Faculty of Engineering & Technology

Electrical & Computer Engineering Department

**Computer Vision**

**ENCS5343**

*Assignment 2*

**Prepared by: Majd Abubaha**

**ID Number: 1190069**

**Instructor: Dr. Aziz Qaroush**

**Section: 2**

**Date: 4/1/2024**

## Contents

## Table of Figures

# 1 Introduction

## 1.1 Content-Based Image Retrieval (CBIR)

Content-Based Image Retrieval (CBIR) is an advanced method that goes beyond using verbal annotations or metadata alone to search and retrieve photos based on their visual content. Enabling users to retrieve photos that visually resemble a query image is the main objective of CBIR systems. In order to find commonalities, this procedure compares and analyzes visual characteristics that have been taken from photos.

## 1.2 Theoretical Background of Color Histograms and Color Moments

Within CBIR, color is essential to the display of images. Basic ideas in the categorization of picture content based on color information are color moments and color histograms.

The distribution of colors in an image is statistically represented by color histograms. A color histogram efficiently represents the color distribution by quantizing the color space into discrete bins and counting the occurrences of pixels in each bin. This provides a brief but useful description of the chromatic content of the image.

Conversely, Color Moments transcend the frequency-based depiction of color. They include statistical metrics that shed light on the central tendency, spread, and asymmetry of color distributions, such as mean, standard deviation, and skewness. By taking into account not just the frequency of color but also its fluctuations and patterns of distribution, color moments provide a more complex description of the color content in photographs.

# 2 Dataset

This assignment was made using the **Wang database**.

This classic dataset comprises 1,000 photos in ten distinct categories. Easy to use and popular for assessing fundamental CBIR algorithms. Every 100 image is a category, for example 0-99 is a category, 100-199 is another category, and so on.

# 3 Task 1

## 3.1 System Implementation

### 3.1.1 General Architecture and Functionality of the Modules

The implemented CBIR system has a modular design that includes essential parts for ranking, feature extraction, distance calculation, and result display.

- **Feature Extraction Module:** Every image in the collection has its color characteristics extracted by the system. In particular, it generates a feature vector for every image by computing the mean and standard deviation characteristics for every color channel.
- **Distance Computation Module:** The system calculates the dissimilarity between the query image's color features and all of the dataset's pictures using Euclidean distance.
- **Ranking Module:** The dataset photos are ranked according to how similar they are to the query image using the calculated distances.
- **Result Visualization Module:** The system presents the top-ranked retrieval results and the query picture as a visual representation. This makes it easier to comprehend the retrieval process in its entirety.

### 3.1.2 Implementation Details

Using NumPy for numerical calculations, Matplotlib for result presentation, and the OpenCV library for image processing, the system is written in Python. Important roles consist of:

- **cv2.imread():** Opens file paths to load pictures.
- **cv2.split():** Split picture channels for separate processing.
- **np.mean() and np.std():** to find the color channel values' mean and standard deviation.
- **np.linalg.norm():** Find the Euclidean distance between two feature vectors.
- **np.argsort():** Sort indices using obtained distances.
- **plt.imshow(), plt.title(), and plt.axis():** Plot characteristics are specified and pictures are seen.

### 3.1.3 Functionality Overview

The complete CBIR process is included in the CBIRSystemColorFeatures class. Images are loaded during initialization, and each image is subjected to feature extraction. After that, the algorithm calculates distances, rates the outcomes, and shows the query image next to the retrievals that are scored highest. Using a given query picture, the run_experiment_once method performs a retrieval experiment to show off the capabilities of the system.

## 3.2 Experimental Setup and Results

### 3.2.1 Evaluation Metrics

- **Precision:** measures how well-relevant retrievals among the top-ranked results are retrieved.
- **Recall:** Evaluates how well the system can locate and retrieve every pertinent image in the collection.
- **F1-Score:** Provides a composite measure for retrieval efficacy by balancing recall and accuracy.

### 3.2.2 Experimental Setup

In a single experiment, the CBIR system analyzed the query image (133.jpg). The query image's color features were taken out and compared to the dataset's image features using the Euclidean distance method. The query image was displayed with the top retrievals in a visualization created after the results were ordered according to the computed distances.

### 3.2.3 Results and Analysis

The experiment effectively demonstrated the operation of the CBIR system and shed light on the accuracy of its retrieval. The query picture and the top-ranked retrievals were shown in the displayed results, providing a visual depiction of the system's functionality.


*Figure 1. CBIR System Using Color Features*

#### 3.2.4 Limitations and Future Improvements

One of the existing system's drawbacks could be its dependence on basic color characteristics, which could reduce its ability to discriminate. To improve retrieval accuracy, future research might investigate more complex feature extraction methods and sophisticated distance measurements.

## 4 Task 2

### 4.1 System Implementation

#### 4.1.1 General Architecture and Functionality of the Modules

Color histograms are the main emphasis of the developed Content-Based Image Retrieval (CBIR) technology. The CBIRSystemColorHistogram class, which is built with the parameters dataset_path and num_pins (bins in the color histogram), encapsulates the CBIR system. The system's main modules are as follows:

- **Initialization Module:** __init__(self, dataset_path, num_pins) loads picture paths, extracts features, and labels the system to start.

- **Feature Extraction Module:** The system is initialized with __init__(self, dataset_path, num_pins), which loads picture paths, extracts features, and assigns labels.
- **Histogram Calculation Module:** Calculate_color_histogram(self, picture) computes histograms and normalizes images after converting them to float32.
- **Distance Computation Module:** The function compute_distance(self, query_features) calculates the distance between query and dataset pictures by use of histogram intersection.
- **Ranking Module:** The query picture is not included in the descending order of results ranked by rank_results(self, distances, query_index), which is based on computed distances.
- **Results Display Module:** Grid visualization of query and retrieval results is achieved using display_results(self, ranked_results, query_index).

