Deep learning feature map for content based image retrieval system for remote sensing application

Swati Jain*
CSE Department, Institute of Technology,
Nirma University,
Ahmedabad, Gujarat, India
Email: swati.jain@nirmauni.ac.in
*Corresponding author

Tanish Zaveri
EE Department, Institute of Technology,
Nirma University,
Ahmedabad, Gujarat, India
Email: Ztanish@nirmauni.ac.in

Kinjal Prajapati and Shailee Patel
CSE Department, Institute of Technology,
Nirma University,
Ahmedabad, Gujarat, India
Email: kinjal1292@gmail.com
Email: shelypatel@gmail.com

Abstract: This paper proposes a model for content based image retrieval system (CBIR), in which handcrafted feature set is replaced with feature set learnt from deep learning, convolutional neural network (CNN) for image retrieval. Feature map obtained from CNN is of high dimension, which makes the matching process expensive in terms of time and computation. Hence to recapitulate information in smaller dimension statistical values of the feature maps are calculated. Statistical values like entropy and contrast are taken as characteristic value of each feature map, and these values are used as features for similarity measure in CBIR system. The retrieval performance are compared, when feature map from deep neural net is considered and when statistical values of the feature map are used. The performance parameter considered are normalised rank and number of relevant images retrieved. The proposed approach is experimented with UC Merced Landuse Landcover Datasatset and the results obtained establishes that statistical features give better results.

Keywords: CBIR; convolutional neural network; deep learning; feature map.

1 Introduction

Content based image retrieval (CBIR) system has been an active research area for more than two decades now, still the field is evolving and there is huge scope for researchers to establish it as a matured system. The CBIR system comprises of feature extraction of images followed by matching and retrieving relevant images in order of its similarity with the query image. Features of an image are extracted, considering image as pixel matrix and processed to represent the information. Feature representation plays very important role in CBIR systems to obtain relevant images. The handcrafted feature set for an image, may not exactly represent human perception, this is termed as semantic gap. For all these years since its inception, researchers have been working and proposed various feature representation techniques for CBIR systems, but this still remains a challenging issue because of the semantic gaps between low-level pixel information of images and high-level human perception. Although the handcrafted features are proven to be efficient for the systems but have limitations in terms of covering all the characteristics of an image.

Deep learning (DL) which is evolving as a promising branch of machine learning, demonstrates high-level abstractions using deep networks which has multiple non-linear transformations. DL networks acts like human brains and learns concepts similar to human perception. DL learns image features automatically and maps the images directly to the output using its domain knowledge; thus overcoming the limitations of handcrafted features.

In recent years, DL, specially convolutional neural network (CNN) has proven its capabilities in the field of image classification and pattern recognition, but its application in the field of CBIR is still limited. In this paper, the authors have demonstrated use of CNN for CBIR systems. The feature maps obtained using CNN are used for CBIR application.
2 Related work

Over the years since the advent of CBIR various features are defined and designed to capture distinguishing characteristics in an image. These features are grouped into colour, texture and edge. Amongst these three feature categories, colour feature has always been more obvious and intuitive. Different colour features which researchers have already experimented are colour coherence vector (CCV) Pass, Zabih and Miller (1997), colour moment Kodituwakku and Selvarajah (2004), colour histogram, and so on. Edge features are captures the edges in the image. The edge descriptors are better able to define the edges in an image. Some of the edge features are edge direction histogram and edge coherence vector Gao et al. (2008). Gabor filters Zhang et al. (2000) and co-occurrence matrix Haralick (1979) are like classic features which exist for years now. These feature sets have been used in texture identification so satellite images is one of the domain where this has been used. Both these texture descriptors are time tested in gray scale images. A step ahead to define the texture was done in by Erchan and morphological features are proposed in Aptoula (2012) Aptoula (2014). Morphological covariance as operator is used to find textures. Circular covariance histogram and rotation invariant point triplets (RIT) are morphological texture descriptors Aptoula (2014). There are many more feature sets introduced, the limitation of this feature sets is that they do not adapt with the image dataset. Also the feature set do not adapt with the concept of image classification in the same image dataset. A good survey of CBIR is presented in Liu et al. (2007) which indicated the limitation of low-level feature set and need of incorporating learning, for efficient CBIR systems. In the last few years, researchers have started working on a completely new dimension called DL, which is expected to change the way the authors perceive the working of CBIR systems. DL in neural networks (NNs) can be used for supervised learning, unsupervised learning and reinforcement learning Schmidhuber (2015). Human learns the patterns by sequentially directing attentions to various relevant portions of the available data unlike DL systems.

