DESIGN AND DEVELOPMENT OF AN INTELLIGENT IMAGE RETRIEVAL SYSTEM FOR HETEROGENEOUS DATASET

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

In

Computer Science and Engineering

By

Ramandeep Kaur

Registration Number : 41700166

Supervised By

Dr. V. Devendran (22735)

Computer Science and Engineering (Professor)

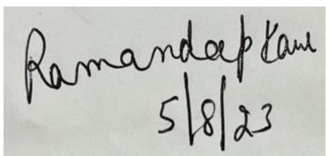
Lovely Professional University

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2023

DECLARATION

I, hereby declared that the presented work in the thesis entitled “Design and Development of an Intelligent Image Retrieval System for Heterogeneous Dataset” in fulfillment of degree of **Doctor of Philosophy (Ph.D.)** is outcome of research work carried out by me under the supervision of Dr. V. Devendran, working as Professor, in the Department of Intelligent Systems, Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.



(Signature of Scholar)

Name of the scholar: Ramandeep Kaur

Registration No.:41700166

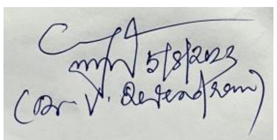
Department/school: Computer Science and Engineering

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled "Design and Development of an Intelligent Image Retrieval System for Heterogeneous Dataset" submitted in fulfillment of the requirement for the reward of degree of **Doctor of Philosophy (Ph.D.)** in the Computer Science and Engineering, is a research work carried out by Ramandeep Kaur, (41700166.), is a bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.



(Signature of Supervisor)

Name of supervisor: Dr. V. Devendran

Designation: Professor

Department/school: Intelligent System/ Computer Science and Engineering

University: Lovely Professional University

ABSTRACT

Content-Based Image Retrieval is a technique that enables to organize the digital photos according to their visual characteristics. It is used in computer vision techniques to solve the issue of retrieving images from huge databases. Content-Based Image Retrieval involves searching a database of photographs for the pictures that look like the query picture. Images are retrieved using features taken directly out of the image data rather than keywords or annotations. With the help of Feature Extraction Technique, number of required images can be extracted from the database based on the query image. Fuzzy Cmean, Kmean, Gray-Level Co-Occurrence Matrix (GLCM), Local Binary Pattern (LBP), Principal Component Analysis (PCA), and Scale Invariant Feature transform (SIFT) are used in this research study to extract features. The main motive of this work is to improve the efficiency of Content-Based Image Retrieval system by eliminating characteristics from query and database images. These extracted features undergo many processing stages, such as optimal feature selection, which gets rid of redundant information. The Markov Chain Rule Feature Optimization approach is used to choose the informative features from the features. For similarity search, Hausdarff Distance along with Gradient Boosting (GBT) machine learning techniques improves matching accuracy and retrieval speed. The tests use publicly accessible datasets termed CIFAR-10, CIFAR-100, Web-crawled misc1 and Web-Crawled misc2 and compare them to existing methodologies with performance parameters such as accuracy, precision, sensitivity, specificity and F-Score and in compare to the state of the art, we have got 97.71% accuracy and they got 93.89% and 94.79%.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. V. Devendran for always providing me his valuable time and support. Through his immense knowledge and patience, I was able to accomplish all my tasks on time. He has been actively interested in my work and has always been available to advise me. I am very grateful for his patience, motivation and enthusiasm.

I would like to express my gratitude to the Lovely Professional University (Punjab), India's School of Computer Science and Engineering for providing this doctoral programme. Throughout my research period, the institute and division were always very helpful in providing all of the necessary services and equipment.

I am also thankful to Honourable Chancellor Mr. Ashok Mittal, Worthy Pro Chancellor Mrs. Rashmi Mittal, Pro-Vice Chancellor Dr. Loviraj Gupta and other higher officials for supporting me directly and indirectly.

I am grateful to all of the reviewers who provided timely feedbacks and comments on my research publications, which helped me to improve the quality of this research. I want to express my gratitude to all researchers referenced in my thesis, whose study findings offered me with a battleground from which to choose study directions.

I want to mention here my gratitude towards my parents and also my husband Mr. Amandeep Singh who went through lot of ups and downs, but still continued to shower their endless love and care which always boosted my morale to carry forward this journey and make my efforts visible.

Ramandeep Kaur
5/8/23

Ramandeep Kaur

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABC	Artificial Bee Colonies
ANN	Artificial Neural Network
ASPE	Asymmetric Scalar Product Preserving Encryption
AUC	Area Under the Curve
CBIR	Content Based Image Retrieval
CCF	Comprehensive Color Feature
CCM	Color Correction Matrix
CNN	Convolutional Neural Network
CV	Cross Validation
DBN	Deep Belief Network
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DoG	Difference of Gaussian
DT	Decision Tree
FC	Fully Connected

FCM	Fuzzy C-Mean
FE	Feature Extraction
FN	False Negativie
FP	False Positive
FPR	False Positive Rate
GBT	Gradient Boosting
GCH	Global Color Histogram
GLCM	Gray-Level Co-occurrence Matrix
HE	Homomorphic Encryption
HIS	Hue Intensity Saturation
HOG	Histogram of Oriented Gradients
HSV	Hue Saturation Value
IRM	Integrated Region Matching
KNN	K- Nearest Neighbor
K-L	Kullback-Leibler Divergence
LBP	Local Binary Pattern
LCH	Local Color Histograms
LSVRC	Large Scale Visual Recognition Challenge
MCFS	Markov Chain Feature Selection
MI	Mutual Information
ML	Machine Learning

MSE	Mean Square Error
NN	Neural network
PCA	Principal Component Analysis
PHOG	Pyramid Histogram of Gradients
PIO	Pigeon Inspired Optimization
PLSA	Probabilistic Latent Search Algorithm
POS	Particle Swarm Optimization
PSNR	Peak Signal-to-Noise Ratio
QBIC	Query Based Image Content
RBMs	Restricted Boltzmann Machines
ROC	Reciever Operating Characteristic
SGLDM	Spatial Grey Level Dependence Matrix
SIFT	Scale-Invariant Feature Transform
SVM	Support Vector Machine
TCCNN	Three Channel Convolutional Neural Network
TN	True Negative
TP	True Positive
TPR	True Positive Rate
VV	Vector Variance

CHAPTER 1

INTRODUCTION

Technique known as content-based image retrieval (CBIR), which searches images from vast picture databases using visual contents, has been the subject of extensive research over the past ten years. There are many applications where it is necessary to identify a desired picture across several databases, including automatic face detection, finger print recognition, and medical diagnostics, to name a few. Early techniques, to retrieve images depended on the arbitrary process of manually textualizing of images and it was very time consuming process. This method, which uses terms associated with photographs to obtain visual data, has a number of drawbacks. It takes a lot of time and is really tedious. It is challenging to adequately express in words the numerous diverse picture kinds' contents. Due to the subjective nature of keywords, the semantic gap between user requests and the retrieval systems cannot be closed and it is very doubtful on the retrieval system's accuracy. The keyword used to describe photographs becomes insufficient in large datasets. The technology known as CBIR, or content-based image retrieval, is efficient. Utilizing visual clues, it searches image databases to find the required photographs. Numerous techniques and procedures are used for this. Color [19], texture [14] [27], shape [52], and other visual aspects [49] of images are carefully examined for indexing and conveying the contents of the image. These inconsequential details of an image are closely related to its content. To compare the query image to images in the database, these picture contents may be retrieved from the image and utilized in conjunction with other statistical techniques [57] [62]. Many aspects of an image query are used in content-based retrieval systems to search the database for associated picture features. A CBIR structure uses illustrative material from the photos that are displayed as low level highlights, such as shading, surface, form, and geographic regions, in order to verbally describe the images in the databases. The relevant pictures in the structure improve when an illustration image or represent is included as a contribution. By posing questions along these lines, we may be able to avoid describing the picture material in words and obtain a detailed understanding [63] of the facts used in the drawing. Several of the agent CBIR systems have a function called Query by Image Content (QBIC).

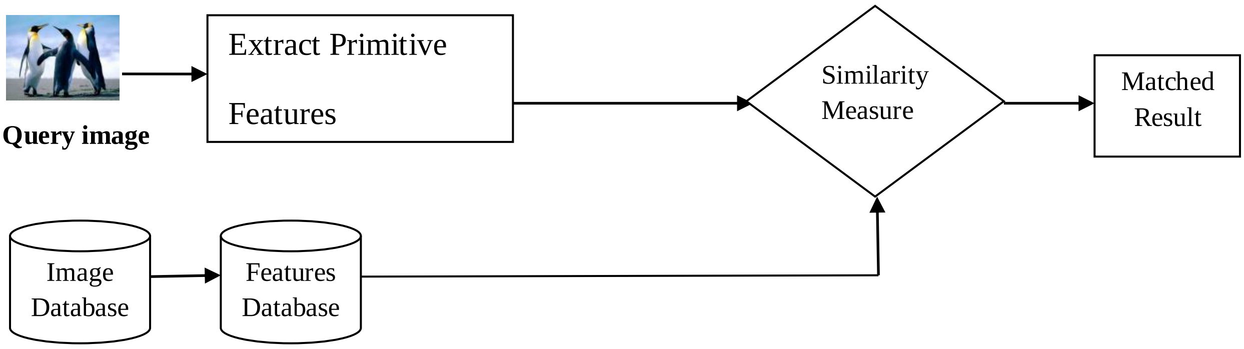


Figure 1.1: Content Based Image Retrieval Architecture

A multidimensional component vector in a CBIR system refers to visual data like shade, surface, shape, and geographic regions (Figure 1.1). An element database is created using the element vectors from the database's pictures. The restoration procedure begins when a customer submits to the building a case study or depiction of the protest. The query image is changed in accordance with the internal representation of the highlight vector using an element extraction procedure that is comparable to the one used to generate the component database. The comparability evaluation is used to split the element vectors for the objective and question images in the component database. The revitalization is finally finished using an ordering system that facilitates efficient picture database searches. Client pertinence feedback has recently been used into efforts to improve the revival process in order to provide more meaningful recovery outcomes on a perceptual and semantic level.

1.1CONTENT BASED IMAGE RETRIEVAL (CBIR)

Images can be used to overcome the constraints of text-based image retrieval systems if they can be represented as computer-accessible mathematical vectors (feature vectors), which include rich visual information like colour, texture, and shape. An interaction between a low level framework and an abnormal state framework (the human mind) is referred to as "content-based image retrieval" (CBIR). A computer can accomplish limited visual activities at far greater rates than the human brain, which is speed-limited yet capable of conducting complex visual discernment. By communicating to visual picture material in the form of naturally derived picture highlights that require no manual involvement, a CBIR eliminates the requirement for human interaction in the component extraction process. These automated include extraction techniques take a lot of time, are challenging, and usually depend on a particular place. As a result, CBIR can look at various expansions, rearrangements, and nonexclusive methods for reducing computing complexity [37].

Image processing algorithms may analyze and modify the visual data in a picture to create a feature vector. In order to produce feature vectors as small as is practicable while yet translating the visual content of photographs into the proper characteristic vectors rapidly, impressively, better, and more effectively, other image processing domains are being researched [27].

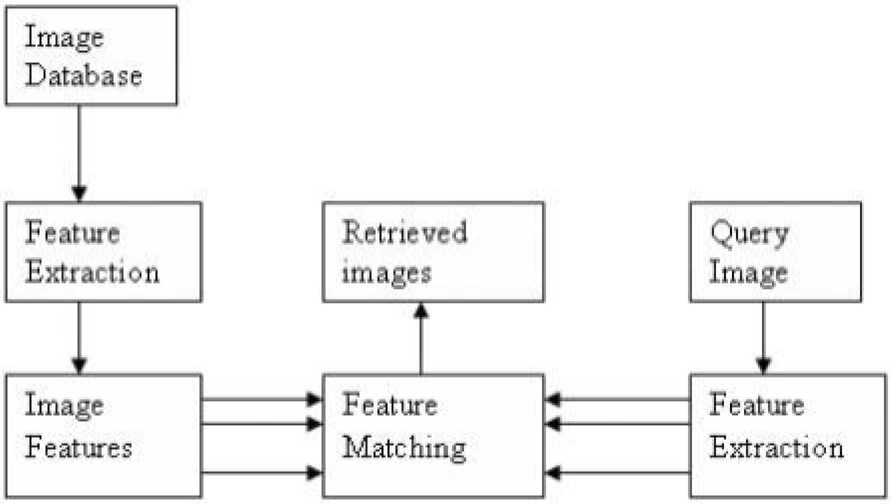


Figure 1.2: Different Steps of Content Based Image Retrieval System

Typical CBIR systems can classify and retrieve pictures from image databases without the involvement of a human by extracting attributes from the images, such as colour, texture, and form, and searching for related images with the same function vectors. Systems for retrieving images that are content-focused work in two steps. In order to properly capture the visual information of each image in the database, a collection of features, known as a feature vector, is constructed in the first stage, feature extraction (FE). A characteristic vector has a substantially smaller dimension than an established snapshot. The magnitude of the similarity between the query image and each image in the database using their characteristic vector is evaluated in the second segment in order to determine which photos are the most similar.

1.2 APPLICATIONS OF CBIR

CBIR has enough progress and certainty to be helpful for applications that can be certified, even if some of the fundamental and critical problems in the field have not yet been overcome. Online picture sharing has grown highly popular as a result of the millions of images and diverse material that are each hosted by Picasa, Flickr, Facebook, and Orkut. YouTube has ignited a new multimedia rebellion through the sharing and broadcast of videos. The present media attention given to science, as seen by articles in Scientific American, Discovery News, and on CNN, serves to showcase the potential for real-world applications of CBIR and image analysis. There are several possible applications for CBIR innovation that have been noted.

Some of these are included in the list below:

Methods for tracking geography, thermometers, ethereal and cosmic imagery, historical locations, and antiquarianism.

A few applications that might profit from this technology include medical imaging (2D/3D information), graphic design/promotion, publishing, trademark databases, criminal investigations, picture archiving and retrieval, video archiving and access, and Internet image searches.

Some software domains set restrictions on the type of picture data in order to allow for the incorporation of a priori information in the retrieval process. A visual look founded matching concept is essential since no a priori abilities may also be recognised for the retrieval of fundamental snapshot information. However, creators of modern systems usually forget to give a precise definition of matching.

The CBIR system has following steps:

1. Create a database: To get it ready for testing, use the built-in databases or store photographs in your own database.
2. Provide a Query Picture: Provide a query image to retrieve associated images from a database.
3. Feature Extraction: This technique involves querying an image and extracting its important characteristics using a variety of visual properties, such as colour, texture, and edge features.
4. Feature Matching: The features that are most comparable to those in the query image are utilised to extract the matching image of those features. The Manhattan distance, Euclidean distance, chi-square distance, and other metrics are used to compare the query image and database images.
5. Evaluate the Results: Evaluate the images that were produced using parameters like sensitivity, Precision,specificity, F_Score and accuracy rate.

.

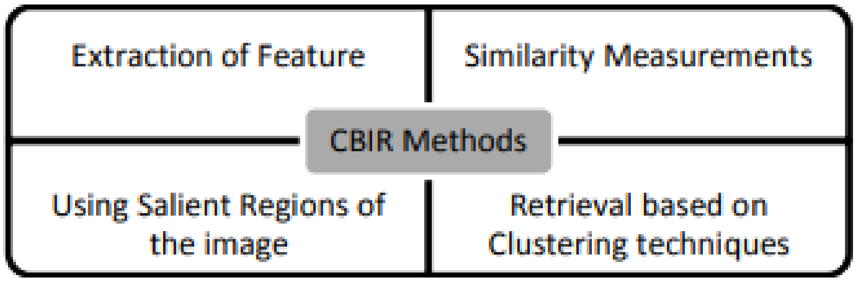


Figure 1.3 Representations of CBIR Methods

1.3 CONTENT BASED IMAGE RETRIEVAL RESEARCH PERSPECTIVE

Due to the lack of a definition of picture interpretation for the human eye as being vital in many applications of image retrieval, research in this subject is always being driven by either building on prior work or pursuing novel routes. In this method, the study focuses on mathematically characterizing images by extracting feature in the form of a vector and utilizing these descriptors to match between pairs of photos. Although there are additional study domains addressed by image retrieval, the major areas of attention are the effective production and matching of objective picture attributes.

1.3.1 Image Feature Extraction

The bulk of CBIR systems use feature extraction to preprocess their data. The way the picture is displayed as a collection of pixel ideals does not adequately reflect how humans understand images semantically. Feature extraction creates the feature vector of the image as mathematical descriptors using the raw pixel representation of the image. Advances are being made in the derivation of new features based on the visible image content of pictures in order to more correctly capture the visible picture content (such colour, texture, and shape). In order to speed up photo retrieval, methods for minimizing the function vector are also being researched. Investigating the computations of near-by, intermediate and global characteristic vectors focused on the various image areas under consideration and their usefulness in image retrieval [60]. The use of colour spaces other than the traditional RGB (which more closely mimic human vision) for characteristic vector extraction is another topic of research in CBIR characteristic extraction.

To capture the fine detail and recurrent patterns of surfaces inside a photograph, texture facets are used. Effective texture discrimination is a skill that humans possess. The usage of texture illustrations as mathematical descriptors for type feature vectors is intrinsically assigned.

Additionally, the research on the role of these texture elements in area-distinctive picture retrieval is complete. This research is particularly important because of how closely these texture features relate to the underlying semantics in contexts like aerial photos and scientific imaging. How to take an image's shape and extract its information to build function vectors is another hot field of research.

Closing the semantic gap between how computers represent images as feature vectors and how human vision systems comprehend images is the primary goal of feature extraction in CBIR. The research efforts in CBIR may be attempting to increase the efficiency and retrieval performance of image retrieval systems by enhancing feature vector extraction techniques to make them accurate and rapid. The majority of the work discussed in the thesis is focused on efficient picture feature extraction because it is one of the key study facets of content-based image retrieval.

1.3.2 Image Similarity Measures used in CBIR

How comparable two photos are may be determined by comparing the difference between their respective function vectors; unlike images do have a bigger change value than similar ones. Several similarity measures were devised in light of the numerous distinct photo retrieval frameworks that were envisioned. A variety of unique conditions, such as neighbourhood linearity, noise resistance, computation efficacy, and contract with semantics, must be met by the similarity measure. A variety of practical methods, such as the use of machine learning techniques, region-based approaches, linear and non-linear formats for similarities, and vector and non-vector feature representations, may be used to categorize picture similarity data.

The Euclidian, Weighted Euclidian, Hausdorff, Earth Movers, Mallows, Kullback-Leibler, IRM (integrated region matching), Kullback-Leibler (K-L) divergence, Mahalanobis, Vector Variance (VV), and Bhattacharyya distances are only a few of the distance metrics that have been planned thus far. Each of these methods for measuring distance has distinct benefits and drawbacks.

Over a wide range of distance measurements, there are many different reasons that vary, including the type of input system, the complexity of the calculation, and whether the measure is metric. Future photo matching algorithms may adopt a variety of novel approaches. To develop what is sometimes referred to as customized photo search, the subjectivity must be more fully included into picture matching systems. Beyond semantics, this may also involve aesthetics and individual preferences for diversity and content. For the purpose of customizing and compressing the entire spectrum of identifying images, researchers are looking for methods to improve the capacity of nonlinear representation manifolds.

1.3.3 Clustering and Classification

A set of visual components that work well on one kind of image could not work the same way on another kind of image. To solve this issue, classification and clustering are used. Three categories may be used to categorize feature vectors for images: static, picture-savvy versatile and client-savvy versatile. Static component vectors are built same for each picture. The way the photographs are grouped or bundled provides information about adaptable element vectors that are picture aware. Images are initially divided into a number of categories before being turned into feature vectors. Clustering is the creation of collections of images in image databases that are used to focus an investigation. The grouping of photos according to their informational or aesthetic value has long been a subject of research interest. Clustering allows for more efficient storage as well as speedier and quicker retrieval. Clustering is done using low level characteristics, and user involvement is limited. The process of classifying photographs using knowledge or experience from the past is known as classification. The classification technique is not intended for extensible photo databases since it depends on prior data. Photos from classes for which the classification method has not been trained are not classified. More recently, an approach [36] influenced by material science has also used semantic-delicate highlights to categorize pictures as either photographs or photos with realistic representations.

The study in the clustering and classification domain aims to make sure that the additional vigour provided by diverse element portrayal does not compromise the ordering and retrieval abilities.

1.3.4 CBIR with Relevance Feedback

Because there is not a clear framework for presenting abnormal state picture semantics independent of subjectivity of recognition, the client's criticism is the greatest source of information for understanding case-specific inquiry semantics. A system for question modification known as importance criticism looks for client and query specific semantics and alters the replies as necessary. This CBIR approach with relevant feedback calls for substantial user involvement. Given the tremendously diversified user base, the main challenge in developing such a paradigm is the growing user engagement.

1.3.5 Multimodal Image Retrieval

A type of retrieval in which a query may be expressed as a synthesis of many media is referred to as "multimedia retrieval." All of the following are supported: photo retrieval, text retrieval, free text retrieval, portrait retrieval, video retrieval, and any conceivable combination of these. Several strategies have been devised so far for obtaining the images and the text that goes with them. Even the different strategies for audio and text retrieval were devised individually. If a user needs a search that is specifically focused on multimedia, the currently employed methods created for personal media types fall short. Multimodal fusion is necessary to make such person queries entertaining. Despite the availability of two separate median dependently, it may be difficult to combine appropriate retrieval strategies for multimodal retrieval. Fusion research looks on issues related to multimodal endeavours to determine the best tactics and units to use.

1.3.6 Semantic Image Retrieval

The fact that all of the present scenarios rely on visual matching to evaluate semantic matching, which is challenging given the semantic gap between lower-level data and higher-level concepts, is one problem. The difference between the knowledge that may be drawn from observable data and the significance that the same data has for a person confronting a given circumstance is known as the semantic gap. To do this, distinct classes are developed differently for semantic image search approaches and relevance courses are created.

1.4 RESEARCH OBJECTIVES

1. To design an indexing technique using CNN Features to categorize and label the Images.
2. To extract the content-based image features using different clustering techniques.
3. To design an optimization technique to minimize the searching and matching the query image over feature heterogeneous dataset.
4. To compare the results with other standard feature extraction techniques and different matching algorithms.

1.5 THESIS ORGANIZATION

Chapter 2 discusses the previous work done in the field of the CBIR Along with it; the technological classification has been also presented.

Chapter 3 discusses about the various feature extraction and clustering Techniques.

In Chapter 4, we have computed the performance of different machine learning classifiers with different feature extraction Techniques.

Chapter 5 suggests about optimization based techniques, which helps to reduce the searching and matching the query image and we have computed the performance of our proposed optimization technique and also compared the results with existing optimization techniques.

Chapter 6 focuses on performance of different matching algorithms.

Chapter 7 concludes our thesis and gives an insight into the future scope.

CHAPTER 2
LITERATURE SURVEY

With the increase in datasets, it is becoming more difficult to extract images from databases as a result of space and time complexity. Image processing is the most effective method for detecting images from the database. Various publications presenting image retrieval are presented and discussed in this chapter. We have conducted a thorough assessment of the literature on various, feature extraction and classification algorithms for image retrieval. The chapter also discusses the research gaps found in the existing literature. A benchmark dataset used in our work has been discussed in detail.

2.1 FEATURE EXTRACTION

Features are the qualities that go into describing an object. Global features and local features are the two main categories of features. Three types of global features include colour histograms, texture histograms, and overall image colour scheme. The local characteristics for the separated regions may be sorted into texture, colour, and shape aspects as part of the segmentation process. These features are extracted from the photos and then utilized to compare the images in order to perform matching and retrieval. The feature extraction method Pyramid Histogram of Gradients (PHOG) can be useful for both color-based feature extraction and texture-based feature extraction [20]. It is also possible to extract information about texture and colour using the Color Co-Occurrence Method [28]. For the component H S I, three colour co-occurrences were made.

2.1.1 Color features

Because colour appeals to the human visual system, it is one of the most essential and fundamental visual aspects of a picture. Color is a fundamental component of the visual content. The majority of images, contents, or items in an image are easier for people to recognise thanks to their distinctive colours. Histograms, which show the amount of pixels belonging to each colour within a range, can be used to depict these images. In most circumstances, a histogram that separates the data range into identical bins is created in this circumstance. Each bin is determined by counting the total number of pixels with identical or nearly identical colour values. As a result, picture retrieval typically makes use of colour features. Several approaches have been developed to extract pictures based on colour similarity. Most of these methods adhere to the same fundamental principle. Each photograph that is kept in the database has its quality examined. Photographs may be shown with their colour histograms using the Global Color Histogram (GCH), and the distance between two photos' colour histograms can be used to determine how similar they are to one another. The photo will appear incorrectly using this approach. They are disadvantages in this situation since GCH is sensitive to colour distortion and intensity changes. Local Color Histograms (LCH), another technique, break the image into blocks and compute the histogram for each independently. As a result, these histograms show the image. Each block from the first picture is matched with a block from the second image at the same location to compare the two images. The sum of all distances will be equal to the distance between the two photographs. Since it helps to exhibit the image in more detail, this approach enables us to compare the picture areas. Color spaces are used to generate the colour histograms. The system's primary colour channels are red, green, and blue [39]. The HSV colour space is used to differentiate between chromaticity and intensity because colour histograms are sensitive to changes in lighting. A thorough colour feature that employs excessive green and red indices and even illumination to identify image [21] [29] [40] is very beneficial.

The image is transformed into the HSV and L*a*b* colour spaces. To separate the parts of H and b*, this is done. In light of this, it is conceivable to employ (ExR, H, b*), which is undoubtedly a distribution of Comprehensive Color Feature (CCF). Different parts have been separated by using the HSV [45] colour space. By matching the colour and pixel values, the backdrop is separated. To make segmentation more effective, [30] uses the HSV colour space, and [46] was used as a reference for the approach. Converting RGB images to L*a*b* colour space [22] brings them closer to human vision.

2.1.2 Texture features

The term "texture" refers to a property that is utilized to distinguish the interest areas and then classify or categorize those portions. Locating colours or intensities in an image is one way that texture provides spatial information. In essence, texture is a pattern of localized changes in the intensity of the picture. The geographical distribution of intensity levels in neighbouring areas determines texture. A point cannot have a texture. We may use adjectives like fine, coarse, grainy, smooth, etc. to describe a texture. The structure and tone of the texture often exhibit these qualities. Structure is produced by the spatial relationships between pixels, whereas tone is dependent on pixel intensity. Texture segmentation and classification are two essential words in texture analysis. The process of automatically determining the boundaries between various texture parts in a picture is known as texture segmentation. Texture classification is the process of identifying a certain textured object among various texture classes. Every place could have a distinct trait or feature. These statistical methods are useful when the texture primitives, also known as micro textures, are small. One example of this is a statistical method like GLCM, which can also compute attributes like contrast, entropy, correlation, etc. These statistical approaches work best when the texture primitives, sometimes called micro textures, are tiny. The Law's textual features provide an alternative strategy. It is really a set of convolution kernels selected from a set of vectors L5, E5, S5, R5, and W5. The letters L5, E5, S5, R5, and W5, respectively, stand for level, edge, spot, ripple, and wave. The parallel lines show a picture's texture. Text in images was analyzed using the Gabor filter [39], which combines band pass and linear filters. A Gabor filter is made up of a complex oscillation and a Gaussian Envelope function. In light of this, it determines the frequency component in a certain direction in a specific region of an image. The Gabor-based Haralick feature extraction filter has been utilized in [39] to extract texture features and identify edges. Using Gabor wavelets, it extracted features. In this case, the variables are radial centre frequency, standard deviation, and direction. To address the dimensionality difficulties, it is vital to restrict the size of the Gabor filter because they are frequently employed for signal processing. Eight characteristics are retrieved using the traditional Haralick method at a 45° angle of rotation. A variation of this proposes feature extraction with every 30 degree rotation. The aforementioned version enables the extraction of 13 distinct traits as a consequence. As a result, identifying leaf traits is easier. The grey level colour co-occurrence method (GLCM) was one of the first techniques to evaluate the spatial relationship between two pixels [40, 41]. It is also known as a "gray-level spatial dependence matrix", counting the instances of two pixels with particular values and spatial relationships in a picture. The importance of contrast, correlation, energy, entropy, and homogeneity are only a few of the important considerations.

2.1.3 Grey Level Co-occurrence Matrix

The grey level co-occurrence matrix (GLCM) approach, also known as the spatial grey level dependency matrix (SGLDM) methodology, focuses on pixel pairings in specific spatial linkages and takes second order statistics into consideration. To achieve this, co-occurrence matrices are employed. By comparing the grey levels of two neighbouring pixels, the GLCM may be used to derive the texture of an image. A combined distribution of grey level pairings between neighbouring pixels in a picture is produced by the GLCM. In order to describe visual textures, Haralick proposed statistical formulae that might be obtained from the co-occurrence matrix. The structure of the visual texture may be inferred by statistically selecting the pattern of grey-levels in respect to other grey-levels. Weighted averages serve as a good representation of the normalized co-occurrence. The relative significance of an item may be described by multiplying its grid components by a weighted mean multiplier. The dimension N of the aforementioned matrix represents the quantity of grey levels in the image. The possibility that a single pixel with value I will be next to a pixel with value j is represented by the matrix element [I j], which is derived by dividing the matrix by the sum of these evaluations. Each result in the matrix therefore represents the possibility that a pixel with value I will be located close to a pixel with value j. The image is used to create the equation for the specified co-occurrence matrix (i, j).

2.1.4 Local Binary Pattern (LBP)

By partitioning the space around each pixel into binary values, the Local Binary Pattern (LBP) texture operator [42] provides labels to individual pixels in a picture. LBP system may be seen as a blend of morphological interpretations, usually inconsistent statistical interpretations, and texture evaluation. The LBP operator's most significant feature in practical applications may be its unexpected nature mixed with monotonic level changes, such as brightness variations. Its operational simplicity, which enables it to analyze photographs under challenging real-time circumstances, is another significant aspect. On the premise that texture has two balanced aspects, a pattern and its strength, both of which are present locally the basic LBP operator was developed. The operator uses the centre value as a threshold and operates in a 3x3 neighbourhood. The thresholded values and the values provided by the relevant pixels are added to create an LBP code. Since the neighbourhood is made up of 8 pixels, $(2)^8 = 256$ different labels may be created by comparing the grey values of the epicentre with the surrounding pixels. The mean of the grey levels above (or equal to) the centre pixel is subtracted from the mean of the grey levels below it to get the contrast measure (C). The contrast is initialized to 0 if all eight of the centre pixel's thresholded neighbours have the same value. (0 or 1). LBP code variants are used as segmentation or grouping characteristics. The LBP operator turns a picture into a set of integer labels that, on a small scale, reflect the image's finer characteristics. Then, for further analysis, these labels or their attributes, the distribution being one of the most common are added to the image. In addition to the generally used grayscale still photographs, the operator has been expanded to handle colour (multi-channel) images, movies, and volumetric data.

2.1.5 Scale Invariant Feature Transform (SIFT)

SIFT [23] is used to represent local attributes via feature vectors. It employs an image-based picture descriptor for recognition. It made use of picture-based key points and descriptions to identify items.

2.1.6 Shape features

What we interpret as having visual properties in photos are shape features. The classification of shape characteristics may be done using the border and region parameters. These categories can also include strategies for extracting form features from regions and boundaries. A collection of numbers that are produced to represent a given form characteristic is known as a shape descriptor. A descriptor seeks to examine the form qualities in a way that is congruent with human perception. These descriptors often take the shape of vectors. Region-based techniques and contour-based methods are two categories into which these descriptors may be categorized. In feature extraction approaches based on borders, basic feature descriptors like perimeter, diameter, eccentricity, curvature, etc. are incorporated. Additional descriptors for the extraction of boundary features include statistical moments and Fourier descriptors. The feature extraction approaches based on regions include the regional area, roundness, regional focus, and topological descriptors. Border and region-based descriptors are both helpful and interchangeable, which essentially means one may be used as the foundation for calculating the other. For the identification and categorization of items, shape representation is a crucial issue. The depiction of form is provided with respect to the relationship between such components. These characteristics match the things spatial, morphological, and dimensional characteristics. Every image that is saved in the database has the form characteristics added to it. Later, additional systems or tactics for shape representation make advantage of these shape qualities to organize the helpful data in index structures for systematic retrieval. The connecting edges or borders, for instance, express the properties of the form object. As a result, the shapes edges may be manipulated. These shape boundaries are afterwards automatically divided into a set of boundary points, which are frequently applied in machine vision techniques for form identification. For some applications, a particular set of characteristics may result in the greatest results. In the event of a different set of applications, they might not, though. Therefore, the shape representation should only fetch those attributes that experts deem essential for that particular application. The properties of the form must adhere to a number of conditions. They must first establish powerfully distinguishing traits. The items key qualities should be preserved when the form representation estimates the things. The features of the shape should not be altered by transformations like scaling, translation, and rotation. They must be able to distinguish between intuitively similar but not technically related goods. They need to be free of all forms of deformities. Additionally, they need to be easy to deduce. The effectiveness of any shape feature extraction method depends on the pre-segmentation effect. When designing a complete Content-Based Image Retrieval system, it could be challenging to utilise an object's shape efficiently (CBIR). The shape has a significant impact on finding related objects in a photograph.

