Ain Shams University
Faculty of Engineering
Computers and Systems Department
4th Year, Electrical Engineering

CSE412: Selected Topics in Computer Engineering

CBIR/CBVR Project Report

Student Name	Student ID
Khaled Mohamed	1600506
Omar Abd-El-Aziz	1600885
Mahmoud Osama Mahmoud Soliman	1601294
Mahmoud Swilam	1601306
Marwan Mahmoud Badawy	1601346
Mostafa Amgad Ahmed Saeed	1601401
Mostafa Gamal Zakaria	1601405
Mohamed Mahmoud Abdallah Al Ansary	1601269

Cairo, 2021

Introduction

Multimedia Retrieval is one of the most important and fastest growing research areas in the field of Multimedia technology, The large and growing amount of digital data, and the development of the Internet highlight the need to develop sophisticated access methods that provide more than just simple text-based queries. Many programs have been developed with complex mathematics algorithms to allow the transformation of image or video data in a way that enhances searching accuracy. However it becomes difficult when dealing with large sets of multimedia data.

Table of Contents

List of Figures

Detailed project description

Out desktop application is a multi-feature content-based multimedia retrieval engine. It is capable of retrieving images and video sequences that look similar. Besides providing multiple common and state of the art retrieval mechanisms based on mean color, histogram and color grid.

We started by designing our system which will include interface with the user , the backend algorithms and the database which we will use in our application, then we chose the programming language that we will use which is java as it has a lot of helper classes, the interface is a desktop application implemented by JavaFx, the database is local database we chose MySQL as query language and at last we choose the 3 techniques for image mean color ,histogram and color grid and 1 technique for video which is Naïve video similarity(NVS).

Beneficiaries of the project

- Dermatologists use is to display a set of visually similar images with a pathology-confirmed diagnosis for a given query skin image.

- The project helps physicians, patients, and other users make trustworthy and accurate classifications of skin diseases based on visually similar cases.

- CBIR algorithms using machine learning with high accuracy rates

- CBVR can help policemen capture robbers by searching their faces against criminal databases from surveillance videos.

- Other domains for CBIR Applications:
*Architectural and engineering design
*Art collections
*Crime prevention
*Geographical information and remote sensing systems
*Intellectual property
*Medical diagnosis
*Military

Detailed analysis

Kind: Class

Name: InsertImage

Scope: Public

Description: used to extract features from the image like mean color, Histogram and color layout and insert the image in the database.

Kind: Class

Name: InsertVideo

Scope: Public

Description: used to extract key frames based on sequential comparison from the video and extract features from the key frames like mean color, Histogram and color layout and insert the Video and the key frames in the database.

Kind: Class

Name: SearchImage

Scope: Public

Description: used to retrieve images from the database based on the visual features like mean color, histogram and color layout.

Kind: Class

Name: SearchVideo

Scope: Public

Description: used to retrieve Video from the database based on Naïve video similarity technique in which we compare the two key frames sequence and find at least one similar frame then divide the number of similar frames over the total number of frames.

Kind: Function

Name: run

Class: InsertImage

Scope: Public

Input: String args,

Connection conn,

int id,

String title,

String info

Output: void

Description: this function is used to extract features and insert the image by url and its features

---

Kind: Function

Name: run

Class: InsertVideo

Scope: Public

Input: String args,

Connection conn,

int id,

String title,

String info

Output: void

Description: this function is used to extract key frames, extract from them the visual features and then insert the image by url, key frames and features.

Kind: Function

Name: mean

Class: SearchImage

Scope: Public

Input: String args,

Connection conn,

Output: image url

Description: this function is used to retrieve image from the database based on color mean.

Kind: Function

Name: grid

Class: SearchImage

Scope: Public

Input: String args,

Connection conn,

int width,

int height

Output: image url

Description: this function is used to retrieve image from the database based on color grid.

Kind: Function

Name: Hist

Class: SearchImage

Scope: Public

Input: String args,

Connection conn,

float compare

Output: image url

Description: this function is used to retrieve image from the database based on histogram.

Kind: Function

Name: Search

Class: SearchVideo

Scope: Public

Input: String args,

Connection conn,

Output: Video url

Description: this function is used to retrieve Video from the database based on key frame extraction and comparing the two key frame sequences.

Detailed description of the adopted techniques

Image:

Mean color:

It is general primitive technique, in which we extract pixel color information for each component in which mean color of component(R, G, B) =sum of that components for all pixels /number of pixels

Histogram:

It is general primitive technique, in which we extract the histogram of the image the histogram is the frequency count of each component, it is most common used feature for representation.

Color layout:

It is general semantic technique, in which we divide the whole image into sub blocks then extract the features from each block individually.

Video:

Key frame extraction by using sequential comparison:

Extracting key frames for action recognition purposes is a challenging task, as compression is to be achieved without losing the impression of the action. Insufficient or wrong key frames can lead to confusion in the type of action. Proposed two-level key frame extraction algorithm uses an adaptive threshold technique to identify most dissimilar frames as key frames. At the first level, global features based on intensity histogram and at second level local features computed from wavelet decomposition are used as similarity measures.

To tackle the problem of large memory space required for storing the videos, video summarization or key frame extraction techniques are used, Video summarization is a technique which represents the video with a fewer number of salient frames. Frames carrying maximum information are identified and stored as key frames.

•Each frame is compared to the KF that has been extracted previously, when the differences is high, then this frame is indicated as a new KF.

•A new key frame can be extracted by computing the differences between the color histogram of the current frame and the previous key frame.

Algorithm:

1. Read the test video and convert the frames to grey scale.

2. Read $Framei$ and $Framei$+1

3. Find intensity histogram $Hisi$ and $Hisi$+1 for frames read in step 2.

4. Compute the absolute difference between $Hisi$ and $Hisi$+1

5. Calculate the sum of differences obtained over all the bins of histogram and store as Histogram coefficients.

6. Find the mean $\bar{x}$ and standard deviation σ of histograms coefficients obtained for all the frames.

7. Compute threshold value from mean and standard deviation values obtained in step 6 using Eq. (4)

$Th=k1*\bar{x}+k2*\sigma$ (4)

Where k1 and K2 are constants which are found empirically.

8. Compare the Histogram coefficient values of all the frames with the threshold value to identify most dissimilar frames as key frames.

Program Interface

User is able to select CBIR or CBVR when entering the Example query, inside he can further select the specific technique as shown in the User Guide.

