Content-Based Image Retrieval System

Abhishek Kumar, Abhishek Kumar Sah, Arpit Das, Saumyak Raj

Abstract

Content Baes Image Retrieval system is the retrieval of images similar to a query image. The general method to achieve this is to extract useful image features using some deep learning neural networks and retrieve similar images from the dataset. An important aspect of a good result is the selection of a suitable similarity score.
In this project, we used a deep learning network to extract features and then pre-cluster the images based on these features which greatly improves the retrieval time without much compromising with the accuracy.

1 Introduction

Obtaining similar photos from a huge database in response to a query image is referred as Content-Based Image Retrieval (CBIR). The standard technique is to locate similar photos by analyzing some of the images' attributes. The features chosen have a considerable influence on the system's success. These properties should be used to describe the photo's content. As a result, high-level attributes are necessary, while low-level attributes like pixel values are ineffectual.
A distance metric is used to determine the similarity (or dissimilarity) between a query image (Q) provided by the user and a database image (I) stored in the system. A smaller calculated distance indicates a higher degree of similarity.
In 1995, IBM released the QBIC (Query By Image Content) system, which was the first commercial version of the CBIR technology. Users can search using user-created sketches, example photos, and drawings.
Lohite et al. employed the SVM (Support Vector Machine) classifier to optimize the outcome utilizing the frequently used color, texture, and edge properties of the images.
The performance of CBIR has improved since the introduction and evolution of Deep Learning Neural Networks, because deep models allow us to extract higher-level features alongside low-level attributes from images, overcoming the semantic gap.
After analyzing several aspects of images, such as geometry, color, and texture, Khokhar et al. described how Back-propagation Feedforward Neural Network (BFNN) may be used for classification in CBIR.
Finally, an Artificial Neural Network was employed to retrieve photos. Part-based weighting aggregation (PWA) was introduced by Xu et al. for CBIR. As part detectors, this PWA uses discriminative filters from deep convolutional layers.

2 Methods Used

Images are represented by a 3-dimensional array of numbers that represent the pixel values of the image encoded in RGB space. But we cannot use these pixel values directly to semantically compare images because these are low level features. So, we need to extract high-level features to reduce the “semantic gap”. For this, we used some pre-trained deep learning convolutional neural network models. For this purpose, we present a brief description of the neural networks.

2.1 Convolutional Neural Networks (CNN)

It is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. The layers of a CNN have neurons that are arranged in 3 dimensions: width, height, and depth. The neurons in a layer are connected to a small region

of the preceding CNN layer, unlike to all the neurons which is a norm in a fully-connected neural network. A simple CNN is a sequence of layers, and every layer of a CNN transforms one volume of activation to another by passing through certain differentiable functions. There are mainly three types of layers in CNN architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. The Fully-Connected Layer at the end returns the output of the CNN, which is the score of different classes. As we go deeper into the CNNs the model starts to learn high-level features from the low-level features, and the last-second layer contains the most useful features extracted from the image, which we can use for the purpose of comparison of images.

2.2 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

2.3 Clustering

Cluster analysis, or clustering, is an unsupervised machine learning task.

It involves automatically discovering natural grouping in data. Unlike supervised learning (like predictive modeling), clustering algorithms only interpret the input data and find natural groups or clusters in feature space. A cluster is often an area of density in the feature space where examples from the domain (observations or rows of data) are closer to the cluster than other clusters. Images in the same cluster will have similar features. This will imply that the images have similar content. This will slightly decrease the performance of the method but will provide a great boost to the speed of image retrieval which will be worth it in most use cases, especially for the larger database.

There are many good clustering algorithms but we use “K-Means Clustering”.

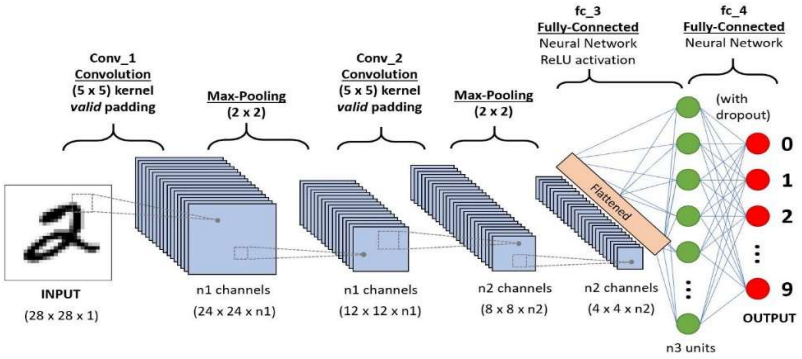


Figure 2: A Convolutional Neural Network

It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision tasks given the vast computing and time resources required to develop neural network models for these problems and the huge jumps in the skill that they provide on related problems.

Taking advantage of the method we used Convolutional Neural Networks trained on ImageNet Dataset, which consists of more than **14 million** images and **20,000** classes. The models have been trained with state-of-art techniques and a lot of resources.

We use models trained on ImageNet to extract high-level features of the query images as well as the images in the dataset because this allows us to take advantage of the highly refined models.