Shape attributes have not advanced as much as colour and texture components because of their innate capacity to represent pictures.

2.1.7 Clustering based segmentation

The clustering-based approaches are unsupervised, which sets them apart from classification algorithms. The user cannot be accessed with the pre-selected set of features or classes in this scenario. The essential hidden information in the data, such as clusters and groups that are frequently unidentified from an analytical perspective, is extracted from the data with the use of clustering-based algorithms. These techniques are intended to divide the image into clusters or different groups of pixels with related characteristics. These pixels are grouped together into bundles or clusters in order to make each cluster's constituent parts more comparable to one another than those of other clusters. K-means clustering and Fuzzy-C means clustering are two of the various clustering algorithmic approaches (FCM). Because it is simple and efficient in terms of computing, K-means clustering is a recommended and often used approach. One form of the process is the K-means clustering approach, which requires less repetitions overall. In contrast to K-means clustering, a new method called FCM permits the linkage of pixel points with several classes with varied degrees of association or membership. An FCM algorithm version optimizes the steady processing time.

- **K mean Clustering:** One of the most straightforward unsupervised learning methods for resolving clustering concerns is K Mean [20]. Finding the centroid of k and assigning points to the clusters are the two key components. A displayed image is classified using a specified count of a priori clusters. The method now begins by splitting the image space into k groups of k centroids, or k pixels. The next step is to allocate each pixel to a group based on its size or distance from the cluster. The Euclidean distance is considered in this situation. Once every single pixel has been assigned to each cluster, the centroids are shifted and then redistributed. Until the centroids can no longer shift, the same operations are repeated. When this algorithm converges, K groups are found in the areas of the image where the individual pixel values exhibit some degree of resemblance. The minimization of an objective function is required for the K-means clustering method. One of the distinctive features of K-means clustering is the complete mutual exclusion between the clusters and the components.

- **FCM:** The linking of the pixels or data points to a number of clusters is made possible by this approach. In contrast, a group of pixels may be a part of several clusters, while their levels of involvement within each group may vary. An optimization function is used by an FCM algorithm, and its minimization indicates the point at which the program converges. The areas of the image where the individual pixels show some degree of resemblance but also show a different degree of membership or affiliation with the other groups are where C groups are formed when this method converges. Using the traditional Fuzzy C-means approach of clustering; a portion of the picture may be split into many clusters or many clusters. In this case, it is important to minimize an objective function.

Calculated is the grey distance [31] between the cluster centre and the pixels. By doing so, the fuzzy associatively value of the pixel is determined, which benefits segmentation in general. Indicators of poor fuzzy associatively include strong grey distances, whereas indicators of high fuzzy associatively include low grey distances. In order to improve the target function, FCM uses weighted grey and spatial qualities as part of the spatial information [31].

2.2 CLASSIFICATION

This approach of supervised learning is used in statistics and machine learning, where the computer programmed learns from the data given to it first before making observations and categorizing them. The aim of a classification technique is to categorize or classify a given set of data. Data may be categorized both in organized and unstructured formats. Identifying the class of the presented data points is the first step in understanding this phenomenon. Labels, categories, or goals are frequently used to express the classifications. Problems including face identification, voice recognition, handwriting analysis, document categorization, and other common classification issues are among the difficulties covered. The categorization can also be divided into binary or multi-class categories. A binary classification has only two practicable outcomes while a multi-class classification consists of more than two outcomes.

2.2.1 Support Vector Machines (SVM)

SVM is a linear model that may be used to address problems with classification and regression. It may, however, resolve both linear and non-linear problems. It is a classifier that represents the training data as points in a space that has been as extensively split into classes as is practical. It seeks a line that separates the data from the two classes, in other words. When possible, this technique generates a line that isolates or separates those groups from the input data. After indicating the class and space they should go in, the most recent points are then added to the area. SVM [28] [51] is a supervised learning method that detaches the data points of classes and may be used to solve classification and regression problems. The biggest hyper plane, which is caused by the greatest distance from the closest training data point, identifies good separation. The advantage of SVM is that it performs well in high dimensional settings.

2.2.2 K nearest neighbor (KNN)

Knowledge-Based Neural Networks, or KNN [39], is a technique that may be used to address classification and regression-related problems. This approach presupposes the presence of neighbouring items that are comparable. When there is little or no prior understanding of data layout, it is frequently used. The data has to be loaded first. After that, k's value is initialized. The number of iterations is performed, starting at 1 and increasing up to the total amount of training data points. The distance between each line in the training data and the test data is then determined (typically as a Euclidean distance). According to the distance values, the estimated distances are then ordered in ascending order.

2.2.3 Logistic Regression

Logistic regression is a crucial part of regression analysis. Before learning about logistic regression, let's first talk about regression analysis. In essence, it looks for correlations between dependent and independent variables using a predictive modeling method. Regression analysis includes several subcategories, including linear regression and logistic regression. When using logistic regression to predict binary outcomes, only two classes are viable. In most cases, binary class categorization issues are the only ones that can be solved using logistic regression, sometimes referred to as binomial logistic regression. It may also be used for multiclass classification, though. It also goes by the name of multinomial logistic regression. Logistic regression uses an equation, much like a linear regression.

2.2.4 Naïve Bayes

There are really other supervised learning algorithms using the same basic idea as Naive Bayes [2]. There is a misguided notion that every pair of qualities must be conditionally independent. In other words, it indicates that none of the features in a class are necessary for the existence of any one attribute. This approach is quite helpful when working with huge datasets. This classifier's theoretical underpinning is the Bayes theorem.

2.2.5 Artificial Neural network

The neural architecture of the brain is fundamentally used to create primitive electrical networks known as artificial neural networks. Each record is handled individually, and its classification is contrasted with the research classification. Errors are communicated back into the network for each record in order to modify the algorithm for later iterations.

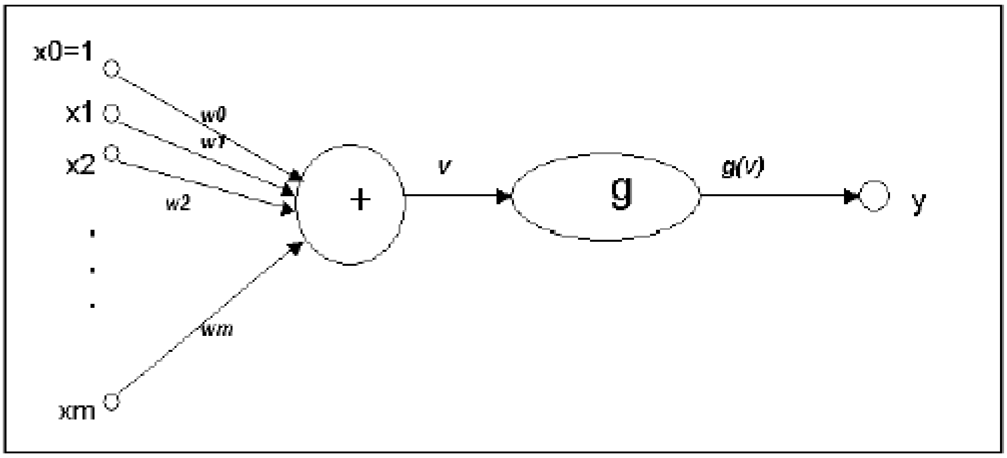


Figure 2.1: Artificial Neural Network

The input values are represented by xi, while the related weights are represented by wi. Using a function (g), the weights are added together, and the output displays the result (y). During the training session, the actual class to which each record truly belongs is known. Real values can be sent to the output nodes.

The value 1 might be given to the correct class, while a 0 could be given to the incorrect class. One of ANN's most distinctive aspects is iterative learning, which is a crucial part of the technology. The input data is transferred or fed to the network one record at a time, and any related weights are also adjusted. The same process is carried out afterwards. By changing the weights, the neural network may teach itself to predict the proper class label for the incoming data. Its three primary layers are the input layer, the hidden layer, and the output layer.

2.2.6 Convolutional Neural Network (CNN)

Convolutional neural networks, a type of neural network, are similar to other neural networks in all ways (as shown in Figure 2.2). CNN was created especially for the analysis of visual data. Their organisational structure is hence more streamlined and composed of two key parts. A convolutional neural network is composed of four different kinds of layers: the convolutional layer, the pooling layer, the ReLU correction layer, and the fully connected layer.

- **The Convolutional Layer:** The convolutional layer, which is always at least the first layer, is a crucial component of CNN. Its objective is to choose traits from the photographs that are provided to it as input. Convolution filtering is used to do this, in which a window displaying a characteristic is "dragged" over the image to determine the convolution result between the characteristic and each component of the digital image. A characteristic is viewed as a filter in this sense, and the two ideas are interchangeable. As a result, the convolutional layer computes the convolution of each input image with each filter using a variety of input images. The filters compare the characteristics of a tree to those we want in our photographs. For any combination of an image and a filter, we have a feature map that displays the location of the features in the image. The region of the image that corresponds to the higher value has a stronger reflection of the feature.

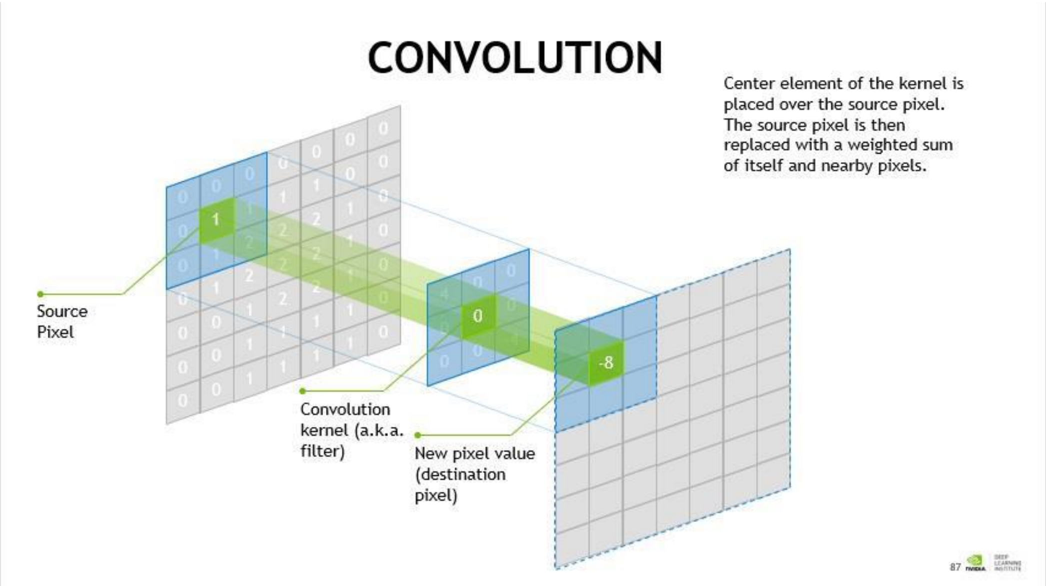


Figure 2.2: Convolutional layer [69]

- **The pooling layer:** This layer is frequently stacked between two convolutional layers. There are several approved feature maps, and the pooling function is used on each of them. The size of the photos remains largely same during the pooling process but is decreased. The image is split into regular cells in order to do this, with the highest value from each cell being saved. Tiny rectangular cells are generally employed in practice to prevent losing too much data. The most common possibilities are 3x3 cells that are separated by a 2 pixel step or 2x2 neighbouring cells that don't cross over (thus overlapping). The pooling layer of the network reduces the amount of calculations and variables. By doing this, over learning is prevented and network efficiency is increased. To identify a dog, for example, we don't even need to know where its ears are precisely; all we need to know is that they are almost directly above the head.

- **The ReLU correction layer:** ReLU (Rectified Linear Units) refers to the real nonlinear function that ReLU(x)=max defines (0,x). The ReLU correction layer replaces any negative input values with zeroes. It serves as a means of activation. It is shown in Figure 2.3.

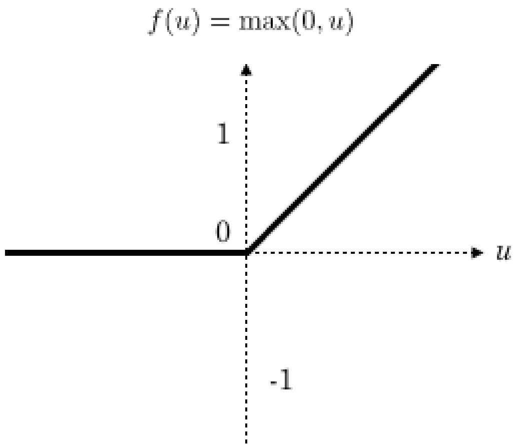


Figure 2.3: ReLU

- **The fully-connected layer:** The top layer of a neural network, whether convolutional or not, is frequently a fully-connected layer, which is not exclusive to CNNs. This layer creates a new output layer using a vector as its input. To do this, the incoming input data are processed using a linear combination and maybe an activation function. The last completely connected layer assigns the picture to one of the N categories in the described image processing challenge, classifying it as an input to the system. The likelihood that a certain category will fit the provided image is represented by each element of the vector. Each input component is multiplied by its weight in the fully-connected layer, the results are added, and the probabilities are calculated using an activation function (logistic if N=2, softmax if N>2). Here, the weights matrix is increased by adding the input vector in the opposite direction. The idea of being "fully-connected" means that all input values are related to all target values. Gradient back-propagation is used to learn weight values in a manner similar to how convolution layer filters are learned by CNNs. A class and feature locations in the image are connected by the totally connected layer. The entry table also links to a feature map for a particular feature since it is the outcome of the layer preceding it.

The high values indicate the feature's placement in the picture, which is more or less correct based on the pooling.

2.2.7 Deep convolution neural network (DCNN)

For autonomous learning from practice datasets, a CNN is utilized. Multiple layers and neurons are possible in CNNs. Deep Convolution Neural Networks (DCNN) [21], [41] is a classification system with four layers that includes a convolution layer with 20 filters, a layer with 100 filters, a layer with 1000 filters, and a layer with 1500 filters. Four neurons are used by the SoftMax function in the output layers by using DCNN [43]. CNN was used to classify different features of images [32].

2.2.8 Three Channel Convolution Neural Network (TCCNN)

Three colour channels, three entirely connected CNNs, Softmax, and an output layer make up a particular sort of CNN. TCCNN has been used to forecast vegetable leaf disease [16]. One of these components is sent to the TCCNN channels. The convolutional features of each CNN are subsequently fed into the subsequent convolutional layer and pooling layer. After that, a fully linked fusion layer is utilized to merge the traits to produce a deep level feature map that may be used to identify images. In the end, categorization is done using a softmax layer.

In the past few years, some fields have used more than twenty features. However, in recent years, research projects have used thousands of features. Due to many functions, there are challenging tasks because of irrelevant and verbose variables.

In contrast to its [37] linked keywords or labels, CBIR incorporates an image's contents or qualities, such as colour, texture, and borders. The literature on diverse picture retrieval systems is thoroughly reviewed in this book, along with the underlying concepts, useful methods, and unmet research requirements. In this paper, retrieval techniques using HSV, Color Moment, HSV and Color Moment, Gabor Wavelet and Wavelet Transform, and Edge Gradient are examined and used. The combination of colour, texture, and edge information in a picture is recommended as a retrieval technique. Several factors, including sensitivity, specificity, retrieval score, error rate, and accuracy, are utilized to assess how well the recommended approach and the investigated image retrieval algorithms work. The results of a performance evaluation experiment demonstrate that the recommended strategy outperforms alternative approaches.

Due to the massive amount of data being collected, the deep belief network (DBN) deep learning approach is employed in article [19] to extract the features and categorize the data. The simulation results comparing the proposed approach to a control group show a considerable improvement in performance. The most recent methods for image retrieval and image processing were covered in article [27]. One of the most fascinating and fast growing areas of image processing research is content-based picture retrieval (CBIR). Some of the approaches under investigation include object-based picture retrieval, Bayesian image retrieval systems, content-based image retrieval employing a mix of multimedia components recorded with metadata, and image retrieval augmentation.

In article [60] of this publication, we offer a method for retrieving images from a database of engineering/computer-aided design (CAD) models. The programmed also makes use of an image's shape information in addition to its 3D information. In Paper [36], they presented a review of the literature on Texture, Color, Shape, and Region-based Content Based Image Retrieval (CBIR) techniques. They also look at some of the most advanced CBIR tools.

They introduced[52] semantically sensitive integrated matching for picture libraries, an image retrieval system that employs semantics classification approaches, which is based on integrated region matching, image segmentation, wavelet-based feature extraction, and simplicity. An image's representation is made up of a number of regions that roughly correspond to objects and may be identified by their colour, texture, form, and location, just like in earlier region-based retrieval systems.

For region-based picture retrieval, a fuzzy logic technique known as UFM (unified feature matching) was presented in a publication [49]. Their retrieval technique represents each image as a group of segmented parts, where each segment is identified by a fuzzy feature (fuzzy set) that reflects the characteristics of colour, texture, and form. As a result, an image may contain a number of loosely defined location-related properties.

Via completely using similarity information, a novel approach known as cluster-based retrieval of images by unsupervised learning was developed in article [57] to increase user engagement with image retrieval systems (CLUE). By using a graph-theoretic clustering technique on a set of photographs close to the query, CLUE may be able to extract image clusters. In CLUE, clustering is dynamic.

They created a new model they called TSI-pLSA by extending pLSA (as used to visual words) to incorporate spatial information in a translation- and scale-invariant manner in their paper [62]. Their method can manage search engine results with a high percentage of pointless photos and a substantial intra-class fluctuation.

A unique low level characteristic that might be used for indexing and retrieval is extracted from the photographs in article [63]. Using a property called "Color and Edge Directivity Descriptor," colour and texture are integrated in a histogram.

In the study [47], three feature vectors for picture retrieval were provided. Additionally, a feature selection technique is demonstrated to choose the best characteristics in order to improve detection rates and simplify calculations associated with picture retrieval. The first and second picture characteristics in this study, the difference between pixels of scan pattern (DBPSP) and the colour co-occurrence matrix, or CCM are based on colour and texture information.

A unique approach for teaching discretized local geometry representation based on minimizing the average re-projection error in the space of ellipses was described in article [54]. The representation of each feature only needs 24 bits.

They [38] take the significant picture marking issue into account. Profound Convolutional Neural Networks (DCNN) is among the best-in-class methods for thick picture marking problems due to their ability to learn specific highlights (such as multi-class semantic division). In this study, they provide a thick picture naming method that utilizes DCNNs in conjunction with a help vector classifier. They compare highlights to a wide range of names derived from varied, unique datasets with explicit forecasting aims using a classifier based on DCNN outputs. The primary goal of employing a help vector classifier is to evaluate the efficacy of utilizing different kinds of representations to foresee class names that are not specifically relevant to the desired task.

One of the [61] issues with PC vision is providing a pixel-by-pixel dense labelling of a given image as the output. One instance of thick picture marking is the work of semantic separation. The goal of semantic division is to provide a name to each pixel in a picture that is consistent with the item classes to which it belongs. The objective of mathematical naming, a distinct strategy, is to identify each pixel in accordance with its mathematical class (for example sky, vertical, flat). Learning a classifier that categorizes each pixel according to a variety of physically based categories is a common method for solving these challenging picture tagging challenges

In this article's [14] image retrieval system for privacy-preserving pictures, the Asymmetric Scalar Product Preserving Encryption (ASPE) and Homomorphic Encryption (HE) approaches are created in order to accomplish safe search for the encrypted image retrieval system. K Nearest Neighbor is used in Asymmetric Scalar-Product Preserving Encryption (ASPE), which considerably raises processing costs. Asymmetric Scalar Product Preserving Encryption (ASPE) used the K-means algorithm to streamline the descriptors of a big database with more than 10,000 photos in order to speed up search times. This method additionally included trapdoor validation to make sure the trapdoor was real during the searching phase. Therefore, our solution provides more secure photo retrieval in the cloud by combining Asymmetric Scalar Product Preserving Encryption (ASPE) with homomorphic encryption (HE).In our system, a single vector served as the representation of each image. However, it has a considerable impact on the high dimensional descriptor's processing costs especially when the encrypted function is used.

A novel and successful technique for content-based picture retrieval was developed by Banharnsakun, et al. [20] by integrating gray-level co-occurrence matrix with the artificial bee colony, or "GLCM-ABC." Artificial bee colonies (ABC) were used to identify and recover the particular type of material surface, and to extract the components of texture, the gray-level co-occurrence matrix was used. Objective of this study was to improve photo retrieval accuracy compared to existing techniques. From computed results, they conclude that the hybrid level co-occurrence matrix with the artificial bee colony approach works better than other conventional techniques. The capabilities of the suggested solution could yet be improved, though.

In this study, a Content Based Image Retrieval (CBIR) system, which uses a genetic algorithm and annealing simulation was suggested by Alsmadi et al. [28] to get the required images from databases. Specifically, to extract the color characteristics, YCbCrcolor, discrete wavelet transform and Canny edge histogram were used; for shape features, RGB color were extracted by using neutrosophic clustering technique, and Canny edge method; and texture features were extracted using Grey Level Co-Occurance Matrix (GLCM). Then, by using a similarity metric based on a metaheuristic algorithm, photos connected to the query image were successfully retrieved. In Corel image datasets, the suggested Content Based Picture Retrieval (CBIR) method beat competing systems and demonstrated significant retrieval image outcomes in terms of recall rates and accuracy. In terms of recall rates and accuracy, it also fared better overall than other current systems. Future research on filtering techniques is advised so that the Content Based Image Retrieval (CBIR) can produce more precise findings.

In this investigation, the multi-extraction was carried out by Garg and Dhiman [39], who employed a Particle swarm optimization to eliminate majority of distinguishing features from collected data. On a COREL dataset with ten categories, classification tests were performed, and the results were presented using precision, recall, measure and accuracy as the four presentation metrics. Three well-known classifiers, including decision trees, support vector machines and K-nearest neighbors, were compared for validation purposes (KNN). There are four steps in the suggested procedure. The first was decomposition where channels R, G, and B, each received a separate multi-scale decomposition using Discrete Wavelet Transform (DWT). The second used the collection of functions to concatenate the R, G, and B channels.

The third step was a feature reduction procedure that used the Particle swarm optimization algorithm to get the prominent features. Final step involved classification in which the category of the examined photos was evaluated using three classifiers. Support Vector Machine (SVM), which had high parameter values for every performance criterion, was the best optimizer, according to the trial findings. The feature dimension, however, was little and required a high calculation cost.

Filters, wrappers and embedding methods are three basic classes that make up FS technology. [40]. A filtering technique considers only data properties by disregarding the classifier type. In a filter-based strategy, Relief [54] is an iteratively monitored approach that estimates the quality of functionality based on how close their values can distinguish data samples from each other. Another filtering method is Fisher's method [63]. It calculates the trait's score as the ratio of interclass separation to interclass variance and scores the trait individually. The Mutual Information (MI)-based approach considers mutual information between values distribution for a particular characteristic and membership in a particular class [45]. In unsupervised learning, the method that is primarily used is the Laplacian Score for geometric structure. The nearest-neighbor graph is constructed for multi-cluster data, MCFS (LS) [29], where important features are determined by their locality power [53]. They take into account a norm regularized discriminative feature selection (UDFS), which chooses the features with the highest discrimination out of the whole collection of features. In this method [49], each feature acts as a node in the graph for the feature selection; hence a path is used for the most important features based on the centrality score. For the wrapper method, SVM-RFE(RFE) [40] which uses a sequential backward elimination method to select features assigns a feature a high rank only if it completely divides the sample using a linear SVM for embedded techniques. The Haralick Features, also known as the Gray-Level Co-Occurrence Matrix (GLCM), Local Binary Pattern (LBP), and Histogram of Oriented Gradients (HOG) features, are retrieved in this [32] research effort. As a result, this study uses the feature selection technique known as Pigeon Inspired based Optimization (PIO).The same processes used for training are applied when a user submits a test collection of photos. The suitable images are then discovered by applying artificial intelligence to compare the test set of images with the training images. In this study, contour-based shape feature extraction approaches and picture moment extraction methodologies are used to obtain form [16] features and shape invariant features. Color, grey, texture, and shape features are mixed with extracted features to create the informative features utilizing PSO. The random forest classifier has been trained to find the target image for the provided query image. These extracted features go through multiple layers of processing, such as optimal feature selection for redundancy removal and optimal weighted linear combination for fusion.

Detail of the journal/ Book / Book chapter/ website link	Year of Publication	Indexing (Scopus/ SCI index etc.)	Main findings or conclusion relevant to proposed research work
Zhu C, Bichot C, Chen L) Multi-scale color local binary pattern for visual object class recognition.Inproceedings of International Conference of Pattern Recognition.	2010	SCI	The experimental results on the PASCAL Visual Object Classes (VOC) 2007 image Benchmark showed that these proposed novel operators have gained significant accuracy improvement, and are more promising for real-world object recognition Tasks.
R JVCI, Guo J, Prasetyo H, Su H.Image Indexing using the color and bit pattern feature fusion. Journal of Visual Communication Image Representation	2013	Scopus	To encode the local visual appearance, an approach based on vector quantisation (VQ) is introduced. The distributions of the VQ code indices are then used to index/retrieve the images. The new method can not only be used to achieve effective image indexing and retrieval; it can also be used for image

			compression. Based on this method, indexing and retrieval can be easily and conveniently performed in the compressed domain without performing decoding operation.
Walia E, Vesal S, Pal A. An effective and Fast hybrid framework for color image retrieval. Journal of sensing and Imaging	2014	SCI	Experimental results demonstrate that the proposed indexing method is superior to the former Block Truncation Coding (BTC) image retrieval system and the other existing methods. The ODBTC method offers an effective way to index an image in a content-based image retrieval system, and simultaneously it is able to compress an image efficiently. Thus, this system can be a very competitive candidate in image retrieval applications.
Schroder, J., Goetze, S., & Anemuller, J. Spectro-Temporal Gabor Filterbank Features for Acoustic Event Detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing	2015	IEEE	Feature Extraction Technique: Gabor -Filter(Optimized GFB parameter for acoustic event detection.)
Mistry, Y., Ingole, D. T., & Ingole, M. D. Content based image retrieval using hybrid features and various distance metric. Journal of Electrical Systems and	2017	Springer	Feature Extraction Technique: Color (To enhance the accuracy of binary statistical image

Information Technology			features, color and edge directivity descriptor features are used to develop an efficient CBIR system.)
Ali, A., & Sharma, S.Content based Image retrieval using feature extraction with machine learning. (ICICCS).	2017	Springer	Feature Extraction Technique:Visual(To check The similarity, a deep neural network is trained and the validation and texting phases are carried out accordingly.)
Srivastava P, Khare A. Utilizing Multiscale Local binary pattern for content-based image retrieval. International Journal of Multimedia Tools & Applications	2017	Springer	Local Binary Pattern of different combinations of eight neighbourhood pixels is computed at multiple scales. The final feature vector is constructed through Gray Level Co-occurrence Matrix (GLCM). Advantage of the proposed multiscale LBP scheme is that it overcomes the Limitations of single scale LBP and acts as more robust feature Descriptor. It efficiently captures large scale dominant features of some textures which single scale LBP fails to do and also overcomes some of the limitations of other multiscale LBP Techniques.
Unar, S., Wang, X., Wang, C., & Wang, Y. A decisive content based image	2019	Google Scholar	Feature Extraction Technique: Visual and

Retrieval approach for feature fusion in visual and textual images. Knowledge-Based Systems.			Textual (It supports three methods of recovery: image query, keywords and both, in order to obtain identical images.)
Liu, Z., Lai, Z., Ou, W., Zhang, K., & Zheng, R. Structured optimal graph based sparse feature extraction for semi-supervised learning. Signal Processing	2020	Google Scholar	Feature Extraction Technique: Gabor-filter(To achieve small- scale representation with an improved sample-to-Feature ratio.)
Ashkan Shakarami & Hadis Tarrah, An efficient image descriptor for image classification and CBIR	2020	Scopus	Accuracy comparison with different feature extraction techniques along with different classifiers.
M.Buvana,K.Muthumayil,T.Jayasankar Content-Based Image Retrieval based on Hybrid Feature Extraction and Feature Selection Technique Pigeon Inspired Based Optimization	2021	SCOPUS	In this research study, Gray-Level Co-Occurrence Matrix (GLCM)along with Local Binary Pattern (LBP), and histogram of oriented gradients (HOG) features. Feature selection technique called Pigeon Inspired based Optimization (PIO) is used along with Artificial Neural Network (ANN) Classifier and compared existing techniques in terms of accuracy, precision, recall and F-measure.
S. Manoharan, L. Velmurugan, V. Chandran, P. Sasikumar,Content –Based Image Retrieval Using Colour,Advanced	2022	SCI/SCI SCOPUS	Colour, gray advanced texture, shape features and Random Forest Classifier

Texture, Shape Features and Random Forest Classifier with Optimized Particle			with optimized PSO provides efficient retrieval of images in a large-scale database. The matching accuracy and the speed of image retrieval are improved by an ensemble of machine learning algorithms for the similarity search.

Table 2.1: Comparison Chart for Literature Survey

2.3 RESEARCH GAPS

1. The first research gap is related to the volume of CBIR database. If we go with manual annotation is very expensive, some and complicated to be applied for large dataset of images.

2. The second research gap is linked to the performance of system that include description of images and labeling comes out to be very subjective because same image might have different level, like a picture on beach which comes under beach, Sky and water labels, problem with these things open door for new research.

3. The third research gap is about semantic gap, CBIR system is develop on feature vector of visual shown image but humans compare any two images based upon its semantic, which leads to unsatisfactory result and create research gap in reality and system developed. The results which come might perfect as per feature matching but not satisfactory by human based upon their visualizations.

4. The fourth research gap is about the CBIR system accuracy, As we keep on adding more algorithm like feature selection and evolution theory which optimized the features to speed up the techniques is where we lost our important information which stored in image features to give desired results. Less the features more chances to get deviated from original label of image.

2.4 DATASETS

The MATLAB programming language is used to create the suggested algorithm. The computer has an Intel i7 processor with 4GB of RAM and runs Windows 11. Every test is run to assess the system's performance as a whole. We have used Heterogeneous Datasets. We have included a summary of datasets and experimental findings that were utilizing the suggested CBIR framework in this section. We have used CIFAR-10, CIFAR-100, Web–Crawled misc1, and Web-crawled misc2 heterogeneous datasets as described below:

2.4.1 CIFAR 10 Dataset

Sixty thousand 32x32 pictures totaling ten classes with a total of 6000 images each make up the CIFAR10 dataset. There are 50,000 training images and 10,000 test images. The 10,000 images in the dataset are split into five training batches and one test batch, respectively. Each randomly chosen class has exactly 1000 photographs in the test batch. The remaining photos are placed randomly on the training deck but some decks may contain more classes of photos than others.

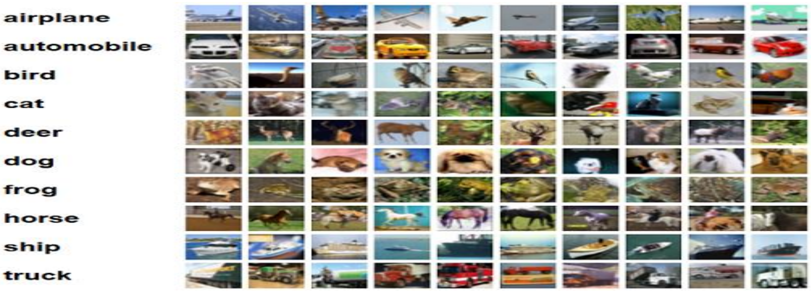


Figure 2.4: sample Images of CIFAR 10 Dataset

2.4.2 CIFAR 100 Dataset

There are 100 classes in the CIFAR-100 dataset, and there are 600 photos in each class. Each of the 20 super classes, which make up the classes, has 100 test photographs and 500 training photos. These pictures are labeled as "fine" and ‘‘coarse’’. The term fine indicates the class it belongs to, while as label coarse gives information about its super class.

Figure 2.5: Sample images of CIFAR-100 Dataset

2.4.3 Web Crawled misc1 Dataset

Ten thousand test photographs in the low-resolution web-crawled misc database1 are used in WBIIS for study comparison.

Figure 2.6: Sample Images of web-crawled misc database1

2.4.4 Web Crawled misc2 Dataset

There are 1000 test photographs in the low-resolution web-crawled misc database2 (SIMPLIcity's test database).