In today’s world with large set of images being added every moment, it is observed that hand-designed features will take huge effort and time in image retrieval. Whether it is audio, visual or text data, in every application features plays an important role. To address this issue, Andrew Ng and his team are actively working on DL algorithms which automatically learns feature representation (from unlabelled data) and thus saving from time-consuming human feature learning Ng (2015). Inspired by the cortical (brain) computations the algorithms based on massive artificial neural networks are developed. Andrew, as a part of his work found and led a project at Google for building a huge DL algorithm which resulted in highly distributed neural network with more than 1 billion parameters which were trained on 16,000 CPU cores. This resulted into self learning of high-level concepts like ‘cats’ from the unlabelled YouTube videos.

Andrea Vedaldi and Karel Lenc Vedaldi and Lenc (2014) have implemented DL using CNN for MATLAB. The toolbox was designed to emphasise on simplicity and flexibility. It contains easy to use functions and provides routines for computing linear convolutions with filters, feature pooling. It supports efficient computation for both CPU and GPU and also allows to train complex models on very large data sets such as Imagenet.

Another MATLAB toolbox for DL was implemented by Palm in Palm (2015). This toolbox uses only CPU and directly takes images as input and trains the network, this is specifically designed for MNIST dataset which is a characters dataset. This toolbox has been implemented for the application of handwriting recognition. An open problem is addressed in Wan et al. (2014) about how DL can be used in CBIR systems. In this paper, an attempt is made to address an open problem of deploying the newly discovered power of DL to solve a long existing problem of CBIR. The authors explored the possibility of using state-of-the-art DL techniques for learning feature representations and similarity measures.
For the purpose of speeding up the image retrieval processes Lin et al. (2015) has proposed a DL algorithm which generates binary hash codes. The binary codes are learnt by including an extra hidden layer for depicting the latent concepts that dominates various class labels. The experimentation is performed on various data sets and seen that the approach outperforms the state-of-the-art hashing methods.

2.1 Deep learning

Neural networks have been around from last 25–30 years and they have always been good for learning weights in the network with one hidden layer. In Mungara (2014) the authors used the three-layered neural network for training, i.e., colour, texture and edge features of the image and retrieved the relevant images based on this trained data. The set of features are trained resulting into fused feature set and then the similarity is measured over this fused feature. They have found improvement in the recall and retrieval rate after the classification using feed forward back propagation neural network.

There are several instances of using neural networks for learning the similarity Shereena and David (2014); Mungara (2014); Muneesawang and Guan (2001), and the concept. Efforts have been made for improving the results by using multi-layer neural network and similar algorithms like back propagation, but no considerable advantage has been reported so far, by even adding more than one hidden layer. One of the reasons being, the error which is back propagated through the layers with the jumbled nonlinear interactions makes it tough for the lowest layers to understand what needs to be done to reach what the output layer is targeting. DL is better than other approaches for speech, images, and so on, because of the series of hidden layers between the input and the output layer. Supervised learning has the images previously annotated and features are manually extracted, while in DL there are no annotations when it comes to images. Images are directly fed as input to the network and rest all is left on the network. Features are learnt gradually by the network and then the classification is done.

