JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Efficient Content Based Image Retrieval System Using Decision Tree Classifier and Particle Swarm Optimization

[1]Syed Qamrul Kazmi, [2]Munindra Kumar Singh, [3]Saurabh Pal
[1]Research Scholar, [2]Assistant Professor, [3]Professor
[1]Department of Computer Applications,
[1]V.B.S Purvanchal University, Jaunpur, India

Abstract: Content-Based Image Retrieval (CBIR) is a technology and methodology used in computer vision and information retrieval to retrieve images from databases based on their visual content rather than metadata or text-based descriptions. Images are indexed and retrieved in CBIR systems utilizing their intrinsic visual qualities such as color, texture, shape, and other visual characteristics. Art galleries, the fashion business, medical imaging, satellite image analysis, and other industries use CBIR. Designing effective and efficient feature extraction approaches that capture the most significant parts of the visual content while minimizing computing cost is one of the problems in CBIR. Furthermore, finding the best similarity measuring metric is critical for accurate retrieval results. In this paper we have used Decision Tree Classifier and particle swarm optimization for creating a decision tree model to classify images based on their visual qualities which is the first step in using a decision tree for content-based image retrieval (CBIR). Particle Swarm Optimization is inspired by the social behavior of birds and fish. It involves a population of particles that move through a search space to find optimal solutions. Each particle adjusts its position based on its own experience and the experiences of other particles.

***IndexTerms* - Decision Tree Classifier, Content based image retrieval, Particle Swarm Optimization, Data mining**

I. INTRODUCTION

Using a Decision Tree for Content-Based Image Retrieval (CBIR) involves creating a decision tree model to classify images based on their visual features. Content-Based Image Retrieval focuses on finding images similar to a query image based on their content attributes, such as color, texture, and shape. The use of decision tree for CBIR consists of feature extraction, feature representation, data preparation and data splitting. Extraction of Relevant Features from photos involves extracting relevant features from photos that will be utilized as input to the decision tree. Color histograms, texture features (e.g., Haralick features), form descriptors (e.g., moments), and any other relevant visual data may be included. We have feature representation where each image should be represented as a vector of feature values. In the feature space, each feature corresponds to a dimension [1]. The next step is preparation of data which involves making a dataset with each data point consisting of an image vector and a class label or target. The class label shows the image's category or similarity degree. Data splitting is the process of dividing a dataset into training and testing sections. The decision tree will be built using the training subset, and its performance will be evaluated using the testing subset. Furthermore there is decision tree construction, model evaluation, query processing, retrieval and ranking and last but not the least we have post processing and refinement. Decision tree is created using the training data. Based on the extracted visual cues, a decision tree will be constructed to predict class labels. Tuning the decision tree algorithm's hyperparameters [2], such as the maximum depth of the tree, the minimum samples required to split a node, and other relevant factors. This is an important step in avoiding overfitting and guaranteeing proper generalization. Model Evaluation uses the testing data, assess the performance of the decision tree model. Precision, recall, F1-score, and accuracy are common CBIR evaluation criteria [17]. To find similar photos for a given query image, extract the same features from the query image and classify it using the trained decision tree. The search will be guided by the decision tree, which will navigate through its branches based on the feature values [3]. Retrieval and Ranking consists of navigating the decision tree to locate images with feature values that are similar to the query image. Sort the photos according to their similarity scores or categorization probability. Images that have a higher similarity score are thought to be more relevant to the query. At last refinement and post-processing is done which as depending on the returned findings, you can use post-processing techniques to further refine the ranking and improve the relevancy of recovered photos.

Particle Swarm Optimization (PSO) is a computational optimization technique inspired by birds' and fish's social behavior, notably how they move in groups to find optimal trajectories. It's a common algorithm for optimization and search issues, and it's frequently used in artificial intelligence and machine learning. The technique employs a population of "particles," each of which represents a possible solution to the optimization issue. These particles wander around in a search space, affected by their own best-known position (personal best) as well as the best-known position [14] of every particle in the population (global best). This collective motion is regulated by mathematical formulae that replicate the interactions of the particles.