Figure 2.7: Sample Images of web-crawled misc database2

CHAPTER 3

FEATURE EXTRACTION AND CLUSTERING TECHNIQUES

The motive of this chapter is to extract the features of images using different clustering Techniques. We have used different feature extraction techniques along with clustering Techniques to get the extracted features.

3.1 FEATURE EXTRACTION TECHNIQUES AND CLUSTERING TECHNIQUES

Feature Extraction techniques are used to extract the features of a given query image and then clusters are made on the basis of similarity. Different Feature extraction and clustering techniques are given below:

3.1.1Fuzzy C Mean

Numerous geo-statistical data [44] analysis issues may be solved using the FCM application. For each collection of numerical data, this application creates fuzzy partitions and prototypes. These partitions can be used to confirm existing substructures or to infer substructure in undiscovered data. An objective function called the generalized least-squares is utilized as the clustering criteria to aggregate subsets. This program's features include accepting varying numbers of clusters, a choice of three norms (Euclidean, Diagonal, or Mahalonobis), an adjustable weighting factor that effectively adjusts susceptibility to noise, and outputs that contain many metrics of cluster validity. Clustering is carried out automatically depending on the area of interest.

Instead, the soft clustering method is used to identify the "soft partition" of a dataset. In a "soft partition," a datum might make up a large number of clusters partly. Since the input space may be bigger than the dataset, a soft partition is not always a fuzzy partition. However, the division created by the bulk of soft clustering techniques is both fuzzy and soft. Point x is ensured to have a membership degree of 1 in each cluster in one particular application of soft clustering, i.e.

$$\sum_j \mu_{c_f}(x_i) = 1 \forall_{x_i} \in X \quad \text{... (3.1)}$$

When soft partitions meet this extra need, they are referred to as being limited. Constrained soft partition is created using fuzzy C-Means, the most used fuzzy clustering method. To enable limited soft partition, the goal function J1 of hard cmeans has been expanded in two different ways:

(1) The fuzzy membership degree in the cluster was taken into consideration throughout the computation.

(2) As a weight exponent, a new parameter m has been added to fuzzy membership. The expanded objective function is as follows, as Jm indicated:

$$J_m(P,V) = \sum_{i-1}^{k} \sum_{x_k \in X}$$

… (3.2)

Using C1, C2,..., Ck to divide the dataset X in a fuzzy manner, where k is the number of clusters. The weight parameter specifies how much the partial members of a cluster influence the clustering result. In order to discover a decent partition, fuzzy c-means seeks a prototype vi that minimizes the objective function Jm.

The fuzzy c-means algorithm must additionally seek for a membership function ci that minimizes Jm in contrast to hard c-means. A restricted fuzzy partition C1, C2,... Ck can only be the local minimum of the objective function Jm if the following conditions are satisfied:

$$\mu c_1(x) = \frac{1}{\sum_{j=1}^{k} \left(\frac{\|x - v_i\|^2}{\|x - v_j\|^2} \right)^{\frac{1}{m-1}}} \quad \text{… (3.3)}$$

$$1 \leq i \leq k, x \in X$$

Keep in mind the following crucial information about the FCM algorithm:

For m > 1, it ensures convergence.

It locates the goal function Jm's local minimum.

The initial prototype as well as the variables m and c all affect how FCM performs on a given dataset.

Pseudocode of Fuzzy C Mean

Given steps tells about how Fuzzy CMean actually works.

Step 1: Each item's frequency in the Data is shown by Variable H.

Create vector I=min(Data):max in step two (Data)

Step 3: Select a centroid at random, preferably 2.

Step 4: Create a membership matrix

$U_{ij=1/}\sum^{c}_{k=1}[I_i-C_j]/ [I_i-C_k]^{2/m-1}$

Step5: calculate the cluster center:

$C=\Sigma^{n}_{i}=_{1}\ U^{m}*H*I/\ \Sigma^{n}_{i}=_{1}\ U^{m}*H$

Step6: Stop if $C^{(k-1)}- C^{k}<\varepsilon$, otherwise move on to Step4.

3.1.2 Gray Level Co-occurrence

Image analysis methods [33] include the Gray Level Co-occurrence Matrix (GLCM) and related texture feature estimates. The GLCM is a table that displays the frequency with which specific combinations of grey levels co-occur in an image or image fragment made up of pixels with varying intensities (a distinct grey level). When calculating texture features, the GLCM data is employed to produce a measurement of the intensity variation, or image texture, at the pixel of interest.

Pseudocode of GLCM

1. ASM

ASM serves as a representation of an image's homogeneity.

$$ASM = \sum_{i=1}^{L}\sum_{j=1}^{L}\left(GLCM(i,j)\right)^2 \qquad \ldots (3.4)$$

2. IDM

It is measured how homogeneous an image with the same degree of grey is.

$$IDM = \sum_{i=1}^{L}\sum_{j=1}^{L}\frac{\left(GLCM(i,j)\right)^2}{1+(i-j)^2} \qquad \ldots (3.5)$$

3. Correlation

This is an expression of the magnitude of the linear dependency between the grey levels on the picture.

$$IDM = \sum_{i=1}^{L}\sum_{j=1}^{L}(i-\mu i^{-1})(i-\mu j^{-1}) \qquad \ldots (3.6)$$

4. Entropy

A grey level irregularity's size

$$Entropy = \sum_{i=1}^{L}\sum_{j=1}^{L}\left(GLCM(i,j)\right)log\left(GLCM(i,j)\right) \qquad \ldots (3.7)$$

5. Contrast

Declares the size of picture pixel-gray level variations' presence.

$$Contrast = \sum_{i=1}^{L}\sum_{j=1}^{L}|i,j|^2 GLCM(I,J) \qquad \ldots (3.8)$$

3.1.3 K mean

The dataset is divided into [56] K pre-specified, unique, non-overlapping subgroups (clusters) via an iterative method, and each data point is assigned to a single cluster. While keeping the clusters as distinct (and apart) as is practical, it aims to make the intra-cluster data points as similar as is practical. The process for allocating data points to clusters ensures that the sum of the squared distances between each data point and the cluster centroid, or the average value of all the data points in the cluster, cannot be more than one. The homogeneity (similarity) of the data points inside a cluster increases as the variety inside the cluster decreases.

The following is how the kmeans algorithm works:

•The required number of clusters is K.

• Initialize the centroids after generating a random distribution for the dataset and choosing K data points for the centroids without changing any of them.

• Till the centroids are consistent, keep iterating. In other words, the data points' grouping has not changed.

• It is necessary to sum up the squared distances between each data point and each centroid.

• Place each data point in the closest cluster (centroid).

Get the cluster centroids by averaging the related data points for each cluster.

To overcome the problem, Kmeans uses the expectation-maximization strategy. The nearest cluster is given the data points in the E-step. The centroid of each cluster is calculated using the M-step. The mathematical steps we can use to solve it are discussed below.

The intended behaviour is:

W_{ik}=1 if data point xi is a member of cluster k; else,

$$j = \sum_{i=1}^{m} \sum_{k=1}^{k} w_{ik} \left\| x^i - \mu_k \right\|^2 \ldots.. \qquad \ldots (3.9)$$

w_{ik}=0. Moreover, k is the centroid of the cluster.

This minimization problem has two parts. We begin by assuming k as fixed and reducing J in regard to w_{ik}. Then, we minimize J w.r.t. k and consider wik to be fixed. Technically speaking, we first differentiate J based on w_{ik}, and then we update cluster allocations (E-step). Following the previous phase's cluster assignments, we differentiate J w.r.t. k and recompute the centroids (M-step). Therefore, E-step is:

$$\frac{\partial j}{\partial w_{ik}} = \sum_{i=1}^{m}\sum_{k=1}^{k}\left\|x^i - \mu_k\right\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 if k = argmin_j \left\|x^i - \mu_k\right\|^2 \\ 0 otherwise \end{cases} \quad \ldots (3.10)$$

In other words, the cluster with the smallest sum-squared distance from its centroid should receive the data point xi.

To recompute the centroid of every cluster to reflect it to the new assignments, M-Step is used.

$$\frac{\partial j}{\partial_{\mu_k}} = 2\sum_{i=1}^{m} w_{ik}(x^1 - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}} \ldots.. \quad \ldots (3.11)$$

In our work, we have taken 2 clusters. Because our dataset consist of two things, one is background and another is things in it like animals, bus, bird etc.

Pseudocode of K mean

Below [31] is a description of the K-means algorithm.

K-Mean()

Set the objects in the Row and the characteristics in the Column to initialise the matrix M.

Set the initial centroids of clusters C1, C2,..., Ck in matrix C.

Set the Group matrix G to zero (0)

As long as (G(i)!= G(i+1))

Find the updated centroid coordinate

Find the separation between each object and the centroids.

Rearrange the Group Matrix

3.1.4 Principal Component Analysis

Principal component analysis [32], or PCA, is a well-liked method for decreasing the dimensionality of huge data sets. This is how PCA works: it takes a huge collection of variables and condenses them into a smaller set while keeping most of the data from the larger set. As the number of variables in a data collection rises, accuracy always declines; nevertheless, dimensionality reduction can be accomplished by trading some accuracy in favour of simplicity. Smaller data sets allow machine learning algorithms to assess data far more rapidly and effectively since there are less irrelevant aspects to account for.

Pseudocode of PCA

1. Ignore the class labels and take the entire dataset of dd-dimensional samples.

2. We should calculate the dd-dimensional mean vector, or the means for all of the dataset's dimensions.

3. For the whole set of data, calculate the scatter matrix (or covariance matrix).

4. Determine the eigenvalues (ee1, ee2,..., d) and associated eigenvectors (ee1, ee2,..., d).

5. Choose the kk eigen vectors with the greatest eigen values and arrange the eigen vectors in decreasing order to construct a dkdk-dimensional matrix (where every column represents an eigen vector)

6. To map the samples into the new subspace, use this dkdk eigenvector matrix. This can be expressed mathematically as yy=WWTxxyy=WWTxx, where yyyy is the converted k1k1-dimensional sample in the new subspace and xxxx is a d1d1-dimensional vector representing one sample.

3.1.5 Local Binary Pattern

The region around each [71] pixel is thresholded to form the pixel labels in an image, and the texture operator interprets the result as a binary integer. A popular approach in many applications, the LBP texture operator has a significant capacity for discrimination and is computationally straightforward. A unified approach for these models might be seen as a solution to the tension between the statistical and structural models of texture analysis. In practical applications, the LBP operator's resilience to monotonic gray-scale oscillations caused, for instance, by changes in illumination, may be its most important quality. Another significant feature is its computational simplicity, which permits photo analysis under challenging real-time circumstances.

To create the LBP, powers of two weights are multiplied. In Figure 3.1, the computation of LBP is shown. The LBP has a spectrum of grey values from 0 to 255.

$$LBP_{P,R}(x,y) = \sum_{i=0}^{P-1} s(g_i - g_c)2^i; s(x) = \begin{cases} 1, if x \geq 0 \\ 0, otherwise \end{cases}$$

… (3.12)

where the letters gc and gi stand for the centre and surrounding pixels' equivalent shades of grey. We use the notation (P, R), where P stands for the number of pixels in the immediate area and R stands for the circle's radius.

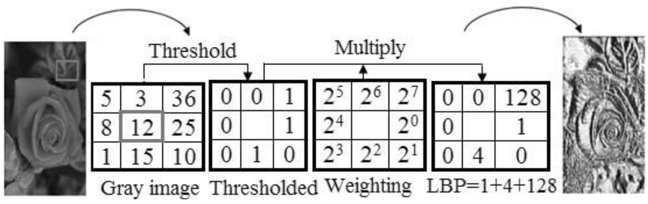


Figure 3.1: Calculation of the LBP code

Figure 3.2 displays many samples with various elemental concentrations (P, R). The initial LBP has been extended with three new techniques: uniform (u2), invariant rotation (ri), and uniform invariant rotation (uir) (riu2).

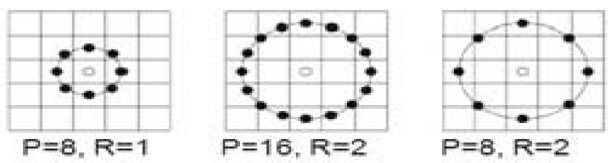


Figure 3.2: Circularly symmetric neighbour sets for various types of neighbours (P, R)

The LBP is regarded as uniform if the circular binary string only experiences two bitwise conversions from one to zero or zero to one. For instance, whereas 00000000 and 1000001 are uniform patterns, 11100101 and 10100001 are not. The original LBP had 256 bins, whereas the uniform LBP, also known as LBPu2, has 59 bins.

Pseudocode of LBP

1. Change the picture to grayscale.

2 For each pixel (gp) in the image, choose the P neighbourhoods that are in the vicinity of the core pixel. gp's coordinates are calculated using the formula (gc x-Rsin(2p/P),gc y+Rcos(2p/P)).

3. Make the P neighbours' threshold the gc pixel in the centre.

4. Set to 1 if the neighbouring pixel's value is higher than or equal to the centre pixel's value; otherwise, set to 0.

5. Determine the LBP value right now: A binary number with digits close to the centre pixel should be written counterclockwise. This binary value has the term LBP-central pixel code (or its decimal equivalent).

3.1.6 Scale Invariant Feature Transform

SIFT is a method for finding forms such as circles, blobs, and corners. A picture can also be resized using it. Regardless of the size or orientation of the image, it can recognise some details. SIFT is extremely stable and unaffected by alterations in size and rotation. The SIFT keypoints provide a ton of information.

Scale space extremes detection, key point localization, orientation assignment, and description creation are the four main steps of the SIFT approach.

A.SIFT descriptor

The SIFT [62] method was developed by Lowe to extract the most stable IPs from an image and provide a vector descriptor using the Difference of Gaussian (DoG). The algorithm is explained in the phrases that follow.

1) Scale-Space Extrema Detection: IPs are recovered with scale and orientation invariance using step pyramids constructed at various scales of the Gaussian function.

2) Finding Points of Interest: Several IP candidates are found during the preliminary stage.

To improve the endurance of the IPs, this stage involves deleting those with poor localisation along edges or low contrast.

3) The orientation assignment

Each recognized IP is given one or more orientations in the previous stage. A gradient histogram with 36 intervals is produced as a result of this IP orientation.

4) A synopsis of the important subject

According to Lowe, a 16-by-16-pixel frame should have many histograms surrounding an IP. In most IP standards, 16 orientation histograms are employed, grouped in a 4x4 grid. According to Figure 3.3, each histogram has eight orientations, producing a feature vector with 128 elements.

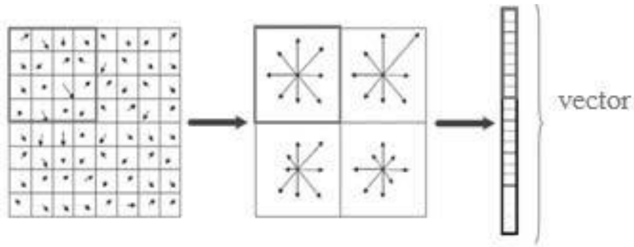


Figure 3.3: SIFT descriptor extraction

Pseudocode of SIFT

Step 1: Locate [69] the area of peaks in the Scale-area. Determine the approximate size of the salient characteristic points (also known as keypoints) using a special smoothing version of the same image known as the Scale Space by varying the Gaussian sigma's dimensions.

Step 2: Localization of Keypoints - Find the precise location of a Keypoint. The maxima or minima in the stack of DoG pictures, or the "scale-space-pyramid," are the important spots. By doing this, you learn the keypoint's scale and placement.

Step 3: Determine the orientation of each keypoint in step three's orientation assignment. Keep each feature's theta, scale, and placement intact.

Step 4: Keypoint Descriptor. A tiny area surrounding the keypoint is split into n × n cells (often n = 2). A gradient orientation histogram is created in each cell, each of which is 4 x 4. The gradient magnitude and a Gaussian weighting function with = 0.5 times window width are used to weight each histogram entry. The dominating orientation of the keypoint is taken into consideration while sorting each gradient orientation histogram.

3.2 PROPOSED METHODOLOGY

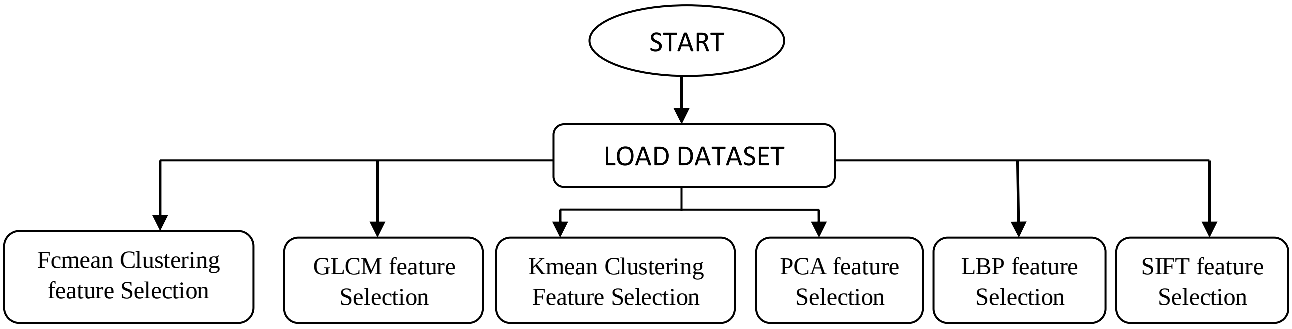


Figure 3.4: Various Feature Extraction and Clustering Techniques

We conduct a systematic search for articles in the Scopus and Web of Science databases for the years 2010–2022. We review a variety of feature selection and feature extraction approaches that have been successfully applied in the Content Based Image Retrieval. We find that FC mean, GLCM, K mean, PCA, LBP and SIFT are the most widely used feature selection and extraction techniques with the best prediction accuracy for various Content Based Image Retrieval applications.

First we will load the dataset. Then we will do the normalization of the dataset.

After normalization, our work flow is divided into six parts. One is using FCM clustering, GLCM, Kmean clustering, PCA, LBP and SIFT.

Clustering techniques are unsupervised learning methods of grouping similar from dissimilar data types. Therefore, these are popular for various data mining and pattern recognition purposes. However, their performances are data dependent. In this work, we have tested the performances of a soft clustering (e.g., Fuzzy C means or FCM) and a Hard clustering technique (e.g., K-means or KM). Distance measure is the heart of any clustering algorithm to compute the similarity between any two data. FCM produces most compact clusters, while KM yields most distinct clusters.

When implementation performed, we got 204 features in FC Mean ,13 in GLCM, 199 in Kmean ,51 in LBP, 99 in PCA and 48 in SIFT respectively.

CHAPTER 4

PERFORMANCE OF MACHINE LEARNING CLASSIFIERS

The motive of this chapter is to show the performance of different machine learning classifiers. We have used different feature extraction techniques along with clustering Techniques in previous chapter. Along with those techniques, we have shown the performance of different classifiers in this chapter. For example, Support Vector Machine, K Nearest Neighbor, Decision Tree, Gradient Boosting Algorithm and Alexnet have been used. Apart from this, indexing technique is applied on the extracted features using Alexnet and most matched images will be retrieved from the dataset.

4.1 PERFORMANCE TEST OF INDIVIDUAL CLASSIFIERS

In this section, performance of the individual classifiers (SVM, KNN, DT and GT) have been analyzed. These classifiers have been used in conjunction with the proposed approach in Chapter 3.

4.1.1 Support Vector Machine

SVM [22], a supervised learning technique, may be utilised to address classification and regression problems. Due to its greater recognition rate when compared to other classification systems, it has attracted several academics from all over the world. The main application is utilising a hyperplane in a high-dimensional feature space to classify two classes. The basic goal is to find a hyperplane in an n-dimensional space that can group the vertices into related groups. The term "hyperplane" means:

$$w.x + b = 0 \qquad \dots (4.1)$$

where the weight vector (w), which is parallel to the hyperplane, is defined. Bias or threshold is the definition of the parameter b. SVM employs the "one-versus-rest" method for multi-class classification. There are created n classifiers, each of which trains data from two classes, totaling n classes * (n classes - 1). The decision function shape option enables you to convert the output of "one-versus-one" classifiers into a "one-vs-rest" decision function of shape in order to give a consistent interface with other classifiers. This study's SVM makes use of a rbf kernel with a regularization parameter set to 1. The form of the decision function is set to ovr.

4.1.2 K- Nearest Neighbor

Another supervised learning strategy for classification problems is KNN [34]. It takes no effort to acquire any knowledge. Its most common use is to classify a data point based on the classification of its neighbours. In this investigation, two neighbours were utilised. The weight function that is used for prediction is set to "uniform" by default. It means that each neighborhood's points are equally weighted. The default setting for the closest neighbour computation algorithm is "auto." Based on the variables supplied to the fit technique, "auto" will try to determine which of the three methods BallTree, KDTree, and brute force search is the best match. The distance unit used is the euclidean distance.

4.1.3 Decision Tree

The decision tree is the most efficient and popular classification and prediction method. Each internal node in a decision tree represents an attribute test, each branch represents the outcome of the test, and each leaf node (terminal node) represents the class label. Flowcharts are similar to decision trees, a sort of tree structure.

A tree can be "taught" by breaking the source setup into subgroups depending on an attribute value test. The process of repeating this procedure on each derived subset is known as recursive partitioning. The recursion ends when the split no longer improves the predictions or when each member of a subset at a node has the same value for the target variable. The building of decision trees classifiers is ideal for exploratory knowledge discovery since it does not need parameter configuration or domain knowledge. Decision trees can be used to manage the high-dimensional data. Decision tree-based classifiers are typically accurate .Decision tree induction is an inductive method for learning classification.

4.1.4 Gradient Boosting Algorithm

Boosting methods are machine learning methods that help to overcome the challenges of processing extensive data. This approach sequentially combines various predictors with some degree of contraction. Each predictor is applied to the committee's residuals generated by the previous predictor and forms the final prediction based on a subset of variables used to classify samples into different categories. It starts with making a leaf instead of a tree or stump. This sheet represents the initial estimates of the weights for all samples. When predicting continuous values like weights, the first guess is the mean. Then it builds a tree. Then again built another tree based on the previous errors.

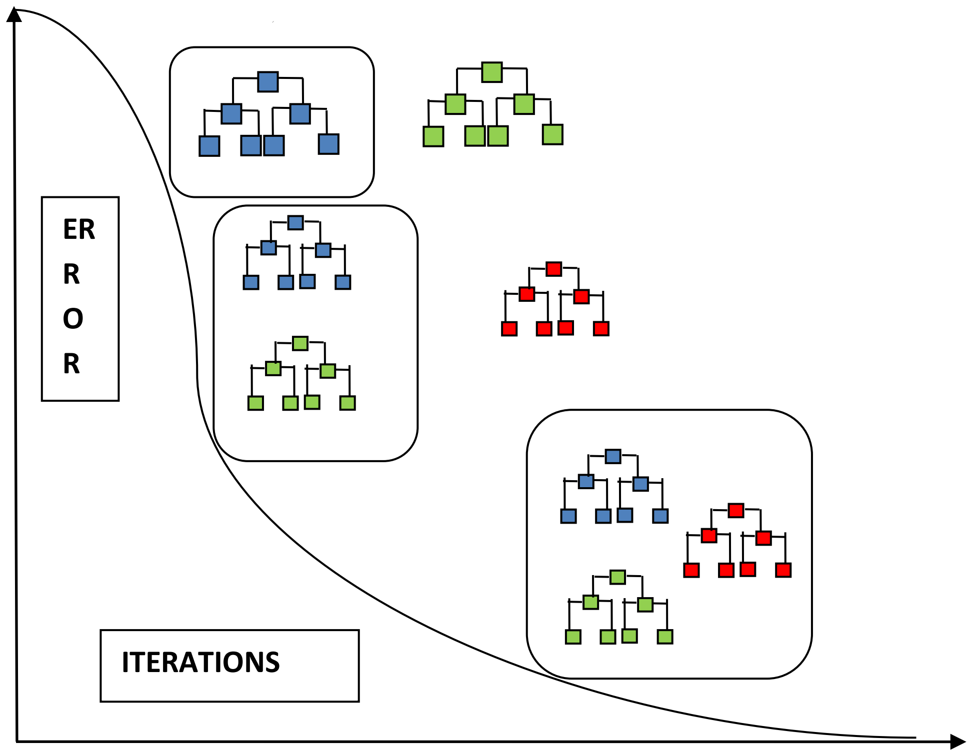


Figure 4.1: Representation of Gradient Boosting Algorithm

4.2 CBIR IMPLEMENTATION USING MACHINE LEARNING AND DEEP LEARNING

All data is limited in deep learning approaches, and the contents are taught by segmenting their features into a deep bottom. The database maintains a unique, individual data centre with just the most important elements. As the data is presently being analyzed, the deep learning technique performs at its peak and plays a smart content extraction role. The soft computing phenomenon known as "deep learning" includes a subcategory that enables the recovery of information from millions of dispersed images. The ability of a content-based image retrieval system to find pictures is significantly influenced by the feature representation and similarity measurement.

In order to attempt to model high-level abstractions in data, deep learning is a family of machine learning algorithms that makes use of deep architectures made up of several non-linear transformations. Deep learning is structured in a deep architecture that analyses information via a number of phases of transformation and representation, in contrast to traditional machine learning techniques, which frequently use "shallow" structures.

Instead of relying on manually created features made with domain knowledge, deep learning techniques allow a system to automatically examine deep architectural characteristics at various levels of abstraction from data and learn complicated functions that directly transfer raw sensory input datum to the output. sophisticated neural network To push the limits of machine learning, several scientists have developed an advanced artificial neural network termed deep learning. Information extraction utilizing high level abstraction is the main goal of this deep learning technology. This method commonly employs multistage nonlinear transformers, which resemble numerous cascading neural networks. As part of high level data abstractions carried out with distributed representation, data will be appraised with a variety of dimensions and attributes. A hierarchy of explanatory elements is employed to implement each abstraction, with a single prior level of produced knowledge being used to construct a number of sublevels of information. All deep learning networks consist of multistage nonlinear transformations, therefore to speed up learning; restricted Boltzmann machines (RBMs) are stacked into multistage layers with just one hidden layer in each layer.

.

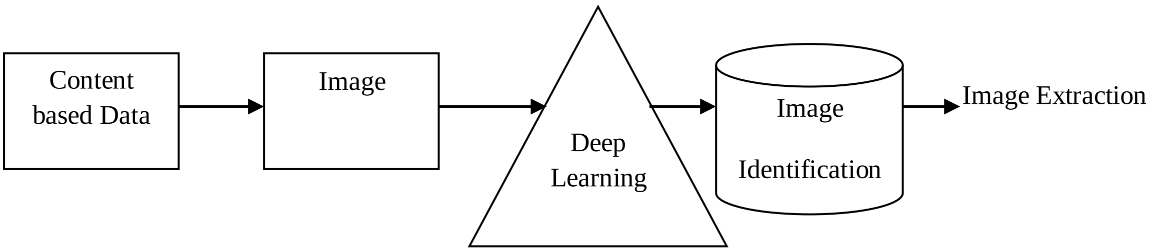


Figure 4.2: CBIR system with Deep Learning Method

In contrast to typical machine learning approaches, which commonly employ "shallow" structures, deep learning is built in a deep architecture that analyses input via a number of steps of transformation and representation. Deep learning techniques enable a system to learn complex functions that directly transfer raw sensory input datum to the output without relying on human-crafted features created using domain knowledge by automatically exploring deep architectural characteristics at different levels of abstraction from data. Deep learning approaches have recently been shown to be useful in a number of applications, including speech recognition, object identification, and natural language processing, among others. Feature representations from images may be extracted using deep learning algorithms, and representations can be compared to CBIR tasks.

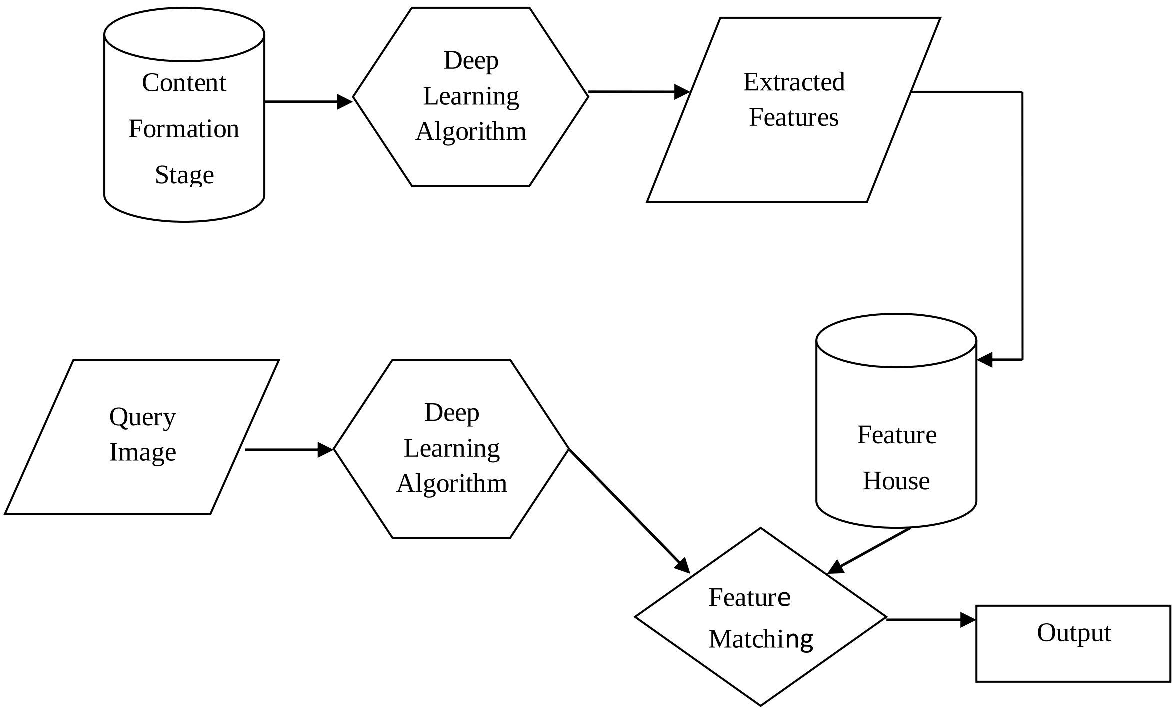


Figure 4.3: CBIR Working with Deep Learning Technique

4.3 DEEP LEARNING BASICS

Deep learning uses machine learning (ML) methods that are modelled after the structure and operation of neural networks or the human brain. Due to its intended purposes and structural makeup, it was referred to as an Artificial Neural Network (ANN).DL is a method of machine learning that focuses on instructing systems via examples [4]. Humans and this are comparable. We learn new things through examples or experiences. The instructor, the master, or the learned employs examples while instructing a student, apprentice, or novice; the more examples provided, the more the pupil learns. A DL algorithm therefore performs better as data size grows. Deep learning is a neural network-based technique; therefore it doesn't need specialists to manually extract traits because it learns features and tasks directly from the training data [5]. Since deep learning techniques make use of neural network topologies, deep learning models are occasionally referred to as deep neural networks (DNN) [6]. The number of ANN's hidden layers is indicated by the term "deep." For the analysis of image data, Convolutional Neural Networks (CNN), a common type of DNN, is the best option. Why is DL a part of this inquiry? Since DL models are more accurate than humans in categorizing images, this ML strategy has lately gained popularity and is currently among the most widely used models in AI. As a result of the high-performance processing made feasible by GPU, we can train deep networks more quickly. Another aspect is the recent increase in accessibility of the label data that deep learning requires. When there are enough annotated photos available, deep learning approaches have been demonstrated to be more effective than conventional ML techniques [7]. Photo classification is getting more and more important as deep learning (DL) develops quickly. With a promising future ahead of it, data analysis utilizing DL will surely have an international influence in the years to come.

4.3.1 Convolutional Neural Networks (CNNs or ConvNets)

Humans may be able to determine an animal's species based on its high-level morphological characteristics, such as the shape of its head, ears, eyes, mouth, legs, and other body components. On the other hand, a computer has a different perspective on the world. It looks for minuscule physical properties like corners and curves in order to do classification before employing a series of convolutional procedures to produce abstract concepts. A computer builds high-level physical attributes from the returned low-level physical information in order to identify the object. This is done by using CNN. An image processing and classification-focused network design for DL is called CNN [4]. It is one of the most popular and widely used DL networks [13]. CNN is to thank for DL's present enormous popularity. CNN's main advantage over its forerunners is that it automatically recognizes key elements without human input, which makes it the most popular [5].

Each layer of CNN transforms the data it gets from the layer below into more complex data that is then sent to following levels for further generalisation. By removing the pixel values from the input photographs and applying certain mathematical operations to the data over a number of layers, the network creates a classification for the images [14]. With activation functions serving as its hidden layer, this specific neural network (NN) features fully connected, convolutional, and pooling layers. Figure 4.4 [24] illustrates the architecture of CNN, which combines the feature extraction and classification blocks. When separating and identifying the various elements of an image, CNN uses a convolution tool and a pooling method. The output from the feature extraction block, or the extracted features, is used by the fully connected layer at the classification block to anticipate the class of the picture [14].

