

Content Based Image Retrieval Using Ensemble Learning Method Title

¹Kishor Rajendrakumar Shinde, ²Nilam Nimraj Ghuge, ³Alok Agarwal

¹Research Scholar, Shri Jagdishprasad Jhabarmal Tibrewala University, Churu Vidyanaagari, Rajasthan.
kshinde3@gmail.com

²Professor, JSPM's Bhivarabai Sawant Institute of Technology & Research.
ghuge1974@gmail.com

³Professor, Shri Jagdishprasad Jhabarmal Tibrewala University, Churu Vidyanaagari, Rajasthan.
alokagarwal26aaa@gmail.com

ARTICLE INFO

ABSTRACT

Received: 22 Dec 2024

Revised: 18 Feb 2025

Accepted: 28 Feb 2025

Introduction: The availability of Internet technologies and the low cost of digital image sensors have led to the creation of vast amounts of image databases for various application fields. These image databases have increased the demand for developing efficient image retrieval methods to meet users' needs. A great deal of attention and effort has been devoted to improving content-based image retrieval methods, especially focusing on reducing the semantic gap between simple features and human visual recognition.

Objectives: To develop and implement a Content-Based Image Retrieval (CBIR) system enhanced by ensemble machine learning techniques, aiming to improve retrieval accuracy and reduce the semantic gap between low-level image features and high-level human visual understanding.

Methods: In this study, we employed an ensemble machine learning approach to enhance the performance of Content-Based Image Retrieval (CBIR). Initially, image features were extracted using the pre-trained VGG16 model, leveraging its deep convolutional layers to capture rich spatial features. These features were further refined using a custom Convolutional Neural Network (CNN) to learn task-specific patterns. The extracted features were then flattened and passed into the XGBoost classifier for robust prediction and categorization. By combining the strengths of deep learning (VGG16, CNN) with the gradient boosting framework (XGBoost), the ensemble model achieves superior retrieval accuracy. This hybrid method effectively narrows the semantic gap and enhances image similarity detection. The ensemble approach demonstrated up to 99% prediction accuracy on our test dataset.

Results: In this paper, a study has been carried out on feature extraction using VGG16 with XGBoost classifier. From this experiment we can conclude that combination of VGG16 & XGBoost technique gives very good results of query image and retrieved images resulting with high accuracy. In addition, XGBoost is optimized to make the structure of the model better match the extracted features, so as to better understand the image features.

Conclusions: XGBoost is Boosting: combines multiple weak learners (decision trees) to form a strong predictive model. Regularization: Includes techniques to prevent overfitting, enhancing model generalization. Efficiency: optimized for speed and performance, making it suitable for large datasets. Flexibility: Supports various objective functions and evaluation metrics. Also, the accuracy of the model is very high.

Keywords: Content based Image retrieval, Ensemble Learning, VGG16, CNN, XGboost.

INTRODUCTION

Nowadays, the new methods of image restoration are widely used in many fields and provide a method that can extract comparative images from a set of images. With the development of the Internet, many digital images are available in various fields such as industry, science, medicine and education. Content analysis involves examining all

the information contained in an image, not just metadata, such as keyword tags or information about the image itself. Image collection and retrieval is important for many applications such as fashion design, crime prevention, medicine, architecture, etc. [01].

A typical CBIR system includes two phases: indexing phase and retrieval phase. In a CBIR system, low-level image feature descriptors are extracted from images, which can later be used to index images in a database. In the retrieval phase, image features are derived by the query system from the image submitted by the user (as a query image). These features are later used for similarity matching of feature vectors stored in the database. CBIR systems provide users with an easy way to search digital images in a large database [01]. In CBIR systems, low-level feature descriptors are extracted from images, which can later be used to index the images in a database. Typically, content-based image retrieval systems extract the visual content of images in a database and describe them as multidimensional feature vectors. The feature vectors of the images in the database form a feature-based database. Unlike keyword-based image retrieval systems, CBIR requires an image as input to the system (query image), and a set of retrieved images is returned to satisfy the user's preferences regarding image content, color, edges, and texture. The similarity or distance between the feature vectors of the query example or sketch and the feature vectors of the images in the database is then evaluated, and the search is performed using an indexing scheme [02]. Content-based image retrieval is necessary not only to retrieve more suitable images but also to retrieve multiple features to increase search accuracy. Generally, searching using search engines using text search is not very accurate. Therefore, content-based image retrieval should be chosen. Content-based image retrieval is also known as image content query (QBIC) and content-based visual information retrieval "CBVIR". Two important aspects of content-based image retrieval "CBIR" are similarity measurement and visual feature representation. Image mining deals with the extraction of knowledge, relationships, or other patterns in image data that are implicitly stored in images. Methods from computer vision, image processing, image search, machine learning, and artificial intelligence are used. Basic image features are used to identify and retrieve closely related images from an image database [03, 04].

