Image based Search Engine

Anjali Sharma[1], Bhanu Parasher[1]

[1]*M.Tech., Computer Science and Engineering, Indraprastha Institute of Information Technology, Delhi, India*

---***---

Abstract - *We can see that different kinds of data are floating on the internet from the last couple of years. It includes audio, video, images, and text data. Processing of all these data has been a key interest point for Researchers. To effectively utilize these data, we want to explore more about it. As it is said, an image speaks more than a thousand words, so here in this article, we are working on images. Many researchers showed their interest in Content-based image retrieval (CBIR). CBIR doesn't work on the metadata, for example, tags, image description, etc. However, it works on the details of the images, or we can say the features of the images, for example, color, texture, edges, etc. The retrieval model should produce accurate and quick results. So here in this article, we are presenting our work on CBIR that will address these issues. Now, to make the CBIR model more accurate, the extraction of accurate features is needed. We extracted the image features by the Resnet-50 deep neural network and fed them to our neural network model. We trained the model on a broad set of images that are collected from four different sources. And for speed up, we performed clustering based on the hybrid approach. The proposed algorithm achieved an accuracy of 92.5.*

***Key Words*: Colour Coherent Vector (CCV), Convolutional Neural Network (CNN), Hue-Saturation-Value (HSV), Content Based Image Retrieval (CBIR), Deep Learning (DL).**

1. INTRODUCTION

The image retrieval system attracted great researchers towards it over the years. [3] (Jain and Dhar, 2017) used a pre-trained CNN model to train the shoe and kitchen appliances dataset. The CNN model used is the Inception-v3 model of Google Net deep learning architecture. The authors divided their work into three steps first is the background and unnecessary feature removal while storing all the required features. Then applying the pre-trained CNN model and finally using Euclidean distance to retrieve the images similar to the query. In [4] (Krizhevsky et al., 2012) authors used ImageNet dataset, this dataset consists of 1000 classes overall the dataset is of 1.2 million images. The authors trained the dataset using deep CNN, and this model consists of 8 layers. The first five layers are convolutional layers, and the other three are fully connected layers. Features are extracted from the 7th layer of the network to get the top one error rate of 37.5%. The limitation of this work is the processing power of the machine. The authors used GTX 580 3GB GPUs because of which there is a bound on the network size. The authors in [2] (Chen and Liu, 2015) used the Deep CNN model to resolve the problem of similar cloth retrieval and clothing-related problems. As the dataset is large, a fine-tuned, pre-trained model is used to lower the complexity between training and transfer learning. Domain transfer learning helps in fine-tuning the main idea behind is reusing the low-level and midlevel network across domains. In paper [5] (Venkata and Yadav, 2012), the author proposed a method of image classification based on two features. First is edge detection using the Sobel edge detection filter. The second feature is the colour of an image for which the author used CCV. These features help in finding the similarity between the query image and the dataset image to generate the output. Paper [1] (Brilakis et al., 2005) discusses the use of a Content-based Search engine for the construction image dataset. The author used colour, edge, and texture features to identify the construction image. The Blind Relevance feedback is used for the classification and relevance of the output.

2. Dataset

Our data set consists of 68 classes. Overall, we have 16,000 images of different dimensions. We collected a dataset from 4 different sources: INRIA holidays dataset, GHIM-10k, Caltech 256 and WANG database. Based on the collected dataset, we have defined different classes. The data is splitted into train and test in 80:20 ratio.

Fig -1: Dataset Images

3. Baseline

We implemented an algorithm from literature work that computes feature of the image based on edge filter and colour filter. The 3D colour histogram in HSV colour space is used for colour filter and Canny filter for edge filtering. Colour features are calculated by dividing the image into 6 segments, then for each segment features are calculated and are appended with each other to make the final feature matrix of the image. Now applying an edge filter on the image and appending the matrix formed with the colour filter matrix of the image. Similarly, finding it for testing data and applying the cosine similarity for the output generation and getting overall accuracy of 75.0

4. Methodology

The methodology subsumes two modules that are training and testing.

Training consists of following steps:

4.1 Pre-processing the image

All the dataset images are resized to 244*244 so that it can be used for Resnet-50.

4.2 Feature Extraction Using Resnet-50

The features for each image are extracted using the Resnet-50 pre-trained deep learning network. ResNet-50 is a pre-trained deep learning model that consists of 50 layers. ResNet is a short form of Residual Network. This model comes under CNN, which helps in image classification. Millions of images of 1000 categories are used for training the ResNet-50 model. In place of learning the features of the image, the model learns the residual of the features. Residual is feature subtraction from the input of that layer. It is preferred over other models of DL as they become complicated to train, and the accuracy also starts degrading after a saturation period. However, ResNet overcomes these problems by connecting the nth layer's input to the (n+x) th layer. Figure 2 showing the section 4.2 and 4.3 how features using resnet-50 are feed to neural network.

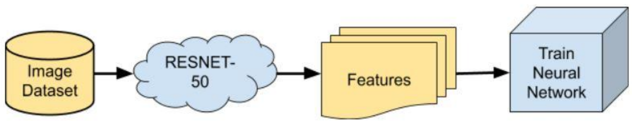

Fig -2: Extracting Resnet Features and Training Neural Network

4.3 Training the Neural Network

The Neural Network used is developed on the Keras library, which runs on top of the TensorFlow platform. Adding three RELU layers with different neurons and, finally, adding the SoftMax activation layer for output generation, inputting the Resnet-50 features and labels for each class to this model. The model is trained for 10 epochs on the training data.