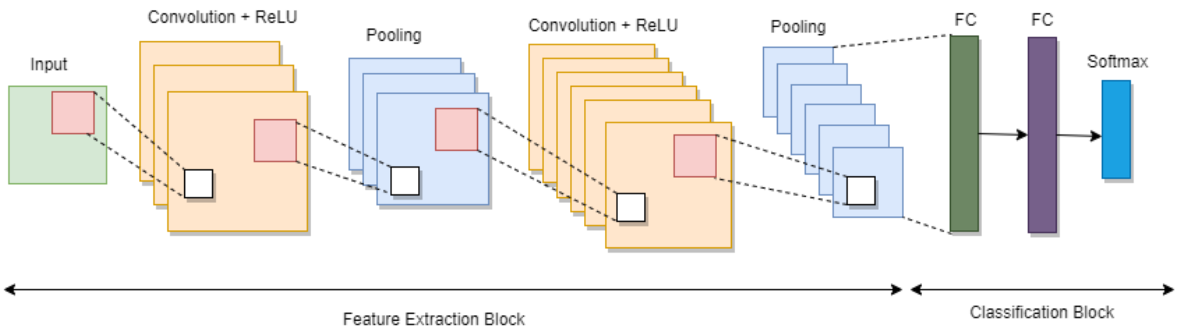


Figure 4.4: The basic CNN architecture [24]

4.3.1.1 Convolutional Layer

The convolutional layer performs convolution and activation processes [8].

Convolution

It is a linear mathematical process to apply a filter matrix over an image matrix. As the filter moves across the image, the local receptive fields, or filter values, are multiplied by the sections of the input image that are proportionate to the filter size [25]. By summing the multiplications from each of the local receptive fields to produce a single value, the output known as the feature map is produced. Figure 4.5 shows this convolutional process in action.

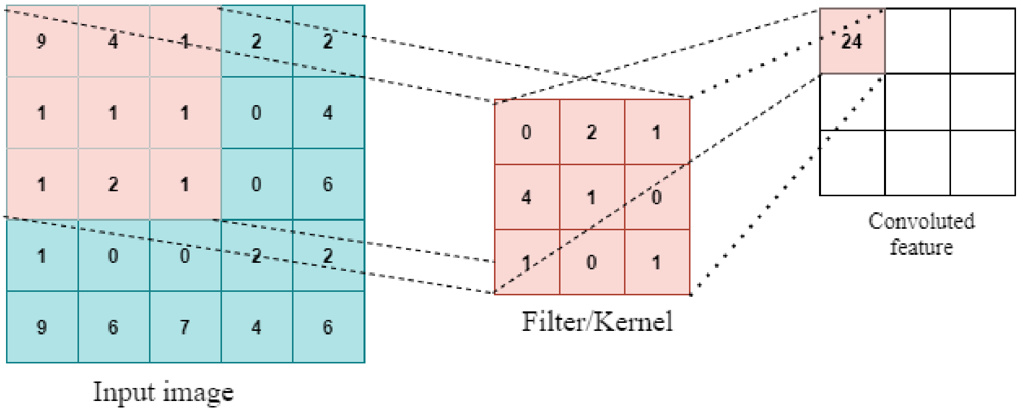


Figure 4.5: Convolution of an input image with a 3×3 kernel [25]

Activation Function

The feature map's elements are changed by the activation function to either the greatest positive value or, in the event of a negative element, to zero. Figure 4.6 provides a diagrammatic explanation of it. The network gains nonlinearity as a result. A popular non-linear activation function is the Rectifier Linear Unit (ReLU) [25]. ReLU has a lower computing burden than other activation functions, which is a benefit [5].

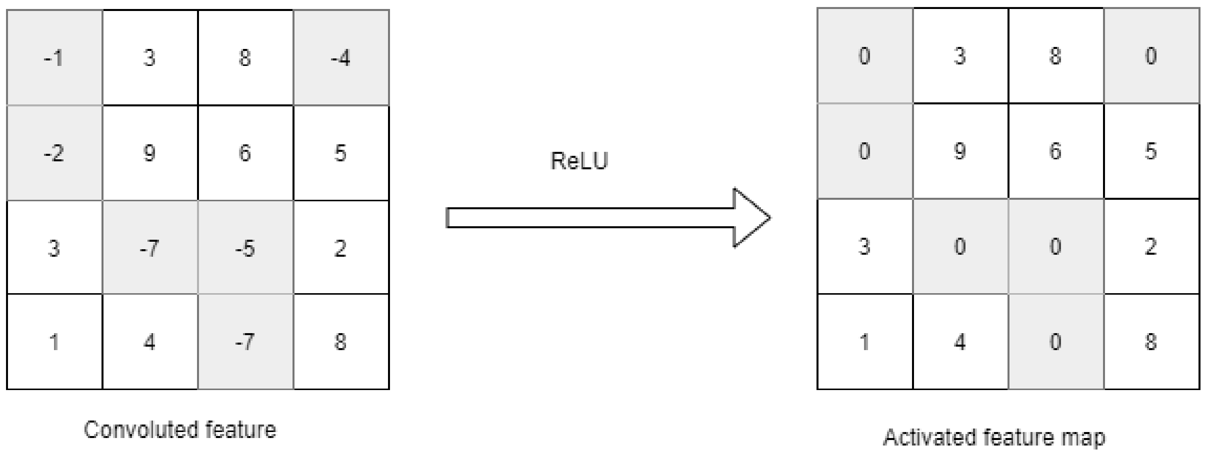


Figure 4.6: Activation operation on the convoluted feature [18]

4.3.1.2 Pooling Layer

By reducing the complex feature map's dimension to a single output, the pooling layer's main job is to implement down sampling [5] [6]. Figure 4.7 depicts pooling at its peak and average. The benefit of this method is that it lowers processing costs while maintaining the characteristics that give an image its uniqueness. Lowering over-fitting is an additional advantage of pooling.

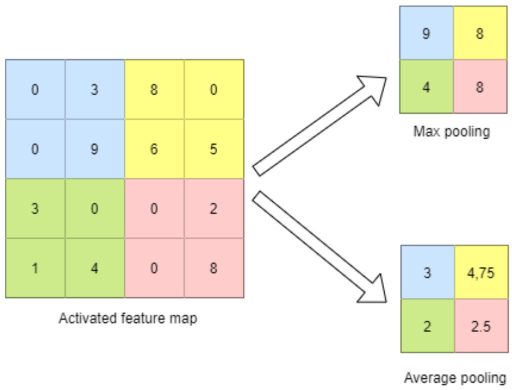


Figure 4.7: Max and average pooling on an activated feature map [24]

4.3.1.3 Fully Connected Layers

After feature extraction, we may divide the images into several groups using fully connected (FC) layers. FC layers are the last phase in CNN's classification process. The feature mappings from the pooling layer of the convolution block are transformed into a one-dimensional array of integers and linked to one or more fully connected layers, each of which has a learnable weight linking each input to each output [25]. Figure 4.8 depicts flattening in graphic form. Without the FC layers, a traditional CNN won't be able to forecast classes. All convolutional layer operations are finished, and the flattened feature map is then sent to an ANN for classification.

Figure 4.9 depicts the FC layers. The entire flattened feature map is provided by the ANN input layer to the FC layers, which use mathematical operations on each FC layer to combine the data into additional features that improve class prediction. We achieve the expected outcomes in the output layer [26].

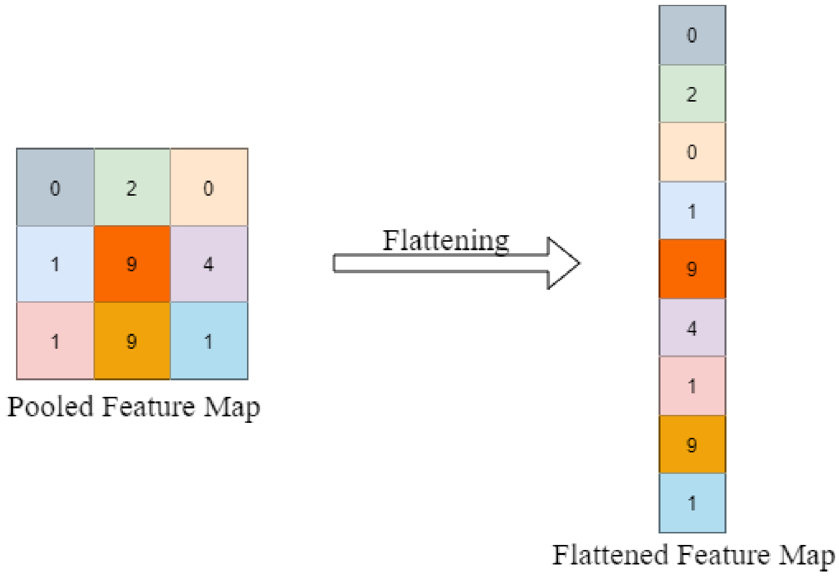


Figure 4.8: Flattening of a 3×3 image matrix to a 9×1 vector [24]

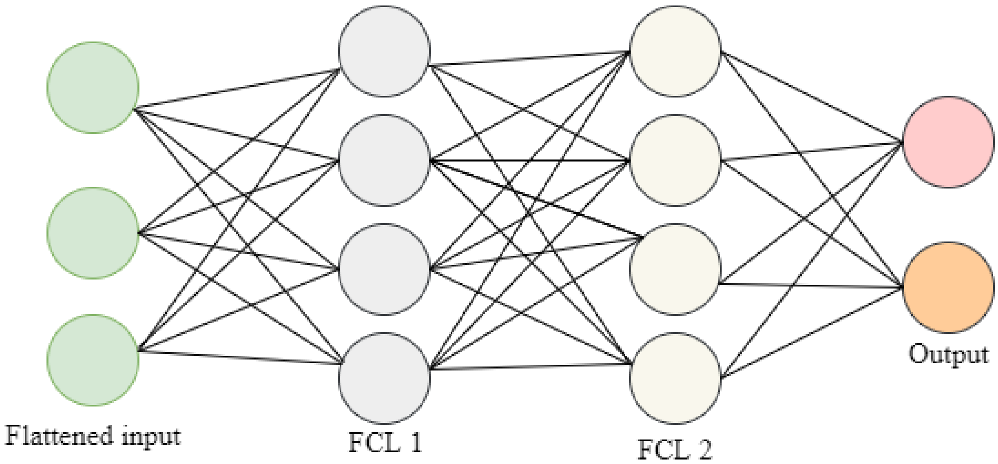


Figure 4.9: The fully connected layers with flattened input [35]

4.3.2 AlexNet and Its Architecture

The AlexNet model is a DL model that is built on CNN architecture. Together with Ilya Sutskever and Geoffrey Hinton, Alex Krizhevsky created the AlexNet model [5][9][35]. CNN was exclusively used for handwritten digit recognition tasks prior to winning the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with an accuracy of 84 percent [9][35]. To improve CNN's capacity for learning, AlexNet expanded the network and applied a number of parameter tuning techniques[5]. Deep CNN architecture holds AlexNet in high regard for its innovative work in picture recognition and classification [5].

Five convolutional layers, three max-pooling layers, two normalisation levels, two fully connected layers, and one softmax layer make up AlexNet's architecture. A diagrammatic representation of the AlexNet architecture is shown in Figure 3.7. A 227x227 pixel input picture is required by the AlexNet model. Filters, a nonlinear activation function, and ReLU make up each convolutional layer [25]. Convolution, pooling, and normalisation are the three algorithms that make up the first two convolutional layers [9]. The fourth and fifth convolutional layers execute convolution alone; no pooling or normalising is done after the second and third convolutional layers. As seen in Figure 4.10, the last convolution layer performs convolution and pooling without normalising.

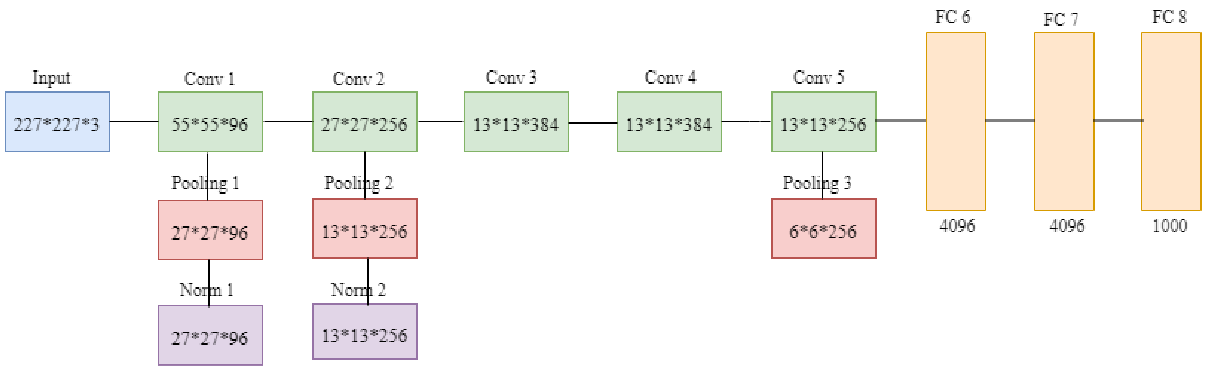


Figure 4.10: AlexNet architecture [9]

4.3.3 Transfer Learning

While the models are hampered by a lack of training data, Deep Learning approaches require a lot of data to function successfully. Since the lack of training data is a problem that the transfer learning approach successfully addresses, it is currently being employed to address this problem. [5] The relationship between the pre-trained model and the novel DL algorithm may be likened to that between a teacher and pupil in traditional language (TL). Before instructing a student, the teacher gains extensive understanding on the subject.

Transfer learning, a convenient and effective learning technique, allows deep learning algorithms to be trained rapidly and efficiently with a little amount of labelled data [10]. This learning method's fundamental tenet is to train a model for new, related tasks by starting with its previous tasks. Researchers in medical imaging use TL because it can be implemented quickly and doesn't require a huge annotated training dataset [7]. Pre-trained models may commonly improve a model's accuracy when it is trained on a smaller dataset and help it fit a certain domain [15]. The transfer learning used in this study was the pre-trained AlexNet mode.

Apart from this, we have extracted the features using Alexnet and after extracting the features, we have designed an indexing Technique, whose flowchart is shown in figure 4.12.

4.4 PROPOSED METHODOLOGY

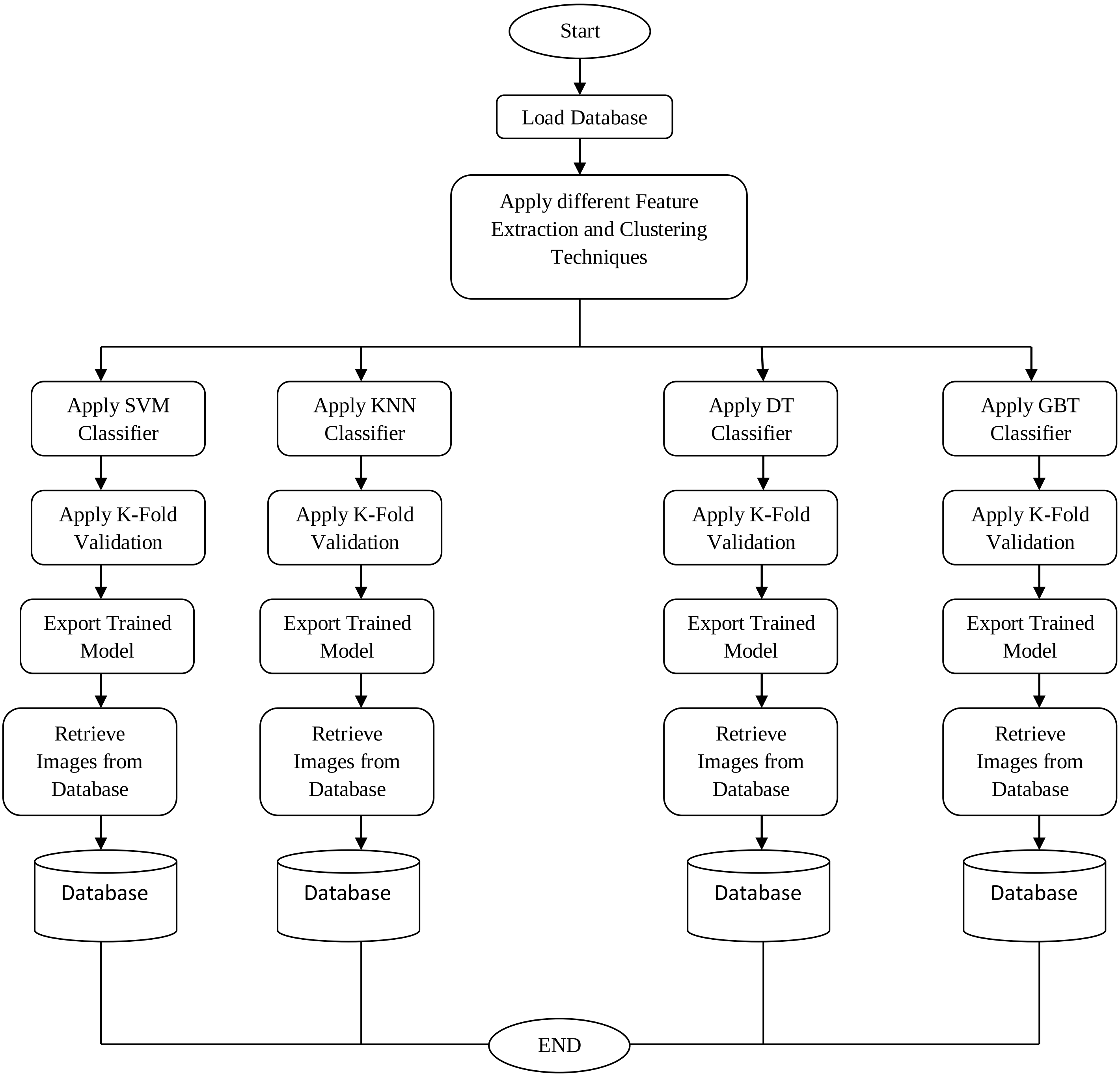


Figure 4.11: Different Classifiers used to classify the Features

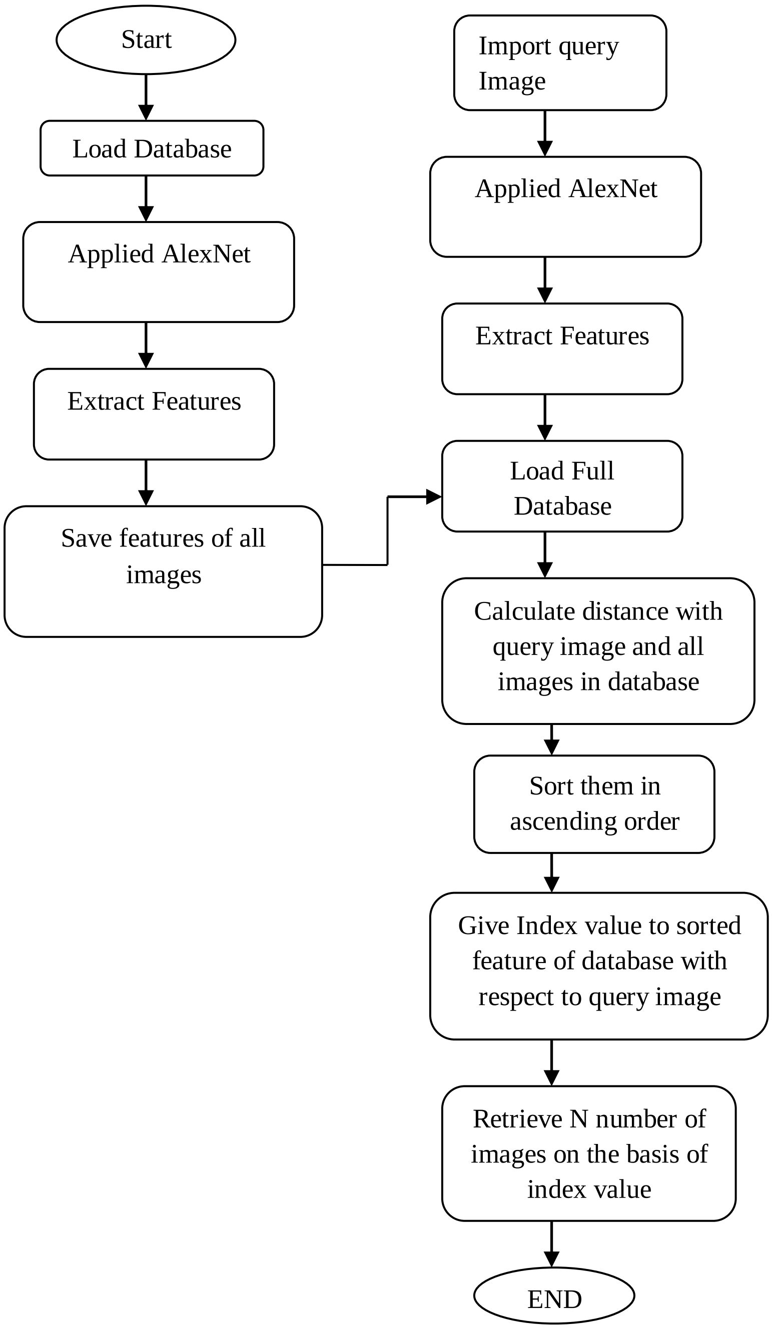


Figure4.12: Flow Chart of Indexing Technique

Pseudo-code of Indexing Technique

To generate indexes, first load the dataset, than apply Alexnet. Than save the dominate features of the dataset for each image. Than save the features of all images with dominating features.

In next step, import the query image, than apply Alexnet. Than extract the dominate features. After feature extraction of queried image, load the whole dataset.

Calculate distance with query image and all images stored in database. After calculation of distance, sort them in ascending order. Give index value to sorted features of database with respect to query image.

Than retrieve N number of images on the basis of index value.

4.5 K- FOLD CROSS VALIDATION

A given data set is split into 5 folds in a 5-fold CV, each of which functions as a testing set at some point. In this instance, the data set is separated into five folds. In the initial iteration, the model is evaluated on the first fold while being trained on the other folds. The left over data is used as the training set in the second iteration, while the second fold is used as the testing set. Up until a test set has been utilized for each of the five folds, this procedure is repeated.

4.6 EXPERIMENT AND RESULT ANALYSIS

We have used different feature extraction techniques along with clustering Techniques and for classification, different classifiers have been used. After the extraction of features using Alexnet, indexing technique is applied on the extracted features and most matched images is retrieved from the dataset.

We have evaluated the effectiveness of the categorization model using a number of assessment metrics, including:

1. Accuracy: It is the proportion of precise predictions to all predictions.

$$\text{Accuracy} = (TP + TN)/(TP + FP + TN + FN) \quad \dots (4.2)$$

2. **Sensitivity:** It is the proportion of true positives to all other positives found in the data.

$$\text{Sensitivity} = TP/(TP + FN) \quad \dots (4.3)$$

3. **Precision:** The ratio is the total predicted positives divided by the sum of true positive and

False positive.

$$\text{Precision} = TP/(TP + FP) \quad \dots (4.4)$$

4. **Specificity**: It is ratio of true negatives to total of true negative and false positive.

$$\text{Specificity} = TN/(TN + FP) \quad \dots (4.5)$$

5. F_Score: It's the harmonic mean of the precision and recall.

$$\text{F\_Score}=2 * (\text{Recall} * \text{Precision})/(\text{Recall} + \text{Precision}) \qquad \ldots (4.6)$$

True Positive Rate (**TPR**) is a synonym for recall, hence the following definition applies:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad \ldots (4.7)$$

False Positive Rate (**FPR**) is characterized as follows:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \qquad \ldots (4.8)$$

4.6.1 Experiments

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
FC Mean+KNN	94.00	94.12	94.10	94.58
GLCM+KNN	92.10	92.00	91.82	92.32
Kmean+KNN	93.00	92.00	92.87	93.72
LBP+KNN	92.49	91.10	91.85	92.94
PCA+KNN	93.10	93.00	92.10	93.54
SIFT+KNN	93.00	93.10	92.10	93.40

Table 4.1 Accuracy of KNN Classifier with different Feature Extraction Techniques

The above Table 4.1 shows the performance accuracy of KNN Classifier with different feature extraction techniques.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
FC Mean+KNN	72	72.50	72.80	72.90
GLCM+KNN	61.50	60.87	60.12	61.60

Kmean+KNN	68.10	68.00	67.82	68.60
LBP+KNN	66.80	62.10	63.70	64.70
PCA+KNN	67.60	67.15	66.50	67.70
SIFT+KNN	66.87	66.78	66.32	67.00

Table 4.2 Sensitivity of KNN Classifier with different Feature Extraction Techniques

The above Table 4.2 shows the performance sensitivity of KNN Classifier with different feature extraction techniques.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
FC Mean+KNN	74.20	73.87	74.10	74.72
GLCM+KNN	60.10	60.05	59.89	60.47
Kmean+KNN	70.26	70.12	70.61	70.62
LBP+KNN	62.10	62.50	62.05	63.50
PCA+KNN	67.00	66.84	65.10	67.05
SIFT+KNN	66.80	66.10	66.10	66.94

Table 4.3 Precision of KNN Classifier with different Feature Extraction Techniques

The above Table 4.3 shows the performance precision of KNN Classifier with different feature extraction techniques.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
FC Mean+KNN	96.80	95.12	95.85	96.99
GLCM+KNN	94.89	94.10	95.10	95.73
Kmean+KNN	96.15	95.16	95.15	96.51

LBP+KNN	95.85	95.10	95.86	96.08
PCA+KNN	96.10	95.78	94.71	96.41
SIFT+KNN	96.00	95.12	95.87	96.33

Table 4.4 Specificity of KNN Classifier with different Feature Extraction Techniques

The above Table 4.4 shows the performance specificity of KNN Classifier with different feature extraction techniques.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
FC Mean+KNN	71.00	70.87	70.10	71.11
GLCM+KNN	59.58	60.00	58.26	59.62
Kmean+KNN	66.59	65.80	65.95	66.95
LBP+KNN	63.85	61.88	61.87	62.59
PCA+KNN	66.01	65.78	65.10	66.05
SIFT+KNN	65.12	65.12	65.10	65.21

Table 4.5 F_Score of KNN Classifier with different Feature Extraction Techniques

The above Table 4.5 shows the performance F_Score of KNN Classifier with different feature extraction techniques.

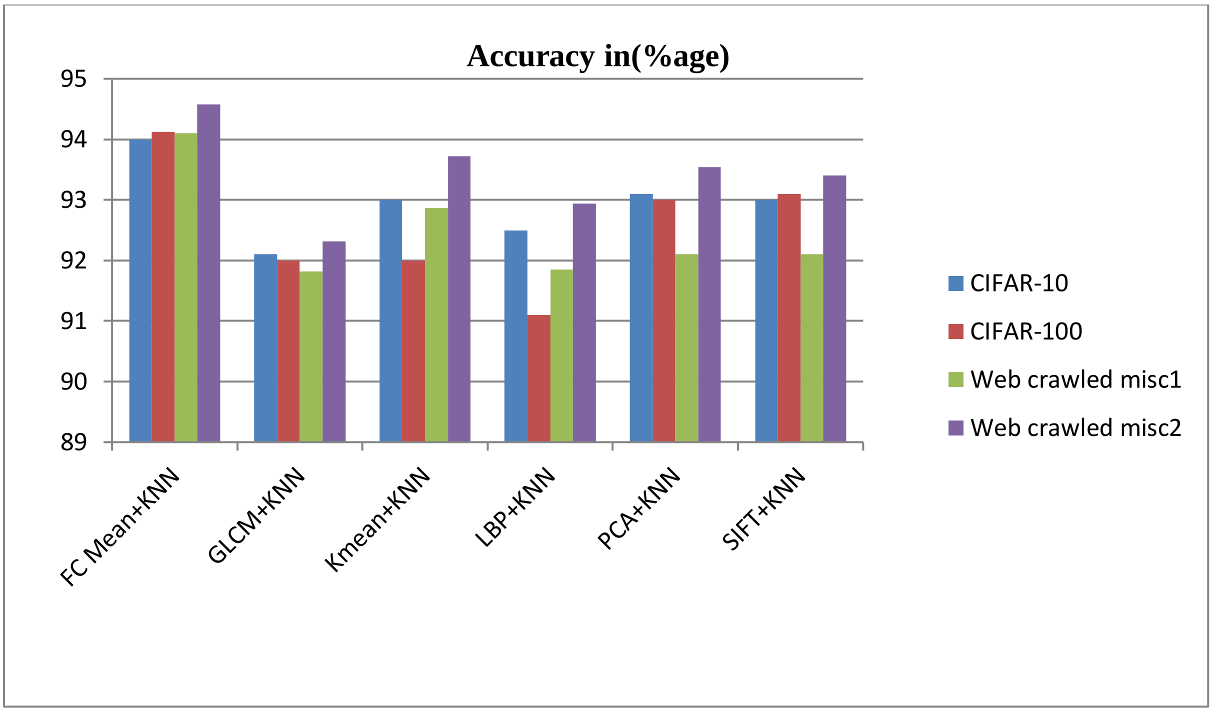


Figure 4.13: Accuracy of KNN Classifier with different Feature Extraction Techniques

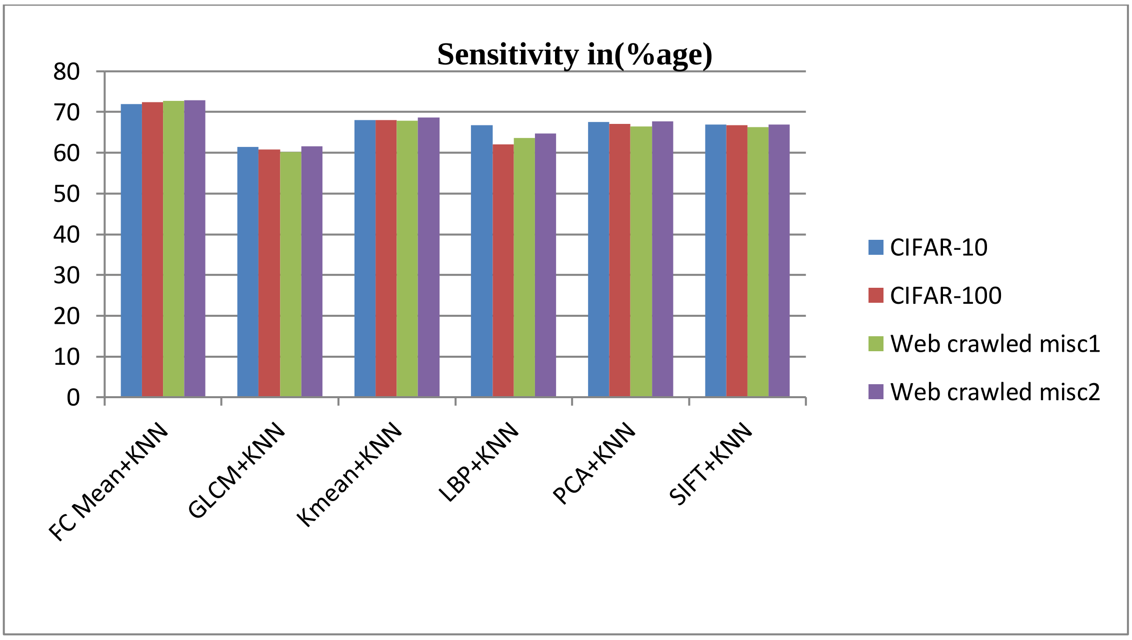


Figure 4.14: Sensitivity of KNN Classifier with different Feature Extraction Techniques