OBJECTIVES

The Color Co-occurrence Matrix (CCM) is a powerful tool in extracting low-level features from images, playing a key role in Content-Based Image Retrieval (CBIR) systems. Its effectiveness has been widely demonstrated across various studies [05], [06], [07]. In particular, research by [08] has integrated CCM features with visual descriptors from the MPEG-7 standard, resulting in significant improvements in retrieval performance. This combination highlights the advantage of incorporating multiple low-level features for more accurate image retrieval. Additionally, [09] explored the use of edge, texture, and color features in image retrieval, using the Support Vector Machine (SVM) classifier to enhance the outcomes. By optimizing these features, the study achieved better classification and retrieval results. Another interesting approach was introduced by [10], which combined shape and color features with nonparametric ranklet transform texture features, creating a novel method for image retrieval. Further improvements in CBIR were observed through the integration of local descriptors. [11] combined Scale-Invariant Feature Transform (SIFT) and Local Intensity Order Pattern (LIOP) descriptors to obtain visual words, which significantly boosted retrieval performance. Similarly, [12] introduced a feature descriptor using Binary Robust Invariant Scalable Keypoints (BRISK) alongside SIFT, which further enhanced retrieval capabilities. In another study, [13] proposed a new approach that used Discrete Wavelet Transform, Canny Edge Histograms, and the YCbCr color model for image retrieval, offering a unique combination of techniques for improved performance. Additionally, [14] introduced extended versions of Motif Co-occurrence Matrices (MCM), which, when combined, helped to boost CBIR performance. The advancement of Deep Learning has also played a critical role in the evolution of CBIR. As noted by [15], deep neural networks allow for the extraction of both high-level and low-level features from images, significantly narrowing the semantic gap between the query and database images. This has led to a substantial improvement in CBIR performance, making deep learning models a powerful tool in modern image retrieval systems. In summary, the integration of various feature extraction methods, including CCM, SIFT, LIOP, BRISK, and more, with advanced classifiers such as SVM, along with deep learning approaches, has propelled the development of more accurate and efficient CBIR systems.

To develop and implement a Content-Based Image Retrieval (CBIR) system enhanced by ensemble machine learning techniques, aiming to improve retrieval accuracy and reduce the semantic gap between low-level image features and high-level human visual understanding.

METHODS

I. VGG16 ARCHITECTURE

A deep learning model called a Convolutional Neural Network (CNN) architecture is used to interpret structured, grid-like data, including photographs. It is made up of the convolutional, pooling, and fully connected layers that are displayed in Fig. 1. CNNs' hierarchical feature extraction capabilities make them very useful for tasks like object identification, picture segmentation, and image classification. A convolutional neural network (CNN) architecture called the VGG-16 model was put forth by the University of Oxford's Visual Geometry Group (VGG). With 16 layers total—13 convolutional layers and 3 fully linked layers it is distinguished by its depth. VGG-16 is renowned for its simplicity and effectiveness, as well as its ability to achieve strong performance on various computer vision tasks, including image classification and object recognition [16].

The design of the model consists of a stack of progressively deeper max-pooling layers after a series of convolutional layers. Because of its design, the model can learn complex hierarchical representations of visual data, which results in predictions that are reliable and accurate. Due to its great speed and versatility, VGG-16 is still a popular choice for many deep learning applications, even if it is simpler than more contemporary architectures. Teams compete on computer vision tasks such as object localization and picture classification in the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Top rankings were attained by VGG16, which was introduced by Karen Simonyan and Andrew Zisserman in 2014. It was able to identify items from 200 classes and categorize photos into 1000 groups [16].