The basic principle behind PSO is to iteratively update the velocity and position of each particle based on these influences, with the goal of finding the best solution by modifying their positions in the search space. The particles converge towards the best or near-optimal solution over time [4]. PSO has been widely employed in several disciplines, including engineering, economics, and machine learning, to tackle optimization problems [15]. It's relatively easy to build, and its exploration-exploitation balance frequently helps it identify solid answers rapidly. However, its performance, like that of any optimization algorithm, can vary depending on the problem at hand and the parameter settings.

II. RELATED WORK

Rastogi et al [4] discussed that the query image's color, shape, edge, and texture features are retrieved using different techniques, and database images are extracted in a similar fashion. Following that, similar images are obtained using a combination of the aforementioned attributes. Finally, the Model Approach [1] is used, which improves system efficiency. Thus, the required relevant photos are recovered from a vast database using the Effective Content Based Image Retrieval (CBIR) based on Model Approach. The suggested CBIR system is assessed by querying different photos, and its effectiveness is evaluated by calculating several parameters to test the efficiency of different methodologies and their combination to improve the performance of the obtained data.

Anita et al [5] discussed in their paper an overview of the relevance feedback (RF), interactive genetic algorithm, and neural network. Relevance feedback improves CBIR capability by closing the semantic gap between low-level features and high-level features. The interactive genetic algorithm is a subset of evolutionary computation that makes the retrieval process more participatory, allowing the user to acquire refined results from database matching to Query Image with his evaluation. When compared to standard implicit feedback, neuro-fuzzy logic-based implicit feedback produces better results. The study discusses recent advances in relevance feedback, interactive genetic algorithms, neural networks in CBIR, various relevance feedback techniques, and CBIR applications.

Arshiya et al [6] commented that CBIR is largely dependent on the calculation of feature representation as well as similarity. They offer a simple yet powerful deep learning system based on Convolutional Neural Networks (CNN) and comprised of feature extraction and classification for quick image retrieval. Many extensive observational studies for a variety of CBIR tasks employing picture databases yield some promising results, revealing some key lessons for enhancing CBIR efficiency. CBIR systems enable another image dataset to find images that are related to a query image. The most prevalent CBIR approach has to be Google's search per image tool.

Arun et al [7] proposed that the image retrieval is preceded by color histogram information. The multiple color axes are grouped into a number of bins for computing an image color histogram. A three-dimensional 256x256x3 RGB histogram would thus have 196608 such bins when reduced to a two-dimensional histogram. When indexing an image, the color of each pixel is determined, and the count of the relevant bin is increased by one. To compare histograms of database picture and query image, specific codes for all histogram bins must first be generated. For RGB histogram bins, (r: 0-255, g: 0-255, b: 0-255) codes were created using this procedure. A mechanism for comparing histograms is required once the images have been quantized.

Poisena et al [1] offered the first image query performance prediction benchmark (iQPP). They created a collection of four data sets (PASCAL VOC 2012, Caltech-101, ROxford5k, and RParis6k) and used two cutting-edge image retrieval models to assess the ground-truth complexity of each query as the average precision or precision@k. Following that, they proposed and assessed novel pre- and post-retrieval query performance predictors, comparing those to existing or adapted (from text to image) predictors. Most predictors do not generalize across evaluation settings, according to the empirical results. Our extensive trials show that iQPP is a difficult benchmark, highlighting an important research need that must be addressed in future work. To encourage future research,

Meenu et al [8] proposed the KNN algorithm that is used to classify images in a database image set using query images, and the candidate images are reduced to images returned after classification, resulting in a shorter execution time and fewer iterations. As a result of the hybrid model of multi query and KNN, the effectiveness of image retrieval in the CBIR system improves. The programming language utilized in this project is C/C++ using Open CV libraries, and the IDE is Visual Studio 2015. The experimental results suggest that our strategy is more effective at improving picture retrieval performance.