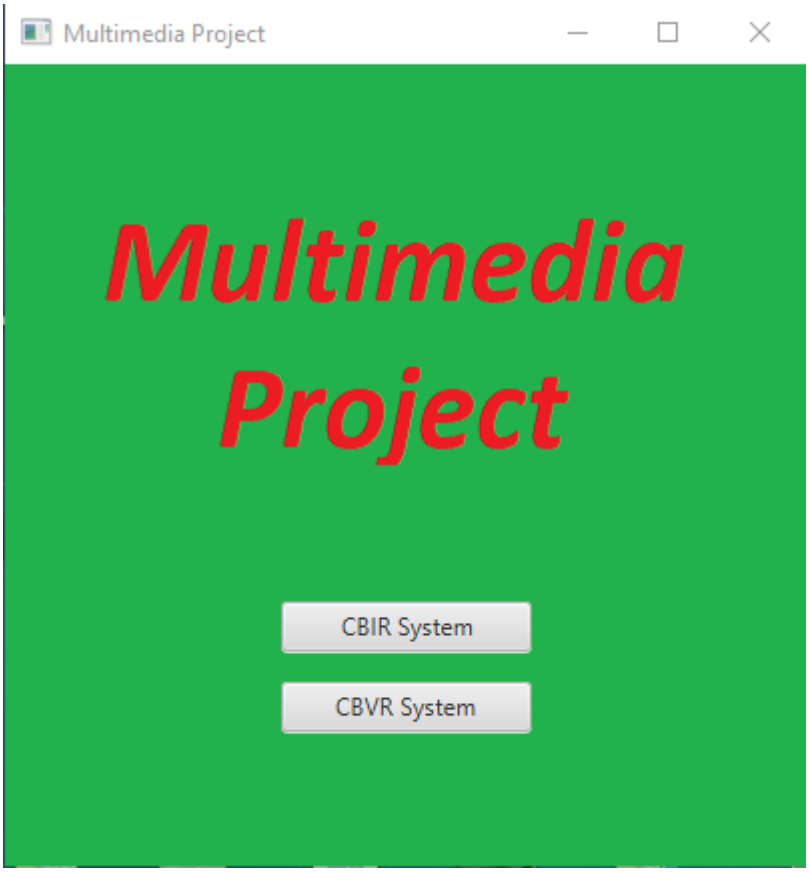

Figure 1: Program Interface

Task breakdown structure

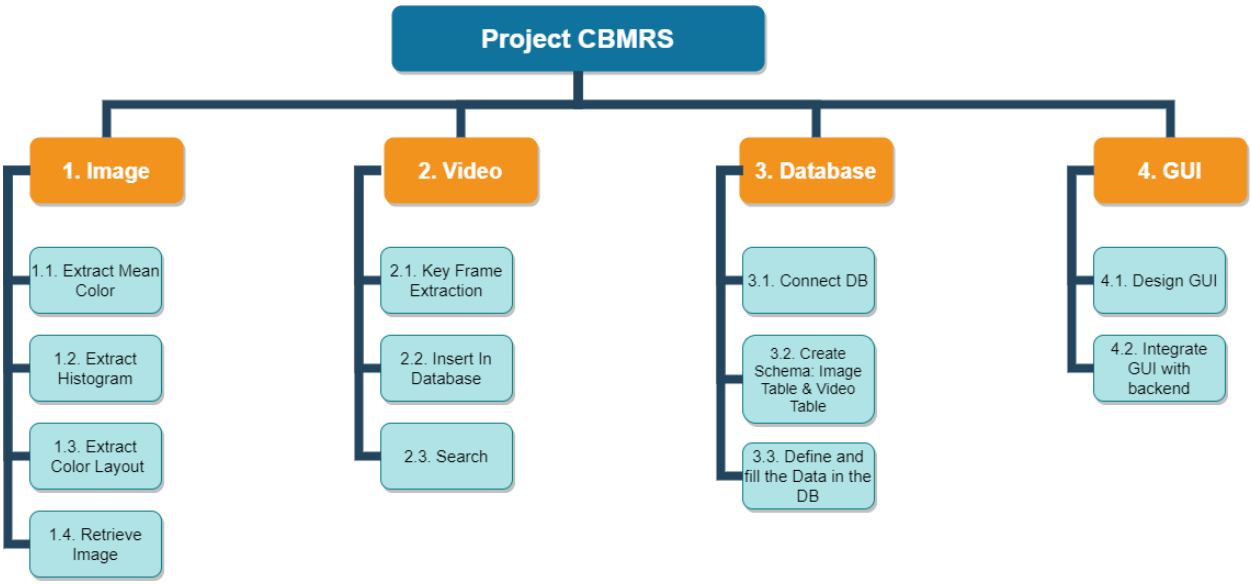

Figure 2: TBD

Time plan

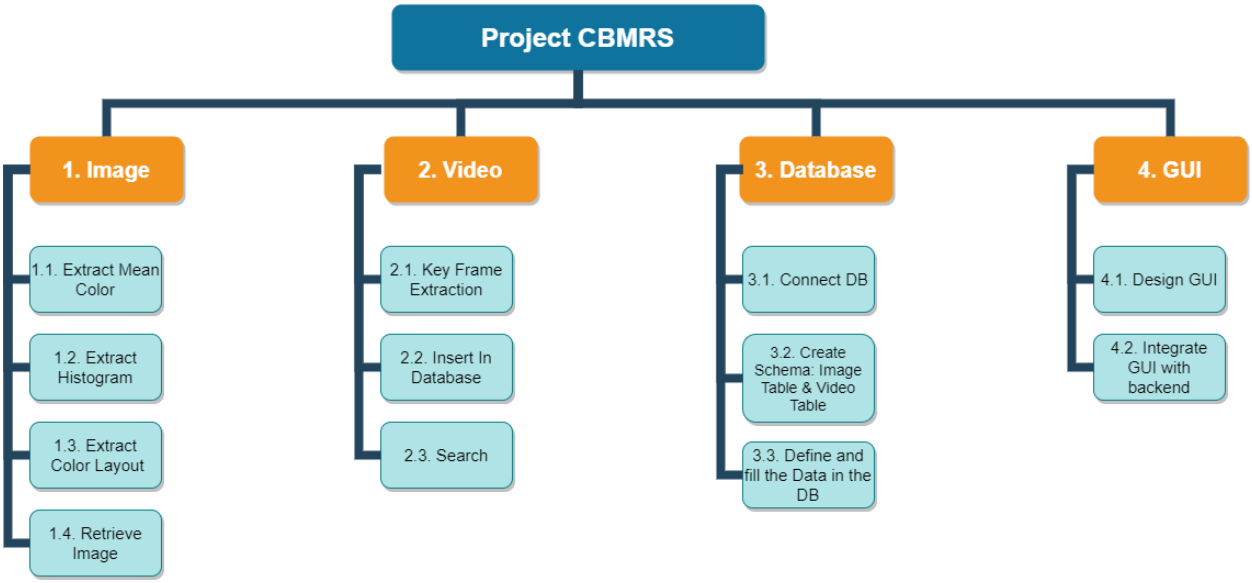

Figure 3: Gantt chart

System architecture

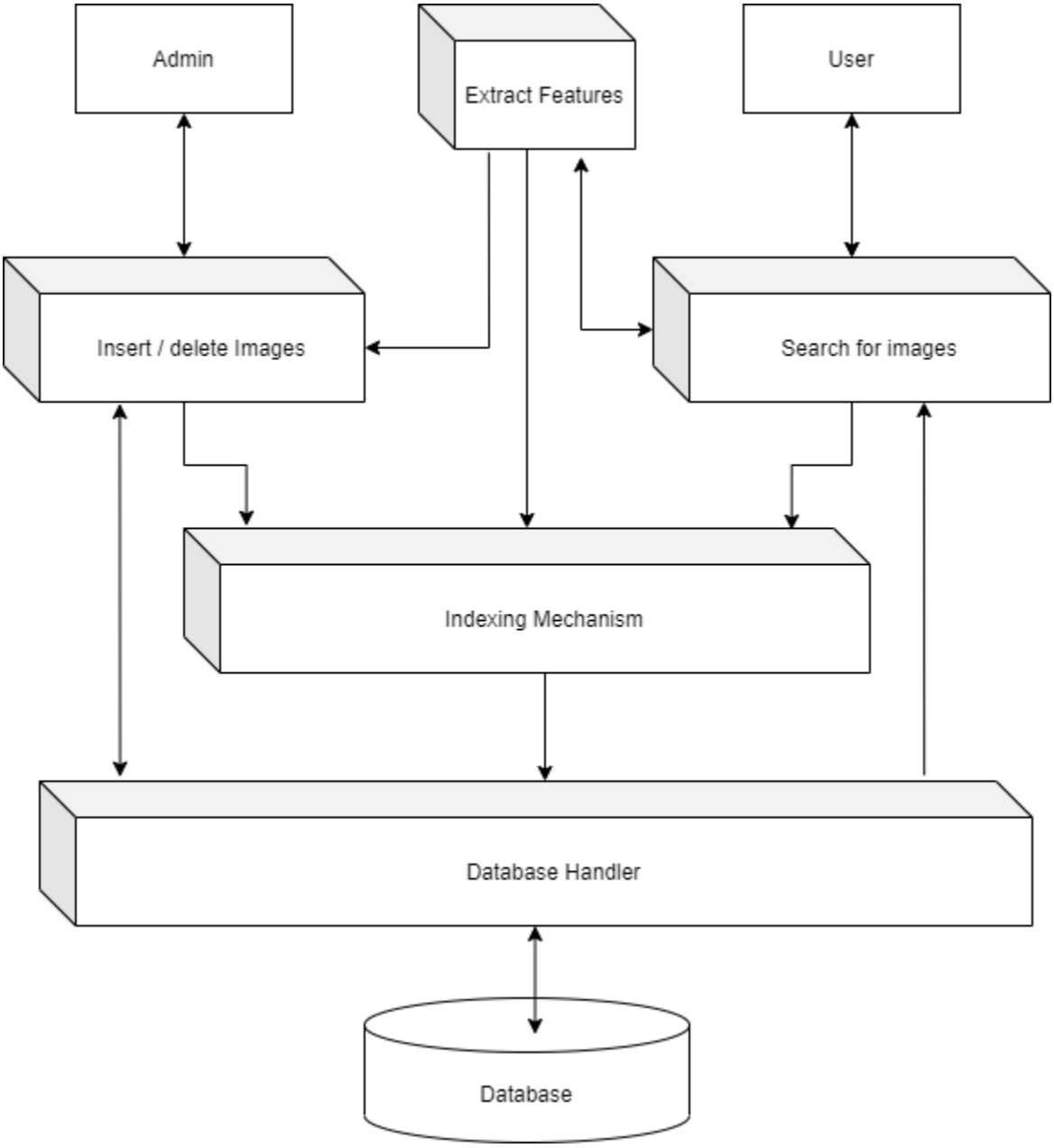

Figure 4: System Architecture

Multimedia Database Design

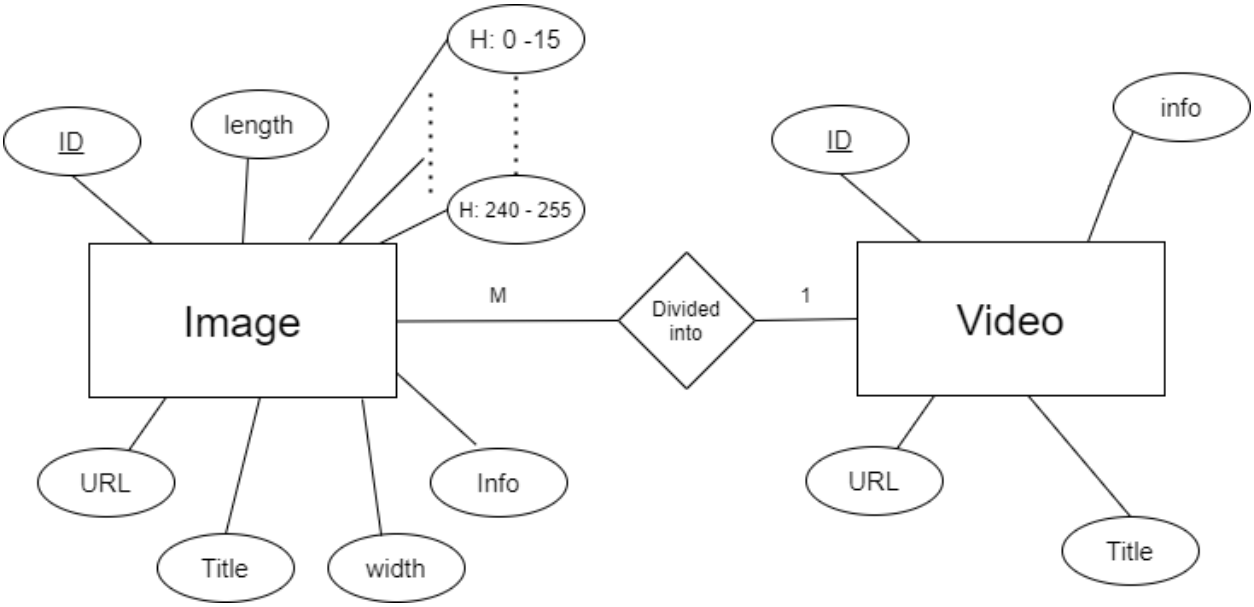

Figure 5: Database Design

System Design

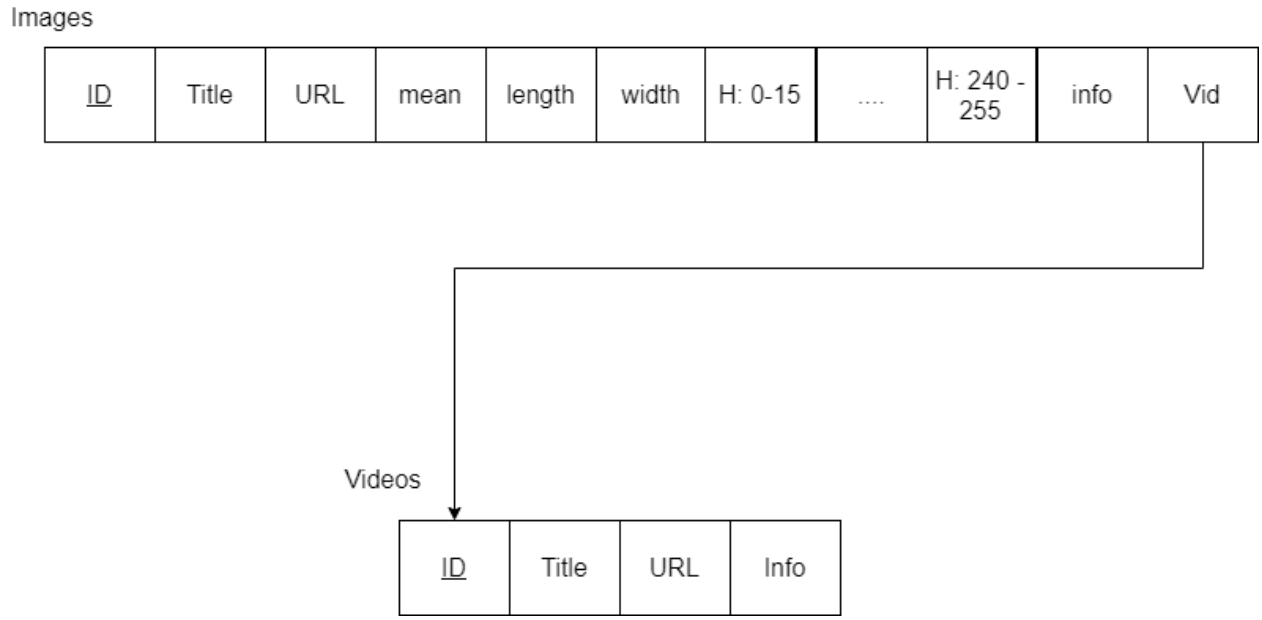

Figure 6: System Design

Testing scenarios and results

Testing CBIR System

Error handling in selecting image

If we didn't choose any image, it will give us an error