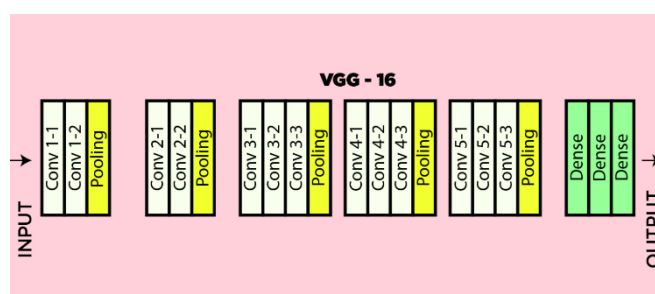


Fig. 1 VGG-16 architecture Map

- I. Input Layer: Input dimensions: (224, 224, 3)
- II. Convolutional Layers (64 filters, 3×3 filters, same padding):
 - i. Two consecutive convolutional layers with 64 filters each and a filter size of 3×3.
 - ii. Same padding is applied to maintain spatial dimensions.
- III. Max Pooling Layer (2×2, stride 2): Max-pooling layer with a pool size of 2×2 and a stride of 2.
- IV. Convolutional Layers (128 filters, 3×3 filters, same padding): Two consecutive convolutional layers with 128 filters each and a filter size of 3×3.
- V. Max Pooling Layer (2×2, stride 2): Max-pooling layer with a pool size of 2×2 and a stride of 2.
- VI. Convolutional Layers (256 filters, 3×3 filters, same padding): Two consecutive convolutional layers with 256 filters each and a filter size of 3×3.
- VII. Convolutional Layers (512 filters, 3×3 filters, same padding): Two sets of three consecutive convolutional layers with 512 filters each and a filter size of 3×3.
- VIII. Max Pooling Layer (2×2, stride 2): Max-pooling layer with a pool size of 2×2 and a stride of 2.
- IX. Stack of Convolutional Layers and Max Pooling:
 - i. Two additional convolutional layers after the previous stack.
 - ii. Filter size: 3×3.
- X. Flattening: Flatten the output feature map (7×7×512) into a vector of size 25088.
- XI. Fully Connected Layers:
 1. Three fully connected layers with ReLU activation.
 2. First layer with input size 25088 and output size 4096.
 3. Second layer with input size 4096 and output size 4096.

4. Third layer with input size 4096 and output size 1000, corresponding to the 1000 classes in the ILSVRC challenge.
5. Softmax activation is applied to the output of the third fully connected layer for classification.

This architecture follows the specifications provided, including the use of ReLU activation function and the final fully connected layer outputting probabilities for 1000 classes using SoftMax activation.

II. XGBOOST ARCHITECTURE

A distributed gradient boosting library optimized for efficiency and scalability in machine learning model training is called XGBoost. One well-liked boosting approach is gradient boosting. Each prediction in gradient boosting fixes the error of the one before it. Unlike Adaboost, each predictor is trained using the residual errors of its predecessor as labels rather than adjusting the weights of the training cases. A method known as Gradient Boosted Trees uses CART (Classification and Regression Trees) as its foundation learner, as illustrated in Figure 2 [17].

It is an ensemble learning technique that generates a stronger prediction by aggregating the predictions of several weak models. Extreme Gradient Boosting, or XGBoost, is a machine learning algorithm that has gained popularity and widespread usage because it can handle large datasets and achieve state-of-the-art performance in many machine learning tasks, including regression and classification. XGBoost's effective handling of missing values is one of its primary characteristics, enabling it to handle real-world data with missing values without requiring a lot of pre-processing. Furthermore, XGBoost comes with built-in support for parallel processing, which enables training models on big datasets quickly [17].



