Project 2: Content-based Image Retrieval

Author: Manushi

Date: February 10, 2024

1. Project Overview:

This project focuses on developing a Content-Based Image Retrieval (CBIR) system, a technology designed to search and retrieve images from a database by analyzing the visual content of the images themselves, rather than relying on textual metadata. The core of the system involves extracting features from a target image and comparing these features with those of images within a database using a specified distance metric. The system is designed to return a list of images ranked by their similarity to the target image. This process involves several steps, including feature extraction, feature comparison, and ranking based on similarity. The project explores various methods of feature extraction, including simple color histograms, texture analysis, and advanced deep learning techniques, and employs different metrics for measuring the similarity between images. By implementing these techniques, the system can accurately identify and retrieve images that are visually like a given target image from a large dataset. This CBIR system has applications in various fields, such as digital image libraries, online shopping, and surveillance, making it a versatile tool in the realm of computer vision and image processing.

2. Baseline Matching:

Baseline matching is the initial approach implemented in the Content-Based Image Retrieval (CBIR) system. It involves extracting a simple feature vector from the center of each image and using the sum of squared differences as the distance metric for comparison.

Example: To illustrate baseline matching, we have a target image(pic.1016.jpg) of a red object. The CBIR system extracts a 7x7 feature vector from the center of the target image, capturing the key color information of the object. It then compares this feature vector with the feature vectors of images in the database. Images with similar color distributions, such as other red objects, will have lower distances and are considered more like the target image.

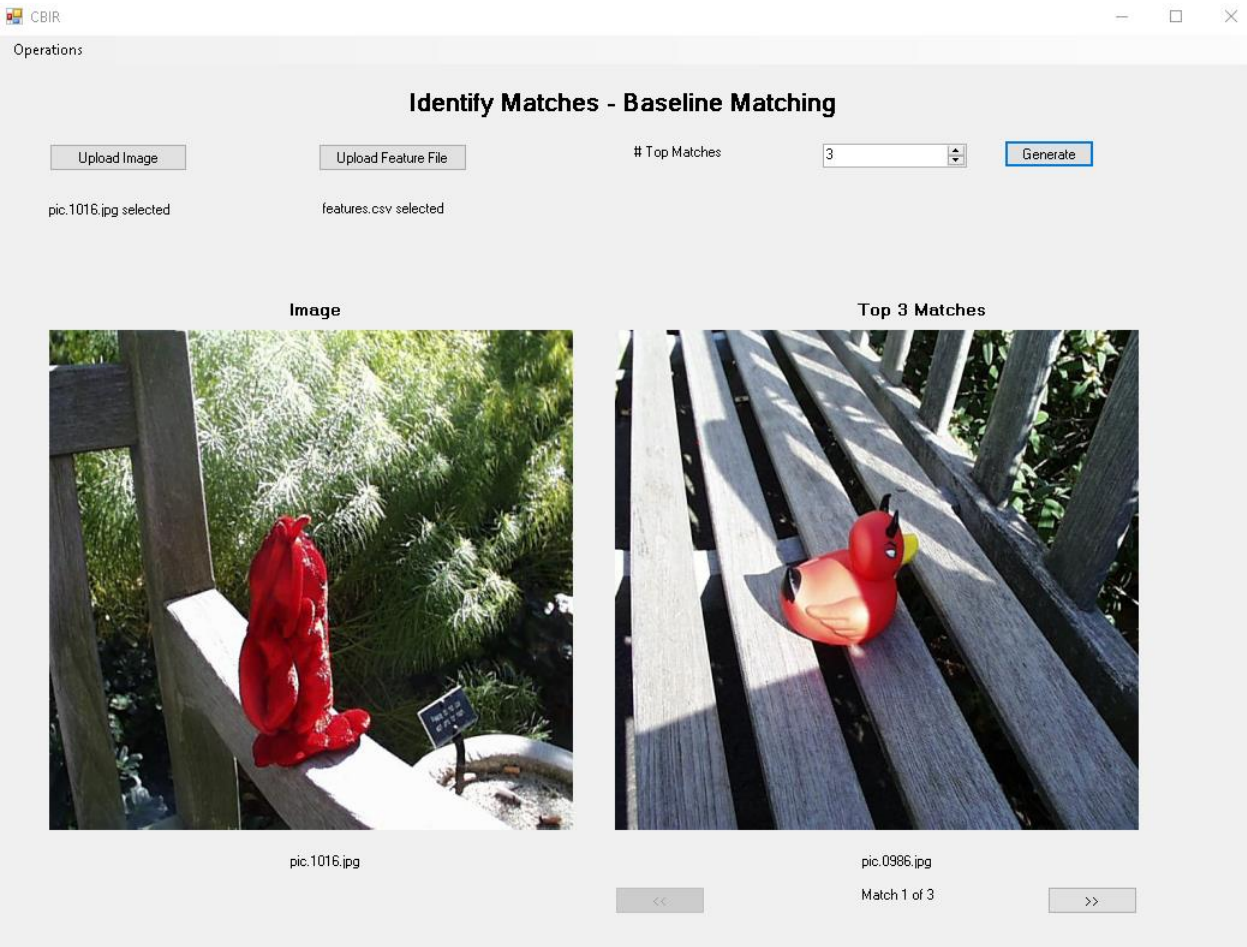

Figure 1: Baseline Matching Top 1 Match

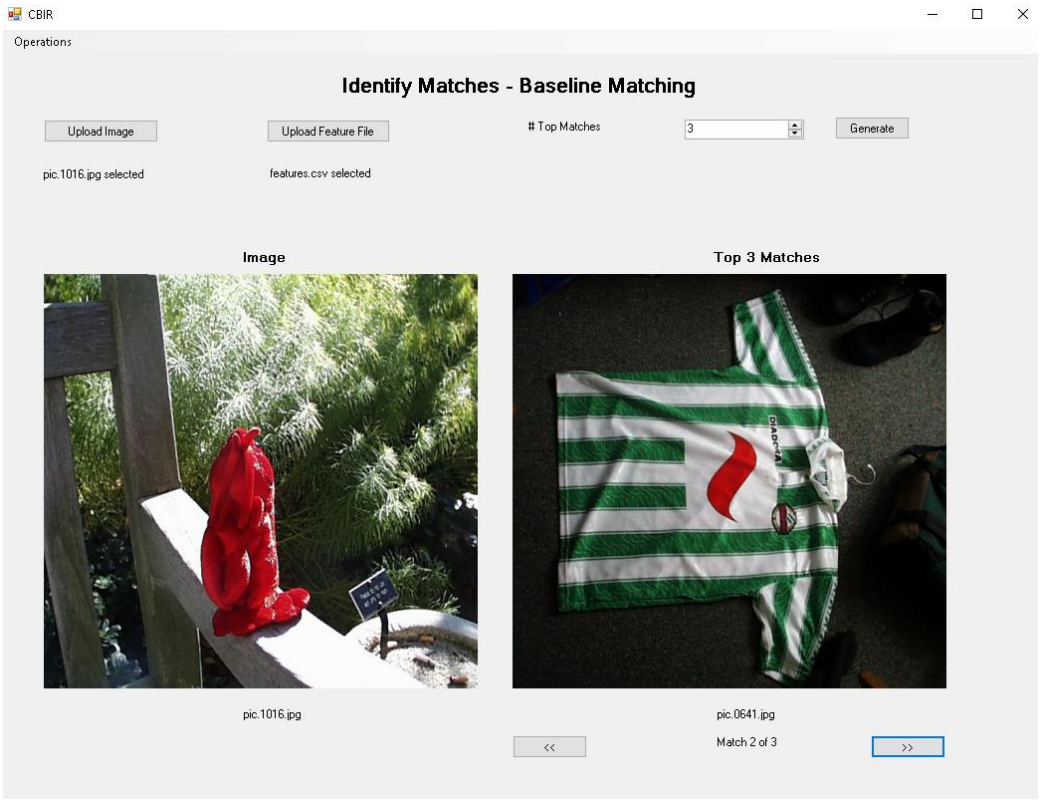

Figure 2: Baseline Matching Top 2 Match

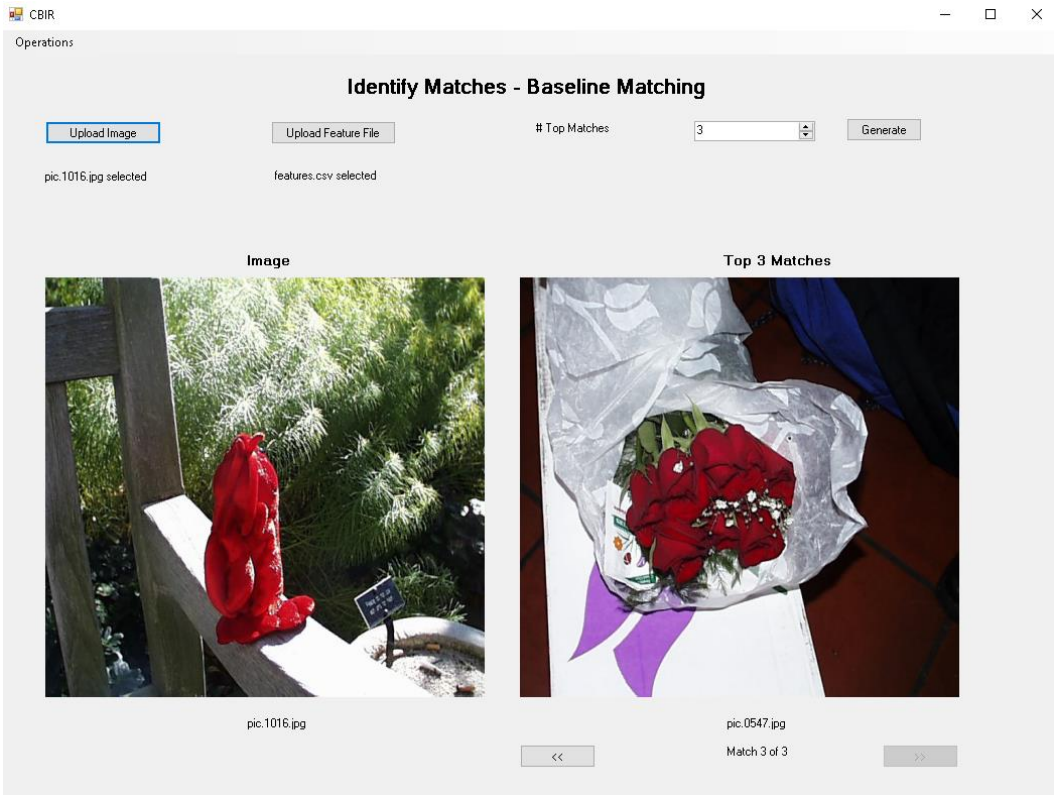

Figure 3: Baseline Matching Top 3 Match

3. Histogram Matching:

Histogram matching is an image retrieval technique that utilizes color histograms as feature vectors and histogram intersection as the distance metric. This method offers a simple yet effective way to compare images based on their color distributions. The choice of "bins per channel" parameter in histogram matching plays a crucial role in determining the granularity of the color histogram and thus affects the level of detail captured in the image representation.

Example:
To illustrate histogram matching, we have a target image(pic.0164.jpg) of a building with a vibrant blue sky in the background. The histogram matching algorithm outputs a ranked list of images from the database that closely resemble the target image of the building with the vibrant blue sky. The top matches represent architectural structures with similar color distributions, aiding in content-based image retrieval tasks focused on identifying scenes with vibrant blue skies and architectural elements.

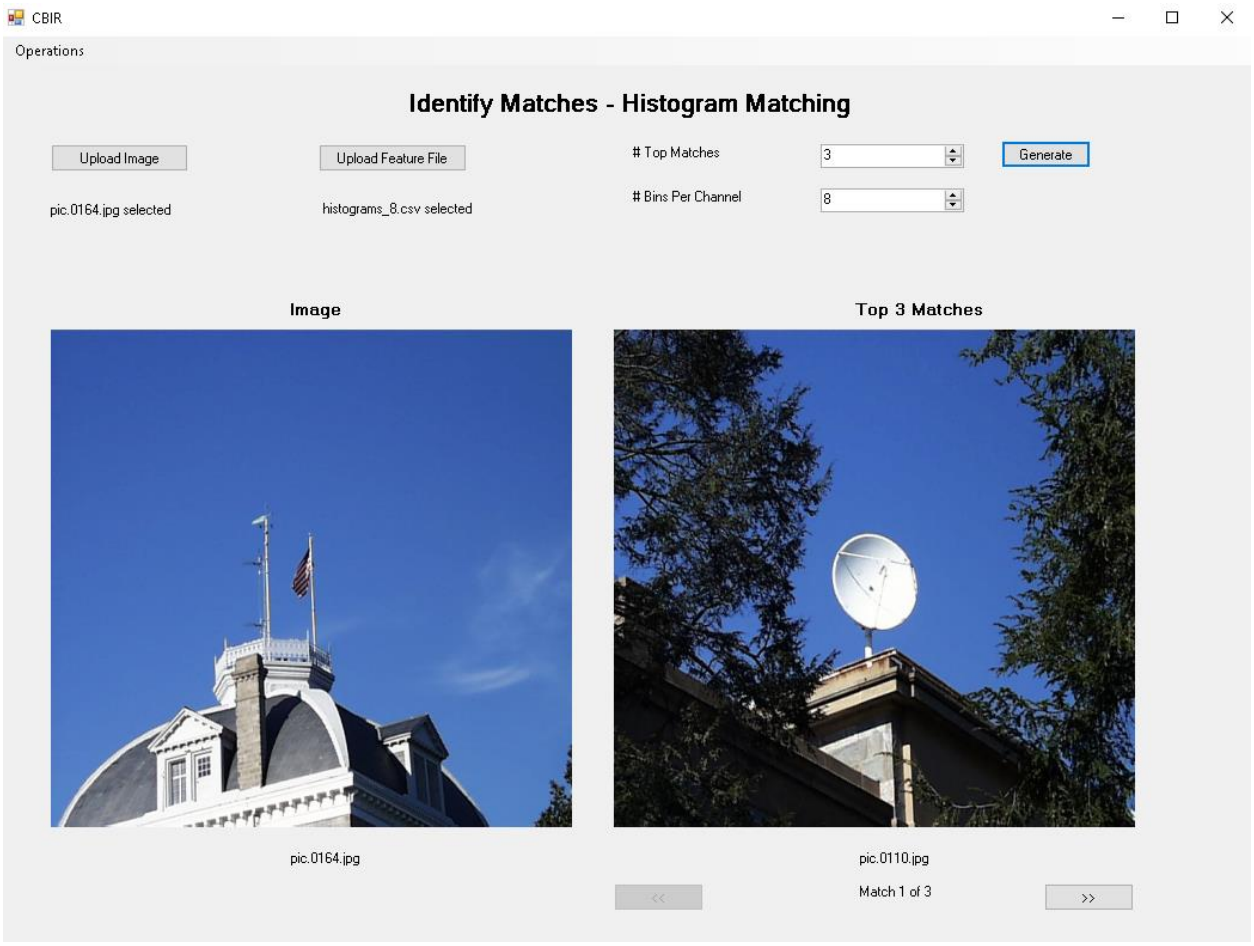

Figure 4: Histogram Matching Top 1 Match

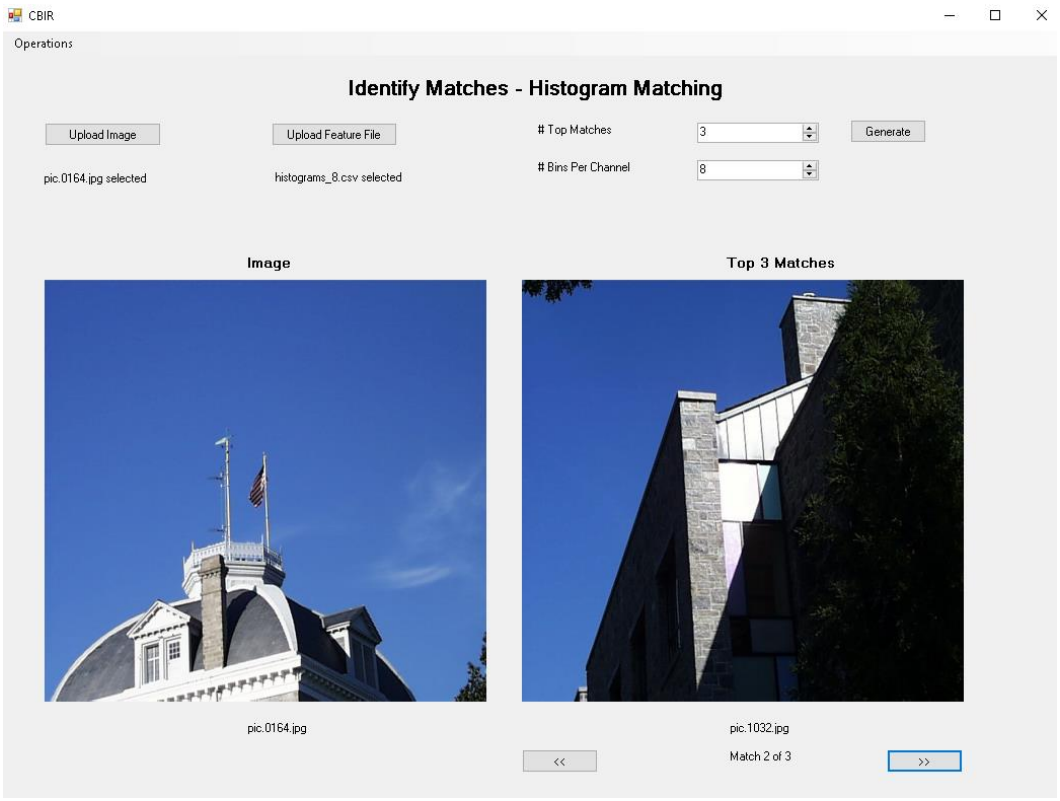

Figure 5: Histogram Matching Top 2 Match

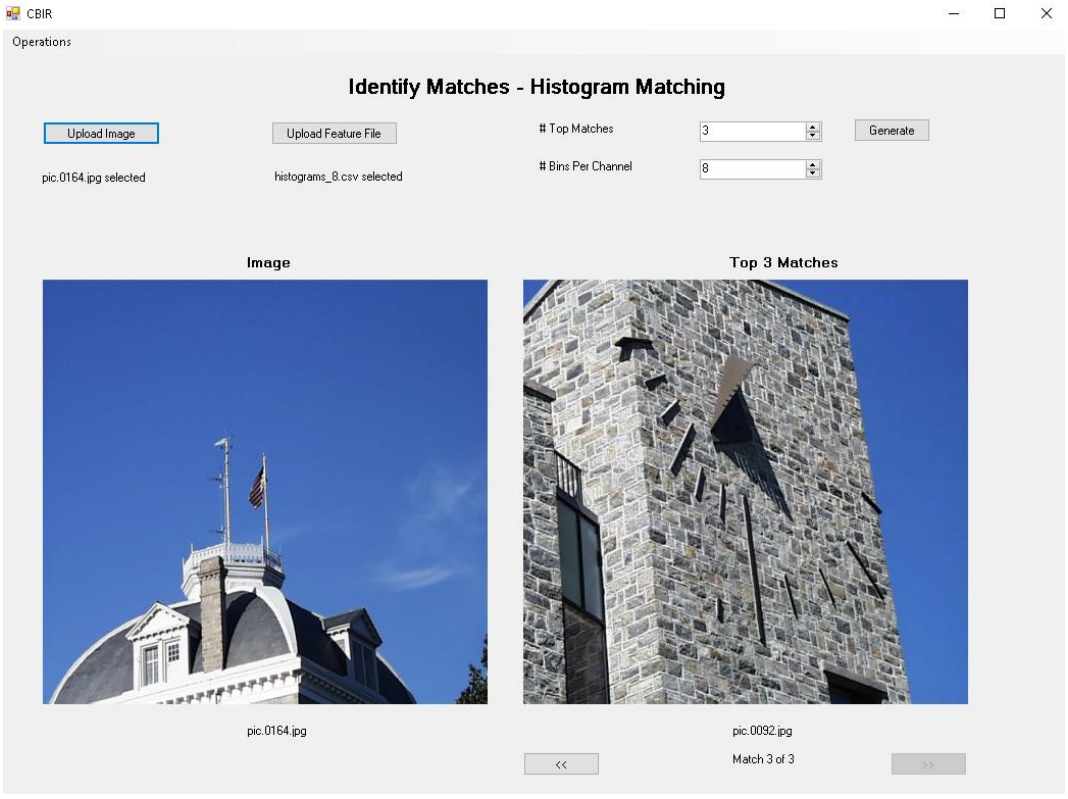

Figure 6: Histogram Matching Top 3 Match

4. Multi-Histogram Matching:

Multi-Histogram Matching is a technique used in Content-Based Image Retrieval (CBIR) systems to find similar images based on multiple histograms computed from different regions of the target image. It combines histogram intersection as the distance metric to compare histograms, providing a robust approach to image similarity assessment. The choice of "bins per channel" parameter in histogram matching plays a crucial role in determining the granularity of the color histogram and thus affects the level of detail captured in the image representation.