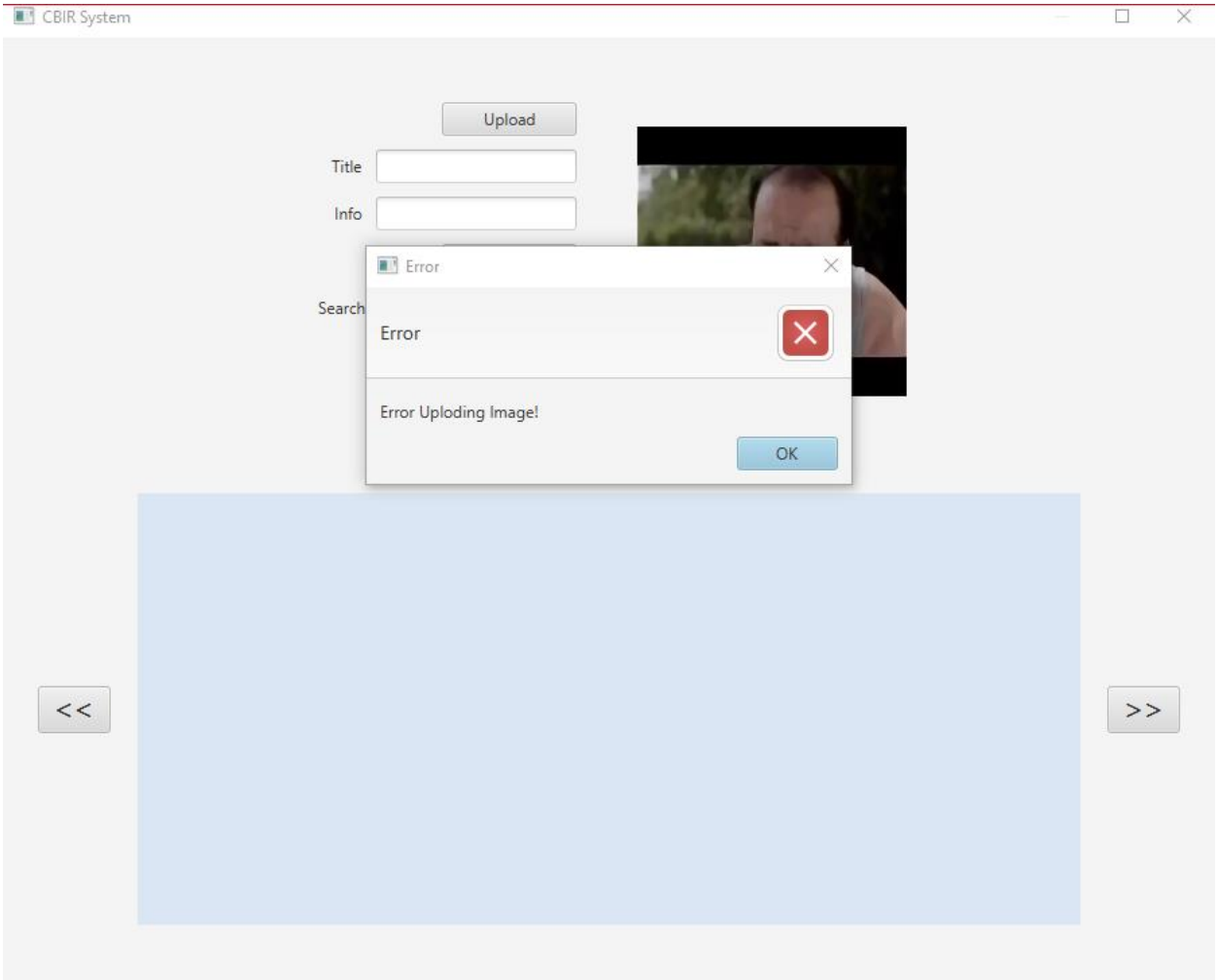

Figure 7: Error handling in selecting image

Error handling in adding image to database

If we didn't write any title for the image, it will give an error

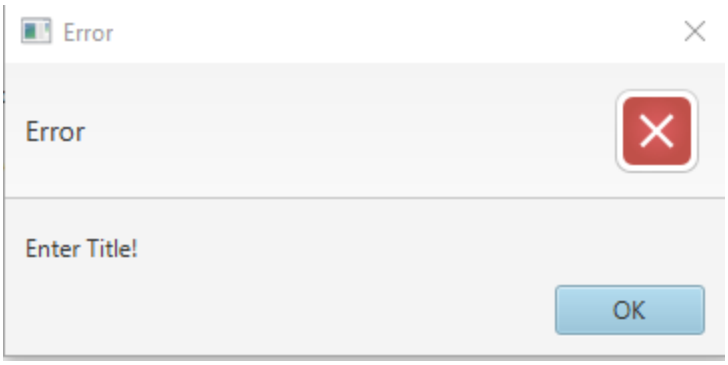

Figure 8: Error handling in adding image to database

Error handling in Histogram Search

For histogram search, if we didn't enter a number in the threshold, it gives an error

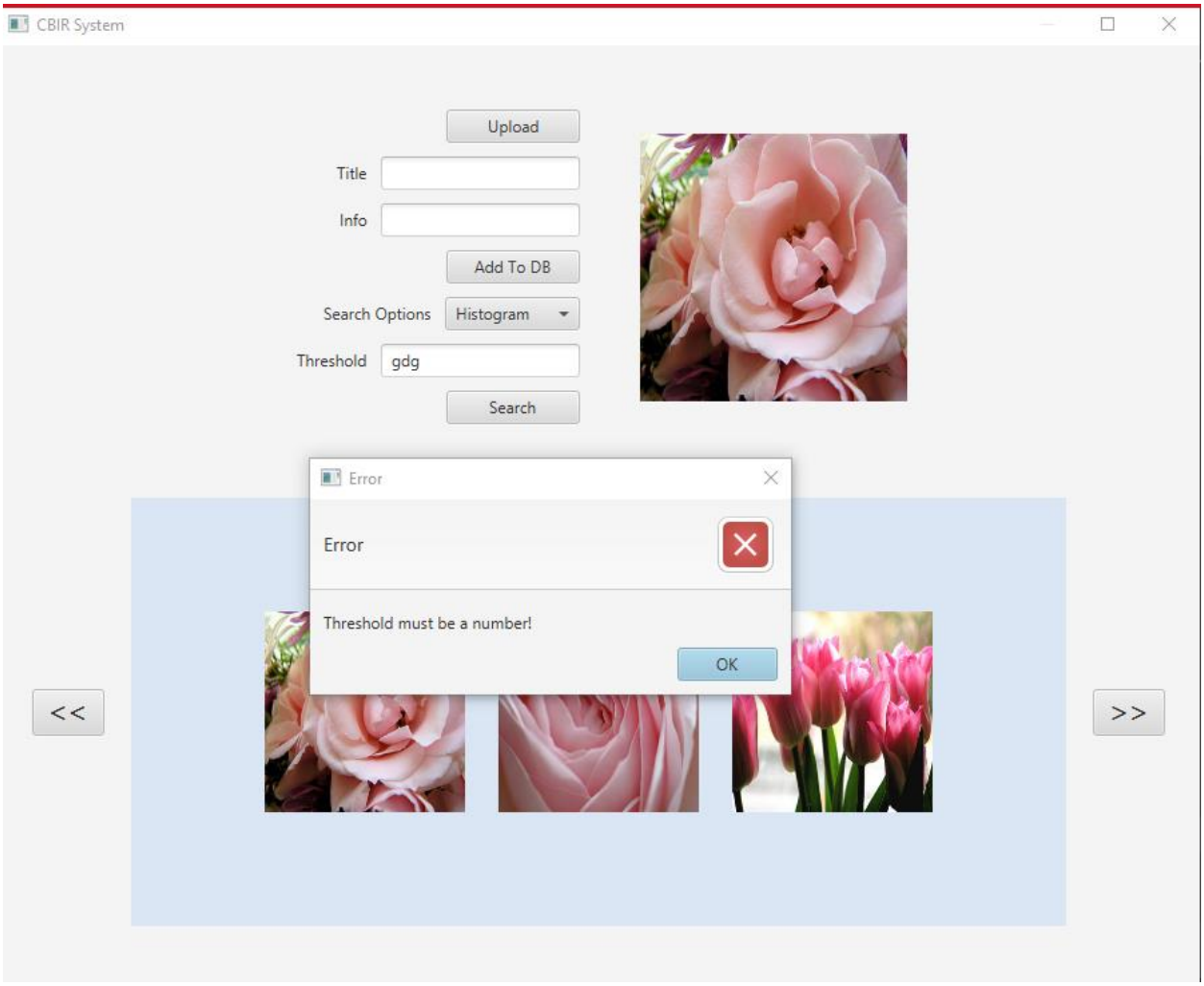

Figure 9: Error handling in Histogram Search

Error handling in Color Layout Search

For Color Layout Search, if we put a float number in the number of blocks lineEdit, it gives an error