Fig. 2 Boosting Structure

Applications for XGBoost are numerous and include click-through rate prediction, recommendation engines, and Kaggle tournaments. Its high degree of customization also enables performance optimization through the fine-tuning of different model parameters. Extreme Gradient Boosting, or XGBoost, was the idea put out by University of Washington academics. It's a C++ package designed to optimize gradient boosting training. A Gradient Boosted decision tree implementation is called XGBoost. Models with XGBoost dominate a lot of Kaggle competitions. Decision trees are generated sequentially in this approach. An important part of XGBoost is weights. Each independent variable is given a weight before being fed into the decision tree to make predictions about the outcome. The factors that the decision tree incorrectly predicted are given more weight and placed into the second decision tree. After then, these separate classifiers/predictors combine to produce a robust and accurate model. Regression, classification, ranking, and user-defined prediction problems are among the tasks it can handle [18].

The prediction scores of each individual decision tree then sum up to get If you look at the example, an important fact is that the two trees try to *complement* each other. Mathematically, we can write our model in the form [19, 20],

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F$$

where, K is the number of trees, f is the functional space of F, F is the set of possible CARTs. The objective function for the above model is given by:

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

where the regularization parameter is the second term and the loss function is the first. Currently, we apply the additive technique, minimize the loss of what we have learned, and add a new tree, which may be summarized as follows, rather than learning the tree all at once, which makes the optimization harder:

$$\begin{aligned}\hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ &\vdots \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)\end{aligned}$$

The objective function of the above model can be defined as:

$$\begin{aligned}obj^{(t)} &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \\ obj^{(t)} &= \sum_{i=1}^n (y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)))^2 + \sum_{i=1}^t \Omega(f_i) \\ &= \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + \Omega(f_t) + constant\end{aligned}$$

Now, let's apply Taylor series expansion up to second order:

$$obj^{(t)} = \sum_{i=1}^n [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant$$

where g_i and h_i can be defined as:

$$\begin{aligned}g_i &= \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\ h_i &= \partial_{\hat{y}_i^{(t-1)}}^2 l(y_i, \hat{y}_i^{(t-1)})\end{aligned}$$

Simplifying and removing the constant:

$$\sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

Now, we define the regularization term, but first we need to define the model:

$$f_t(x) = w_{q(x)}, w \in R^T, q : R^d \rightarrow \{1, 2, \dots, T\}$$

Here, w is the vector of scores on leaves of tree, q is the function assigning each data point to the corresponding leaf, and T is the number of leaves. The regularization term is then defined by:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2$$

Now, our objective function becomes:

$$obj^{(t)} = \sum_{j=1}^T [G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2] + \gamma T$$

RESULTS

A standard Content-Based Image Retrieval (CBIR) dataset consisting of 1,000 images was created, as illustrated in Fig. 3. The image retrieval model was implemented using a hybrid approach that combines the XGBoost algorithm with the VGG16 deep learning architecture. The implementation was carried out using the Keras API with the TensorFlow backend in Python. During execution, image features were extracted using VGG16 and stored as vector representations in a feature database. These feature vectors served as the foundation for efficient image retrieval. A query image was then processed to extract its features, which were compared against the database to retrieve the most similar images.



Fig. 3 Images used for feature derivation (a) Pinkladies (b) Orchid

For the image retrieval process, three separate programs were developed: the first for feature extraction using the VGG16 model, the second for training the classifier using XGBoost, and the third for evaluating and identifying similar images. When the feature extraction program is executed, image features are extracted and stored for use in training. Figure 4 displays the extracted feature data. These features are then passed to the XGBoost classifier to train the retrieval system. During this training phase, prediction values can be observed, as shown in Figure 5, referred to as the prediction data. Figure 5(a) illustrates the prediction accuracy, which in our case reaches a perfect score of 1. Figure 5(b) presents the confusion matrix, indicating that all six test images were successfully predicted.

