Content Based Image Retrieval based on Handcrafted and Deep Learning Methods

Submitted in partial fulfilment of the requirement for the
award of the degree of
DOCTOR OF PHILOSOPHY

Submitted by
Debanjan Pathak
(Roll No. 717149)

Under the guidance of
Dr.U.S.N.Raju
Assistant Professor

Department of Computer Science and Engineering
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
WARANGAL - 506 004, (T.S.) INDIA
June- 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
WARANGAL - 506 004, (T.S.) INDIA

THESIS APPROVAL FOR Ph.D.

This is to certify that the thesis entitled, Content Based Image Retrieval based on Handcrafted and Deep Learning Methods, submitted by Mr. Debanjan Pathak [Roll No.717149] is approved for the degree of DOCTOR OF PHILOSOPHY at National Institute of Technology Warangal.

Examiner

U.S.N.Raju

Research Supervisor
Dr.U.S.N.Raju
Dept. of Computer Science and Engg.
NIT Warangal

Chairman
Dr. S. Ravichandra
Head, Dept. of Computer Science and Engg.
NIT Warangal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
WARANGAL - 506 004, (T.S.) INDIA

CERTIFICATE

This is to certify that the thesis entitled, Content Based Image Retrieval based on Handcrafted and Deep Learning Methods, submitted in partial fulfillment of requirement for the award of degree of DOCTOR OF PHILOSOPHY to National Institute of Technology Warangal, is a bonafide research work done by Mr. Debanjan Pathak [Roll No. 717149] under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

U.S.N.Raju

Research Supervisor
Dr.U.S.N.Raju
Assistant Professor
Dept. of Computer Science and Engg.
NIT Warangal, India

Place: NIT Warangal
Date: 1st July, 2022.

DECLARATION

This is to certify that the work presented in the thesis entitled *‘‘Content Based Image Retrieval based on Handcrafted and Deep Learning Methods''* is a bonafide work done by me under the supervision of Dr.U.S.N.Raju and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others’ ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Debanjan Pathak

Debanjan Pathak

(Roll No. 717149)

Date: 1st July, 2022.

ACKNOWLEDGMENTS

It is with great pleasure that I acknowledge my sincere thanks and deep sense of gratitude to my supervisor Dr.U.S.N.Raju, Department of Computer Science and Engineering for his valuable guidance throughout the course. His technical perception, profound knowledge, sincere effort in guiding a student and devotion to work have greatly charmed and constantly motivated me to work towards the goal. He always gave me ample time for discussions, reviewing my work and suggesting requisite corrections.

I extend my gratitude to all my Doctoral Scrutiny Committee members Dr.Ch.Sudhakar, Dr.K.V.Kadambari, and Dr.G.Kalyan Kumar for their insightful comments and suggestions during my research work. I am immensely thankful to Dr. S.Ravichandra, Head of Dept. of CSE, for providing adequate facilities in the department to carry out the research work. I wish to express my sincere thanks to Prof. N.V. Ramana Rao, Director, NIT Warangal for his official support and encouragement.

I would also like to thank my co-research scholar and friends for their valuable suggestions and for extending selfless cooperation. My gratitude to my family for their unconditional love, support and prayers for my success in achieving the goal. Lastly I am deeply indebted to the almighty for giving the opportunity to pursue this line.

Debanjan Pathak

Dedicated to

My Teachers and Family

ABSTRACT

Research on Content-Based Image Retrieval (CBIR) is being carried out to improvise the existing methods. The main part of CBIR is to extract the features from the given image. The features considered are based on texture, color, and shape information present in the image. The features extracted using traditional methods are known as handcrafted features. These handcrafted features can be considered individually for texture, color, and shape or in some combination of these three in the CBIR process. Currently the research work is focusing on using different CNN models for extracting the features efficiently from an image. The features generated by CNN models imitate human perception through various operations such as convolution and pooling and thus achieve better feature descriptors compared to the handcraft cases. Although CNN models are working well, there is still scope of improvement in terms of feature extraction. Hence in this thesis improving handcraft features is explored and also refining CNN models by making different modification in them. Further, the extracted features from these CNNs are fused with prominent handcraft features. By considering this, in the present work four different approaches for CBIR are proposed.

The first one is, a resolution independent CBIR method based on the fusion of color and texture features. As a decomposition method, the Haar wavelet transform is applied to the input images but the level is determined based on the size of the image. For color feature we use inter-channel voting between hue, saturation and intensity component of an image, as inter-relationship between color and intensity component is independent of the dimension of the image. Gray Level Co-occurrence Matrix (GLCM) on Diagonally Symmetric Pattern (DSP) is computed to get the texture features of an image. To corroborate the performance of the proposed method, it is applied on three natural datasets (Corel-1K, Corel-5K & Corel-10K) and three texture datasets (VisTex, STex & Color Brodatz) as well as their multi resolution versions also.

The second contribution is based on CNN features for CBIR. Three different existing CNN models: AlexNet, VGG, and GoogleNet are used to extract the features and these features are used for image retrieval. The fusion of different CNN architectures along with the handcraft features for CBIR is proposed. As GoogleNet is giving better CBIR performance in terms of accuracy and retrieval time, to extract CNN features, the GoogleNet model is used. As handcraft feature, an improved form of dot-diffused block truncation coding (DDBTC) and Histogram of orientated gradient (HOG) feature are used along with the features used in the first contribution. The proposed method is applied to a total of ten benchmark image datasets. Along with the six image datasets used in first contribution, four more image datasets are used: ImageNet-13K, ImageNet-65K, ImageNet-130K, and UKBench.

The CNNs that are existing can still further be refined to obtain better features for CBIR. This is done in the third contribution. A of total of five refinement for the three of the existing CNNs: GoogleNet, ResNet-50, and DarkNet-53 are proposed. The five refinements proposed are: (i) Residual-GoogleNet: It is an improved version of GoogleNet. In this proposed CNN architecture, Residual connection is used in every inception layer of GoogleNet to solve the 'Degradation Problem'. (ii)Casecade-ResNet-50: It is a modified version of ResNet-50, where the Residual connection is established among the 5 different size reparative blocks: 56×56, 28×28, 14×14, 7×7. (iii)GroupNormalized-Inception-DarkNet-53: It is a refined version of DarkNet-53 is where inception module is incorporated with the basic structure of DarkNet53 which contains three 3×3, one 5×5, and one 1×1 convolution filter instead of only a 3×3 filter used in DarkNet-53. After each convolution layer, Group Normalization (GN) layer is used instead of the BN layer used in DarkNet-53 which makes the training process of the proposed CNN models independent of batch size. A total of five such Inception modules are added with the Darknet-53 structure so that the proposed model can extract more detailed and hierarchical features. (iv) Xception-DarkNet-53: This is another modification of DarkNet-53, where the Xception concept, which is an extension of Inception, is incorporated with DarkNet-53. This model also employs convolution layers of 1×1, 3×3, and 5×5 filter sizes, but instead of a typical 2D convolution operation, 'Grouped Convolution' is used. By stacking three times each of the 1×1, 3×3, and 5×5 size Group convolution filters, one Xception module yields a total of nine group convolution filters. (v) Shuffled-Xception-DarkNet-53: This is the final modification of DarkNet-53, in which the same Xception module is utilized, but one Channel Shuffling layer is inserted between each pair of stacked Group Convolutions of the same size to preserve information flow.

Finally, the fusion of the five refined models along with prominent handcrafted features are used for CBIR in the fourth work. To evaluate the efficiency of our proposed method, five standard performance measures are evaluated: Average Precision Rate (APR), Average Recall Rate (ARR), F-Measure, Average Normalized Modified Retrieval Rank (ANMRR), and Total Minimum Retrieval Epoch (TMRE). and are compared with different existing methods.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

2S-RN	Two-Stage Relation Network Module
ANMRR	Average Normalized Modified Retrieval Rank
APR	Average Precision Rate
ARR	Average Recall Rate
AVF-RN	Aggregated Visual Features Relation Network
AVGRR	Average Retrieval Rank
BLVC	Block Variation of Local Correlation
BN	Batch Normalization
BOF	Bag of Features
BoVW	Bag of Visual Word
BoWBIR	Bag of Words Based Image Retrieval
BPF	Bit Pattern Feature
BTC	Block Truncation Coding
CBIR	Content Based Image Retrieval
CCS	Complementary CNN and SIFT
CHF	Color Histogram Feature
CNN	Convolutional Neural Network
CS_LBP	Centrally Symmetric Local Binary Pattern
DBN	Deep Belief Network
DDBTC	Dot Diffused Block Truncation Coding
DSP	Diagonally Symmetric Pattern
FPGA	Field Programmable Gate Arrays
FC	Fully-Connected
GLCM	Gray Level Co-occurrence Matrix
GN	Group Normalization
GPU	Graphic Processing Unit
HOG	Histogram of Orientated Gradient
HSI	Hue, Saturation, and Intensity.
ILSVRC	ImageNet Large Scale Visual Recognition Challenge

K	One Thousand
LBP	Local Binary Pattern
LECoP	Local Extrema Co-occurrence Pattern
LDP	Local Derivative Patterns
LEP	Local Extrema Patterns
LHdP	Local Hexadeca Patterns
LNP	Local Neighbor Pattern
LOtP	Local Octa Patterns
LR	LeakyRelu
LTrP	Local Tetra Patterns
MBW	Morphological Binary Wavelet
MHDW	Morphological Haar Dilation Wavelet
NMRR	Normalized Modified Retrieval Rank
PCA	Principal Component Analysis
PDE	Partial Differential Equations
QBIC	Query through Image Content
R-CBIR	Relational Content Based Image Retrieval
R-VQA	Relation Visual Question Answering
ReLU	Rectified Linear Activation Function
RF	Relevance Feedback
SeBIR	Semantic Based Image Retrieval
SIFT	Scale-Invariant Feature Transform
SISR	Single Image Super-Resolution
SpBIR	Speech Based Image Retrieval
STex	Salzburg Texture Image Database
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TBIR	Text Based Image Retrieval
TMRE	Total Minimum Retrieval Epoch
ULBP	Uniform Local Binary Pattern
VDSR	Very Deep Super Resolution

VGG-16	Visual Geometry Group-16
VGG-19	Visual Geometry Group-19
VQ	Vector Quantization
YCbCr	Green (Y), Blue (Cb), Red (Cr) (Digial Video Color Space)

Chapter 1

Introduction

With the growth of the internet and multimedia technologies, a massive amount of multimedia data in the form of images, audio, and videos has been used in many fields, including medical care, satellite imagery, video, still image archives, and surveillance systems. As a result, there is an ongoing demand for systems that can properly store and retrieve multimedia data. Many multimedia information storage and retrieval systems have been introduced to meet these demands.

