CONTENT-BASED IMAGE RETRIVAL USING VGG16

Project report in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology
In
COMPUTER SCIENCE ENGINEERING

Submitted By

Shauvik Roy	Enrollment No. 12020009028041
Sayan Mukherjee	Enrollment No. 12020009001317
Soumi Roy	Enrollment No. 12020009028010
Monodeep Saha	Enrollment No. 12020009028074
Debjyoti Dutta	Enrollment No. 12020009028004
Shreya Ghosh	Enrollment No. 12020009028068
Daipayn Saha	Enrollment No. 12020009028003
Subham Das	Enrollment No. 12020009028001
Shivam kumar	Enrollment No. 12020009028039
Amit Ghosh	Enrollment No. 12020009028015

Under the guidance of

Prof. (Dr.) Rajendrani Mukherjee

&

Prof. Bijoya Mukherjee

Department of Computer Science Engineering

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

CERTIFICATE

This is to certify that the project titled **CONTENT-BASED IMAGE RETRIVAL USING VGG16** submitted by **Shauvik Roy** (University Roll No. 12020009028041**), Sayan Mukherjee** (University Roll No. 12020009001317**)**, **Soumi Roy** (University Roll No. 12020009028010**), Monodeep Saha** (University Roll No. 12020009028074), **Debjyoti Dutta** (University Roll No. 12020009028004), **Shreya Ghosh** (University Roll No. 12020009028068), **Daipayn Saha** (12020009028003), **Subham Das** (University Roll No. 12020009028001), **Shivam kumar** (University Roll No. 12020009028039) and **Amit Ghosh** (University Roll No. 12020009028015) students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of requirement for the degree of Bachelor of Computer Science, is a bonafide work carried out by them under the supervision and guidance of Prof. (Dr.) Rajendrani Mukherjee & Prof. Bijoya Mukherjee during 5th Semester of academic session of 2021 - 2022. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original, and its performance is found to be quite satisfactory.

Prof. (Dr.) Rajendrani Mukherjee
Project Guide
Department of Computer Science and Engineering
UEM, Kolkata

Prof. Bijoya Mukherjee
Project Guide
Department of Computer Science and Engineering
UEM, Kolkata

Prof. (Dr.) Sukalyan Goswami
Head Of the Department of Computer Science and Engineering
UEM, Kolkata

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof. (Dr.) Rajendrani Mukherjee and Prof. Bijoya Mukherjee of the Department of Computer Science, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Dr. Sukalyan Goswami, HOD, Computer Science, UEM, Kolkata and all other departmental faculties for their ever-present assistant and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Shauvik Roy
Sayan Mukherjee
Soumi Roy
Monodeep Saha
Debjyoti Dutta
Shreya Ghosh
Daipayn Saha
Subham Das
Shivam Kumar
Amit Ghosh

TABLE OF CONTENTS

CONTENT-BASED IMAGE RETRIVAL USING VGG16
(IMAGE SEARCH ENGINE)

ABSTRACT

Multimedia content analysis is applied in different real-world computer vision applications, and digital images constitute a major part of multimedia data. In last few years, the complexity of multimedia contents, especially the images, has grown exponentially, and on daily basis, more than millions terabyte of image data is uploaded at different servers such as Google, Yahoo, Twitter, Facebook, and Instagram etc. To search for a relevant image from an archive is a challenging research problem for computer vision research community. Most of the search engines retrieve images on the basis of traditional text-based approaches that rely on captions and metadata. In the last two decades, extensive research is reported for content-based image retrieval (CBIR), image classification and image clustering. In CBIR and image classification-based models, high-level image visuals are represented in the form of feature vectors that consists similarities. In this project, we have tried to implement a simple search with the help of pre-trained deep neural network named VGG16 which act as a feature extractor of the image dataset and a matrix normalization algorithm to compare the similarity of feature extracted from query image with the other features of other images stored in the local database.