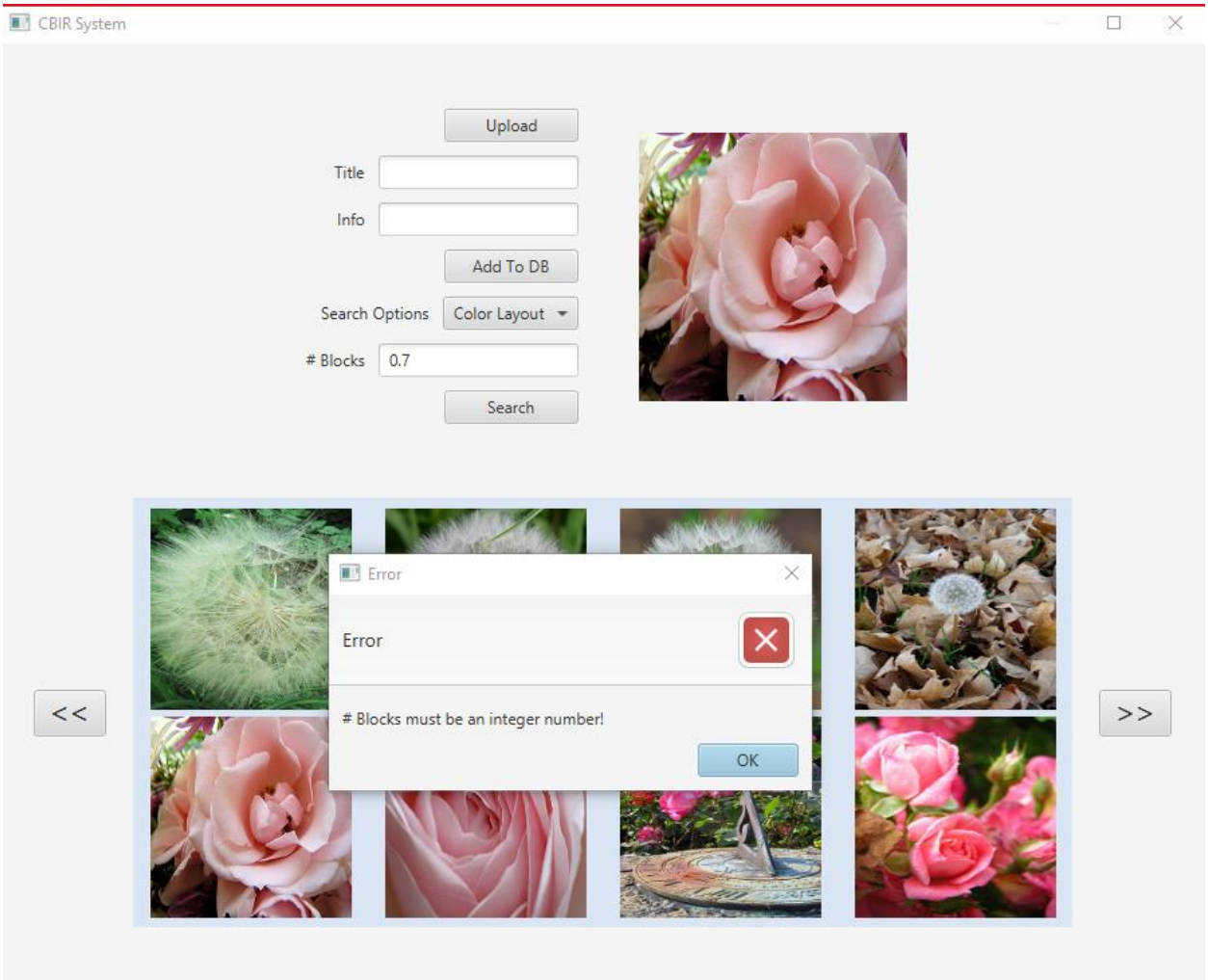

Figure 10: Error handling in Color Layout Search

Testing CBVR System

Error handling in selecting video

If we don't choose any video, it gives an error

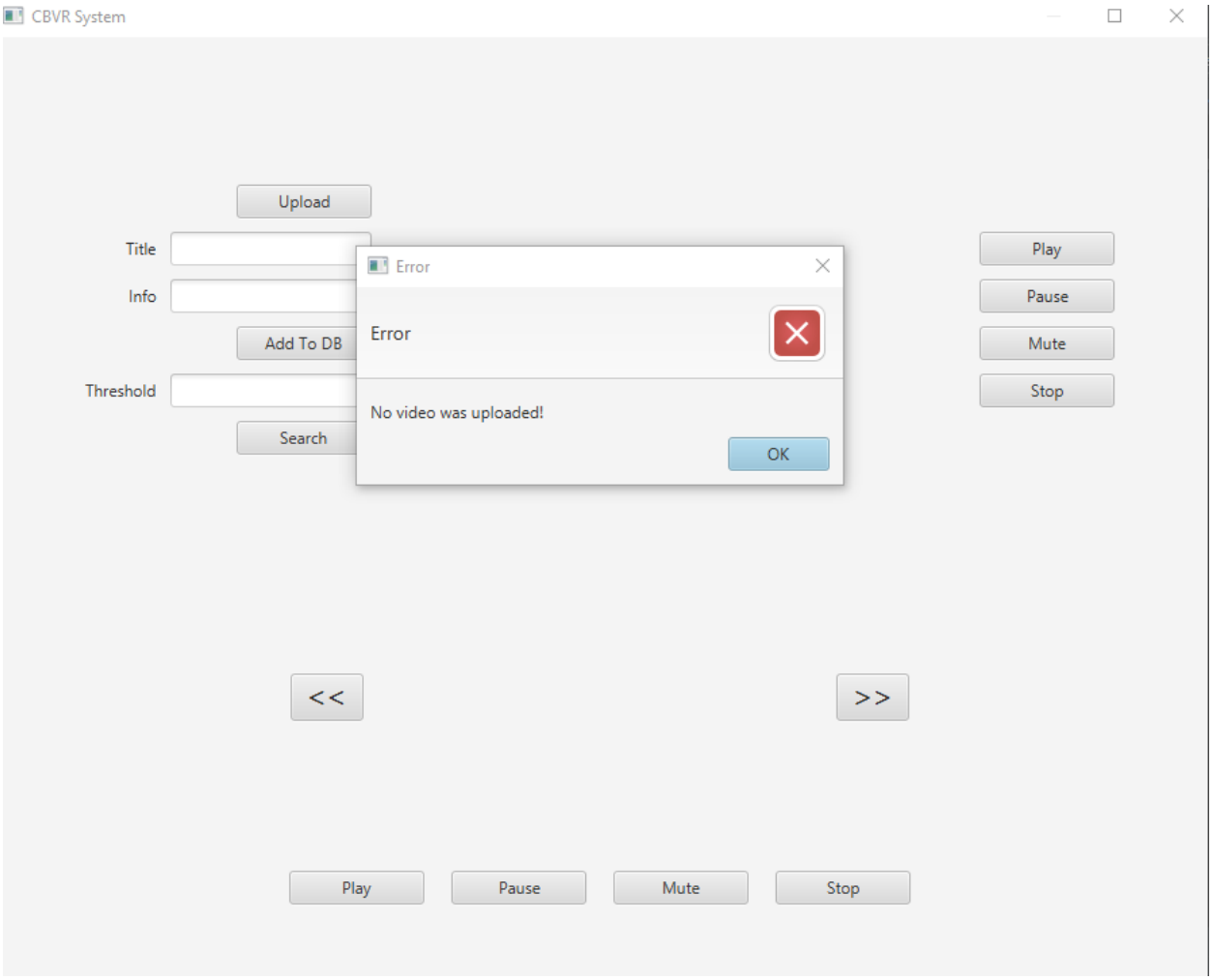

Figure 11: Error handling in selecting video

Error handling in adding video to database

If we didn't choose a video and clicked "Add To DB", it gives an error

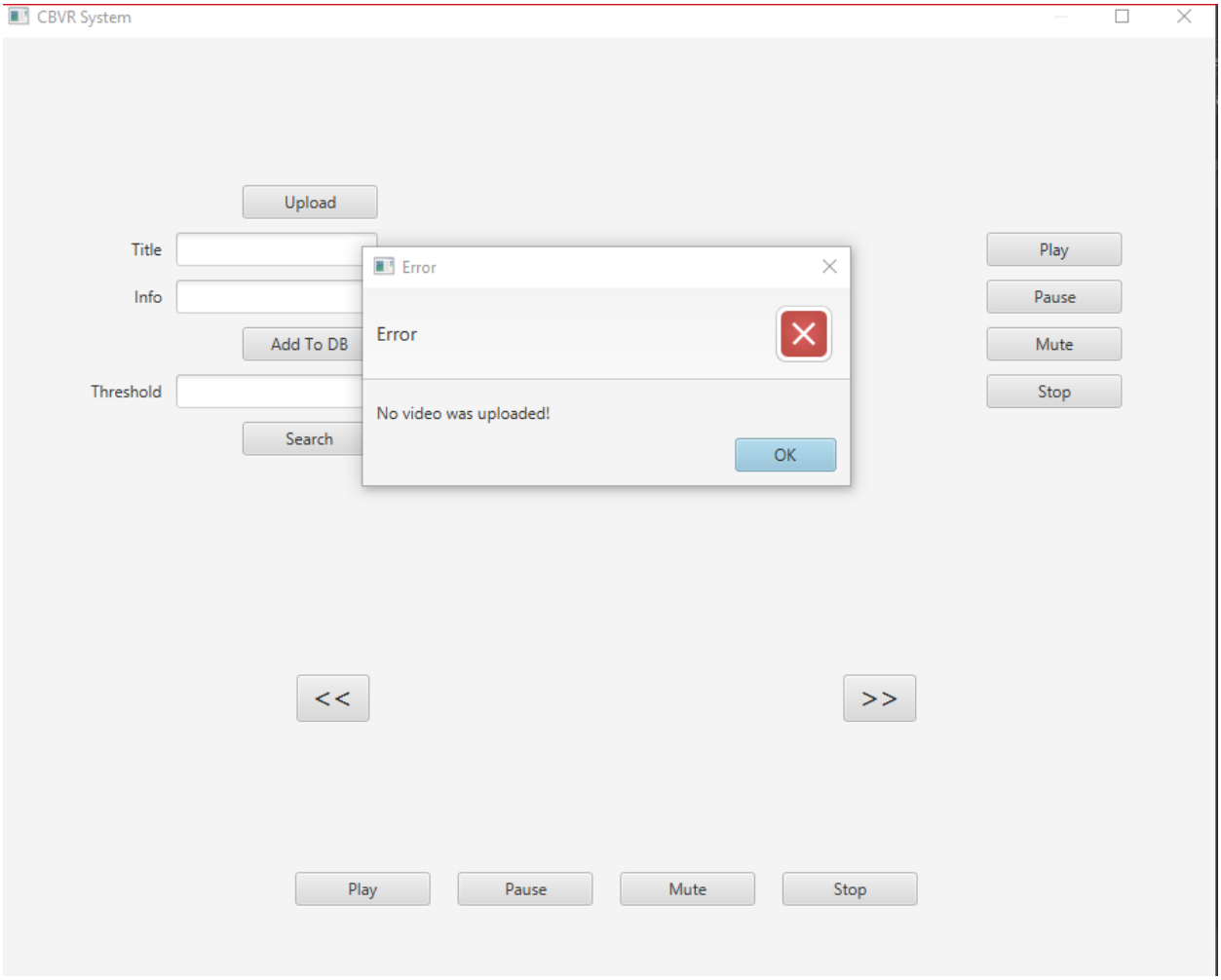

Figure 12: Error handling in adding video to database

Error handling in playing video

If we try to play a video while a video was not uploaded, it will give an error