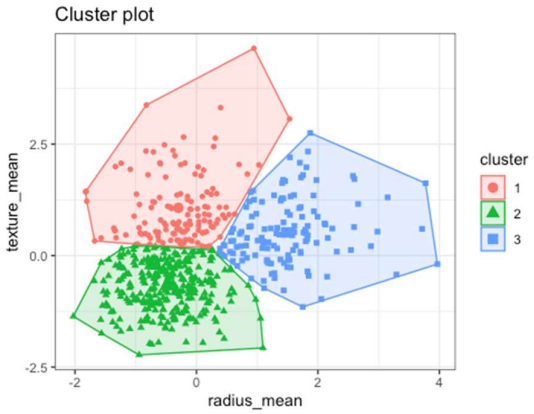


Figure 1: Clustering

2.4 K-Means Clustering

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm. The k-means clustering algorithm mainly performs two tasks: (i) Determines the best value for K center points or centroids by an iterative process. (ii) Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

2.5 Principal Component Analysis

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but dimensionality reduction is to trade a little accuracy for simplicity and most importantly time of execution and computation power. For example, the VG19 model gives a feature vector of size 4096 but using all these features will require a huge computation power and will be slow at execution which is bad for the end-user of CIBR.

2.6 Similarity Score

The feature vectors of images in the database and that of the query image needs to be compared. And for this purpose, a suitable similarity score needs to be used. We are using the "rbf_kernel" scores. RBF stands for the Radial Bias Function. RBF kernels are the most generalized form of kernelization and are one of the most widely used kernels due to their similarity to the Gaussian distribution. The RBF kernel function for two points X_1 and X_2 computes the similarity or how close they are to each other.

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

In the figure, K is the rbf_kernel, x, x' are two input vectors and γ is a hyperparameter.

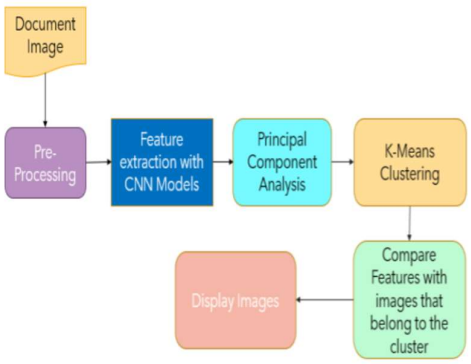
3 Database Used

Flilckr8K contains 8,000 images. The images were chosen from six different Flickr groups, and tend not to contain any well-known people or locations, but were manually selected to depict a variety of scenes and situations.

4 Workflow Of the Method

4.1 Training Phase

During the initial training phase, a model is first initialized with the ImageNet Dataset. The final layer (Fully Connected Layer) layer is removed from the model and the output of the last remaining layer is used as features of the images. The model is then used to extract the features of all the images in our Database after necessary preprocessing. These features have very high dimensions so PCA is applied to the features. Finally, K-Means is used to group all the features into groups of similar images.



4.2 Query

When a query image is received, the same flow is applied to it to extract features and the respective cluster information of the query image. Features of the images are only compared to the features of images in the

cluster of the query image. Finally, the top images with the highest similarity score are displayed.

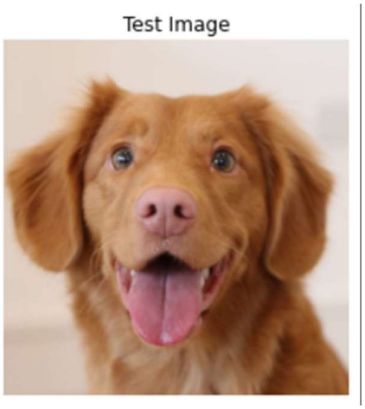
5 Result

We trained our dataset on many different models and compared their results. The different CNN models used are: DenseNet201, InceptionV3, InceptionResNetV2, MobileNetV2, VGG19, Xception. Following table shows the accuracy obtained from these different models:

Model	**Precision (%)**
VGG19	86%
InceptionResNetV2	84.2%
DenseNet201	82.5%
InceptionV3	82%
Xception	80.3%
MobileNetV2	79.4.%

We got the best results from model VGG19, which can be seen in the above table. We created a Django web-app which takes in the query image, and returns the list of all matching images found in the dataset.

Following is an example run with a test image and result images:



6 References:

- https://arxiv.org/pdf/2002.07877v1.pdf
- CS231n: Convolutional Neural Networks for Visual Recognition. URL: http://cs231n.stanford.edu/.
- K. T. Ahmed et al. "Convolution, Approximation and Spatial Information Based Object and Color Signatures for Content Based Image Retrieval". In: 2019 International Conference on Computer and Information Sciences (ICCIS). Apr. 2019, pp. 1–6. DOI: 10.1109/ICCISci.2019.8716437.
- Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. "An Analysis of Deep Neural Network Models for Practical Applications". In: arXiv e-prints, arXiv:1605.07678 (May 2016), arXiv:1605.07678. arXiv: 1605.07678

Work Division

Member	Contribution
Abhishek Kumar Sah	• Participated in proposing solution • Implemented the training and query notebook with a base technique which was further modified by other members according to their purposes. • Integrated the image search algorithm in the Django web-app. • Participated in report making (Methods used)
Saumyak Raj	• Participated in proposing solution • Tried and tested different CNN models to pick the best model. • Implemented Django codebase to make the web-app. This includes forms, views, etc. • Participated in report making (Methods Used)
Arpit Das	• Participated in proposing solution • Tried different clustering and dimensionality reduction algorithms • Implemented the HTML pages in the web-app • Participated in report making (Database used and result)
Abhishek Kumar	• Participated in proposing solution • Experiment with the program, with and without PCA and clustering. • Implemented the HTML pages in the web-app • Participated in report making (Abstract and Introduction)