*A project report on*

**RETRIEVAL ORIENTED DEEP FEATURE LEARNING WITH COMPLEMENTARY SUPER VISION MINING**

*Submitted in partial fulfillment for the award of the degree of*

**B. TECH (Software Engineering)**

*by*

**KATTA. BALAJI (192110707)**

# SAVEETHA SCHOOL OF ENGINEERING,

# SIMATS.

OCTOBER, 2023

**RETRIEVAL ORIENTED DEEP FEATURE LEARNING WITH COMPLEMENTARY SUPER VISION MINING**

*Submitted in partial fulfillment for the award of the degree of*

B.Tech (Computer Science Engineering)

*by*

**KATTA. BALAJI (192110707)**

# SAVEETHA SCHOOL OF ENGINEERING,

# SIMATS

OCTOBER, 2023

## DECLARATION

I am here by declare that the thesis entitled “RETRIEVAL ORIENTED DEEP FEATURE LEARNING WITH COMPLEMENTARY SUPER VISION MINING” submitted

by me, for the award of the degree of B. Tech (Software Engineering) is a record of Bonafede work carried out by me under the supervision of PROF. TERRANCE FREDERICK FERNANDEZ.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place:THANDALM

Date: 14-10-2023

Signature of the Candidate

**K. Balaji**

## CERTIFICATE

This is to certify that the thesis entitled “RETRIEVAL ORIENTED DEEP FEATURE LEARNING WITH COMPLEMENTARY SUPER VISION MINING**”** submitted

by **KATTA.BALAJI (192110707), Saveetha School of Engineering,** **Saveetha Institute of Medical and Technical Sciences**, for the award of the degree B. Tech (Software Engineering) is a record of Bonafede work carried out by him/her under my supervision.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The Project report fulfils the requirements and regulations of **Saveetha Institute of Medical and Technical Sciences**, \_\_\_\_\_\_\_\_\_ and in my opinion meets the necessary standards for submission.

### Signature of the Guide Signature of the HoD

**Internal Examiner External Examiner**

## ABSTRACT

In this system, we suggest incorporating the knn function in a learning-based paradigm into the CNN function with a Siamese framework. The learning goal comprises of two types of loss, i.e., loss of resemblance and loss of loyalty. The first loss embeds the closest image-level neighborhood structure with knn function in CNN feature learning, while the second loss imposes that the CNN feature with the updated CNN model maintains that fidelity from the initial CNN model designed exclusively for classification. The produced CNN function inherits the property of the knn function after learning, which is well-oriented for retrieval of images.

## ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to PROF. TERRANCE FREDERICK FERNANDEZ,ProfessoR, Saveetha School of Engineering, SIMATS, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Software Testing.

I would like to express my gratitude to DR.N. M. VEERAIYAN, Chancellor Saveetha Institute of Medical and Technical Sciences (SIMATS), CHENNAI, Dr. S. Suresh Kumar, DR. SEKAR VISWANATHAN, MR. G V SELVAM, Vice – Presidents Saveetha Institute of Medical and Technical Sciences (SIMATS), CHENNAI, DR. RAMBABU KODALI, Vice – Chancellor, DR. PARTHA SHARATHI MALLICK,Pro-Vice Chancellor and Dr. S. Sumathy, Dean, School of Information Technology & Engineering (SITE), for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Shantharajah S.P, Program chair & HoD/Professor, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Thandalam

Date: 14-10-2023 **katta Balaji**

## TABLE OF CONTENTS

[ABSTRACT 5](#_TOC_250008)

[ACKNOWLEDGEMENT… 6](#_TOC_250007)

[LIST OF FIGURES 9](#_TOC_250006)

LIST OF TABLES 10

[LIST OF ACRONYMS 11](#_TOC_250005)

CHAPTER 1 INTRODUCTION

* 1. [BACKGROUND 12](#_TOC_250004)
  2. [MOTIVATION 13](#_TOC_250003)
  3. PROJECT STATEMENT 13
  4. [OBJECTIVES 13](#_TOC_250002)
  5. [SCOPE OF THE PROJECT 14](#_TOC_250001)

CHAPTER 2 LITERATURE SURVEY

* 1. [SUMMARY OF THE EXISTING WORKS 15](#_TOC_250000)
  2. CHALLENGES PRESENT IN EXISTING SYSTEM 18

CHAPTER 3 REQUIREMENTS

* 1. HARDWARE REQUIREMENTS 19
  2. SOFTWARE REQUIREMENTS 19

CHAPTER 4 ANALYSIS & DESIGN

* 1. EXISTING SYSTEM 24
  2. PROPOSED METHODOLOGY 25
  3. SYSTEM ARCHITECTURE 26
  4. FLOW DIAGRAM… 27
  5. UML DIAGRAMS 28
     1. USECASE DIAGRAM
     2. SEQUENCE DIAGRAM
     3. CLASS DIAGRAM
  6. MODULE DESCRIPTIONS 30

## CHAPTER 5 IMPLEMENTATION & TESTING

* 1. DATASET 33
  2. SAMPLE CODE 33
  3. SAMPLE OUTPUT 43
  4. TEST PLAN & DATA VERIFICATION… 45

## CHAPTER 6 RESULTS

**6.1** RESULT ANALYIS & EVALUATION METRICS 46

## CONCLUSIONS AND FUTURE WORK 50

**REFERENCES** 51

## LIST OF FIGURES

1. SYSTEM ARCHITECTURE 26
2. FLOW DIAGRAM 27
3. USECASE DIAGRAM 28
4. SEQUENCE DIAGRAM 28
5. CLASS DIAGRAM 29

## LIST OF TABLES

EVALUATION METRICS 46

## LIST OF ACRONYMS

SVM -- SUPER VECTOR MACHINE KNN -- K-NEAREST NEIGHBOR

CNN -- CONVOLUTIONAL NEURAL NETWORK SIFT -- SCALE-INVARIANT FEATURE TRANSFORM DWT -- DISCRETE WAVELET TRANSFORM

### BACKGROUND

# Chapter 1

## INTRODUCTION

With the gradual development of internet technology and digital devices billions of people are browsing web and sharing the information in the form of text, images, audio or video. Image Retrieval (IR) is the mechanism by which the images related to query image are searched from the images database and user retrieves the most similar images from the database. This image retrieval system is known as Content Based Information Retrieval system (CBIR) or query by image content retrieval system (QBIC). In technical point of view, there are main three steps in content-based image retrieval system: image representation, image organization, and image similarity measurement. CBIR is to retrieve information from the very big visual database thus organizing the large-scale database efficiently and identifying the relevant results of a given query within acceptable time limits and storage demands is the main goal of image retrieval system.

In this paper, to learn features oriented for image retrieval, we propose to implicitly embed the SIFT feature into CNN feature with a Siamese structure. To enable accurate and efficient retrieval in massive image collections, compact and rich image feature representations are at the core of successful image retrieval. In the past two decades, remarkable progress has been made, focusing more on using primitive color features, texture features, and shape features. These low-level features heavily depend on hand-engineered feature descriptors such as Scale Invariant Feature Transform (SIFT). Witnessing the great success of low-level SIFT feature in instance retrieval and its complementary nature to the semantic aware CNN feature, many approaches fuse these two features in a multi-level and complementary way. In a novel unsupervised method is proposed to update the CNN features by implicitly embedding the information from SIFT features. Specifically, true matches of each database image is identified with strong reciprocal nearest neighbour constraints from SIFT features, and their CNN features are updated to be closer to each other. An iterative update is performed to alternatively revise both CNN and SIFT features in a heuristic way. It successfully embeds the pseudo-supervision from SIFT feature spaces into the CNN representation, and subtly achieve information fusion to the features.