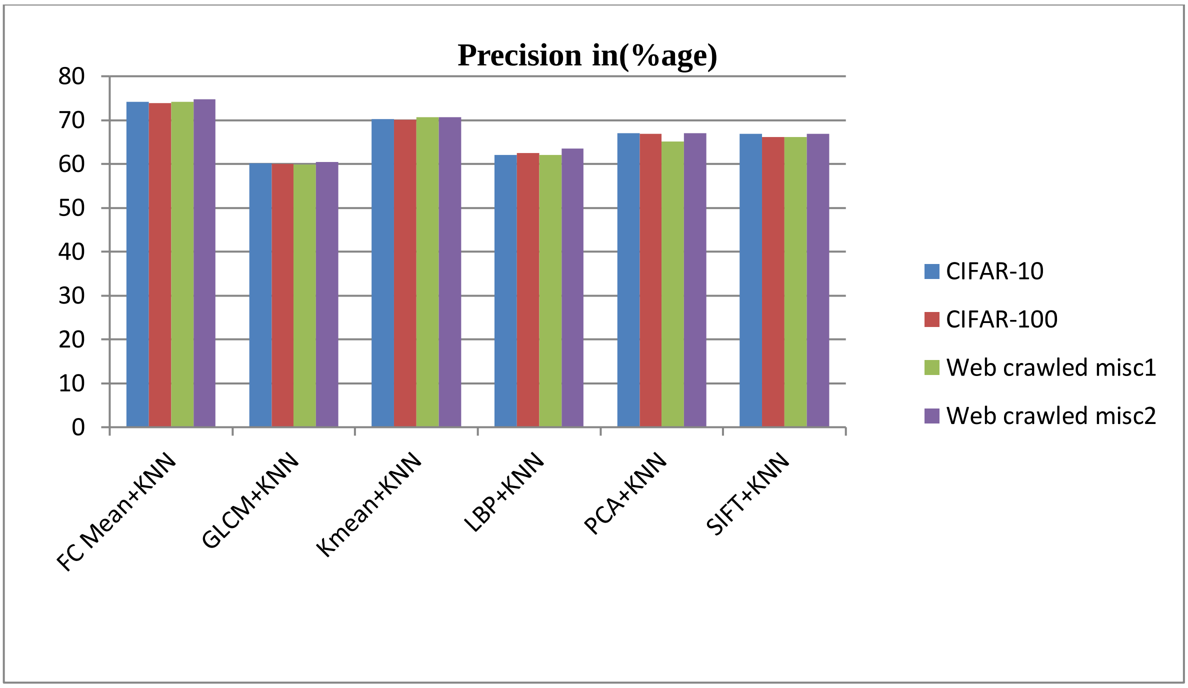


Figure 4.15: Precision of KNN Classifier with different Feature Extraction Techniques

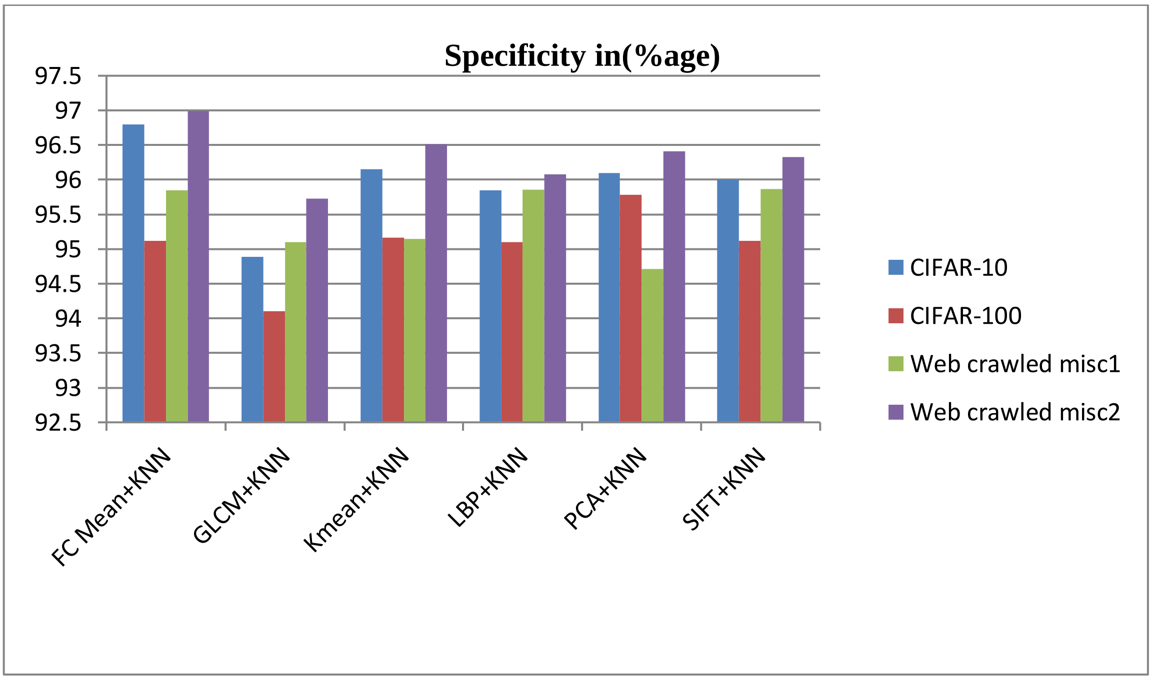


Figure 4.16: Specificity of KNN Classifier with different Feature Extraction Techniques

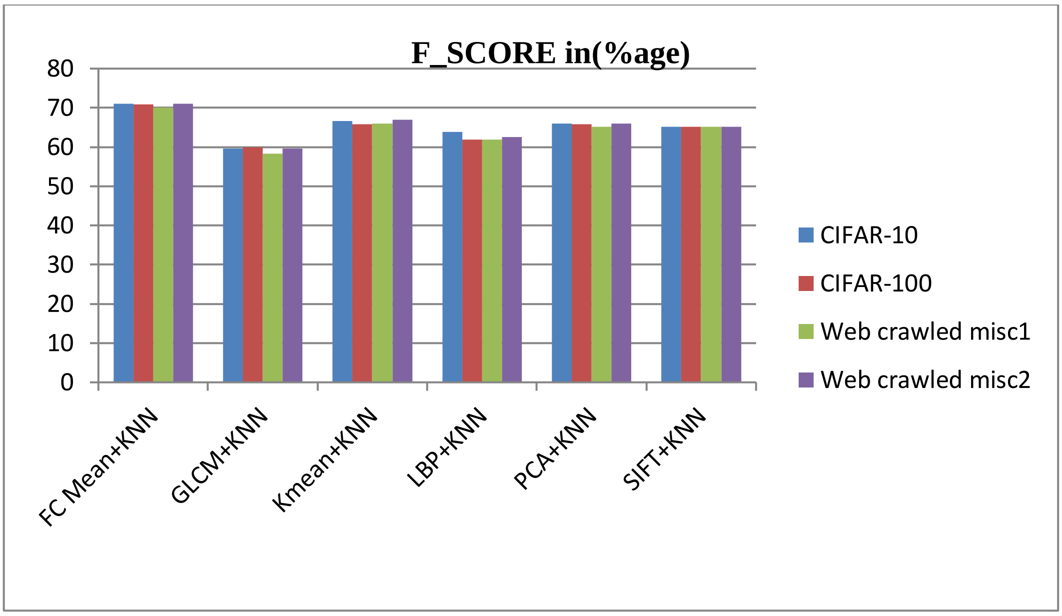


Figure 4.17: F_Score of KNN Classifier with different Feature Extraction Techniques

From Figure 4.13 to Figure 4.17, shows the performance of KNN classifier with different feature extraction techniques.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
FC Mean+SVM	95.10	95.12	95.10	95.68
GLCM+SVM	91.80	90.00	91.00	92.00
Kmean+SVM	94.82	95.00	94.10	95.78
LBP+SVM	92.10	92.00	91.87	92.28
PCA+SVM	93.00	93.10	92.82	93.28
SIFT+SVM	92.81	92.80	93.10	93.84

Table 4.6 Accuracy of SVM Classifier with different Feature Extraction Techniques

The above Table 4.6 shows the performance accuracy of SVM Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+SVM	78.20	78.10	77.16	78.40
GLCM+SVM	60.10	59.67	60.01	60.00
Kmean+SVM	77.80	78.11	77.20	78.90
LBP+SVM	61.00	60.10	60.10	61.40
PCA+SVM	65.80	65.10	65.00	66.40
SIFT+SVM	68.10	68.87	68.50	69.20

Table 4.7 Sensitivity of SVM Classifier with different Feature Extraction Techniques

The above Table 4.7 shows the performance sensitivity of SVM Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+SVM	78.17	78.12	76.89	78.54
GLCM+ SVM	60.00	61.00	60.11	61.23
Kmean+ SVM	78.10	78.89	78.12	79.02
LBP+ SVM	62.37	61.45	61.20	62.54
PCA+ SVM	65.10	64.57	66.05	66.58
SIFT+ SVM	68.00	68.15	68.20	69.17

Table 4.8 Precision of SVM Classifier with different Feature Extraction Techniques

The above Table 4.8 shows the performance precision of SVM Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+SVM	96.10	95.87	96.10	97.60
GLCM+SVM	95.10	94.87	94.88	95.56
Kmean+SVM	96.10	96.10	96.14	97.66
LBP+SVM	94.12	94.12	94.82	95.71
PCA+SVM	94.12	94.50	95.13	96.27
SIFT+SVM	95.10	94.20	94.32	96.58

Table 4.9 Specificity of SVM Classifier with different Feature Extraction Techniques

The above Table 4.9 shows the performance specificity of SVM Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+SVM	77.44	76.50	77.12	78.44
GLCM+SVM	60.00	61.00	60.00	60.50
Kmean+SVM	77.82	77.11	77.12	78.91
LBP+SVM	61.13	60.87	60.00	61.79
PCA+SVM	65.10	60.22	63.00	66.45
SIFT+SVM	64.87	67.12	68.21	69.12

Table 4.10 F_Score of SVM Classifier with different Feature Extraction Techniques

The above Table 4.10 shows the performance F_Score of SVM Classifier with different feature extraction techniques.

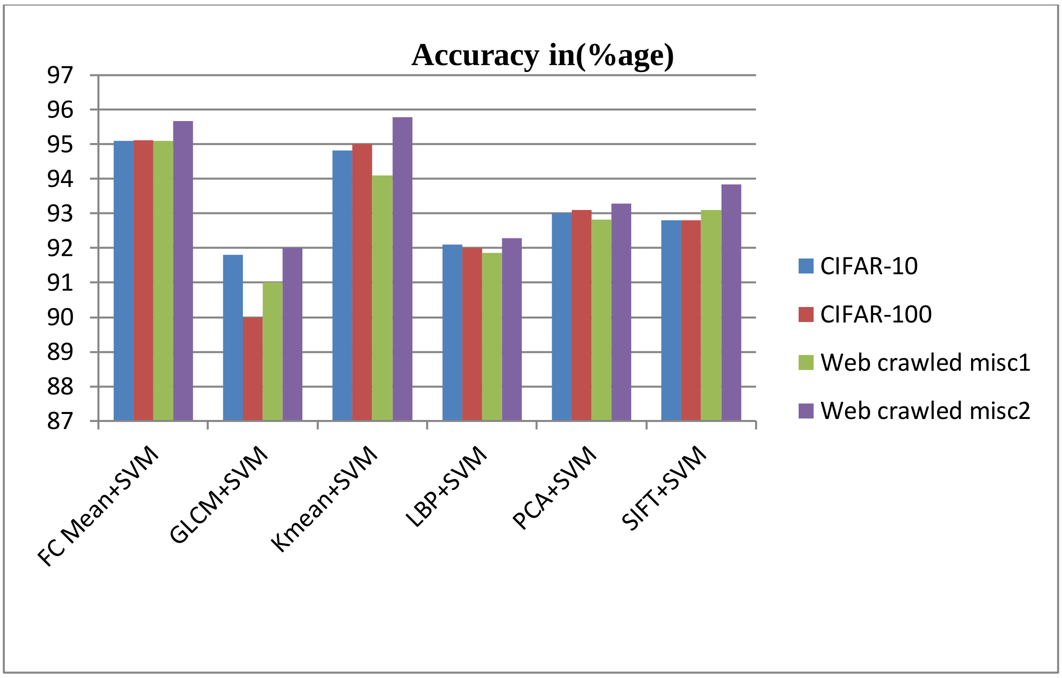


Figure 4.18: Accuracy of SVM Classifier with Feature Extraction Techniques

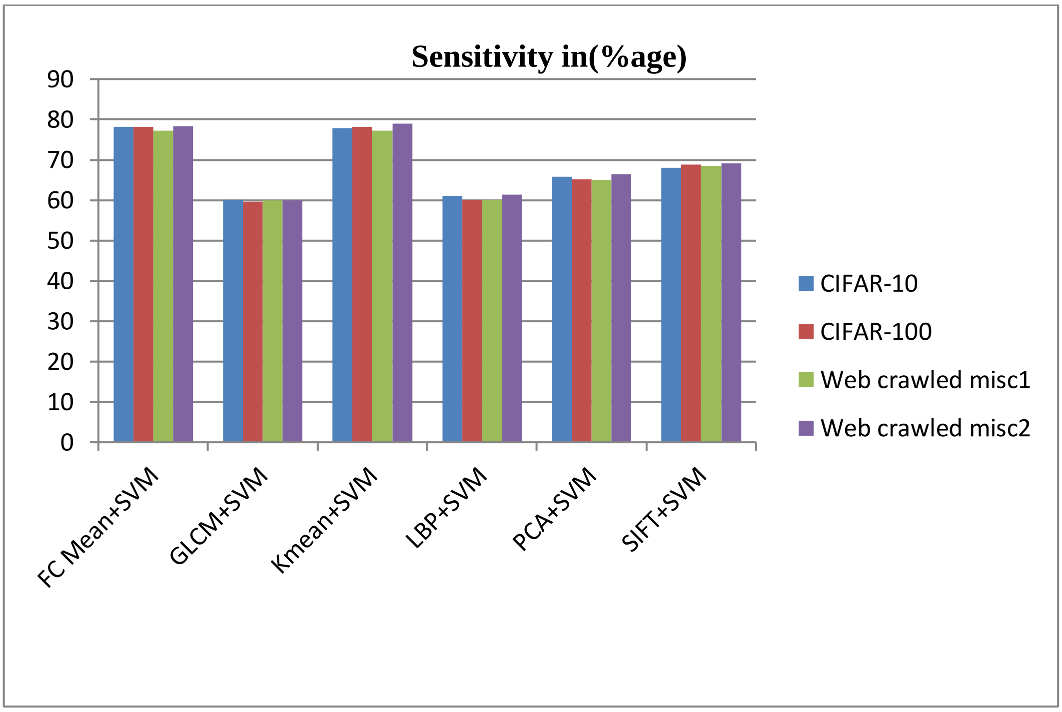


Figure 4.19: Sensitivity of SVM Classifier with different Feature Extraction Techniques

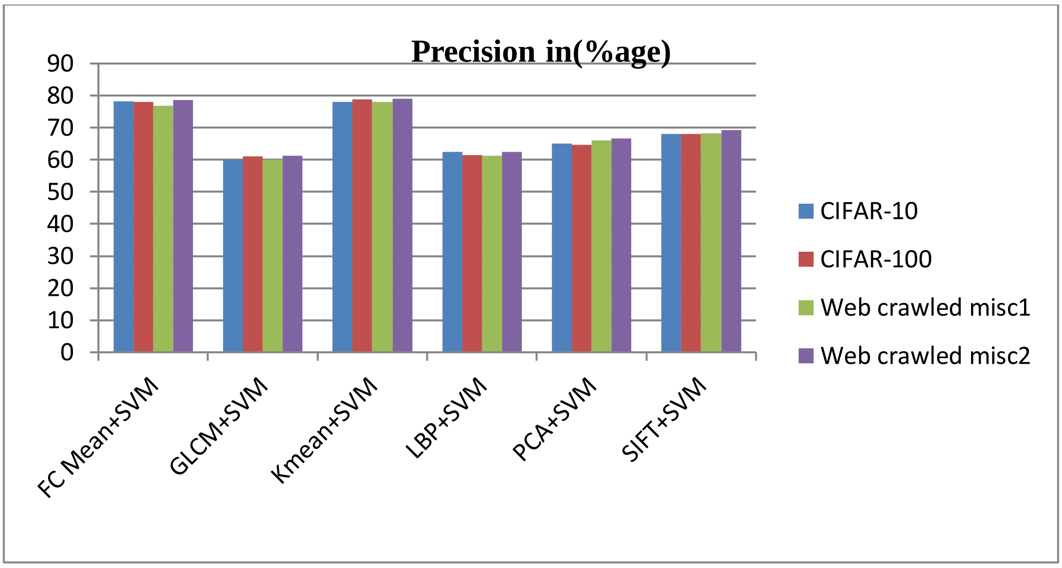


Figure 4.20: Precision of SVM Classifier with different Feature Extraction Techniques

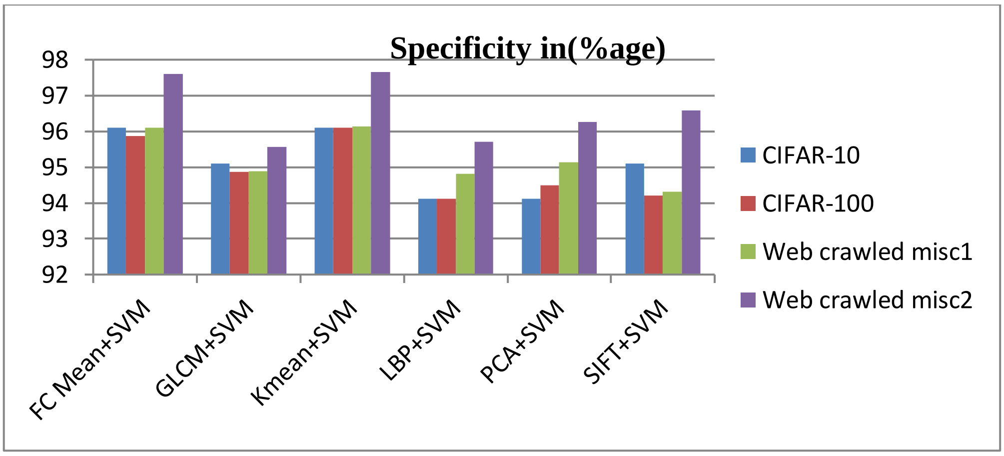


Figure 4.21: Specificity of SVM Classifier with different Feature Extraction Techniques

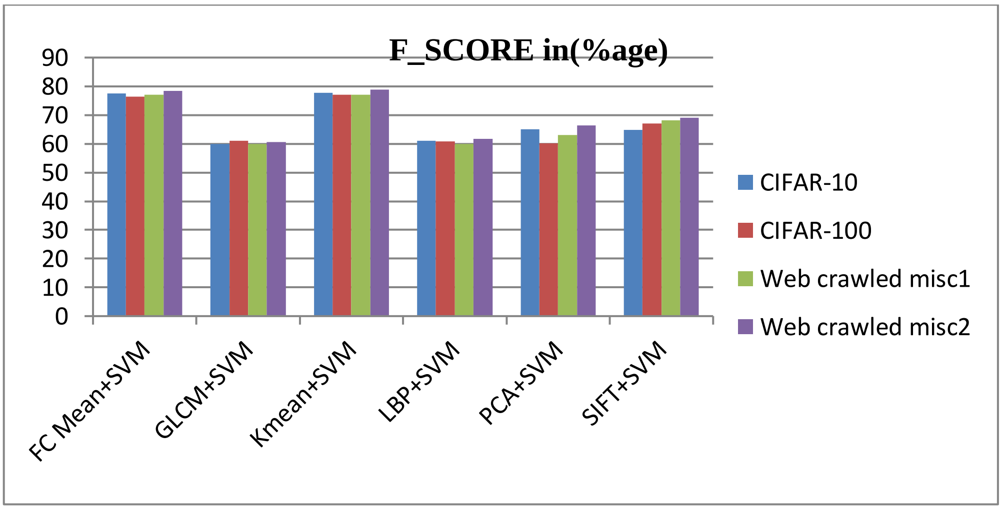


Figure 4.22: F_Score of SVM Classifier with different Feature Extraction Techniques

From Figure 4.18 to Figure 4.22, shows the performance of SVM Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+DT	95.10	95.25	94.12	95.86
GLCM+DT	92.00	91.86	91.89	92.24
Kmean+DT	95.00	94.32	95.10	95.60
LBP+DT	91.00	91.15	92.10	92.62
PCA+DT	93.00	92.10	93.00	93.62
SIFT+DT	92.00	93.10	91.50	93.68

Table 4.11 Accuracy of DT Classifier with different Feature Extraction Techniques

The above Table 4.11 shows the accuracy of DT Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+DT	78.10	79.00	78.32	79.30
GLCM+DT	60.00	60.12	59.87	61.20
Kmean+DT	76.00	75.10	77.10	78.00
LBP+DT	62.15	62.89	63.04	63.10
PCA+DT	67.50	66.50	67.12	68.10
SIFT+DT	68.00	67.00	68.10	68.40

Table 4.12 Sensitivity of DT Classifier with different Feature Extraction Techniques

The above Table 4.12 shows the performance Sensitivity of DT Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+DT	78.87	79.12	78.89	79.23
GLCM+DT	62.10	61.16	61.10	62.30
Kmean+DT	77.12	76.85	76.72	78.37
LBP+DT	67.68	68.10	63.89	64.14
PCA+DT	67.77	68.10	67.10	68.67
SIFT+DT	67.82	68.12	68.12	68.83

Table 4.13 Precision of DT Classifier with different Feature Extraction Techniques

The above Table 4.13 shows the resulting precision of DT Classifier with multiple feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+DT	96.50	96.17	96.10	97.70
GLCM+DT	94.32	94.85	93.85	95.69
Kmean+DT	96.65	96.56	96.10	97.56
LBP+DT	95.64	95.45	95.12	95.90
PCA+DT	95.50	94.12	95.64	96.46
SIFT+DT	96.00	95.82	94.13	96.49

Table 4.14 Specificity of DT Classifier with different Feature Extraction Techniques

The above Table 4.14 shows the performance Specificity of DT Classifier with different feature extraction techniques.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+DT	78.11	79.10	78.12	79.17
GLCM+DT	60.75	60.10	60.10	61.60
Kmean+DT	77.87	76.87	77.12	78.08
LBP+DT	67.32	68.00	62.58	63.47
PCA+DT	68.12	68.10	68.12	68.23
SIFT+DT	67.82	65.55	67.89	68.54

Table 4.15 F_Score of DT Classifier with different Feature Extraction Techniques

The above Table 4.15 shows the performance F_Score of DT Classifier with different feature extraction techniques.

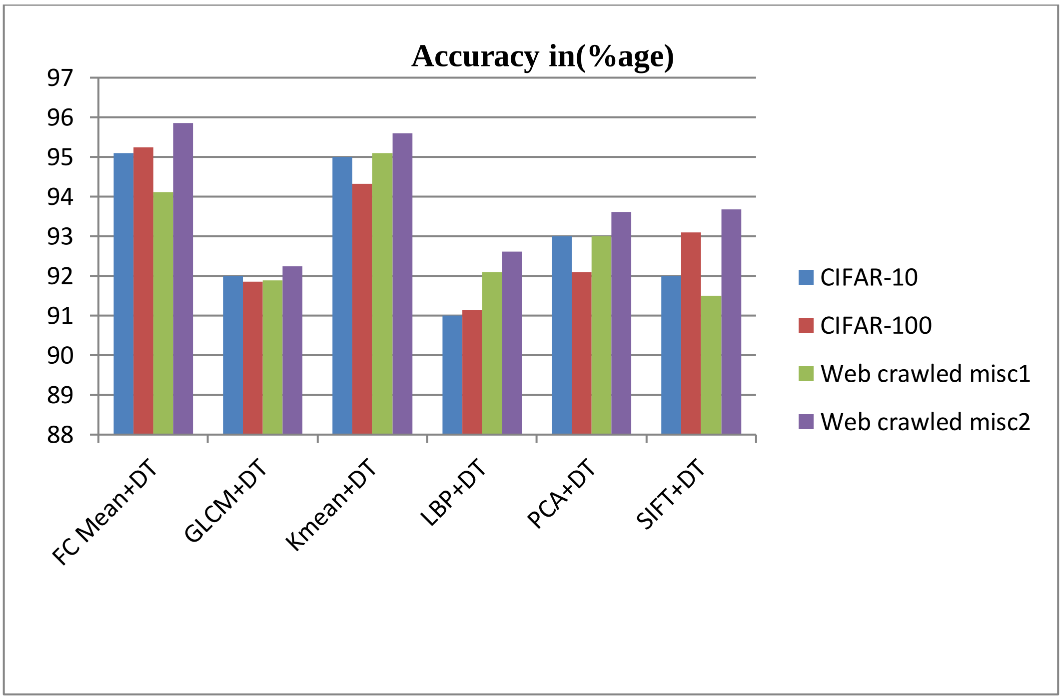


Figure 4.23: Accuracy of DT Classifier with different Feature Extraction Techniques

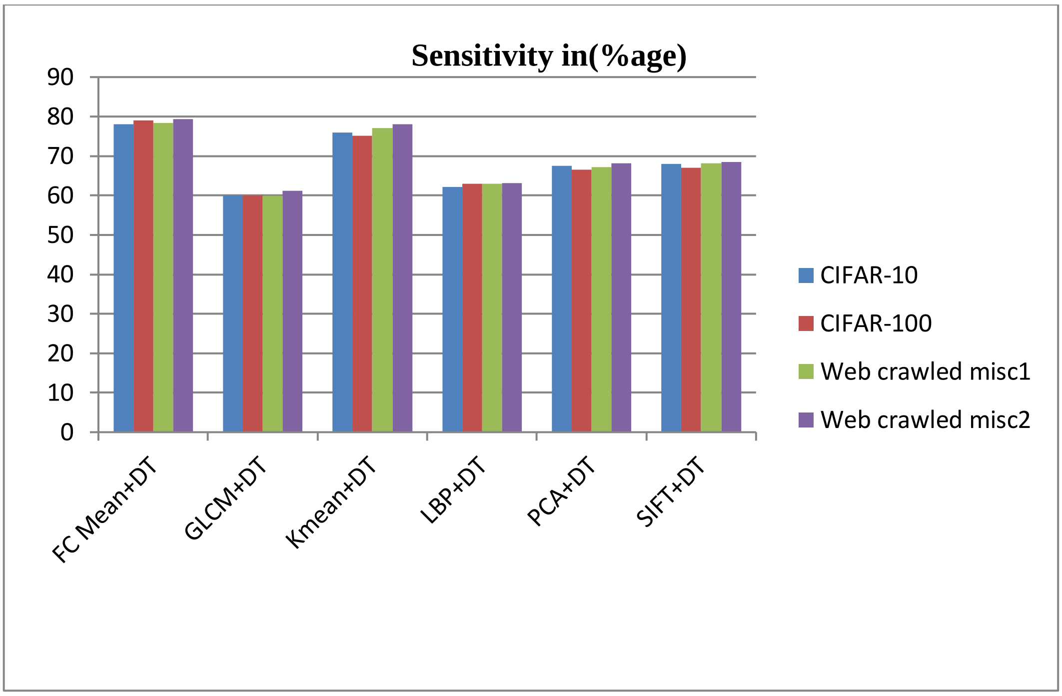


Figure 4.24: Sensitivity of DT Classifier with different Feature Extraction Techniques

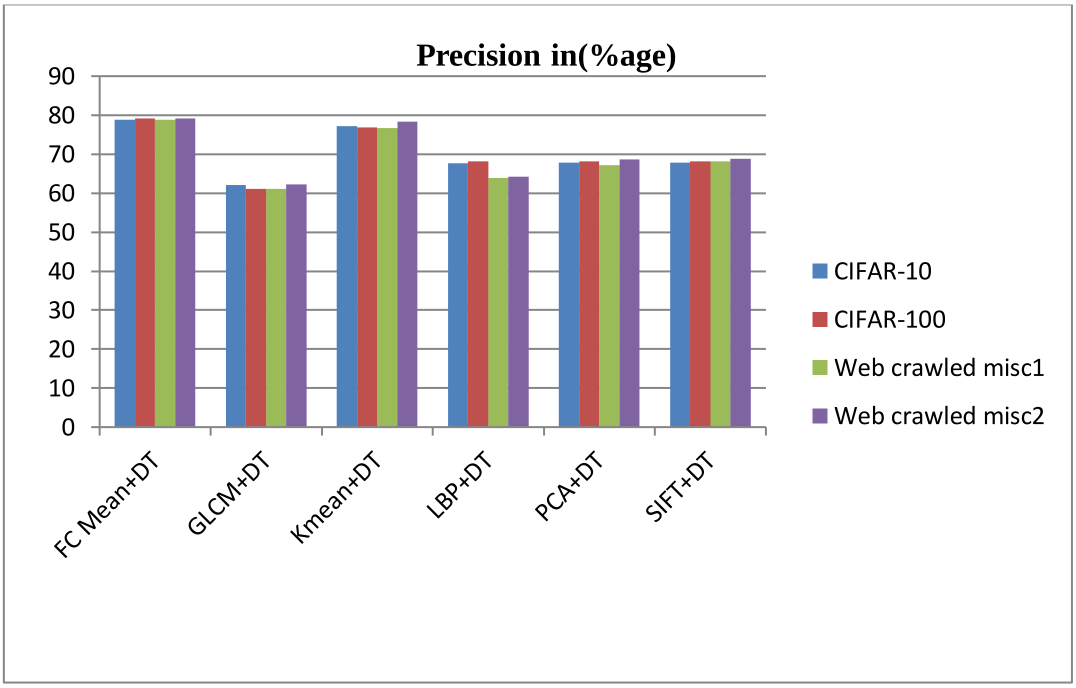


Figure 4.25: Precision of DT Classifier with different Feature Extraction Techniques

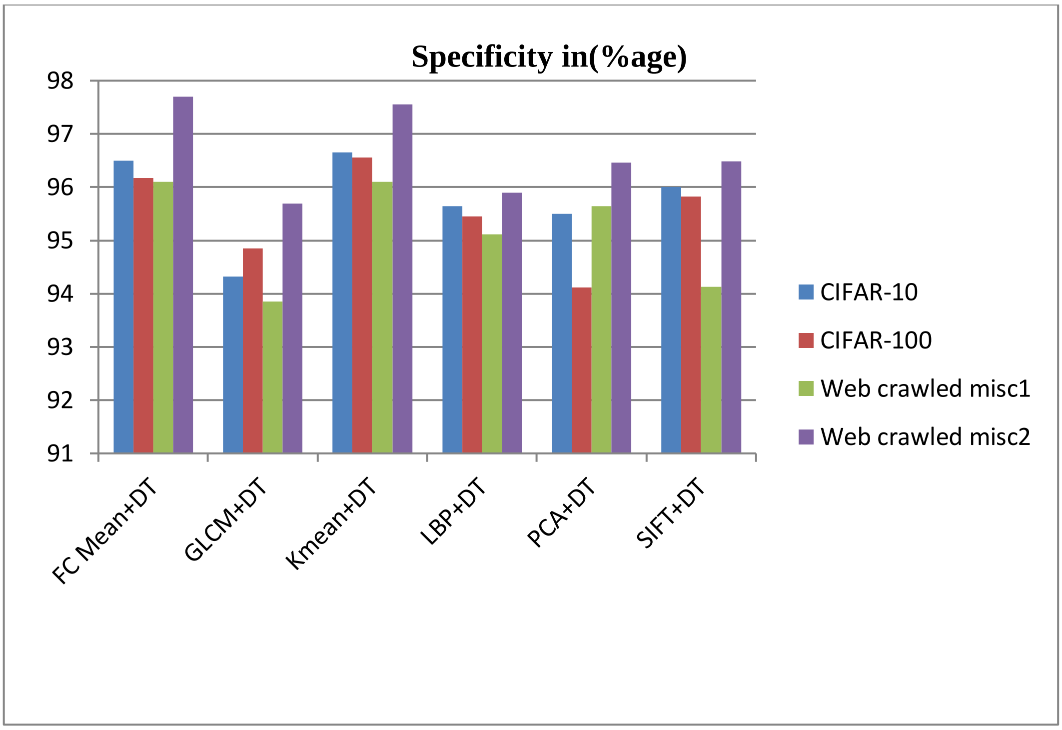


Figure 4.26: Specificity of DT Classifier with different Feature Extraction Techniques

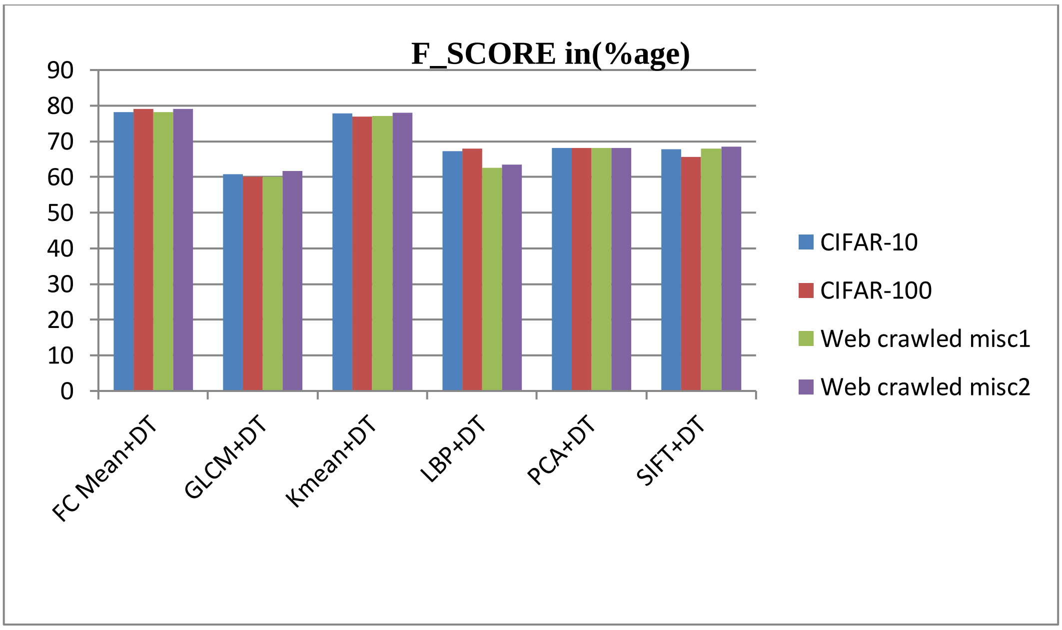


Figure 4.27: F_Score of DT Classifier with different Feature Extraction Techniques

From Figure 4.23 to Figure 4.27, comparison of DT Classifier with different feature extraction techniques.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+GT	96.10	96.20	96.50	97.44
GLCM+GT	92.40	92.30	93	93.50
K MEAN+GT	95.10	95.00	94.20	95.98
LBP+GT	93.80	93.20	94.10	94.60
PCA+GT	94.10	94.82	94.28	95.02
SIFT+GT	94.67	96.10	93.82	94.70

Table 4.16 Accuracy of GT Classifier with different Feature Extraction Techniques

The above Table 4.16 shows the performance accuracy of GT Classifier with different feature selection approaches.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
Fuzzy an+GT	97.15	98.00	97.85	98.58
GLCM+GT	94.50	93.70	92.20	96.39
K MEAN+GT	96.76	96.10	96.11	97.76
LBP+GT	96.50	95.87	95.60	97.00
PCA+GT	96.10	95.89	95.87	97.23
SIFT+GT	96.10	96.05	96.12	97.05

Table 4.17 Sensitivity of GT Classifier with different Feature Extraction Techniques

The above Table 4.17 shows the Sensitivity of GT Classifier with different feature extraction approaches.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+GT	96.20	96.50	96.89	97.21
GLCM+GT	93.00	93.10	91.10	93.11
K MEAN+GT	94.60	94.50	94.12	95.67
LBP+GT	94.20	93.11	94.00	94.24
PCA+GT	93.20	92.85	93.95	94.65
SIFT+GT	94.21	94.10	94.12	94.32

Table 4.18 Precision of GT Classifier with different Feature Extraction Techniques

The above Table 4.18 shows the Precision of GT Classifier with different feature extraction approaches.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC Mean+GT	97.54	98.21	97.51	98.58
GLCM+GT	95.40	95.50	92.13	96.39
K MEAN+GT	96.11	97.10	96.68	97.77
LBP+GT	96.00	95.12	95.89	97.00
PCA+GT	96.25	95.10	96.12	97.23
SIFT+GT	96.10	96.08	96.10	97.06

Table 4.19 Specificity of GT Classifier with different Feature Extraction Techniques

The above Table 4.19 shows the Specificity of GT Classifier with different feature extraction approaches.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
FC mean+GT	96.98	97.50	97.20	97.89
GLCM+GT	94.17	93.25	94.10	94.71
K MEAN+GT	95.60	96.12	95.45	96.69
LBP+GT	94.82	93.18	93.99	95.59
PCA+GT	94.91	94.89	95.10	95.91
SIFT+GT	95.10	94.12	94.10	95.65

Table 4.20 F_Score of GT Classifier with different Feature Extraction Techniques

The above Table 4.20 shows the F_Score of GT Classifier withdifferent feature extraction approaches.