Example:
To illustrate multi-histogram matching, we have a target image(pic.0274.jpg) featuring a building and sky is analyzed by dividing it into two regions. Individual color histograms for each region are combined into a single feature vector. This vector is then compared to those of a database to find images with similar color profiles. The system ranks the images based on how closely their histograms match, identifying the top matches that share color characteristics with the target, like the hue of the sky or the building's facade.

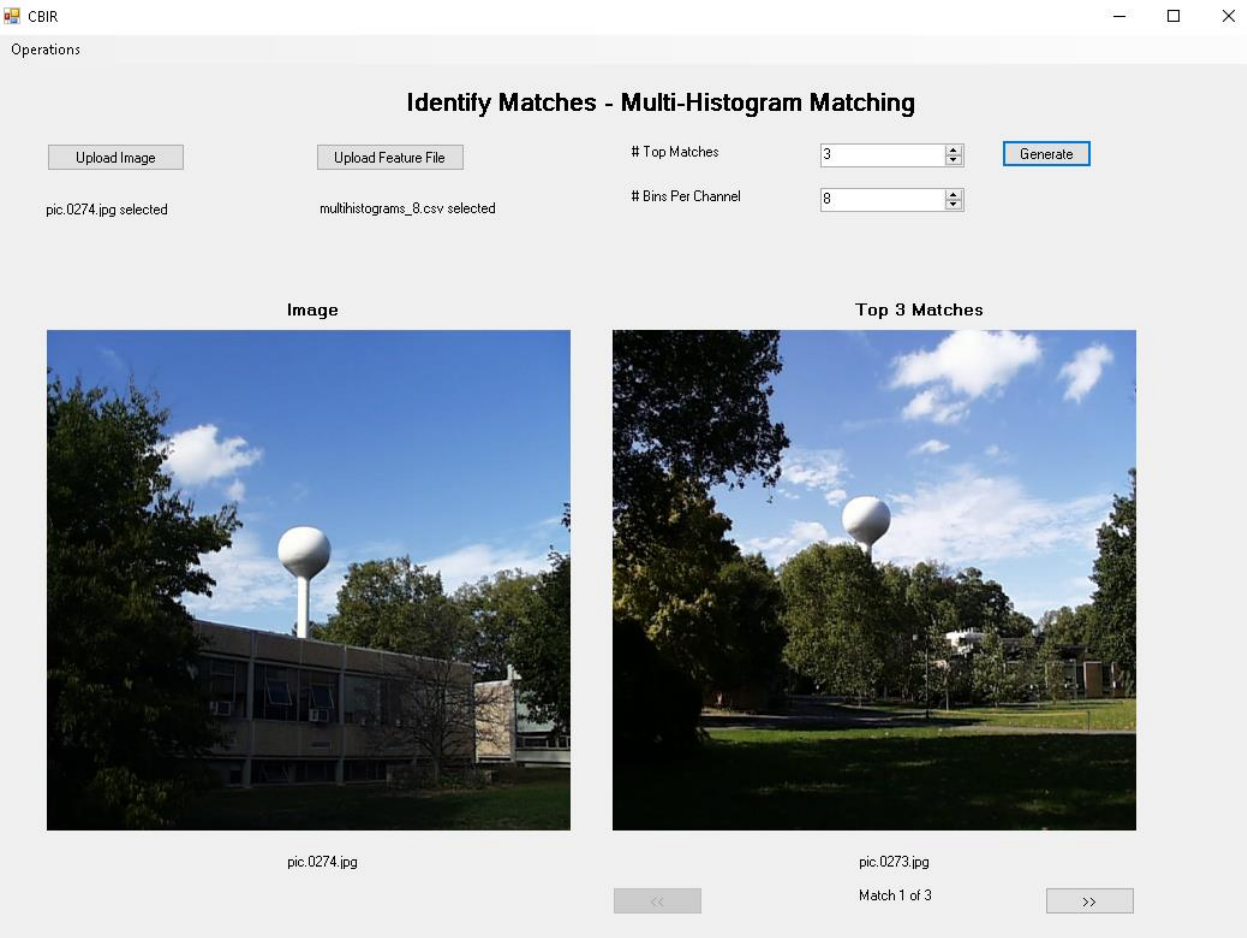

Figure 7: Multi-Histogram Matching Top 1 Match

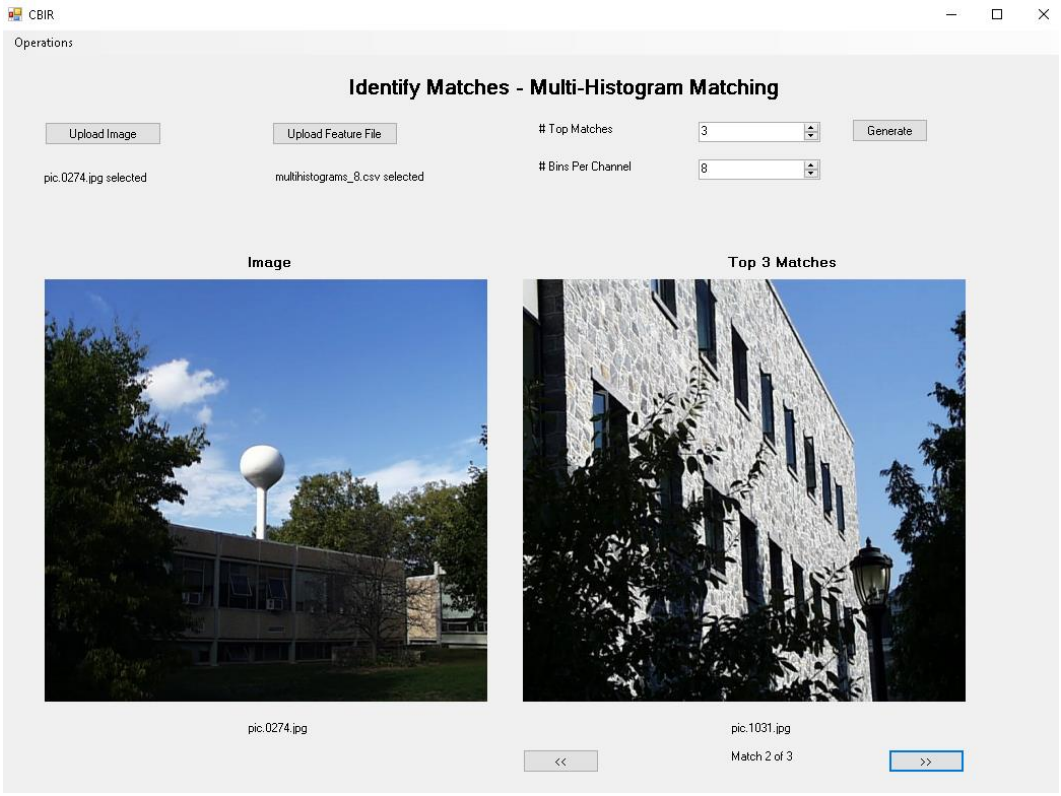

Figure 8: Multi-Histogram Matching Top 2 Match

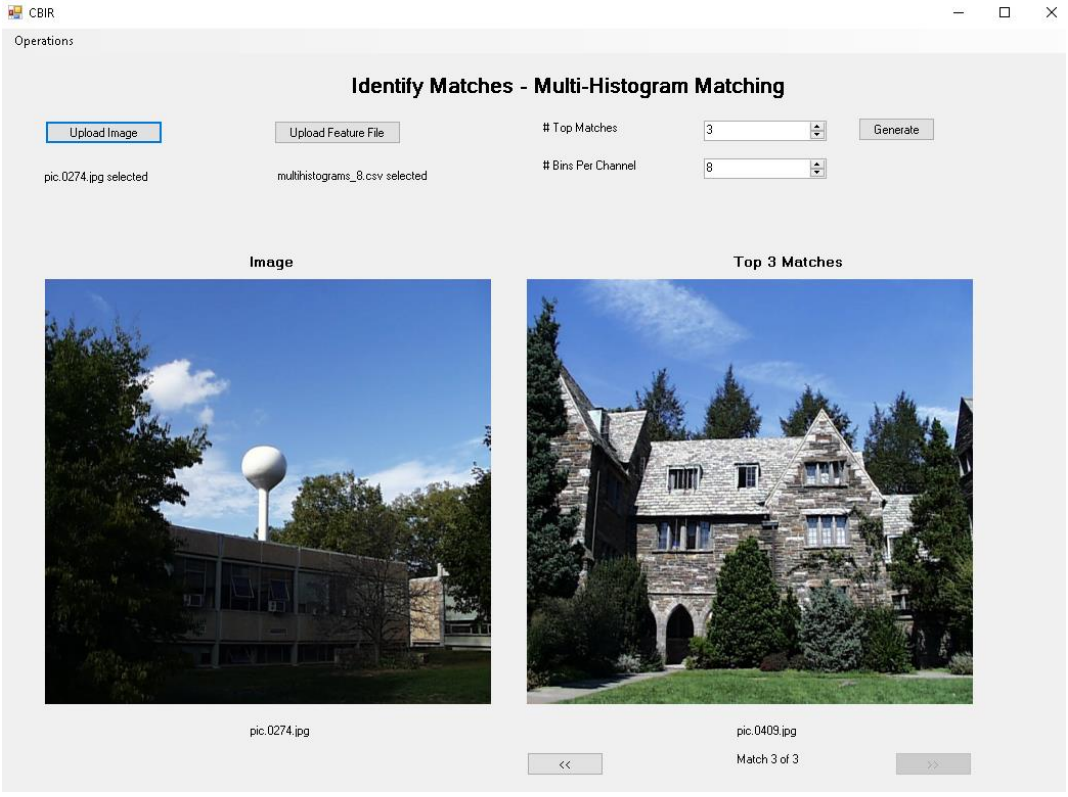

Figure 9: Multi-Histogram Matching Top 3 Match

5. Texture and Color:

Texture and Color Histogram Matching combines whole image color and texture histograms to find similar images. The color histogram captures color distributions, while the texture histogram, calculated using Sobel magnitude, describes texture patterns. The choice of "bins per channel" parameter in the color histogram determines the granularity of color representation, while the "texture bins" parameter controls the level of detail in the texture description.

Example:

To illustrate texture and color, we have a target image(pic.0535.jpg) featuring an indoor dining area with stone walls, wooden features, white chairs, and a white chimney. The stone wall, wooden features, and white furniture create a distinctive texture and color pattern. The CBIR system computes two types of histograms: a color histogram capturing the distribution of colors with a specified number of bins per channel, and a texture histogram capturing patterns and edges using gradient magnitudes with 16 texture bins per channel. Both histograms provide a granular representation of the image's visual features. These histograms are then combined into a single feature vector. The system compares the combined feature vector against the feature vectors of images stored in the database. Based on the computed distances, the system identifies and ranks the top matching images.

Comparison:

- **Histogram Matching:** This method considers only color histograms for matching, potentially overlooking textural nuances. Therefore, the top matches for pic.0535.jpg would be images with similar colors, even if their textures are dissimilar.
- **Multi-Histogram Matching:** By evaluating multiple color histograms from different regions of an image, this technique provides a spatially sensitive color comparison. However, like Histogram Matching, it does not incorporate texture information, which can lead to matches with correct color layouts but varied textures.
- **Texture and Color Matching:** This approach integrates both color and textural attributes, offering a holistic similarity assessment. Matches for pic.0535.jpg derived from this method will more accurately reflect both the color and textural content, ensuring that images with stone textures or wooden elements, as present in the target, are ranked higher.

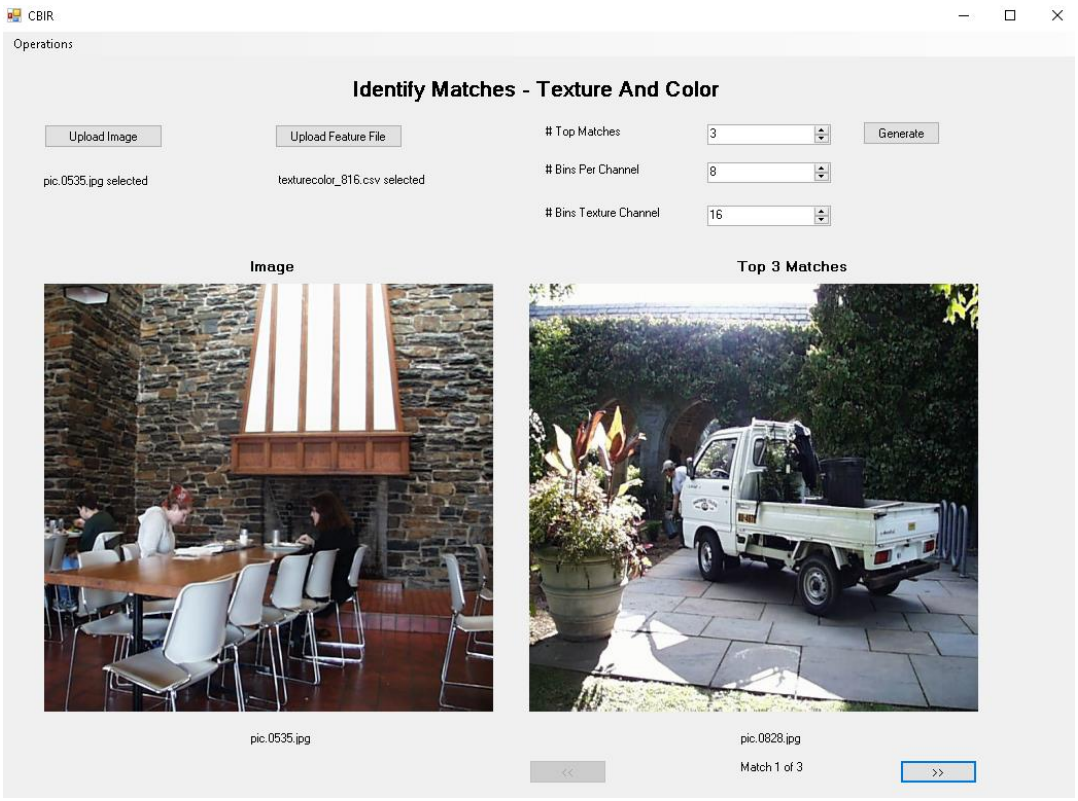

Figure 10: Texture and Color Matching Top 1 Match

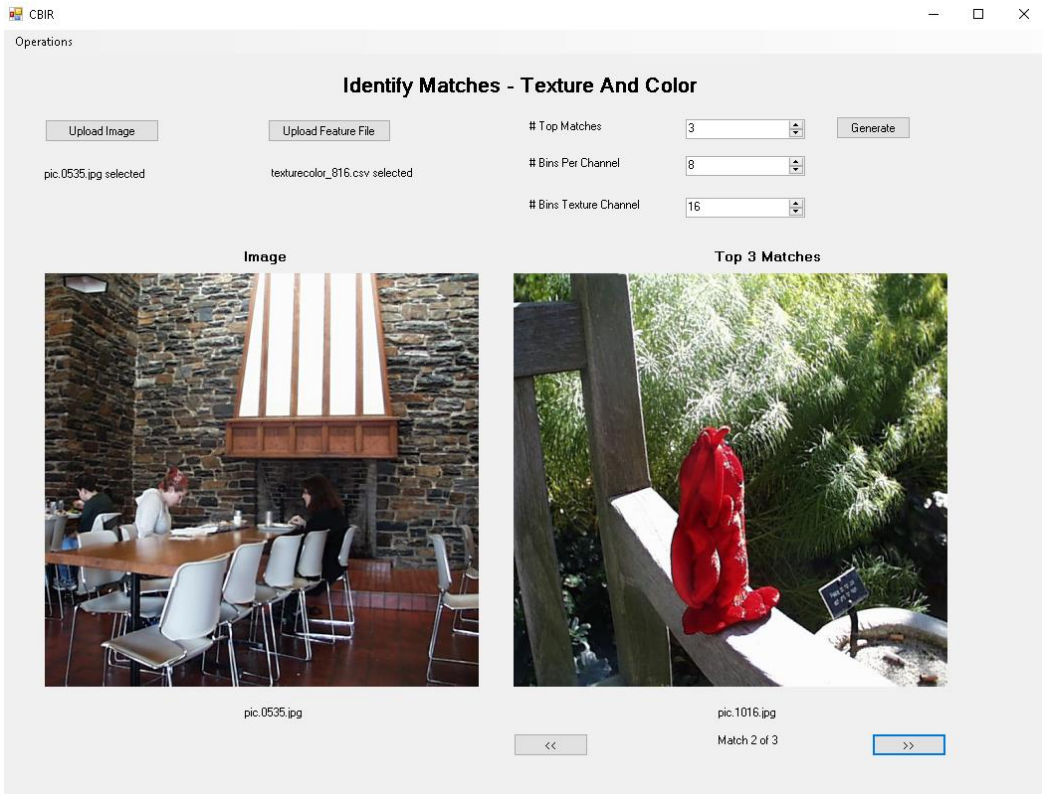

Figure 11: Texture and Color Matching Top 2 Match

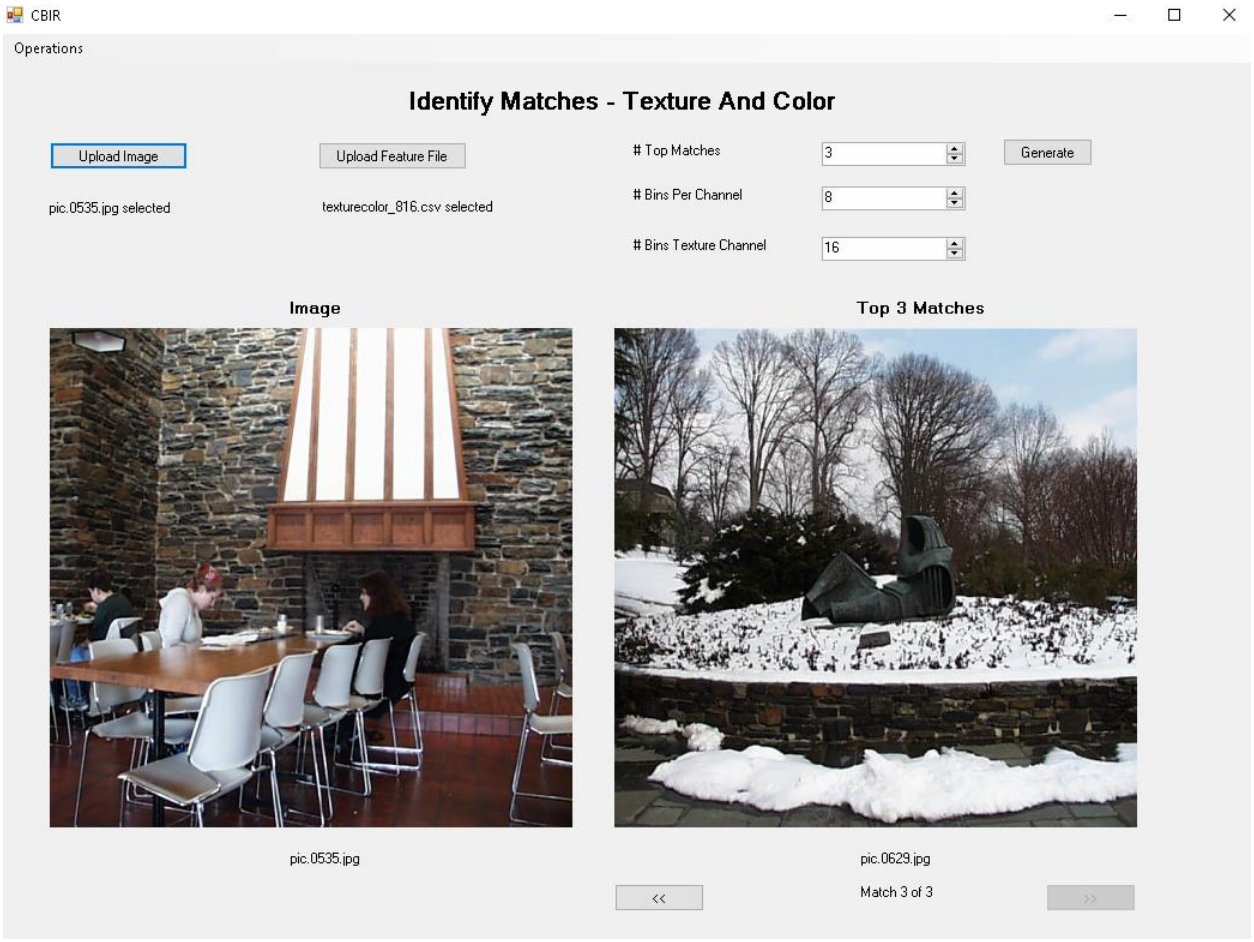

Figure 12: Texture and Color Matching Top 3 Match

6. Deep Network Embeddings:

Deep Network Embeddings Matching utilizes feature vectors extracted from a deep neural network to find similar images. These feature vectors capture high-level semantic information about the content of images and are obtained from the final average pooling layer of a pre-trained ResNet18 deep network trained on ImageNet.