1.1 Image Retrieval

Image retrieval is an example of a multimedia system in which images are browsed, searched for, and retrieved from a huge dataset of digital images. In order to implement retrieval over the annotation words, most of the conventional and widespread image retrieval systems use some techniques of adding metadata to the images, such as captioning, keywords, or descriptions. Five different Image Retrieval methods are commonly used: 1). Text Based Image Retrieval 2). Content Based Image Retrieval 3). Speech Based Image Retrieval, 4). Semantic Based Image Retrieval, and 5). Bag of Words Based Image Retrieval. All these different types of image retrieval methods are shown in Figure 1.1.

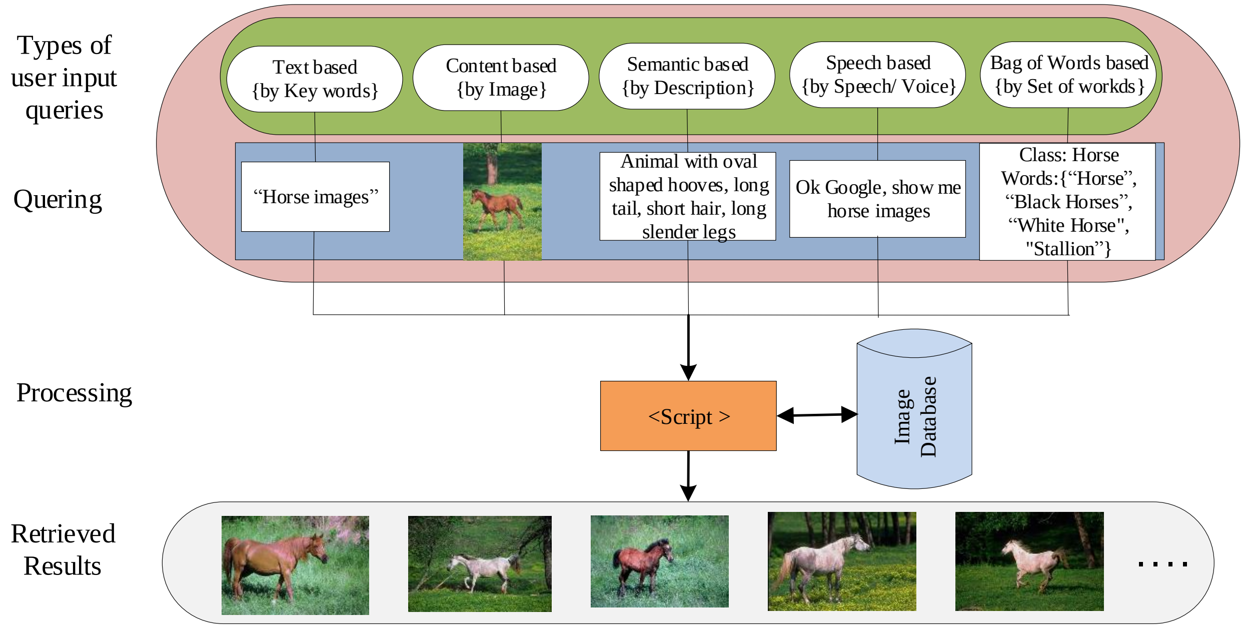

Types of user input queries
Quering
Processing
Retrieved Results
Text based {by Key words}
Content based {by Image}
Semantic based {by Description}
Speech based {by Speech/ Voice}
Bag of Words based {by Set of workds}
"Horse images"
Animal with oval shaped hooves, long tail, short hair, long slender legs
Ok Google, show me horse images
Class: Horse Words:{"Horse", "Black Horses", "White Horse", "Stallion"}
<Script >
Image Database
. . . .

Figure 1.1 Image Retrieval System.

1.1.1 Text Based Image Retrieval

The most often utilized retrieval systems were text-based image retrieval (TBIR) systems, in which the automatic or manual annotation of an image is used as the searching mechanism. A traditional TBIR searches the dataset for the text that surrounds the image, as specified in the query string. The most popular TBIR system is Google Images. Text-based systems are faster because string matching is a less time-consuming procedure. However, expressing the complete visual content of images in words is usually challenging, and TBIR can produce minor findings. Additionally, image annotation is not always appropriate and takes a long time. To find an alternate way to search and overcome the limits imposed, more intuitive and user-pleasant approaches like content-based image retrieval systems were developed.

1.1.2 Content Based Image Retrieval

The Content Based Image Retrieval (CBIR) framework represents the images in the databases by using the visual content of the given images which is calculated from low-level image features like color, texture, and shape information. When the system receives a sample image or sketch as input, it retrieves identical images from the image dataset. This approach eliminates the need of describing the visual content of images in terms of words and it becomes more similar to how people perceive visual data. Query By Image Content (QBIC) [1], Robust CBIR approach [2], and Blob World [3] make up the majority of the other representative CBIR systems.

1.1.3 Speech Based Image Retrieval

An auditory tag is associated with an image in Speech Based Image Retrieval (SpBIR) at the time of capture or afterward. Additional metadata, such as location, user identification, date and time of collection, and image-based attributes, is also recorded with the images. Users can use speech to search for images in their personal repository. [4,5]

1.1.4 Semantic Based Image Retrieval

In Semantic Based Image Retrieval (SeBIR), the image that is sought after is located by taking into account the image's description. The semantic gap is the difference between high-level concepts (or semantics) of user queries and low-level characteristics retrieved and indexed by computers. In other words, it's challenging to connect automated CBIR systems to user requests. Most of the time, the user bases their mental comparison of similarity on high-level abstractions such as activities, entities/objects, events, or even elicited feelings. Retrieval by similarity based on low-level characteristics like color or shape will be ineffective as a result. In other words, human similarity assessments do not meet the criteria set out by the similarity measure of the CBIR system. Additionally, the majority of users find it challenging to directly search or query images using color, texture, and shape criteria. As per various literature, textual or keyword-based queries are commonly used for SeBIR systems, because they are simpler and more straightforward ways to express their information demands, textual or keyword-based queries are typically chosen [6, 7, 8]. However, it is very challenging to teach computers to comprehend or extract high-level concepts from images in the same manner that humans can. [9].

1.1.5 Bag of Words Based Image Retrieval

In the Bag of Words Based Image Retrieval (BoWBIR) approach, an automatic classification operation of an image into several classes is performed. To characterize the conceptual content of the images in each class, keywords or labels have been provided. Thus, labeling an image with keywords is considered to be classifying the image. Scene classification and object classification are two categories of image classification [10]. Object classification, for instance, concentrates on categorizing images into "concrete" categories like "person," "motorcycle," "cat," and so forth. A scene categorization, on the other hand, might be thought of as an abstract word for the scene, such as "sea," "houses," or "sunrise," which can be thought of as an assembly of several physical or entities items as a single entity. [9]

1.2 Introduction to Content Based Image Retrieval

The term CBIR originated in 1992 by T.Kato to describe experiments into automatic retrieval of images from a large dataset. CBIR is an application of image retrieval problem that is searching a digital image from a large dataset. The general CBIR architecture is given in Figure 1. 2. “Content Based” means that the search will analyze the actual contents of the images. The term “Content” in this context might refer to some features that can be derived from that image itself. In a conventional CBIR system, low-level image features such as color, texture, and shape information are used to calculate the feature vector. Convolution Neural Network (CNN) models are utilized in recent research to extract high level features from input images. A feature database is formed by each image in the image dataset. The retrieval procedure is started when a user makes a query using an example image. Using the same extraction method used to create the feature database, the query image is converted to the internal representation of the feature vector. The difference between the feature vectors of the images in the feature database and those of the query image is calculated using similarity measures. The images in the dataset are ranked according to similarity measures. Different performance parameters can be calculated based on these rankings to evaluate the effectiveness of a given CBIR method. Section 1.3 provides a detailed explanation of the traditional image processing techniques for extracting hand-crafted features, and Section 1.4 provides an explanation of the state-of-the-art technologies for extracting features, such as CNN models for CBIR.

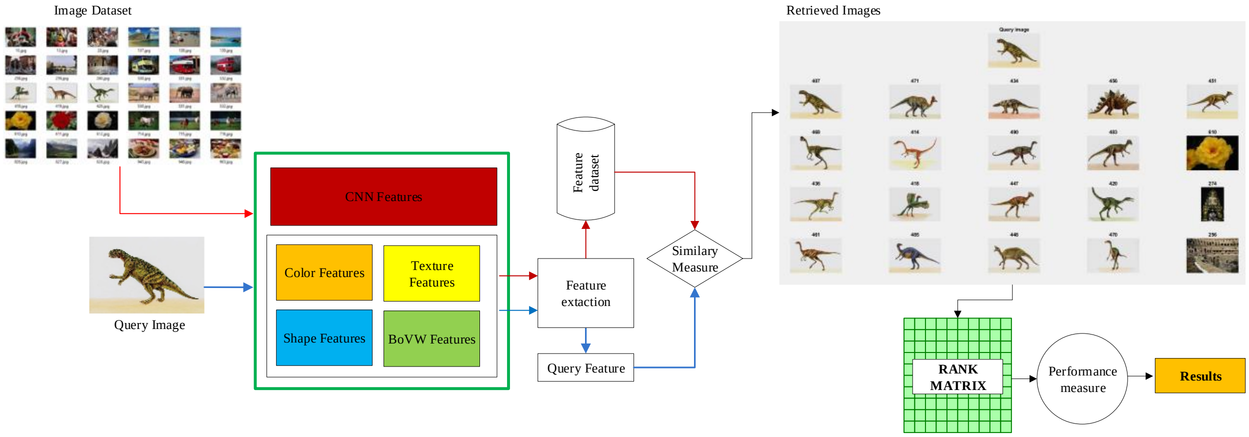


Figure 1.2 Content Based Image Retrieval Architecture.