### MOTIVATION

The major research gap lies for reducing machine training time for training large dataset image samples. The existing databases need further refinement for better retrieval results or new datasets are to be generated to foster upcoming demands. The review of literature spreads light on the major issues to be addressed in the field of CBIR. The some of the issues are: the semantic gap in extracting the meaningful image information from image and then transforming into a visual vector in accordance with human perception is the prime most issue. The images belonging to different classes in the database are to be represented by the feature vectors which may be markedly differently from different users.

### PROBLEM STATEMENT

We developed this project based on google queries and fetching the images or videos, so normally in some engines like opera, and mini browsers fetching the related search images taskes time so for that browsers we implemented this project model that is here using some algorithms which described below we are going to make a project with more accuracy and less error rate we developed this project to overcome problem facing by the browsers.

### OBJECTIVES

The objective is to excerpt visual content of an image inevitably, like color, shape or texture. The CBIR tools can be utilized in numerous applications such as digital libraries, photo sharing sites and crime prevention. There are various techniques for content-based image retrieval is presented in past, which are either based on extracting the individual image feature (color, texture or shape) or combination of these features. Some of the techniques are based on the same image feature extraction, but different in the method of extracting the particular image feature, used canny and Sobel edge detection procedure for mining the shape features of a given image, have utilized color

histogram technique and color correlogram technique for mining texture and color feature of a given image, used DCT (Discrete Cosine Transforms) method for mining texture and color feature of images to retrieve requested image etc.

### SCOPE OF THE PROJECT

With the growth of personal image collection and shared images (images made public from private image collections), the scope for searching and viewing image content is also increasing. The ability to create comments, tag, and annotate images leaves the doors open for integrating visual elements, textual retrieval, and browsing for many indexing and search opportunities. Developing and evaluating adaptable systems for users could be quite challenging since the following need to be considered: human memory, context, and subjectivity. In order to develop excellent personal image retrieval systems, a holistic approach to designing and developing applications must be taken considering all the related variables concerning the final human users of these systems.

# Chapter 2

## LITERATURE SURVEY

* 1. SUMMARY OF THE EXISTING WORKS
     1. TOPIC: EFFECTIVE IMAGE RETRIEVAL BASED ON HIDDEN CONCEPT DISCOVERY IN IMAGE DATABASE

This paper addresses content-based image retrieval in general, and in particular, focuses on developing a hidden semantic concept discovery methodology to address effective semantics- intensive image retrieval. In our approach, each image in the database is segmented into regions associated with homogenous color, texture, and shape features. By exploiting regional statistical information in each image and employing a vector quantization method, a uniform and sparse region-based representation is achieved. With this representation, a probabilistic model based on statistical-hidden-class assumptions of the image database is obtained, to which the expectation- maximization technique is applied to analyze semantic concepts hidden in the database. An elaborated retrieval algorithm is designed to support the probabilistic model. The semantic similarity is measured through integrating the posterior probabilities of the transformed query image, as well as a constructed negative example, to the discovered semantic concepts. The proposed approach has a solid statistical foundation; the experimental evaluations on a database of 10 000 general-purposed images demonstrate its promise and effectiveness

* + 1. TOPIC: A NEW WAY FOR MULTIDIMENSIONAL MEDICAL DATA MANAGEMENT: VOLUME OF INTEREST (VOI)-BASED RETRIEVAL OF MEDICAL IMAGES WITH VISUAL AND FUNCTIONAL FEATURES

The advances in digital medical imaging and storage in integrated databases are resulting in growing demands for efficient image retrieval and management. Content-based image retrieval (CBIR) refers to the retrieval of images from a database, using the visual features derived from the information in the image, and has become an attractive approach to managing large medical image archives. In conventional CBIR systems for medical images, images are often segmented into regions which are used to derive two-dimensional visual features for region-based queries.

Although such approach has the advantage of including only relevant regions in the formulation of a query, medical images that are inherently multidimensional can potentially benefit from the multidimensional feature extraction which could open up new opportunities in visual feature extraction and retrieval. In this study, we present a volume of interest (VOI) based content-based retrieval of four-dimensional (three spatial and one temporal) dynamic PET images. By segmenting the images into VOIs consisting of functionally similar voxels (e.g., a tumor structure), multidimensional visual and functional features were extracted and used as region-based query features. A prototype VOI-based functional image retrieval system (VOI-FIRS) has been designed to demonstrate the proposed multidimensional feature extraction and retrieval. Experimental results show that the proposed system allows for the retrieval of related images that constitute similar visual and functional VOI features, and can find potential applications in medical data management, such as to aid in education, diagnosis, and statistical analysis

* + 1. TOPIC: MEDICAL IMAGE CATEGORIZATION AND RETRIEVAL FOR PACS USING THE GMM-KL FRAMEWORK

This paper presents an image representation and matching framework for image categorization in medical image archives. Categorization enables one to determine automatically, based on the image content, the examined body region and imaging modality. It is a basic step in content-based image retrieval (CBIR) systems, the goal of which is to augment text-based search with visual information analysis. CBIR systems are currently being integrated with picture archiving and communication systems for increasing the overall search capabilities and tools available to radiologists. The proposed methodology is comprised of a continuous and probabilistic image representation scheme using Gaussian mixture modeling (GMM) along with information-theoretic image matching via the Kullback-Leibler (KL) measure. The GMM-KL framework is used for matching and categorizing X-ray images by body regions. A multidimensional feature space is used to represent the image input, including intensity, texture, and spatial information. Unsupervised clustering via the GMM is used to extract coherent regions in feature space that are then used in the matching process. A dominant characteristic of the radiological images is their poor contrast and large intensity variations. This presents a challenge to matching among the

images, and is handled via an illumination-invariant representation. The GMM-KL framework is evaluated for image categorization and image retrieval on a dataset of 1500 radiological images. A classification rate of 97.5% was achieved. The classification results compare favorably with reported global and local representation schemes. Precision versus recall curves indicate a strong retrieval result as compared with other state-of-the-art retrieval techniques. Finally, category models are learned and results are presented for comparing images to learned category models.

* + 1. TOPIC: SUPERVISED LEARNING OF SEMANTIC CLASSES FOR IMAGE ANNOTATION AND RETRIEVAL

A probabilistic formulation for semantic image annotation and retrieval is proposed. Annotation and retrieval are posed as classification problems where each class is defined as the group of database images labelled with a common semantic label. It is shown that, by establishing this one- to-one correspondence between semantic labels and semantic classes, a minimum probability of error annotation and retrieval are feasible with algorithms that are 1) conceptually simple, 2) computationally efficient, and 3) do not require prior semantic segmentation of training images. In particular, images are represented as bags of localized feature vectors, a mixture density estimated for each image, and the mixtures associated with all images annotated with a common semantic label pooled into a density estimate for the corresponding semantic class. This pooling is justified by a multiple instance learning argument and performed efficiently with a hierarchical extension of expectation-maximization. The benefits of the supervised formulation over the more complex, and currently popular, joint modelling of semantic label and visual feature distributions are illustrated through theoretical arguments and extensive experiments. The supervised formulation is shown to achieve higher accuracy than various previously published methods at a fraction of their computational cost. Finally, the proposed method is shown to be fairly robust to parameter tuning.

* + 1. TOPIC: MATCHING AND RETRIEVAL OF TATTOO IMAGES: ACTIVE CONTOUR CBIR AND GLOCAL IMAGE FEATURES

Tattoos provide an important source of biometric information, particularly in gang-related criminal activity. The goal of this paper is the formation of an image analysis tool to match tattoos and to retrieve similar tattoos from a tattoo database. First, an existing content based image retrieval (CBIR) approach for tattoos is reviewed. Then, a new active contour CBIR approach is detailed. This method incorporates vector field convolution active contours for tattoo segmentation, Haar wavelet decomposition for texture analysis, hue-saturation-value histograms for color representation and Fourier shape descriptors for shape characterization. Finally, the glocal (global- local) image feature approach is introduced. Results are provided for two datasets that include both recreational and prison/gang tattoos.