Example:

For instance, we have a target image(pic.0893.jpg) of a water hydrant positioned on lush grass, the embeddings capture its shape, color, and the context provided by its environment. The similarity between the feature vector of this target image and those of images within a database is computed using cosine distance, ensuring that the images with the closest embeddings are deemed most similar.

Comparison:

Contrasting this approach with Histogram Matching, Multi-Histogram Matching, and Texture and Color Matching—which are predicated on manually engineered features such as color distributions and texture gradients—Deep Network Embeddings leverage the inherent ability of neural networks to discern complex patterns and high-level concepts in the visual data. This can result in a more nuanced and semantically rich set of comparable features.

When comparing the top three results for images pic.0893.jpg and pic.0164.jpg:

- **Histogram Matching** might prioritize images with similar color profiles, potentially overlooking the context or comparison of elements within the images.
- **Multi-Histogram Matching** extends this by considering color profiles in different image regions but may still miss out on the semantic relationships between these regions.
- **Texture and Color Matching** considers both color and texture, offering a balanced representation but still within the realm of handcrafted features.
- **Deep Network Embeddings** move beyond these limitations by capturing more abstract features that represent deeper semantic relationships, likely resulting in matches that are visually and contextually more aligned with the target images.

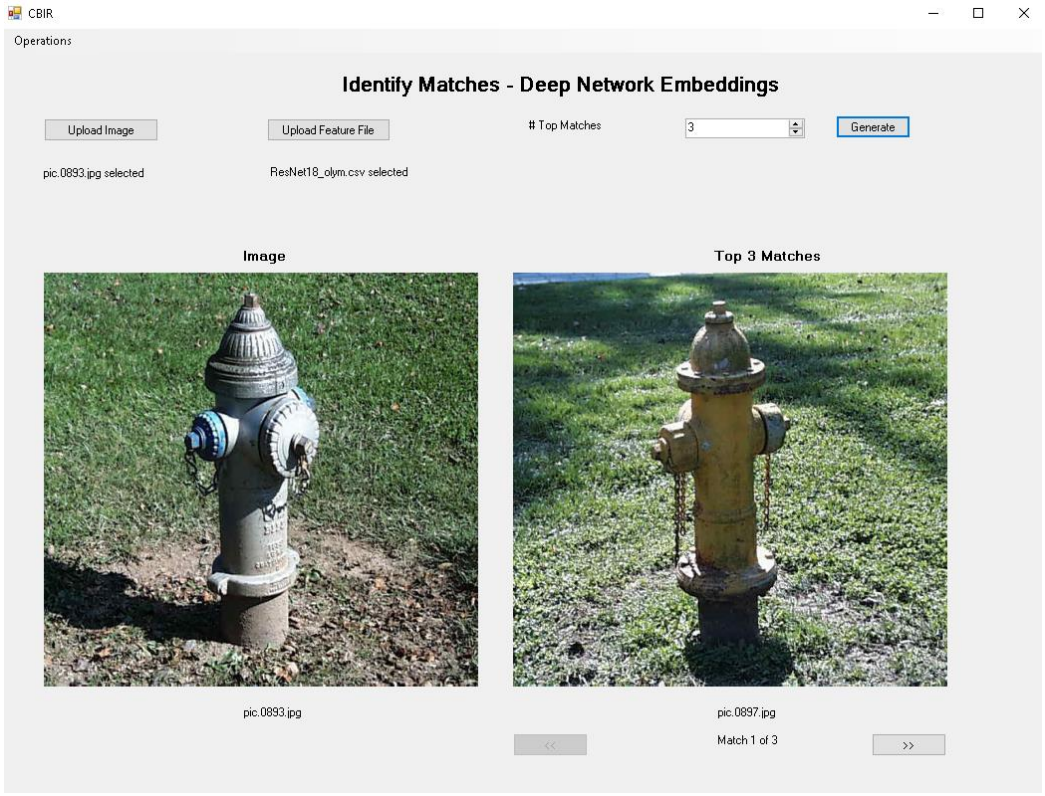

Figure 13: DNN Matching for pic.0893.jpg Top 1 Match

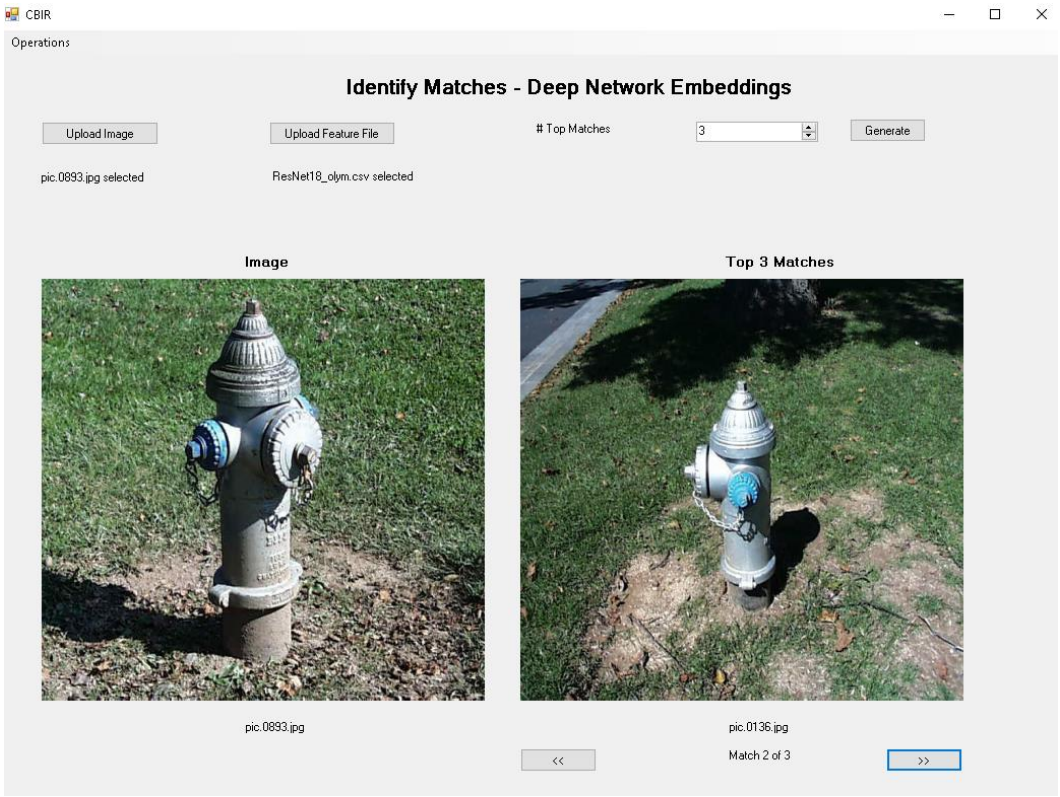

Figure 14: DNN Matching for pic.0893.jpg Top 2 Match

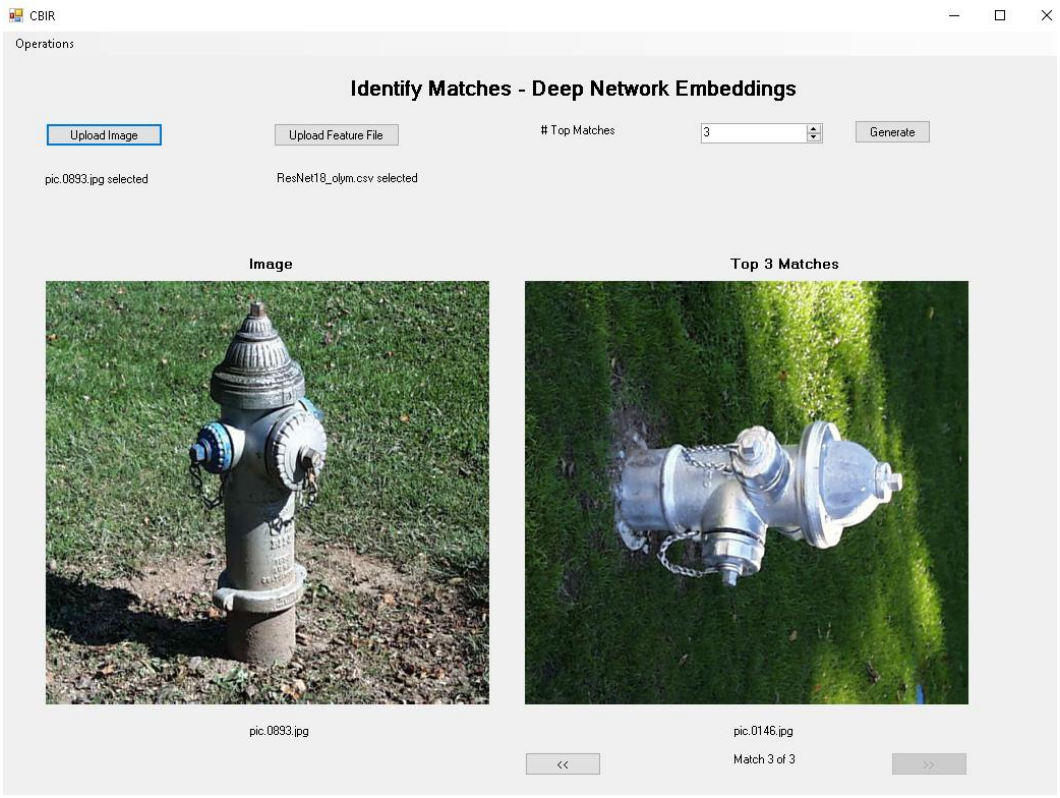

Figure 15: DNN Matching for pic.0893.jpg Top 3 Match

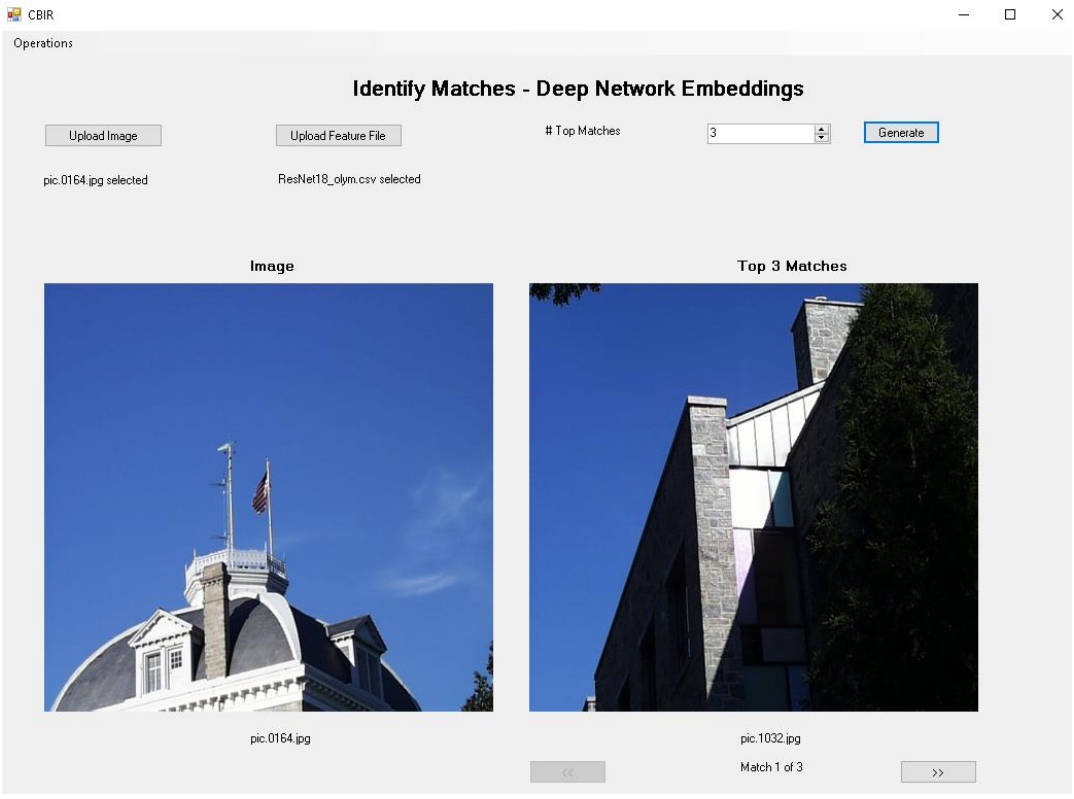

Figure 16: DNN Matching for pic.0164.jpg Top 1 Match

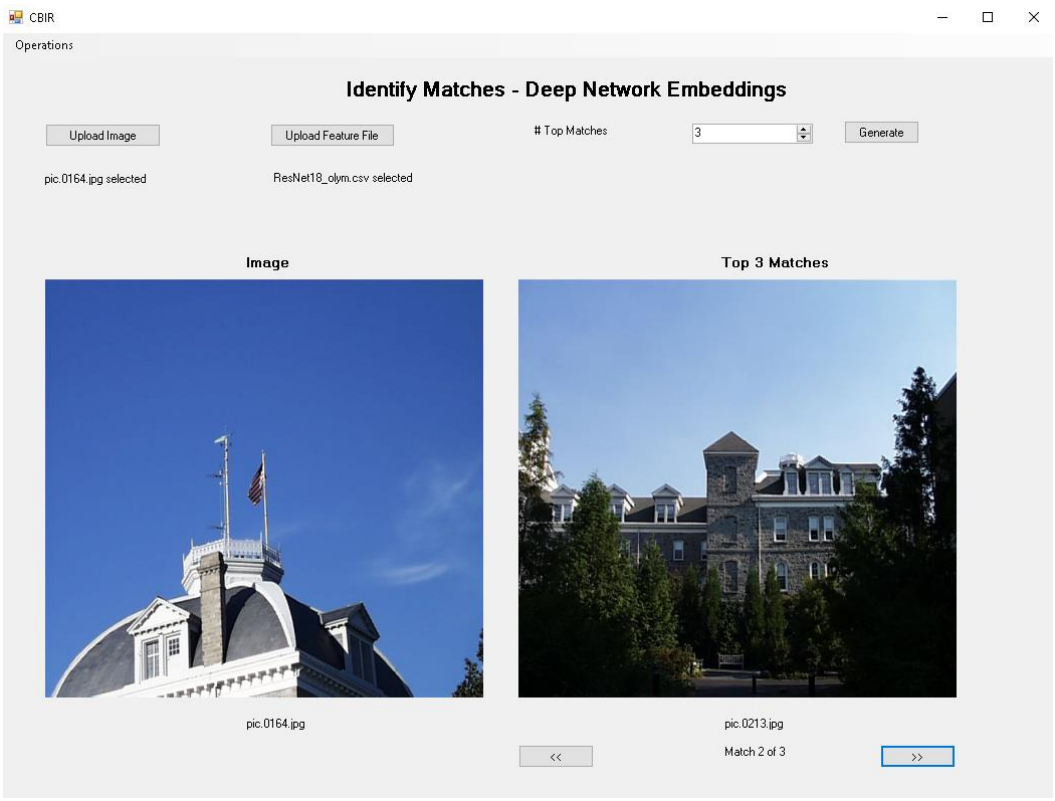

Figure 17: DNN Matching for pic.0164.jpg Top 2 Match

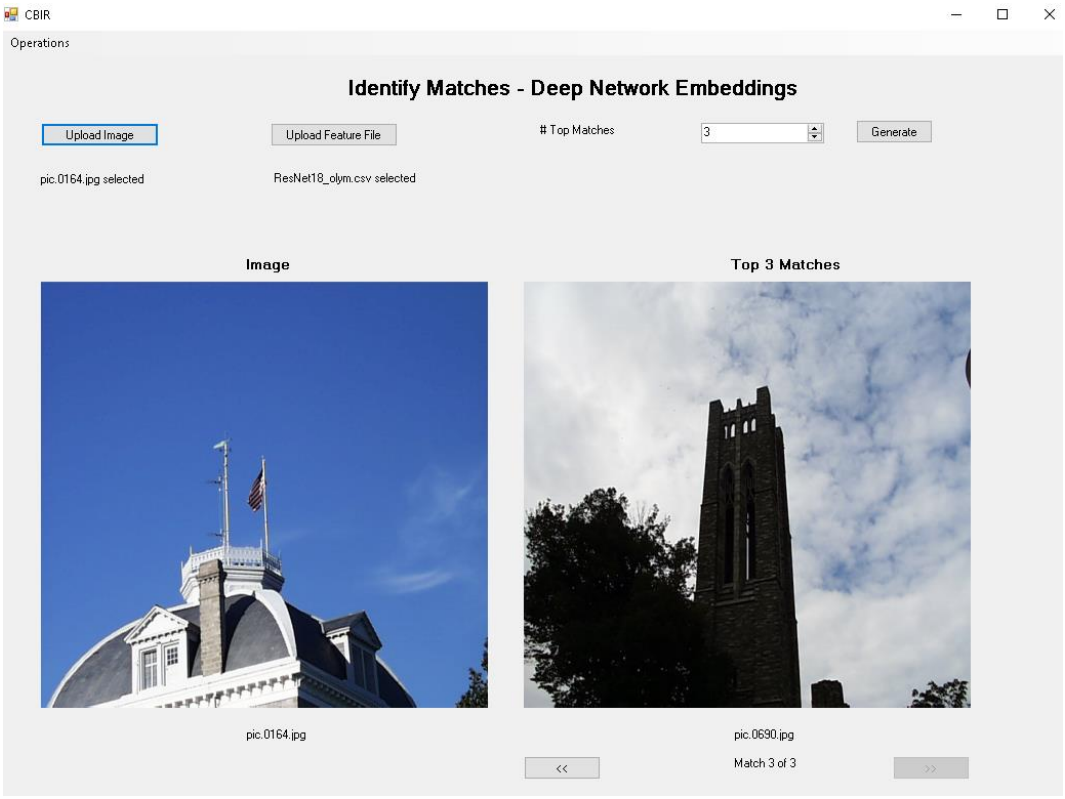

Figure 18: DNN Matching for pic.0164.jpg Top 3 Match

7. Compare DNN Embeddings and Classic Features:

When comparing Deep Neural Network (DNN) embeddings with classic features for content-based image retrieval, there are several factors to consider. The performance of these methods can vary depending on the specific characteristics of the images and the similarity measures used.

For image pic.1072.jpg:

- **Histogram Matching** with pic.0937.jpg as a top match suggests that color distribution is a significant factor in determining similarity.
- **Multi-Histogram Matching** leading to pic.0813.jpg implies that when considering multiple regions, the similarity in color distribution across these specific regions is what defines the match.
- **Texture and Color matching** selecting pic.0701.jpg indicates that the combination of color and texture features in this image resonates more closely with the features of the selected match.
- **DNN Embeddings** pointing to pic.0143.jpg suggests that the high-level abstractions captured by the DNN are different from those captured by the other methods, likely reflecting more complex patterns or semantic content.

For image pic.0734.jpg:

- **Histogram Matching** yields pic.0450.jpg, emphasizing color similarity.
- **Multi-Histogram Matching** with pic.0577.jpg as the match suggests a similarity in the color layout of different regions within the images.
- **Texture and Color matching** points to pic.0105.jpg, indicating a blend of color and texture that is distinctive to these images.
- **DNN Embeddings** suggesting pic.0731.jpg as the closest match implies that the DNN has identified deeper patterns or content features that may not be immediately apparent through basic color or texture analysis.

From these observations, it is evident that DNN embeddings do not always align with the results from classic feature methods. This divergence can be due to DNNs capturing high-level content-based features that may relate to object identity, scene type, or other complex visual constructs beyond basic color and texture. This can sometimes result in more semantically meaningful matches but may also lead to less expected results if the user's notion of similarity is more aligned with color or texture.