2.2 DL architecture

For the purpose of automatic feature learning, this paper concentrates on DL using CNNs. Unlike neural network, the CNN has its neurons arranged in three dimensions, i.e. width, height, and depth as can be seen from Fig 1.

\[
f[x, y] \ast g[x, y] = \sum_{n_1 = -\infty}^{\infty} \sum_{n_2 = -\infty}^{\infty} f[n_1, n_2] \ast g[x - n_1, y - n_2] \tag{1}
\]

**Figure 1** Neural network and convolutional neural network architecture (see online version for colours)
There are three main layers in CNNs.

- **Convolutional layer:** This layer convolves two signals, here it convolves the input image and the predefined kernels. The output maps obtained from convolution, act as different features of the images. As seen from equation 1 element wise multiplication and sum of a filter and a signal(image) is done.

- **Pooling layer:** To make the representation more manageable and small without losing much of information is done in this layer. Mostly down sampling is done to pool the representations.

- **Fully connected layer:** This layer gets the input data which is ready for classification. This layer classifies and gives the results accordingly. Here neurons have full connections to all the activations.

3 Implementation

Experimentation for the retrieval and matching are performed on the CNN architecture as specified in Agrawal (2014). The architecture defined in the toolbox has the architecture of Input-Conv-Pool-FC layers. That means the architecture has one convolutional layer. The number of kernels are 400 for convolutional layer and then down sampling is done on the output of the convolutional layer in the pool layer, which is fed to the fully connected network for classification. Our objective here is not classification, but to check the retrieval results the authors consider the intermediate results as features.

3.1 Flow

Figure 2 shows the architecture of CNN used for experimentation. The detailed explanation is as follows:

- The experimentation is done on UC Merced LULC dataset Yang and Newsam (2010).
- The network is trained on 2100 images of dataset containing 21 categories and tested on 2100 images of dimension $256 \times 256$.
- The input size (re-sized) is of $67 \times 67$. The input is processed in batches of 50 images at a time.
- The kernels of convolutional layer has receptive field of size $8 \times 8$ and such 400 kernels are present hence the output volume turns out as $400 \times 50 \times 60 \times 60$ for 50 images. As the image size reduces from $MXN$ to $(M-W+1)X(N-W+1)$ due to convolution with kernel size $WXW$.
- The pooled layer does sampling which has receptive field of $6 \times 6$ and hence the output is a $10 \times 10$ image., i.e., $(400 \times 50 \times 10 \times 10)$.
- The output of pool layer is fed into fully connected network and the classification is done. The accuracy of classification on our dataset is 98.7.
4 Experimentation

As described in subsection 3.1 all the 2100 images with given class, participate in the learning process. As an outcome of the learning processes, the weights are learnt in such a way that when convolved with the images gives appropriate features. An example of the feature maps obtained as the output of the convolution layer can be seen in Figure 3. The figure describes the output of each of the 400 kernels for image number 38 of agriculture category. These 400 blocks visible in the figure are the output of layer one, when a test image is operated with learned kernel of CNN. Figure 4 is the the set of some selected feature maps, and is quite apparent that the kernels of CNN generates the patterns, which are very similar to a human visualisation of the patterns. Figure 4 shows four different images of agricultural category and their patterns are captured by three different kernels numbered 20, 93 and 202. It can be concluded that these kernels have captured the patterns in a similar way as perceived by human.

Figure 3  Kernels obtained for query image of agriculture category
These output of the convolution layer is then sub sampled, which in turn becomes the input to the fully connected neural network. Input to the fully connected neural network is the feature set, and the authors have used this feature set for CBIR systems. The results obtained using the pooled data set (output of pooled layer) yields not so good results when measured in terms of rank and retrieval images as can be seen in Tables 1 and 2. To improve the retrieval result and to make the feature set compact, the authors have used the concepts contrast and entropy.

4.1 Contrast and entropy

Contrast of an image represents the difference in the intensity values in case of grey scale, which is represented by standard deviation. Standard deviation of the data set represents the measure of variation or dispersion from the average (mean or expected value). A low contrast indicates that the data points tend to be very close to the mean, whereas high contrast indicates that the data points are spread out over a large range of values. The standard deviation is calculated as Malik and Baharudin (2013)

$$\text{StandardDeviation} = \frac{1}{n-1} \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$  \hspace{1cm} (2)

Here $\bar{x}$ is the mean of the intensity values of all bins of four quantised histogram. It describes the brightness of the image. $x_i$ is the intensity value of each pixel and n is the total number of pixels.