4.4 Clustering the Data and defining leaders using Histogram features

Clustering is used within the classes to speed up the process of image retrieval. The mean is computed all over the class and using it some images are picked up as leaders, and then from each image in that class, the cosine similarity score is computed, and finally, assigning the images to the nearest leader. For computing the cosine similarity, we used histogram features. The histogram features are computed by converting the input image from RGB to HSV format and then dividing the image into 6 segments. Finally, computing the histogram for each segment and appending the together. Figure 3 showing the clustering basis on the histogram features and saving the clusters with features.

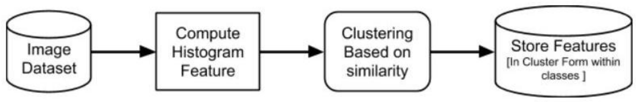

Fig -3: Clustering using the Histogram Features

Testing consists of following steps:

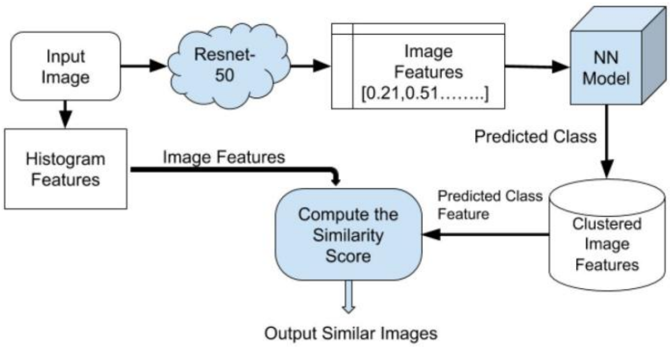

Fig -4: Testing Model

4.5 Class prediction using Trained Model

Firstly, for the input image the features are computed using the Resnet-50 model, and then the class is predicted using the trained neural network. Figure 4 depicting the final purposed running model.

4.6 Cosine similarity computation with Clusters

Now we got the class to which the input image belongs. The second task is to search for similar images in that predicted class. For doing so, the histogram features are computed for the input image, then finding the cosine similarity with the leaders of the predicted class and storing the closest leader.

4.7 Final result selection

In the end, cosine similarity between the input image and all the images in that leader's cluster is computed. Based on that produced as output and image are ranked according to the similarity scores.

5. Results and Conclusion

In Figure 5 we are showing the results for 5 input Query.

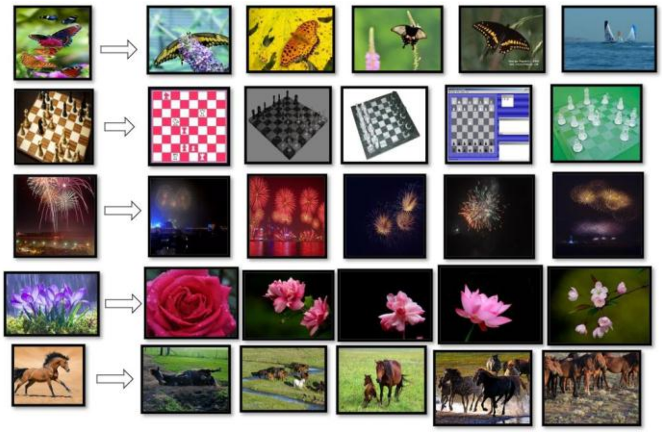

Fig -5: Top 5 Results on 5 Input Images

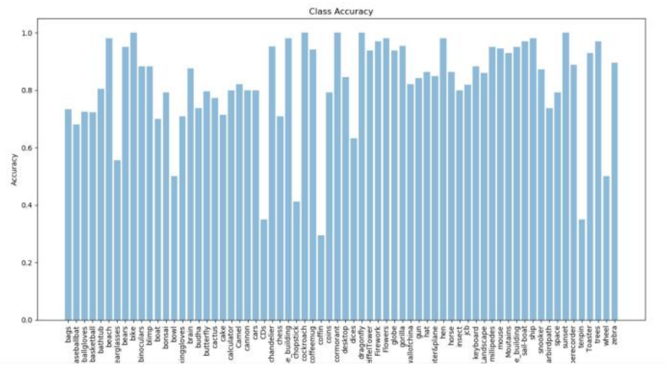

Fig -6: Class wise accuracy plotted using bar chart

Figure 6 showing the accuracy of each class by testing over 5000 input images. On Maximum classes we are achieving accuracy of above 0.87 and the overall accuracy achieved by proposed algorithm is 0.925. The trained neural network predicts the input image class accurately. Also, clustering within a class allows us to not compare the input image with each image of the predicted class. Hence the time complexity of image retrieval decreases.

The Proposed algorithm is simple to understand and can be utilized where a large number of unlabelled images are present, and we want to find similar images out of it. The algorithm is tested and trained on the Nvidia GTX 1050 GPU. In Future work, the size of the dataset can be increased by adding more number of classes. Better training model can be developed by experimenting with the number of layers, and by changing the hyperparameters.

REFERENCES

[1] Ioannis Brilakis, Lucio Soibelman, and Yoshihisa Shinagawa. 2005. Content-based construction site image retrieval. Journal of Computing in Civil Engineering.

[2] J.-C. Chen and C.-F. Liu. 2015. Visual-based deep learning for clothing from large database. ASE Big-Data Social Informatics. ACM, p. 42.

[3] Surbhi Jain and Joydip Dhar. 2017. Image based search engine using deep learning. International Conference on Contemporary Computing.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems.

[5] Divya Ragatha Venkata and Divakar Yadav. 2012. Image query based search engine using image content retrieval. International Conference on Modeling and Simulation (UKSIM-2012).