1.3 Different Handcrafted Features Methods for CBIR

Features of an image can be calculated based on the contents of that image. These features are often expressed as a vector known as the feature vector. Various feature extraction methods exist among the pixels of the image and are based on color information, texture information, shape information, and key point information. If the features are extracted based on a typical image processing method, they are referred to as Handcrafted features. The four types of handcrafted feature extraction methods are based on the 'information in the image' utilized to extract the features from the image. They are as follows: i) color-based method, ii). Texture-based methods, iii) Shape-based methods, and iv) Key point (or interest point) based methods.Various research has been done on each of the categories of CBIR, and some of them are discussed in detail in chapter 2.

1.4 Introduction to CNN Methods for CBIR

The content of an image can also be processed using state-of-the-art technologies to extract the features. To extract the features in these technologies, a number of analysers are applied concurrently. For this, various Convolutional Neural Network (CNN) models are applied. Each

CNN is unique in terms of the number of layers employed to extract features. To extract the features, the existing CNN model with transfer learning can be employed. CNN models that can be utilized for CBIR include AlexNet, VGG, GoogleNet, ResNet, DarkNet-53 and others. In Chapter 2, some of the CNN models utilized for feature extraction are discussed in detail.

1.5 Motivation, Aim and Problem Statement of the Thesis

Motivation:

The following three issues were motivating for doing this research work.

- ❖ The wide use of Convolutional Neural Networks (CNN) for CBIR, which outperforms the existing conventional CBIR methods.
- ❖ Feature fusion of CNN and handcrafted features gives an efficient state-of-art method for CBIR.
- ❖ Various fields of application of CBIR.

The application of CBIR include:

- o Medical Applications: MRI Image analysis, Brain tumor image retrieval, Lung image retrieval, Histological image retrieval, Pathological image retrieval.
- o Remote Sensing Applications: Remote sensing image retrieval, Satellite image retrieval, Geographical image retrieval, Aerial image retrieval.
- o Security Applications: Encrypted image retrieval, Biometric security system using CBIR, Airport video monitoring.
- o Forensic Applications: Glare based image retrieval, Tattoo image retrieval, Camera identification based Image retrieval.
- o Business Applications: E-commerce application, Electronic retailing, Trademark image retrieval, Logos image retrieval.
- o Natural Image Applications: Natural image retrieval, color image retrieval, Texture image retrieval.
- o Other Applications: Ancient Chinese character image retrieval, Image collection organization, Video retrieval using CBIR.

Aim: This thesis aims to provide effective CBIR methods by proposing refined versions of existing CNN models and fusing these CNN features with handcrafted features to introduce new CBIR approaches.

Problem Statement: As the image data is becoming bigger in size, retrieving the required image from a group image is a challenging task. To address this, different new Handcrafted features and Deep Convolutional Neural Network (DCNN) features are needed to get a better CBIR system.

1.6 Objectives and Contributions of the Thesis

This thesis is devoted to improving both existing Deep learning and handcrafted techniques involved in feature extraction. The objectives of the thesis are:

- To develop a new handcrafted feature technique by considering color and texture features for CBIR
- To combine newly developed handcrafted techniques with CNN for efficient CBIR.
- To improvise different CNN architectures for CBIR in terms of Precision, Recall, F-measure, ANMRR, and TMRE.
- To combine the improvised CNN architectures with handcrafted features for better CBIR.

To achieve the given objectives the contributions of the thesis are listed below and it is represented in a structure diagram in Figure 1.3.

- A Resolution Independent Handcrafted Feature Extraction Method for Content Based Image Retrieval.
- A feature fusion based CBIR framework which combines both Deep learning (GoogleNet) and Handcrafted features.
- Modified version of the existing Nets: GoogleNet, Resnet-50, and DarkNet-53 for CBIR.
- A feature fusion based on Modified CNN based features and Handcrafted features for CBIR.

1.7 Different Benchmark Image Datasets used in this Work

To evaluate the proposed method and to show the comparative results with different CBIR methods, in this thesis a total of ten image datasets are used. Among these ten datasets three datasets are natural datasets (Core-1K, Corel-5K, and Corel-10K: *‘K’ stands for one thousand*), three are color-texture datasets (VisTex, STex, and Color Brodatz), the other three are subsets of the ImageNet dataset (ImageNet-13K, ImageNet-65K, and ImageNet-130K), and the tenth image dataset is a near-duplicate dataset (UKBench) in which one group is created by considering only one object taken from a different viewpoint. In addition to these image datasets, multiresolution versions of the natural datasets (Core-1K, Corel-5K, and Corel-10K) and color-texture datasets (VisTex, STex, and Color Brodatz) are also used in Chapter-3 in which we have proposed a resolution independent CBIR approach.

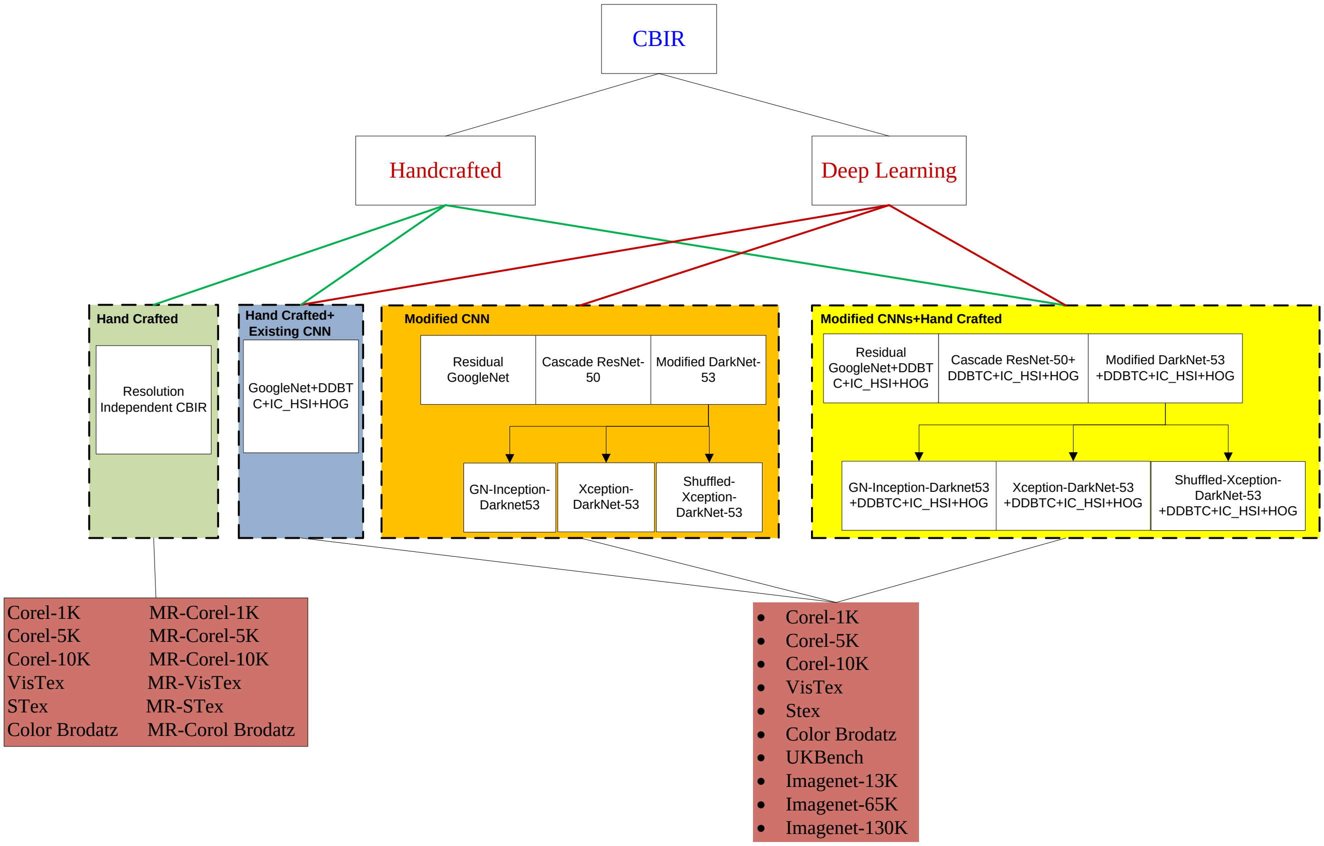

CBIR
Handcrafted
Deep Learning
Hand Crafted
Resolution Independent CBIR
Hand Crafted+ Existing CNN
GoogleNet+DDBTC+IC_HSI+HOG
Modified CNN
Residual GoogleNet
Cascade ResNet-50
Modified DarkNet-53
GN-Inception-Darknet53
Xception-DarkNet-53
Shuffled-Xception-DarkNet-53
Modified CNNs+Hand Crafted
Residual GoogleNet+DDBTC+IC_HSI+HOG
Cascade ResNet-50+ DDBTC+IC_HSI+HOG
Modified DarkNet-53 +DDBTC+IC_HSI+HOG
GN-Inception-Darknet53 +DDBTC+IC_HSI+HOG
Xception-DarkNet-53 +DDBTC+IC_HSI+HOG
Shuffled-Xception-DarkNet-53 +DDBTC+IC_HSI+HOG
Corel-1K MR-Corel-1K
Corel-5K MR-Corel-5K
Corel-10K MR-Corel-10K
VisTex MR-VisTex
STex MR-STex
Color Brodatz MR-Corol Brodatz
Corel-1K
Corel-5K
Corel-10K
VisTex
Stex
Color Brodatz
UKBench
Imagenet-13K
Imagenet-65K
Imagenet-130K

Figure 1.3 Structure Diagram of CBIR Methods and Datasets in this Work.