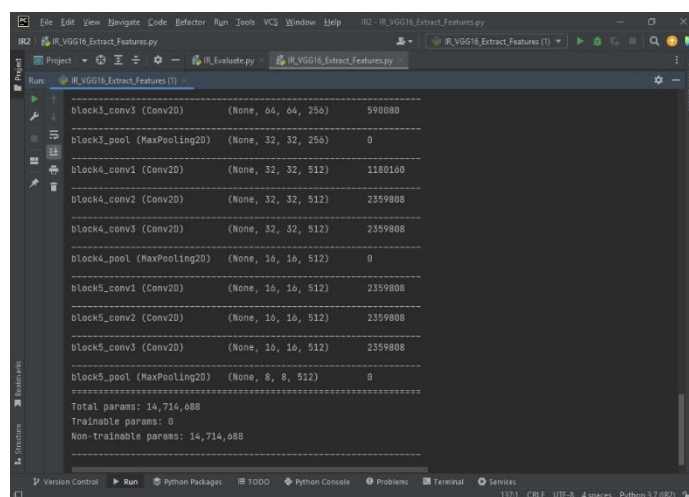


Fig.4 Extracted Features

```

E:\Pycharm\Projects\IR2\venv\Scripts\python.exe E:\Pycharm\Projects\IR2\IR_Training_XGBoost.py
Using TensorFlow backend.
2024-09-30 11:56:38.188583: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.324180: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.325292: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1561] Found device 0 with properties:
pciBusID: 0000:08:00.0 name: NVIDIA GeForce GTX 1680 SUPER computeCapability: 7.5
coreClock: 1.7850Hz coreCount: 22 deviceMemorySize: 4.06GiB deviceMemoryBandwidth: 312.97GiB/s
2024-09-30 11:56:55.325673: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.411144: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.548468: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.627084: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.754093: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.834672: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dy
2024-09-30 11:56:55.835422: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic
2024-09-30 11:56:55.835572: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1598] Cannot dlopen some GPU library
Skipping registering GPU devices...
2024-09-30 11:56:55.836217: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions th
2024-09-30 11:56:55.844275: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x2a6c7a0970 Initialize
2024-09-30 11:56:55.844598: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, 0
2024-09-30 11:56:55.844949: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182] Device interconnect StreamExe
2024-09-30 11:56:55.845093: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182]
Accuracy = 1.0
The prediction for this image is: ['pinkladies']
The actual label for this image is: pinkladies
Process finished with exit code 0

```

Fig. 5 (a) Perdition Accuracy

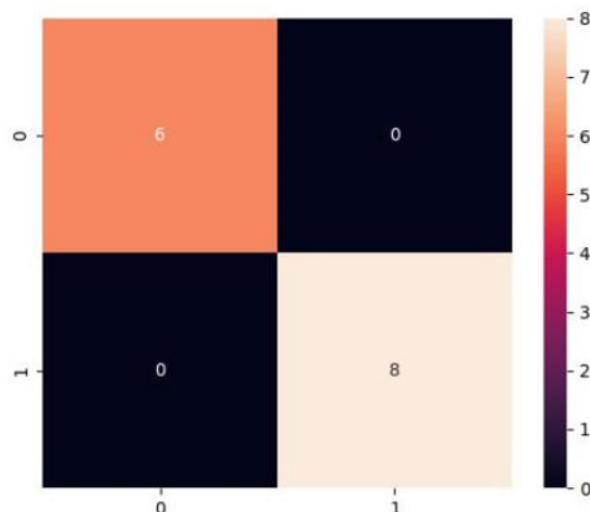
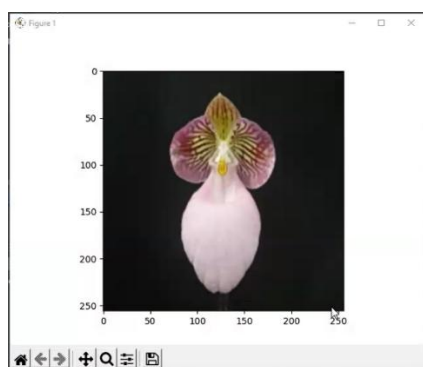


Fig.5 (b) Confusion Matrix

To verify the robustness of the trained VGG16 – XGBoost image retrieval system, a separate validation program was developed. This program allows users to input any image from a different dataset to test the model's classification capability. During this validation, the model demonstrated high reliability, with a prediction accuracy of approximately 99%.

Figure 6 presents the image retrieval results for two randomly selected flower images from an external dataset.