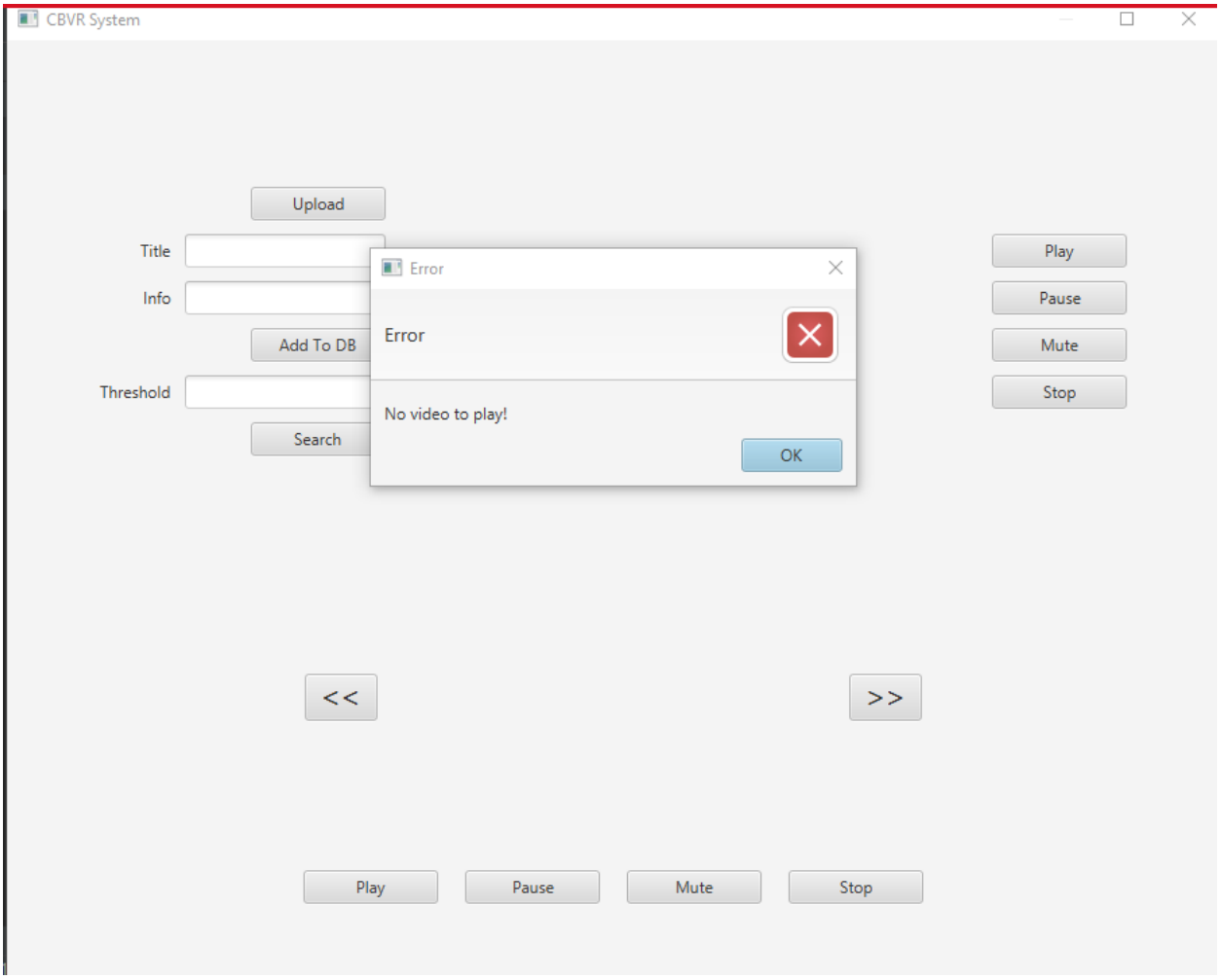

Figure 13: Error handling in playing video

Error handling in NVS Search

If we didn’t enter a number, it will give a threshold error

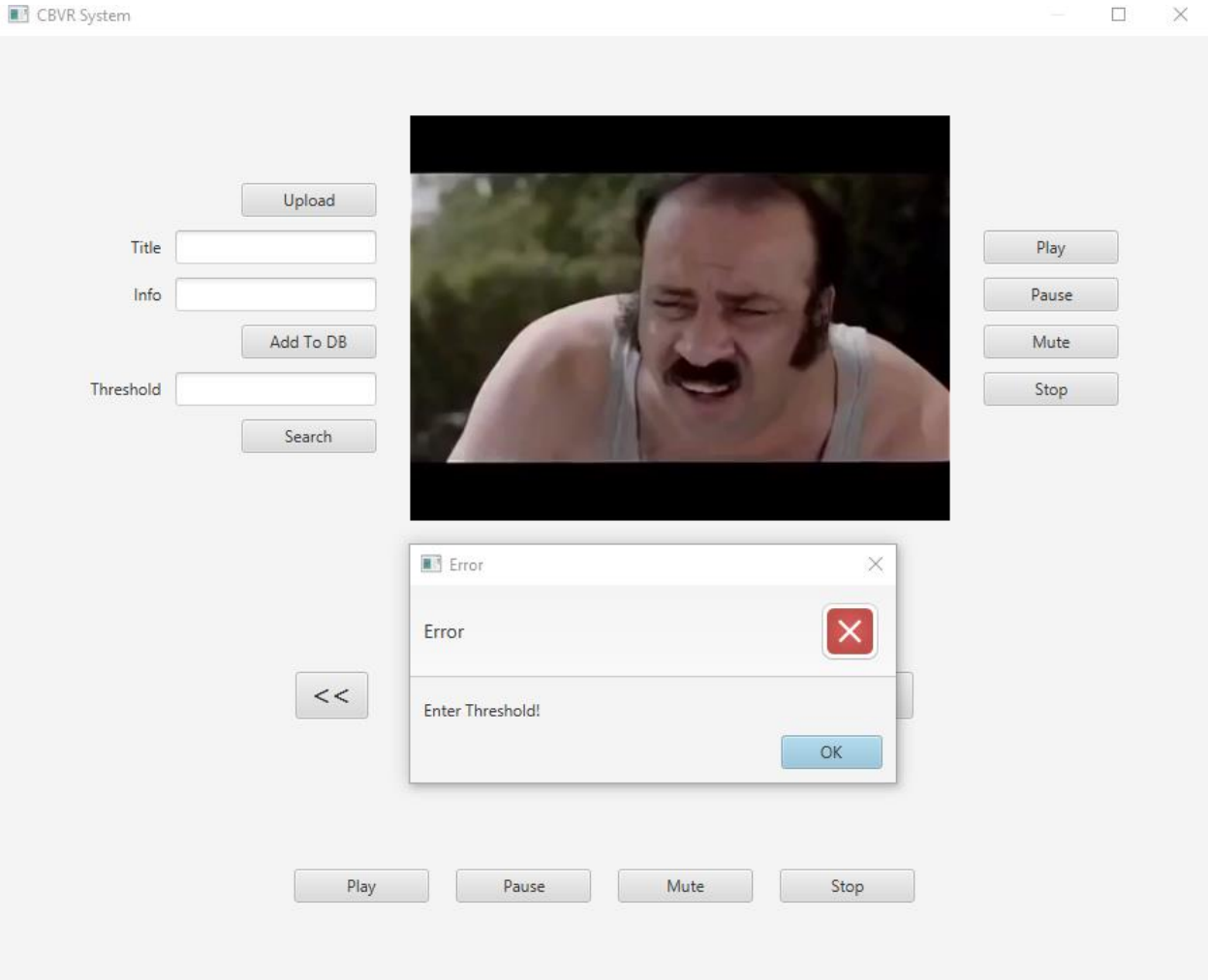

Figure 14: Error handling in NVS Search

End user guide

Interface of the CBIR system

Upload image

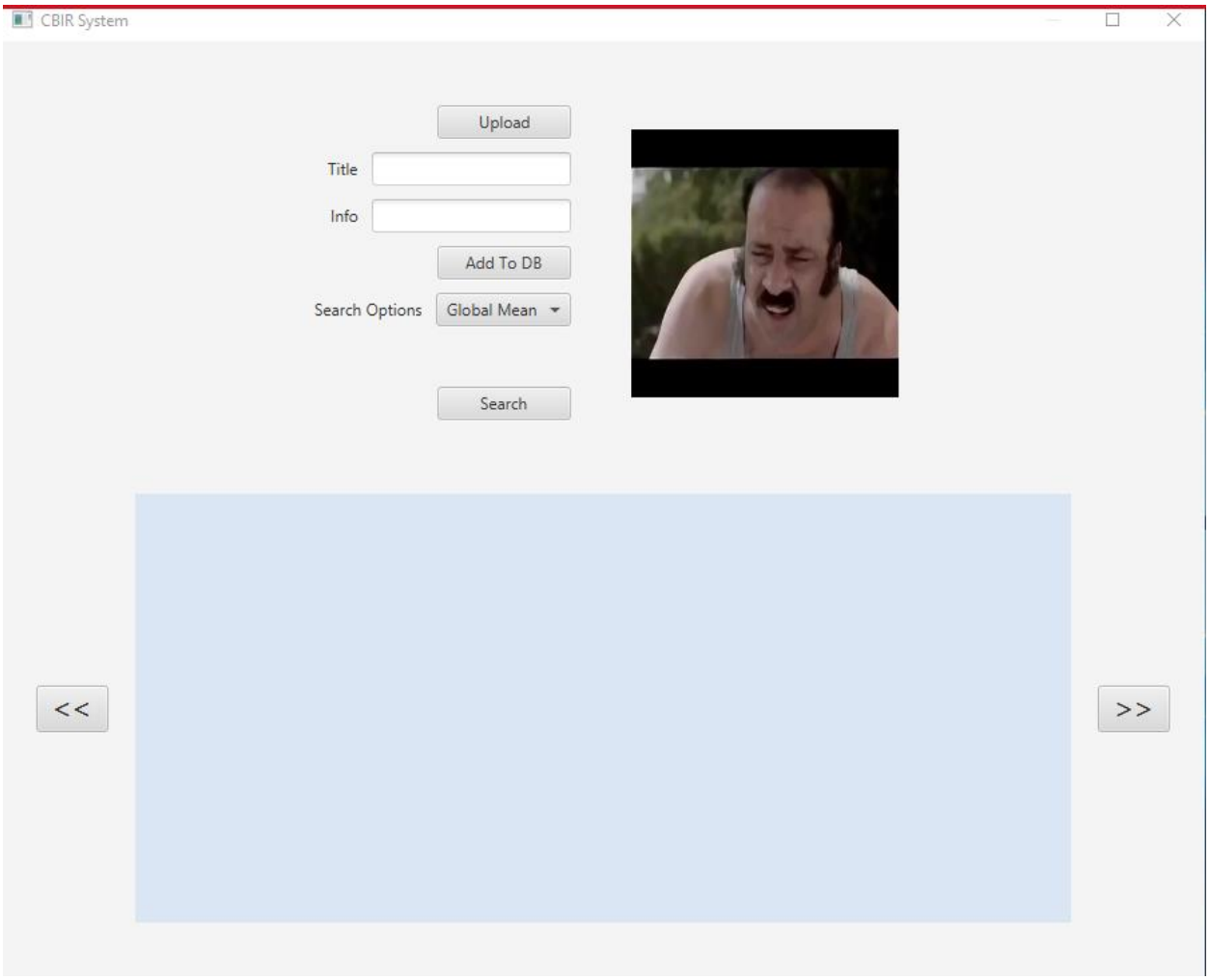

Figure 15: Upload image

After clicking “Upload”, a window pops up to choose the image.
As we can see, we can only select images with extensions “jpg, jpeg, png” and that’s some error handling technique.