In some cases, especially where the image content is less about the objects or semantic content and more about the color and texture, classic feature methods might provide more directly relevant results. For example, if the task is to match images based on a specific color scheme or texture pattern without regard for the objects present, classic features may be more appropriate.

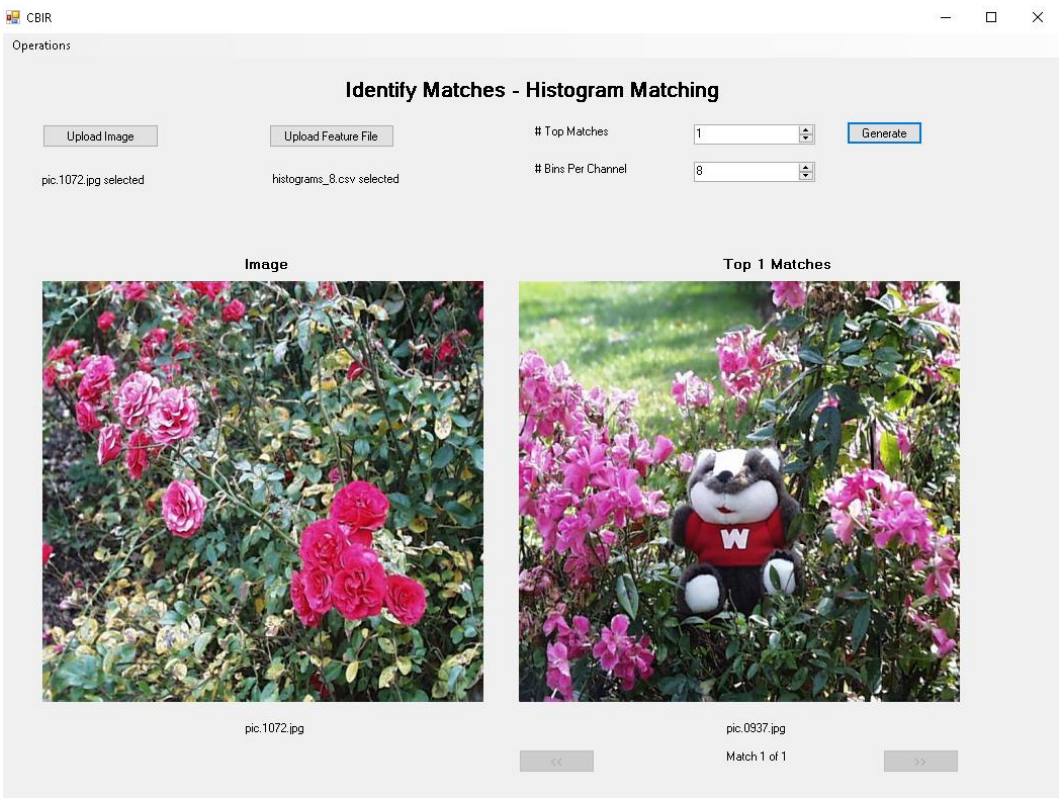

Figure 19: Histogram Matching for pic.1072.jpg

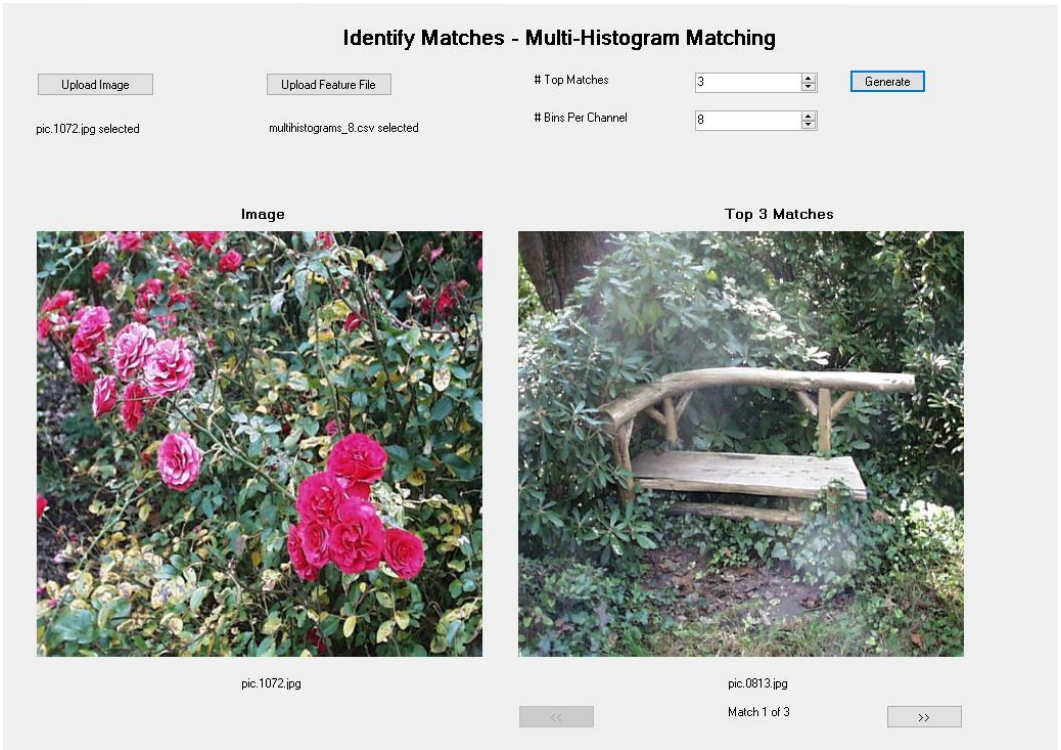

Figure 20: Multi-Histogram Matching for pic.1072.jpg

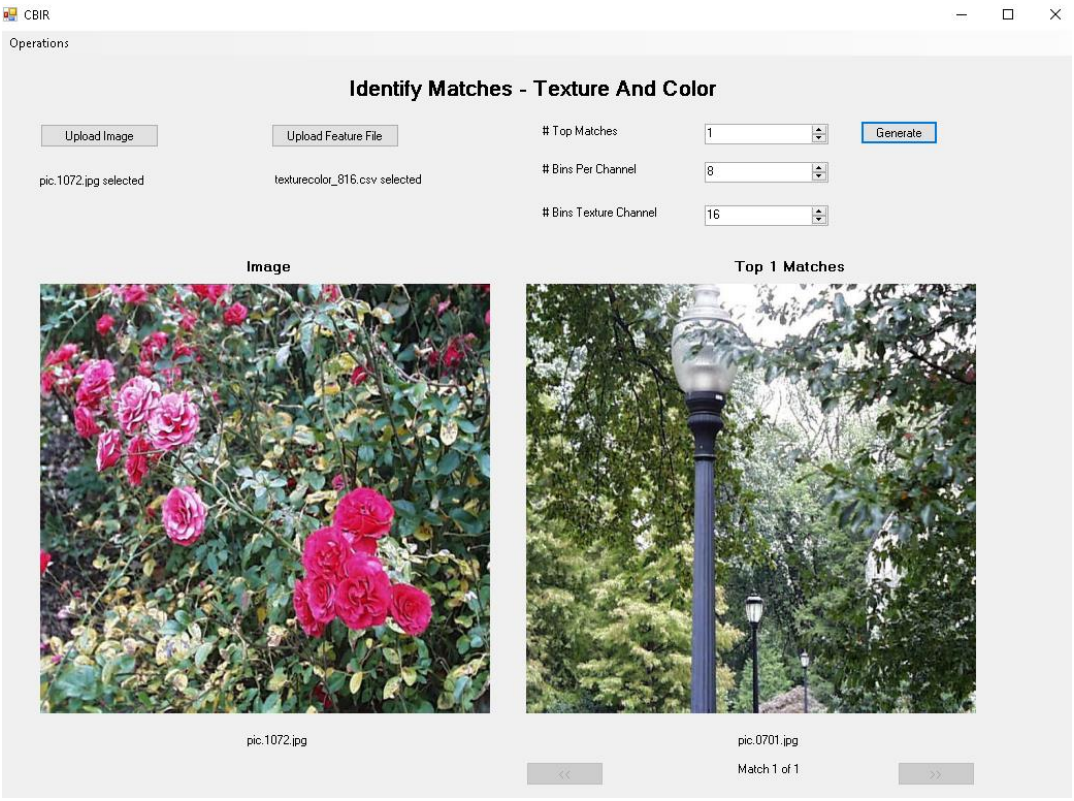

Figure 21: Texture and Color Matching for pic.1072.jpg

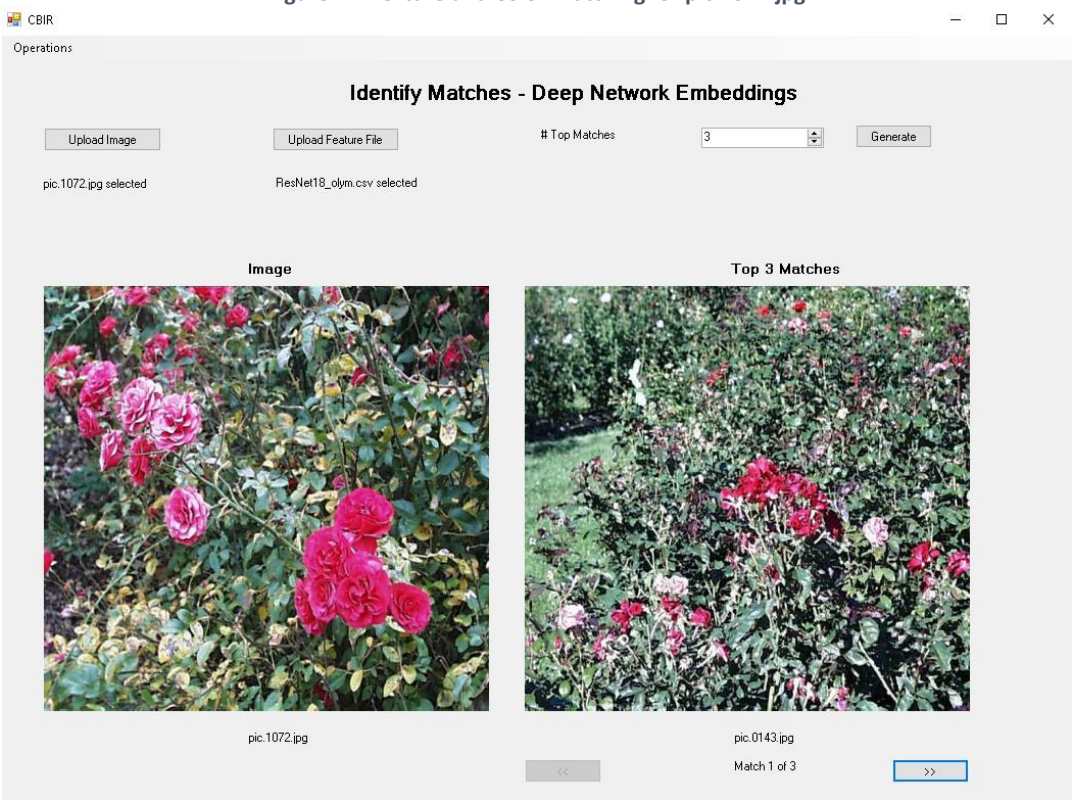

Figure 22: DNN Matching for pic.1072.jpg

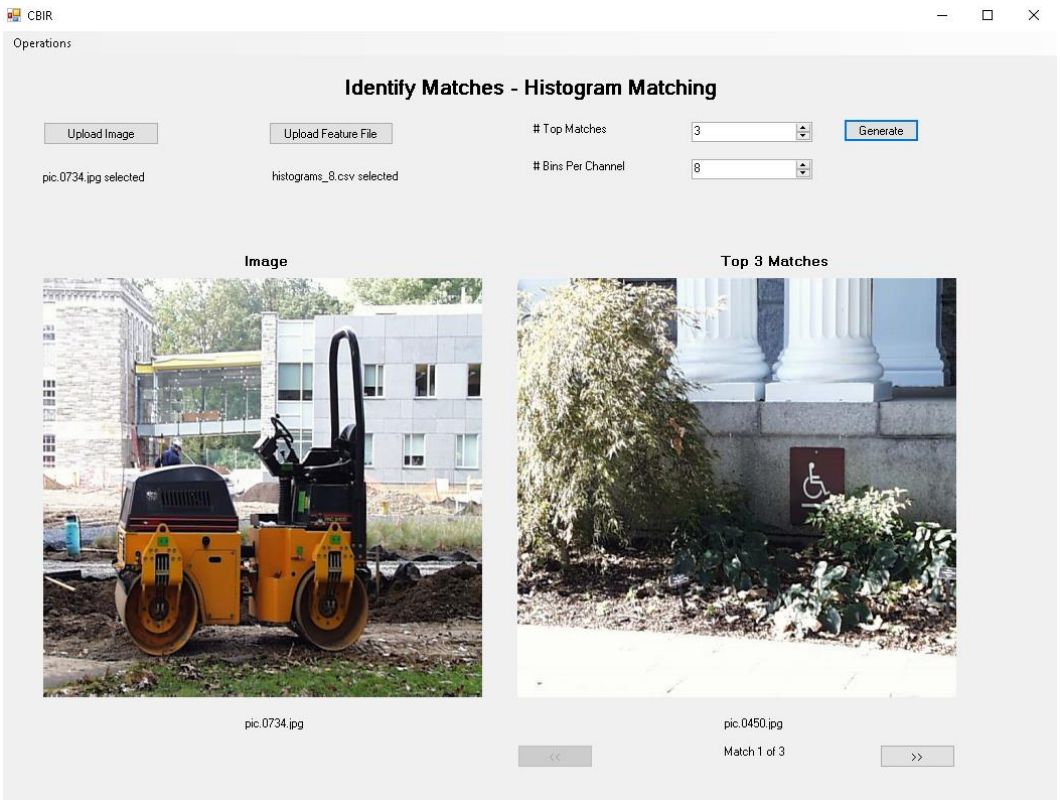

Figure 23: Histogram Matching for pic.0734.jpg

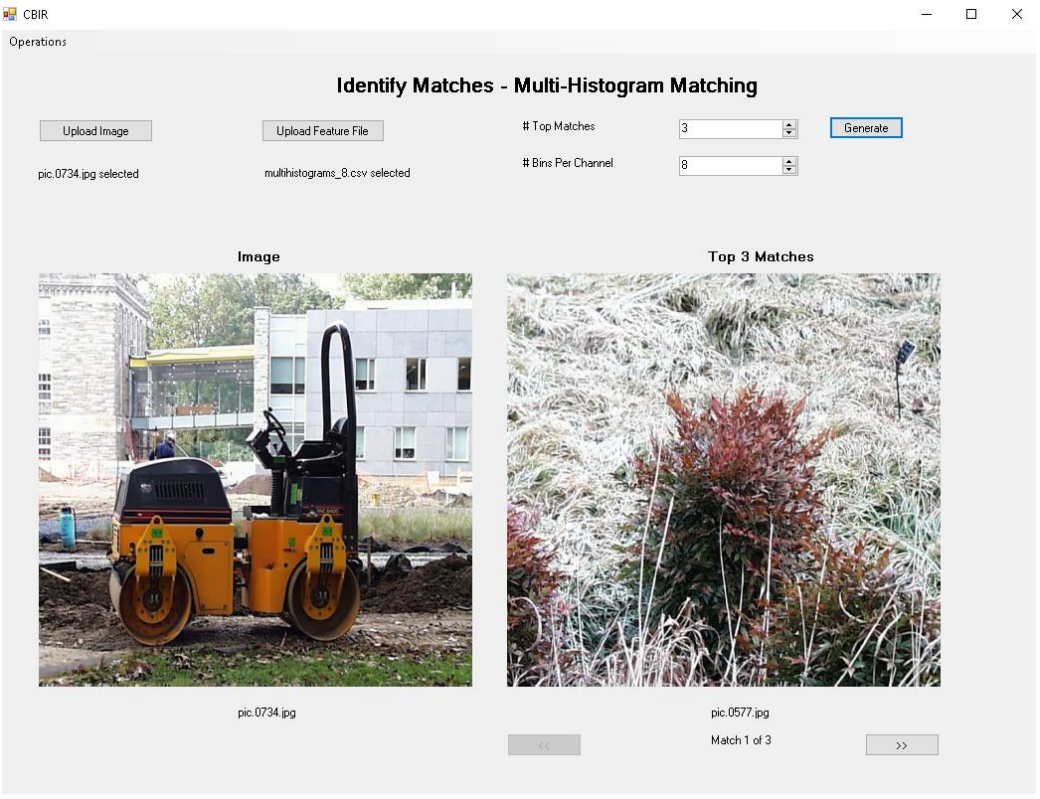

Figure 24: Multi-Histogram Matching for pic.0734.jpg

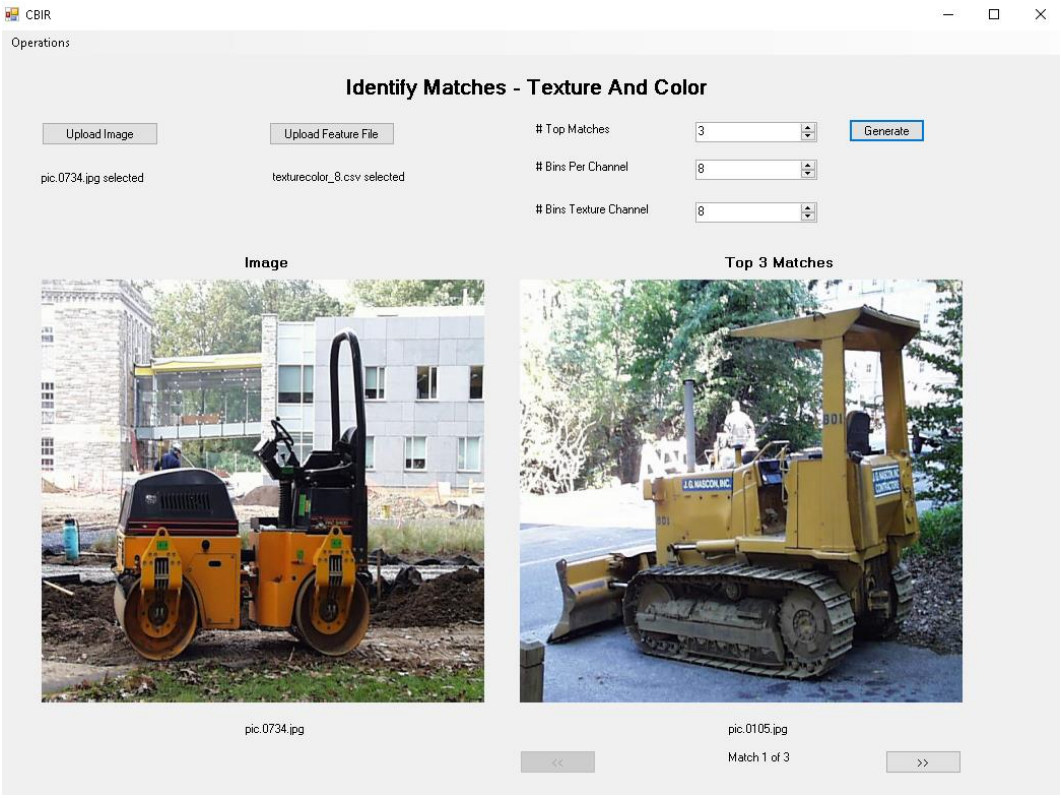

Figure 25: Texture and Color Matching for pic.0734.jpg

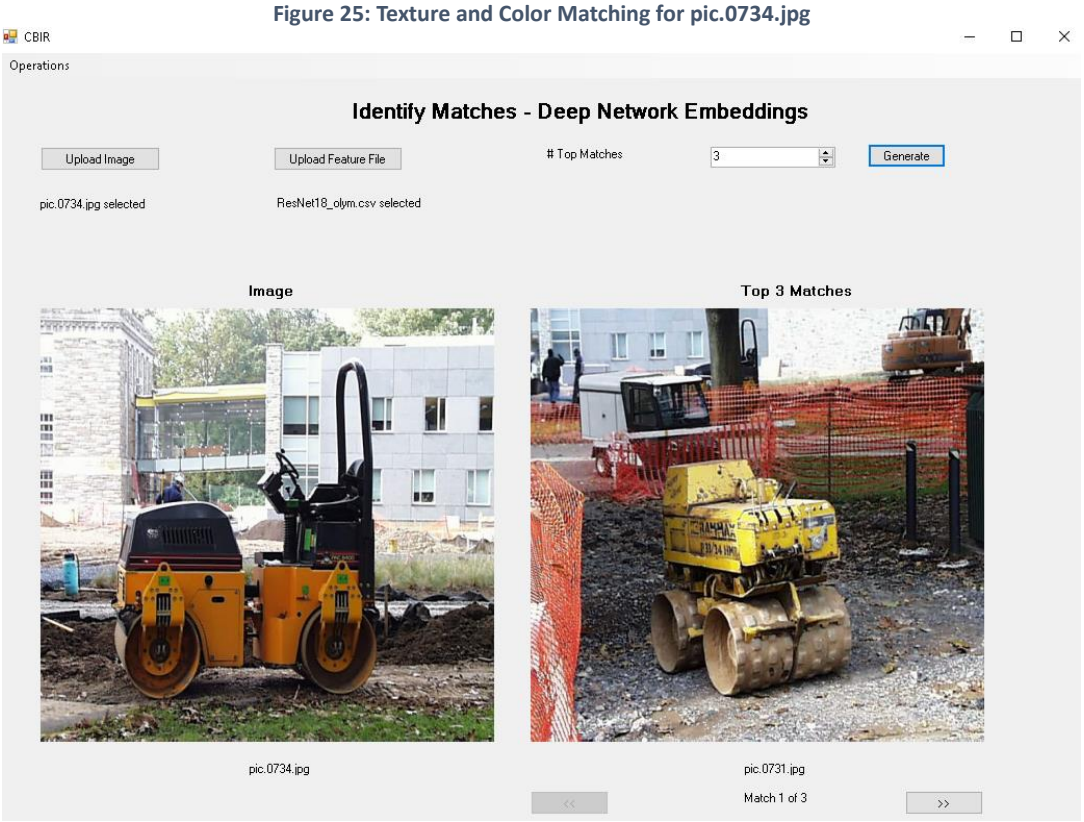

Figure 26: DNN Matching for pic.0734.jpg

8. Custom Design:

The custom designed CBIR system demonstrated the ability to effectively retrieve images with similar visual and semantic content to the target image. By integrating both DNN embeddings and handcrafted features, the system could capture a detailed representation of the images, leading to satisfactory retrieval results that aligned with the designed emphasis on sunset images.