* 1. CHALLENGES PRESENT IN EXISTING SYSTEM
* Performance metrics value is reduced because of SVM and KNN
* Accuracy is less
* Precision and recall is also very less

# Chapter 3

## REQUIREMENTS

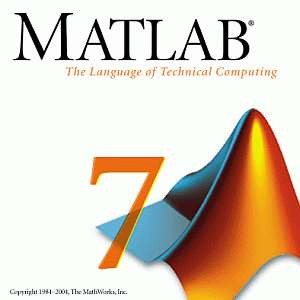
* 1. HARDWARE REQUIREMENTS
     + Processor Type : Pentium -IV
     + Speed : 2.4 GHZ
     + Ram : 4 GB RAM
     + Hard disk : 20 GB HD
  2. SOFTWARE REQUIREMENTS
     + Operating System : Windows 7
     + Software Programming Package : MATLAB R2018a

### INTRODUCTION TO MATLAB:

**MATLAB VERSION REQUIRED- 2018a**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

* + - * Math and computation
      * Algorithm development
      * Data acquisition
      * Modeling, simulation, and prototyping
      * Data analysis, exploration, and visualization
      * Scientific and engineering graphics
      * Application development, including graphical user interface building



MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high- productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most uses of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M – files) that

extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

### The MATLAB system:

The MATLAB system consists of five main parts

### Development Environment:

This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and command window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

### The MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

### The MATLAB Language:

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both “programming in the small” to rapidly create quick and dirty throw-away programs, and “programming in the large” to create large and complex application programs.

### Graphics:

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization,

image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

### The MATLAB Application Program Interface (API):

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Various toolboxes are there in MATLAB for computing recognition techniques, but we are using

**IMAGE PROCESSING** toolbox.

### SOFTWARE DESCRIPTION

**Getting Started**

If you are new to MATLAB, you should start by reading Manipulating Matrices. The most important things to learn are how to enter matrices, how to use the: (colon) operator, and how to invoke functions. After you master the basics, you should read the rest of the sections below and run the demos.

At the heart of MATLAB is a new language you must learn before you can fully exploit its power. You can learn the basics of MATLAB quickly, and mastery comes shortly after. You will be rewarded with high productivity, high-creativity computing power that will change the way you work.

**Introduction** - describes the components of the MATLAB system.

**Development Environment** - introduces the MATLAB development environment, including information about tools and the MATLAB desktop.

**Manipulating Matrices** - introduces how to use MATLAB to generate matrices and perform mathematical operations on matrices.

**Graphics** - introduces MATLAB graphic capabilities, including information about plotting data, annotating graphs, and working with images.

**Programming with MATLAB** - describes how to use the MATLAB language to create scripts and functions, and manipulate data structures, such as cell arrays and multidimensional arrays.

### INTRODUCTION

**What Is MATLAB?**

MATLAB® is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

* + Math and computation
  + Algorithm development
  + Modeling, simulation, and prototyping
  + Data analysis, exploration, and visualization
  + Scientific and engineering graphics
  + Application development, including graphical user interface building



**Chapter 4**

## ANALYSIS & DESIGN

* 1. EXISTING SYSTEM
     + In the existing system, The image retrieval is based on the wavelet features using Machine learning algorithm i.e., SVM and KNN
     + The first loss embeds the closest image-level neighborhood structure with SIFT function in SVM and KNN feature learning, while the second loss imposes that the KNN feature with the updated SVM and KNN model.
     + It maintains that fidelity from the initial SVM and KNN model designed exclusively for classification.
     + The produced SVM and KNN function inherits the property of the SIFT function after learning, which is well-oriented for retrieval of images
  2. PROPOSED METHODOLOGY
     + In the proposed system, to learn retrieval-oriented feature representation, we make full use of the complementary nature of the SIFT feature and CNN feature and transfer the closest neighborhood structure in deep CNN learning to the SIFT-based image level.
     + We obtain training data automatically based on reciprocal neighborhood limitations from SIFT.
     + The learning goal with Siamese network architecture comprises of two types of loss, i.e., SIFT-guided loss of resemblance and CNN-based fidelity

ADVANTAGES OF THE EXISTING SYSTEM

* + - * Performance metrics is increased using CNN
      * Accuracy value is more
      * Precision and recall value is more

### User

* 1. SYSTEM ARCHITECTURE

**IMAGE DATABASE**

### QUERY IMAGES

**PRE-PROCESSING**

GAUSSIAN FILTERING

### QUERY PROCESSING

**FEATURE**

**EXTRACTION(**color,Edge, Texture)

**-**DISCRETE WAVELET TRANSFORM

**FEATURE**

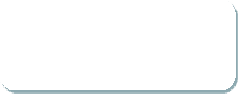
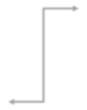
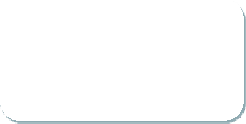
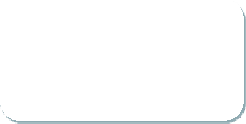
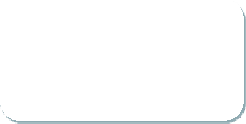
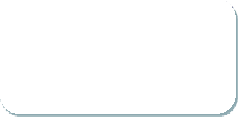
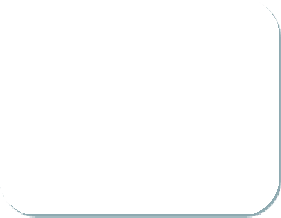
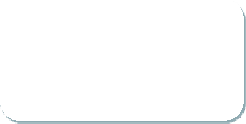
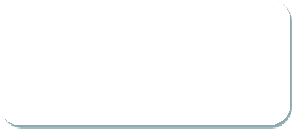
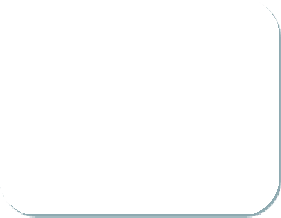
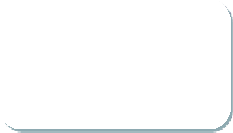
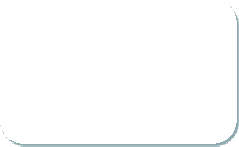
**EXTRACTION(**color,Edge, Texture)

**-**DISCRETE WAVELET TRANSFORM

**INDEXING CLASSIFICATION**

-CONVOLUTION NEURAL NETWORK

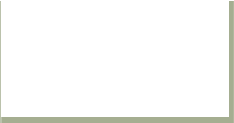
### SIMILARITY MATCHING



**RETRIEVAL RESULT**

IMAGE FEATURE VECTOR DATABASE

### DISPLAY RESULT

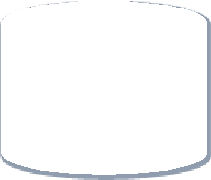


**PRE-**

**PROCESSING**

**QUREY**

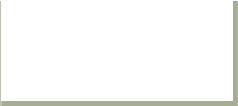
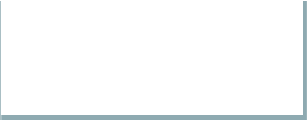
**IMAGE**