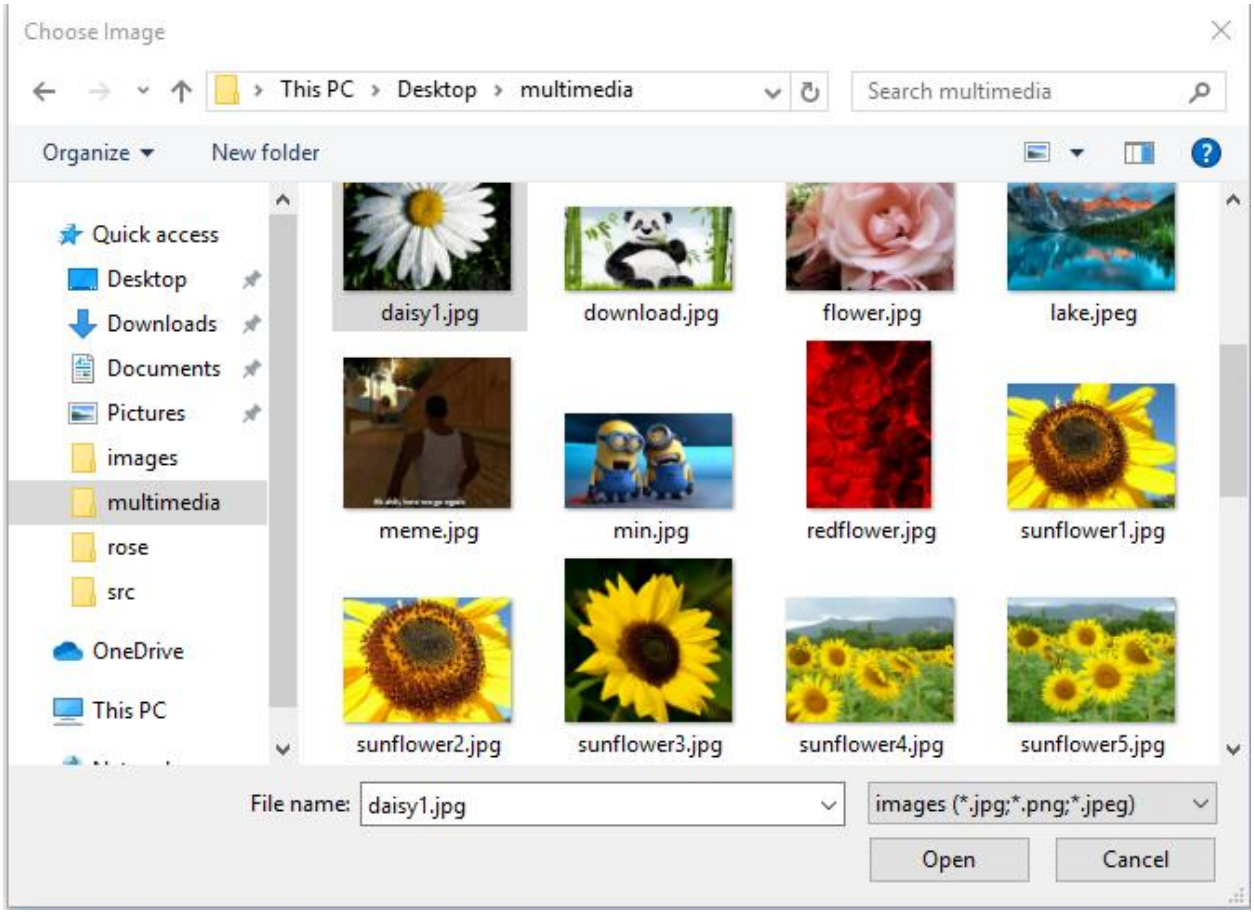

Figure 16: Upload image(2)

Add image to database

After uploading the image, we add a title and some info, the we press “Add To DB” button