1.7.1 Image Dataset-1 (Corel-1K)

The Corel-1K dataset [11], the first image dataset used to compare the performance of the proposed approach to that of other methods currently in use, consists of 10 categories with a total of 100 images each. Africans (1–100), Beaches (10–200), Buildings (20–300), Buses (30–400), Dinosaurs (401–500), Elephants (501–600), Flowers (601–700), Horses (701–800), Mountains (801–900), and Food (901-1000) are ten classes in this dataset. The images in this dataset are either 256×384 or 384×256 in size. Figure 1.4 displays three images for each category, resulting in a total of 30 images from this dataset.

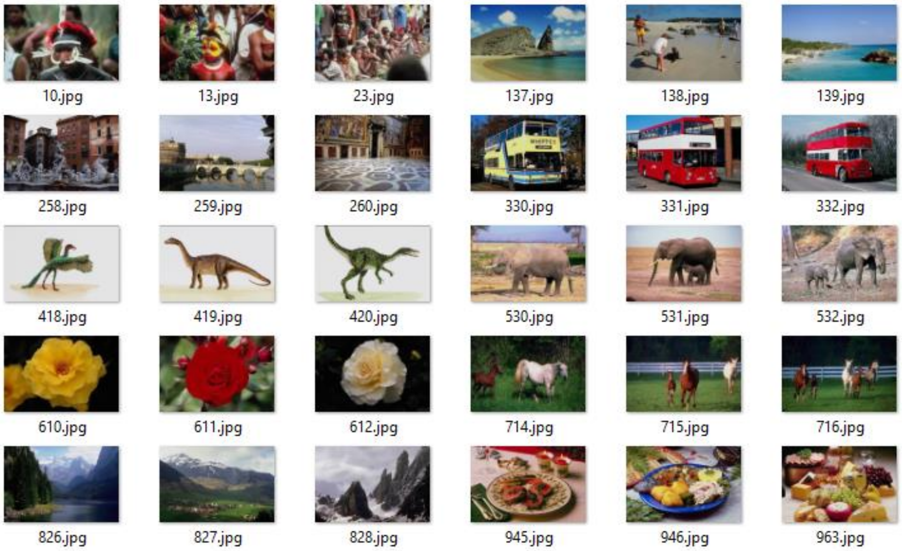


Figure 1.4 Corel-1K Image Dataset Samples (Three Images per Category).

1.7.2 Image Dataset-2 (Corel-5K)

The next dataset considered for experimentation is the Corel-5K image dataset [12]. It has 5,000 images divided into 50 classes of 100 images each. It includes images of animals, such as bears, foxes, lions, and tigers, as well as humans, natural sceneries, buildings, artworks, fruits, and automobiles. The image sizes in this dataset are 187×126, 126×187, 128×192, 192×128, 126×188, or 188×126. Figure 1.5 depicts one image from each class of this dataset.

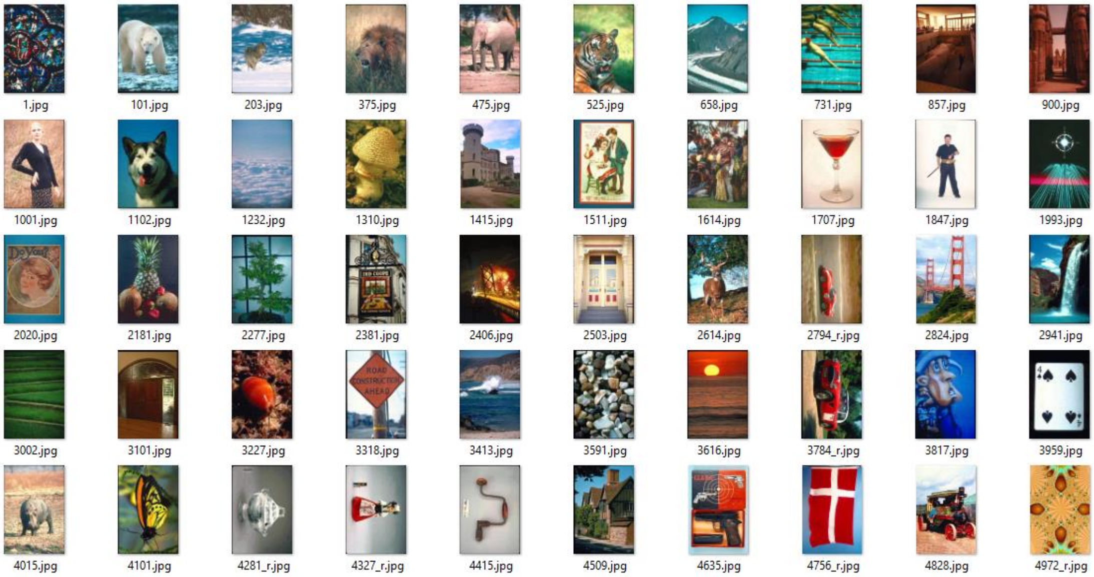


Figure 1.5 Corel-5K Image Dataset Samples (One Image per Category).

1.7.3 Image Dataset-3 (Corel-10K)

The other dataset for the experiment is the Corel-10K image dataset [12], which has 10,000 images divided into 100 classes with 100 images per category. The first 5,000 images are identical to those from Corel-5K. This dataset has a total of 53 different image sizes: 128×192, 192×128, and so on. Because this dataset is a superset of Dataset-2 (Corel-5k), Figure 1.6 shows one image from each of the additional 50 classes.

1.7.4 Image Dataset-4 (VisTex)

The other three image datasets considered to evaluate the performance of the proposed approach are colored texture datasets. Dataset-4 is based on the MIT VisTex dataset [13]. It comprises a total of 484 texture images of varying sizes. Forty texture images with dimensions of 512×512 are being selected for testing. These forty texture images are subdivided into sixteen 128 ×128 nonoverlapping subimages, yielding a total of 640 texture images. Figure 1.7 depicts the forty texture images.

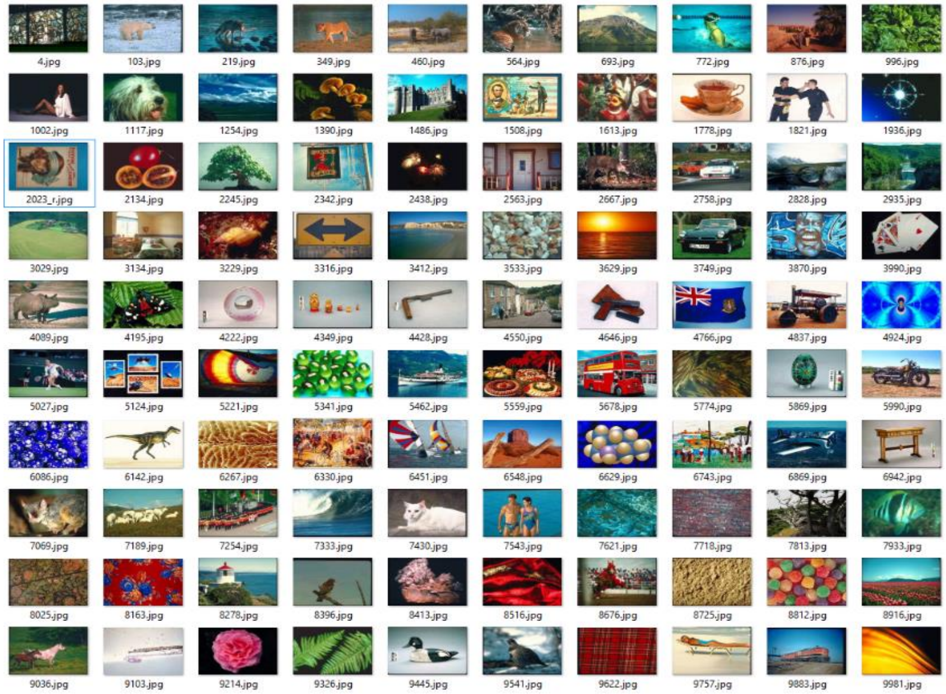


Figure 1.6 Corel-10K Image Dataset Samples (One Image per Category).

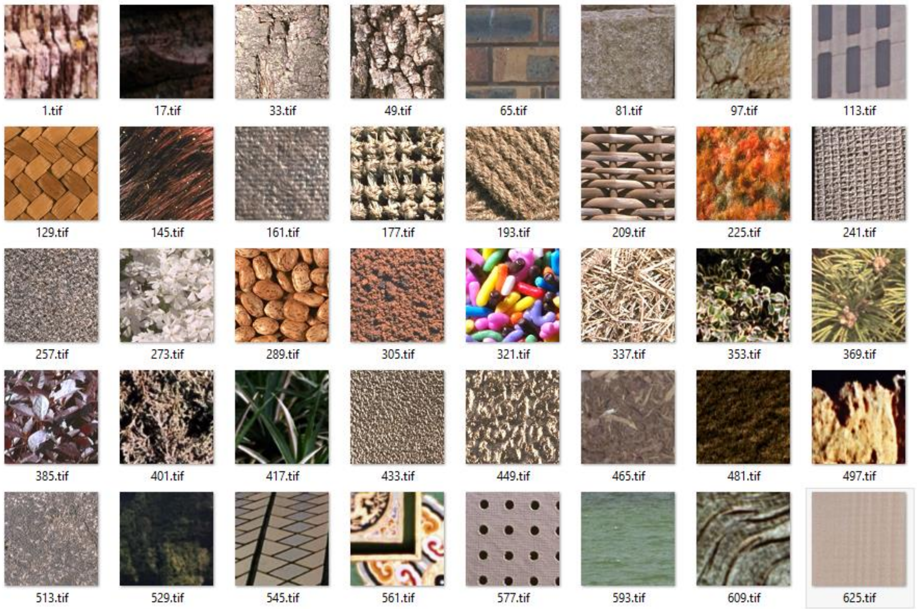


Figure 1.7 Forty VisTex Texture Images Considered.