III. PROPOSED METHODOLOGY

Particle Swarm Optimization (PSO) is a computational technique that replicates the movement of particles in a search space to identify optimal solutions. It does this by adjusting the particles' positions and velocities based on their individual best-known positions and the best-known position of any particle in the population, aiming to converge towards the optimal solution over iterations [9]. PSO is widely employed in optimization issues across numerous fields due to its ability to establish a balance between exploration and exploitation. The following figure 1 represents the proposed methodology where decision tree classifier technique is used to classify color, texture and shape features which can be classified and then it can be optimized using the particle swarm optimization algorithm. Combining a Decision Tree classifier with Particle Swarm Optimization (PSO) [18] is a novel strategy that can be used for feature selection, hyperparameter tuning, or even refining the decision tree's structure. PSO can be used to improve a Decision Tree classifier in the following ways:

3.1 Feature Selection:

PSO can be employed to select the most relevant features for classification. Each particle in the PSO population represents a subset of features, and the fitness function evaluates the performance of a Decision Tree classifier [10] using only those selected features. PSO then iteratively refines these subsets to find the best feature combination.

3.2 Hyperparameter Tuning:

PSO can optimize hyperparameters of the Decision Tree, such as the maximum depth of the tree, minimum samples required for a split, or the criterion for splitting. The fitness function measures the classifier's performance using different combinations of hyperparameters.

3.3 Optimizing Decision Tree Structure:

PSO can even be used to optimize the structure of the Decision Tree itself, such as the number of nodes or splits. Each particle represents a different tree structure, and PSO searches for the tree structure that yields the best classification performance.

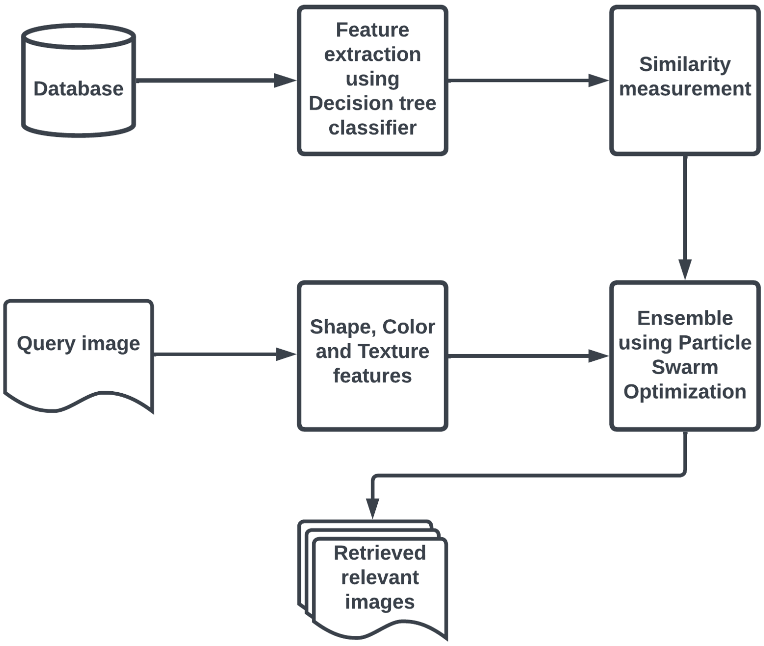


Figure1: The proposed CBIR system

The Particle Swarm Optimization algorithm can be stated as follows which is used to optimize the performance of the decision tree classifier.

Table 1: Particle Swarm Optimization Algorithm

The extracted color, shape and texture features are given as inputs and the relevant images based on these features is the output. The algorithm can be summarized as

1. Create a population of particles with random placements and speeds.
2. Determine the fitness of each particle's current location.
3. Based on their fitness, update each particle's personal best position.
4. Update the global best position depending on all particles' personal best placements.
5. Each particle's velocity and position are updated using equations that consider their current velocity, personal best, and global best.
6. Steps 2–5 must be repeated for a set number of iterations or until a convergence requirement is fulfilled.