**DATASETS**



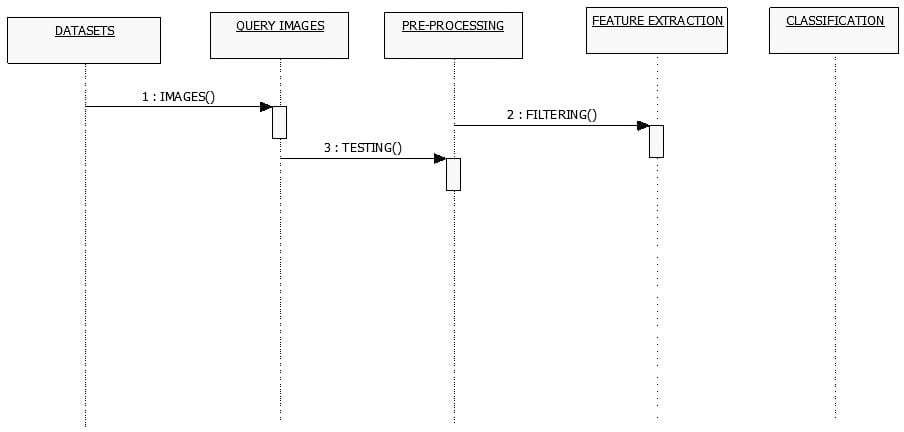
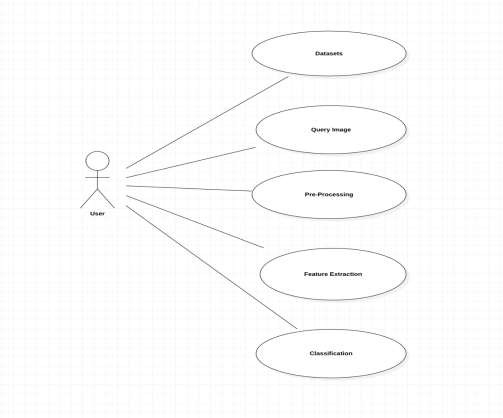
**4.4 FLOW DIAGRAM**



**TRAINING**

**CNN**

**FEATURE EXTRACTION**





4.5.1 USECASE DIAGRAM



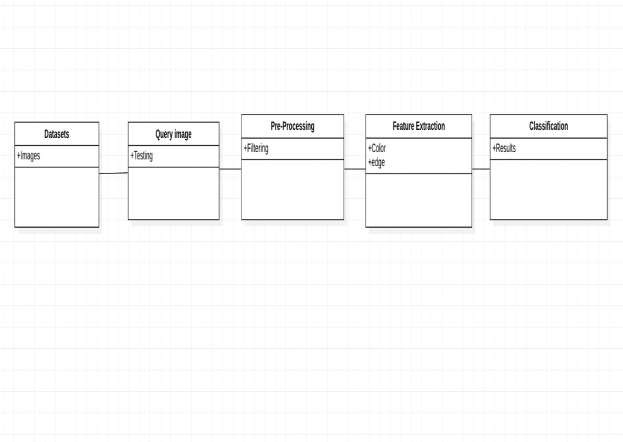
4.5.2 SEQUENCE DIAGRAM



**4.5 UML DIAGRAMS**



4.5.3 CLASS DIAGRAM



4.6 MODULE DESCRIPTIONS

### PRE-PROCESSING:

**Digital image processing** is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, **digital image processing** has many advantages over **analogue image processing**. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our **AI-Computer Vision** models can benefit from this improved data to work on. An image is nothing more than a two-dimensional array of numbers (or pixels) ranging between 0 and 255. It is defined by the mathematical function f (x, y) where x and y are the two co-ordinates horizontally and vertically. The value of f (x, y) at any point is giving the pixel value at that point of an image.

### QUERY IMAGE:

content-based image retrieval, also known as query by image content ([QBIC](https://en.wikipedia.org/wiki/Content-based_image_retrieval#QBIC)) and content-based visual information retrieval (CBVIR), is the application of [computer vision](https://en.wikipedia.org/wiki/Computer_vision) techniques to the [image retrieval](https://en.wikipedia.org/wiki/Image_retrieval) problem, that is, the problem of searching for [digital images](https://en.wikipedia.org/wiki/Digital_image) in large [databases](https://en.wikipedia.org/wiki/Database) (see this survey[[1]](https://en.wikipedia.org/wiki/Content-based_image_retrieval#cite_note-Survey-1) for a recent scientific overview of the CBIR field). Content- based image retrieval is opposed to traditional concept-based approaches (see [Concept-based](https://en.wikipedia.org/wiki/Concept-based_image_indexing) [image indexing](https://en.wikipedia.org/wiki/Concept-based_image_indexing)).

"Content-based" means that the search analyzes the contents of the image rather than the [metadata](https://en.wikipedia.org/wiki/Metadata_(computing)) such as keywords, tags, or descriptions associated with the image. The term "content" in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself. CBIR is desirable because searches that rely purely on metadata are dependent on annotation quality and completeness.

### FEATURES EXTRACTION:

The feature is defined as a function of one or more measurements, each of which specifies some quantifiable property of an object, and is computed such that it quantifies some significant characteristics of the object. The various features classified and currently employed are • General features: Independent features such as color, texture, and shape According to the abstraction level, they can be further divided into: - Pixel-level features: Features calculated at each pixel, e.g. color, location. - Local features: Features calculated over the results of subdivision of the image band of an image segmentation or edge detection. - Global features: Features calculated over the entire image or just regular sub-area of an image. Domain-specific features: Application of dependent features such as human faces, fingerprints and conceptual ones. All features can be coarsely classified into low-level features and high-level features. Low-level features can be extracted directly from the original images, whereas high-level feature extraction depends on low level features. The issue of choosing the features from the extracted vector should be guided by the following concerns:

* The features should carry enough information about the image and should not require any domain-specific knowledge for their extraction.
* They should be easy to compute in order to approach the feasibility of a large image collection and rapid retrieval.
* They should relate well to the human perceptual characteristics since users finally determine the suitability of the images retrieved.

**TRAINING:**

The model is initially fit on a training dataset, that is a set of examples used to fit the parameters (e.g., weights of connections between neurons in [artificial neural networks](https://en.wikipedia.org/wiki/Artificial_neural_networks)) of the model. The model (e.g. a [neural net](https://en.wikipedia.org/wiki/Neural_net) or a [naive Bayes classifier](https://en.wikipedia.org/wiki/Naive_Bayes_classifier)) is trained on the training dataset using a [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) method (e.g. [gradient descent](https://en.wikipedia.org/wiki/Gradient_descent) or [stochastic gradient descent](https://en.wikipedia.org/wiki/Stochastic_gradient_descent)). In practice, the training dataset often consist of pairs of an input [vector](https://en.wikipedia.org/wiki/Array_data_structure) (or scalar) and the corresponding output vector (or scalar), which is commonly denoted as the *target* (or *label*). The current model is run with the training dataset and produces a result, which is then compared with the *target*, for each input vector in the training dataset. Based on the result of the comparison and the specific learning algorithm being used, the parameters of the model are adjusted. The model fitting can include both [variable selection](https://en.wikipedia.org/wiki/Feature_selection) and parameter [estimation](https://en.wikipedia.org/wiki/Estimation_theory).

Successively, the fitted model is used to predict the responses for the observations in a second dataset called the **validation dataset**. The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)) (e.g., the number of hidden units in a neural network). Validation datasets can be used for [regularization](https://en.wikipedia.org/wiki/Regularization_(mathematics)) by [early](https://en.wikipedia.org/wiki/Early_stopping) [stopping](https://en.wikipedia.org/wiki/Early_stopping): stop training when the error on the validation dataset increases, as this is a sign of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to the training dataset. This simple procedure is complicated in practice by the fact that the validation dataset's error may fluctuate during training, producing multiple local minima. This complication has led to the creation of many ad-hoc rules for deciding when overfitting has truly begun.

Finally, the **test dataset** is a dataset used to provide an unbiased evaluation of a *final* model fit on the training dataset. If the data in the test dataset has never been used in training (for example in [cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics))), the test dataset is also called a **holdout dataset**.