### 4.1.2 Implementation Details

Python is used to implement the system, and time libraries, NumPy, Matplotlib, and OpenCV are used. The CBIRSystemColorHistogram class facilitates successful feature extraction and retrieval by effectively organizing and processing pictures.

### 4.1.3 Functionality Overview

With an emphasis on accuracy and efficiency, the CBIR System uses color histograms for content-based image retrieval. The CBIRSystemColorHistogram class, which is initialized with the path to the dataset and the number of bins (pins) for the color histogram, contains its fundamental functionality. After loading the images, their characteristics are retrieved via scaling, RGB conversion, and histogram computation. Using histogram intersection, the distances between the query and dataset pictures are calculated. With the query image excluded, the retrieval results are sorted according to similarity. Users may examine the top retrieval results and the query picture thanks to visualization capabilities. The system provides quantitative assessment by supporting metric computations for accuracy, recall, and F1-score. Through experimentation with various pin arrangements, the system offers insights regarding retrieval performance. The understanding of results is further improved by the ROC curve analysis.

## 4.2 Experimental Setup and Results

### 4.2.1 Evaluation Metrics

This is an illustration of how the f1 score, recall, and precision are calculated by the system:

*Equation 1. Precision*

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

*Equation 2. Recall*

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

*Equation 3. False Negatives*

$$\text{False Negatives} = \text{Total number of images in the category - True Positives}$$

*Equation 4. F1 Score*

$$\text{Precision} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **True Positives:** The number of images from the same category as the query image that were correctly retrieved.
- **False Positives:** The number of images from another category that were correctly retrieved.
- **False Negatives:** The number of images from the same category as the query image that were not retrieved by the system.

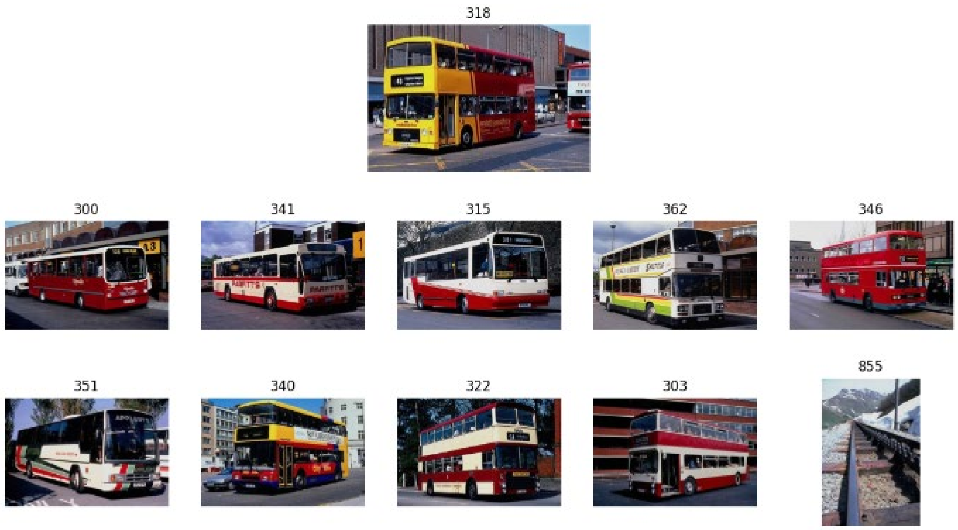


*Figure 2. Query Image With 10 Retrieved Images*

```
Query Image: .\dataset\318.jpg
Precision: 0.9, Recall: 0.08256880733944955, F1 Score: 0.15126050420168066
```

*Figure 3. Calculated Precision, Recall, and F1 Score*

### 4.2.2 Experimental Setup

In order to see how changing the number of pins in the color histogram (num_pins) affects retrieval efficiency and accuracy, tests are conducted. Ten randomly chosen queries are run by the system from the dataset (the threshold 10 could be changed).

### 4.2.3 Results and Analysis

In order to visualize the performance of the system, metrics such as recall, precision, and F1-score are displayed together with the Receiver Operating Characteristic (ROC) curve.

The display function will show the photos that were retrieved.

The system uses precision, recall, and F1-score measures to assess how useful certain color attributes are. The implementation highlights a factor affecting retrieval accuracy: the number of pins in the color histogram.