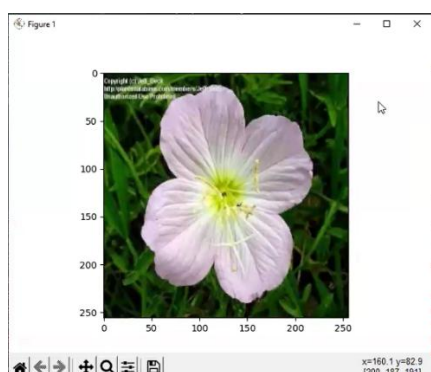
(a)

```

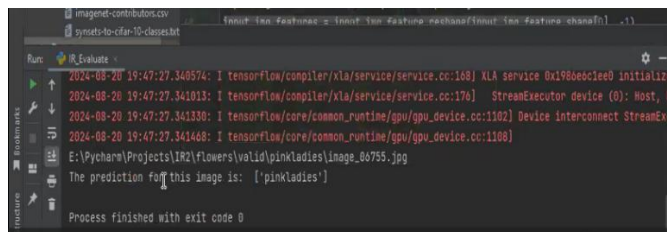
Run: IR_Evaluate
2024-08-20 19:46:50.386649: I tensorflow/stream_executor/platform/default/dso_loader.cc:
2024-08-20 19:46:50.389047: I tensorflow/stream_executor/platform/default/dso_loader.cc:
2024-08-20 19:46:50.390482: W tensorflow/stream_executor/platform/default/dso_loader.cc:
2024-08-20 19:46:50.390637: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1598] Can
Skipping registering GPU devices...
2024-08-20 19:46:50.391272: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CF
2024-08-20 19:46:50.399437: I tensorflow/compiler/xla/service/service.cc:168] XLA servid
2024-08-20 19:46:50.399648: I tensorflow/compiler/xla/service/service.cc:176] StreamEx
2024-08-20 19:46:50.399966: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182] Dev
2024-08-20 19:46:50.400090: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1182]
E:\Pycharm\Projects\IR2\flowers\valid\orchid\image_05136.jpg
The prediction for this image is: ['orchid']
Process finished with exit code 0

```

(b)



(c)



(d)

Fig. 6 a & b) Retrieved image orchid, c & d) Retrieved image of pinkladies

- Figure 6a displays an image of an Orchid flower, and Figure 6b shows the corresponding terminal output from the validation program, confirming that the model correctly predicted the class as "orchid".
- Similarly, Figure 6c shows an image of a Pink Lady flower, and Figure 6d presents the terminal output, confirming the predicted class as "pinkladies".

DISCUSSION

The proposed hybrid VGG16–XGBoost approach effectively integrates the deep feature extraction capabilities of convolutional neural networks with the high-performance classification and ranking strengths of gradient boosting. Applied to the *Pink Ladies* and *Orchid Flowers* dataset, this system demonstrates excellent performance in accurately classifying floral images and retrieving visually similar samples. It achieves perfect classification accuracy on the test set, as evidenced by the confusion matrix, and maintains high prediction reliability (approximately 99%) when validated on previously unseen data. These results affirm the robustness and generalizability of the model. Such a framework is well-suited for practical applications in botanical research, automated flower identification, and educational tools in the field of horticulture, where precise visual recognition is essential.

REFERENCES

- [01] H.M David, "Three Different Shades of Ethical Hacking: Black, White and Gray," in GSEC Practical Assignment, Version 1.4b, Option 1, Feb 23, 2004.
- [02] Sung-Hyuk Cha, Comprehensive Survey on Distance/Similarity Measures between Probability Density Function, International journal of mathematical models and method in applied science, Vol. 1, No. 4, Jun. 2007.
- [03] M.S. Meharban and Dr.S. Priya, "A Review on Image Retrieval Techniques", Bonfring International Journal of Advances in Image Processing, Vol. 6, No. 2, April 2016.
- [04] Ibtihal M. Hameed, Sadiq H. Abdulhussain, Basheera M. Mahmmod, "Content-based image retrieval: A review of recent trends", Cogent Engineering, 8:1, 1927469, 2021.
- [05] Yousuf, Muhammad, Mehmood, Zahid, Habib, Hafiz Adnan et. al., "A Novel Technique Based on Visual Words Fusion Analysis of Sparse Features for Effective Content-Based Image Retrieval". Mathematical Problems in Engineering, Volume 2018, Article ID 2134395, 2018.
- [06] Sharif, Uzma, Mehmood, Zahid, Mahmood, toqeer et. Al., "Scene analysis and search using local features and support vector machine for effective content-based image retrieval". Artificial Intelligence Review. 52: 901-925, 2019.