**CLASSIFICATION:**

The learning algorithms are broadly classified into supervised and unsupervised learning techniques. The distinction is drawn from how the learner classifies data. In supervised learning, the classes are predetermined. These classes can be conceived of as a finite set, previously arrived at by a human. In practice, a certain class of data will be labeled with these classifications. reviewed the classes are then evaluated based on their predictive capacity in relation to measures of variance in the data itself. Some of the examples of supervised classification techniques are Back Propagation Network (BPN), Learning Vector Quantization (LVQ), Self-Organizing Map (SOM), Support Vector Machine (SVM), etc., The basic task of unsupervised learning is to develop classification labels automatically. Unsupervised algorithms seek out similarity between 112 pieces of data in order to determine whether that can be characterized as forming a group. These groups are termed clusters. Unsupervised classification, often called as clustering, the system is not informed how the pixels are grouped. The task of clustering is to arrive at some grouping of the data. One of the very common of cluster analysis is K-means clustering. The unsupervised clustering is a kind of clustering which takes place with minimum input from the operator; no training sample is available and part of the feature space is achieved by identifying natural groupings of the given data. In unsupervised clustering technique, an individual pixel is compared to each cluster to see the closest pixel. Finally, a map of all pixels in the image, classified as to different clusters, each pixel is most likely to belong, is produced. This then must be interpreted by the user as to what the color patterns may mean in terms of classes, etc. that are actually present in the real world scene; this requires some knowledge of the scene's feature or cluster content from general experience or personal familiarity with the area imaged. This section discusses three unsupervised classification techniques namely K-means clustering, PCA based K- means clustering and Fuzzy C-Means clustering (FCM).



Chapter 5

**IMPLEMENTATION & TESTING**



5.1 DATA SET



We downloaded dataset from Uci.education.



5.2 SAMPLE CODE

**Cbires.m**

function varargout = cbires(varargin)

% CBIRES MATLAB code for cbires.fig

% CBIRES, by itself, creates a new CBIRES or raises the existing

% singleton\*.

%

% H = CBIRES returns the handle to a new CBIRES or the handle to

% the existing singleton\*.

%

% CBIRES('CALLBACK',hObject,eventData,handles,...) calls the local

% function named CALLBACK in CBIRES.M with the given input arguments.

%

% CBIRES('Property','Value',...) creates a new CBIRES or raises the

% existing singleton\*. Starting from the left, property value pairs are

% applied to the GUI before cbires\_OpeningFcn gets called. An

% unrecognized property name or invalid value makes property application

% stop. All inputs are passed to cbires\_OpeningFcn via varargin.

%

% \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one

% instance to run (singleton)".

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ... 'gui\_Singleton', gui\_Singleton, ... 'gui\_OpeningFcn', @cbires\_OpeningFcn, ... 'gui\_OutputFcn', @cbires\_OutputFcn, ... 'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1}) gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:}); else

gui\_mainfcn(gui\_State, varargin{:}); end

% End initialization code - DO NOT EDIT

% --- Executes just before cbires is made visible.

function cbires\_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% varargin command line arguments to cbires (see VARARGIN)

% Choose default command line output for cbires handles.output = hObject;

% Update handles structure guidata(hObject, handles);

% UIWAIT makes cbires wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line. function varargout = cbires\_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure varargout{1} = handles.output;

% --- Executes on button press in btn\_BrowseImage.

function btn\_BrowseImage\_Callback(hObject, eventdata, handles)

% hObject handle to btn\_BrowseImage (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

[query\_fname, query\_pathname] = uigetfile('\*.jpg; \*.png; \*.bmp', 'Select query image'); if (query\_fname ~= 0)

query\_fullpath = strcat(query\_pathname, query\_fname);

[pathstr, name, ext] = fileparts(query\_fullpath); % fiparts returns char type

if ( strcmp(lower(ext), '.jpg') == 1 || strcmp(lower(ext), '.png') == 1 ...

|| strcmp(lower(ext), '.bmp') == 1 )

queryImage = imread( fullfile( pathstr, strcat(name, ext) ) );

% handles.queryImage = queryImage;

% guidata(hObject, handles);

% extract query image features addpath('CBIR files\')

queryImage = imresize(queryImage, [384 256]); hsvHist = hsvHistogram(queryImage);

autoCorrelogram = colorAutoCorrelogram(queryImage); color\_moments = colorMoments(queryImage);

% for gabor filters we need gary scale image img = double(rgb2gray(queryImage))/255;

[meanAmplitude, msEnergy] = gaborWavelet(img, 4, 6); % 4 = number of scales, 6 = number of orientations

wavelet\_moments = waveletTransform(queryImage);

% construct the queryImage feature vector

queryImageFeature = [hsvHist autoCorrelogram color\_moments meanAmplitude msEnergy wavelet\_moments str2num(name)];

% update handles

handles.queryImageFeature = queryImageFeature; guidata(hObject, handles);

helpdlg('Proceed with the query by executing the green button!');

% Clear workspace

clear('query\_fname', 'query\_pathname', 'query\_fullpath', 'pathstr', ... 'name', 'ext', 'queryImage', 'hsvHist', 'autoCorrelogram', ... 'color\_moments', 'img', 'meanAmplitude', 'msEnergy', ... 'wavelet\_moments', 'queryImageFeature');

else

errordlg('You have not selected the correct file type'); end

else

return; end

% --- Executes on selection change in popupmenu\_DistanceFunctions.

function popupmenu\_DistanceFunctions\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu\_DistanceFunctions (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu\_DistanceFunctions contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu\_DistanceFunctions

handles.DistanceFunctions = get(handles.popupmenu\_DistanceFunctions, 'Value'); guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.

function popupmenu\_DistanceFunctions\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu\_DistanceFunctions (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor')) set(hObject,'BackgroundColor','white');

end

% --- Executes on selection change in popupmenu\_NumOfReturnedImages.

function popupmenu\_NumOfReturnedImages\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu\_NumOfReturnedImages (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu\_NumOfReturnedImages contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu\_NumOfReturnedImages

handles.numOfReturnedImages = get(handles.popupmenu\_NumOfReturnedImages, 'Value'); guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.

function popupmenu\_NumOfReturnedImages\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu\_NumOfReturnedImages (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor')) set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in btnExecuteQuery.

function btnExecuteQuery\_Callback(hObject, eventdata, handles)

% hObject handle to btnExecuteQuery (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% check for image query

if (~isfield(handles, 'queryImageFeature'))

errordlg('Please select an image first, then choose your similarity metric and num of returned images!');

return; end

% check for dataset existence

if (~isfield(handles, 'imageDataset'))

errordlg('Please load a dataset first. If you dont have one then you should consider creating one!');

return; end

% set variables

if (~isfield(handles, 'DistanceFunctions') && ~isfield(handles, 'numOfReturnedImages')) metric = get(handles.popupmenu\_DistanceFunctions, 'Value');

numOfReturnedImgs = get(handles.popupmenu\_NumOfReturnedImages, 'Value'); elseif (~isfield(handles, 'DistanceFunctions') || ~isfield(handles, 'numOfReturnedImages'))

if (~isfield(handles, 'DistanceFunctions'))

metric = get(handles.popupmenu\_DistanceFunctions, 'Value'); numOfReturnedImgs = handles.numOfReturnedImages;

else

metric = handles.DistanceFunctions;

numOfReturnedImgs = get(handles.popupmenu\_NumOfReturnedImages, 'Value'); end

else

metric = handles.DistanceFunctions; numOfReturnedImgs = handles.numOfReturnedImages;

end

if (metric == 1)

L1(numOfReturnedImgs, handles.queryImageFeature, handles.imageDataset.dataset);

elseif (metric == 2 || metric == 3 || metric == 4 || metric == 5 || metric == 6 || metric == 7 || metric == 8 || metric == 9 || metric == 10 || metric == 11)