1. INTRODUCTION

Due to recent development in technology, there is an increase in the usage of digital cameras, smartphones, and internet. The shared and stored multimedia data are growing, and to search or to retrieve a relevant image from an archive is a challenging research problem. Most of the search engines on the Internet retrieve the images on the basis of text-based approaches that require captions as input. It is near to impossible to apply the concept of manual labeling to existing large size image archives that contain millions of images. The second approach for image retrieval and analysis is to apply an automatic image annotation system that can label image on the basis of image contents.

There is a substantial amount of work that has been done on this topic as detailed in the previously done section improvements still can be achieved as the benchmark has not yet reached human-level accuracy or surpassed that. Using recent advancements both in the usage of convolution kernels in Deep Learning, model training procedures (like hyper-parameter tuning, data augmentation, etc.) and superior feature extraction techniques using pre-trained DCNNs has increase the accuracy of Image Searching Models.

Content-based image retrieval (CBIR) is a framework that can overcome the above-mentioned problems as it is based on the visual analysis of contents that are part of the query image. To provide a query image as an input is the main requirement of CBIR and it matches the visual contents of query image with the images that are placed in the archive, and closeness in the visual similarity in terms of image feature vector provides a base to find images with similar contents. In CBIR, low-level visual features (e.g., color, shape, texture, and spatial layout) are computed from the query and matching of these features is performed to sort the output.

With the advent of deep learning, it was possible to partly or fully eliminate the need for traditional feature extractor like manual pre-processing of the images. Recent methods used Convolution Neural Networks (CNN) [1] and improved the performance for Bangla character and digit recognition on a relatively large-scale dataset. The VGG16 CNN Model [2] proposed consists of combination of 16 layer Deep Convolutional Neural Network with input size of image as 224 x 224 pixel. Our purpose in this project is to present a simple image searching method based on statistical normalization technique and with a powerful pre-trained DCNN for feature extraction. The pre-trained neural network we have used here is VGG16 developed by Simonyan and Zisserman [2] contain several differences from the ones in high-performing entries from the ILSVRC-2012 and ILSVRC-2013 (Zeiler & Fergus) [3].

In the last part of our project, we have tried to store all the features extracted from the image dataset and stored it in a (.npy) database. This feature stored can be further utilize for analyzing the efficiency of our technique used image searching project. The extracted features are all numeric data which were given VGG16 pre-trained network [1] as an output. And at the end we have tried to show our output in a Graphical User Interface (GUI) for user convenience.

2. LITERATURE SURVEY

Krizhevsky et al. [3] trained a deep CNN to classify ImageNet dataset consisting of one 2 million pictures into a thousand completely different categories. The authors worked on a network containing eight layers, wherever initial 5 were convolutional layers, and last 3 were totally connected layers. The authors used the options extracted from the seventh layer to fetch similar photos and achieved the top-1 error rate of 37% and top-5 error rates of 17%. The ConvNet architecture of Simonyan and Zisserman [2] contain several differences from the ones in high-performing entries from the ILSVRC-2012 and ILSVRC-2013 (Zeiler & Fergus; Sermanet et al.) [3] competitions. For comparison with our proposed model, we show the Simonyan and Zisserman model [2] in Figure 3.

Babenko et al. [4] suggested compressing the features using the dimensionality reduction method and attained a good performance. However, because of the high dimensionality of CNN features and inefficiency of similarity computation between two 4096-dimensional vectors. Deep models are used for hash learning. Xia et al. [5] proposed a supervised hashing methodology to check binary hash codes to retrieve pictures exploitation deep learning and disclosed the revolutionary performance of retrieval on datasets that are publicly offered.

3. PROBLEM STATEMENT

We use search engine regularly. When we have queries, we can use the search engine like Google to retrieve the most relevant answer. Most of the query's format is text-based. But not most of the time, the text is quite useful to find relevant answers. For example, you want to search for a product on the internet, in this case, a t-shirt, but you don't know the name of it. How could you find them? Well, you can write the description of that shirt.

The problem for using descriptions is that you will get wide varieties of products. And what makes it worse, they will be not similar with the product that you want to search, so you need a better way to retrieve them. To solve it, we can use the image of the product, extracts its features, and CIBR to retrieve similar products. Content-based image retrieval (CBIR) is a system for retrieving relevant images based on a given image. The system consists of an image query and an image database.