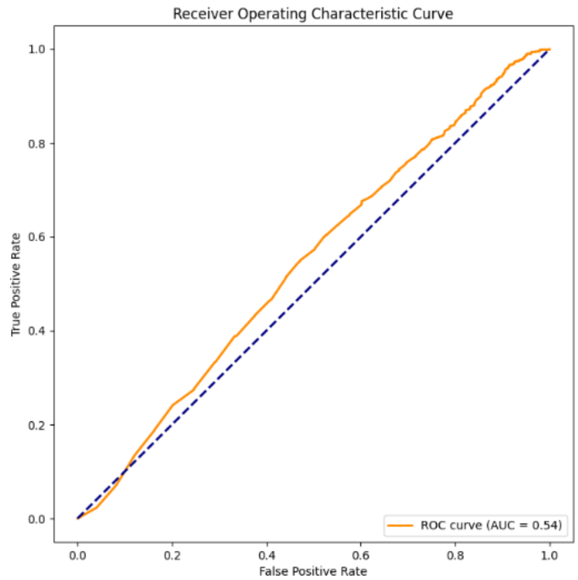


*Figure 4. ROC Curve with 120 Pin and 10 Images Retrieved*

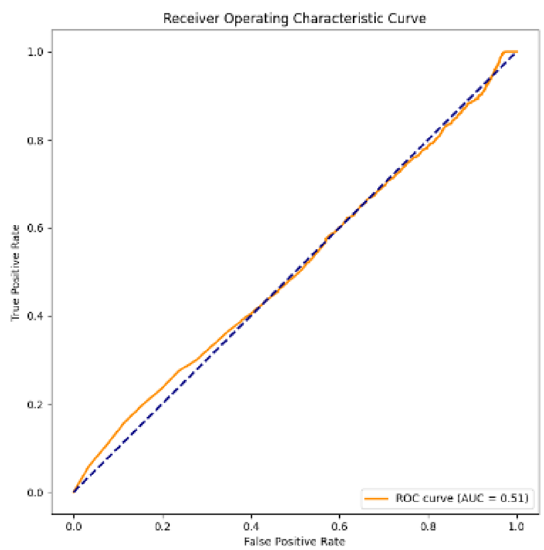


*Figure 5. ROC Curve with 120 Pin and 20 Images Retrieved*

### 4.2.4 Limitations and Future Improvements

The fine-grained retrieval process may not be able to achieve the necessary discriminative power when using color histograms. Investigating more sophisticated feature representations, including features based on deep learning or texture, may improve efficiency.

System performance is greatly impacted by the number of pins (num_pins) used for the color histogram. Sensitivity might be reduced by using an automated tuning process or a more flexible strategy.

# 5 Task 3

## 5.1 System Implementation

### 5.1.1 General Architecture and Functionality of the Modules

Color moments are used as characteristics for picture representation in the developed Content-Based picture Retrieval (CBIR) system. The system is made up of many modules that collaborate to accomplish picture retrieval and assess the system's effectiveness:

- **CBIRSystemColorMoments Class**:
  - **Initialization Module:** Extracts image paths, initializes the system, and accepts the dataset path as input. The extract_color_moments function is then invoked.
  - **Feature Extraction Module (extract_color_moments):** The extract_features function is used to extract color moments features while the feature extraction module (extract_color_moments) iterates over the picture paths and reads photos.
  - **Feature Extraction Method (extract_features):** Utilizing the extract_features function, the feature extraction method generates color moments (mean, standard deviation, skewness) for each channel (R, G, and B) after converting pictures to RGB format. combines these characteristics into a solitary vector.
  - **Distance Calculation Method (compute_distance):** Determines the distances between each query picture and every image in the dataset based on the attributes of the query image.
  - **Result Ranking Method (rank_results):** The photos are ranked according to their distances from the query image using the "rank_results" method of result ranking.
  - **Results Display Method (display_results):** The query picture and the top retrieval results are displayed using the results display method (display_results).

- **Performance Evaluation Method (evaluate_performance):** The method for evaluating performance that calculates the F1 score, precision, and recall for a given query.

- **Experiment Runner Function (run_experiment_color_moments):**
  - **System Initialization:** Using the CBIRSystemColorMoments class, a new instance of the CBIR system is created.
  - **Loop for Executing the Experiment:** Chooses a random image from each category to serve as the query image while iterating through a predetermined number of inquiries.
  - **Query Processing:** Uses the CBIR system to get the picture and extract features for the query image.
  - **Performance evaluation:** Compiles findings for an overall assessment by measuring accuracy, recall, and F1 score for every query.
  - **Standard Measures:** Average accuracy, recall, F1 score, and retrieval time are computed over all queries.
  - **ROC Curve Construction:** For performance visualization, create a Receiver Operating Characteristic (ROC) curve using labels and distances.

### 5.1.2 Implementation Details

The implemented Content-Based Image Retrieval (CBIR) system is developed in Python, leveraging several key libraries and frameworks. OpenCV (cv2) is utilized for image reading and color space conversion, while NumPy (np) is employed for numerical operations and array manipulation. Scikit-learn is utilized for computing ROC curve metrics, specifically roc_curve and auc, and Matplotlib (plt) is employed for visualizing images, performance metrics, and the ROC curve. Time is used to measure the execution time of the retrieval process, and Scipy (skew) is employed for calculating the skewness of color channels. The image representation in the system is based on color moments, including mean, standard deviation, and skewness, calculated for each channel (R, G, B). Evaluation metrics include precision, recall, F1 score, and the construction of a Receiver Operating Characteristic (ROC) curve with Area Under the Curve (AUC). The dataset is organized with images grouped into categories, each containing a specified number of images defined by the variable 'category_size.'

### 5.1.3 Functionality Overview

A flexible method for retrieving and assessing images is the Python-based Content-Based Image Retrieval (CBIR) system. The system extracts color moments, including mean, standard deviation, and skewness, from each picture channel by utilizing OpenCV for image processing, NumPy for numerical operations, and Scikit-learn for metric computation. After that, it calculates the distances between the dataset and the query photos, ranks the results according to these distances, and displays the retrieval results graphically. For each query, the system assesses performance parameters including recall, F1 score, and accuracy while taking category intervals into account when determining correctness. The system performs tests, computing average metrics and creating Receiver Operating Characteristic (ROC) curves for a thorough analysis, with flexibility in dataset route and category size.

## 5.2 Experimental Setup and Results

### 5.2.1 Evaluation Metrics

The f1 score, recall, and precision are calculated as Task 2.

### 5.2.2 Experimental Setup

Using Color Moments with the CBIR system on the selected dataset was the experimental setting. A predetermined number of queries were run through the system, each of which was selected at random from within its category. The category size (category_size) was set to 100. The total performance metrics were then calculated by averaging the results across all queries.