L2(numOfReturnedImgs, handles.queryImageFeature, handles.imageDataset.dataset, metric); else

relativeDeviation(numOfReturnedImgs, handles.queryImageFeature, handles.imageDataset.dataset);

end

% --- Executes on button press in btnExecuteSVM.

function btnExecuteSVM\_Callback(hObject, eventdata, handles)

% hObject handle to btnExecuteSVM (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% check for image query

if (~isfield(handles, 'queryImageFeature')) errordlg('Please select an image first!'); return;

end

% check for dataset existence

if (~isfield(handles, 'imageDataset'))

errordlg('Please load a dataset first. If you dont have one then you should consider creating one!');

return; end

numOfReturnedImgs = get(handles.popupmenu\_NumOfReturnedImages, 'Value'); metric = get(handles.popupmenu\_DistanceFunctions, 'Value');

% call function passing as parameters the numOfReturnedImgs, queryImage and the dataset [~, ~, cmat] = knns(numOfReturnedImgs, handles.imageDataset.dataset, handles.queryImageFeature, metric);

[~, ~, cmat] = CNN(numOfReturnedImgs, handles.imageDataset.dataset, handles.queryImageFeature, metric);

% plot confusion matrix

opt = confMatPlot('defaultOpt'); opt.className = {

'Flowers', 'Beach', 'Monuments', ...

'Buses', 'Dinosaurs', 'Elephants', ...

'Flowers', 'Horses', 'Mountains', ... 'Food'

};

opt.mode = 'both';

figure('Name', 'Confusion Matrix'); confMatPlot(cmat, opt); xlabel('Confusion Matrix');

% --- Executes on button press in btnPlotPrecisionRecall.

function btnPlotPrecisionRecall\_Callback(hObject, eventdata, handles)

% hObject handle to btnPlotPrecisionRecall (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

if (~isfield(handles, 'imageDataset')) errordlg('Please select a dataset first!'); return;

end

% set variables numOfReturnedImgs = 20;

database = handles.imageDataset.dataset;

metric = get(handles.popupmenu\_DistanceFunctions, 'Value'); precAndRecall = zeros(2, 10);

for k = 1:15

randImgName = randi([0 999], 1); randStrName = int2str(randImgName);

randStrName = strcat('images\', randStrName, '.jpg'); randQueryImg = imread(randStrName);

% extract query image features

queryImage = imresize(randQueryImg, [384 256]); hsvHist = hsvHistogram(queryImage);

autoCorrelogram = colorAutoCorrelogram(queryImage); color\_moments = colorMoments(queryImage);

% for gabor filters we need gary scale image img = double(rgb2gray(queryImage))/255;

[meanAmplitude, msEnergy] = gaborWavelet(img, 4, 6); % 4 = number of scales, 6 = number of orientations

wavelet\_moments = waveletTransform(queryImage);

% construct the queryImage feature vector

queryImageFeature = [hsvHist autoCorrelogram color\_moments meanAmplitude msEnergy wavelet\_moments randImgName];

disp(['Random Image = ', num2str(randImgName), '.jpg']);

[precision, recall] = svmclassify(numOfReturnedImgs, database, queryImageFeature, metric);

precAndRecall(1, k) = precision; precAndRecall(2, k) = recall;

end

figure;

plot(precAndRecall(2, :), precAndRecall(1, :), '--mo'); xlabel('Recall'), ylabel('Precision');

title('Precision and Recall');

legend('Recall & Precision', 'Location', 'NorthWest');

% --- Executes on button press in btnSelectImageDirectory.

function btnSelectImageDirectory\_Callback(hObject, eventdata, handles)

% hObject handle to btnSelectImageDirectory (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% select image directory

folder\_name = uigetdir(pwd, 'Select the directory of images'); if ( folder\_name ~= 0 )

handles.folder\_name = folder\_name; guidata(hObject, handles);

else

return; end

% --- Executes on button press in btnCreateDB.

function btnCreateDB\_Callback(hObject, eventdata, handles)

% hObject handle to btnCreateDB (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

if (~isfield(handles, 'folder\_name')) errordlg('Please select an image directory first!'); return;

end

% construct folder name foreach image type pngImagesDir = fullfile(handles.folder\_name, '\*.png'); jpgImagesDir = fullfile(handles.folder\_name, '\*.jpg'); bmpImagesDir = fullfile(handles.folder\_name, '\*.bmp');

% calculate total number of images num\_of\_png\_images = numel( dir(pngImagesDir) ); num\_of\_jpg\_images = numel( dir(jpgImagesDir) );

num\_of\_bmp\_images = numel( dir(bmpImagesDir) );

totalImages = num\_of\_png\_images + num\_of\_jpg\_images + num\_of\_bmp\_images;

jpg\_files = dir(jpgImagesDir); png\_files = dir(pngImagesDir); bmp\_files = dir(bmpImagesDir);

if ( ~isempty( jpg\_files ) || ~isempty( png\_files ) || ~isempty( bmp\_files ) )

% read jpg images from stored folder name

% directory and construct the feature dataset jpg\_counter = 0;

png\_counter = 0;

bmp\_counter = 0; for k = 1:totalImages

if ( (num\_of\_jpg\_images - jpg\_counter) > 0)

imgInfoJPG = imfinfo( fullfile( handles.folder\_name, jpg\_files(jpg\_counter+1).name ) ); if ( strcmp( lower(imgInfoJPG.Format), 'jpg') == 1 )

% read images

sprintf('%s \n', jpg\_files(jpg\_counter+1).name)

% extract features

image = imread( fullfile( handles.folder\_name, jpg\_files(jpg\_counter+1).name ) ); [pathstr, name, ext] = fileparts( fullfile( handles.folder\_name,

jpg\_files(jpg\_counter+1).name ) );

image = imresize(image, [384 256]); end

jpg\_counter = jpg\_counter + 1;

elseif ( (num\_of\_png\_images - png\_counter) > 0)

imgInfoPNG = imfinfo( fullfile( handles.folder\_name, png\_files(png\_counter+1).name )

);

if ( strcmp( lower(imgInfoPNG.Format), 'png') == 1 )

% read images

sprintf('%s \n', png\_files(png\_counter+1).name)

% extract features

image = imread( fullfile( handles.folder\_name, png\_files(png\_counter+1).name ) ); [pathstr, name, ext] = fileparts( fullfile( handles.folder\_name,

png\_files(png\_counter+1).name ) );

image = imresize(image, [384 256]); end

png\_counter = png\_counter + 1;

elseif ( (num\_of\_bmp\_images - bmp\_counter) > 0)

) );

imgInfoBMP = imfinfo( fullfile( handles.folder\_name, bmp\_files(bmp\_counter+1).name

if ( strcmp( lower(imgInfoBMP.Format), 'bmp') == 1 )

% read images

sprintf('%s \n', bmp\_files(bmp\_counter+1).name)

% extract features

image = imread( fullfile( handles.folder\_name, bmp\_files(bmp\_counter+1).name ) ); [pathstr, name, ext] = fileparts( fullfile( handles.folder\_name,

bmp\_files(bmp\_counter+1).name ) );

image = imresize(image, [384 256]); end

bmp\_counter = bmp\_counter + 1; end

hsvHist = hsvHistogram(image);

autoCorrelogram = colorAutoCorrelogram(image); color\_moments = colorMoments(image);

% for gabor filters we need gary scale image img = double(rgb2gray(image))/255;

[meanAmplitude, msEnergy] = gaborWavelet(img, 4, 6); % 4 = number of scales, 6 = number of orientations

wavelet\_moments = waveletTransform(image);

% construct the dataset

set = [hsvHist autoCorrelogram color\_moments meanAmplitude msEnergy wavelet\_moments];

% add to the last column the name of image file we are processing at

% the moment

dataset(k, :) = [set str2num(name)];