Target Image pic.002.jpg:

- Top Matches: pic.005.jpg, pic.003.jpg, pic.012.jpg, pic.006.jpg, pic.007.jpg
- Observation: These images were likely identified as top matches due to their strong sunset color characteristics and potential horizon line features, matching the emphasis of the custom-designed feature vector.

Target Image pic.007.jpg:

- Top Matches: pic.003.jpg, pic.018.jpg, pic.005.jpg, pic.014.jpg, pic.004.jpg
- Observation: Like pic.002.jpg, the top matches for pic.007.jpg would have been selected based on their sunset color profiles and edge features aligning with the horizon detection part of the feature vector.

Target Image pic.011.jpg (non-Sunset Image):

- Top Matches: pic.020.jpg, pic.013.jpg, pic.015.jpg, pic.004.jpg, pic.018.jpg
- Observation: In a distinctive evaluation, when presented with pic.011.jpg—an image featuring a collection of vegetables—the system's retrieval significantly deviated from the sunset theme. The top matches included a mix of images without the sunset color prominence or horizon lines, indicating the feature vector's ability to differentiate and rank images based on their dissimilarity to the target image's theme. This test case highlights the system's robustness and its discriminative efficacy, ensuring a diverse and accurate retrieval across various image queries.

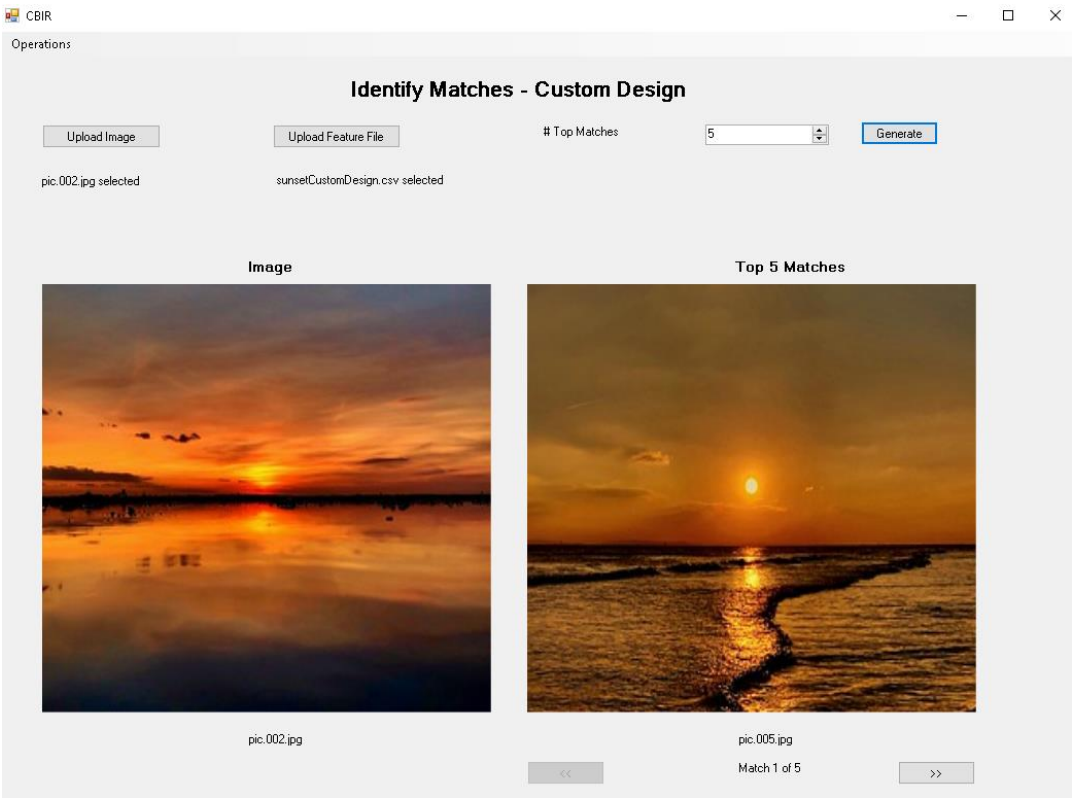

Figure 27: Custom Design Matching for pic.002.jpg displaying Top 1 Match

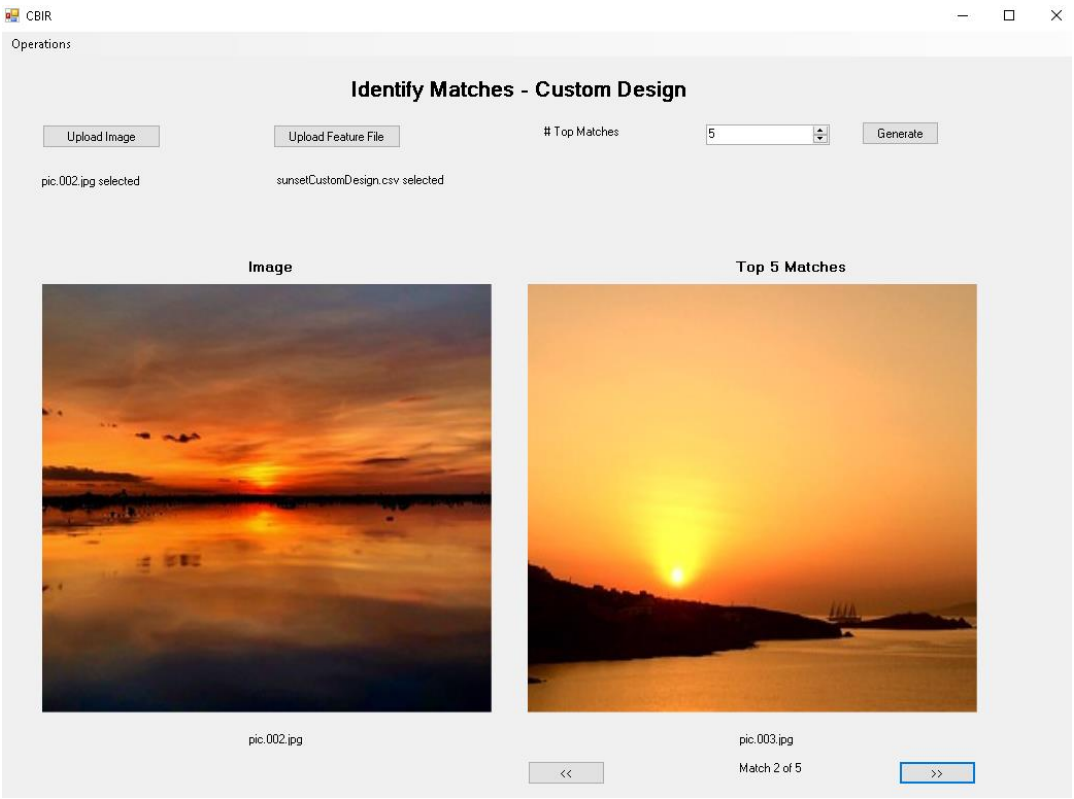

Figure 28: Custom Design Matching for pic.002.jpg displaying Top 2 Match

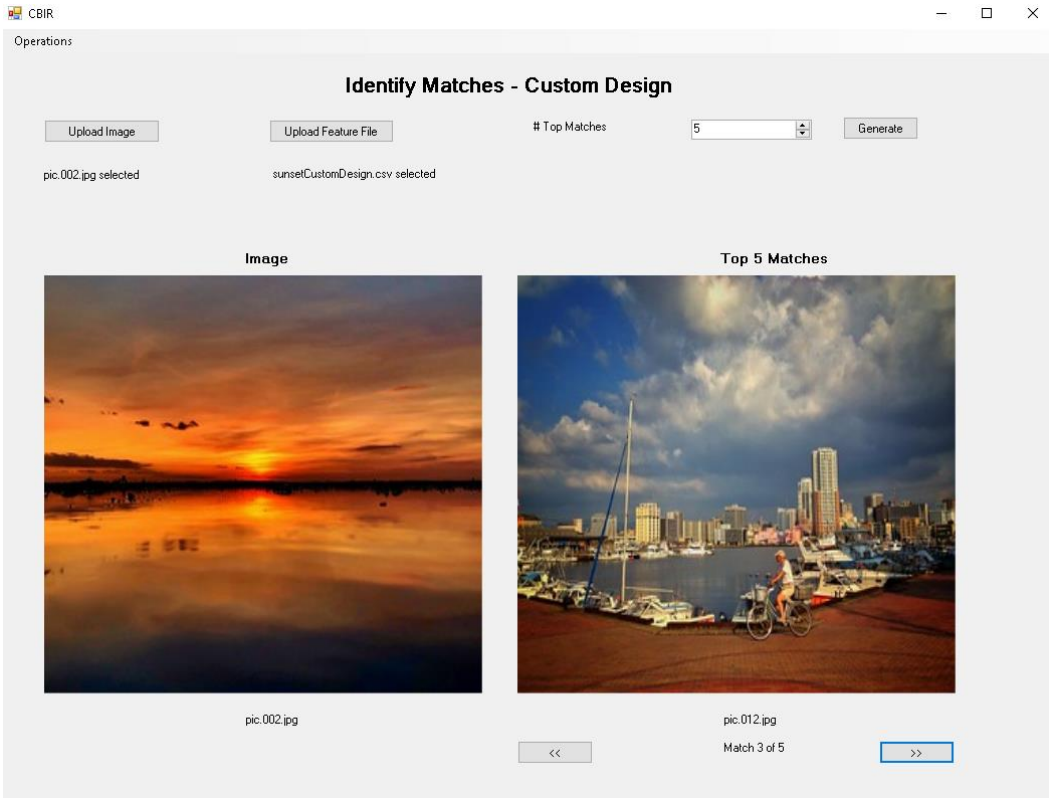