1.7.5 Image Dataset-5 (STex)

Another color-texture picture dataset, Salzburg Texture Image Dataset (STex), is used for the experimental analysis [14]. This dataset contains 476 texture images, each of which is 512×512 in size. Each of the 476 texture images is again divided into 16 non-overlapping images of size 128×128. As a result, it produces 7,616 texture images. Figure 1.8 depicts some of the texture images from this dataset.

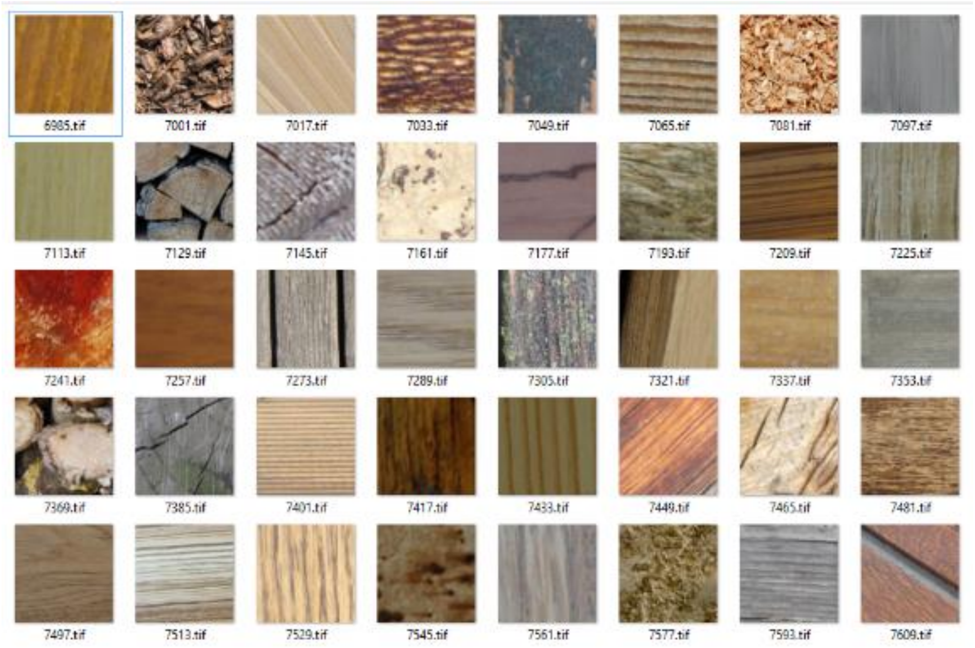


Figure 1.8 Fifty of the STex Texture Images Considered.

1.7.6 Image Dataset-6 (Color Brodatz)

Color Brodatz texture image dataset [15] is considered as the last color texture image dataset. It contains a total of 112 distinct color texture images, each of which is 640×640 pixels in size. 25 non-overlapping texture images having a size of 128×128 are created from each of these 112 images, yielding a total of 2800 texture images. Figure 1.9 depicts some of the texture pictures from this collection.

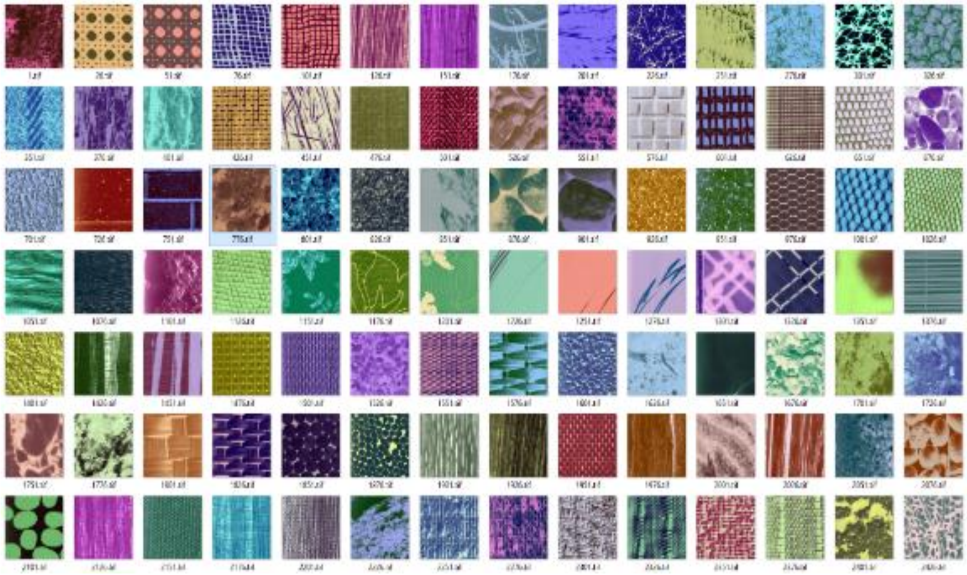


Figure 1.9 Sixty of the Color Brodatz Texture Images Considered.

1.7.7 ImageNet Dataset

ImageNet is an image dataset organized according to the WordNet hierarchy. ImageNet consists of 1,41,97,122 images organized into 21,841 classes [16]. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was an annual computer vision contest between 2010 and 2017[17]. In ILSVRC-2017 for object detection challenge, a subset of the ImageNet dataset is used. This subset contains a total of 12,81,167 training images divided into 1,000 classes, where the number of images in each class varies from 732 to 1,300 [17]. This research work has used three subsets of this training dataset to show retrieval accuracy of the proposed method, which contains 10, 50, and 100 classes, respectively.

1.7.7.1 Image Dataset-7 (ImageNet-13K)

Ten classes from the ImageNet training dataset are selected in this dataset in an interval of 100 classes. Each class contains 1,300 images resulting in a total of 13,000 images. Some of the images from this dataset are shown in Figure 1.10.

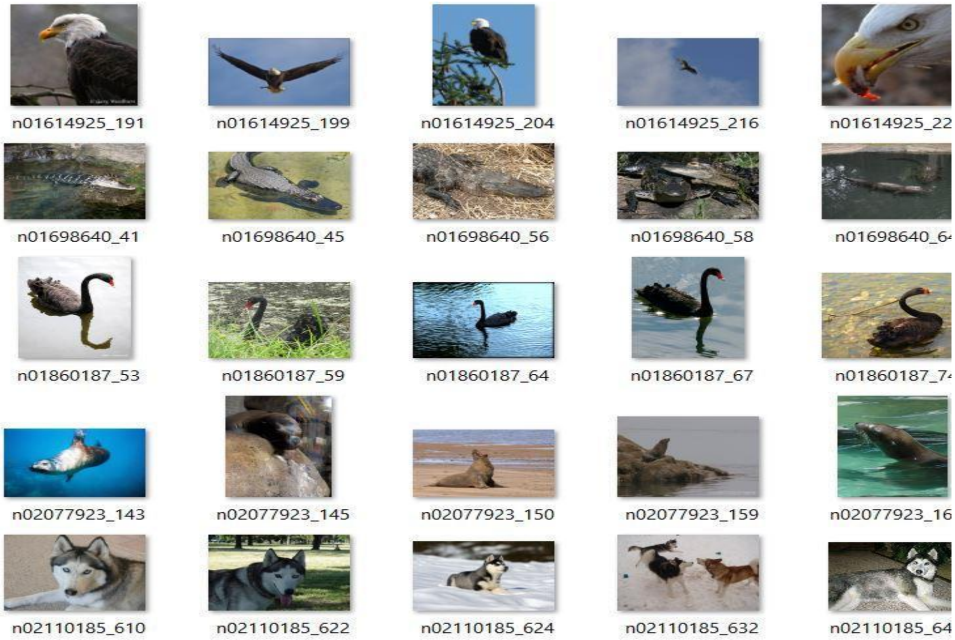


Figure 1.10 Sample Images of ImageNet-13K Dataset.

1.7.1.2 Image Dataset-8 (ImageNet-65K)

Fifty classes from the ImageNet training dataset are selected in this dataset in an interval of 20 classes. Each class contains 1,300 images resulting in a total of 65,000 images. Some of the textures images from this dataset are shown in Figure 1.11.

1.7.1.3 Image Dataset-9 (ImageNet-130K)

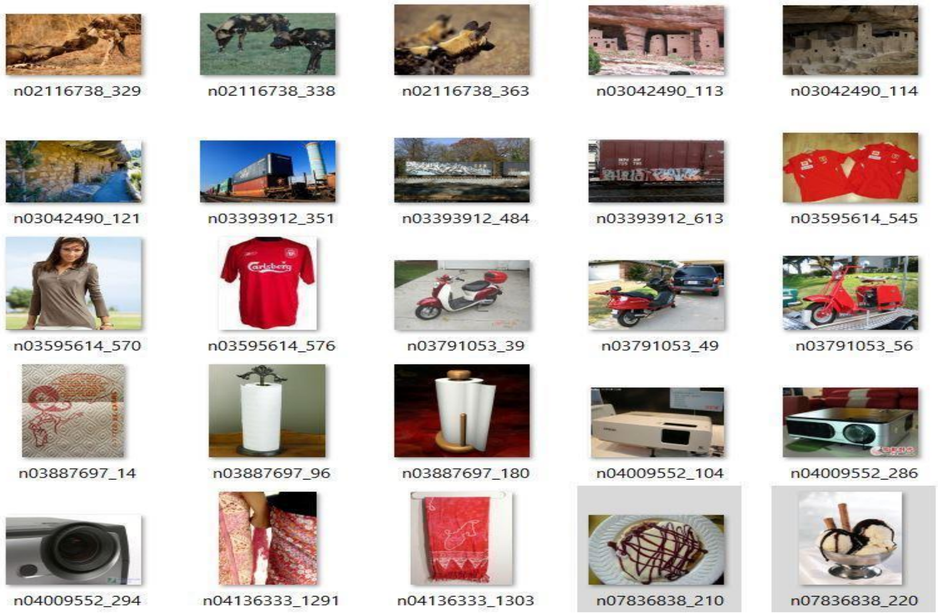


Figure 1.11 Sample Images of ImageNet-65K Dataset.