4. EXPERIMENTAL SETUP

In this project we tackle the objective of Image Search Engine. It is a method to search for information, products or other things through images, without having to give detail about anything in the query search. It mainly contains three steps: 1) to collect and pre-process the images to form a proper dataset; 2) the mining process of extracting silent features from the images and store them in database for further use of searching or normalization; and 3) in the third part, the method will ask for a query image and tries a couple of matrix normalization method and provide the desired output a probability score.

4.1 Dataset

The Dataset used in this project is a hybrid dataset. Initially, it was a CBIR Dataset (Content Based Image Retrieval Dataset) from Kaggle [6]. But later, to increase the size of the dataset and to diversify the dataset, we have used some multiple object datasets and merged all of them to build a huge workable dataset. After all the pre-processing, our hybrid image dataset is best fit for our project evaluation because of its large sample size, inferior data quality (when compared to other available datasets which suit our purpose since it helps us better generalize) and large variance; and also, because the output classes are balanced.

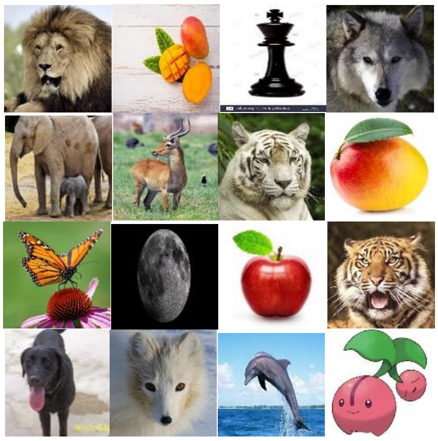


Fig 1: Sample Images from the Dataset

After discarding mistakes and scribbles, 4,141 images of different categories were included in the final dataset. The dataset consisted of 25+ categories of dataset, which consisted of different animals, fruits, cartoons and different regularly seen daily life objects were collected, digitized and pre-processed before running the feature extraction code. Here we have presented on 16 out of many categories of images present in our training dataset. So that all the images are successfully pre-processed for smooth running of our particular algorithm. These images were pre-processed by applying colour-inversion, noise removal and edge-thickening filters. All the images of dataset were resized into 224 x 224px to fit them VGG16 pre-trained model. The above give Figure 1 is small example of our hybrid dataset.

4.2 Deep Convolution Neural Network (DCNN)

Deep Neural Networks (DNNs), more commonly referred to as Deep Learning, employ deep NN architectures to automatically learn hierarchy of features from raw input data without the need for feature engineering [7]. Loosely inspired by how the mammalian brain uses different areas of the cortex to abstract different levels of features when given an input percept, deep learning methods are characterized by deep architectures with several hidden layers that allow them to learn many levels of abstraction, as opposed to shallow architectures with 1 or 2 hidden layers.

Recently, it has been proved that Deep Convolutional Neural Network (DCNN)is very effective for large-scale object recognition. [10] However, it needs a lot of training images. DCCNs have shown to be highly effective in processing visual data, such as images and videos. DCNNs take raw input data at the lowest level and transforms them by processing them through a sequence of basic computational units to obtain representations that have intrinsic values for classification in the higher layers [8]. A DCNN typically consists of three-layer types (Figure 2): convolution layers, pooling layers, and fully connected layers. A convolutional layer is parametrized by the number of channels, kernel size, stride factor, border mode, and the connection table. The convolution layer takes the input image and applies convolution filter on it to produce the output image. Multiple convolutional layers are used to take into consideration the spatial dependencies among image pixels. The max-pooling layer is used to make the neural network more invariant and robust which lead to lead to faster convergence and better generalization. It is common to use multiple fully connected layers after several rounds of convolution and the resulting structure of the last convolutional layer is flattened before connecting to the following fully connected layer.

DCNNs typically require large, annotated image datasets to achieve high predictive accuracy. However, in many domains, acquisition of such data is difficult and labelling them is costly. In light of these challenges, the use of well-established pre-trained DCNNs such as VGG-16, AlexNet, and GoogLe-Net has shown to be very useful for solving
cross domain image classification problems through the concept of transfer learning and fine-tuning [9]. The idea behind transfer learning is that it is cheaper and efficient to use deep learning models trained on ‘‘big data” image datasets (like ImageNet) and ‘‘transfer” their learning ability to new classification scenario rather than train a DCNN classifier from scratch [8]. With adequate fine-tuning, pretrained DCNN has been shown to outperform even DCNN trained from scratch model [8].