Entropy measures the uncertainty of the intensity level distribution of the histogram bins. The high value of entropy indicates the distribution among the greater intensity levels.
in the image. The entropy is low for simple image and high for the complex image. The entropy can be calculated as Malik and Baharudin (2013)

\[
Entropy = -\sum_{i=1}^{n} (P(i) \log_2[P(i)])
\]  

(3)

Here P(i) stands for the probability of possible intensity i in its neighborhood pixel. Here the authors have calculated the entropy of 10 x 10 image kernels.

4.2 Results

The authors have compared the retrieval results using two approaches first one is the feature maps obtained using trained neural network termed as pooled featureset. The pooled feature set has dimensions 400 x 2100 x 10 x 10, hence 400 x 10 x 10 for each image of the data set, which is not feasible. The other feature set is the statistical parameter of the pooled feature set. Hence contrast and entropy of each of the kernel is calculated, and now we have a new feature set with 400 feature count. The features are matched using the Euclidean distance metric as in equation 4. The performance of CBIR is evaluated using the Normalised rank of the relevant images retrieved. Hence both, the number of relevant image retrieved and position of these relevant images is considered for evaluating the performance of CBIR system. If Q and DB are the feature set for query image and the database image respectively then the Euclidean Distance is calculated as in equation Howarth and Rüger (2005)

\[
Euclidean(Q, DB) = \sqrt{\sum_{i=1}^{n} ((Q_i - DB_i)^2)}
\]  

(4)

In case of Euclidean distance, distance value 0 indicates the perfect match and 1 indicates the complete mismatch, hence more the distance less is the similarity. In CBIR, for performance evaluation, the rank/position of the retrieved image is equally important as the number of relevant retrieved images. Hence, the authors have calculated the rank for each query image as proposed in Müller et al. (2001)

\[
\tilde{Rank} = \frac{1}{NN_R} \left( \sum_{i=1}^{N_R} R_i - \frac{N_R(N_R - 1)}{2} \right)
\]  

(5)

Here \(R_i\) - position of the relevant image among the retrieved.

\(N_R\) -No of relevant images in the database

N-Total no of images

The value of the rank is ranging from 0 to 1 where 0 indicates perfect match and 1 indicate the complete mismatch. The above expression calculates the deviation from the ideal condition when all the \(N_R\) relevant images are at the top positions.

In Table 2 each row indicate the result for each of the 21 classes of image in the data set. The values in the table indicate the average value of the rank of retrieval results for all the 100 images, taken as query image one at a time, in each class. Here the improvement in the rank value for each image category is clearly observed for the features with the contrast and entropy values as compared to that of pooled feature set. The rank value in
Deep learning feature map for remote sensing data