Figure 29: Custom Design Matching for pic.002.jpg displaying Top 3 Match

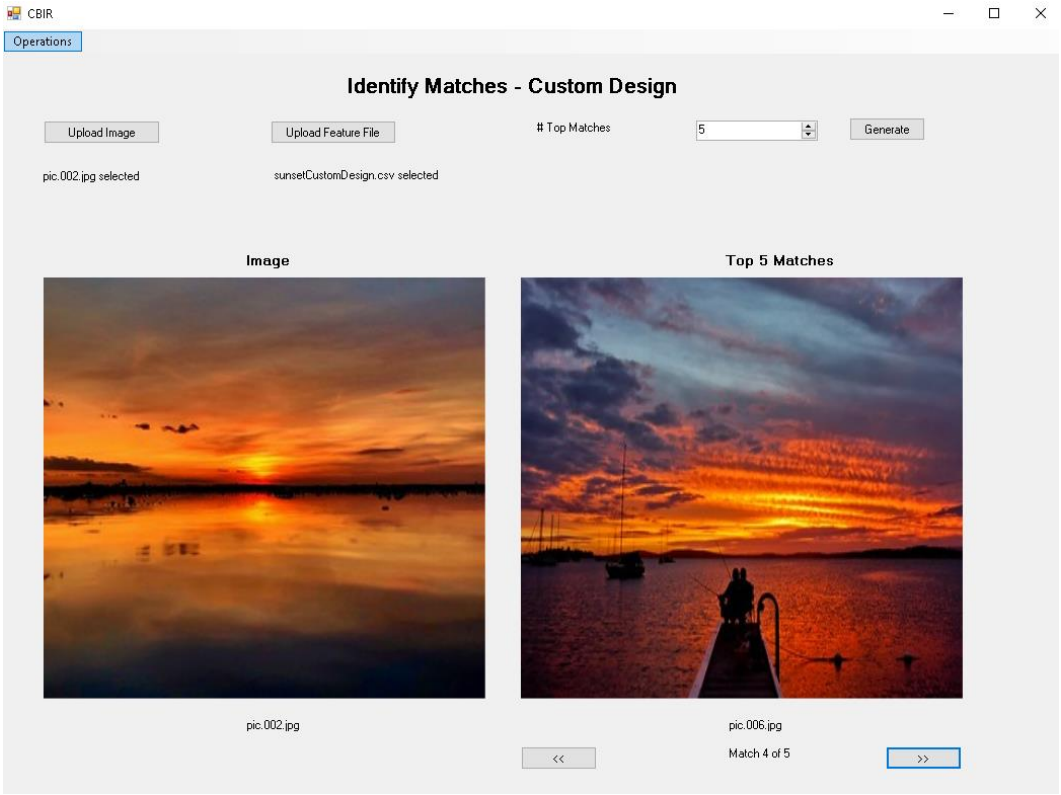

Figure 30: Custom Design Matching for pic.002.jpg displaying Top 4 Match

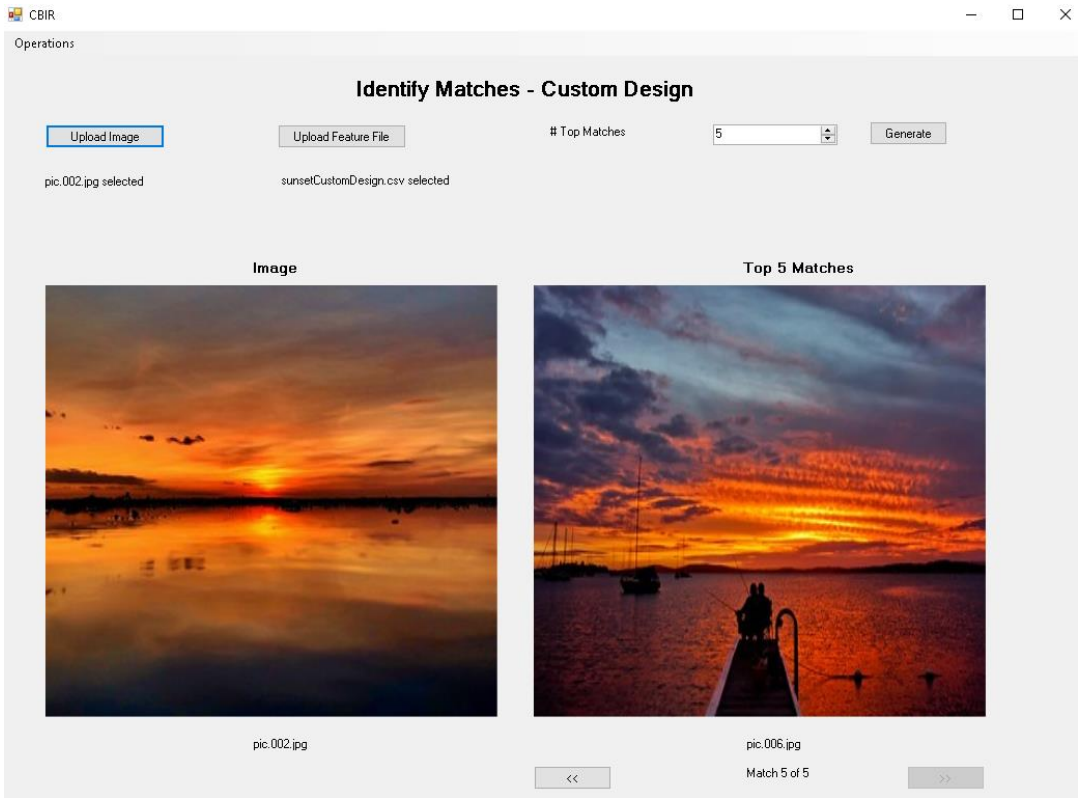

Figure 31: Custom Design Matching for pic.002.jpg displaying Top 5 Match

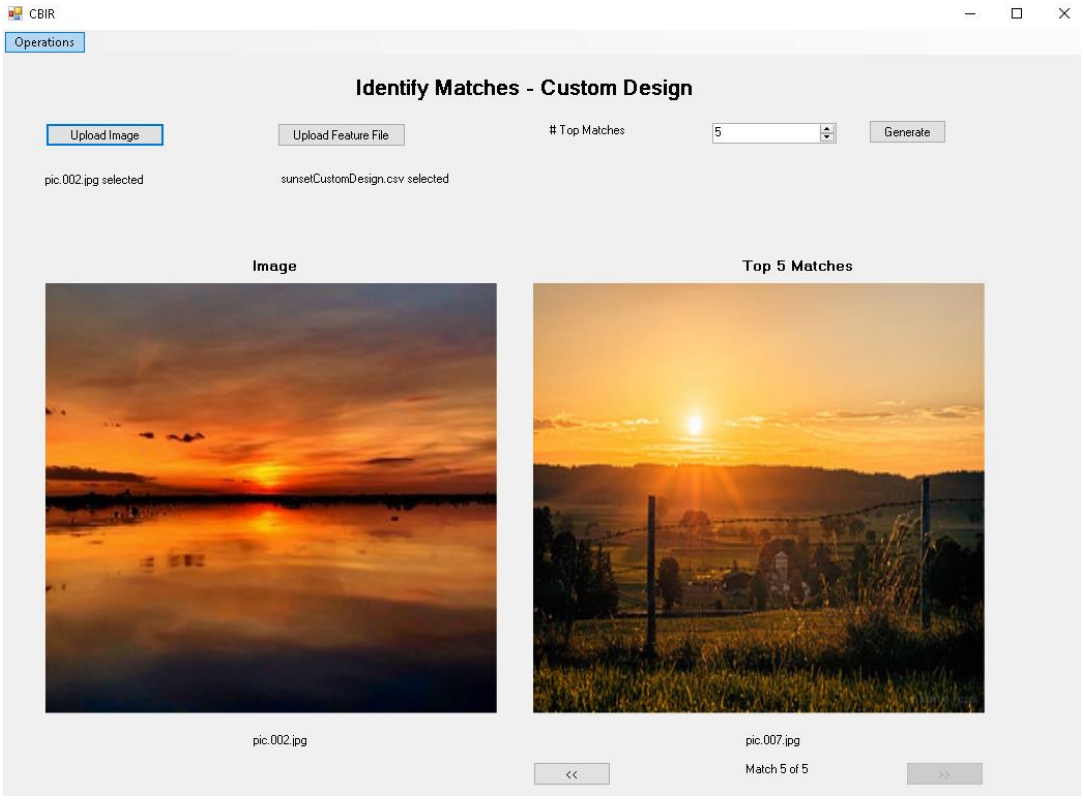

Figure 32: Custom Design Matching for pic.002.jpg displaying Top 5 Match

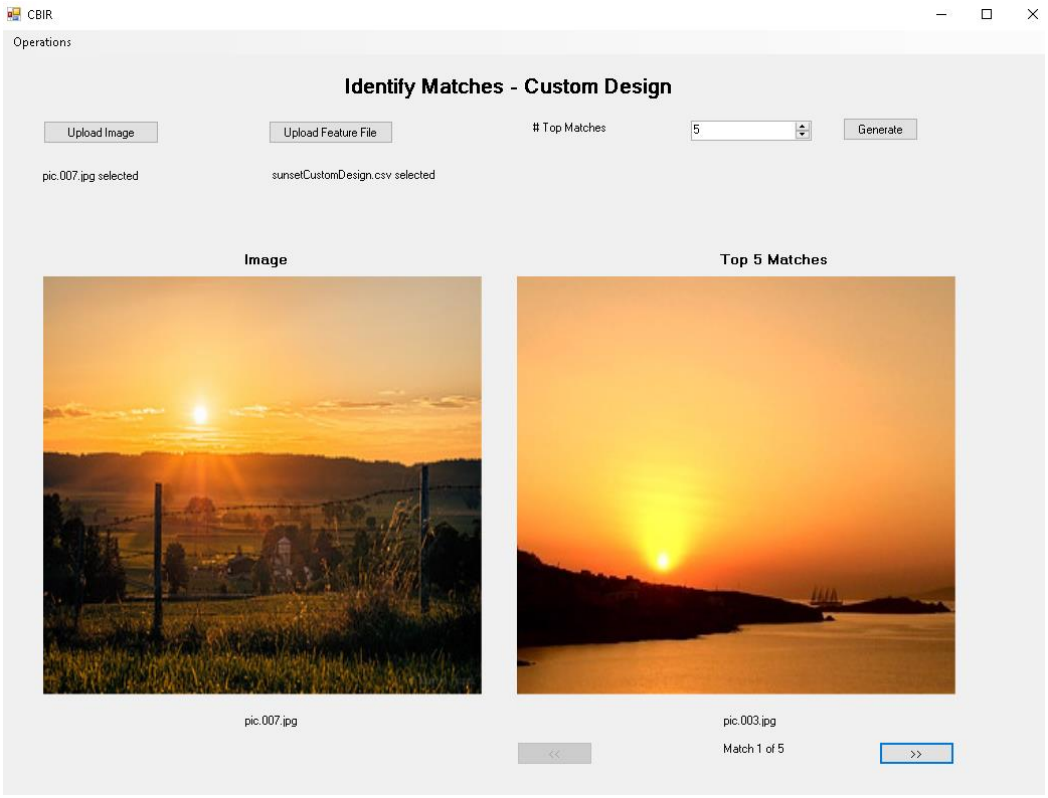

Figure 33: Custom Design Matching for pic.007.jpg displaying Top 1 Match

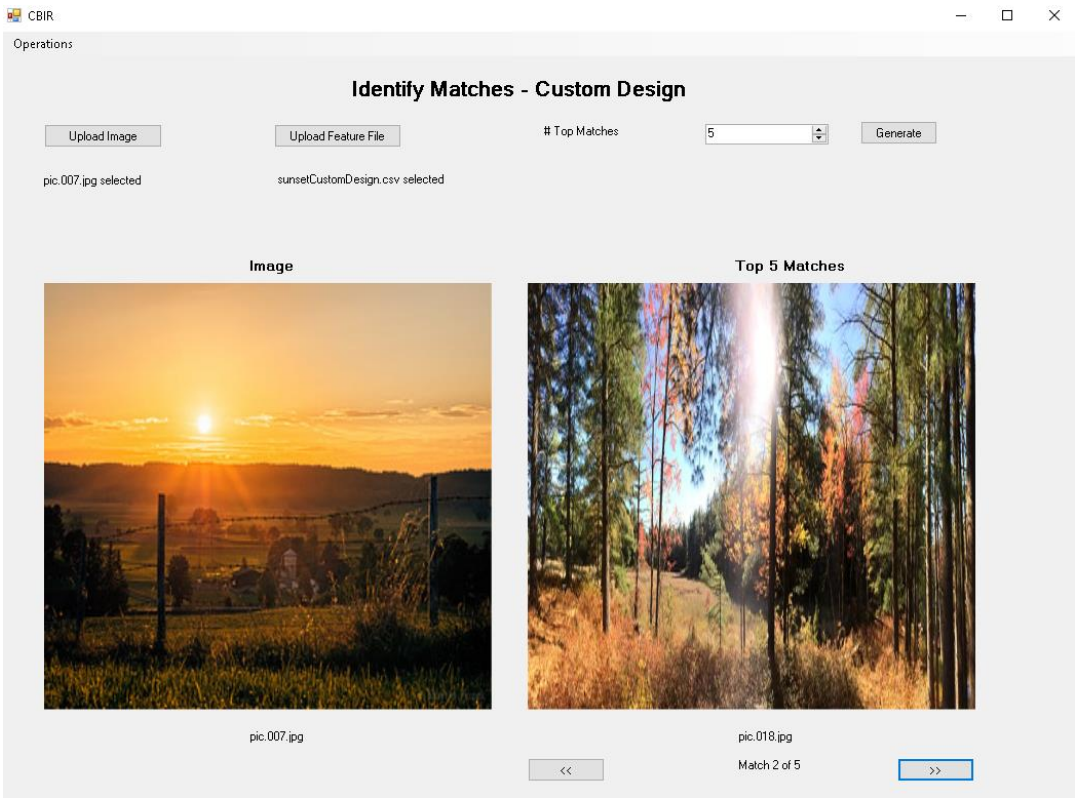

Figure 34: Custom Design Matching for pic.007.jpg displaying Top 2 Match

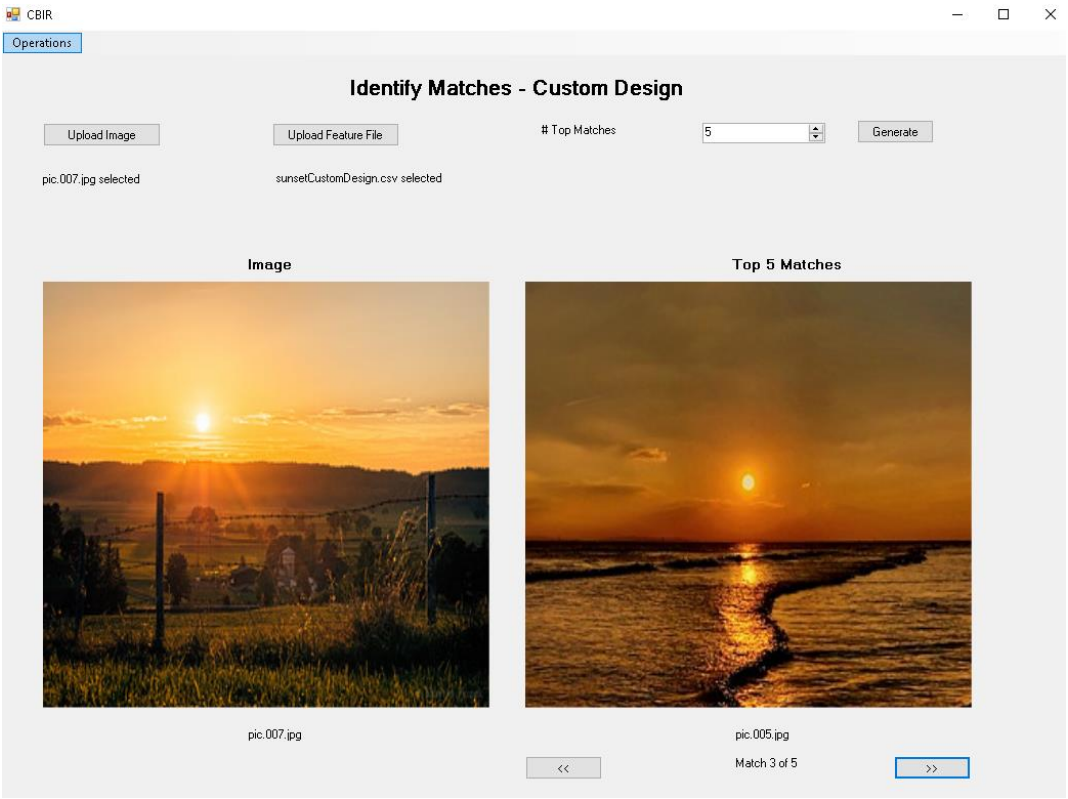

Figure 35: Custom Design Matching for pic.007.jpg displaying Top 3 Match

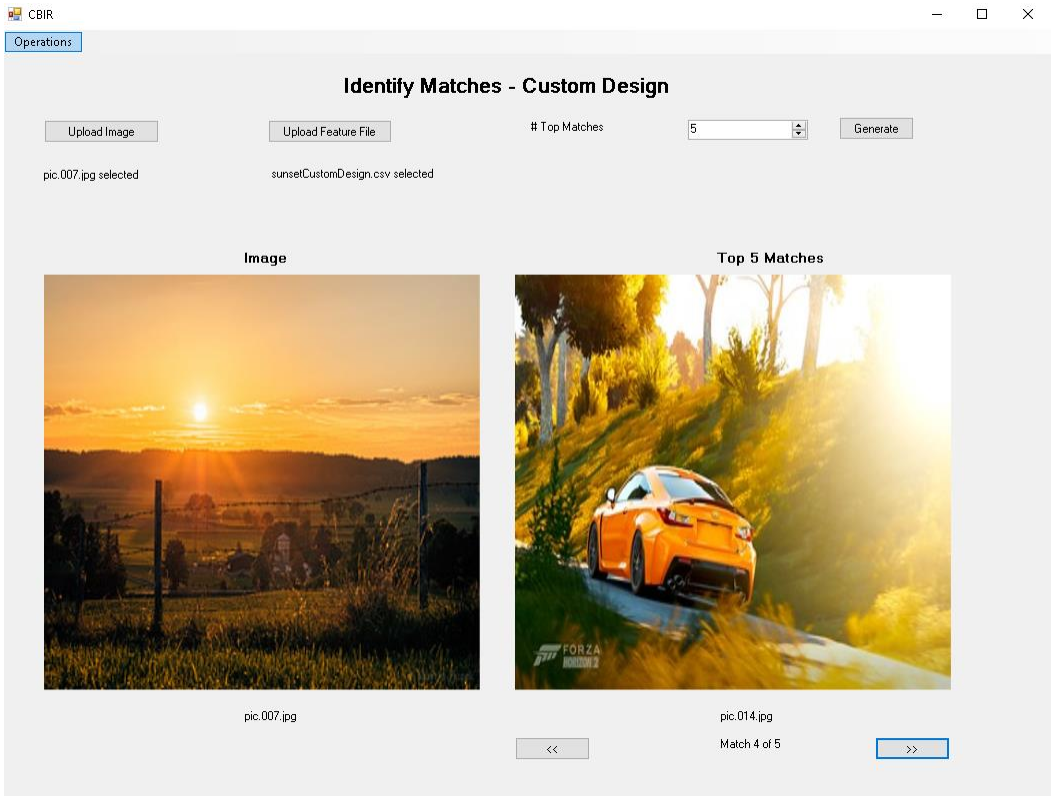

Figure 36: Custom Design Matching for pic.007.jpg displaying Top 4 Match

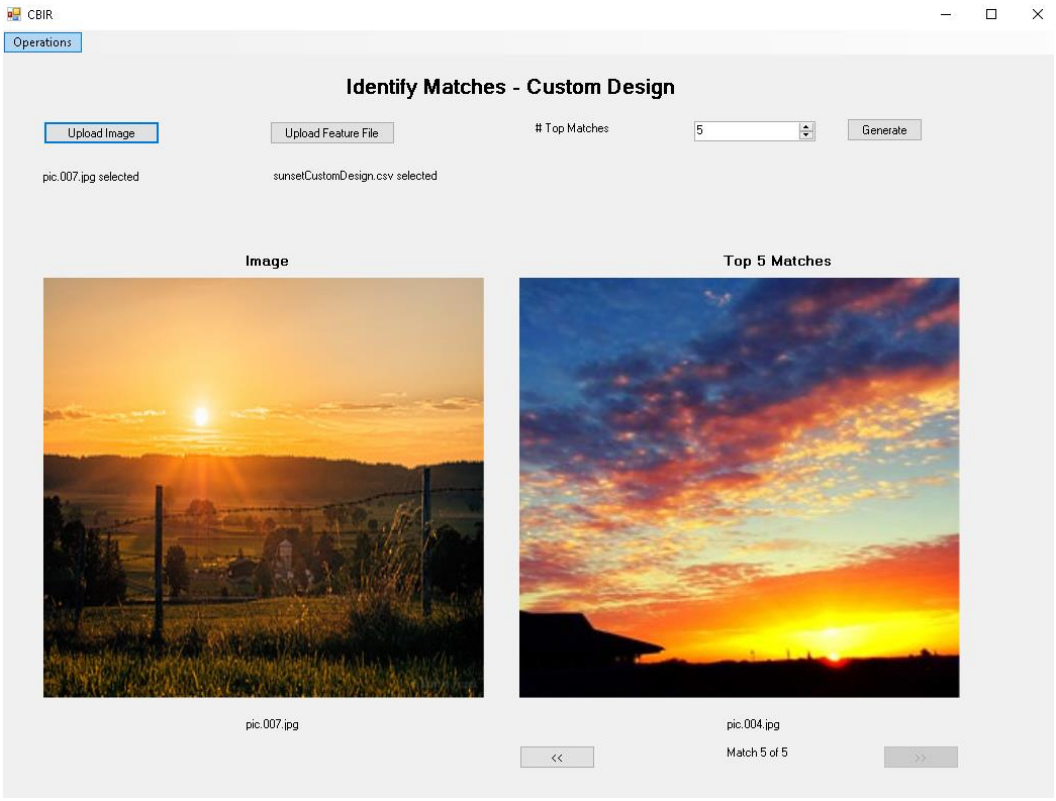

Figure 37: Custom Design Matching for pic.007.jpg displaying Top 5 Match

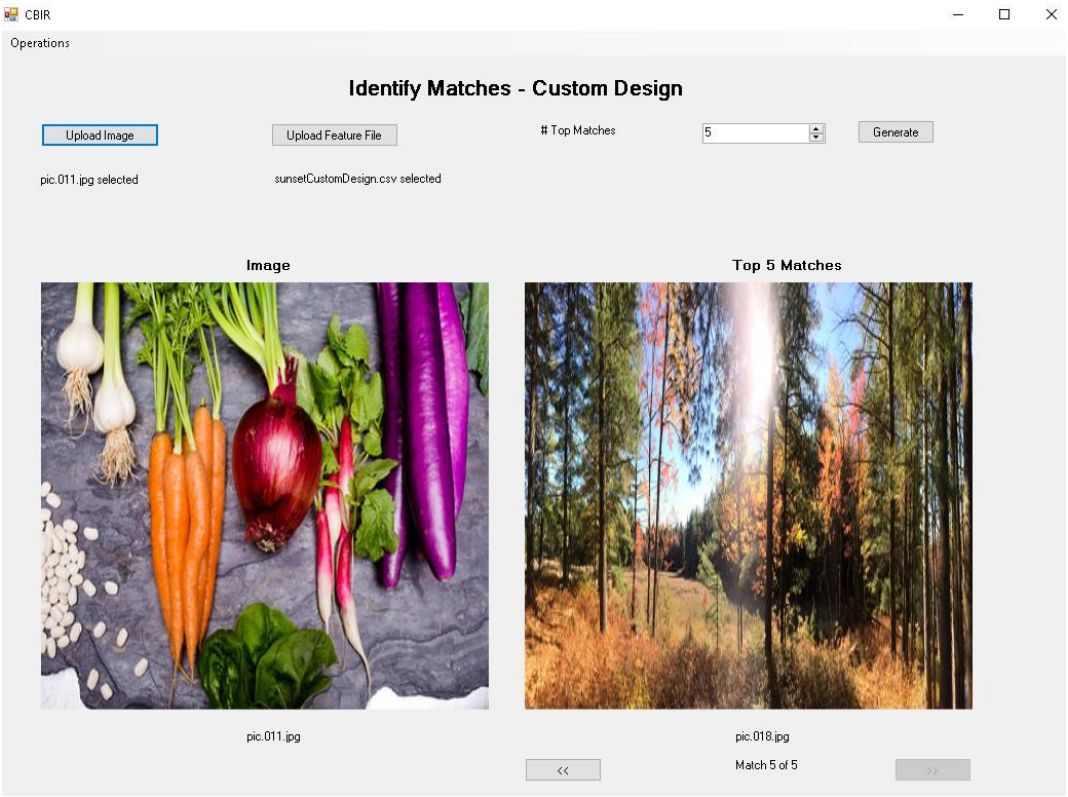

Figure 38: Custom Design Matching for pic.011.jpg displaying Top 1 Match

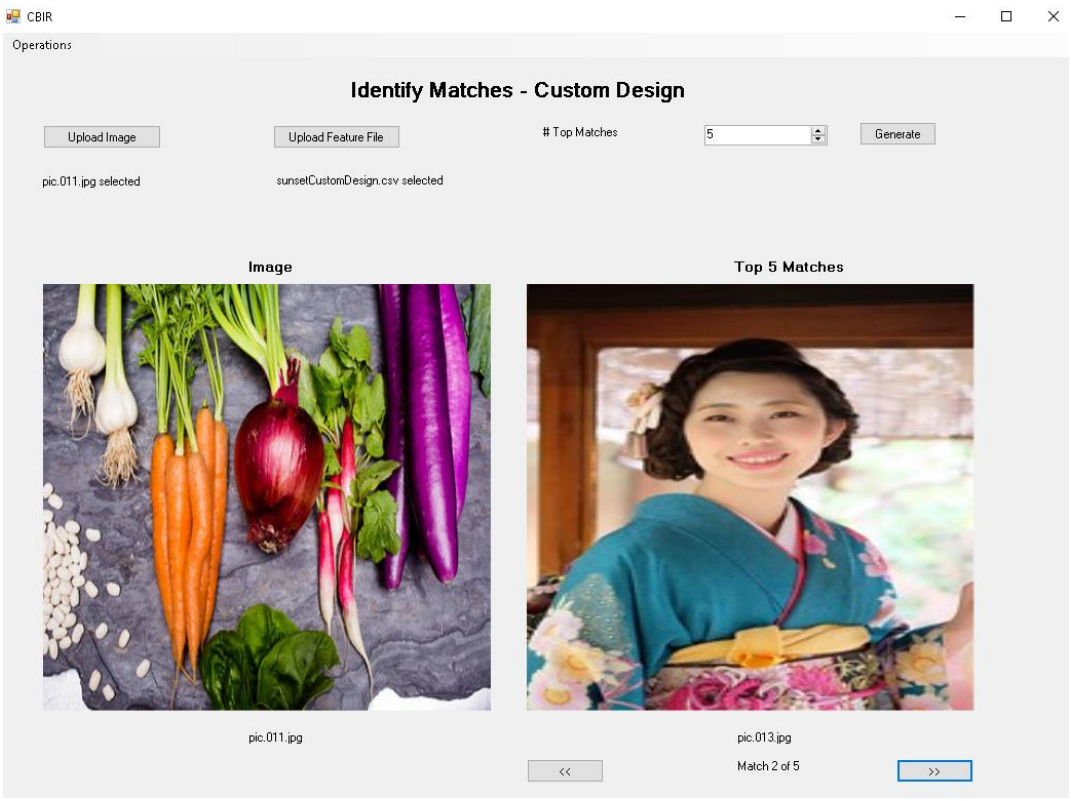

Figure 39: Custom Design Matching for pic.011.jpg displaying Top 2 Match

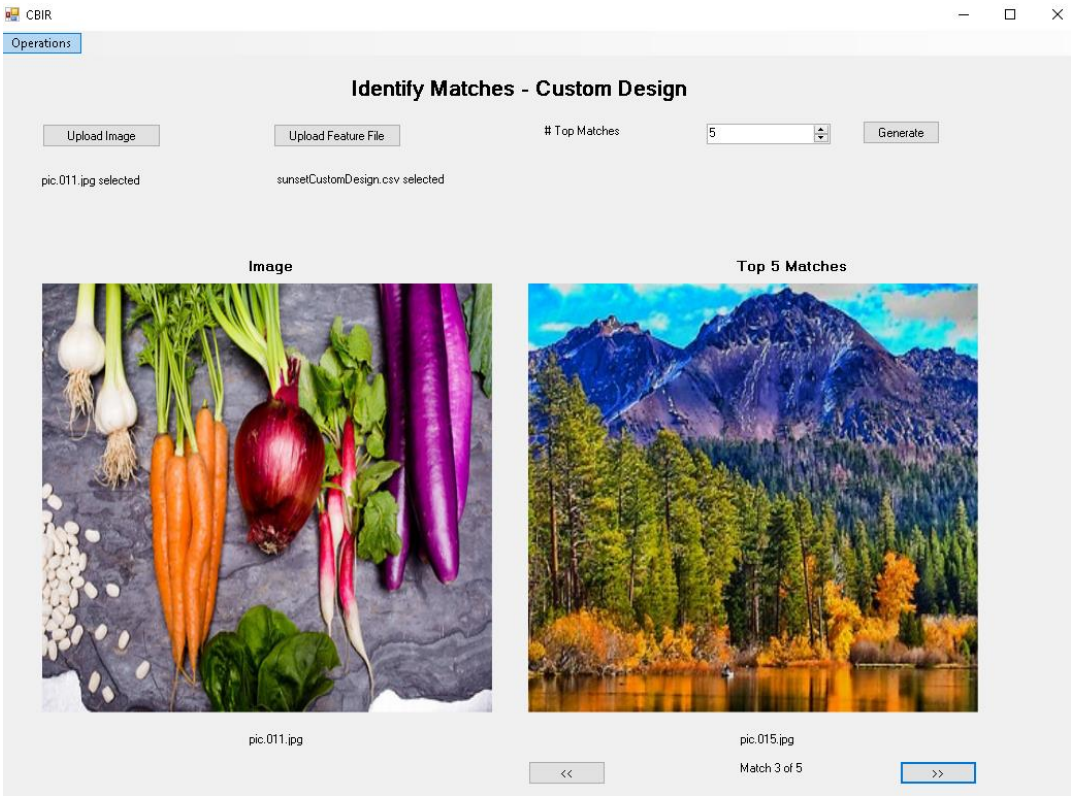

Figure 40: Custom Design Matching for pic.011.jpg displaying Top 3 Match

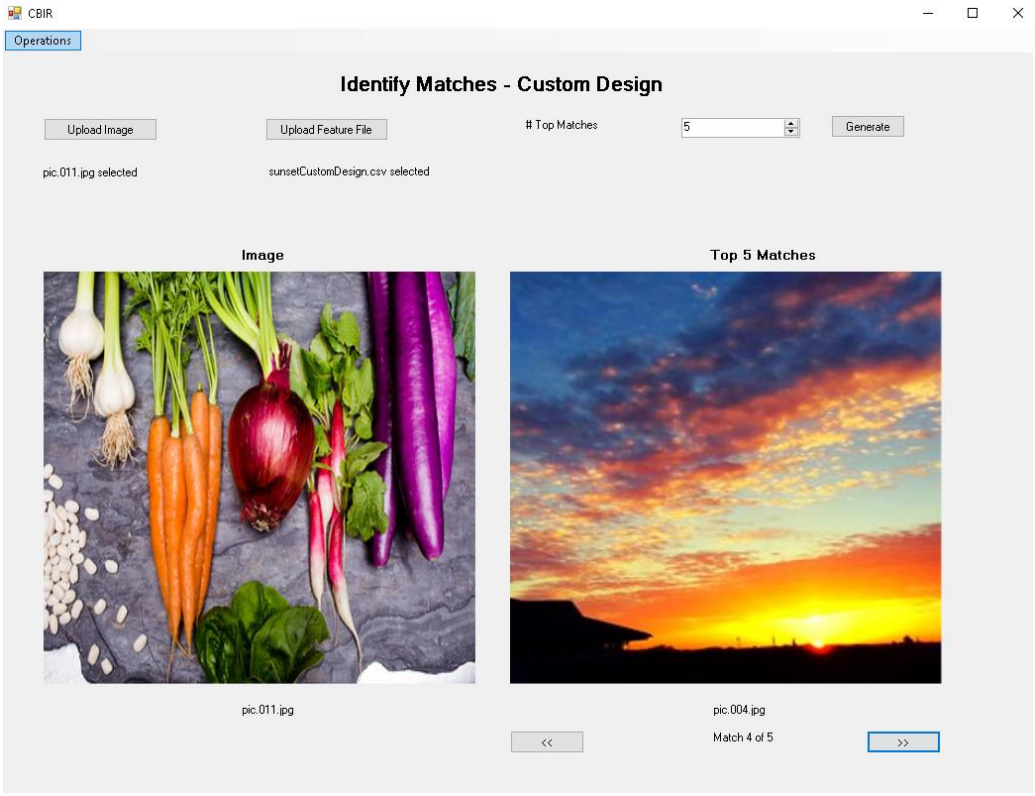

Figure 41: Custom Design Matching for pic.011.jpg displaying Top 4 Match

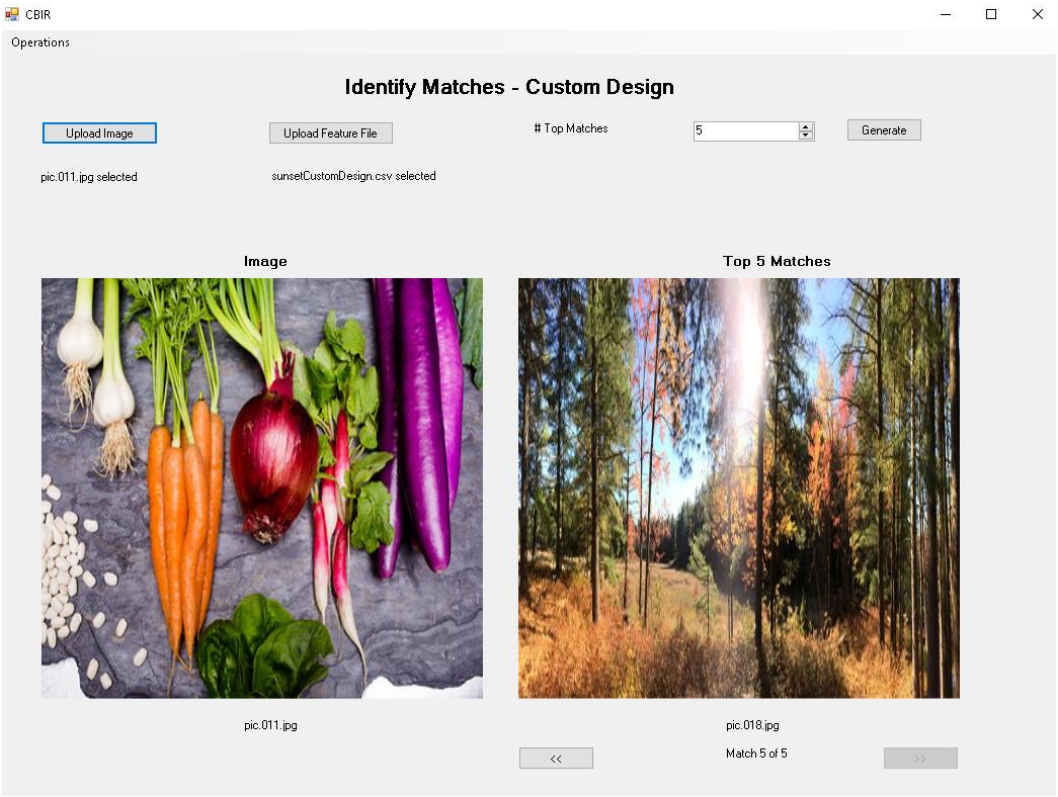

Figure 42: Custom Design Matching for pic.011.jpg displaying Top 5 Match

9. Extensions:

9.1 Custom Design - CBIR with Face Detection Features

The CBIR system has been tailored to enhance the retrieval of images based on the presence and characteristics of faces within them. Utilizing a combination of color histograms, texture features, deep neural network embeddings, and specifically crafted face features, the system aims to match images containing faces with higher accuracy.

Target Image pic.0006.jpg (Classroom Setting):

- Top Matches: pic.0002.jpg, pic.0012.jpg, pic.0001.jpg, pic.0003.jpg, pic.0044.jpg
- Observations:
    - The matches were primarily determined by the detection and features of faces within the images.
    - pic.0002.jpg and pic.0001.jpg feature prominent individuals, suggesting that the face detection component effectively influenced the retrieval results.
    - pic.0012.jpg, pic.0003.jpg, and pic.0044.jpg contain facial features that were recognized by the system, despite differences in image composition and context, showcasing the model's ability to generalize the presence of faces across varied scenarios.

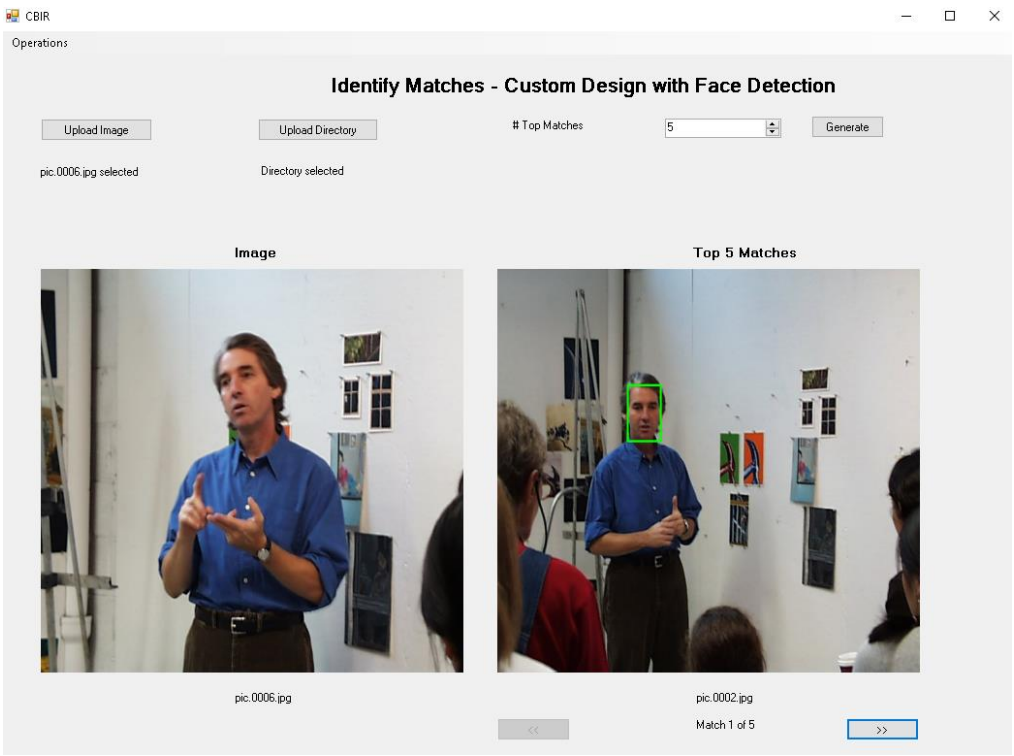

Figure 43: Custom Design with Face Detection showing Top 1 match.

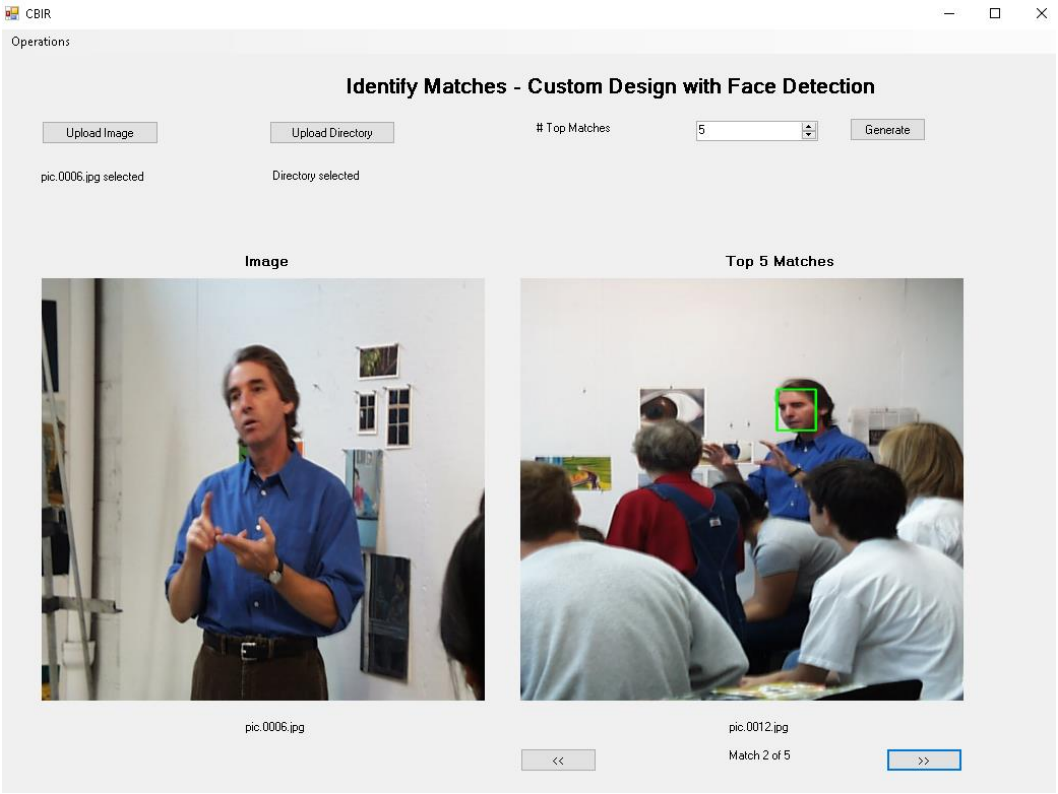

Figure 44: Custom Design with Face Detection showing Top 2 match.

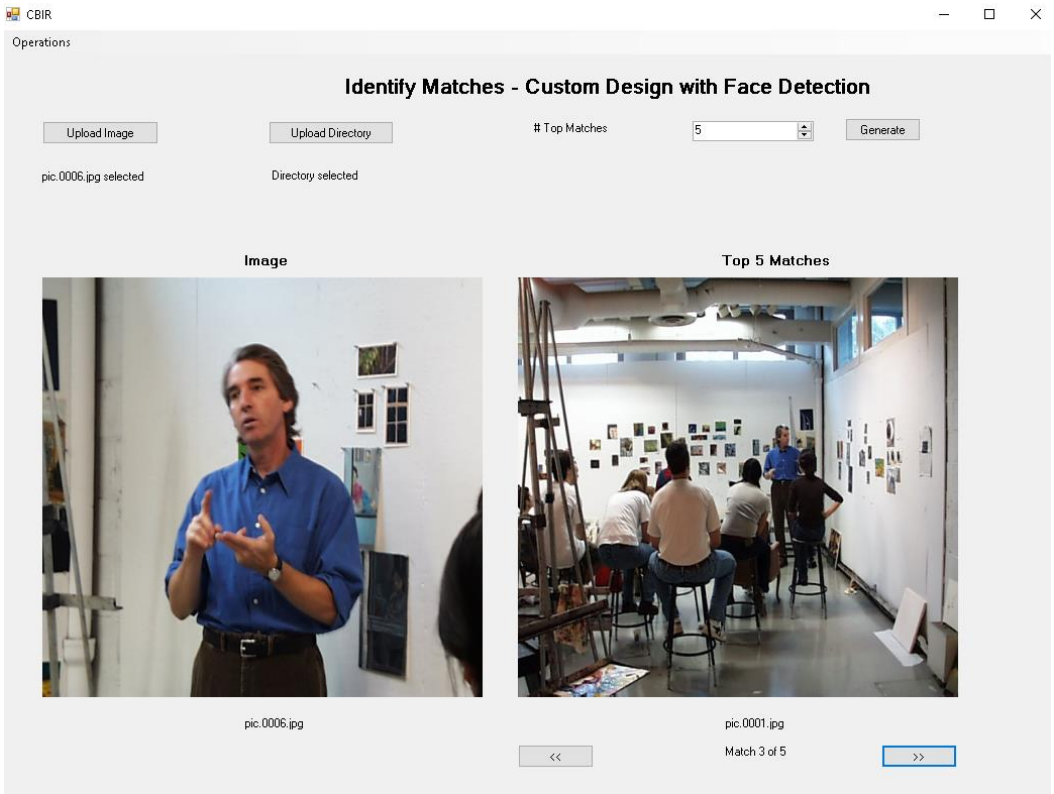

Figure 45: Custom Design with Face Detection showing Top 3 match.

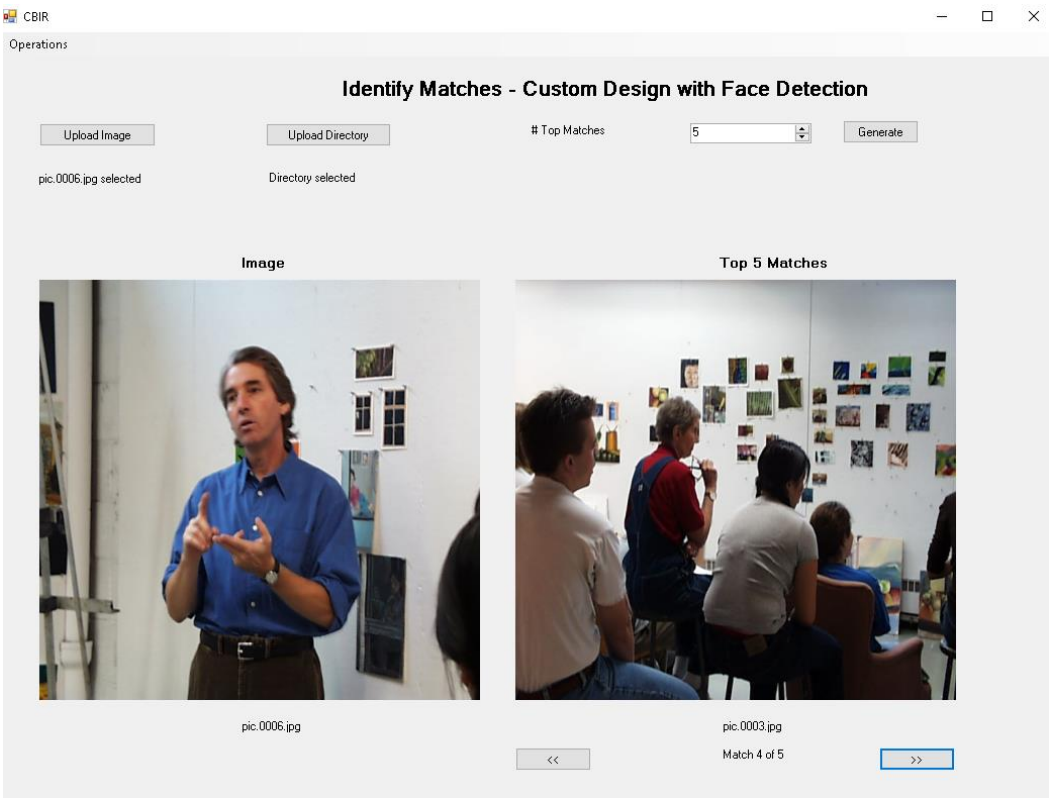

Figure 46: Custom Design with Face Detection showing Top 4 match.

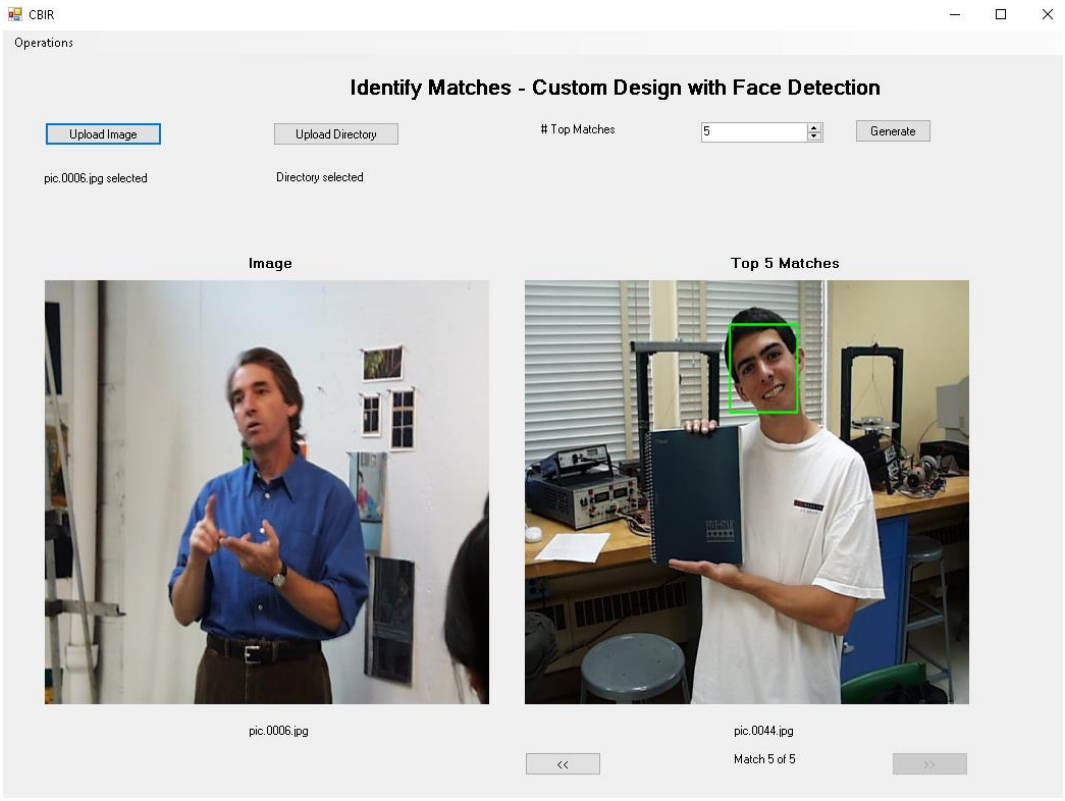

Figure 47: Custom Design with Face Detection showing Top 5 match.

