WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SIMILAR IMAGE FINDER USING ML

Mrs. C. Agjelia Lydia[1], Madhusree M[2], Naveen K[3], Nihal N[4],Sangeetha M[5]

Department of Computer Science and Engineering, Bachelor of Engineering, Sri Shakthi Institute of Engineering andTechnology Coimbatore-641062

ABSTRACT :

This project focuses on developing an intelligent and user-friendly system that can efficiently analyze and compare images to identify those with similar content. By leveraging advanced technologies such as computer vision, machine learning, and image processing algorithms, the system offers enhanced accuracy and performance in detecting image similarity. The Similar Image Finder provides users with the ability to upload an image and receive real-time results showing visually similar images from a defined dataset. Key features include image preprocessing, feature extraction using techniques like histogram analysis or deep learning models, and efficient comparison mechanisms to deliver accurate results. The system is supported by a secure database for storing and managing image metadata ensuring organized and optimized performance.

Keywords— Intelligent system, User-friendly Image analysis,Image comparison,Similar images,Machine learning, Image processing algorithms,Image preprocessing

INTRODUCTION

- A Similar Image Finder using Machine Learning is an application that takes an input image and finds other images that are visually similar. This kind of system is commonly used in search engines, e-commerce, and digital asset management.
- A Similar Image Finder uses machine learning and computer vision techniques to compare images based on their content rather than metadata or filenames. The main goal is to extract meaningful features from images and use those features to measure similarity.
- This system leverages machine learning algorithms, specifically deep learning techniques, to analyze the features of images and compare them based on visual content rather than metadata or keywords. By extracting meaningful patterns, the system can effectively match images that share similarities in content, texture, color, and structure.
- By automating and streamlining the image comparison process, the Similar Image Finder eliminates the need for manual searching and promotes precision in tasks like duplicate detection, content verification, and image organization.

Objective:

- The objective of a **Similar Image Finder using Machine Learning** is to create a system capable of identifying and retrieving visually similar images from a large dataset based on a given query image.
- Leverage pre-trained models like ResNet, VGG, or Inception to avoid training from scratch and benefit from learned image representations.
- Compute the distance (e.g., cosine similarity, Euclidean distance) between the query image and images in the dataset.
- Images with smaller distances are considered more similar.
- Ensure the system is scalable for handling large image databases, supporting efficient image retrieval even with millions of images.
- Design the system for fast, real-time performance when matching images, allowing for instant retrieval of similar images.
- Calculate image similarity using metrics like cosine similarity, Euclidean distance, or Manhattan distance between feature vectors.
- Provide a user-friendly interface for users to input an image and see visually similar images returned as results.

LITERATURE SURVEY

1. Content-Based Image Retrieval (CBIR):

The foundation of Similar Image Finder systems lies in Content-Based Image Retrieval (CBIR), a concept that enables image search based on visual content rather than metadata. Pentland et al. (1996) introduced early systems like QBIC (Query By Image Content), which used attributes such as color, texture, and shape for similarity detection.

2. Feature Extraction Techniques:

Effective image comparison depends on accurate feature extraction. Traditional methods like SIFT (Lowe, 2004), SURF (Bay et al., 2006), and HOG (Dalal & Triggs, 2005) laid the groundwork for image feature representation. These techniques analyze key points and edge orientation to describe image patterns. Recent research explores deep feature extraction using convolutional layers in CNNs (Convolutional Neural Networks), which significantly hand-crafted features in identifying subtle similarities (Krizhevsky et al., 2012).

3. Deep Learning in Image Similarity:

With the rise of deep learning, models such as VGGNet, ResNet and MobileNet have been successfully applied to image similarity tasks. Studies by Zhang et al. (2019) emphasize the use of transfer learning and feature vector embeddings from pre-trained CNNs for fast and accurate image retrieval. Siamese networks, introduced by Bromley et al. (1994), have also gained traction for similarity detection due to their ability to learn distance metrics between image pairs.

METHODOLOGY

Manual Search and Comparison:

Traditionally, identifying visually similar images involves manually browsing through large collections and visually comparing each image. This process is not only time-consuming but also prone to human error and inefficiency, especially when dealing with large datasets.

Keyword-Based Search Limitations:

Most existing search engines and image databases rely on metadata or keywords (like filenames, tags, or descriptions) to locate similar images. This method fails when the metadata is missing, inconsistent, or does not represent the visual content accurately, making it unreliable for true image similarity detection.

Lack of Content-Based Filtering:

Basic image viewers or file managers do not support content-based image filtering. They cannot analyze or compare visual elements such as color histograms, texture, or object shapes. This limits the ability to find visually similar images unless manually reviewed.