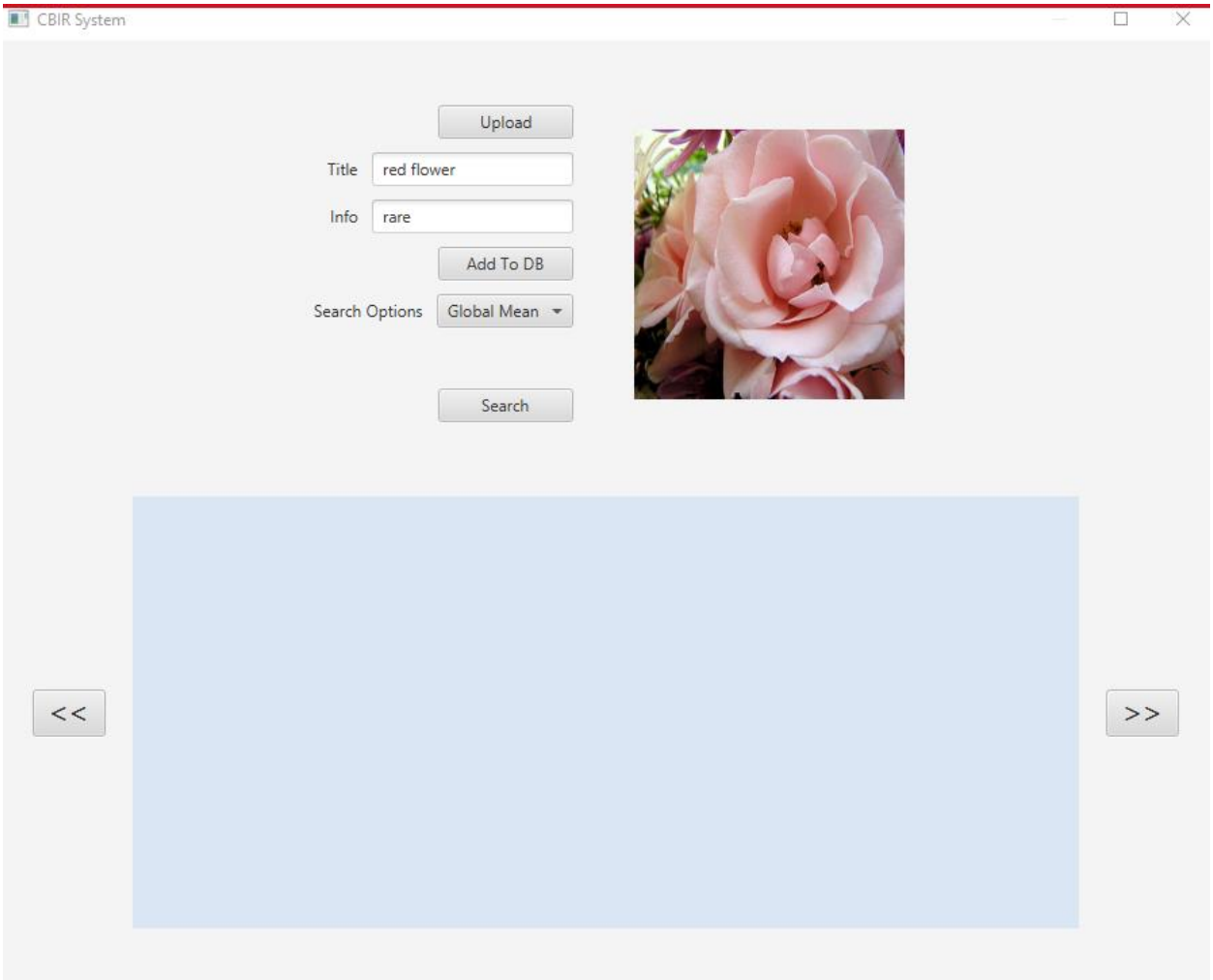

Figure 17: Add image to database

Mean Color Search

We can search by mean color

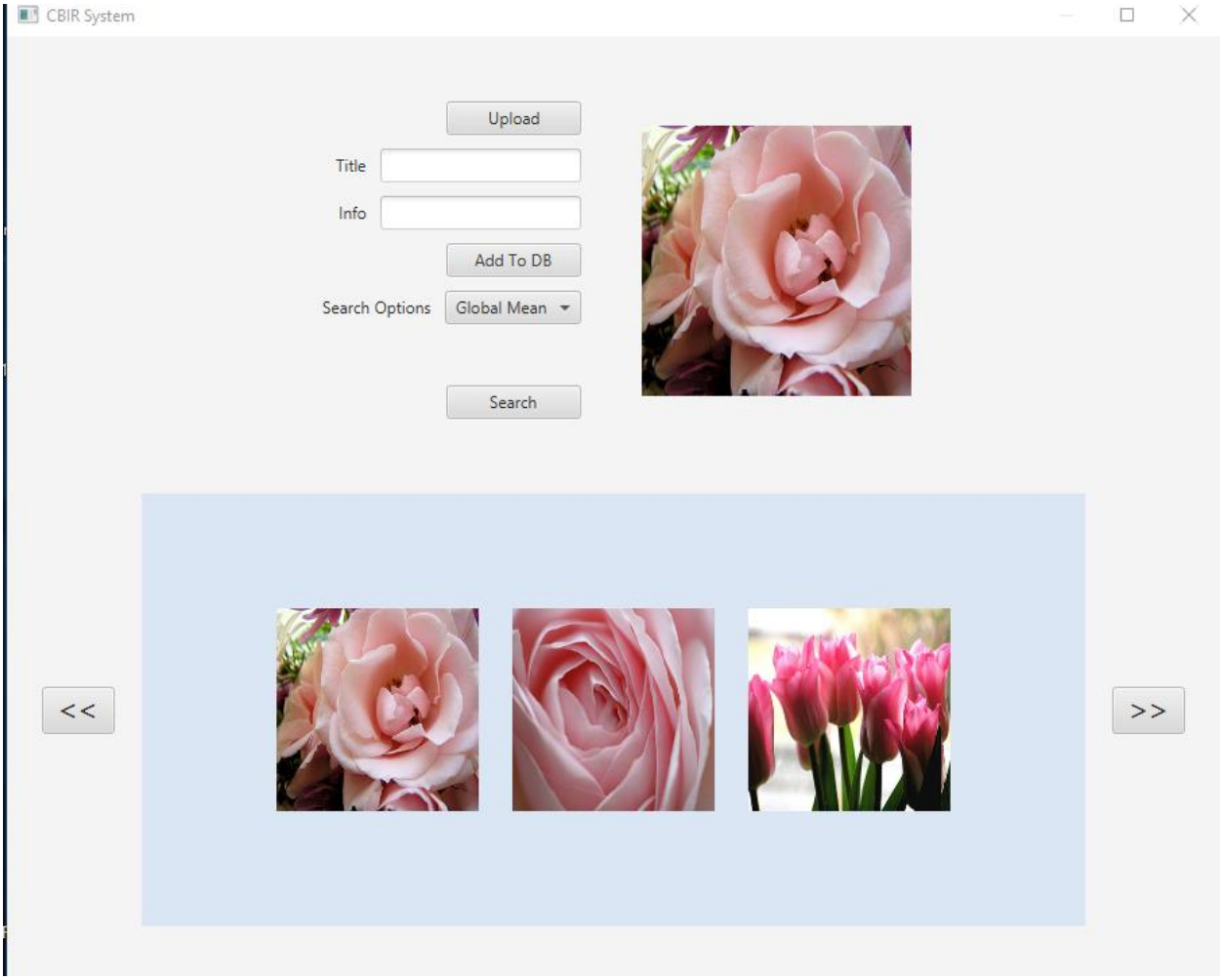

Figure 18: Mean Color Search

Histogram Search

We can search by histogram, we must first enter the threshold

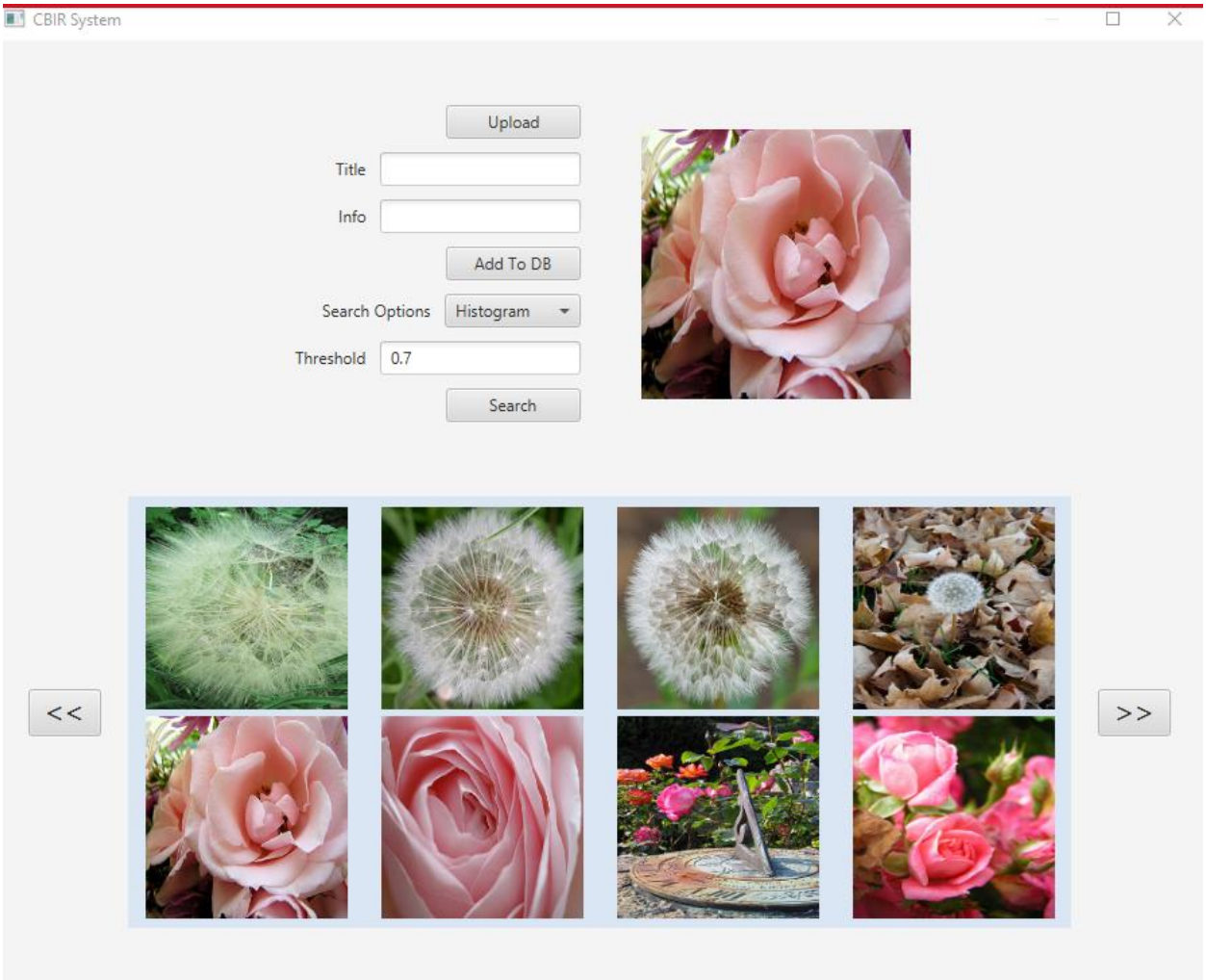

Figure 19: Histogram Search

Color Layout Search

We search by color layout, it takes input images, then divides it into small color grids, and we search by color layout by comparing each grid from query image with corresponding grid in model image

