition, and level-three decomposition, respectively.

In contrast, CNN compares an image block by block. These blocks appear as the features of an image. By calculating the features of an image at roughly the same position, CNN yields a good result when used as a retrieval system. Each feature in an image looks like a mini image. A feature matches the common aspects of an image. In CNN, when the network does not know where to extract the images features from, it tries using a few possible positions for extracting the image features of an image and functions as a filter/mask. In this process, we use a convolutional layer.

A. Convolution Layers
Convolution is basically an integration of two functions and shows how one function modifies the shape of the other function. The image is in the form of pixels and can be represented in the matrix form. Fig. 3 shows the operation of the convolution layer. By using the sliding window technique, we multiplied the feature detector with the input image. The size of the feature detector used depends on the size of the input image. Although there is some information loss and we do not obtain all the values of the resulting matrix, the purpose of the feature detector is to detect certain features from certain integral parts of the image.

For obtaining the first convolutional layer, we generated various feature maps. On the basis of the values of the feature detector, we obtained certain image features, such as sharpen, blur, edge enhance, edge detect, and emboss. To optimise the convolution layer, we used a shared weights model to reduce the number of unique weights used to train the model per layer and consequently, perform the matrix calculations.

The rectifier neuron, e.g. rectified linear unit (ReLU) is one of the several keys to the recent achievement of deep networks. When compared with conventional sigmoid functions, ReLU accelerates the convergence of training and leads to better solutions. In fact, building models using the sigmoid functions forces the model to some knowledge loss. Instead of the sigmoid functions, deep learning networks use ReLUs for the hidden layers. These networks output 0 if the input is less than 0; otherwise, the output is raw data. However, when dealing with the classification problems, ReLUs cannot help much. To overcome this, we used a softmax function. The softmax function compresses the output of each unit to be between 0 and 1 and is equivalent to a categorical probability distribution.

When dealing with the probabilistic classification, it is in fact better to use the cross-entropy loss than the squared error. By reducing the cross-entropy loss, we could maximise the confidence in the correct class.

B. Pooling Layers

Spatial invariance is obtained by using pooling layers, which minimises the resolution of the feature maps and controls the over-fitting. Each feature map of the previous layer corresponds to a pooled feature map. This can be achieved by using the sliding window technique. Pooling is also called down-sampling.

The technique consists of two pooling operations: subsampling and max pooling. We used the max pooling technique. The subsampling function can be expressed as follows:

\[
    a_j = \tanh \left( \beta \sum_{N=n^x n^y} a_i n^x n^y + b \right)
\]

It takes the average of all the inputs in the window function and multiplies it by a scalar \( \beta \). To the results, a bias \( b \) is added. The resulting value is then passed through the non-linearity.

The max pooling function can be represented as follows:
\[ a_j = \max_{n=1}^{N} (a_i^{m,n} u(n, n)) \]

It takes the maximum input value in the window function. The result is a feature map with a very low resolution. Thus, the size of the input image was reduced. The overlapping pooling window step size affected the recognition confidence. In the pooling layer, the max pooling method was used. An experimental comparison of the sub-sampling method to the max-pooling method revealed good results for capturing the spatial invariance in the image.

CNNs consist of pooling layers which condense in a similar kernel and map the yields of adjacent groups of neurons. Customarily, the areas are bridged by utilizing the nearby non-overlap pooling units. To be progressively exact, a pooling layer has a network of pooling units divided by s pixels, and each pooling unit outlines an area having size focused at the area of the pooling unit\[4\]. During training, we observed the overlapping pooling models and found that they became marginally increasingly difficult to over-fit.

Flattening is a procedure for changing over the convolutional part of the CNN output into a 1D feature vector. Fully-associated (Dense) layer, and have full activation with all authorizations, in the earlier layer as we found in normal systems. Their activation can be determined in a straightforward manner by using matrix multiplication followed by the corresponding bias offset. All the layers of the CNN model can be prepared by using the back-propagation algorithm. We used a similar system for error propagation and weight adaptation in fully connected, convolutional, and pooling layers.

3. Proposed algorithm:

A block diagram of the proposed method is shown below along with the detailed algorithm of the method

1. Read an input RGB colour image from a collection of images in the Corel database.
2. Resize the image into 128 × 128 pixels, and convert the input RGB image into a grey-scale image.
3. Apply 1-level DWT.

Figure 4: Proposed method for feature extraction
4. Divide the image using DWT into four sub-bands, namely LL band, LH band, HL band, and HH band.
5. Consider the LL band of the image.
6. Apply a CNN architecture training model to the LL band and construct the feature vector. Create a feature vector of the given query image, which is similar to the feature vector of the database images.
7. Use the Euclidean distance measure to compare the distance between the feature vectors of the query image and the database images. The image with the smallest distance between the feature vector of the query image and that of the database images is the most similar image to the query image