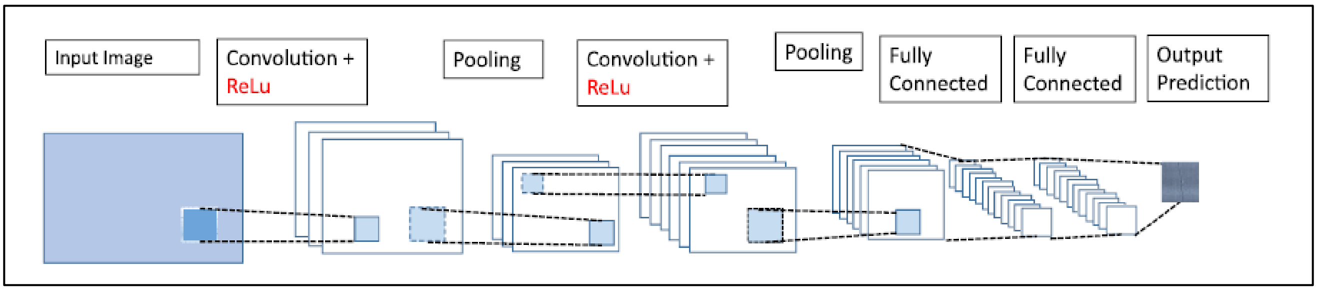


Fig. 2: Schematic of Convolution neural network architecture.

Image Reference: K. Gopalakrishnan, Siddhartha K. Khaitan, Alok Choudhury, Ankit Agrawal, Deep Convolution Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection, p 324. [12]

4.3 VGG16 Model Architecture

VGG16 model [2] is a pre-trained Convolution Neural Network architecture which was used to win ILSVR (Imagenet) competition in 2014. It is considered to be one of the excellent vision model architectures. Most unique thing about VGG16 is that instead of having a large number of hyper-parameters they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout thewhole architecture. In the end it has 2 fully connected layers followed by a softmax for output. The 16 in VGG16 [2] refers to it has 16 layers that have weights. This network is a pretty large network and it has about approximately 138 million parameters.