### 5.2.3 Results and Analysis

#### *5.2.3.1 Task 3.1*

One of the outcomes is shown here:


*Figure 6. Query With 10 Retrieved Images for Task 3.1*

After that, the system was put to the test by having it retrieve 10, 20, and 30 images. The curves are as follows, in order:

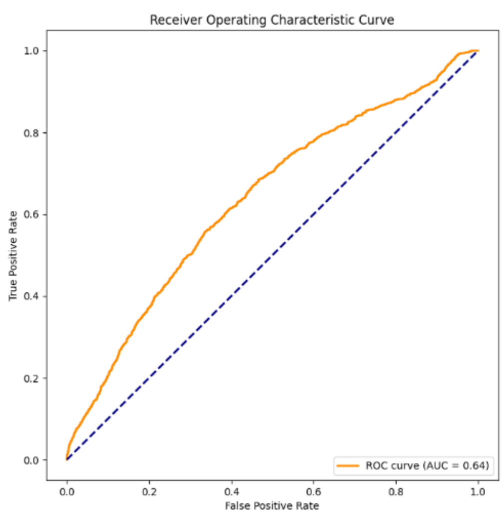


*Figure 7. ROC Curve for 10 Retrieved Images*

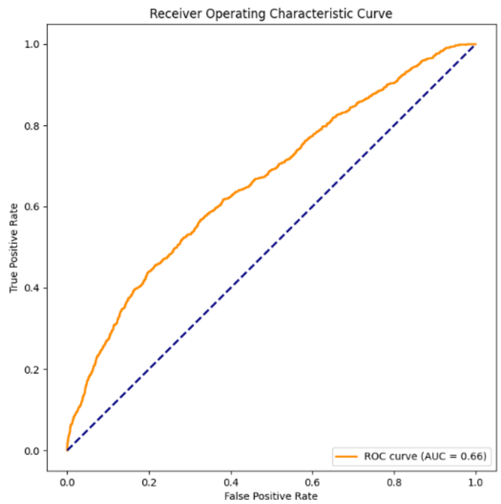


*Figure 8. ROC Curve for 20 Retrieved Images*

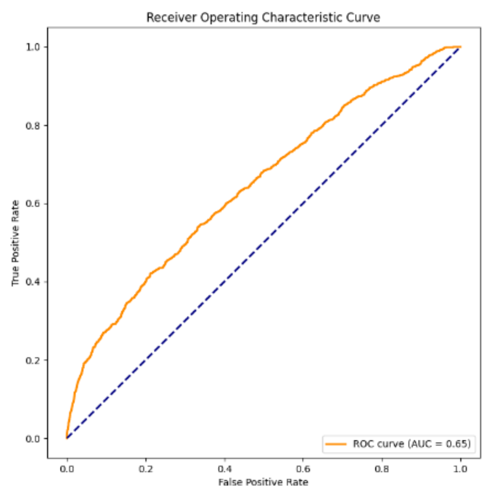


*Figure 9. ROC Curve for 30 Retrieved Images*

The outcomes of the trial showed how well the CBIR system retrieved pertinent photos. The system's discriminative capacity was further illustrated by the ROC curve, and its total performance was quantified by the Area Under the Curve (AUC).

##### *5.2.3.2 Task 3.2*

With weighted components for improved feature representation, the code given creates a Content-Based Image Retrieval (CBIR) system with Color Moments as distinguishing characteristics.

Color Moments in the Content-Based Image Retrieval (CBIR) system are shaped in large part by the weights that are included in the feature extraction process. The weights are applied to each RGB channel's mean, standard deviation, and skewness components in the code that is given. Users can choose how these features contribute to the total feature vector by giving them varying weights.

The code defines weights like [1.0, 0.5, 10.0], which are specifically used to modify the importance of each color moment. Each RGB channel is given a weighted mean, standard deviation, and skewness value. These weighted components are then concatenated to create the final feature vector.

The way the weights may highlight or underline certain facets of the Color Moments determines how they work. For example, skewness with a larger weight (10.0) indicates that skewness is seen more significant in terms of image differentiation. On the other hand, a smaller weight (like 0.5) for the standard deviation lessens its influence on the feature representation as a whole.

One of the outcomes is shown here:

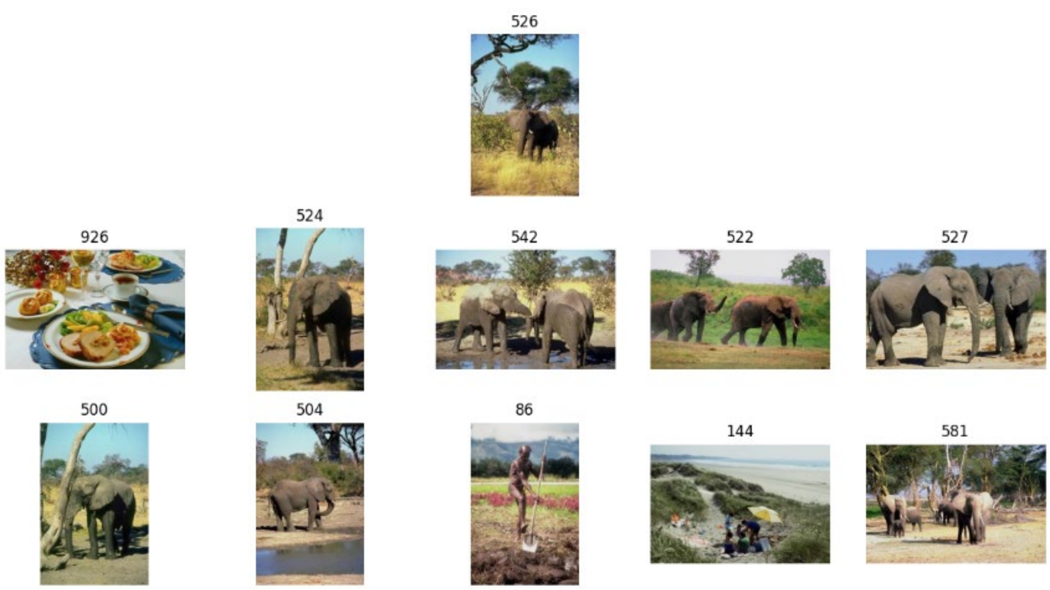


*Figure 10. Query With 10 Retrieved Images for Task 3.2*

After that, the system was put to the test by having it retrieve 10, 20, and 30 images. The curves are as follows, in order:

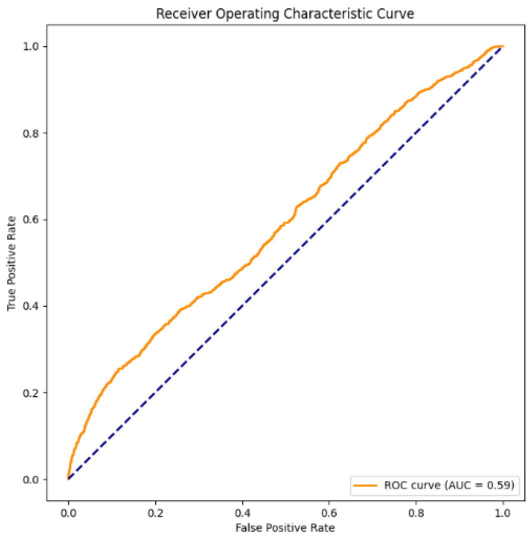


*Figure 11. ROC Curve for 10 Retrieved Images*

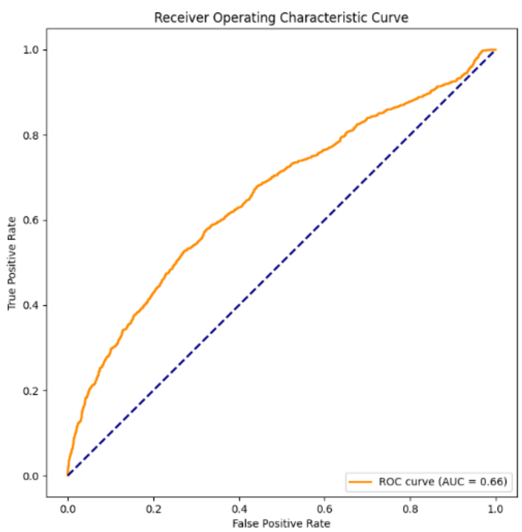


*Figure 12. ROC Curve for 20 Retrieved Images*

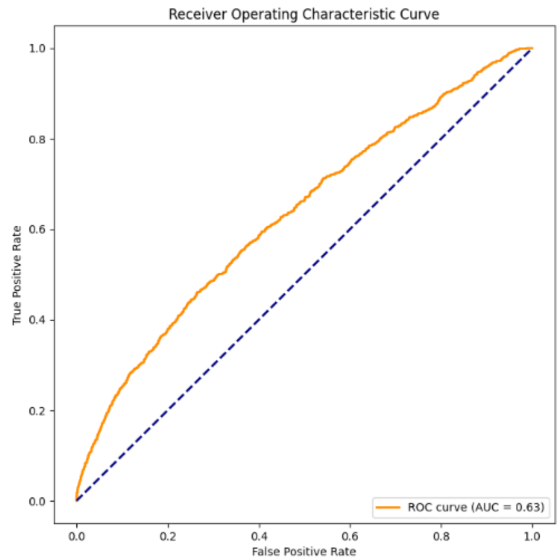


*Figure 13. ROC Curve for 30 Retrieved Images*

#### 5.2.3.3 Task 3.3

Similar to task 3.2, but additional moments, such as the median, mode, and kurtosis, were included.

One of the outcomes is shown here:

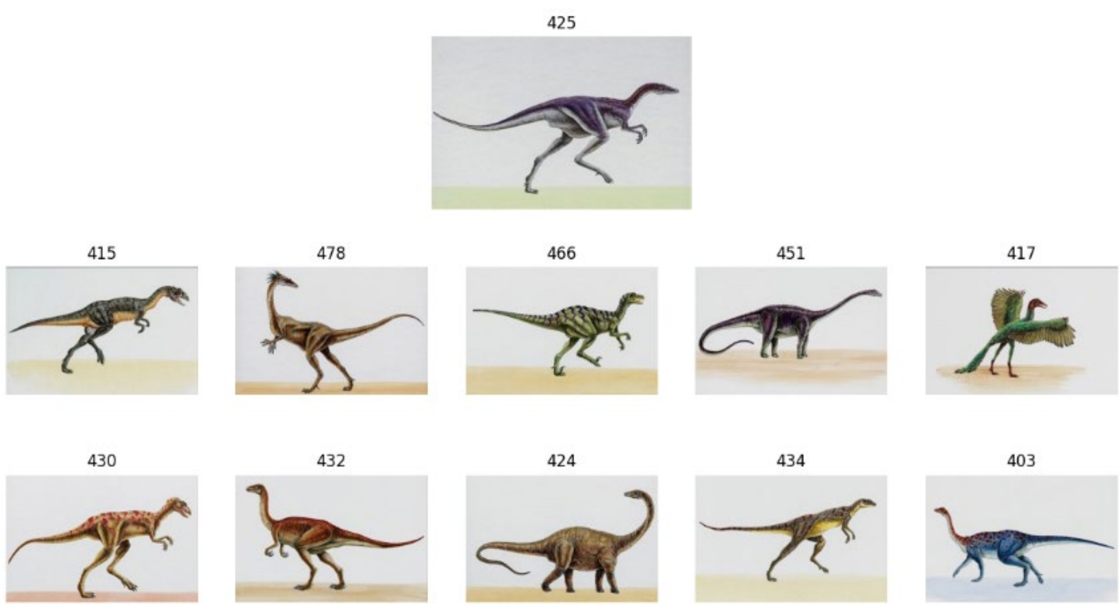


*Figure 14. Query With 10 Retrieved Images for Task 3.3*

After that, the system was put to the test by having it retrieve 10, 20, and 30 images. The curves are as follows, in order:

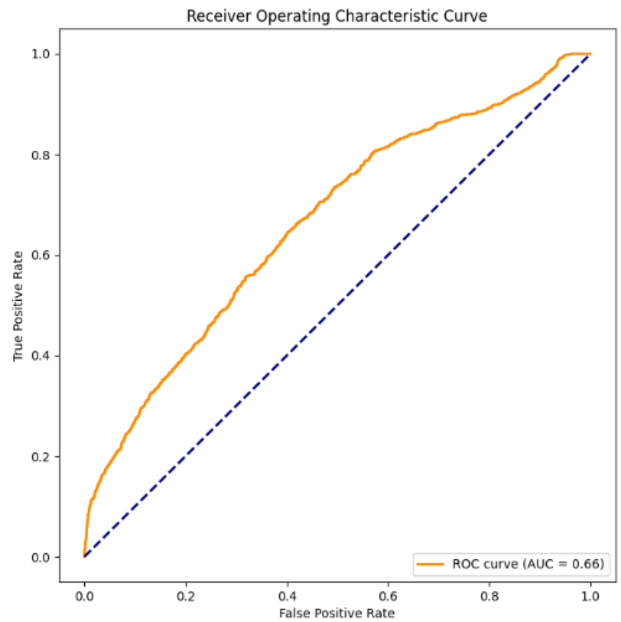


*Figure 15. ROC Curve for 10 Retrieved Images*

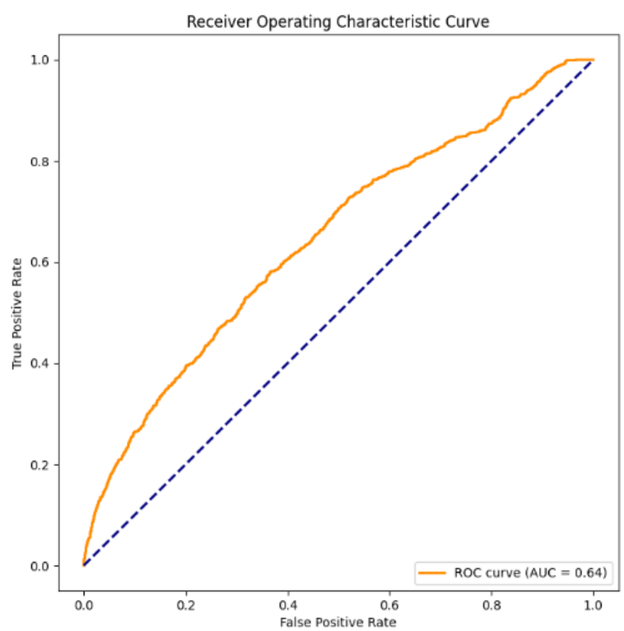


*Figure 16. ROC Curve for 20 Retrieved Images*

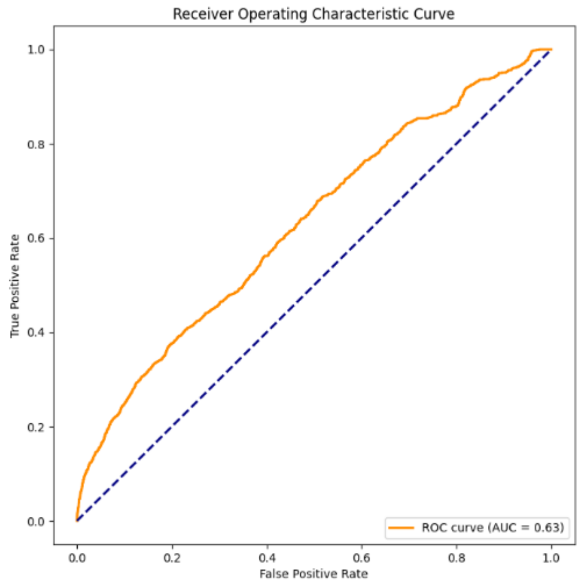


*Figure 17. ROC Curve for 30 Retrieved Images*

Then the weight of the mean was changed from 1 to 10:

```
weights = [10.0, 0.5, 1.0, 2.0, 3.0, 5.0] # Adjust the weights based on importance
```

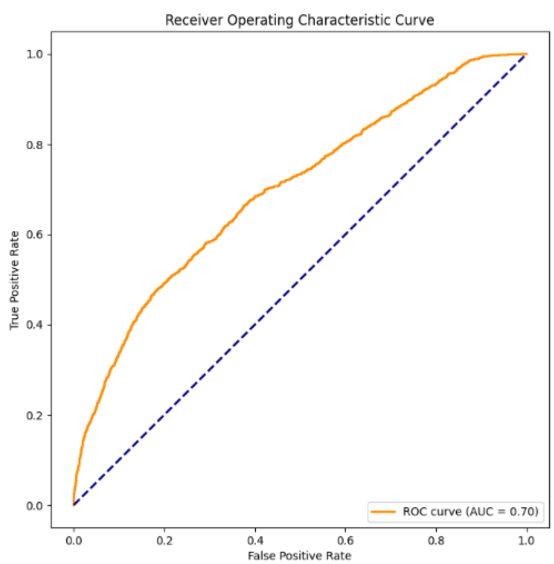


*Figure 18. ROC Curve for 10 Retrieved Images After Changing the Weight of Mean*

### 5.2.4 Limitations and Future Improvements

#### 5.2.4.1 Limitations

- **Limited Feature Set:** The only feature descriptors used by the algorithm for each channel (R, G, and B) are color moments (mean, standard deviation, and skewness). Although useful in some situations, this small feature set might not be able to fully capture intricate patterns, textures, or structural details seen in photos, which could result in retrieval accuracy issues, particularly when dealing with different datasets.
- **Sensitivity of the Weighting Scheme:** To apply varying weights to the calculated moments, a weighting scheme is included in Task 3.2, the expanded version of the system. The features of the dataset and the relative significance of each instant have a significant impact on how successful this weighting technique is. Retrieval performance may suffer from an inefficient weighting scheme, and determining the ideal set of weights may be difficult and dataset-dependent.
- **Scalability Issues:** Because of its linear search technique, the system's performance may suffer when handling bigger datasets or more complicated queries. The retrieval time may become a limiting issue when the number of photos rises. To overcome scalability concerns, more effective indexing or retrieval techniques may need to be investigated.