The first is the location vector of the particle's best solution (fitness) so far [11]. The fitness level is also saved. This position is known as pbest. Another "best" location that the particle swarm optimizer keeps track of is the best position acquired thus far by any particle in the population. This best position is the current global best, and it is referred to as gbest. Following the determination of the two best values, the position and velocity of the particles are updated using the following two equations:

$$\begin{aligned} v_i^k &= w v_i^k + c_1 r_1\left(\text{ pbest }_i^k - x_i^k\right) + c_2 r_2\left(\text{ gbest }^k - x_i^k\right) \\ x_i^{k+1} &= x_i^k + v_i^{k+1} \end{aligned} \tag{1}$$

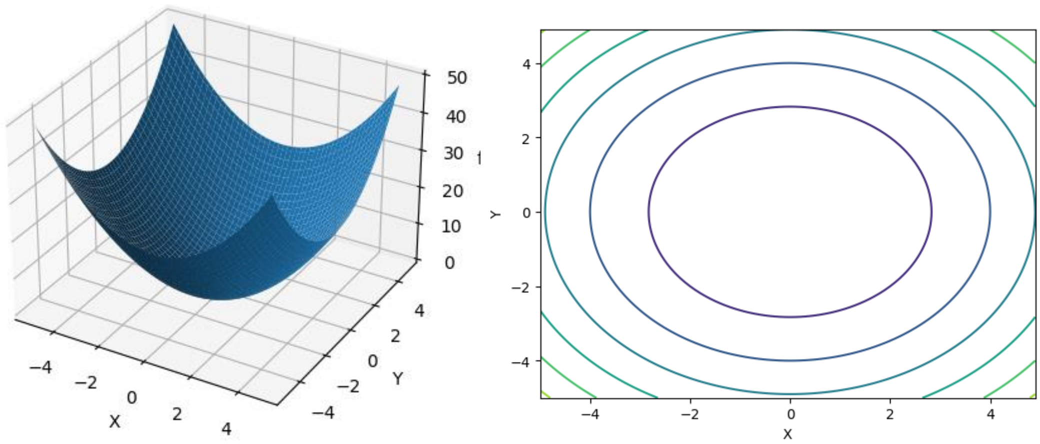
Where v_i^k is the velocity of the i th particle at the k th iteration and x_i^k is the th particle's current solution (or position) of the i th particle at the k th iteration [12]. Also c1 and c_2 are both positive constants, and r_1 and r_2 are two random variables with uniform distributions [19] ranging from 0 to 1. The inertia weight in this equation shows the effect of the prior velocity vector on the new vector.

The De Jong function is a commonly used benchmark function in optimization and mathematical optimization problems. There are several variations of the De Jong function, but a common one is the De Jong's Function 1, also known as De Jong's Sphere Function:

$$f(x) = \prod_{i=1}^{n} x_i^2 \qquad (2)$$

Where f(x) is the value of the function

x is vector of n variables, often representing the coordinates in the vector space. To create surface and contour plots of the De Jong function, you need to choose a specific dimensionality[20] and define a range of values for each variable Then, you calculate the function values for all combinations of within the specified ranges and create the plots. The figure obtained by creating these plots using Python and the matplotlib library [13] is given as follows.



(a) Surface Plot (b) Contour Plot

Figure 2: The De Jong function surface and contour plot

3.11 Evaluation of Performance

Precision is the most commonly used measures for performance evaluation CBIR system. The Precision is calculated as follows:

$$\text{Precision} = \frac{\textit{Number of relevant images retrieved}}{\textit{Number of retrieved images}} \qquad (3)$$

In general, any CBIR approach returns a fixed number of positive integer pictures. This is known as the system's scope. A precision value is assigned to each image in the database, which is then averaged across all photos in the database. The greater the scope, the more relevant photos are retrieved, resulting in lower precision numbers.