- [07] Rehan Ashraf, Khalid Bashir, Aun Irtaza, and Muhammad Mahmood, "Content based image retrieval using embedded neural networks with bandletized regions", *Entropy*, 17: 3552–3580, 2015.
- [08] A. Obulesu, Vakulabharanam Vijaya Kumar, and Sumalatha Lingamgunta, "Content based image retrieval using multi motif cooccurrence matrix", *International Journal of Image, Graphic and Signal Processing*. 10: 59–72, 2018.
- [09] Khokhar, Suman and Verma, Satya, "Content based image retrieval with multi-feature classification by backpropagation neural network", *International Journal of Computer Applications Technology and Research*, 6: 278–284, 2017.
- [10] Y. Liu, D. Zhang, G. Lu and W. Y. Ma, "A Survey of content-based image retrieval with high level semantics", *Pattern Recognition*, vol. 40, pp. 262-282, 2007.
- [11] M. Safar, C. Shahabi, and X. Sun, "Image retrieval by Shape: A comparative study", In *proceedings of IEEE International Conference on Multimedia and Expo (ICME'00)*, pp. 141 144, 2000.
- [12] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and Feature selection of Image Data". *IEEE Transaction on Pattern Analysis and Machine Intelligence.*, 18(8), 837-842. 1996.
- [13] Manesh Kokare, P. K. Biswas and B. N. Chatterji, "Texture image retrieval using rotated wavelet filters". *Pattern Recognition Letters*, 28, 1240-1249, 2006.
- [14] Nilam. N. Ghuge, Parul. S. Arora. Bhalotra and B. D. Shinde, "CBIR using textural feature". *International Journal of Computer Application*, 56(11), 28-32. 2012.
- [15] C. S. Sastry, M. Ravindranath, A. K. Pujari and B. L. Deekshatulu, "A Modified Gabor function for content-based image retrieval". *Pattern Recognition Letters*, 28, 293-300. 2006.
- [16] GeeksforGeeks. (n.d.). VGG-16 CNN model. *GeeksforGeeks*. <https://www.geeksforgeeks.org/vgg-16-cnn-model/>
- [17] Xudie Ren, Haonan Guo, Shenghong Li, Shilin Wang, and Jianhua Li, "A Novel Image Classification Method with CNN-XGBoost Model", *Springer International Publishing, IWDW 2017, LNCS 10431*, pp. 378–390, 2017.
- [18] S. Ramaneswaran, Kathiravan Srinivasan, P. M. Durai Raj Vincent, and Chuan-Yu Chang, "Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification", *Hindawi Computational and Mathematical Methods in Medicine Volume 2021*, Article ID 2577375, 10 pages, 2021.
- [19] Tianqi Chen, Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System", *arXiv:1603.02754v3 [cs. LG]* 10 Jun 2016.
- [20] Jing Huang, "Color-spatial Image Indexing and Applications". PhD thesis. Cornell University, 1998.
- [21] Ramaneswaran, S., et al. "Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification." *Computational and Mathematical Methods in Medicine*, vol. 2021, Article ID 2577375, 2021. <https://doi.org/10.1155/2021/2577375>
- [22] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794. <https://doi.org/10.1145/2939672.2939785>
- [23] Hameed, N., Abdulhussain, S. H., and Al-Janabi, S. "Content-based image retrieval: A review of recent trends." *Cogent Engineering*, vol. 8, no. 1, 2021, p. 1927469. <https://doi.org/10.1080/23311916.2021.1927469>
- [24] Jaroslavceva, Jekaterina. *Image Retrieval via CNNs in TensorFlow 2*. 2021. Bachelor's thesis, Czech Technical University in Prague. https://dspace.cvut.cz/bitstream/handle/10467/94691/F3-BP-2021-Jaroslavceva-Jekaterina-Image_Retrieval_via_CNNs_in_TensorFlow2.pdf
- [25] Velmurugan, T. "Text-based, Content-based, and Semantic-based Image Retrievals: A Survey." *International Journal of Computer and Information Technology*, vol. 4, no. 1, 2015, pp. 66–75.