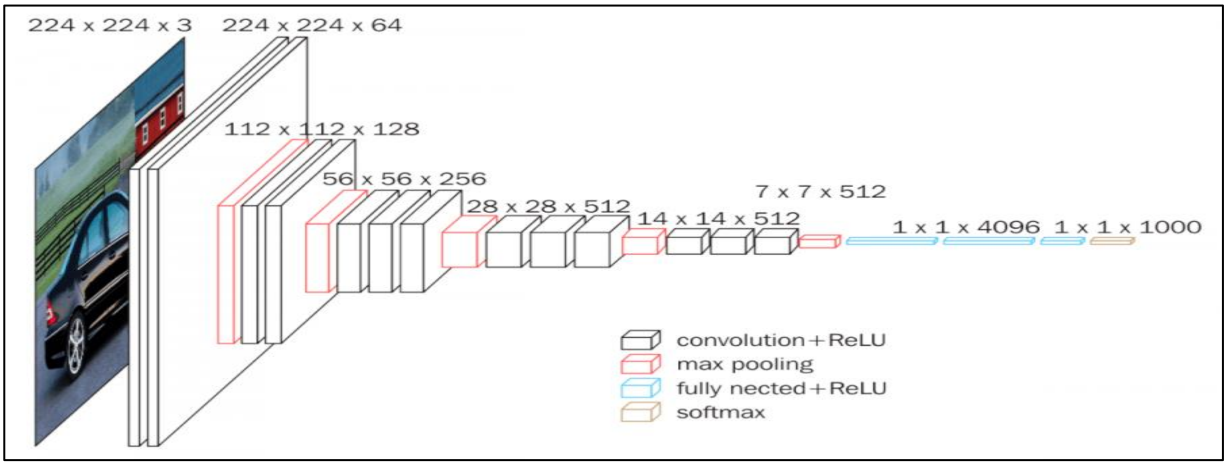
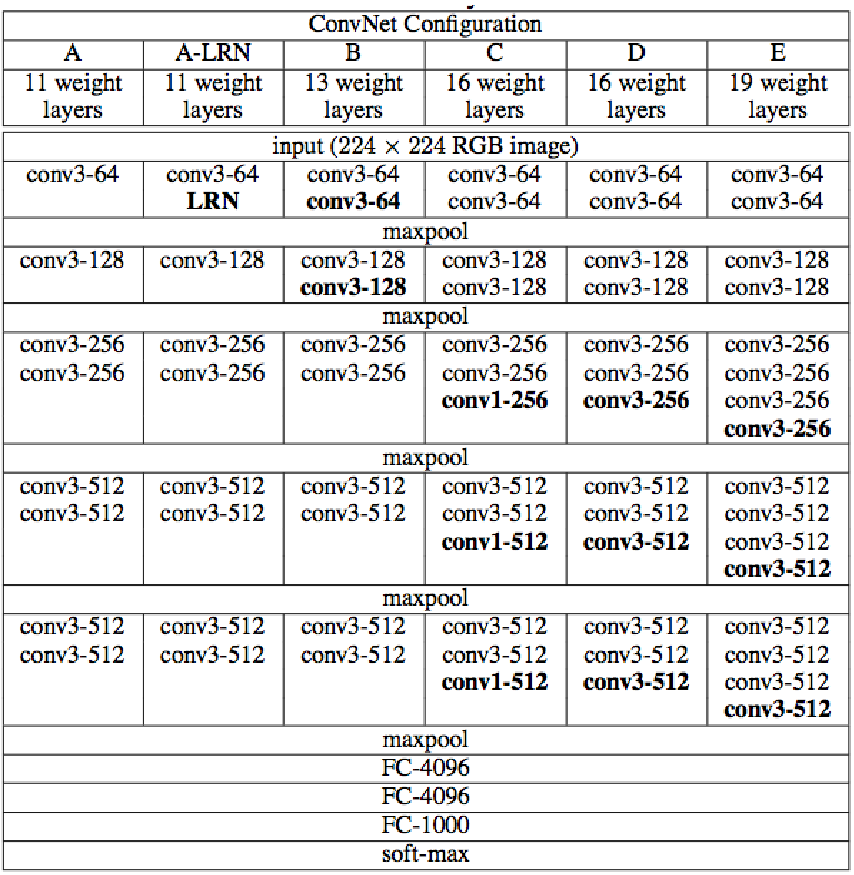


Fig 3: Architecture of VGG16

The table below listed different VGG architectures. VGG16 was identified to be the best performing model on the ImageNet dataset. The input to any of the network configurations is considered to be a fixed size 224 x 224 image with the RGB channels. The only pre-processing done is normalizing the RGB values for every pixel and resizing. This is achieved by subtracting the mean value from every pixel. We can see that there are 2 versions of VGG- 16 (C and D). There is not much difference between them except for one that except for some convolution layers, (3, 3) filter size convolution is usedinstead of (1, 1). These two contain 134 million and 138 million parameters respectively. size of the output activation map is the same as the input im- age dimensions. The activation maps are then passed through spatial max pooling over a 2 x 2-pixel window, with a stride of 2 pixels. This halves the size of the activations. Thus, the size of the activations at the end of the firststack is 112 x 112 x 64.

VGG16 network and further improves on following aspects: (1) small model size, (2) faster speed, (3) uses residual learning for faster convergence, better generalization, and solves the issue of degradation, (4) matches the recognition accuracy of the non-compressed model on the very large-scalegrand challenge MIT Places 365-Standard scene dataset. In comparison toVGG16 the proposed model is 88.4 percent smaller in size and 23.86 per-cent faster in the training time. This supports our claim that the proposed model inherits the best aspects of VGG16 and further improves upon it.In comparison to SqueezeNet our proposed framework can be more easily adapted and fully integrated with the residual learning for compressing various other contemporary deep learning convolutional neural network models Broader impact of our work could improve the performance in specializedtasks such as video-based surveillance, self-driving cars, and mobile GPU applications [11].