#### 5.2.4.2 Potential Improvements for Future Work

- **Feature Diversity:** By include more features than only color moments, the system will be better able to capture a wider range of picture attributes. Integrating deep learning-based features, texture features, or shape descriptors can yield a more thorough representation and increase the system's accuracy over a larger variety of picture formats.
- **Indexing and Retrieval Optimization:** Using more sophisticated indexing and retrieval methods, including tree-based structures or locality-sensitive hashing (LSH), may greatly increase system performance, especially when working with big datasets. These enhancements can improve scalability and lessen the retrieval process's temporal complexity.

# 6 Task 4

## 6.1 System Implementation

### 6.1.1 General Architecture and Functionality of the Modules

- **System BoVW_LBP Initialization (CBIRSystem):**
  - **Goal:** Configures LBP (Local Binary Pattern) and BoVW (Bag of Visual Words) features in the CBIR system.
  - **Functionality:**
    - Sets the cluster size and dataset path during system startup.
    - Assigns labels depending on category after extracting LBP features for every image in the collection.
- **Feature Extraction (extract_lbp_features, extract_bovw_lbp_features):**
  - **Goal:** Extract LBP characteristics from pictures.
  - **Functionality:**
    - **Extract_lbp_features:** This function takes a picture, normalizes the histogram, computes LBP, and transforms it to grayscale.
    - **Extract_bovw_lbp_features:** Assigns labels after iterating through pictures and extracting LBP features.
- **Distance Calculation and Ranking (compute_distance, rank_results):**
  - **Goal:** Order the results based on the distance measured between the query and database photos.
  - **Functionality:**
    - **Compute_distance:** Determines the Euclidean separation between the database pictures and the query's LBP histograms.
    - **Rank_results:** Removes the query image from the ranking list after sorting the distances.
- **Display Results (display_results):**
  - **Goal:** Shows the top-ranked retrieval results and the query picture.
  - **Functionality:** Plots the top 10 retrieval results in the second and third rows, and the query picture in the first.

- **Performance Evaluation (evaluate_performance):**
    - **Goal:** Assesses the system's F1 score, recall, and accuracy.
    - **Functionality:**
        - Calculates true positives, false positives, true negatives, and false negatives.
        - Determines the F1 score, precision, and recall by using the recovered photos as a basis.
- **Experiment Execution (run_experiment_bovw_lbp):**
    - **Goal:** Carry out tests and assess the CBIR system's functionality.
    - **Functionality:**
        - Generates a CBIRSystemBoVW_LBP instance.
        - Iteratively searches through query photos, retrieves information, and assesses performance indicators.
        - Determines the time, F1 score, average accuracy, and recall.
        - ROC curve and average metrics are displayed.

### 6.1.2 Implementation Details

The Content-Based Image Retrieval (CBIR) system is implemented in Python with the use of a number of modules and frameworks to make certain jobs easier. NumPy is used for numerical computations on arrays; Scikit-learn is used to compute metrics like AUC, ROC curve, and clustering; Matplotlib is used to visualize metrics and ROC curves; and Scikit-image is used to extract local binary pattern features from images. OpenCV is used for image processing operations. The CBIR system's strong functionality and effectiveness are largely due to these libraries, which offer vital capabilities for metric calculation, image analysis, numerical processing, and visualization.

### 6.1.3 Functionality Overview

The CBIR system makes effective use of a number of modules and functions to retrieve images. The CBIR system is first configured with Bag of Visual Words (BoVW) and Local Binary Pattern (LBP) features using the Initialization module (CBIRSystemBoVW_LBP). Subsequently, each picture in the dataset is subjected to LBP features extraction, which involves normalizing histograms and allocating labels according to categories. The Euclidean distance between the query and database image LBP histograms is determined by the Compute Distance module. The ranked list is sorted and returned by the Rank Results module, with the query picture removed. The query picture and the top-ranked retrieval results are shown visually by the Display Results module.

Precision, recall, and F1 score are evaluated in the Evaluate Performance module while true positives, false positives, true negatives, and false negatives are taken into account. Lastly, tests are carried out, performance metrics are assessed, and average metrics are shown with a Receiver Operating Characteristic (ROC) curve by the Run Experiment module.

## 6.2 Experimental Setup and Results

### 6.2.1 Evaluation Metrics

The f1 score, recall, and precision are calculated as Task 2 and Task 3.

### 6.2.2 Experimental Setup

Specified cluster size is used to establish the BoVW-LBP CBIR system. Within each category, query photos are randomly selected for experimentation. For every experiment, the retrieval time is assessed.