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

IV. RESULTS AND DISCUSSION

4.1 Results of CBIR approaches

In this section, we present all of our findings, including the best architecture, our CBIR system's retrieval performance in relation to sample query images taken from our datasets, the precision of category-level image retrieval for our datasets, and a comparison of the proposed method's precision with that of some other recent CBIR approaches.

Throughout the study, we draw on two independent datasets, each with its own collection of categories and image types. The suggested models partition the datasets into training and test images using 80% of the training photos and % of the test images. The Caltech101 dataset contains images from 101 object categories (e.g., “butterfly”, “leopard” and “airplane” etc.) and a background category that contains the images not from the 101 object categories. For each object category, there are about 40 to 800 images, while most classes have about 50 images. The resolution of the image is roughly about 300×200 pixels.

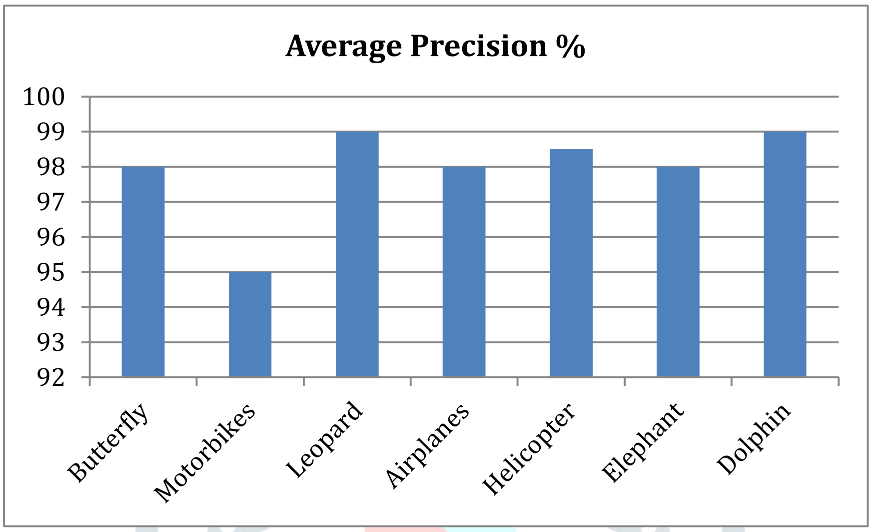


Figure 3: Precision values of different classes from Caltech 101 dataset

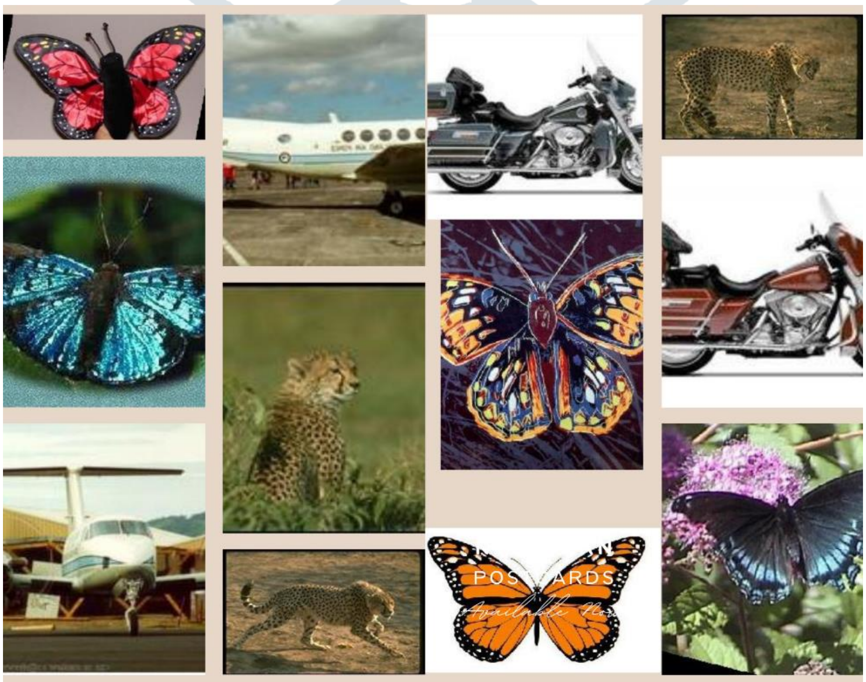


Figure 4: Sample images from Caltech 101

The second dataset is Oxford 102 dataset which contains 102 classes. Each class consists of between 40 and 258 images.

Figure 5: Example of classes from Oxford 102 dataset

4.2 Result comparison with other recently proposed algorithms

The proposed method outperformed other methods as compared in the Table 1. The precision % obtained for the Caltech 101 dataset was 98.25 whereas for Oxford 102 dataset the average precision % achieved was 98.79. The results have been summarized and it has been concluded that the proposed method performance is far superior to other methods over the same datasets.

Table 2: Result comparison of proposed model with other methods

Methods	Average Precision % Caltech 101	Average Precision % Oxford 102
Proposed Method	**98.25**	**98.79**
Ravi et al	96.00	95.40
Fadaei et al	95.13	96.15
Arshiya et al	92.30	92.20
Arun et al	91.35	89.45
Eduard et al	89.43	97.10
Meenu et al	93.16	96.25

The Comparison of average precision between our proposed method and recent papers' methods were carried on Caltech 101 and Oxford 102 datasets. The two datasets are publicly available on Kaggle website.

V. CONCLUSION