Hundred classes from ImageNet training dataset are selected in this dataset in an interval of 10 classes. Each class contains 1,300 images resulting in a total of 1,30,000 images. Some of the images from this dataset are shown in Figure 1.12.

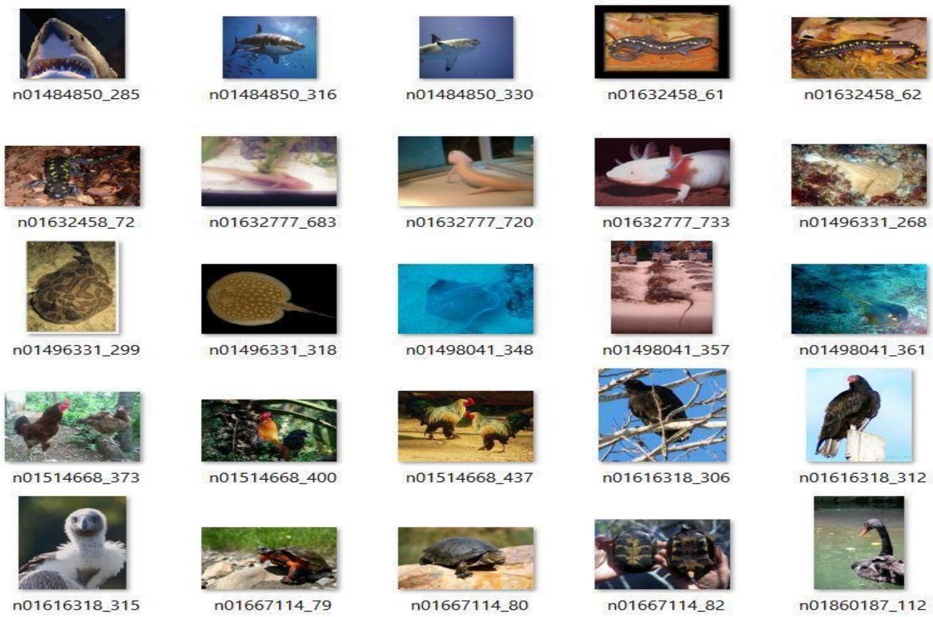


Figure 1.12 Sample Images of ImageNet-130K Dataset.

1.7.8 Image Dataset-10 (UKbench)

The UKBench dataset from Henrik Stewenius and David Nister contains 10,200 images of 2,550 groups with every four images at size 640×480. The images are rotated, blurred, and have a tendency for computer science motives. Every group of this dataset contains the same image with different rotations. Some of the textures images from this dataset are shown in Figure 1.13.

Multiresolution Corel dataset

In the multiresolution version of Corel-1K, four different resolution images are created for each class using bilinear interpolation. In this dataset, the resolution of the first 25 images is not changed. For the remaining images in the dataset, the resolution is reduced by a factor of two for every four images of each class which results in 192×128 or 128×192, 96×64 or 64×96 and 48×32 or 32×48 images respectively. The same strategy is followed for Multiresolution Corel-5K and Corel-10K. Some sample images of multiresolution Corel-1K, Corel-5K, and Corel-10K are given in Figure 1.14, 1.15, and 1.16 respectively.

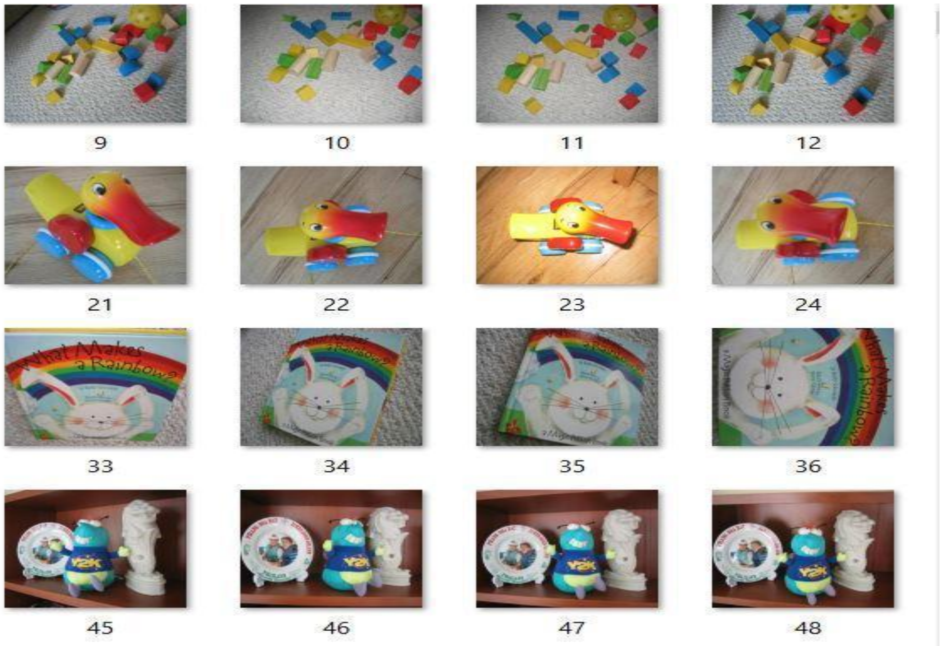


Figure 1.13 Sample Images of UKBench Dataset.

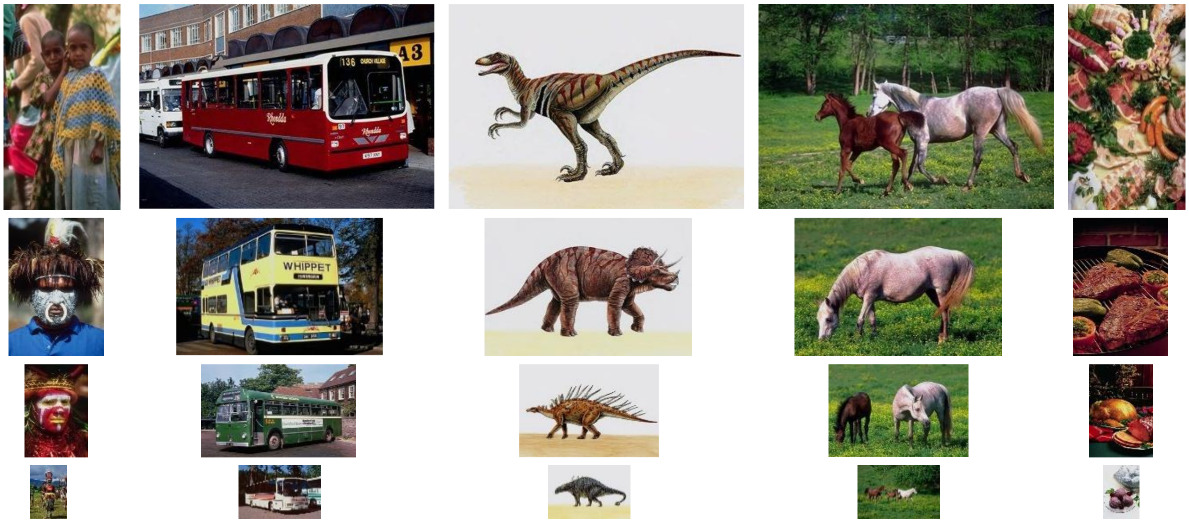


Figure 1.14 Sample Images of Multiresolution Corel-1K Dataset.

Multiresolution Color-Texture

To create the multiresolution version of VisTex and STex image datasets, four different resolution images are created for each class using bilinear interpolation. In these Multiresolution versions, the resolution of the first 4 images is not changed. For the remaining images in the dataset, the resolution is reduced by a factor of two for every four images of each class which results in 64×64, 32×32 and 16×16 images respectively. Some sample images of multiresolution Vistex and STex are given in Figure 1.17 and 1.18 respectively.

In the Multiresolution Color Brodatz image dataset, a total of five different resolution images are created. The resolution of the first five images is not changed. For the remaining images in the dataset, the resolution is reduced by a factor of two for every five images of each class which results 64×64, 32×32, 16×16, and 8×8 images respectively Some sample images of multiresolution Color Brodatz is given in Figure 1.19.