In Table 4.16-4.20, after extracting the features, and then by classifying the features using Gradient Boosting Algorithm, different performance parameters have been calculated. It has been observed here that with Fuzzy C mean feature clustering technique, accuracy achieved for CIFAR-10 is 96.10%, for CIFAR-100, it is 96.20%, for Web-Crawled misc1, it is 96.50% and for Web-Crawled misc2 it is 97.44%. Sensitivity achieved for CIFAR-10 is 97.15%, for CIFAR-100, it is 98.00%, for Web-Crawled misc1, it is 97.85% and for Web-Crawled misc2 it is 98.58%. Precision achieved for CIFAR-10 is 96.20%, for CIFAR-100, it is 96.50%, for Web-Crawled misc1, it is 96.89% and for Web-Crawled misc2 it is 97.21%. Specificity achieved for CIFAR-10 is 97.54%, for CIFAR-100, it is 98.21%, for Web-Crawled misc1, it is 97.51% and for Web-Crawled misc2 it is 98.58%. F_Score achieved for CIFAR-10 is 96.98%, for CIFAR-100, it is 97.50%, for Web-Crawled misc1, it is 97.20% and for Web-Crawled misc2 it is 97.89%.

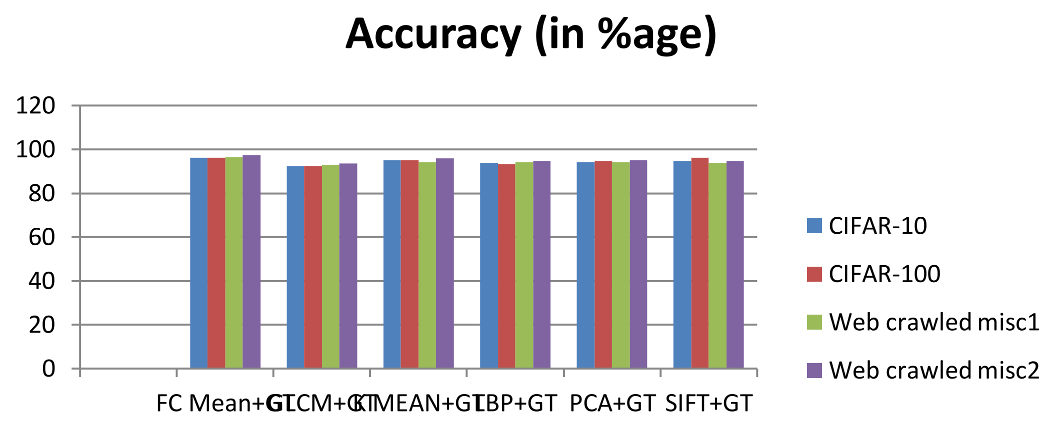


Figure 4.28: Accuracy of GT Classifier with different Feature Extraction Techniques

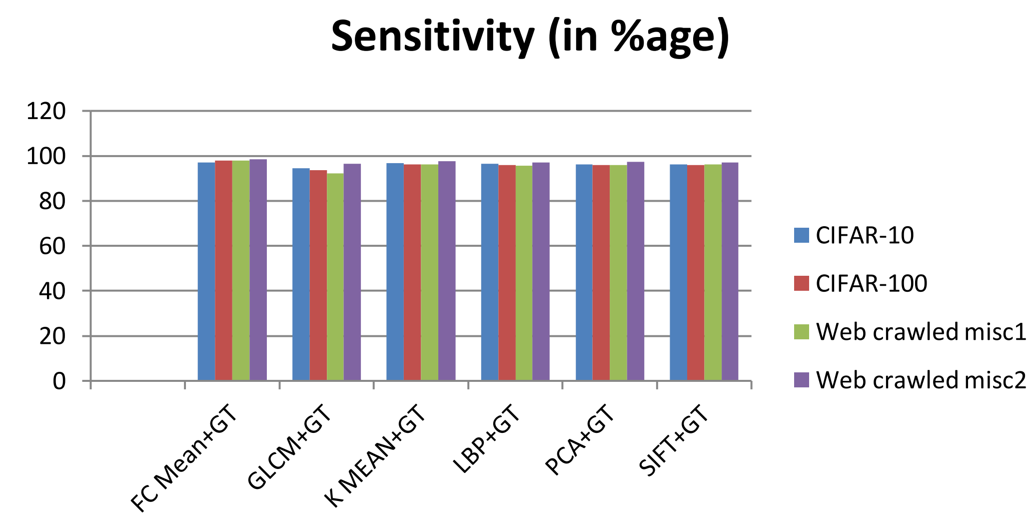


Figure 4.29: Sensitivity of GT Classifier with different Feature Extraction Techniques

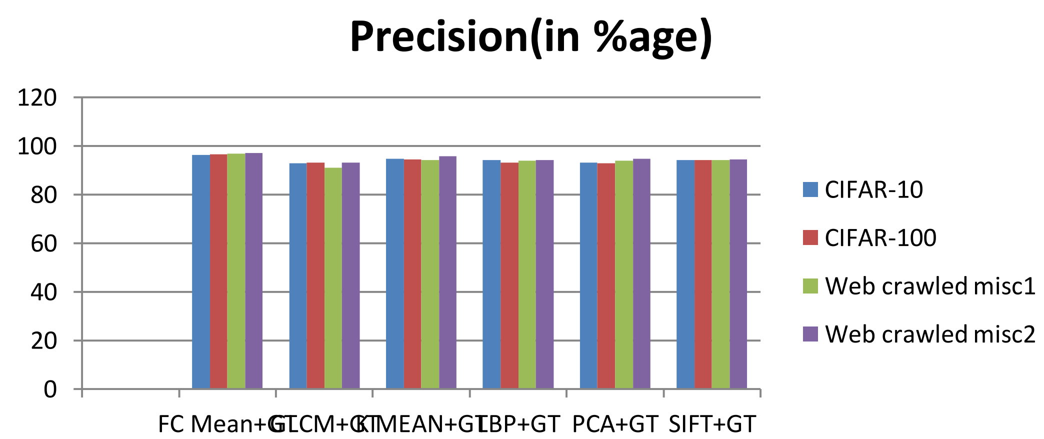


Figure 4.30: Precision of GT Classifier with different Feature Extraction Techniques

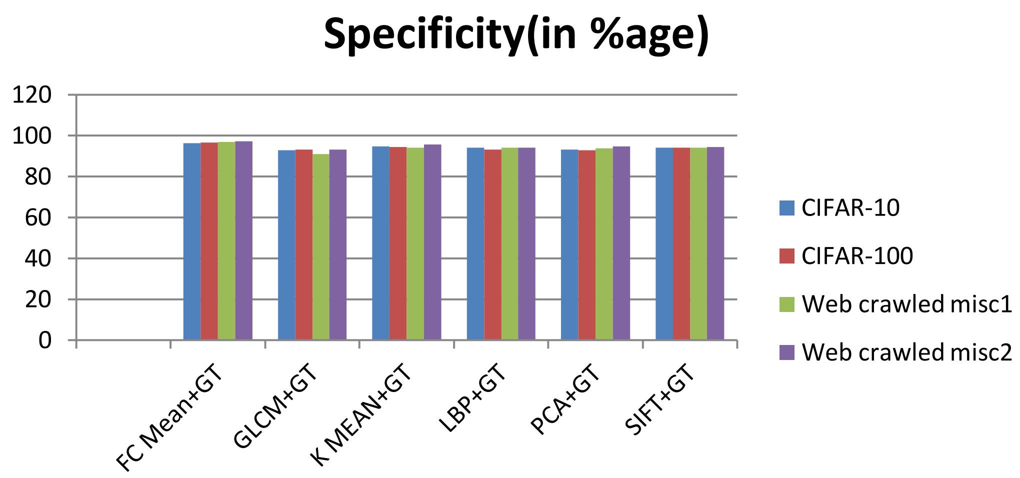


Figure 4.31: Specificity of GT Classifier with different Feature Extraction Techniques

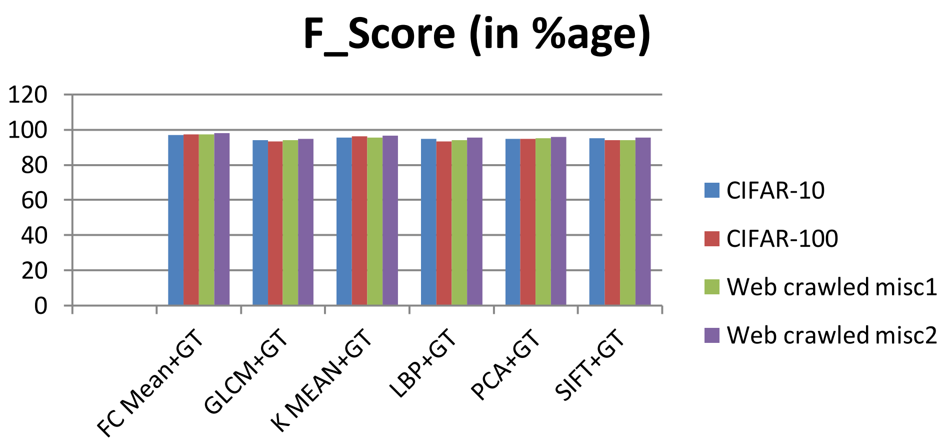


Figure 4.32: F-Score of GT Classifier with different Feature Extraction Techniques

In Figure 4.28-4.32, after extracting the features, and then by classifying the features using Gradient Boosting Algorithm, different performance parameters have been calculated. It has been observed here that with Fuzzy C mean feature clustering technique, accuracy achieved for CIFAR-10 is 96.10%, for CIFAR-100, it is 96.20%, for Web-Crawled misc1, it is 96.50% and for Web-Crawled misc2 it is 97.44%. Sensitivity achieved for CIFAR-10 is 97.15%, for CIFAR-100, it is 98.00%, for Web-Crawled misc1, it is 97.85% and for Web-Crawled misc2 it is 98.58%. Precision achieved for CIFAR-10 is 96.20%, for CIFAR-100, it is 96.50%, for Web-Crawled misc1, it is 96.89% and for Web-Crawled misc2 it is 97.21%. Specificity achieved for CIFAR-10 is 97.54%, for CIFAR-100, it is 98.21%, for Web-Crawled misc1, it is 97.51% and for Web-Crawled misc2 it is 98.58%. F_Score achieved for CIFAR-10 is 96.98%, for CIFAR-100, it is 97.50%, for Web-Crawled misc1, it is 97.20% and for Web-Crawled misc2 it is 97.89%.

Apart from this, we have also shown the results corresponding to AlexNet CNN Classifier. After extracting the features, we have done indexing and then ranking of the images features in ascending order is done. On the basis of their index values, top ten images are retrieved based on the query image.

Figure 4.33: Retrieved images of Web-Crawled misc1 Dataset based on the indexing

Figure 4.34: Retrieved images of Web-Crawled misc2 Dataset based on the indexing

Confusion Matrix

A classification technique for classifying models is a confusion matrix. To evaluate a classification models performance on a set of test data, utilise a square matrix table. The confusion matrix in Figure 4.35 shows what our model got right and wrong. The samples that the model successfully identified are represented by the numbers along the diagonal (the green boxes), whereas the samples that the model incorrectly classified are represented by the numbers along the off-diagonal (the red boxes) (the pink boxes). The percentage of correctly predicted samples and incorrectly predicted samples is shown, respectively, by the green and pink values in the white boxes.The misclassification rate is indicated by the red percentage value, while the classification accuracy is indicated by the green% number in the only grey box at the bottom of the diagonal.

Figure 4.35 Confusion Matrix of Dataset

ROC curve (**receiver operating characteristic curve**)

It is a graph that shows how well a classification model performs at every level of categorization. On this curve, two parameters are displayed:

- True Positive Rate
- False Positive Rate

TPR vs. FPR are shown for various classification criteria on a ROC curve. More objects are classified as positive when the classification threshold is lowered, which raises the number of both False Positives and True Positives. The accompanying figure displays an example of a ROC Curve.

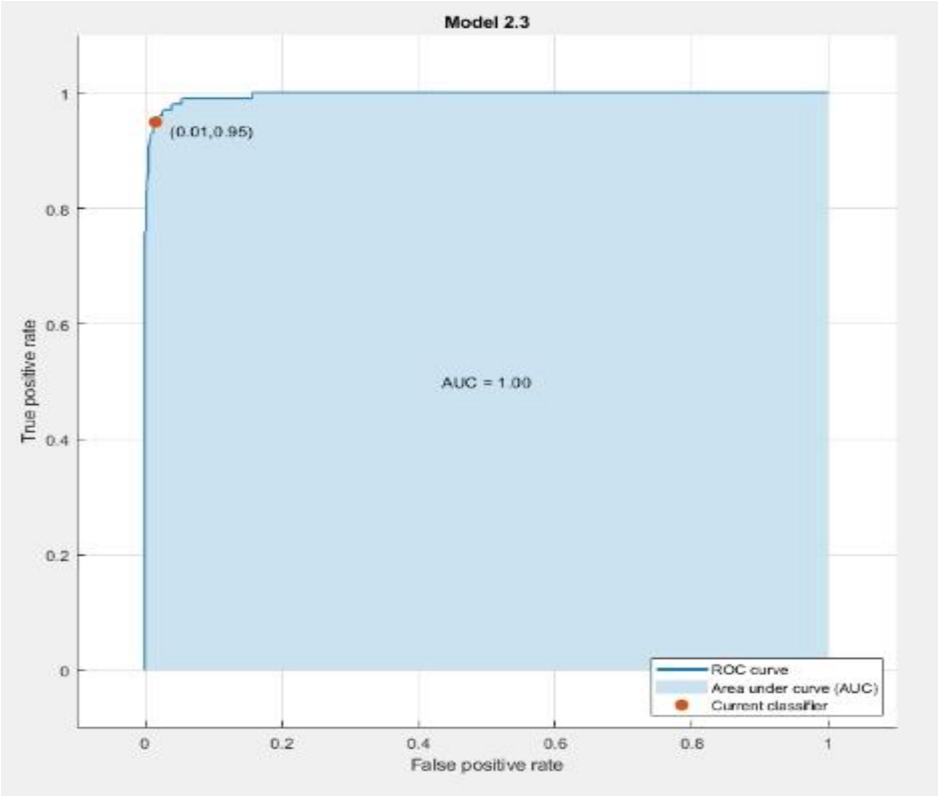


Figure 4.36: ROC Curve For Results

AUC provides a thorough analysis of performance using all relevant categorization criteria. The AUC is the likelihood that a model would judge a randomly chosen positive example more favourably than a randomly chosen negative example. Take a look at the examples below, which are presented in increasing order of logistic regression predictions from left to right.

4.6.2 Result Analysis

In this chapter, we have extracted the features using AlexNet CNN Feature Extractor and by using Fuzzy c Mean, K mean, GLCM, PCA, LBP and SIFT. By using AlexNet CNN Classifier, we got total 4096 Features. We also observe that by use of Fuzzy C mean, we are getting better results as compare to other feature extraction Techniques along with Gradient Boosting Classifier.

FCM produces most compact clusters as compare to other clustering techniques. Gradient Boosting reduces error mainly by reducing bias (and also to some extent variance, by aggregating the output from many models) it has lower number of hyper parameters to be tuned. Therefore, it is faster to have a best setting model. It updates the weights based on the gradients, which are less sensitive to outliers, which yield to the better performance as compare to other techniques.

4.7 SUMMARY

In this chapter we have used different Classifiers with different feature extraction and clustering techniques. We have observed that Fuzzy C mean has improved the results greatly in the suggested approach. It means the combination of Fuzzy C mean along with Gradient Boosting Algorithm, have given us the best results in terms of accuracy, sensitivity, precision, specificity and F_Score. So to optimize the features in next chapter, we will proceed with FC mean feature clustering Technique.

CHAPTER 5

PERFORMANCE OF OPTIMIZATION TECHNIQUES

The motive of this chapter is create an optimization approach that will enable us to increase the accuracy and to shorten the time spent for searching and matching the query image with images that are already stored in different datasets following by feature extraction using various methods. We have proposed a Markov chain rule-based feature selection (MCPL-FS) after the ranking step. This study employs a probabilistic latent graph-based feature selection technique which performs the ranking phase and also takes into account all potential subsets of features as pathways on a graph, successfully avoiding the combinatorial problem.

When implementation performed, we got 204 features in FC Mean ,13 in GLCM, 199 in Kmean ,51 in LBP, 99 in PCA and 48 in SIFT respectively. After this, we have selected Fuzzy CMean, as we got higher accuracy with it and then we applied feature optimization technique on it and we got 179 features after the optimization.

5.1 PROPOSED METHODOLOGY

The CBIR system calculates a feature vector for each image in the database and stores it in the database based on the visual feature. Firstly, the feature vector will be calculated whenever a user executes a query. Using the similarity method, the query vector is to be compared to all the vectors in the visual feature database and the closest image to the input image, known as the query image is given to the user. Therefore, the performance of the CBIR system is highly dependent on the features extracted from the image.

5.1.1 Automated ranking of features by using the Markov chain rule

To illustrate the working of the Markov Chain Rule, a weather model is considered as an example. It provides the transition probability given the current state. If the weather is cloudy, there is a probability of 50% of having rain and a 40% chance of having a sunny day. Thus, it shows that the probability of being sunny gives that to have cloudy depends only on the previous state, which was cloudy. Therefore, it does not consider the prior state.

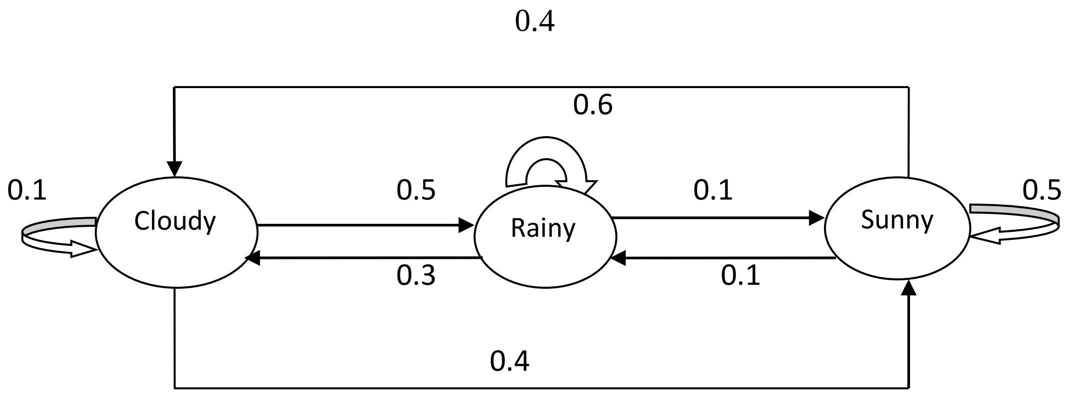


Figure 5.1: Markov Chain Rule Basic Model

As already discussed, extraction and selection representing images semantic substance play a central role in CBIR. This study proposed a probabilistic latent graph algorithm for feature selection.

The proposed PLSA considers data subsets of feature as a node and edge as a relation between nodes. We have considered those features as connecting nodes which make a path. The three essential components of this strategy are as follows:

Pre-processing: On raw feature distribution$\overrightarrow{Xi}$, a quantization process is applied and their values are mapped into a smaller group of sets which is a countable nominal. During this phase, each raw feature $\overrightarrow{Xi}$ is assigned to a descriptor fi.

- **Graph-Weighting*:*** The feature fi and a weighted edge between feature models are represented by nodes in an undirected, fully connected graph. Based on stochastic latent semantic analysis, characteristics xi and xj are likely to be associated (PLSA). As a mixture of conditional independent multinomial distributions for parameter estimation, fi, the weights are automatically learned [46]. Calculate the likelihood of each concurrent event in fj. A modelled expected value maximization (EM) algorithm is used.

5.1.2 Ranking Technique

Markov chain rule feature selection ((MRC-FS)) is applied [49] which works based on considering paths among nodes that can investigate the redundancy of any features in any set.

T_s is the training set with a set of feature distributions T_s = { $\overrightarrow{Ts1}$., ..., $\overrightarrow{Tsn}$.}, built undirected graph G, where edges indicate features represented by nodes and relationships between node pairs. An adjacency matrix A_m is used, it explains how the weighted edges of each element ofA_m, $1 \leq y,z \leq n$behave. Each weight represents the propensity of traits ~xy and ~xz to be strong choices.

$$a_{ij}= \phi(\sim x_y,\sim x_z), \qquad \text{... (5.1)}$$

Where $\phi(\sim x_y,\sim x_z)$ describes the potential function which is used by the PLSA-inspired framework.

5.1.3 Graph weighting using Markov chains and random walks using co-occurrences

This section gives us information of the algorithm which we have proposed and is based on ''Absorbing Random walks.'' Markov chain rule set of nodes is known as states and every move is known as a step. We supposed T_{ransp} as a matrix of transition probabilities of the Markov chain. Here s_i denotes chain with the current state, while as s_j denotes the moves with the successive state and t_{ij} at the next step with a probability.States which are once reached can never make a transition out of (i.e.,t_{ii}= 1). Transient mean a state, i.e., not absorbing. Transition Matrix can be expressed using equation (5.2).

$$T_{rans}=\begin{bmatrix} I & 0 \\ RS & A \end{bmatrix} \qquad \text{... (5.2)}$$

Transition probabilities from non-absorbed state to absorbed state are specified by the ''RS,'' which is known as rectangular matrix, probabilities from the non-absorbed state to the non-absorbed state are specified by ''AI'' which is a square submatrix. The Identity matrix is denoted by ''I,'' while as, for rectangular matrix ''0'' is used.

If ''a'' is in the absorbed state and ''b'' is in the non-absorbed state, R is b * a, A is b * b, I is a * a, and 0 is a * b.By iterative multiplication of the trans matrix.

$$T^2_{rans} =\begin{bmatrix} I & 0 \\ RS & A \end{bmatrix}\begin{bmatrix} I & 0 \\ RS & A \end{bmatrix}$$

$$=\begin{bmatrix} I & 0 \\ RS + ARS & A * A \end{bmatrix} \qquad \text{… (5.3)}$$

$$T^3_{rans}=\begin{bmatrix} I & 0 \\ RS + ARS & A * A \end{bmatrix}\begin{bmatrix} I & 0 \\ RS & A \end{bmatrix}=\begin{bmatrix} I & 0 \\ R + ARS + A * ARS & A * A * A \end{bmatrix} \qquad \text{… (5.4)}$$

By induction, we got the following expression presented in equation (5.5):

$$T^l_{rans}=\begin{bmatrix} I & 0 \\ (I + A + A * A + \cdots + Al - 1)RS & Al \end{bmatrix} \qquad \text{… (5.5)}$$

$A^l \to 0$ as $l \to \infty$. Thus

$$T^{\infty} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ CR & \mathbf{0} \end{bmatrix} \quad \dots (5.6)$$

$$C = \mathrm{I} + A + A^2 + \dots + A^{\infty} = (I - A)^{-1} \quad \dots (5.7)$$

To represent fundamental matrix for the absorbing chain is represented by matrix ‘‘C.’’

Based on the degree of relevance to sort these features automatically ranked framework is used to rank the training data.

We considered the relevancy and irrelevancy of the proposed method and introduced an unobserved class variable CV = $\{cv_1, cv_2\}$.

For each feature f_{eature}, there is a distribution $P_{rob}(V \backslash f_{eature})$ over a predetermined number of tokens.

$$P_{\mathrm{rob}}\,(t_{oken}|f_{eature}) = P_{\mathrm{rob}}\,(t_{oken}|z_1)\;P\,(z_1|f_{eature}) + P_{\mathrm{rob}}\,(t_{oken}|z_2)\;P\,(z_2|f_{eature}) \quad \dots (5.8)$$

For features, $P_{\mathrm{rob}}\,(f_{eature}) = {}^{\mathrm{Yn}}P_{\mathrm{rob}}\,(t_{oken}|z_1)\;P_{\mathrm{rob}}\,(z_1|f_{eature}) + P_{\mathrm{rob}}\,(t_{oken}|z_2)\;P_{\mathrm{rob}}\,(z_2|f_{eature})\,^{\circ}$ … (5.9)

To compute these parameters for PLSA, we have used maximum likelihood shown in the below equation (5.10).

$$\mathrm{L} = {}^{\mathrm{XX}}Q(f_{eature},t_{oken})\log[P_{\mathrm{rob}}\,(t_{oken}|f_{eature})] \quad \dots (5.10)$$

$Q(f_{eature},t_{oken})$ is used to denote the total number of times token t_{oken} came in feature f_{eature}. For the calculation of optimal parameters, EM algorithm is used.

E-step:

$$P_{rob}(v|f_{eature},t_{oken}) = \frac{P_{rob}(v)\;P_{rob}(f_{eature}|v)\;P_{rob}(t_{oken}|v)}{P_{rob}(v1)\;P_{rob}(f_{eature}|v_1)\;P_{rob}(t_{oken}|v_1) + P_{rob}(v_2)\;P_{rob}(f_{eature}|v_2)\;P_{rob}(t_{oken}|v_2)} \quad \dots (5.11)$$

M-step:

$$P_{rob}(t_{oken}|v) = \frac{\sum_{f\,eature} Q(f_{eature},t_{oken})\;P_{rob}(v|f_{eature},t_{oken})}{\sum_{feature,t'oken} Q(f_{eature},t'_{oken})\;P_{rob}(v|f_{eature},t'_{oken})} \quad ,$$

$$P_{rob}(f_{eature}|v)=\frac{\sum_{token}Q(f_{eature},t_{oken})\ P_{rob}(v|f_{eature},t_{oken})}{\sum_{feature,t'oken}Q(f'_{eature},t_{oken})\ P_{rob}(v|f'_{eature},t_{oken})}\quad ,$$

$$P_{rob}(v) = \frac{\sum f_{eature,token}Q(f_{eature},t_{oken})\ P_{rob}(v|f_{eature},t_{oken})}{\sum_{feature,,token}Q(f_{eature},t_{oken})} \quad \ldots (5.12)$$

Initialize $P(t_{oken}|v)$ to link v_1 to abstract *relevancy* and v_2 for *irrelevancy*. The probability distribution $P\ (V = v1|f_{eature})$ can be used to weigh the graph.

5.2 COMPARATIVE OPTIMIZATION ALGORITHMS

Pigeon Inspired Optimization Algorithm

A number of optimization issues, including as trajectory planning for aerial robots, three-dimensional robots, an autonomous landing system, and a Proportional–integral–derivative (*PID*) development controller, have been successfully solved using Pigeon Inspired Optimization (PIO) techniques, according to recent study.

Whale Optimization Algorithm

It is challenging to choose a subset of crucial important points since the definition of brightness is also suggested by the choice of an element, an attribute, or a group of variables for the model's creation. The Whale Optimization Algorithm **(**WOA) algorithm is used to determine who is included in this proposed structure. Due to the rapid drop in differential diversity, the majority of them quickly converge in the direction of the optimal neighbourhood, and the first WOA is not a better case, which is a significant problem for large-scale global enhancement (LSGO) handled by metaheuristic computing (MAS). Due to the great productivity of hunting all over the world, the Levy flight course has been actively used in previous research in MA to speed up integration and avoid the tight agreement of Optima. Therefore, Levy flying is employed at Whale Optimization Algorithm (WOA) to evacuate almost optimum, allowing for the distinction of population diversity.

5.3 FLOWCHART OF PROPOSED TECHNIQUE

First we will load the dataset. Than features will be extracted by using FC mean. After feature extraction, to optimize the features, Markov Chain Rule is applied. For the classification purpose, Gradient Boosting Algorithm is applied. For validation, K-cross validation is applied to compute the required parameters like accuracy, Sensitivity, Precision, Specificity and F_Score.