### 6.2.3 Results and Analysis

The average precision, recall and F1-score were calculated over the experiments and presented. Additionally, a Receiver Operating Characteristic (ROC) curve was plotted to visualize the trade-off between a false positive rate and a true positive rate. The area under the ROC curve (AUC) is provided as a summary metric. One of the outcomes is shown here:

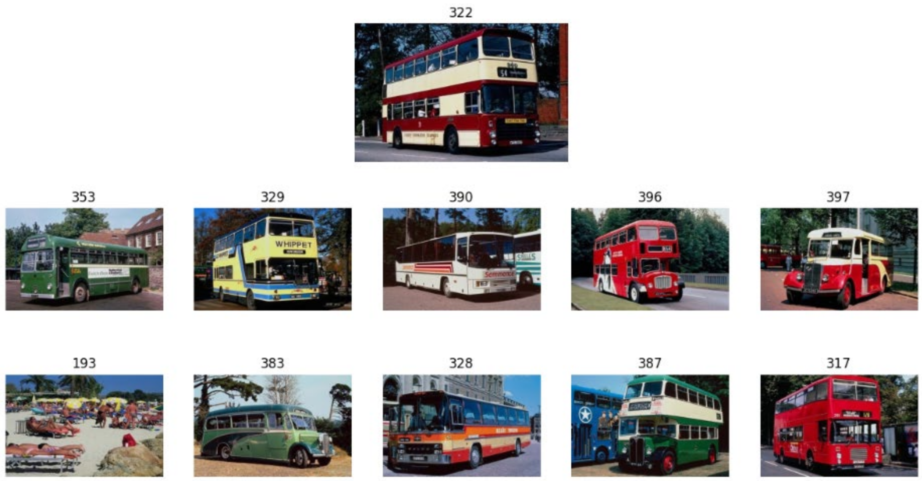


*Figure 19. Query With 10 Retrieved Images for Task 4*

After that, the system was put to the test by having it retrieve 10, 20, and 30 images. The curves are as follows, in order:

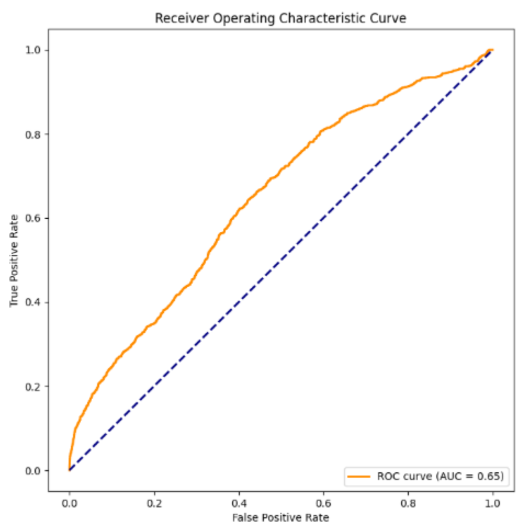


*Figure 20. ROC Curve For 10 Retrieved Images*

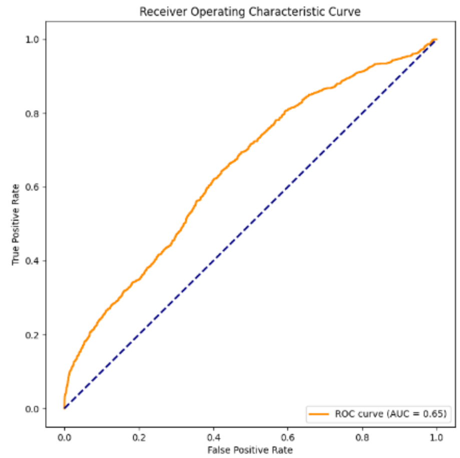


*Figure 21. ROC Curve For 20 Retrieved Images*

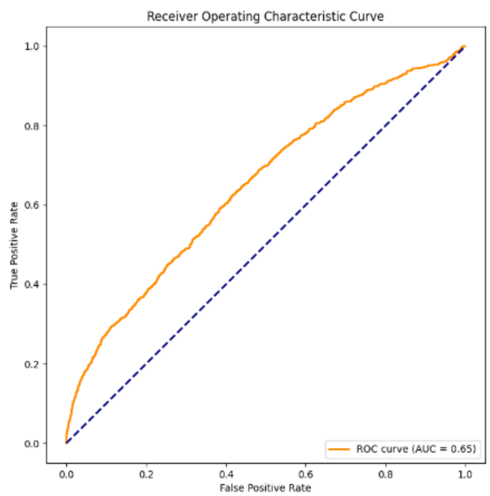


*Figure 22. ROC Curve For 30 Retrieved Images*

### 6.2.4 Limitations and Future Improvements

Potential biases in the dataset, reliance on particular parameters (such as cluster size), and the usage of a single feature combination are some of the drawbacks of the existing approach. Further research endeavors may encompass broadening the range of datasets, refining parameters, and investigating supplementary feature combinations to augment retrieval precision and efficacy. For better feature representation, the integration of sophisticated deep-learning models may also be taken into consideration.

# 7 Conclusion

## 7.1 Key Findings and Achievements

In summary, a Content-Based Image Retrieval (CBIR) system that makes use of color features—more especially, color histograms and color moments—has been successfully constructed in this assignment. The system demonstrated the efficacy of the chosen color descriptors by showcasing noteworthy successes in picture retrieval based on visual information.

## 7.2 Overall Effectiveness of Color Features

When characterizing picture content, the use of color features—specifically, color histograms and color moments—proved to be reliable and effective. Color moments provided a level of complexity by integrating statistical measurements, whilst color histograms provided a succinct depiction of color distributions. This combination emphasized the usefulness of color characteristics in CBIR tasks by improving the precision and accuracy of picture retrieval.

## 7.3 Insights and Recommendations

Even though the assignment produced encouraging findings, there is still much to be learned about CBIR research and development. A possible research direction is to adjust the weighting techniques used for the color moments. The weights attributed to skewness, mean, and standard deviation are important, and more research might determine the best weight combinations for various datasets or picture kinds.

For further applications, it is also necessary to address the CBIR system's scalability and flexibility. Examining the system's functionality with more extensive datasets and a range of picture classifications might yield important information about its practicality. Furthermore, investigating the use of machine learning methods to dynamically modify feature weights according to the dataset's properties may improve the CBIR system's flexibility.