% clear workspace

clear('image', 'img', 'hsvHist', 'autoCorrelogram', 'color\_moments', ... 'gabor\_wavelet', 'wavelet\_moments', 'set', 'imgInfoJPG', 'imgInfoPNG', ... 'imgInfoGIF');

end

% prompt to save dataset uisave('dataset', 'dataset1');

% save('dataset.mat', 'dataset', '-mat');

clear('dataset', 'jpg\_counter', 'png\_counter', 'bmp\_counter'); end

% --- Executes on button press in btn\_LoadDataset.

function btn\_LoadDataset\_Callback(hObject, eventdata, handles)

% hObject handle to btn\_LoadDataset (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA) [fname, pthname] = uigetfile('\*.mat', 'Select the Dataset');

if (fname ~= 0)

dataset\_fullpath = strcat(pthname, fname); [pathstr, name, ext] = fileparts(dataset\_fullpath); if ( strcmp(lower(ext), '.mat') == 1)

filename = fullfile( pathstr, strcat(name, ext) ); handles.imageDataset = load(filename); guidata(hObject, handles);

% make dataset visible from workspace

% assignin('base', 'database', handles.imageDataset.dataset); helpdlg('Dataset loaded successfuly!');

else

errordlg('You have not selected the correct file type'); end

else

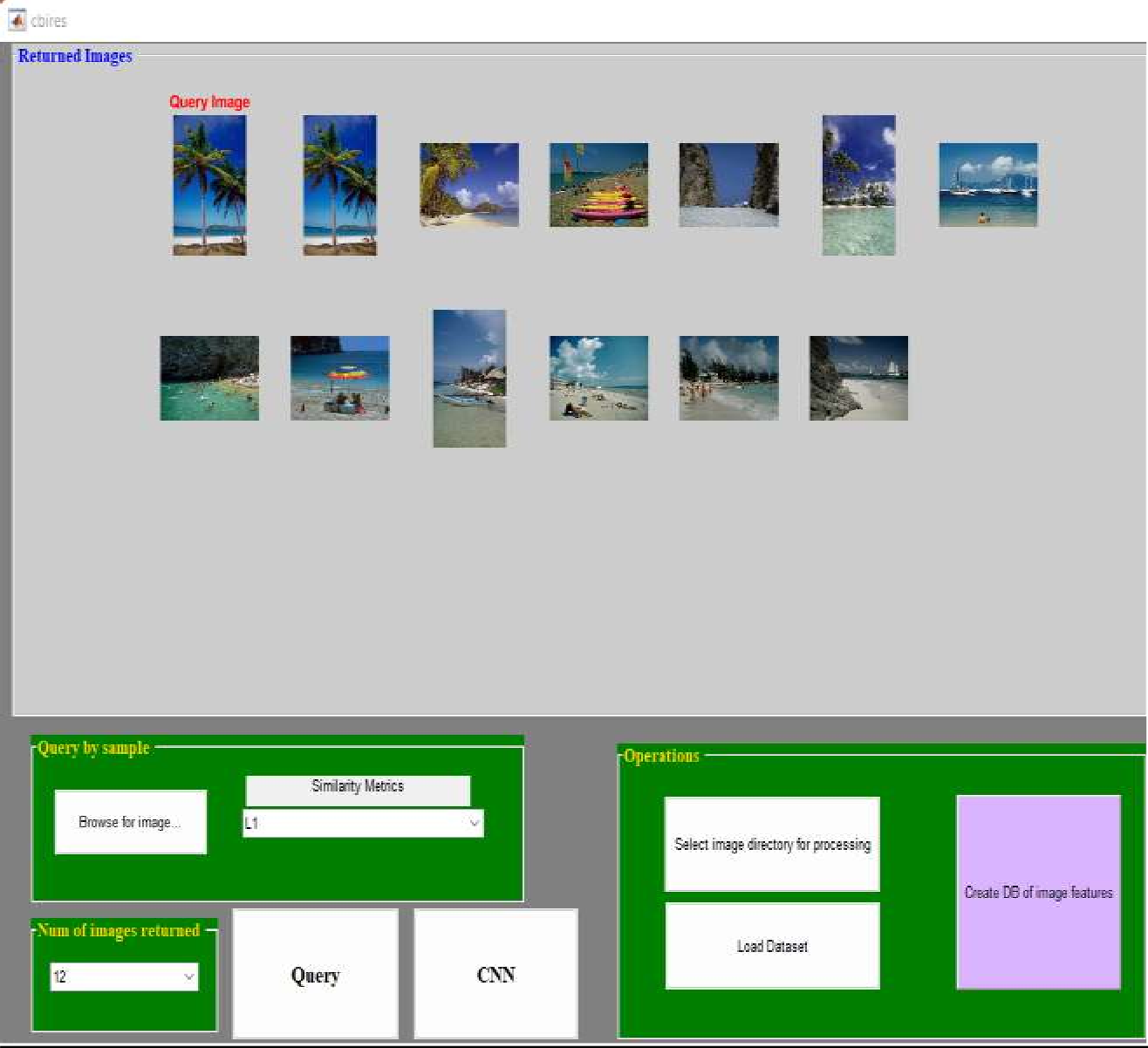
return; end

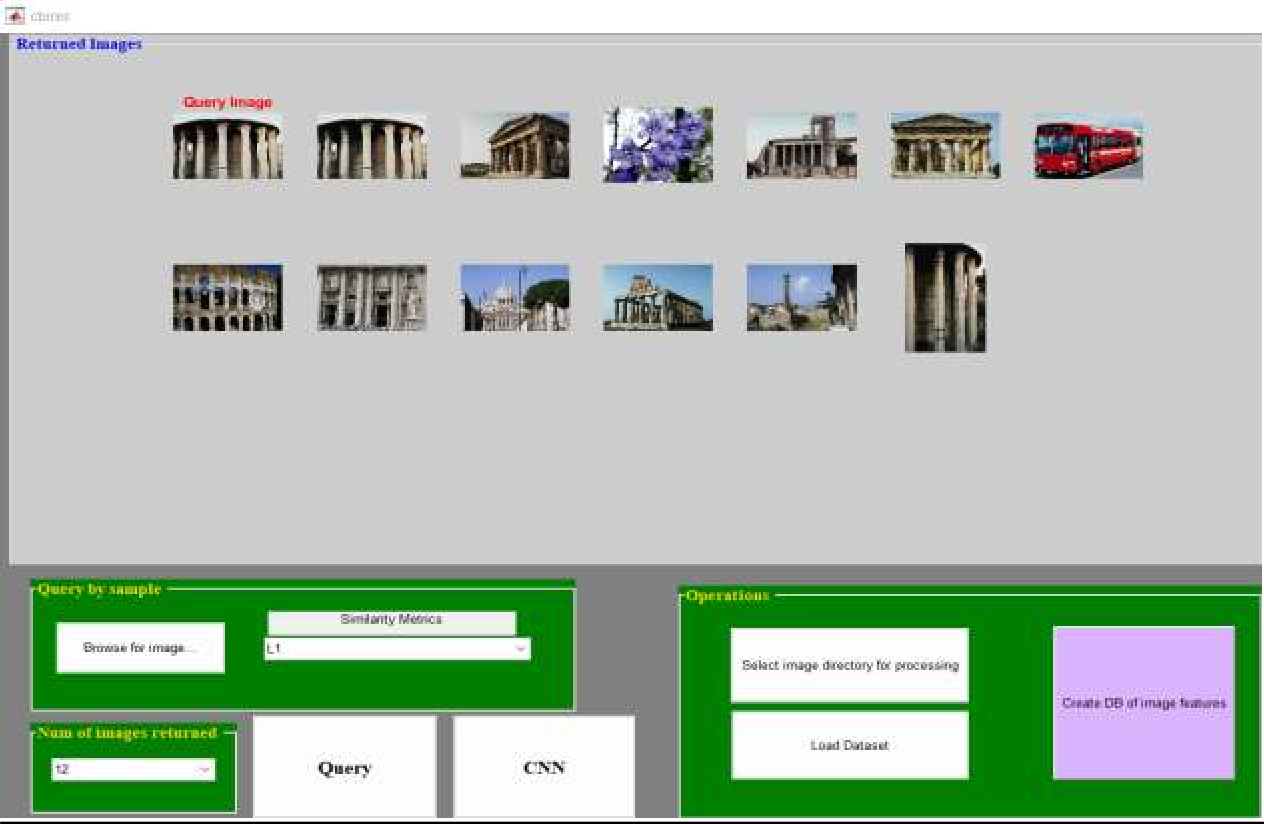


5.3

**SAMPLE OUTPUT**









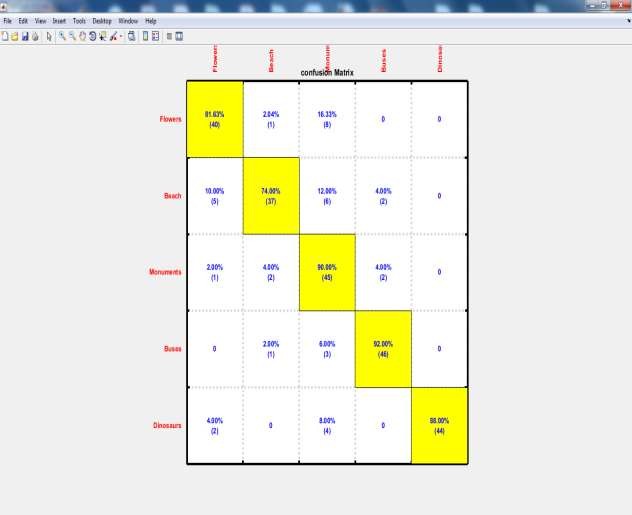