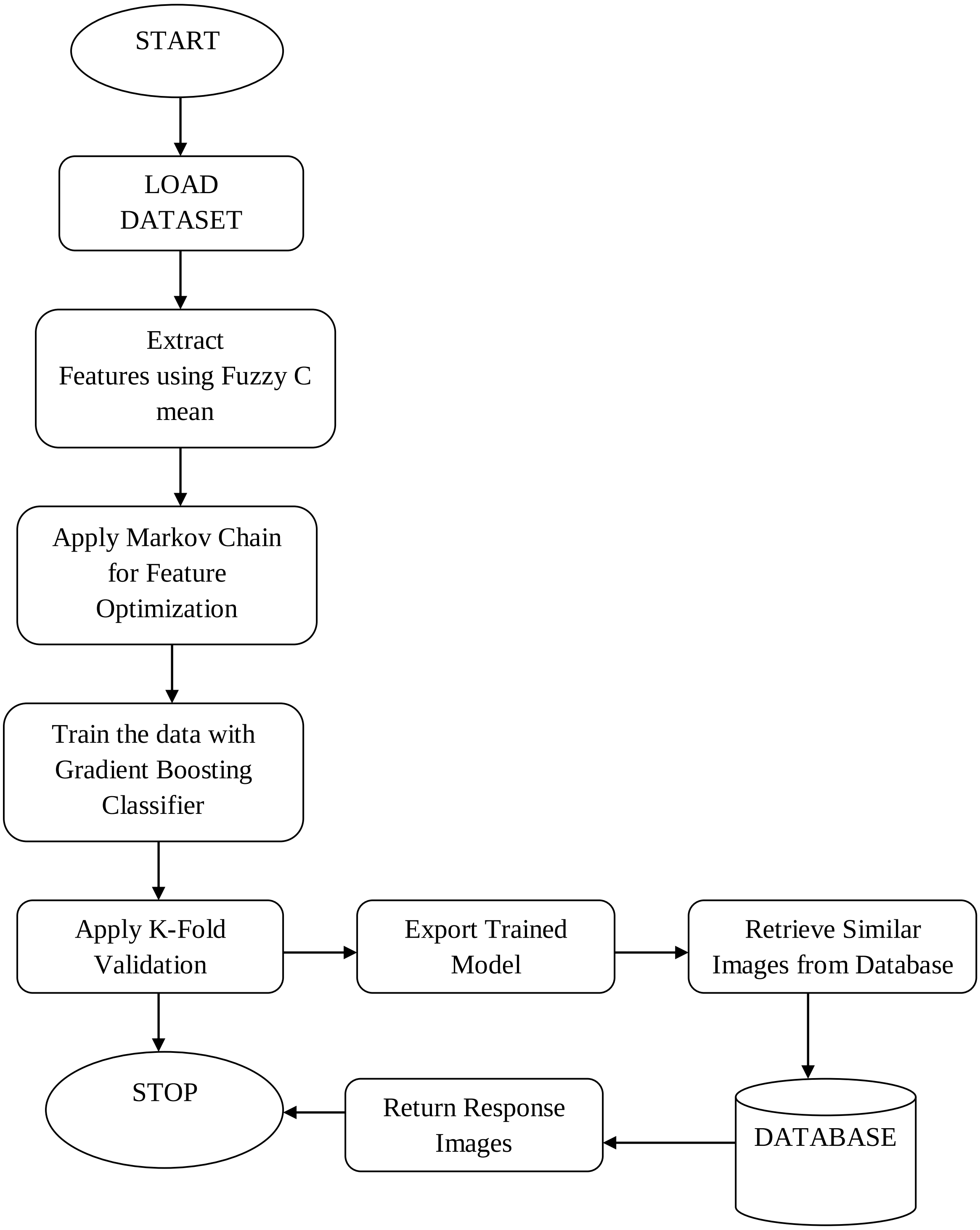


Figure 5.2: Flowchart for working of Feature Extraction and Feature Optimization Technique

5.4 EXPERIMENT AND RESULT ANALYSIS

The motive of this chapter is to optimize the features of images using feature optimization technique to minimize the searching time and to increase the performance.

We have used Fuzzy C mean along with markov chain rule to optimize the features and for classification, Gradient Boosting classifier is used and most matched images will be retrieved from the dataset.

We evaluated the effectiveness of the categorization model using a number of assessment metrics, including:

1. Accuracy: It is the proportion of precise predictions to all predictions.

$$\text{Accuracy} = (\mathrm{TP} + \mathrm{TN})/(\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}) \qquad \ldots (5.13)$$

2. **Sensitivity:** It is the proportion of true positives to all other positives found in the data.

$$\text{Sensitivity} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \qquad \ldots (5.14)$$

3. **Precision:** The ratio is the total predicted positives divided by the sum of true positive and False positive.

$$\text{Precision} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP}) \qquad \ldots (5.15)$$

4. **Specificity**: It is ratio of true negatives to total of true negative and false positive.

$$\text{Specificity} = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}) \qquad \ldots (5.16)$$

5. F_Score: It’s the harmonic mean of the precision and recall.

$$\text{F\_Score} = 2 * (\mathrm{Recall} * \mathrm{Precision})/(\mathrm{Recall} + \mathrm{Precision}) \qquad \ldots (5.17)$$

5.4.1 Experiments

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
GLCM+MC+GT	92.10	92.00	92.90	92.70
KMean+MC+GT	97.00	96.10	96.10	97.30
LBP+MC+GT	93.00	92.81	93.87	93.80
PCA+MC+GT	94.30	94.99	94.78	95.60
SIFT+MC+GT	94.99	94.37	94.10	96.20
MCPL-FS	97.10	97.10	97.00	97.90

Table 5.1 Accuracy of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

The above Table 5.1 shows the performance accuracies of different feature extraction techniques with feature optimization along with GT Classifier and it is clearly shown that we are getting highest accuracy with our proposed Technique i.e. MCPL-FS.

Features Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
GLCM+MC+GT	93.10	93.05	92.10	95.94
KMean+MC+GT	97.10	96.50	96.87	98.51
LBP+MC+GT	96.20	95.16	94.20	96.57
PCA+MC+GT	96.50	96.00	96.10	97.57
SIFT+MC+GT	96.50	96.87	96.87	97.89
MCPL-FS	98.00	98.10	98.12	98.85

Table 5.2 Sensitivity of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

The above Table 5.2 shows the performance sensitivities of different feature extraction techniques with feature optimization along with GT Classifier and it is clearly shown that we are getting highest accuracy with our proposed Technique i.e. MCPL-FS.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
GLCM+MC+GT	92.80	93.00	91.00	92.29
KMean+MC+GT	96.10	95.10	95.10	97.07
LBP+MC+GT	93.00	92.87	93.87	93.40
PCA+MC+GT	94.10	93.10	94.78	95.29
SIFT+MC+GT	94.89	94.54	94.10	95.89
MCPL-FS	97.10	97.00	96.99	97.71

Table 5.3 Precision of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

The above Table 5.3 shows the performance precisions of different feature extraction techniques with feature optimization along with GT Classifier and it is clearly shown that we are getting highest accuracy with our proposed Technique i.e. MCPL-FS.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
GLCM+MC+GT	95.15	95.20	92.05	95.94
KMean+MC+GT	97.12	97.21	97.58	98.50
LBP+MC+GT	95.87	94.89	95.16	96.56
PCA+MC+GT	97.10	95.35	96.82	97.65
SIFT+MC+GT	96.89	96.32	96.50	97.89
MCPL-FS	98.11	98.25	97.57	98.83

Table 5.4 Specificity of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

The above Table 5.4 shows the performance specificities of different feature extraction techniques with feature optimization along with GT Classifier and it is clearly shown that we are getting highest accuracy with our proposed Technique i.e. MCPL-FS.

Features Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
GLCM+MC+GT	93.80	93.10	93.99	93.90
KMean+MC+GT	96.12	96.58	96.50	97.10
LBP+MC+GT	94.17	93.00	93.17	94.98
PCA+MC+GT	95.10	95.12	95.37	96.10
SIFT+MC+GT	95.54	94.87	94.88	96.05
MCPL-FS	97.20	98.11	97.30	98.27

Table 5.5 F_Score of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

The above Table 5.5 shows the performance F_Scores of different feature extraction techniques with feature optimization along with GT Classifier and it is clearly shown that we are getting highest accuracy with our proposed Technique i.e. MCPL-FS.

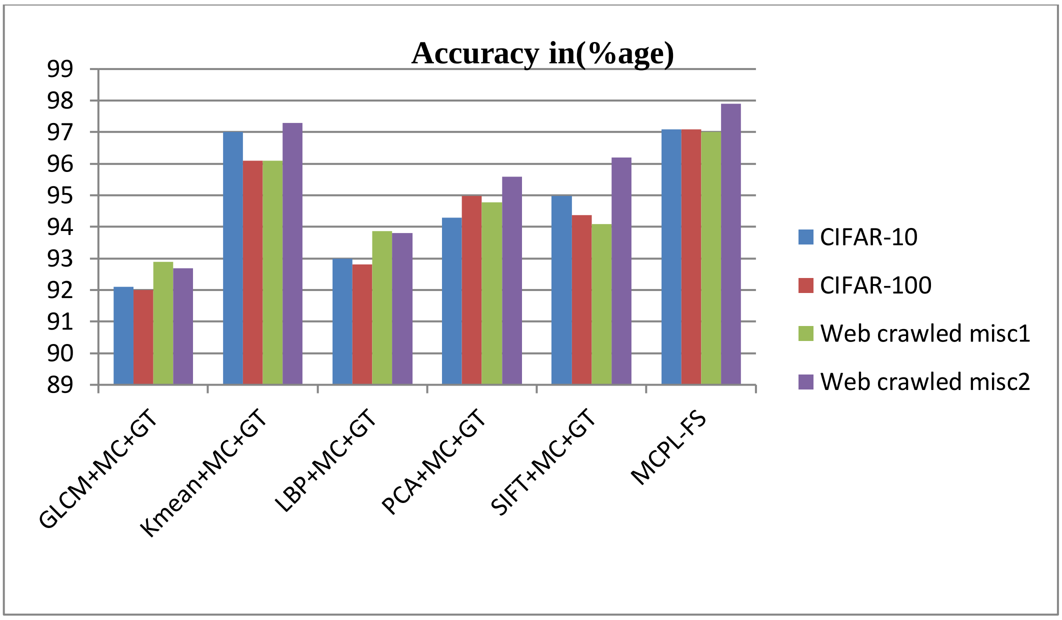


Figure 5.3: Accuracy of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

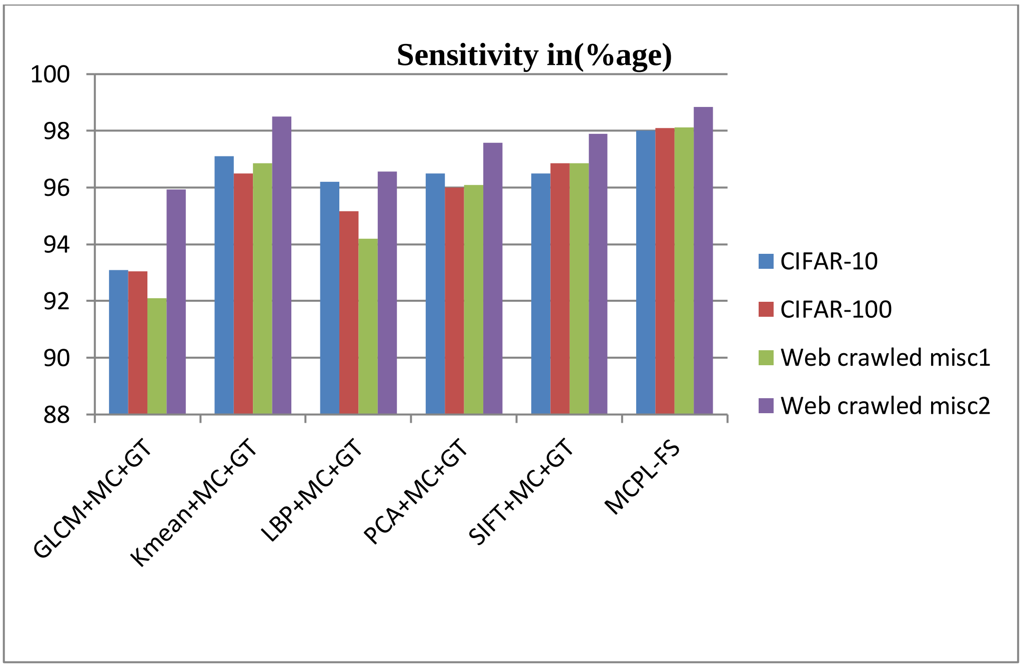


Figure 5.4: Sensitivity of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

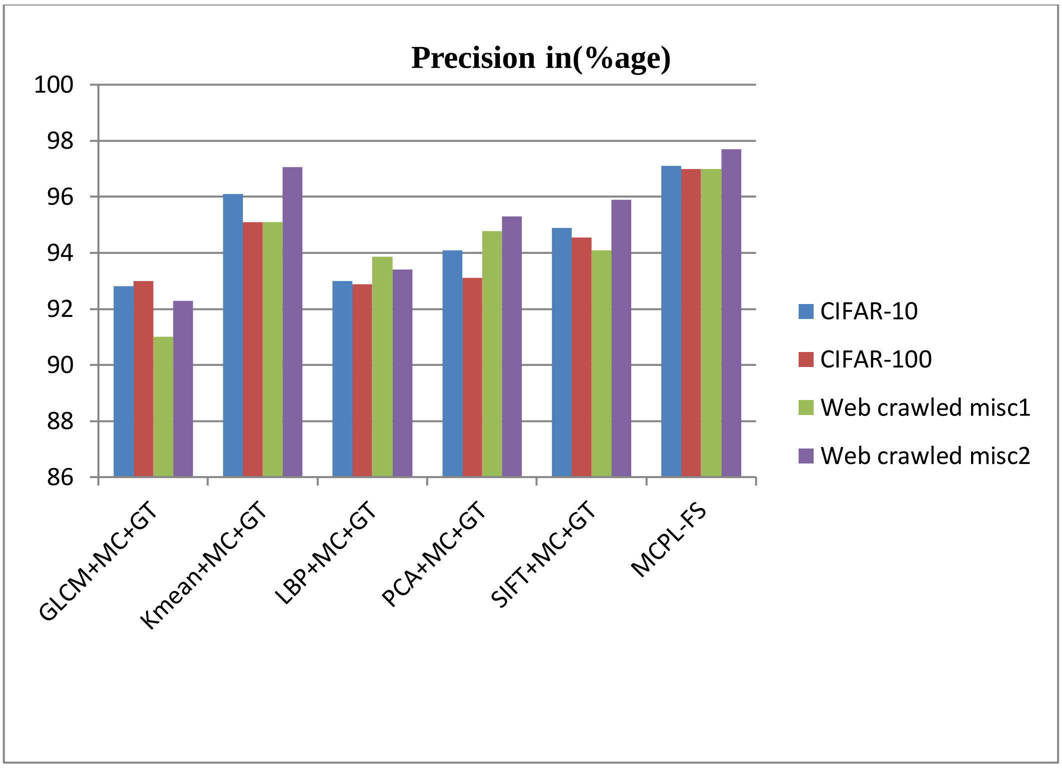


Figure 5.5: Precision of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

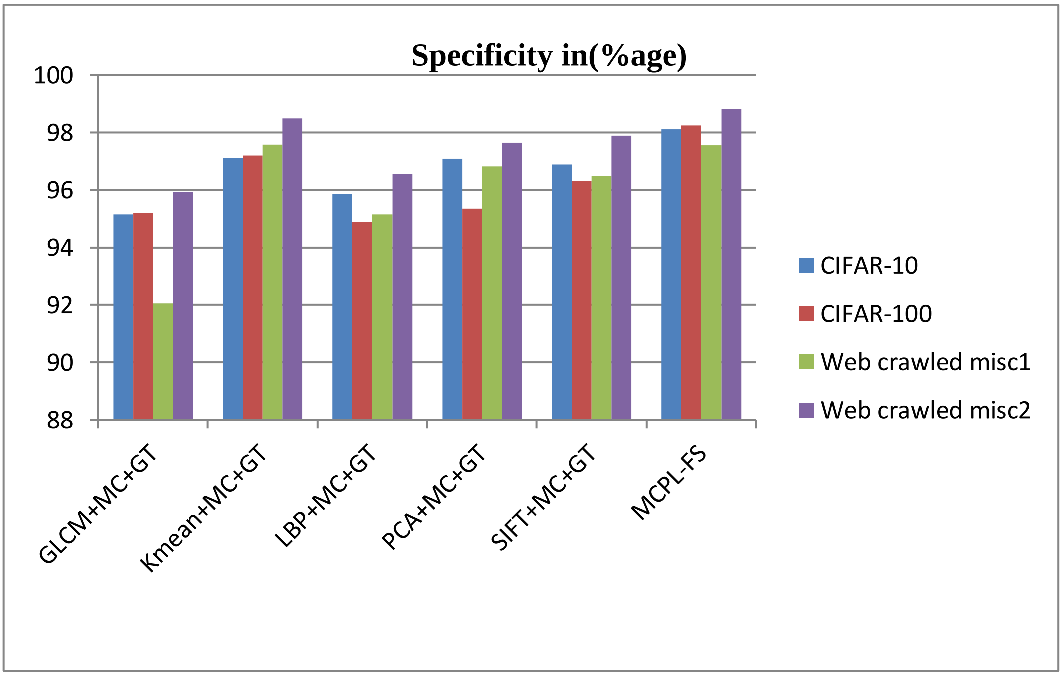


Figure 5.6: Specificity of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

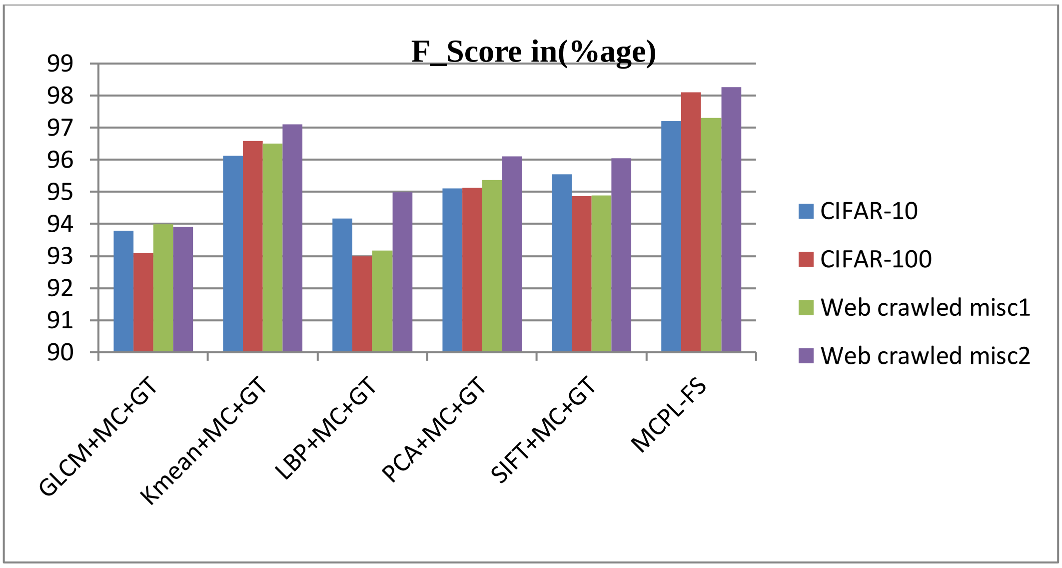


Figure 5.7: F_Score of different Feature Extraction Techniques with Feature Optimization along with GT Classifier

From above given Figure 5.3-Figure 5.7, it is clearly shown that we are getting highest performance with the help of MCPL-FS as compare to other Techniques.

Feature Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
Fuzzy CMean+GT	96.10	96.20	96.90	97.44
MCPL-FS	97.10	97.10	97.00	97.90

Table 5.6 Accuracy of FC mean and MC Optimization Technique

The above Table 5.6 shows the performance accuracy of feature extraction and optimization approach and it is clearly shown in Table that we are getting highest accuracy with the help of MCPL-FS as compare to other Techniques.

Feature Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
Fuzzy CMean+GT	97.15	98.00	97.85	98.58
MCPL-FS	98.00	98.10	98.12	98.85

Table 5.7 Sensitivity of FC mean and MC Optimization Technique

The above Table 5.7 shows the performance Sensitivity of feature extraction and optimization approach and it is clearly shown in Table that we are getting highest sensitivity with the help of MCPL-FS as compare to other Techniques.

Feature Extraction Methods	**CIFAR-10**	**CIFAR-100**	**WEB CRAWLED MISC1**	**WEB CRAWLED MISC2**
Fuzzy CMean+GT	96.20	96.50	96.89	97.21
MCPL-FS	97.10	97.00	96.99	97.71

Table 5.8 Precision of FC mean and MC Optimization Technique

The above Table 5.8 shows the performance Precision of feature extraction and optimization approach and it is clearly shown in Table that we are getting highest precision with the help of MCPL-FS as compare to other Techniques.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
Fuzzy CMean+GT	97.54	98.21	97.51	98.58
MCPL-FS	98.11	98.25	97.57	98.83

Table 5.9 Specificity of FC mean and MC Optimization Technique

The above Table 5.9 shows the performance Specificity of feature extraction and optimization approach and it is clearly shown in Table that we are getting highest specificity with the help of MCPL-FS as compare to other Techniques.

Feature Extraction Methods	CIFAR-10	CIFAR-100	WEB CRAWLED MISC1	WEB CRAWLED MISC2
Fuzzy CMean+GT	96.98	97.50	97.20	97.89
MCPL-FS	97.20	98.11	97.30	98.27

Table 5.10 F_Score of FC mean and MC Optimization Technique

The above Table 5.10 shows the performance F_Score of feature extraction and optimization approach and it is clearly shown in Table that we are getting highest F_Score with the help of MCPL-FS as compare to other Techniques.

In Table 5.6-5.10, after extracting the features and optimizing the features and then by classifying the features using Gradient Boosting Algorithm, different performance parameters have been calculated. It has been observed here that with Fuzzy C mean feature clustering technique and markov chain rule feature optimization(MCPL-FS), accuracy achieved for CIFAR-10 is 97.10%, for CIFAR-100, it is 97.10%, for Web-Crawled misc1, it is 97.00% and for Web-Crawled misc2 it is 97.90%. Sensitivity achieved for CIFAR-10 is 98.00%, for CIFAR-100, it is 98.10%, for Web-Crawled misc1, it is 98.12% and for Web-Crawled misc2 it is 98.85%. Precision achieved for CIFAR-10 is 97.10%, for CIFAR-100, it is 97.00%, for Web-Crawled misc1, it is 96.99% and for Web-Crawled misc2 it is 97.71%. Specificity achieved for CIFAR-10 is 98.11%, for CIFAR-100, it is 98.25%, for Web-Crawled misc1, it is 97.57% and for Web-Crawled misc2 it is 98.83%. F_Score achieved for CIFAR-10 is 97.20%, for CIFAR-100, it is 97.30%, for Web-Crawled misc1, it is 98.27% and for Web-Crawled misc2 it is 97.89%.

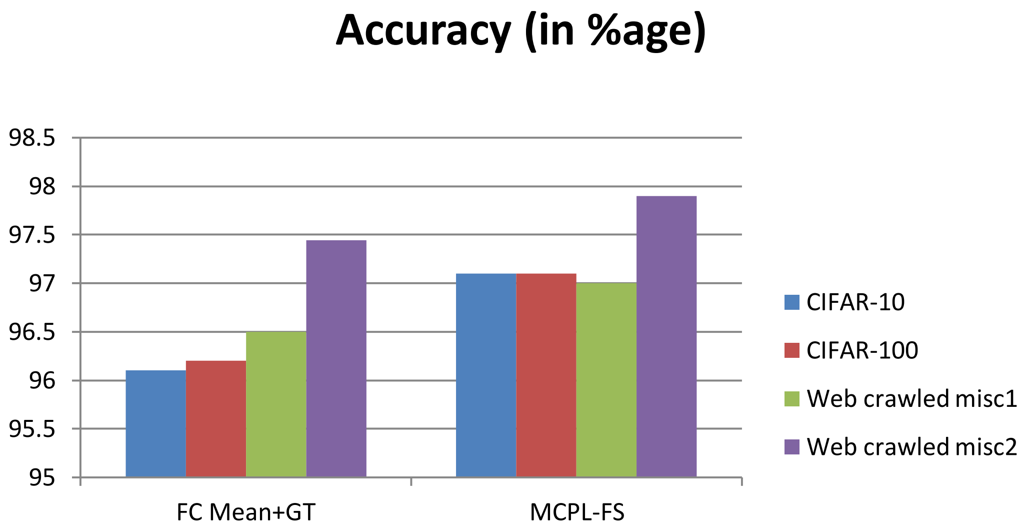


Figure 5.8: Accuracy comparison along with Feature Optimization Technique

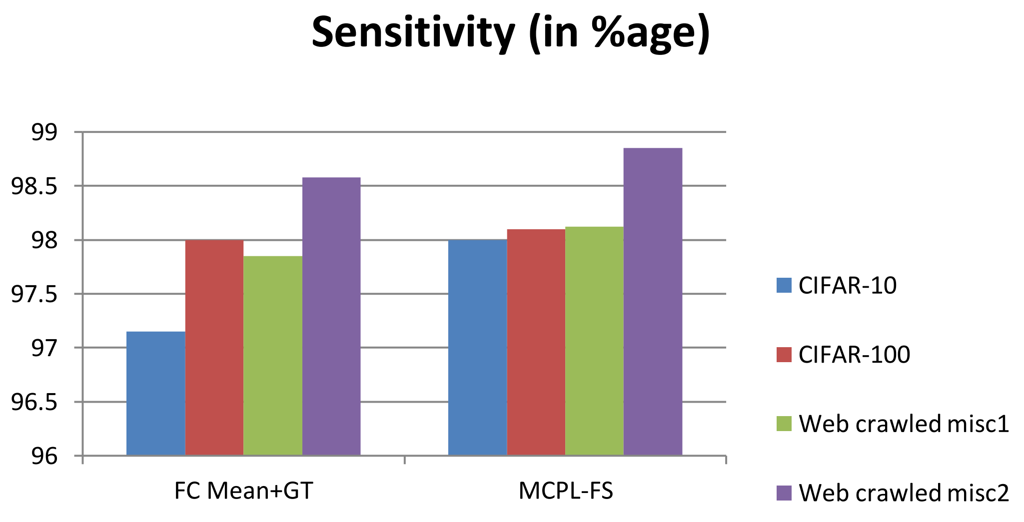


Figure 5.9: Sensitivity comparison along with Feature Optimization Technique

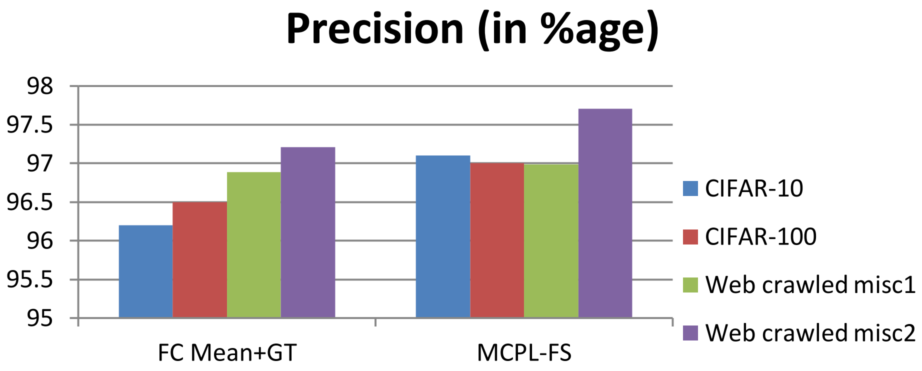


Figure 5.10: Precision comparison along with Feature Optimization Technique

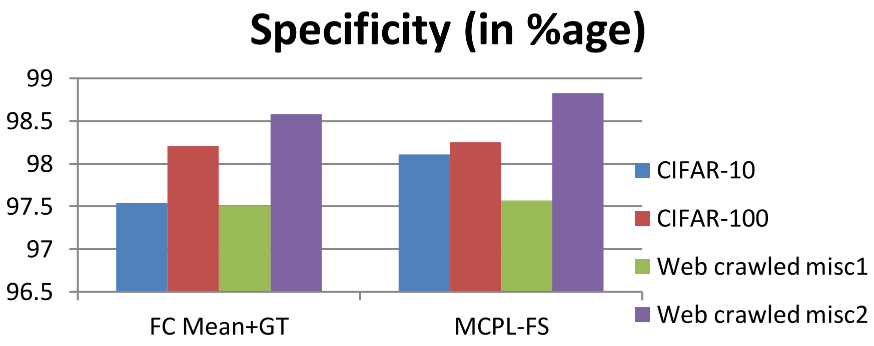


Figure 5.11: Specificity in Percentage for Feature Optimization Technique

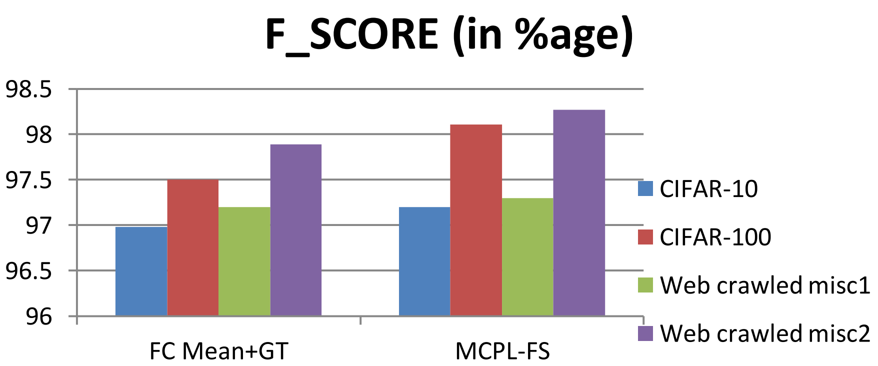


Figure 5.12: F_Score in Percentage for Feature Optimization Technique

In Figure 5.8-5.12, after extracting the features and optimizing the features and then by classifying the features using Gradient Boosting Algorithm, different performance parameters have been calculated. It has been observed here that with Fuzzy C mean feature clustering technique and markov chain rule feature optimization, accuracy achieved for CIFAR-10 is 97.10%, for CIFAR-100, it is 97.10%, for Web-Crawled misc1, it is 97.00% and for Web-Crawled misc2 it is 97.90%. Sensitivity achieved for CIFAR-10 is 98.00%, for CIFAR-100, it is 98.10%, for Web-Crawled misc1, it is 98.12% and for Web-Crawled misc2 it is 98.85%. Precision achieved for CIFAR-10 is 97.10%, for CIFAR-100, it is 97.00%, for Web-Crawled misc1, it is 96.99% and for Web-Crawled misc2 it is 97.71%. Specificity achieved for CIFAR-10 is 98.11%, for CIFAR-100, it is 98.25%, for Web-Crawled misc1, it is 97.57% and for Web-Crawled misc2 it is 98.83%. F_Score achieved for CIFAR-10 is 97.20%, for CIFAR-100, it is 97.30%, for Web-Crawled misc1, it is 98.27% and for Web-Crawled misc2 it is 97.89%. From above given results, it is clearly shown that we are getting highest performance with the help of FC mean feature extraction algorithm along with optimization (MCPL-FS) as compare to without optimization technique.

Comparison with others Work

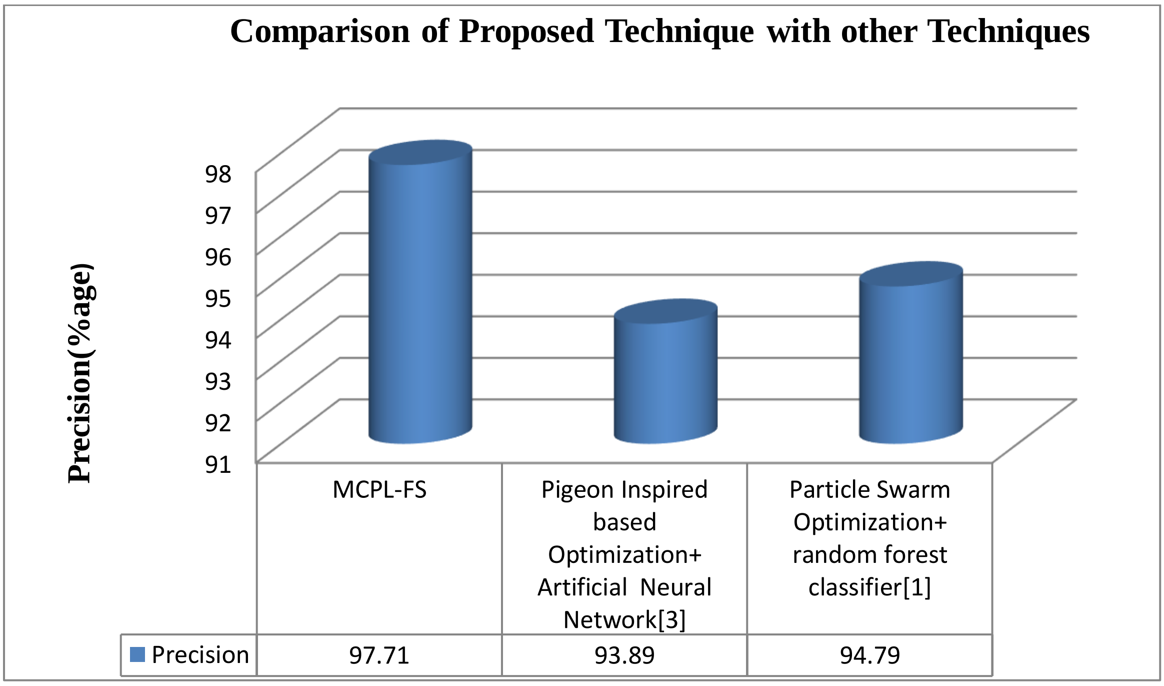


Figure 5.13: Precision comparison of proposed Technique with other Technique

In Figure 5.13, we have compared our Technique (MCPL-FS) with other's proposed Techniques and on the basis of precision, we are getting better results.

5.4.2 Result Analysis

We observe that by use of Fuzzy C mean and Markov chain rule, we are getting better results as compare to without feature optimization Technique. Apart from this, we are also getting better results than other optimization Techniques.

Markov Chain algorithms have the ability to explore complex and high-dimensional search spaces effectively. They can handle objective functions with multiple local optima or even discontinuous surfaces, where traditional optimization algorithms might get stuck in a suboptimal solution. It often considered global optimization techniques. By performing random walks in the search space, they can sample from the entire space, allowing them to converge to global optima with a suitable exploration strategy.