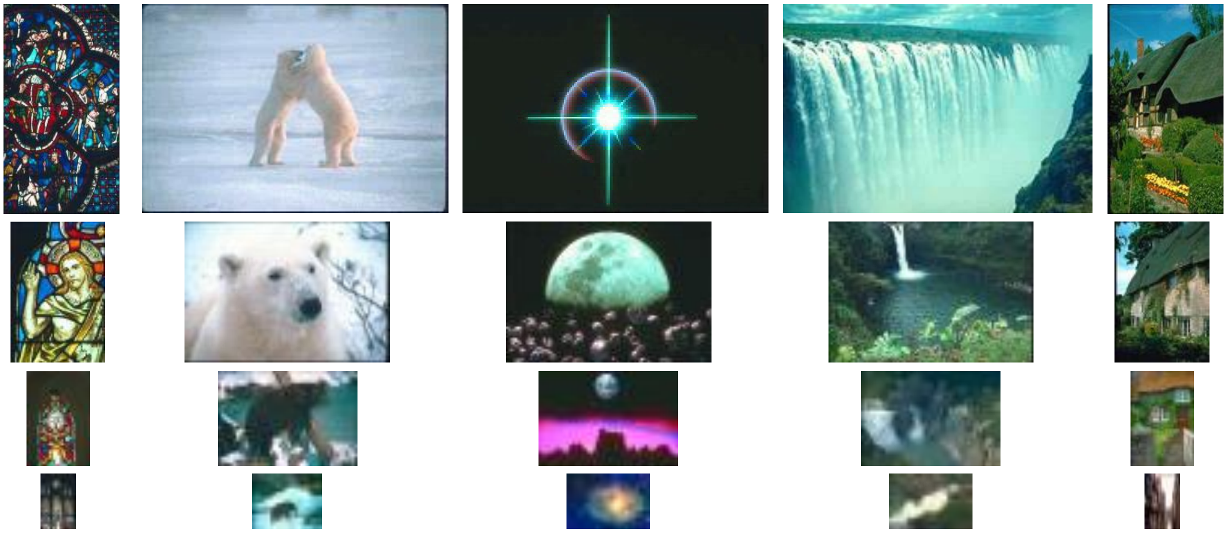


Figure 1.15 Sample Images of Multiresolution Corel-5K Dataset.

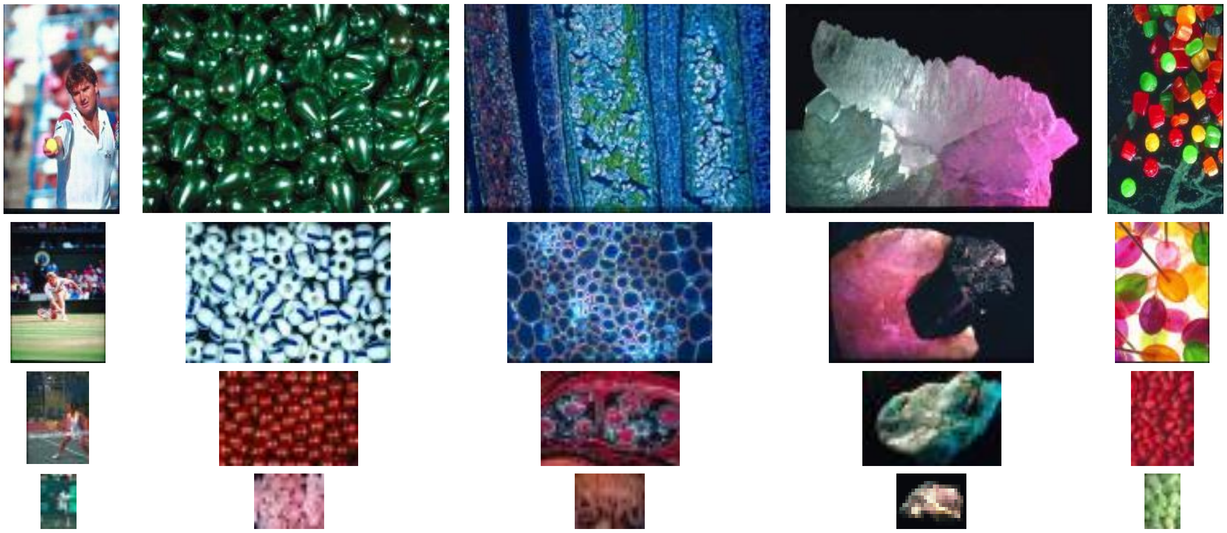


Figure 1.16 Sample Images of Multiresolution Corel-10K Dataset.

Multiresolution Color-Texture

To create the multiresolution version of VisTex and STex image datasets, four different resolution images are created for each class using bilinear interpolation. In these Multiresolution versions, the resolution of the first 4 images is not changed. For the remaining images in the dataset, the resolution is reduced by a factor of two for every four images of each class which results in 64×64, 32×32 and 16×16 images respectively. Some sample images of multiresolution Vistex and STex are given in Figure 1.17 and 1.18 respectively.

In the Multiresolution Color Brodatz image dataset, a total of five different resolution images are created. The resolution of the first five images is not changed. For the remaining images in the dataset, the resolution is reduced by a factor of two for every five images of each class which results 64×64, 32×32, 16×16, and 8×8 images respectively Some sample images of multiresolution Color Brodatz is given in Figure 1.19.

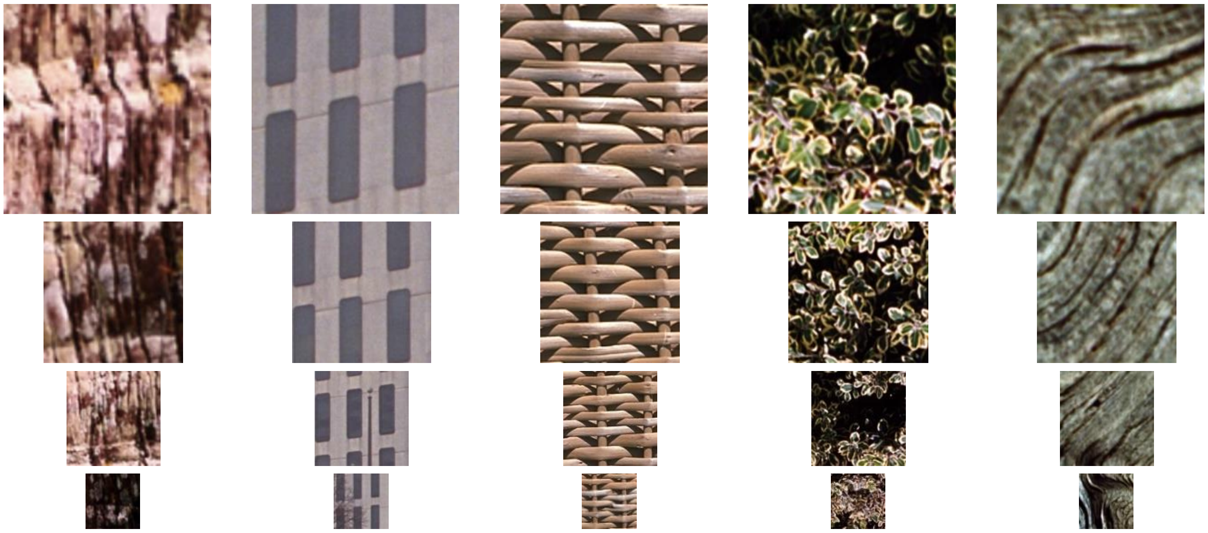


Figure 1.17 Sample Images of Multiresolution VisTex Dataset.

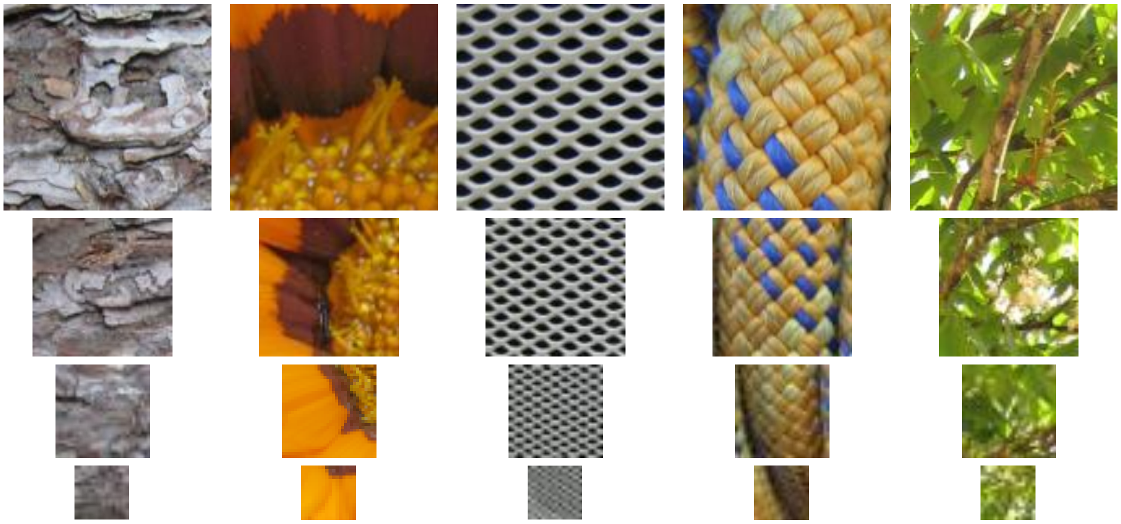


Figure 1.18 Sample Images of Multiresolution STex Dataset.

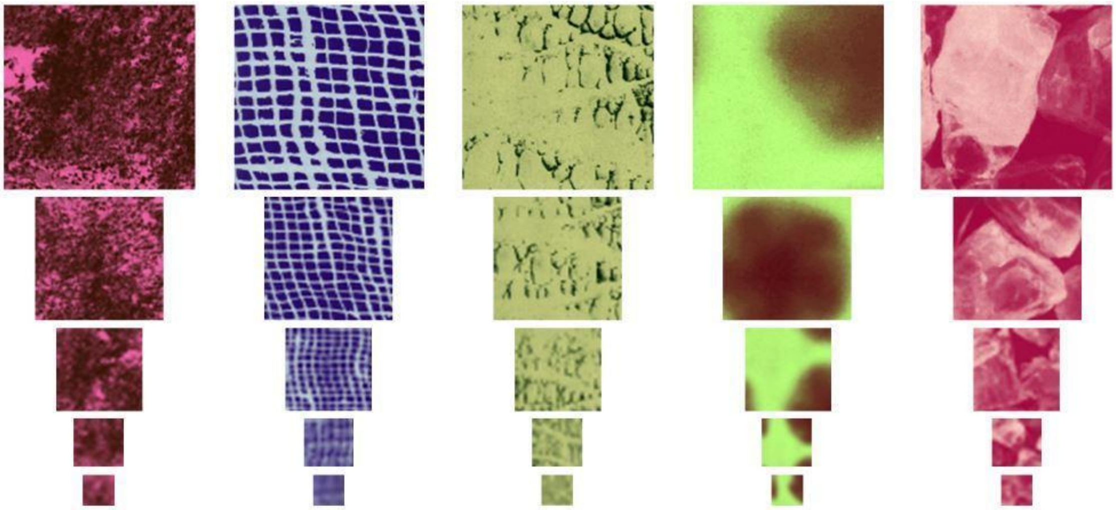


Figure 1.19 Sample Images of Multiresolution Color Brodatz.