Limited Use of Deep Learning Models:

Some image retrieval systems use traditional computer vision algorithms (e.g., SIFT, ORB), but they lack accuracy and robustness in identifying complex visual patterns. These methods struggle with variations in lighting, angle, and object scale, leading to low precision in similarity detection.

EXISTING SYSTEM

1. Traditional Content-Based Image Retrieval (CBIR)

How It Works:

Feature Extraction: Traditional systems rely on handcrafted features such as color histograms, textures (using methods like Gabor filters), and shapes (using methods like edge detection).

Matching Algorithms: These systems compare the extracted features of the query image with those in the database using similarity measures (e.g., Euclidean distance, cosine similarity).

Drawbacks:

Limited Feature Representation: Handcrafted features are often limited in capturing complex visual patterns, such as abstract shapes or contextual **information.**

Low Accuracy for Complex Datasets: In large-scale and diverse datasets, CBIR struggles to maintain accuracy due to its reliance on simple feature descriptors.

Scalability Issues: As the database grows, matching becomes computationally expensive due to the need for exhaustive comparisons.

2.Deep Learning-Based Image Retrieval

How It Works:

Deep Convolutional Neural Networks (CNNs): These systems leverage CNNs to automatically extract complex features from images. A model is trained on a large dataset to learn high-level representations that improve similarity search.

Transfer Learning: Pretrained models such as VGG16, ResNet, and Inception are fine-tuned to suit specific datasets, reducing the need for extensive training data.

Drawbacks:

High Computational Cost: Training deep learning models requires significant computational resources, including powerful GPUs. The need for constant retraining further increases costs.

Need for Labeled Data: Deep learning models generally require large labeled datasets to achieve good performance, which might not always be available, especially in niche domains.

Overfitting: Deep models may overfit on specific data distributions and fail to generalize well to unseen data if not properly regularized.

Long Training Times: Training deep learning models from scratch can take days or even weeks, depending on the size of the dataset.

PROPOSED SYSTEM

1. Hybrid Model with Multi-Stage Feature Extraction

Objective: To combine the strengths of both handcrafted and deep learning-based features for more robust image retrieval.
How It Works:

- Stage 1: Handcrafted Features – Extract basic low-level features such as color histograms, texture, and edges using traditional computer vision techniques.
- Stage 2: Deep Features – Use pre-trained Convolutional Neural Networks (CNNs), such as ResNet, EfficientNet, or MobileNet, to extract high-level features from images.
- Stage 3: Fusion Layer – Combine handcrafted features and deep learning features using a feature fusion mechanism (e.g., concatenation, attention mechanism, or weighted sum) to create a more comprehensive feature vector.

Benefit: This multi-stage approach ensures that both simple and complex image patterns are captured, improving retrieval accuracy across different types of images.

2. Automated Image Similarity Detection: The proposed system leverages deep learning (MobileNetV2) to extract feature vectors from images and compute similarity scores automatically. It removes the need for manual visual comparison, significantly saving time and increasing accuracy.

3. Custom Dataset Support : Users can upload a query image and a custom dataset of images for comparison. The system identifies and ranks the top visually similar images based on calculated feature distances, making it applicable in educational, forensic, and commercial fields.

4. Normalized Similarity Scoring: The system uses Euclidean distance between feature vectors to calculate similarity. Images with lower distance values are more visually similar, and the top results are displayed for user analysis, providing clear and objective results.

5. User-Friendly Web Interface: Built with Flask and HTML templates, the interface allows users to upload images and view results effortlessly. The output includes the original query image and thumbnails of the top five similar images, making the system intuitive for non-technical users.

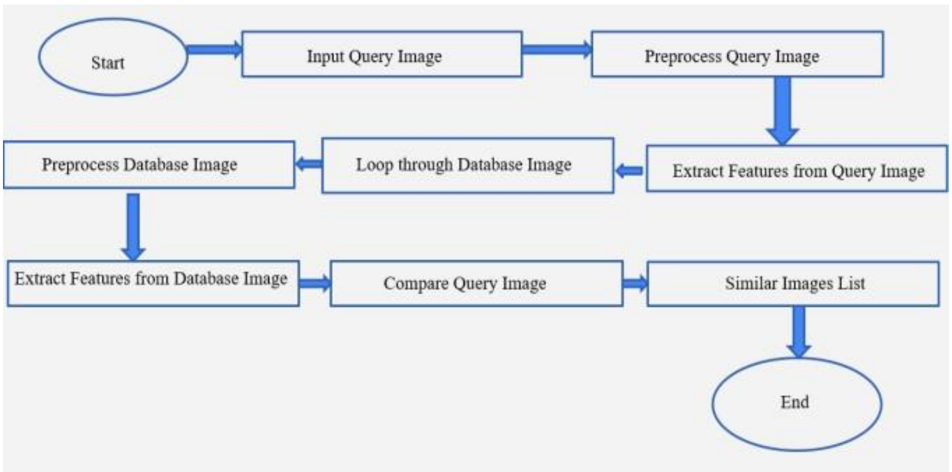
SYSTEM REQUIREMENTS

HARDWARE SPECIFICATIONS:

• 8 -16 GB RAM,
• Intel (or)AMD Ryzen processor,
• 500 GB hard disk space,
• An efficient internet service.