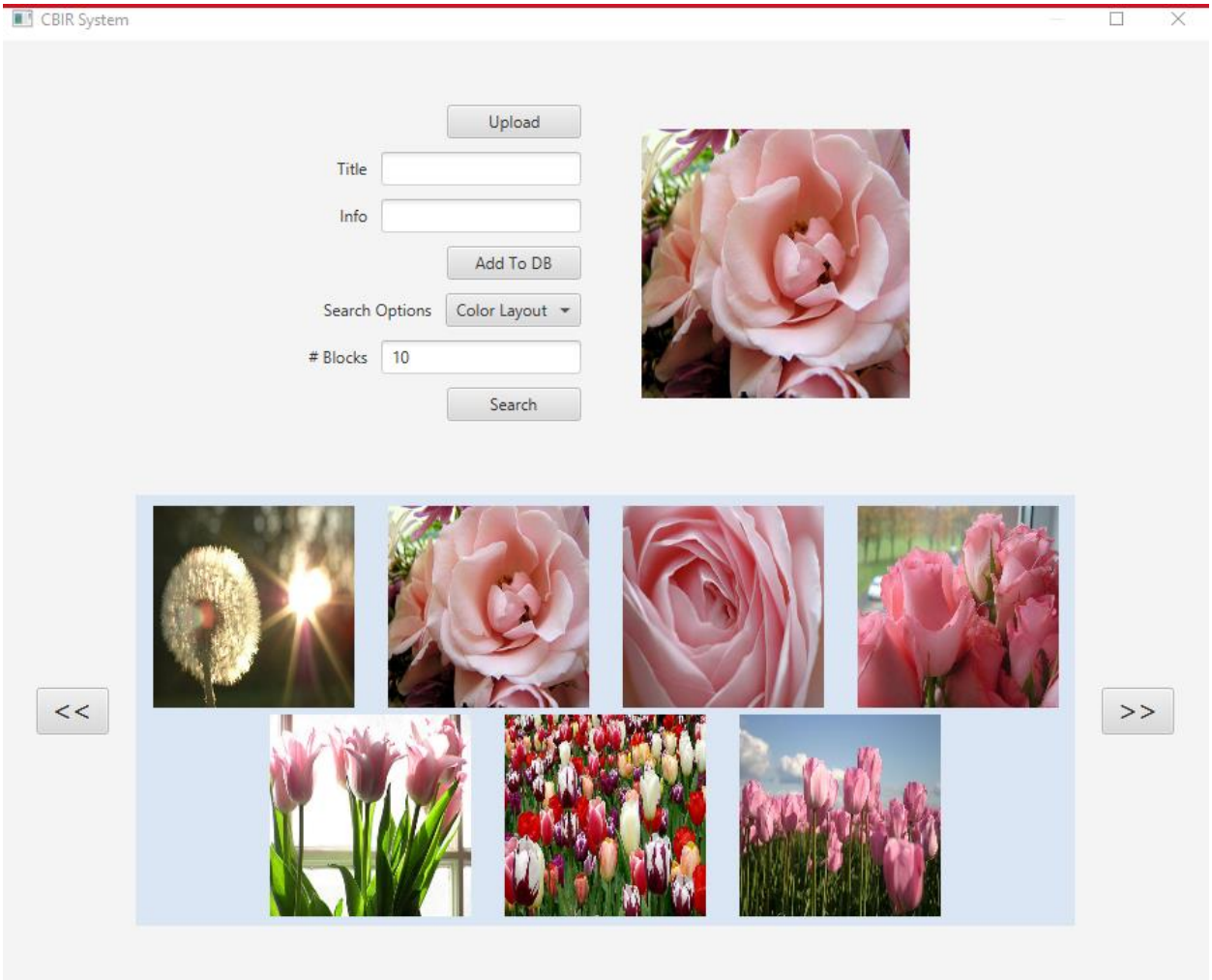

Figure 20: Color Layout Search

Interface of the CBIR system

Upload Videos

We can upload videos using “Upload” button

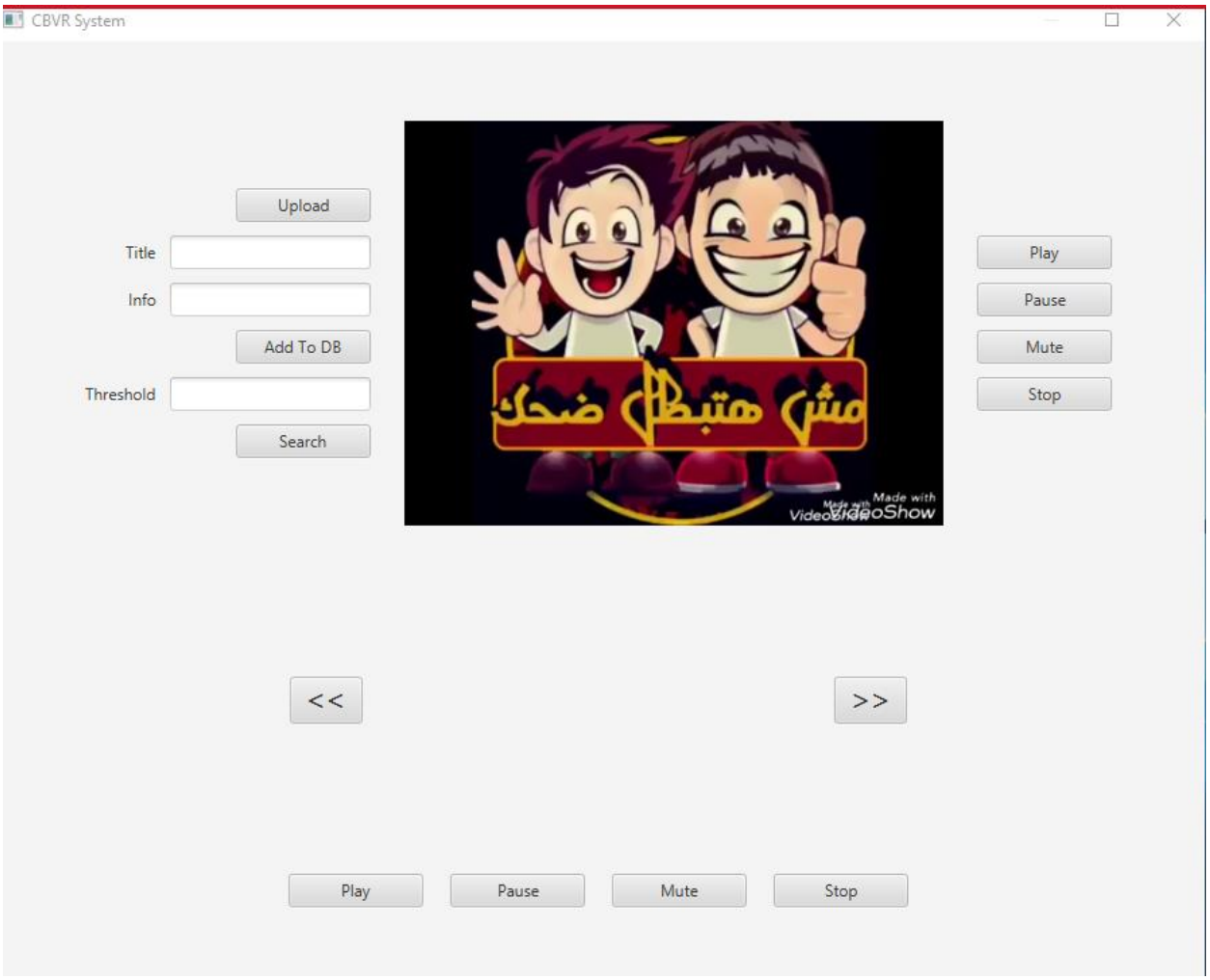

Figure 21: Upload Videos

We can only upload mp4 videos

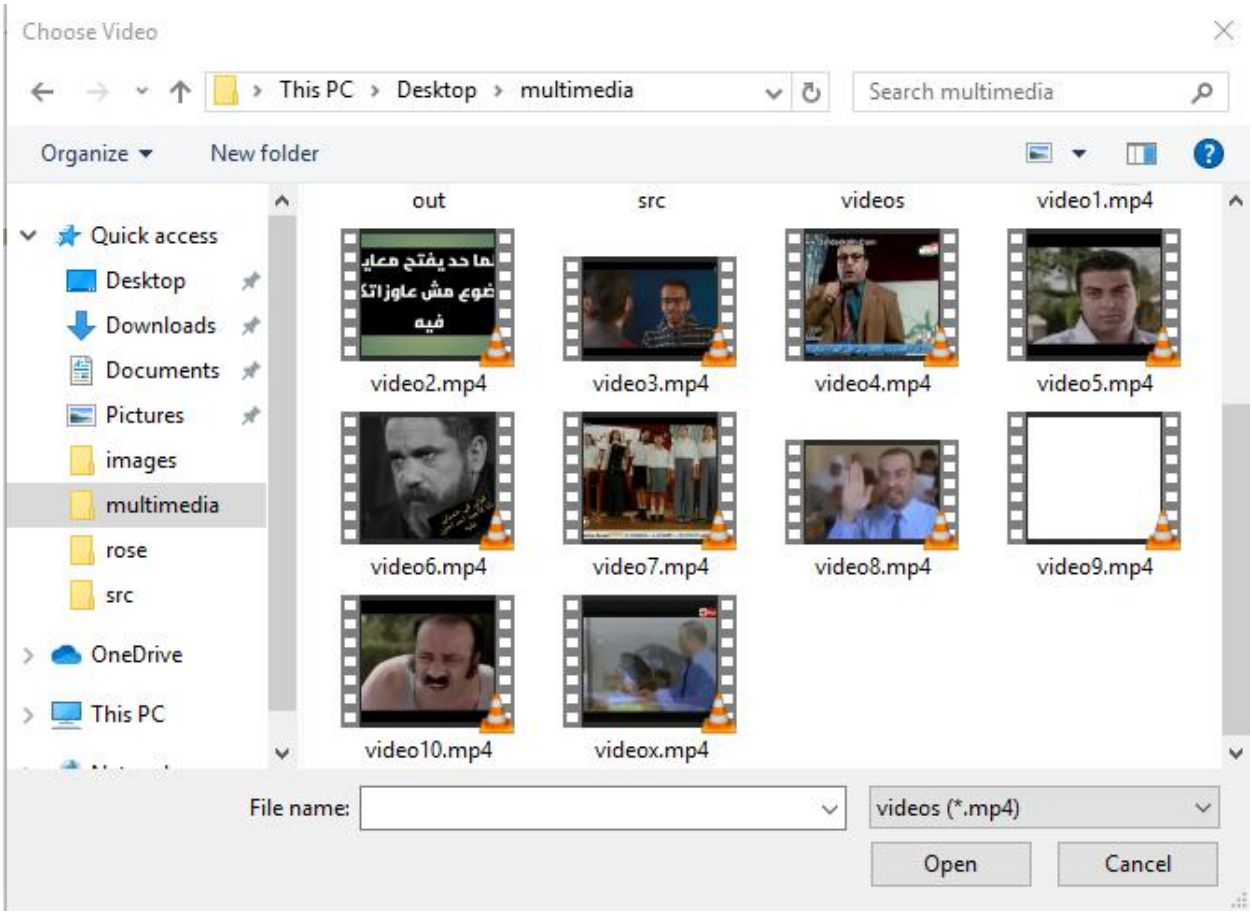

Figure 22: Upload Videos(2)

Add a video to database

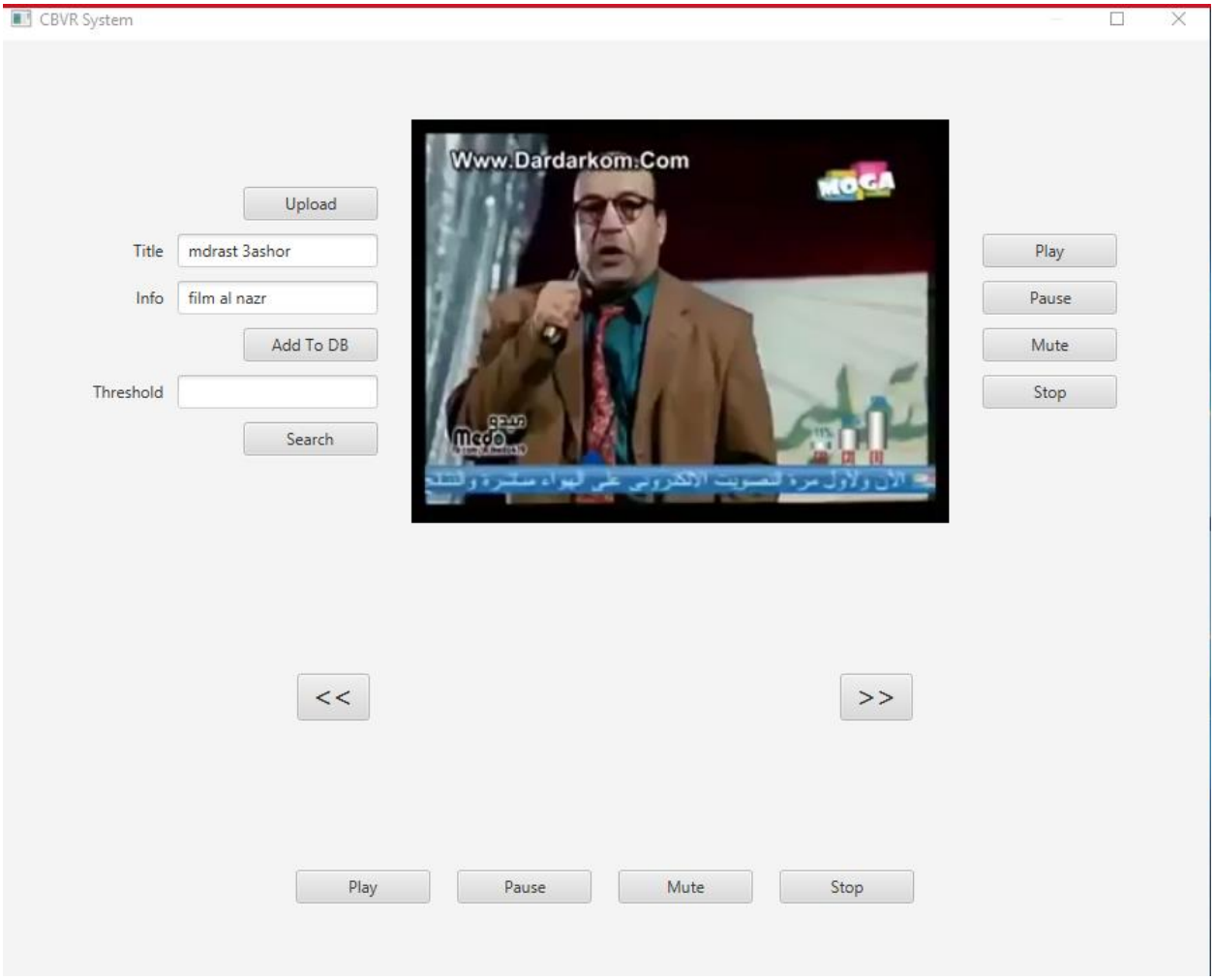

Figure 23: Add a video to database

NVS Search

We can search a video by NVS.
We can play/pause/mute/stop videos from the query video, or from the model video.

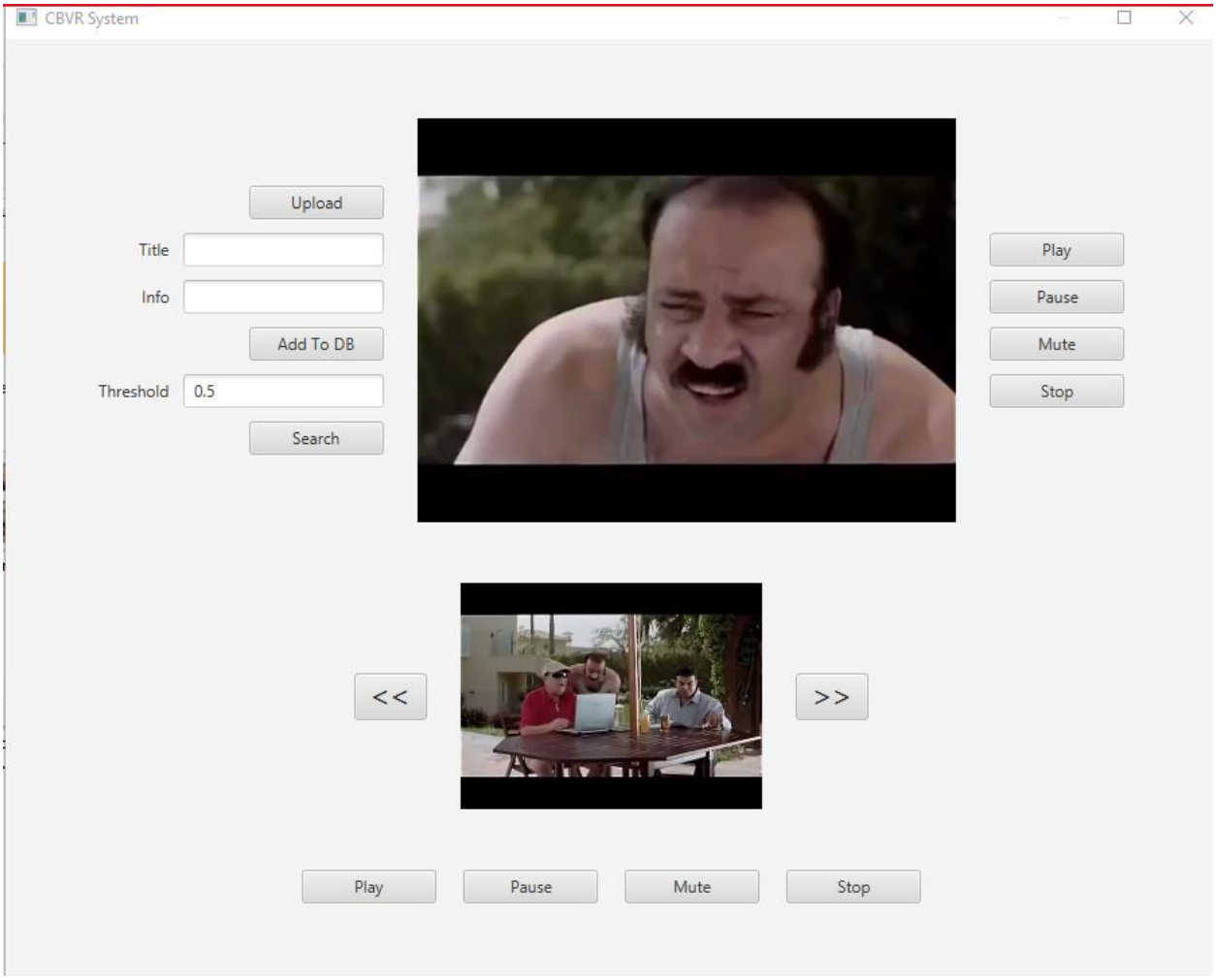

Figure 24: NVS Search

Conclusion

Image databases, once an expensive proposition, in terms of space, cost and time has now become a reality.

• Image databases, store images of a various kinds.

• These databases can be searched interactively, based on image content or by indexed keywords.

Searching by keywords (text) is fast process but it has a lot of constrains such as

• Problem of image annotation

  – Valid only for one language –with image retrieval this limitation should not exist

• Problem of human perception

  – Subjectivity of human perception
  – Too much responsibility on end-user

• Problem of abstract needs

  – Queries that cannot be described at all that depend on visual features of images.

Searching by content is better because we search by the visual features itself so we extract better multimedia content but the process is computationally expensive if there is large dataset so we must make a well-designed architecture and database and fast code in order to decrease the time of the insertion and retrieval process

References

[1] Z. Li and M. S. Drew, Fundamentals of Multimedia, Prentice Hall, 2nd Edition, Springer, 2014.

[2] Guoyong Duan, Jing Yang and Yilong Yang, Content-Based Image Retrieval Research, Department of Computer Science and Information, Guizhou University, Guiyang, 550025, China

[3] A. W. Smeulders, M. M. Worring, A. Gupta, and R. Jain, Content-Based Image Retrieval at the End of the Early Years, IEEE Trans. Pattern Anal. Machine Intell., vol.22 no.12, pp1349-1380, 2000.

[4] J.R. Smith, Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression PhD thesis, Columbia University, 1997