case of building is 0.4419 for pooled feature which is improved to 0.2394 and 0.2418 when calculated for the contrast and entropy features, respectively. In case of normalised rank higher value means higher deviation from the ideal condition and lower value means better performance. The decreasing value demonstrates the improving performance.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Retrieval results in terms of number of relevant image retrieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Category</td>
<td>Count_10</td>
</tr>
<tr>
<td>Agriculture</td>
<td>9</td>
</tr>
<tr>
<td>Airplane</td>
<td>10</td>
</tr>
<tr>
<td>Baseball diamond</td>
<td>2</td>
</tr>
<tr>
<td>Beach</td>
<td>8</td>
</tr>
<tr>
<td>Building</td>
<td>4</td>
</tr>
<tr>
<td>Chaparral</td>
<td>2</td>
</tr>
<tr>
<td>Dense residential</td>
<td>3</td>
</tr>
<tr>
<td>Forest</td>
<td>3</td>
</tr>
<tr>
<td>Freeway</td>
<td>2</td>
</tr>
<tr>
<td>Golfcourse</td>
<td>2</td>
</tr>
<tr>
<td>Harbour</td>
<td>3</td>
</tr>
<tr>
<td>Intersection</td>
<td>2</td>
</tr>
<tr>
<td>Medium residential</td>
<td>6</td>
</tr>
<tr>
<td>Mobile home park</td>
<td>2</td>
</tr>
<tr>
<td>Overpass</td>
<td>6</td>
</tr>
<tr>
<td>Parking lot</td>
<td>1</td>
</tr>
<tr>
<td>River</td>
<td>2</td>
</tr>
<tr>
<td>Runway</td>
<td>1</td>
</tr>
<tr>
<td>Sparse residential</td>
<td>2</td>
</tr>
<tr>
<td>Storage tanks</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Retrieval results for the dataset in terms of rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Category</td>
<td>Pooled Feature (Rank)</td>
</tr>
<tr>
<td>Agriculture</td>
<td>0.1691</td>
</tr>
<tr>
<td>Airplane</td>
<td>0.055</td>
</tr>
<tr>
<td>Baseball diamond</td>
<td>0.3239</td>
</tr>
<tr>
<td>Beach</td>
<td>0.1671</td>
</tr>
<tr>
<td>Building</td>
<td>0.4419</td>
</tr>
<tr>
<td>Chaparral</td>
<td>0.1994</td>
</tr>
<tr>
<td>Dense residential</td>
<td>0.4803</td>
</tr>
<tr>
<td>Forest</td>
<td>0.4138</td>
</tr>
<tr>
<td>Freeway</td>
<td>0.3458</td>
</tr>
<tr>
<td>Golfcourse</td>
<td>0.2618</td>
</tr>
<tr>
<td>Harbour</td>
<td>0.3414</td>
</tr>
<tr>
<td>Intersection</td>
<td>0.3819</td>
</tr>
<tr>
<td>Medium residential</td>
<td>0.1266</td>
</tr>
<tr>
<td>Mobile home park</td>
<td>0.4269</td>
</tr>
<tr>
<td>Overpass</td>
<td>0.4851</td>
</tr>
<tr>
<td>Parking lot</td>
<td>0.3479</td>
</tr>
<tr>
<td>River</td>
<td>0.4707</td>
</tr>
<tr>
<td>Runway</td>
<td>0.4456</td>
</tr>
<tr>
<td>Sparse residential</td>
<td>0.4737</td>
</tr>
<tr>
<td>Storage tanks</td>
<td>0.208</td>
</tr>
</tbody>
</table>

Similarly, Table 1 shows of the average value of the number relevant images retrieved for each query image out of top 10, 20, 30, 40 and 50 for each class of image. A good
improvement is observed for the pooled contrast and entropy features over the normal pooled features. The count value for the forest category is 3, 4, 5, 5, 6 out of top 10, 20, 30, 40 and 50 respectively for normal pooled features which is improved to 8, 15, 20, 25, 29 for contrast features and 7, 12, 16, 18, 20 for entropy features.

5 Conclusion

In the last few years, there is an emerging paradigm in CBIR systems which allows system to learn feature extraction kernel. This is made possible by the use of DL, CNNs and its architecture that embeds feature extraction with it. Here in the presented work, we trained CNN kernels for feature extraction, and obtained the feature map from the middle layers. The statistical parameters like contrast (standard deviation) and entropy of the feature map are calculated to use it as feature set for the purpose of image comparisons. Feature map as it is, and its statistical parameters, one at a time was used as features in conventional CBIR model. Retrieval results was compared in both the cases. It was observed that the results are better when statistical parameters of feature map obtained from CNN are used as features. This finding opens up a new alley in the field of image feature representations. More evocative parameters can be experimented to represent the high dimension feature map from the CNNs that improves the retrieval performance.

Acknowledgement

The authors are thankful to Institute of technology, Nirma University, for supporting this research work.

References

Deep learning feature map for remote sensing data