SOFTWARE SPECIFICATIONS:

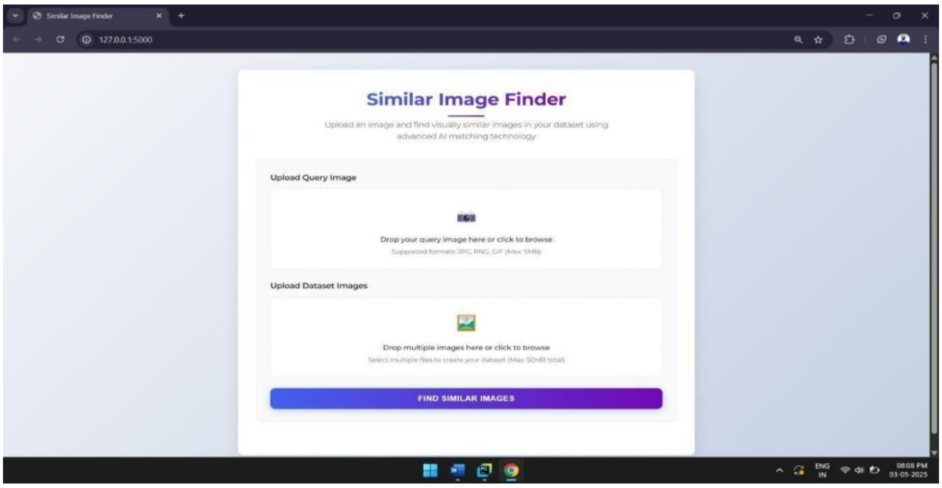
• Operating System: Windows, MacOS, Linux and above.
• Languages Used: HTM, CSS,JavaScript, Python.
• Tools: PyCharm.
PROCESSING



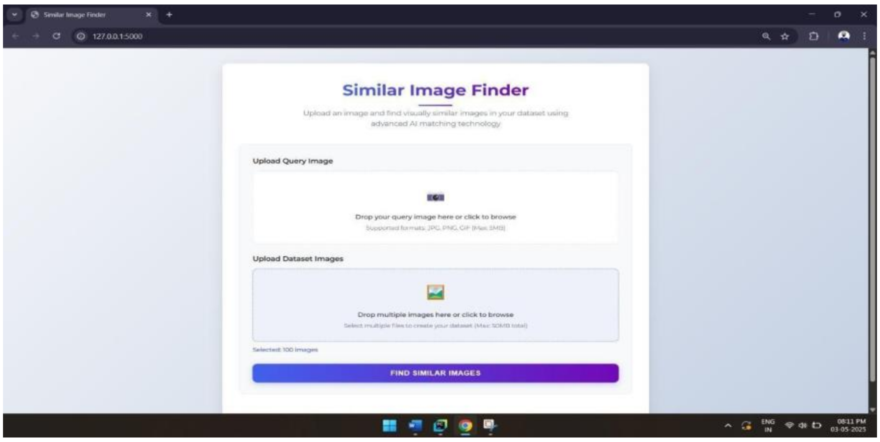
MODE OF DESCRIPTION

1. Main Interface
2. Import Dataset
3. Attach Image
4. Upload Successful
5. Result Overview

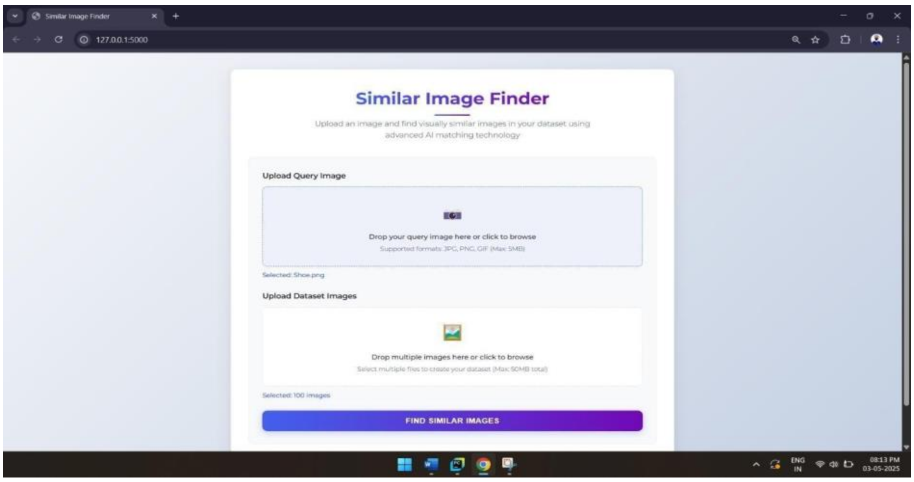
Main Interface :