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig 4: Different VGG Architecture

4.4 Matrix Normalization

In statistics, normalization can have a range of meanings. In the simplest cases, normalization of ratings means adjusting values measured on different scales to a notionally common scale, often prior to averaging. In more complicated cases, normalization may refer to more sophisticated adjustments where the intention is to bring the entire probability distributions of adjusted values into alignment. In the case of normalization of scores in educational assessment, there may be an intention to align distributions to a normal distribution. A different approach to normalization of probability distributions is quantile normalization, where the quantiles of the different measures are brought into alignment. In our project, we will mostly dependent Frobenius Normalization based 2-norms. We have particularly used "scipy.linlag.norm()" function in Python for the calculation of Matrix Norm.

The Frobenius norm, sometimes also called the Euclidean norm (a term also used for the vector L^2- norm), is matrix norm of an matrix defined as the square root of the sum of the absolute squares of its elements. The Frobenius norm is defined by

$$||A||_F \equiv \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{i\,j}|^2}$$

Notice that one can think of the Frobenius norm as taking the columns of the matrix, stacking them on top of each other to create a vector of size m × n, and then taking the vector 2-norm of the result.

The Frobenius norm can also be considered as a vector norm. It is also equal to the square root of the matrix trace of AA^H, where A^H is the conjugate transpose. This is further improving the Gradient based learning on visual images or documents [13].

5. RESULT ANALYSIS

The output returned by our particular method gives an accurate result with a probabilistic score every given output image listed above that image. This score was given after the Frobenius matrix normalization. After extracting the features, we have tried to calculate the distance between the query and all images. In a alternative method, we can also use the Euclidean distance or l2 norm to measure it. If the number is getting smaller, the pair of images is similar to each other. In the below images (Figure 5 & 6) the algorithm returns the expected output of similar images of the query image which 'elephants. Here the output is a combination of 30 images similar to the query image with their relative probabilistic score mentioned above every image.

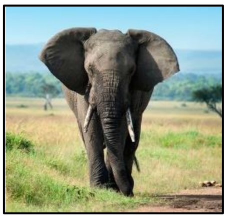


Fig 5: Image given as Query

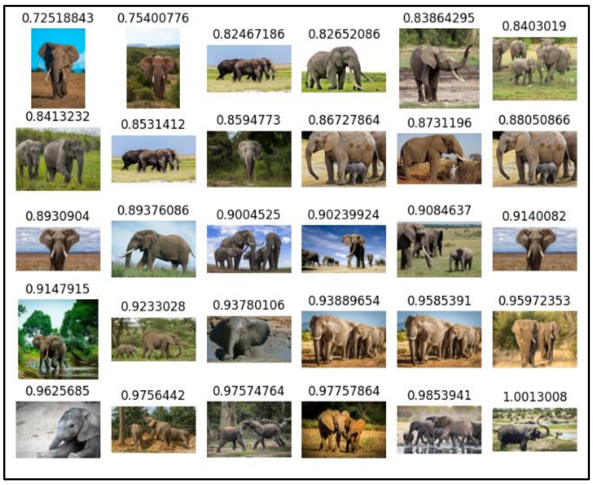


Fig 6: 30 Similar Images as Output

In the below images (Figure 7 & 8) the algorithm doesn't returns the expected output. The output of 30 images is similar to the query image. But here the query image was a 'Giraffe', as that particular category of images were not feed into the database so the algorithm returns nearly similar images here it is Tigers and Zebra. Here lies a disadvantage of our algorithm which cannot identify the images which are not present in the database.