5.5 SUMMARY

To summarize, we have done comparison of different Feature Optimization methods. Along with this, we have used feature extraction techniques with the Markov Chain Rule and Gradient Boosting Classifier. We have observed that Feature Optimization Technique has improved the results greatly in terms of accuracy, sensitivity, precision, specificity and F_Score, the best results were given by the combination of Fuzzy C mean and Markov Chain Rule along with Gradient Boosting Algorithm (MCPL-FS).

CHAPTER 6

PERFORMANCE OF MATCHING ALGORITHMS

In Chapter 5, once we loaded the dataset, features were extracted using FC mean. After feature extraction, Markov Chain Rule was applied to optimize these features. For the classification purpose, Gradient Boosting Algorithm was applied. While for validation, K-cross validation is used to compute performance parameters like accuracy, sensitivity, precision, specificity and F_Score. Apart from this, in Chapter 6, different similarity matching techniques like Wasserstein Distance, Indexing Distance, Discrete Frechet Distance and Bhattacharyya Distance have been applied to get the retrieval time for comparison purpose.

6.1 DIFFERENT SIMILARITY MEASURE ALGORITHMS

6.1.1 Hausdorff Distance Matching

To compare the similarities between query image and the images, which are already stored in dataset, Hausdarff similarity measure method is used.

By assessing the percentage of points in one set that are close to points in the other, this measurement assesses how similar two point sets are to one another. In order to determine if two point sets resemble one another, it is necessary to examine two parameters: the minimum distance at which points must be regarded near to one another, and the percentage of points that are (at most) this close to points in the other set. There are no paired points in the two sets being compared, which distinguishes this distance metric from correspondence-based approaches like point matching and binary correlation.

Pseudocode of Hausdorff Distance Matching

To get the Hausdorff Distance between P and Q, use the formula hd = HausdorffDist. (P,Q)

Hausdorff = HausdorffDist (P,Q)

Hausdorff = HausdorffDist (...,lmf)

HausdorffDist(...,[],'visualize'] is equivalent to [hd D].

Calculates the Hausdorff distance, hd, between two sets of points, P and Q. (which could be two trajectories). There must be an equal number of columns (dimensions) in the matrices of sets P and Q, but not necessarily an equal number of rows (observations).

The Directional Hausdorff Distance (dhd) is defined as: dhd(P,Q) = max p c P [min q c Q [||p-q||]].

Dhd identifies the point p from the set P that is the farthest from any other point in Q and finds the distance between p and its nearest neighbour in Q.

The Hausdorff Distance is Max(dhd(P,Q),dhd(Q,P)).

The matrix D represents the separation between the nth point in P and the mth point in Q. (n,m).

If P and Q are big enough, the matrix of distances between them, D, will be too big to store in memory. If creating the D matrix will need more memory than we have allotted, the function will verify and use a quicker version of the code instead. In this scenario, the raw matrix D will be returned. The method may be forced to disregard the D matrix even for minuscule P and Q if the function is run with the optional third lmf variable set to 1. The function will be forced to return the D matrix if lmf is set to 0. The code could choose the appropriate mode on its own when lmf is set to [).

6.1.2 Wasserstein Distance

It is used to measure the distance between two probability distributions. It considers both probabilities and also distance between different outcome events.

Pseudocode of Wasserstein Distance

The formula wsd = WS DISTANCE yields the 1-Wasserstein distance between the discrete probability measures u and v that correspond to the sample vectors u samples and v samples (u samples, v samples).

The formula wsd = WS DISTANCE yields the p-Wasserstein distance between the discrete probability measures u and v that correspond to the sample vectors u samples and v samples (u samples, v samples, p). P can only be either 1 or 2.

6.1.3 Indexing Distance

It is an Euclidean Distance, which gives the distance between two points.

Pseudocode of Indexing Distance

It's basically a Euclidean distance. It offers the separation or straight line distance between two points. Let's assume that there are two points on a two-dimensional plane, (x1, y1)(x1, y1) and (x2, y2)(x2, y2). The formula for Euclidean distance is displayed here.

$$d = \sqrt{[\,(x_2 - x_1)^2 + (y_2 - y_1)^2\,]} \qquad \dots (6.1)$$

6.1.4 Frechet Distance

To calculate the similarity between two curves, A and B, Frechet Distance is used. It is used to define the minimum cord-length to join a point travelling forward along A and a point travelling forward along B.Although, rate of travel may not be uniform for either points.

Pseudocode of Frechet Distance

Two sets of points, P and Q, combine to form polygonal curves with dimensions arranged in columns and vertices arranged in rows (data points). The points of the curve should be in the same order as they do in P and Q.

The coupling measure, commonly known as the discrete Frechet distance, is given back in cm. When P = Q, it is 0, and as the curves diverge farther, it rises positively.

The user can provide a function to be used to determine the separation between points in P and Q using the optional dfcn parameter. The L2 norm is applied if none are given.

The coupling sequence, or collection of steps along each curve that must be taken to achieve the least coupling distance, cm, is the secondary output, or cSq. The outcome is shown as a matrix, where column 1 has the index for each point in P and column 2 contains the index for each point in Q.

6.1.5 Bhattacharyya Distance

The Bhattacharyya distance calculates how similar two probability distributions are to one another. It has a close relationship to the Bhattacharyya coefficient, which assesses how much two statistical samples or populations overlap.

Pseudocode of Bhattacharyya Distance

Z = Bhattacharyya The function Distance determines the one-dimensional Bhattacharyya distances between two independent subsets of the data set X that are categorised in line with the logical labels in I. (X,I). The Bhattacharyya distance is a metric for assessing qualities in terms of their ability to discern between two classes of data, such as data from healthy and failing equipment. The data in X have a Gaussian distribution, which is the presumption utilised in the distance calculation.

6.2 FLOWCHART OF PROPOSED DIFFERENT MATCHING ALGORITHMS

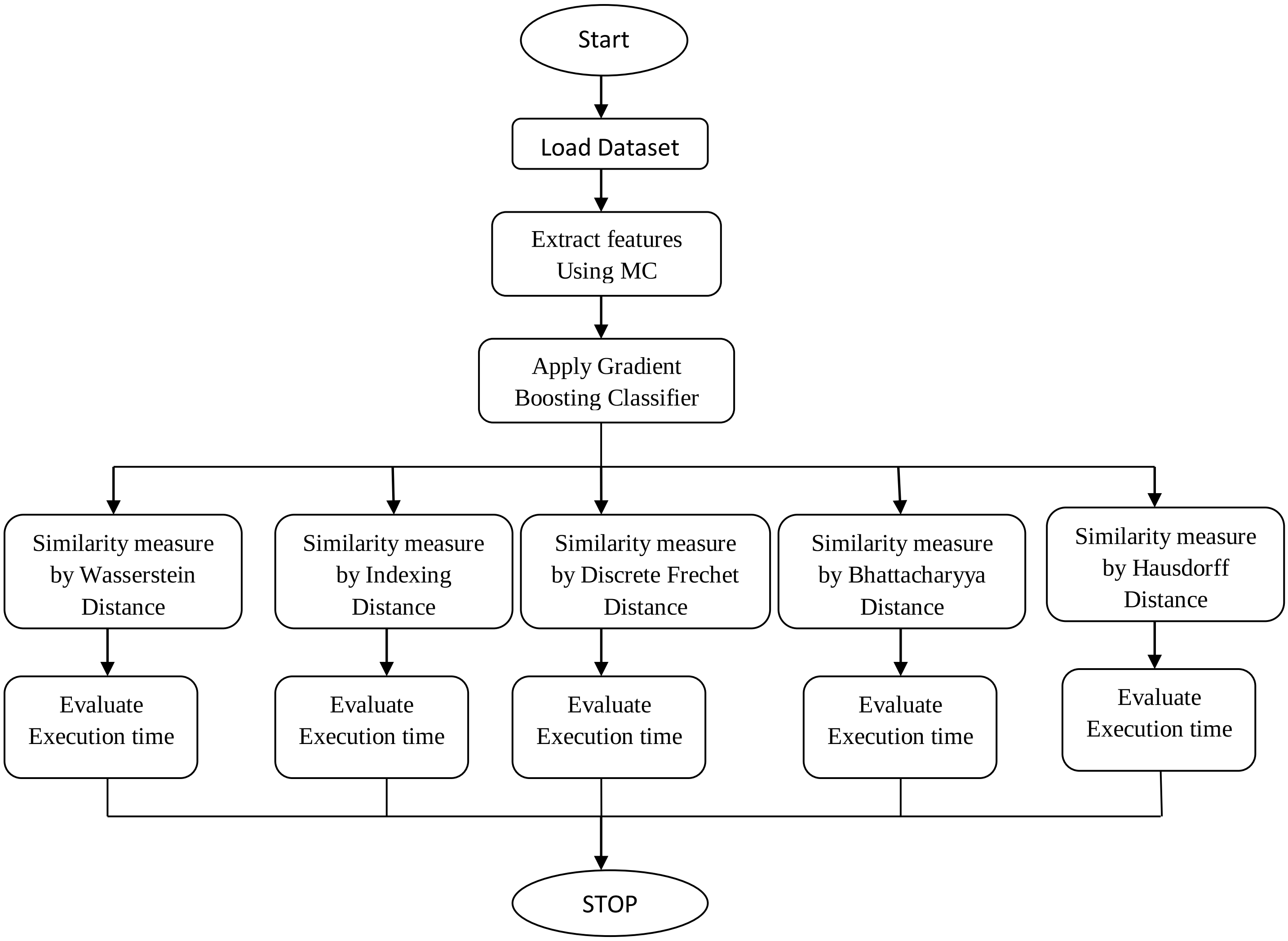


Figure 6.1 Flowchart for the comparison of different Matching Algorithms

In above Figure 6.1, after extracting and optimizing the features, comparison between different similarity measures techniques like Wasserstein distance, indexing distance, discrete frechet distance, Bhattacharyya distance and Hausdorff distance has been computed to get the execution time to retrieve the images from the database on the basis of query image to get the similar images.

6.3 EXPERIMENT AND RESULT ANALYSIS

Different similarity matching techniques like Wasserstein Distance, Indexing Distance, Discrete Frechet Distance and Bhattacharyya Distance have been applied to get the retrieval time for comparison purpose.

6.3.1 Experiments

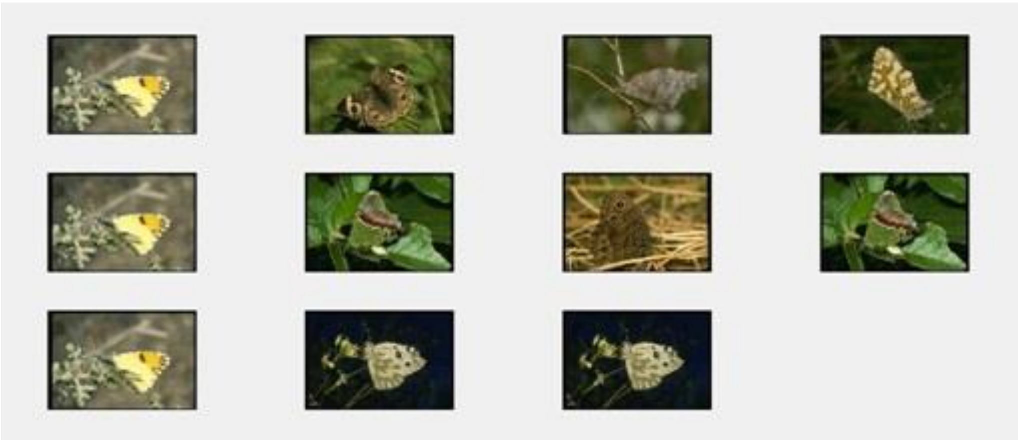


Figure 6.2 Retrieved images along with Fuzzy Cmean, MC and GT classifier by using Hausdorff Distance matching Algorithm

Above are the first 10 matched images from the database has been retrieved after the optimizing features and by similarity matching by using Hausdarff Distance similarity matching.

Similarity Measures	WITHOUT OPTIMIZATION	WITH OPTIMIZATION
Wasserstein Distance	0.8987	0.7178
Indexing Distance	0.6766	0.6582
Discrete Frechet Distance	0.7231	0.6813
Bhattacharyya Distance	0.6896	0.656
Hausdorff Distance	0.6763	0.6302

Table 6.1 Execution Time in seconds along with Fuzzy Cmean, MC and GT classifier by using different similarity measure Algorithm

From above Table 6.1, it is clearly shown that our proposed technique along with Hausdorff distance take less time to retrieve images as compare to other matching Techniques.

Figure 6.3: Execution Time in seconds along with Fuzzy Cmean, MC and GT classifier by using different similarity measure Algorithm

6.3.2 Result Analysis

On the basis of similarity measures, we have found the execution time with and without optimization technique also and we have seen that we are getting less execution time with optimization technique and by using Hausdorff Distance matching.

Hausdorff distance can provide better results compared to other distance metrics like Euclidean distance or Manhattan distance, especially in scenarios where the shape or arrangement of points is crucial. Hausdorff distance focuses on the maximum distance, making it less sensitive to outliers or noise in the data. It accounts for the overall shape similarity rather than being affected by individual data points.

6.4 SUMMARY

After extracting the features using Fuzzy C Mean and then optimized the features by using Markov Chain rule and Classifying the Features using Gradient Boosting Classifier and by applying the different similarity matching algorithms, we have founded that our proposed method (MCPL-FS) is best and it takes less time for the execution as compare to other similarity matching algorithms.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

This work attempted to process content based image retrieval. This searches the related images from the databases on the basis of query image. This research also tried to develop an efficient automated system for Content Based Image Retrieval without an expert‘s help or any manual intervention. This work focuses more on extraction of features and optimization of features, which helps to increase the accuracy of image retrieval and reduces the retrieval time. The results were comparatively satisfactory. The results obtained were given below.

7.1 SUMMARY OF RESULTS AND CONCLUSION

Brief summary of chapters is given here, discussing the proposed approaches and the results. In Chapter 3, we were able to extract the features using various feature extraction and clustering Techniques. From Chapter 4, we were able to investigate the best feature extraction Technique and Gradient Boosting Classifier based on the accuracy, sensitivity, precision, specificity and F_Score. It has been observed here that with Fuzzy C mean feature clustering technique, accuracy achieved is 96.10% using CIFAR-10, 96.20%, using CIFAR-100, 96.50% using Web-Crawled misc1, 97.44% using Web-Crawled misc2 datasets. Sensitivity achieved for CIFAR-10 is 97.15%, for CIFAR-100, sensitivity is 98.00%, for Web-Crawled misc1, it is 97.85% and for Web-Crawled misc2, it is 98.58%. Precision achieved for CIFAR-10 is 96.20%, for CIFAR-100, it is 96.50%, for Web-Crawled misc1, it is 96.89% and for Web-Crawled misc2 it is 97.21%. Specificity achieved for CIFAR-10 is 97.54%, for CIFAR-100, it is 98.21%, for Web-Crawled misc1, it is 97.51% and for Web-Crawled misc2 it is 98.58%. F_Score achieved for CIFAR-10 is 96.98%, for CIFAR-100, it is 97.50%, for Web-Crawled misc1, it is 97.20% and for Web-Crawled misc2 it is 97.89%.

In Chapter 5, with selected feature extraction technique in chapter 4, a markov chain rule was applied to optimize the features and to improve the performance. It has been observed here that with Fuzzy C mean feature clustering technique and markov chain rule feature optimization, accuracy achieved for CIFAR-10 is 97.10%, for CIFAR-100, it is 97.10%, for Web-Crawled misc1, it is 97.00% and for Web-Crawled misc2 it is 97.90%. Sensitivity achieved for CIFAR-10 is 98.00%, for CIFAR-100, it is 98.10%, for Web-Crawled misc1, it is 98.12% and for Web-Crawled misc2 it is 98.85%. Precision achieved for CIFAR-10 is 97.10%, for CIFAR-100, it is 97.00%, for Web-Crawled misc1, it is 96.99% and for Web-Crawled misc2 it is 97.71%. Specificity achieved for CIFAR-10 is 98.11%, for CIFAR-100, it is 98.25%, for Web-Crawled misc1, it is 97.57% and for Web-Crawled misc2 it is 98.83%. F_Score achieved for CIFAR-10 is 97.20%, for CIFAR-100, it is 97.30%, for Web-Crawled misc1, it is 98.27% and for Web-Crawled misc2 it is 97.89%. From above given results, it is clearly shown that we are getting highest performance with the help of FC mean feature extraction algorithm along with optimization as compare to without optimization technique.

We have extended our work further in Chapter 6. The proposed approach (MCPL-FS) was applied with different similarity measure algorithms. We have also given the execution time with and without using optimization techniques. We can understand that we are getting less execution time with optimization technique and Hausdorff Distance matching.

7.2 FUTURE SCOPE

- This work can be extended with different optimization techniques for the improvement.
- Matching algorithms and indexing algorithms can be the scope of further research.
- Quick retrieval is always a challenge in the CBIR system as we have different types of images and the complex images.
- Time and Space Complexity can also be analyzed for all the optimization and matching algorithms to speed up the CBIR process.
- Big data concepts may be added in the CBIR research.

LIST OF PUBLICATIONS

- Ramandeep Kaur, V. Devendran, “Content based image retrieval: A Review”, International Journal of Innovative Technology and Exploring Engineering, vol: 9(10).P.NO. 222-228. (Scopus Indexed 2019) **(UGC)**, 2020.
- Ramandeep Kaur, V. Devendran, “Study on Labeling of images using CNN features”, International Conference on Computer Science, Machine Learning and Artificial Intelligence ,P.NO.19-25,2021. **(International Conference)**
- Ramandeep Kaur, V. Devendran, “Study on Labeling of images using CNN features”, International Journal of Advance Computational Engineering and Networking, pp. 1-7, Vol:9(4), 2021. **(Non Scopus)**
- Ramandeep Kaur, V. Devendran, “Image Matching Techniques: A Review”, Information and Communication Technology for Competitive Strategies, Vol.401, pp. 785-795, December 2021. **(Springer)**
- Ramandeep Kaur, V. Devendran, “Markov Chain Latent Space Probabilistic For Feature Optimization Along With Hausdarff Distance Similarity Matching In Content-Based Image Retrieval”, The Imaging Science Journal, February 2023. **(SCIE)**
- Ramandeep Kaur, V. Devendran, “Fuzzy Markov chain latent space probabilistic decision-making for feature optimization in content-based image recognition”, Soft Computing, July 2023. **(SCIE)**

BIBLIOGRAPHY

[1] Manoharan Subramanian, Velmurugan Lingamuthu, Chandran Venkatesan, Sasikumar Perumal, "Content Based Image Retrieval using Colour,Gray,Advanced Texture,Shape Features and Random forest classifier with optimized Particle Swarm Optimization" ,International Journal of Biomedical Imaging,2022.(DOI:10.1007/978-981-15-7394-1_2)

[2] D. Mohapatra, J. Tripathy, T. K. Patra, "Rice Disease Detection and Monitoring Using CNN and Naive Bayes Classification", Advances in Intelligent Systems and Computing, vol. 1248, 2021. (DOI:10.1007/978-981-15-7394-1_2)

[3] M. Buvana, K. Muthumayil, T. Jayasankar, "Content-Based Image Retrieval based on Hybrid Feature Extraction and Feature Selection Technique Pigeon Inspired based Optimization", Annals of the Romanian Society for Cell Biology, vol. 25(1), 2021.

[4] Mishra, M., Choudhury, T., Sarkar, T., "CNN based efficient image classification system for smartphone device", Electronic letters on computer vision and image analysis 0(0), 2021. (DOI: https://doi.org/10.21203/rs.3.rs-428430/v1)

[5] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions", Journal of Big Data 8, 2021. (DOI: https://doi.org/10.1186/s40537-021-00444-8)

[6] Kugunavar, S. and Prabhakar, C. J., "Convolutional neural networks for the diagnosis and prognosis of the coronavirus disease pandemic",Vis Comput Ind Biomed Art; 4(1):12,2021. (DOI:10.1186/s42492-021-00078-w0)

[7] Abbas, A., Abdelsamea, M.M., Gaber, M.M.,"Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network", Appl. Intell., 2021.

[8] TianboWu, ChenTang, MinXu, NianHong, ZhenkunLei, "ULNet for the detection of coronavirus (COVID-19) from chest X-ray images",Computers in Biology and Medicine,vol. 137,2021.(DOI: 10.1016/j.compbiomed.2021.104834,)

[9] Ibrahim, A.U., Ozsoz, M., Serte, S. et al., "Pneumonia classification using deep learning from chest X-ray images during COVID-19", Cogn Comput., 2021. (DOI:https://doi.org/10.1007/s12559-020-09787-5)

[10] Chaddad, A., Hassan, L., Desrosiers, C.,"Deep CNN models for predicting COVID-19 in CT and X-ray images", J. Med. Imag, 8(S1)014502,2021.(DOI:https://doi.org/10.1117/1.JMI.8S1.014502.)

[11] Li, J.S., Liu, I.H., Tsai, C.J., Su, Z.Y., Li, C.F. and Liu, "C.G. Secure content-based image retrieval in the cloud with key confidentiality", IEEE Access, 8, 2020.

[12] Anan Banharnsakun, "Artificial bee colony algorithm for content based image retrieval", Computational Intelligence, vol. 36(1), 2020.

[13] Dhillon, A., Verma, G.K, "Convolutional neural network: a review of models, methodologies and applications to object detection", Prog Artif Intell 9, 2020. (DOI: https://doi.org/10.1007/s13748-019-00203-0)

[14] Vinh, T. Q., Duy, L. H., and Nhan, N. T., "Vietnamese handwritten character recognition using convolutional neural network", IAES International Journal of Artificial Intelligence vol. 9,2020. (DOI: 10.11591/ijai.v9.i2.)

[15] Andrea Yoss, "Transfer Learning using Pre-Trained AlexNet Model and Fashion-MNIST", Retrieved on Sept. 16, 2021 from https://towardsdatascience.com/transfer-learning-using-pre-trained-alexnet-model-and-fashion-mnist-43898c2966fb, 2020.

[16] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition", Cognitive Systems Research., vol. 53, 2019.(DOI: 10.1016/j.cogsys.2018.04.006)

[17] Jyoti S.Patil, "SIFT: A Comprehensive", International Journal of Recent Technology and Engineering, vol.7 (5S4), 2019.

[18] Rokas Balsys. Convolutional Neural Networks (CNN) explained. Retrieved on Nov. 24, 2021 from https://pylessons.com/CNN-tutorial-introduction May, 2019.

[19] R. Rani Saritha ,Varghese Paul · P. Ganesh Kumar, "Content based image retrieval using deep learning process",Cluster Computing,2018.(DOI: https://doi.org/10.1007/s10586-018-1731-0)

[20] S. Zhang, H. Wang, W. Huang, and Z. You, "Plant diseased leaf segmentation", 2018. (DOI: 10.1016/j.ijleo.2017.11.190)

[21] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network",

Computers and Electronics in Agriculture, vol. 154, 2018.(DOI: 10.1016/j.compag.2018.08.048)

[22] S. Kaur, S. Pandey, and S. Goel, “Semi-automatic leaf disease detection and classification system for soybean culture”, IET Image Processing, vol. 12, 2018.(DOI: 10.1049/iet-ipr.2017.0822)

[23] C. S. Hlaing and S. M. Maung Zaw, “Tomato Plant Diseases Classification Using Statistical Texture Feature and Color Feature”, 17th IEEE/ACIS International Conference on Computer and Information Science, 2018.(DOI: 10.1109/ICIS.2018.8466483)

[24] Sumit Saha, “A Comprehensive Guide to Convolutional Neural Networks”, the ELI5 way. Retrieved on Nov. 24, 2021 from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, 2018.

[25] Yamashita, R. et al, “Convolutional neural networks: An overview and application in radiology”, Insights into Imaging, 2018. (DOI:https://doi.org/10.1007/s13244- 018-0639-9)

[26] Towards data science. Convolutional neural network. Retrieved October 20, 2021 from https://towardsdatascience.com/convolutional-neural-network, 2018.

[27] Nitika Seth, Sonika Jindal, “A Review on content based image retrieval”, International Journal of Computers and Technology, vol. 15, 2017.

[28] V. Singh and A. K. Misra, “Detection of plant leaf diseases using image segmentation and soft computing techniques”,Information Processing in Agriculture, vol. 4(1),2017.(DOI: 10.1016/j.inpa.2016.10.005)

[29] J. Ma, K. Du, L. Zhang, F. Zheng, J. Chu, and Z. Sun, “A segmentation method for greenhouse vegetable foliar disease spots images using color information and region growing”, Computers and Electronics in Agriculture, vol. 142, pp. 110–117,2017.(DOI: 10.1016/j.compag.2017.08.023)

[30] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, “Automatic crop detection under field conditions using the HSV colour space and morphological operations”, Computers and Electronics in Agriculture, vol. 133,2017.(DOI: 10.1016/j.compag.2016.11.021)

[31] X. Bai, X. Li, Z. Fu, X. Lv, and L. Zhang, “A fuzzy clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images”,Computers and Electronics in Agriculture ,vol.136,2017.(DOI: 10.1016/j.compag.2017.03.004)

[32] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, “Identification of rice diseases using deep convolutional neural networks”, Neurocomputing, vol. 267, 2017.(DOI: 10.1016/j.neucom.2017.06.023)

[33] Muhammad Sipan, “Image Block Matching Based on GLCM (Gray levelCo-occurence Matrix) Texture Feature on Grayscale Image Auto Coloring”, International Seminar on Intelligent Technology and Its Application, 2017.

[34] M. Suresha, K. N. Shreekanth, and B. V. Thirumalesh, “Recognition of diseases in paddy leaves using knn classifier”, 2nd International Conference for Convergence in Technology, 2017.(DOI: 10.1109/I2CT.2017.8226213)

[35] Krizhevsky A, Sutskever I, Hinton GE., “Imagenet classification with deep convolutional neural networks”, Commun ACM. ,vol.60(6),2017.

[36] Syed Hamad Shirazi ,Arif Iqbal Umar ,SaeedaNaz , Noor ul Amin Khan, Muhammad Imran Razzak ,King Saud bin Abdulaziz ,Bandar AlHaqbani , “Content-Based Image Retrieval Using Texture Color Shape and Region”,International Journal of Advanced Computer Science and Applications, vol. 7, 2016.

[37] Kaur, M., &Sohi, N., “A novel technique for content based image retrieval using color, texture and edge features”,International Conference on Communication and Electronics Systems ,2016.(DOI:10.1109/cesys.2016.7889955)

[38] Islam, Md Amirul, Neil Bruce, and Yang Wang, "Dense image labeling using deep convolutional neural networks." 13th Conference on computer and robot vision IEEE, 2016.

[39] B. VijayaLakshmi and V. Mohan, “Kernel-based PSO and FRVM: An automatic plant leaf type detection using texture, shape, and color features”, Computers and Electronics in Agriculture, vol. 125,2016.(DOI: 10.1016/j.compag.2016.04.033)

[40] S. Prasad, S. K. Peddoju, and D. Ghosh, “Multi-resolution mobile vision system for plant leaf disease diagnosis”, Signal, Image and Video Processing, vol. 10(2), 2016.(DOI: 10.1007/s11760-015-0751-y)

[41] S. Gharge, P. Singh, and Á. D. Á. Á. Neural, “Image Processing for Soybean Disease Classification and Severity Estimation”, Emerging Research in Computing, Information, Communication and Applications,2016.(DOI: 10.1007/978-81-322-2553-9)

[42] X. E. Pantazi, D. Moshou, A. A. Tamouridou and S. Kasderidis, “Leaf Disease Recognition in Vine Plants Based on Local Binary Patterns and One Class Support Vector

Machines",International Federation for Information Processing,2016.(DOI: 10.1007/978-3-319-44944-9_27)

[43] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection", Frontiers in Plant Science, vol. 7, 2016. (DOI: 10.3389/fpls.2016.01419)

[44] Hind Rustum Mohammed, "Improved fuzzy c-mean algorithm for image segmentation", International Journal of Advanced Research in Artificial Intelligence, Vol. 5(6), 2016.

[45] W. Kazmi, F. J. Garcia-Ruiz, J. Nielsen, J. Rasmussen, and H. Jørgen Andersen, "Detecting creeping thistle in sugar beet fields using vegetation indices", Computers and Electronics in Agriculture, vol. 112, 2015.(DOI: 10.1016/j.compag.2015.01.008)

[46] W. Yang, S. Wang, X. Zhao, J. Zhang, and J. Feng, "Greenness identification based on HSV decision tree",Information Processing in Agriculture, vol. 2, 2015.(DOI: 10.1016/j.inpa.2015.07.003)

[47] K. Sumithra, S. Buvana, and R. Somasundaram, "A Survey on Various Types of Image Processing Technique", International Journal of Engineering and Research, vol. V4, 2015. (DOI:10.17577/ijertv4is030552)

[48] G. Roffo, S. Melzi, and M. Cristani., "Infinite feature selection", In Conf., IEEE InternationalConference on Computer Vision, 2015.

[49] P. Manipoonchelvi and K. Muneeswaran, "Multi region based image retrieval system",Indian Academy of Sciences, 2014.

[50] D. Majumdar, A. Ghosh, D. K. Kole, A. Chakraborty and D. D Majumder, "Application of Fuzzy C-Means Clustering Method to Classify Wheat Leaf Images Based on the Presence of Rust Disease", Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications, 2014. (DOI: 10.1007/978-3-319-11933-5)

[51] R. Zhou, S. Kaneko, F. Tanaka, M. Kayamori, and M. Shimizu, "Disease detection of Cercospora Leaf Spot in sugar beet by robust template matching",Computers and Electronics in Agriculture, vol. 108,2014.(DOI: 10.1016/j.compag.2014.07.004)

[52] M. Fakheri, T. Sedghi, M. G. Shayesteh1 and M. C. Amirani, "Framework for image retrieval using machine learning and statistical similarity matching techniques," IET Image Process, 2013.

[53] Y. Chen, X. Li, A. Dick and A. v. d. Hengel, "Boosting Object Retrieval With Group Queries," IEEE, 2012.

[54] R. Arandjelovi´c and A. Zisserman, "Three things everyone should know to improve object retrieval," IEEE, 2012.

[55] Q. Gu, Z. Li, and J. Han.,"Generalized fisher score for feature selection", CoRR, abs/1202.3725, 2012.

[56] Kharabela Parida, " Feature Extraction Using K-means Clustering: An Approach & Implementation", International Journal of Computer Information Systems, vol. 2(2), 2011.

[57] H. J´egou, M. Douze, C. Schmid and P. P´erez, "Aggregating local descriptors into a compact image representation",IEEE, , 2010.

[58] D. Cai, C. Zhang, and X. He., "Unsupervised feature selection for multi-cluster data", In

Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010.

[59] H. Liu and H. Motoda., " Computational Methods of Feature Selection", Chapman and Hall, 2008.

[60]Amit Jain, RamanathanMuthuganapathy, and KarthikRamani, "Content-Based Image Retrieval Using Shape and Depth from an Engineering Database", Springer-Verlag Berlin Heidelberg ,2007.

[61] J. Shotton, J. Winn, C. Rother, A. Criminisi. "TextonBoost:Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation", European Conferenceon Computer Vision, 2006.

[62] Y. Chen, J. Z. Wang and R. Krovetz, "CLUE: Cluster-Based Retrieval of Images by Unsupervised Learning," IEEE, 2005.

[63] R. Fergus, L. Fei-Fei, P. Perona and A. Zisserman, "Learning Object Categories from Google's Image Search", IEEE, 2005.

[64] X. He, D. Cai, and P. Niyogi., "Laplacian score for feature selection",In Advances in Neural Information Processing Systems, 2005.

[65] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. "Gene selection for cancer classification using support vector machines. ",Mach.Learn., 2002.

[66] M. Zaffalon and M. Hutter, "Robust feature selection using distributions of mutual information", In UAI, 2002.

[67] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik., "Gene selection for cancer classification using support vector machines", Mach.Learn.,2002.

[68] T. Hofmann., " Probabilistic latent semantic analysis", In Proceedings of the Fifteenth Conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers ,1999.

[69] Using Convolutional Neural Networks for Image Recognition Retrieved on 1 September 2019, "Convolutional layer‖, https://www.embedded-vision.com".

[70]Implementing a Principal Component Analysis Retrieved on 3 May 2020," https://sebastianraschka.com/Articles/2014_pca_step_by_step.html#principalcomponent-analysis-pca-vs-multiple-discriminant-analysis-mda".

[71] Local Binary Pattern Algorithm: The Math Behind It Retrieved on 25 October 2020,

“https://medium.com/swlh/local-binary-pattern-algorithm-the-math-behind-it%EF%B8%8F-edf7b0e1c8b3".

[72] Gradient Boosting Trees for Classification: A Beginner’s Guide Retrieved on December 2021,"https://medium.com/swlh/gradient-boosting-trees-for-classification-a-beginners-guide-596b594a14ea