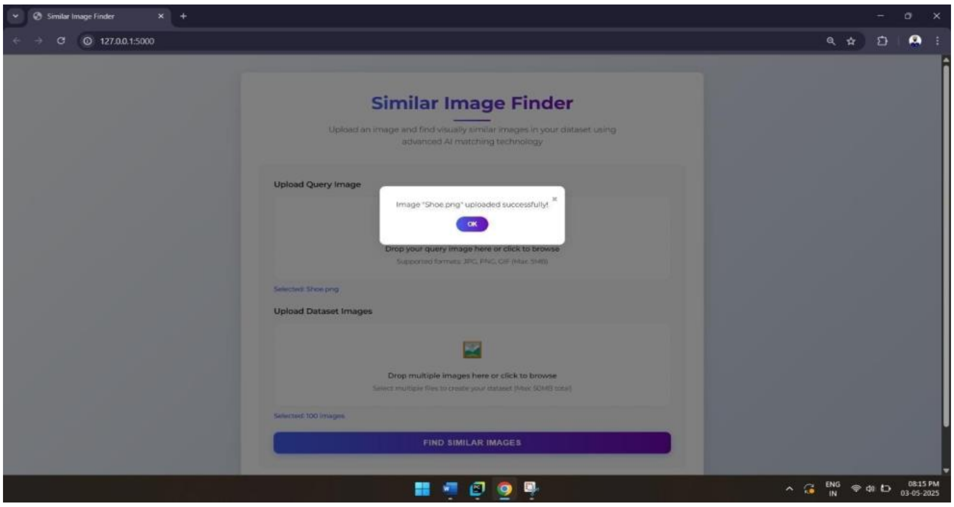
Import Dataset

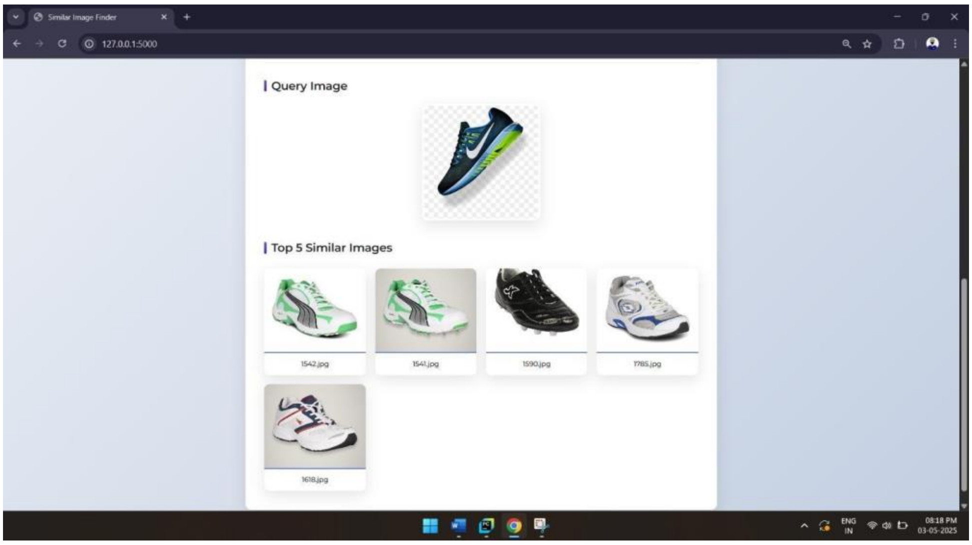


Attach Image :



Upload Successful





Conclusion :

The **SIMILAR IMAGE FINDER** project effectively addresses the need for a reliable and efficient system to identify visually similar images from a given dataset. By leveraging deep learning-based feature extraction using the MobileNetV2 model and implementing an image similarity algorithm based on Euclidean distance, the system provides accurate and consistent results across various test scenarios.

Overall, the SIMILAR IMAGE FINDER project contributes meaningfully to the field of image processing by delivering an accessible, accurate, and efficient solution for visual similarity detection, and lays the foundation for future expansion into more advanced and scalable image recognition systems

REFERENCES

1. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." Advances in Neural Information Processing Systems. 2012.
2. Simonyan, Karen, and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv preprint arXiv:1409.1556 (2014).
3. He, Kaiming, et al. "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.
4. Szegedy, Christian, et al. "Going Deeper with Convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

5. Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing Data using t-SNE." Journal of Machine Learning Research 9.Nov (2008): 2579-2605.