The proposed model performed well on the basis of color, shape and texture for the decision tree classifier. The features are combined to analyze the use of image retrieval by using particle swarm optimization and turned out to be more efficient in retrieval of images in relevance as well as the execution time can be reduced for databases with limited number of images having size of 600 and 800 etc. The execution time may depend on the size of the database used and type of images the database contains. In larger databases the execution time can be a little bit more but it can be reduced by other optimization techniques and using deep learning models. As a future research the use of Artificial Swarm Optimization and Fuzzy feed forward neural networks can be used for a better performance.

REFERENCES

[1] Eduard Poesina, Radu Tudor Ionescu, and Josiane Mothe. 2023. iQPP: A Benchmark for Image Query Performance Prediction. In Proceedings of Arxiv (Preprint). ACM, New York, NY, USA

[2] Md. Mohsin Kabir, Adit Ishraq1, Kamruddin Nur, and M. F. Mridha, Content-Based Image Retrieval Using AutoEmbedder, Journal of Advances in Information Technology Vol. 13, No. 3, June 2022.

[3] Mhd Furqan, Muhammad Ikhsan, Rizki Saidah Srg, Analysis Of Content-Based Image Retrieval (Cbir) And Decision Tree Methods For Identification Types Of Coffee Seeds, http://infor.seaninstitute.org/index.php/infokum/index JURNAL INFOKUM, Volume 10, No.4, October 2022 ISSN : 2302-9706.

[4] Ravi Rastogi, Rohan Appasaheb Borgalli, An Effective Content Based Image Retrieval (CBIR) System based on Model Approach, IJSRD - International Journal for Scientific Research & Development| Vol. 7, Issue 11, 2020 | ISSN (online): 2321-0613

[5] Anita N. Ligade, Manisha R. Patil, Content Based Image Retrieval Using Interactive Genetic Algorithm with Relevance Feedback Technique, Anita N. Ligade et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5610-5613

[6] Arshiya Simran, Shijin Kumar P.S and Srinivas Bachu, Content Based Image Retrieval Using Deep Learning Convolutional Neural Network, IOP Conf. Series: Materials Science and Engineering 1084 (2021) 012026 IOP Publishing doi:10.1088/1757-899X/1084/1/012026

[7] Arun JB and Reshu Choudhary, Image Mining for Retrieval of Image using Histogram Technique, International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

[8] Meenu , Sonika Jindal, Enhanced Multiquery System Using Knn For Content Based Image Retrieval, I S S N 2 2 7 7 - 3 0 6 1 V o l u m e 1 6 N u m b e r 1 , International Journal of Computers & Technology.