Figure 5. CNN training model

A. CNN Architecture
The architecture of the proposed neural network consists of three layers: the convolution layer, the pooling layer, and the fully connected layer. In addition, to extract the feature information from more different aspects, more convolution kernels are used, but doing so leads to a larger number of parameters and a very high training time for the neural network. When compared with the non-saturated non-linearity, the saturated nonlinearity is slow. Therefore, nowadays, in the CNN domain, an activation function is used. The CNN model mainly uses ReLUs. In CNN, the extraction of image features is performed layer by layer; i.e. the study, extraction, and abstraction of the image are performed layer by layer. If for the network architecture the extracted feature is more then it means that gives more abstract description of the image, so it is the more useful contribution to the image retrieval. In AlexNet, the Fc8 layer feature information is extracted for the image retrieval process
Table I: Step-by-step sizes of CNN training model

| Layer Type   | Image Size | Feature Maps | Kernel Size |
|--------------|------------|--------------|-------------|
| Input        | 128x128    | 1            | -           |
| Convolutional| 128x128    | 16           | 3x3         |
| Max pooling  | 64x64      | 16           | 2x2         |
| Convolutional| 64x64      | 32           | 3x3         |
| Max pooling  | 32x32      | 32           | 2x2         |
| Convolutional| 32x32      | 64           | 3x3         |
| Max pooling  | 16x16      | 64           | 2x2         |

4. Experimental Results

We used the Corel-1K database for testing the CNN architecture. This database consists of 10 categories of 1000 RGB colour images

![Figure 6. Corel Database](image-url)
Figure 7: Retrieval results of proposed technique when input query image was from (a) Rose and (b) Horse.

Table II: Performance measures of proposed technique

| categories | Recall | Precision | F-SCORE |
|------------|--------|-----------|---------|
| People     | 84.05  | 81.91     | 82.97   |
| Beach      | 84.28  | 74.62     | 79.16   |
| Buildings  | 78.57  | 80.84     | 79.69   |
| Buses      | 97.14  | 88.27     | 92.49   |
| Dinosaurs  | 97.14  | 98.55     | 97.84   |
| Elephants  | 85.71  | 89.54     | 87.58   |
| Flowers    | 92.71  | 96.95     | 94.73   |
| Horses     | 94.28  | 91.65     | 92.95   |
| Mountains  | 80     | 91.62     | 85.42   |
| Food       | 85.71  | 88.23     | 86.95   |
| Overall    | 87.96  | 88.22     | 88.09   |

Figure 8: Precision Comparison of various CBIR methods

5. Conclusion

In this paper, we presented an algorithm to retrieve similar images using an integration of the DWT and CNN architectures. We conducted an experiment on the Corel-1K database and achieved precision and recall of 88.22% and 87.96%, respectively, for the correctly classified images in the test data. The Euclidean distance measure was used as the similarity metric to retrieve images similar to the query image from the database. Moreover, we implemented an
integration of DWT and CNN, because if only CNN were used to train the data, it would have taken more time and using traditional algorithms alone would have resulted in compromising the accuracy or performance.

6. REFERENCES

[1] AShah, Amjad & Naseem’, Rashid & , Sadia & Iqbal, Shahid’ & Arif Shah’, Muhammad. (2017).” Improving CBIR accuracy using convolutional neural network. for feature extraction”. 1-5. 10.1109/ICET.2017.8281730.

[2] Vipin Tyagi. "Content-Based Image Retrieval" , Springer Nature, 2017

[3] scholarworks.rit.edu

[4] Y Gao’, He Li’, J Dong’, Guang F. "A deep convolutional network for medical image super-resolution", 2017 Chinese Automation Congress (CAC), 2017

[5] Nam. Vu, Cuong P. “Traffic Incident Recognition Using Empirical Deep Convolutional Neural Networks Model”, Springer Nature, 2018

[6] W Zhou’, , Zhen. feng S’, and Qimin C’. "Deep feature representations for high-resolution remote sensing scene classification”, 2016 “4th International workshop on earth observation & remote sensing applications.” (EORSA).- 2016

[7] Corel database http://wang.ist.psu.edu/~jwang/test1.tar

[8] en.wikipedia.org/A.

[9] Alzu’bi’, A. Amira.’., and N Ramzan’. , “Semantic content-based image retrieval: A comprehensive study,” ‘journal of Visual communication & image Representation’, vol. 32, no. July, pp. 20–54, 2015.

[10] J. Z. Zhang”, “CBIR using colour & edge direction features.”,2010 2 nd International Conference on Advanced Computer Control., pp. 459–462, 2010’.

[11] H Liu, ‘Baoan Li’, Xueqiang Lv, Yue Huang. "Image Retrieval Using Fused Deep Convolutional Features", Procedia Computer Science, Volume 107 Issue C, PP.749-754 April 2017, ISSN: 1877-0509 EISSN: 1877-0509

[12] X.-Y. Wang’, B.-B. Zhang’, “Content-based image retrieval by integrating color and texture features,” Multimedia tools and Applications’, vol. 68, no. 3, pp. 545–569, 2012.

[13] Chhaya Singh, Brijesh Kumar, Anju Rani, Kamla Dhyani, Raj Singh. "Biodiversity and conservation Ferns Diversity in different forests of Dehradun district ". International Journal of Pharmacy Research & Technology 7 (2017), 01-07.

[14] J J Foley, Paul Kwan. "Chapter 583- Feature Extraction in Content-Based Image Retrieval”, IGI Global, 2015.