9.2 Graphical User Interface for CBIR System

The CBIR system has been significantly enhanced with the introduction of a Graphical User Interface (GUI), making it more user-friendly and accessible. The GUI serves as an interactive platform that allows users to engage with the system beyond the limitations of a command-line.

Figure 48: Demonstrate GUI - User selects Generate Feature Vector for Multi-Histogram Matching

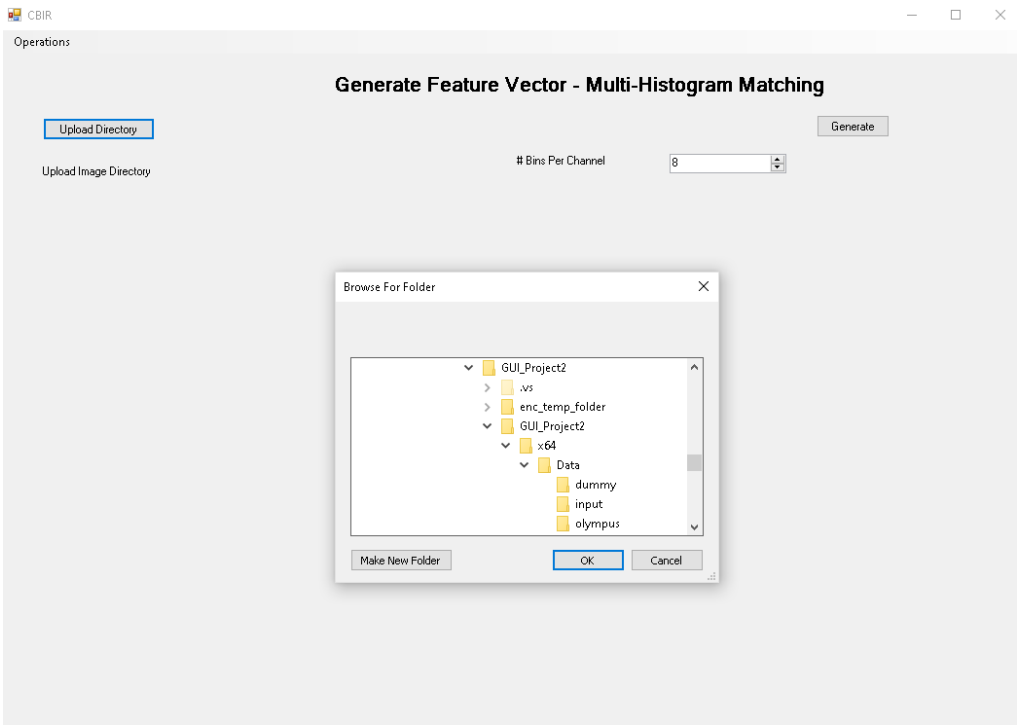

Figure 49: Demonstrate GUI - User upload Image Directory

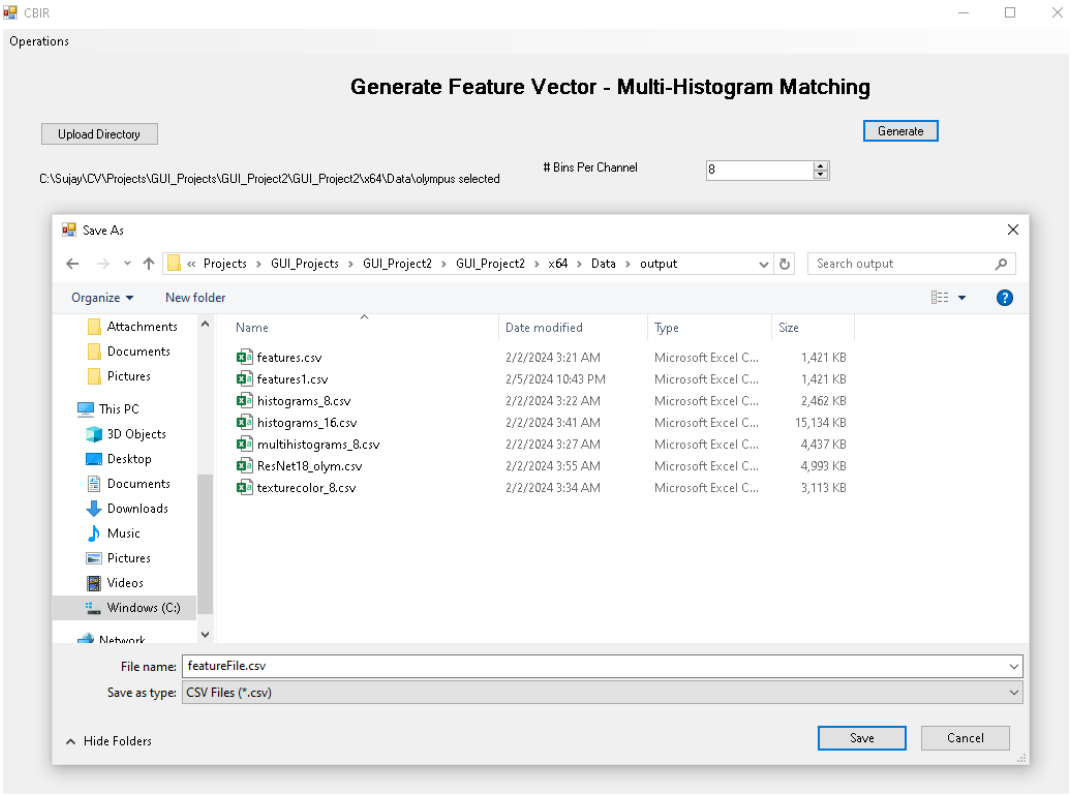

Figure 50: Demonstrate GUI - User generating and saving the Features Vector file

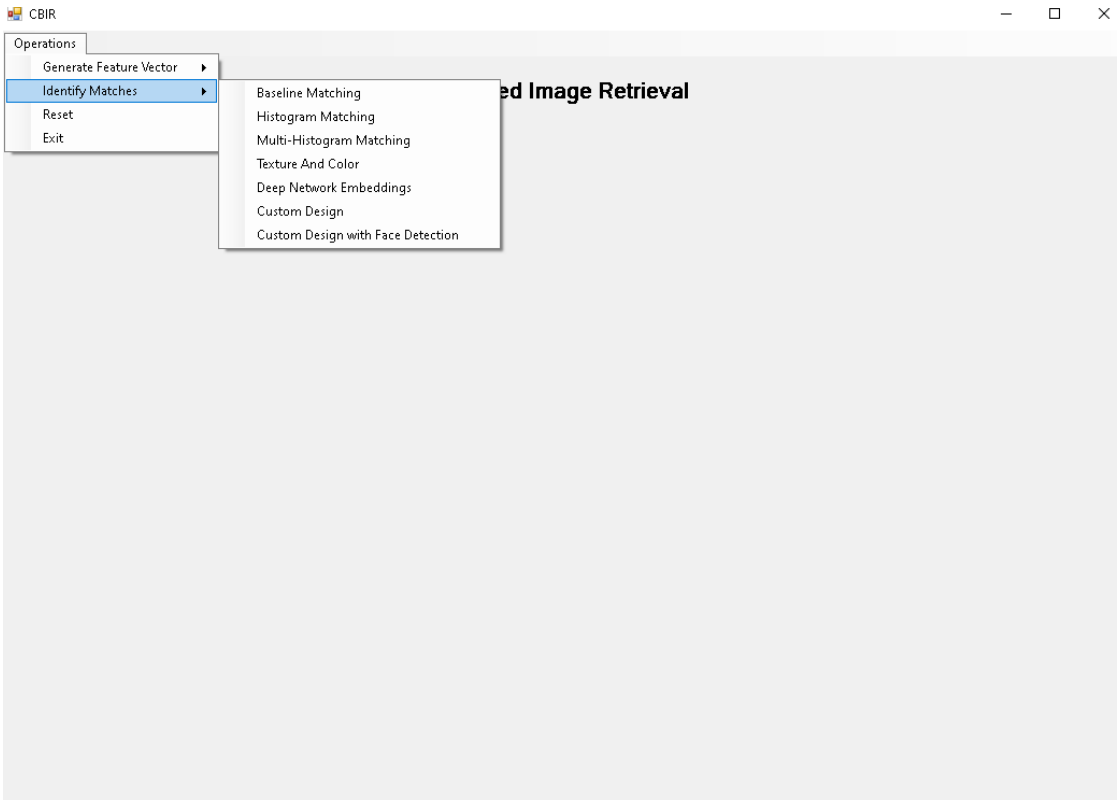

Figure 51: Demonstrate GUI - User selects Identify Matches to match the top results.

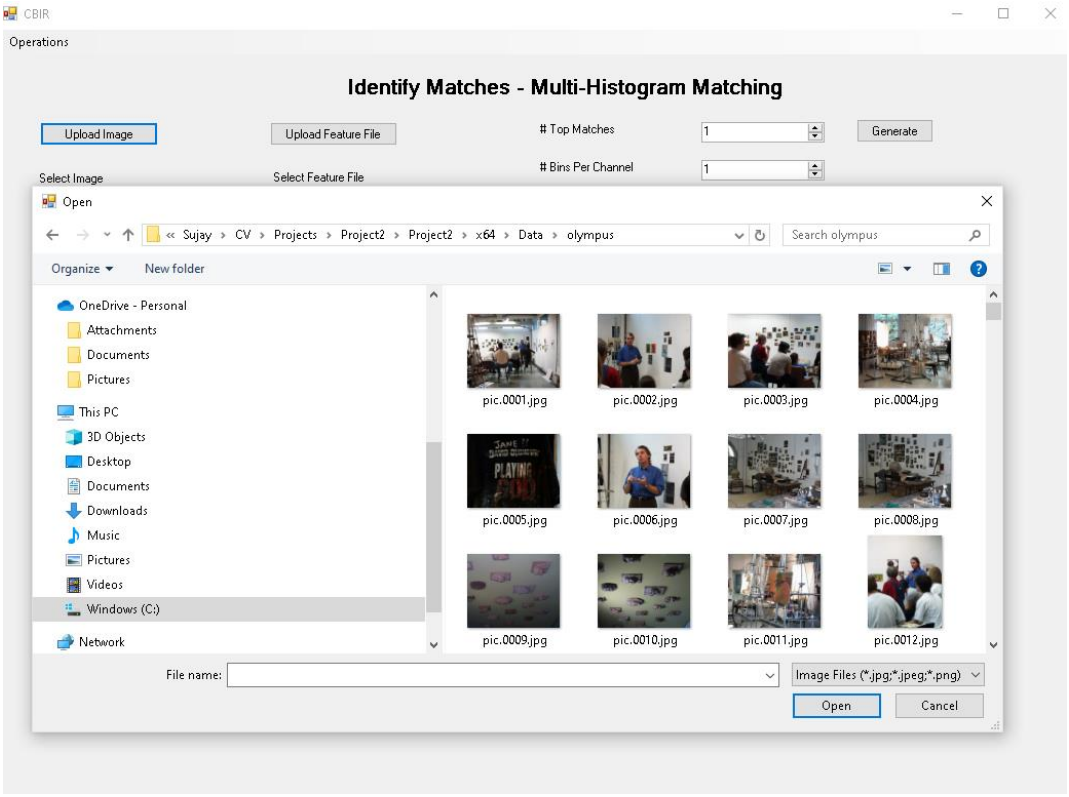

Figure 52: Demonstrate GUI - User uploading an image.

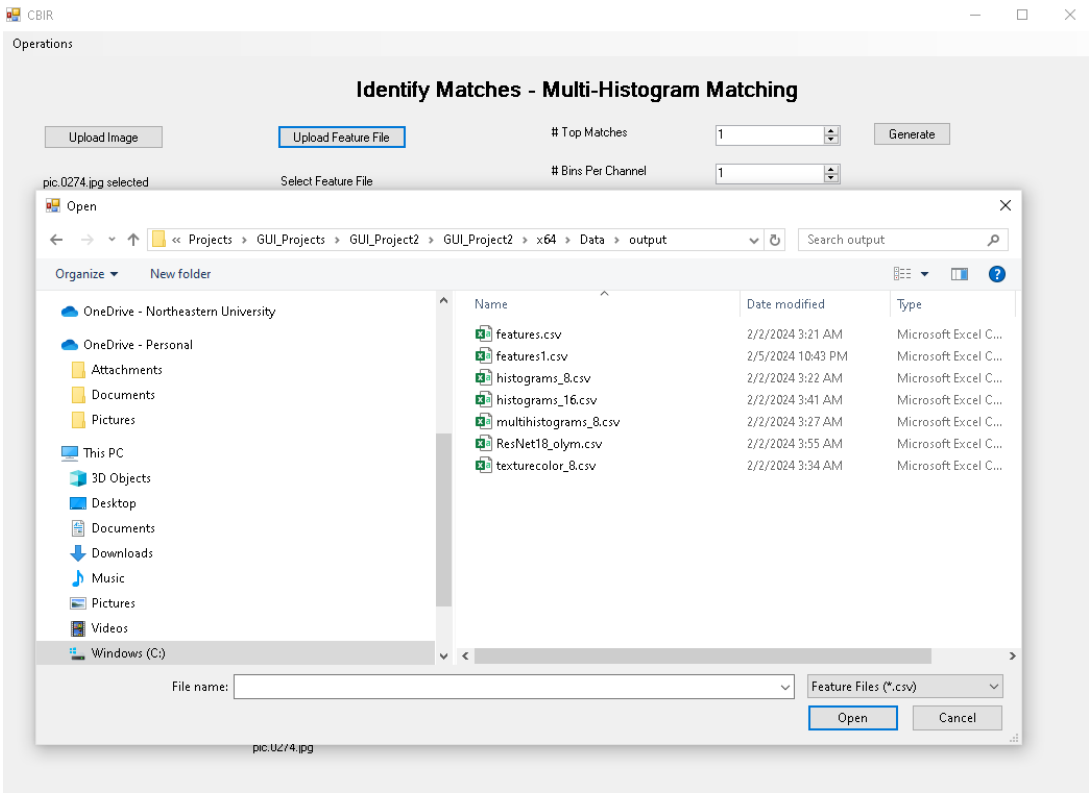

Figure 53: Demonstrate GUI - User uploading Feature vector file.

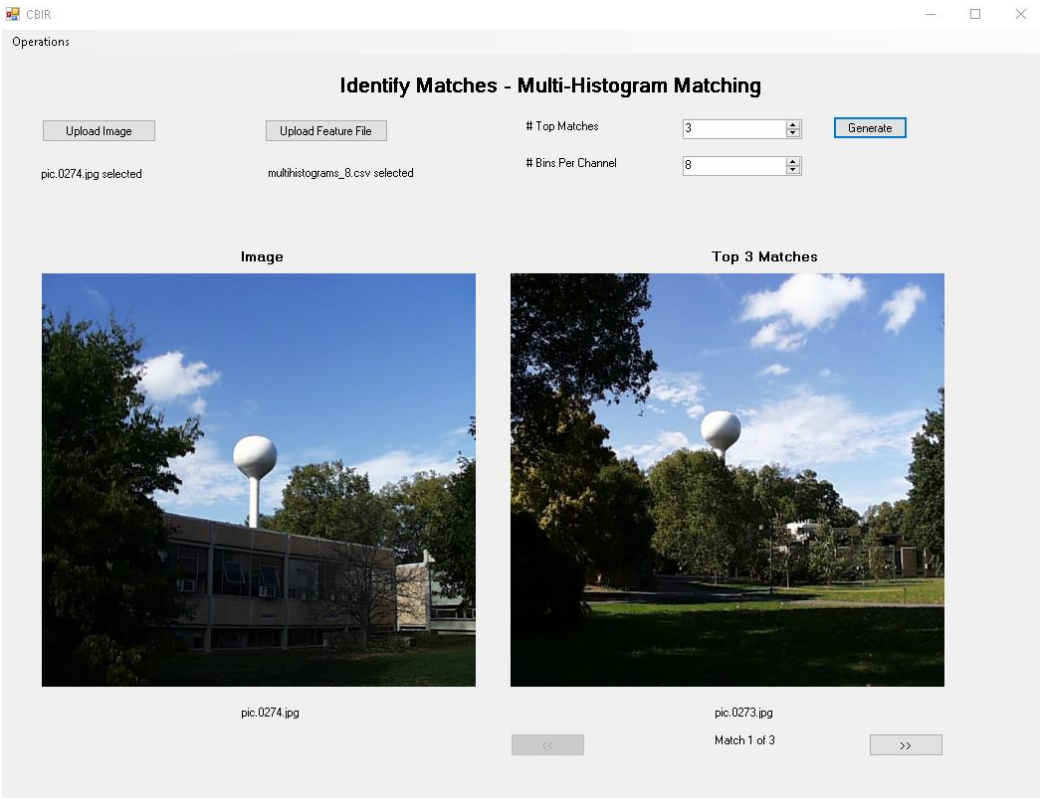

Figure 54: Demonstrate GUI - User generating the result and showing the matches.

A Short Reflection of What I Learned

Taking on the task of creating a Content-Based Image Retrieval (CBIR) system has been a rewarding experience that has allowed me to use what I have learned outside of the classroom. By exploring the intricacies of image retrieval methods, ranging from basic color histograms to intricate deep neural network embeddings, I was able to gain a nuanced comprehension of the problems and methods in computer vision.

I was able to gain direct experience with the nuances of parameter tweaking and how it affects retrieval system efficiency by using C++ to develop these algorithms. I was able to understand the complexities involved in feature extraction and image representation thanks to this exercise, which acted as a link between theoretical ideas and practical applications.

The integration of multiple disciplines, including mathematics and computer science, became evident as the project progressed. I learned that the construction of an effective CBIR system is not just about programming but also involves understanding the mathematical foundations that underlie feature representation and similarity measures.

Evaluating the performance of the retrieval system underscored the importance of robust metrics that consider not only the visual similarity but also the relevancy and precision of the results returned. This aspect of the project highlighted the iterative nature of developing a CBIR system, where continuous refinement and testing play a pivotal role.

The creation of a graphical user interface (GUI) was a transformative part of the project, as it illuminated the significance of user experience in the field of computer vision. The GUI served as a vital link between the underlying algorithms and the end-users, emphasizing the need for intuitive navigation and clarity in presenting results.

Furthermore, exploring the future directions of CBIR, such as the application of artificial intelligence and machine learning, was particularly exciting. It opened my eyes to the potential for these systems to revolutionize the way we interact with and analyze visual information.

In conclusion, this project was a comprehensive and hands-on learning experience that significantly enhanced my understanding of image processing and computer vision techniques.

Acknowledgements

This project was enriched and made possible by a range of resources and materials that provided invaluable guidance and insight. I would like to acknowledge the following sources:

Textbooks:

- "Computer Vision: Algorithms and Applications, 2nd Edition" by Richard Szeliski: This comprehensive guide served as the cornerstone of the project, offering crucial insights into computer vision principles that are fundamental to CBIR systems.
- Additional readings and literature provided context and depth to the algorithms and techniques implemented throughout the project.

Websites:

- The OpenCV Documentation (https://docs.opencv.org/master/): Provided thorough and practical insights into the application of OpenCV functions and methodologies pertinent to image retrieval.
- Stack Overflow (https://stackoverflow.com/): A critical resource for resolving specific coding challenges and learning from the experiences of a global developer community.
- GitHub Repositories (https://github.com/): The open-source projects and code snippets available were instrumental for understanding and implementing various features of the CBIR system.

Personal Acknowledgements:

- I would also like to express my gratitude to my instructors who provided guidance and support throughout this project. Their feedback and suggestions were crucial in overcoming challenges and enhancing the project's outcomes.

These resources, combined with the support from my academic environment, played a vital role in the successful execution of this project. I am grateful for the wealth of knowledge and expertise shared by these sources.