[9] Fatima Shaheen1, R. L. Raibagkar, Efficient Content-Based Image Retrieval System with Two-Tier Hybrid Frameworks, Applied Computer Systems ISSN 2255-8691 (online) ISSN 2255-8683 (print) December 2022, vol. 27, no. 2, pp. 166–182 https://doi.org/10.2478/acss-2022-0018 https://content.sciendo.com.

[10] Kommineni Jenni, Satria Mandala, Mohd Shahrizal Sunar, Content Based Image Retrieval Using Colour Strings Comparison, 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15).

[11] Konstantin Schall, Kai Uwe Barthel, Nico Hezel, Klaus Jung, Gpr1200: A Benchmark For General-Purpose Content-Based Image Retrieval, arXiv:2111.13122v1 [cs.CV] 25 Nov 2021.

[12] Krishna Moorthy, Kannedari Gopikrishna, Rambabu M, Image Processing In Data Mining By Using Color K-Means Clustering, National Conference on Engineering, Science, Technology in Industrial Application and Significance of Free Open Source Softwares. National Conference Proceeding NCESTFOSS Dec 2017| ISSN: 2320-2882.

[13] Maneela Jain, Pushpendra Singh Tomar, Manish Shrivastava, Content based Image Retrieval using SVM-ID3, International Journal of Computer Applications (0975 – 8887) Volume 87 – No.17, February 2014.

[14] Nitin Arora, Aditya Kakde, Subhash C. Sharma, An optimal approach for content‑based image retrieval using deep learning on COVID‑19 and pneumonia X‑ray Images, Int J Syst Assur Eng Manag (March 2023) 14(Suppl. 1):S246–S255 https://doi.org/10.1007/s13198-022-01846-4.

[15] Ramesh Babu Durai, Dr.V.Duraisamy, A Generic Approach To Content Based Image Retrieval Using Dct And Classification Techniques, International Journal on Computer Science and Engineering Vol. 02, No. 06, 2010, 2022-2024.

[16] R.SahayaNanthini, M. Sunganya, D.Saravanan, A.Jesudoss, Content Based Image Retrieval Using Image Feature, International Journal of Power Control and Computation(IJPCSC) Vol 6. No.1 – Jan-March 2014 Pp. 13-17 ©gopalax Journals, Singapore available at : www.ijcns.com ISSN: 0976-268X.

[17] Sarmad T. Abdul-Samad, Sawsan Kamal, Image Retrieval Using Data Mining Technique, Iraqi Journal of Science, 2020, Vol. 61, No. 8, pp: 2115-2125 DOI: 10.24996/ijs.2020.61.8.26.

[18] Shereena V.B. and Julie M. David, Content Based Image Retrieval: Classification Using Neural Networks, The International Journal of Multimedia & Its Applications (IJMA) Vol.6, No.5, October 2014,

[19] Suneel Kumar, Manoj Kumar Singh, Manoj Kumar Mishra, Improve Content-based Image Retrieval using Deep learning model, 4th International Conference on Intelligent Circuits and Systems Journal of Physics: Conference Series 2327 (2022) 012028 ,IOP Publishing doi:10.1088/1742-6596/2327/1/012028.

[20] Zakariya, S. M.; Ali, Rashid; and Ahmad, Nesar (2012) "Unsupervised Content Based Image Retrieval by Combining Visual Features of an Image With A Threshold," International Journal of Computer and Communication Technology: Vol. 3 : Iss. 3, Article 1. DOI: 10.47893/IJCCT.2012.1131, Available at: https://www.interscience.in/ijcct/vol3/iss3/1