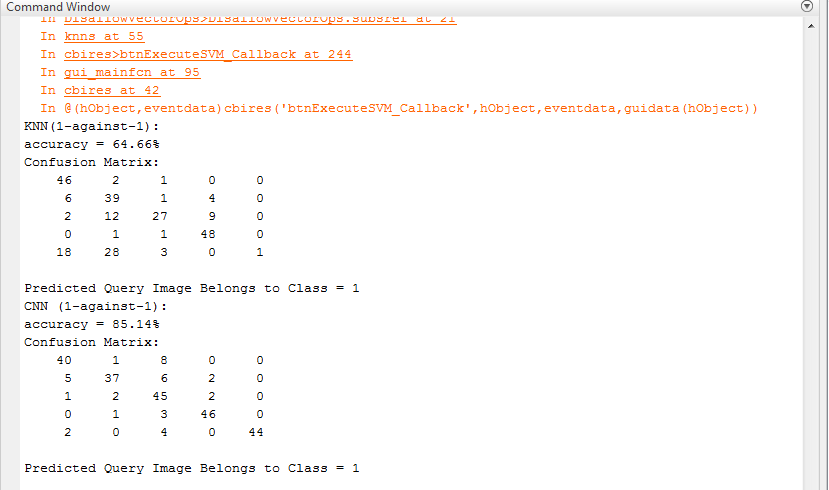


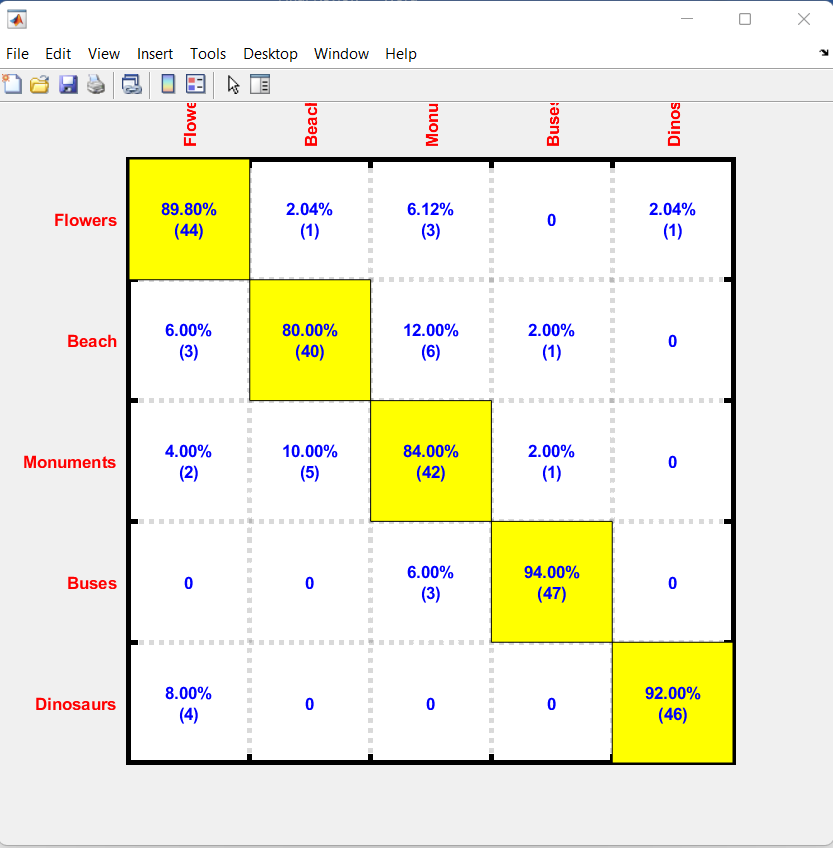
**Chapter 6**

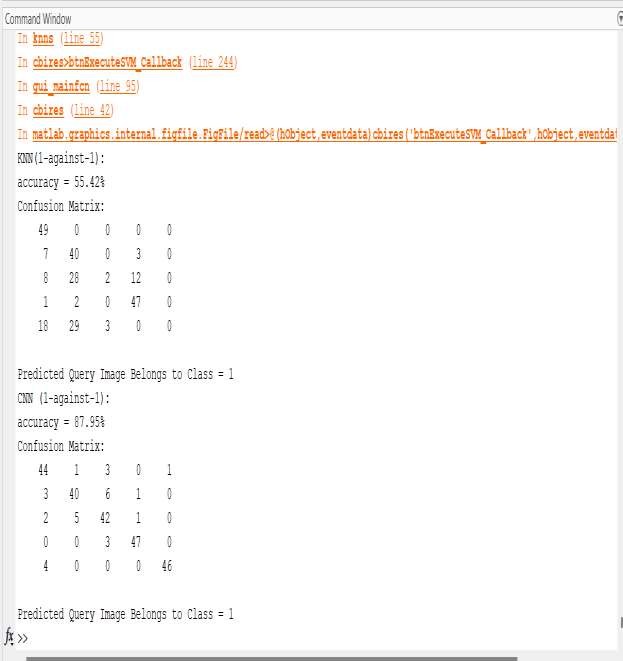
**RESULT ANALYSIS & EVALUATION METRICS**

6.1 RESULTS











Chapter 7

# Conclusion & Future Work

The updated CNN model maintains that fidelity from the initial CNN model designed exclusively for classification. The produced CNN function inherits the property of the SIFT function after learning, which is well-oriented for retrieval of images. The convolution neural networks (CNN) being a deep learning method works more effectively with more datasets. This project compared the query image with the database and look for the image matching. And will have output as graph for the separate class.



REFERENCES

1. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision (IJCV), vol. 60, no. 2, pp. 91– 110, 2004.
2. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005, pp. 886–893.
3. A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 806–813.
4. Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in European Conference on Computer Vision (ECCV), 2014, pp. 392–407.
5. A. Babenko and V. Lempitsky, “Aggregating local deep features for image retrieval,” in IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1269–1277.