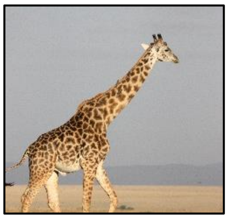


Fig 7: Image given as Query

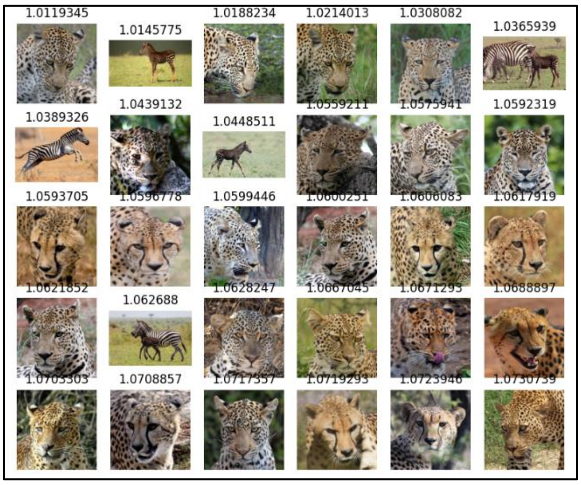


Fig 8: 30 Similar Images as Output

6. CONCLUSION AND FUTURE SCOPE

We explained Content-Based Image Retrieval (CBIR) and its difference from Text-Based Image Retrieval. Content-based image retrieval (CBIR) is a system for retrieving relevant data from images based on a given image [12]. The system consists of an image query and an image database. The process of the system will begin by extracting features on all images, whether it's the query or the image database by using a feature extraction algorithm. Then, the system will calculate similarities between the query with all images on the database. At the end, the system will retrieve all the images that have a great similarity with the query.

Modern CBIR systems use deep convolutional networks to extract features from a query image and compare them to those of the database images. The most similar images constitute the query's result. For extracting features, there are lots of options to choose. In this case, we can use a convolutional neural network (CNN) based pre-trained model named VGG16 for extracting those features. This model has a great capability to capture patterns than any other algorithms thanks to its convolutional layer that captures the neighbor for each instance of the data. Convolutional Neural Network consists of layers, such as convolutional layer for feature extraction, pooling layer for sampling the features, and fully-connected layer for doing prediction.

After we retrieve all of the images, now we can extract the features from all images using CNN and save those features on (.npy) format for later use. To specify the architecture, we will use VGG-16 architecture and pretrained weight from the ImageNet. After we extract features, we calculate the distance between the query and all images. And the algorithm is successful to return the desired output with respect to the input query image. The project was inspired from Irfan Al. Khalid blog [14].

REFERENCES

[1] LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), p.1995.

[2] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[3] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), pp.84-90.

[4] Babenko, A., Slesarev, A., Chigorin, A. and Lempitsky, V., 2014, September. Neural codes for image retrieval. In *European conference on computer vision* (pp. 584-599). Springer, Cham.

[5] Xia, R., Pan, Y., Lai, H., Liu, C. and Yan, S., 2014, June. Supervised hashing for image retrieval via image representation learning. In *Twenty-eighth AAAI conference on artificial intelligence*.

[6] https://www.kaggle.com/datasets/theaayushbajaj/cbir-dataset

[7] Rahman, M.M., Akhand, M.A.H., Islam, S., Chandra Shill, P., Hafizur Rahman, M.M.: Bangla handwritten character recognition using convolutional neural network. Int. J. Image Graph. Signal Process. 7(8), 42–49 (2015).

[8] S. Bai, Growing random forest on deep convolutional neural networks for scene categorization, Expert Syst. Appl. 71 (2017) 279–287, https://doi.org/10.1016/j.eswa.2016.10.038.

[9] H.-C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imaging 35 (2016) 1285–1298.

[10] Wataru Shimoda and Keiji Yanai. An analysis on visual recognizabilityof onomatopoeia using web images and dcnn features.

[11] Hussam Qassim, Abhishek Verma, and David Feinzimer. Compressed residual-vgg16 cnn model for big data places image recognition. In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pages 169–175. IEEE, 2018.

[12] Alkhawlani M, Elmogy M, El Bakry H. 2015. Text-based, Content-based, and Semantic-based Image Retrievals: A Survey. *International Journal of Computer and Information Technology*.

[13] LeCun Y, Bottou L, Bengio Y, Haffner P. 1998. Gradient Based Learning Applied to Document Recognition. *Proc. Of The IEEE.*

[14] https://towardsdatascience.com/build-an-image-search-engine-using-python-ad181e76441b