1.8 Different Performance Measure for Comparing CBIR Methods

Different performance measures can be used for the performance evaluation of a CBIR method. Precision, Recall, F-measure, Average Normalized Modified Retrieval Rank (ANMRR), and Total Minimum Retrieval Epoch (TMRE) are used in this work as the performance measures.

Significance of these parameters:

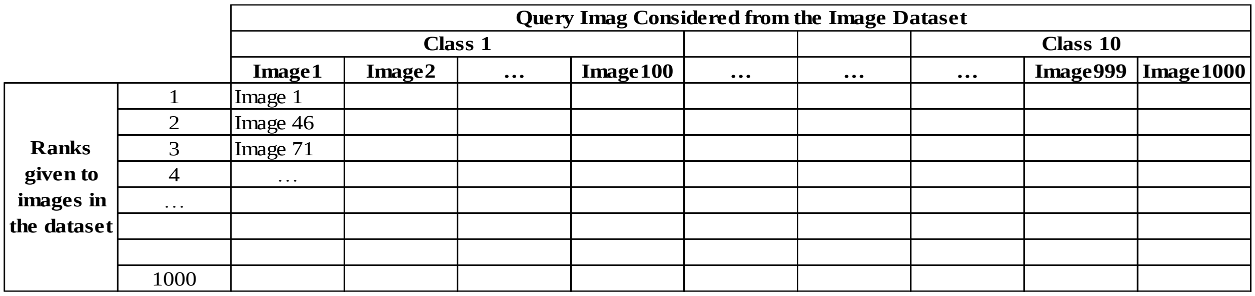
- Precision gives us the accuracy of *m* correctly retrieved images out of *n* selected images.
- Recall gives us the accuracy of *m* correctly retrieved images out of total images of the same group.

- The F-measure is a measure of a method’s accuracy on a dataset. The F-measure is a way of combining the precision and recall of the method, and it is defined as the harmonic mean of the method’s precision and recall.
- Average Normalized Modified Retrieval Rank (ANMRR): It is also used to measure the retrieval accuracy. This parameter is mainly useful when the number of images in each group is not the same.
- Total Minimum Retrieval Epoch (TMRE) is used to determine the minimum number of images needed to traverse to find all the same group images of the input image. Ideally the *n* should be the perfect value of MRE, where *n* is the total number of images in the group.

At the time of evaluation, a feature vector is extracted from each query image. Feature vector comparison is performed among the feature vector of the query image and that of all images of the image dataset. This comparison is done based on distance measure (in this thesis d1-distance used) given in equation 1.1 In this equation $F_{db}(S)$ represents the feature vector of the dataset images and $Fq(S)$ is the ‘query’ image’s feature vector

$$d\left(db_i, q\right) = \sum_{s=1}^{len} \left| \frac{F_{db_i}(s) - F_q(s)}{1 + F_{db_i}(s) + F_q(s)} \right| \qquad (1.1)$$

Based on the distances, a rank matrix of size $N \times N$ is formed, where N is the total number of images in the dataset as shown in Figure 1.20. The value of each cell of the rank matrix is given as Rank(k,i) which is k^{th} similar image with respect to i^{th} query image.



		Query Imag Considered from the Image Dataset								
		Class 1						**Class 10**		
		Image1	**Image2**	**…**	**Image100**	**…**	**…**	**…**	**Image999**	**Image1000**
Ranks given to images in the dataset	1	Image 1								
	2	Image 46								
	3	Image 71								
	4	…								
	…									
	1000									

Figure 1.20 Rank Matrix Representation for 1,000 Images Dataset.

Precision & Recall:

Precision is described as the ratio of total number of relevant images retrieved to the number of images being retrieved for a query image. Recall is described as the ratio of total number of relevant images retrieved for a query image to the total number of images in that category. Let N_{ic}

be the total number of images in a category C_i. Let a total of n images be retrieved from a dataset for a query image i from that category, then precision and recall are calculated by using equations 1.2 to 1.4. The average precision and average recall for j^{th} category using equations 1.5 and 1.6. If the total number of categories is N_c, then Average Precision Rate(APR) and Average Recall Rate(ARR) for n images are calculated by using equations 1.7 and 1.8.

$$P(i,n) = \frac{1}{n}\sum_{k=1}^{n} f_3(Rank(k,i)) \qquad (1.2)$$

$$R(i,n) = \frac{1}{N_{ic}}\sum_{k=1}^{n} f_3(Rank(k,i)) \qquad (1.3)$$

$$f_3\left(Rank(k,i)\right) = \begin{cases} 1, & rank(k,i) \in C_i \\ 0, & else \end{cases} \qquad (1.4)$$

$$P_{avg}(j,n) = \frac{1}{N_{ic}}\sum_{i=1}^{N_{ic}} P(i,n) \qquad (1.5)$$

$$R_{avg}(j,n) = \frac{1}{N_{ic}}\sum_{i=1}^{N_{ic}} R(i,n) \qquad (1.6)$$

$$APR(n) = \frac{1}{N_c}\sum_{j=1}^{N_c} P_{avg}(j,n) \qquad (1.7)$$

$$ARR(n) = \frac{1}{N_c}\sum_{j=1}^{N_c} R_{avg}(j,n) \qquad (1.8)$$

F-measure:

Precision and Recall are good performance measures. To understand the relationship between these two measures, we calculate the third performance measure *F-measure* as given in equation 1.9.

$$F - measure(n) = \frac{(2 \times APR(n) \times ARR(n))}{(APR(n) + (ARR(n))} \qquad (1.9)$$

***Average Normalized Modified Retrieval Rank (ANMRR)**:*

To measure the retrieval accuracy, we used ANMRR measure which can be calculated using equations 1.10. The average retrieval rank (AVGRR) is calculated using equation 1.11. Now, normalized modified retrieval rank (NMRR) for query image *i*, can be calculated using equation 1.12. Finally, the average NMRR for the entire dataset can be calculated as given in equation 1.13.

$$RR(k,i) = \begin{cases} f_5(k,i), \quad if \quad f_5(k,i) \le f_6(i) \\ \quad 1.25 \times f_6(i), \qquad\qquad else \end{cases} \tag{1.10}$$

$\forall k \in C(i)$, where $C(i)$ is the set of all the images of the category C$_i$,

$f_5(k,i) = x, where\ Rank(x,i) = k$ and

$f_6(i) = (4 \times N_{ic}, 2 \times \{max(N_{ic})\ , \forall i\})$.

$$AVGRR(i) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} RR(k,i) \tag{1.11}$$

$$NMRR(i) = \frac{AVG(i) - 0.5[1 + N_{ic}]}{1.25 f_5(i) - 0.5[1 + N_{ic}]} \tag{1.12}$$

$$ANMRR = \sum_{i=1}^{DB} NMRR(i) \tag{1.13}$$

where, $DB = N_c \times N_{ic}$, total number of images in dataset.

Total Minimum Retrieval Epoch *(TMRE):*

In addition to the above mentioned four performance measures, one new performance measure is introduced for overall performance evaluation of the entire dataset, which is known as minimum retrieval epoch ratio. It is defined as an average of ratios between the average number of images to be traversed to get all the relevant images for each query image of a category and total number of images of that category. To get TMRE for the entire dataset, initially, minimum retrieval epoch MRE(i) for each query image can be calculated by using the equation 1.14. Average MRE for each category $AMRE(C_i)$ can be calculated using equation 1.15. Then AMRE value is divided by total number of images in corresponding category to make performance measure more generalize for any dataset. AMRE Ratio for one category $AMRER(C_i)$is calculated using equation 1.16. Finally, AMRER for entire dataset i.e. TMRE is calculated using equation 1.17.

$$MRE(i) = \max(k), \forall k \exists Rank(k,i) \in C_i \tag{1.14}$$

$$AMRE(C_i) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} MRE(i) \tag{1.15}$$

$$AMRER(C_i) = \frac{AMRE(C_i)}{N_{ic}} \tag{1.16}$$

$$TMRE = \frac{1}{N_c} \sum_{C_i=1}^{N_c} AMRER(C_i) \qquad (1.17)$$

1.9 Organization of the Thesis

The rest of the chapters of this thesis are organized as follows: Chapter-2 describes detailed literature review on content based image retrieval (CBIR) using different handcraft features and CNN features. Chapter-3 proposes a resolution independent feature extraction method for CBIR and results on six image datasets are given and compared. Chapter-4 proposes a CBIR method using Feature Fusion of Deep Learning and Handcraft features. All the ten image datasets are used for evaluating this method. In Chapter-5, a total of five modified CNN architectures for CBIR are given along with the results on all the ten image datasets. However, in Chapter-6, CBIR using fusion of refined CNN's features along with handcraft features is given with results. To visualize the performance of all the proposed methods, in Chapter-7 the consolidated results for all the ten image datasets along with conclusion and future scope are discussed.

Chapter 2

Related Work

This chapter discusses various content based image retrieval (CBIR) techniques proposed by various researchers. The methods which are related to our proposed methods are explained in detail whereas other methods are mentioned with proper references. The extraction of color feature methods, texture feature methods, Bag of Visual Word (BoVW) feature methods, and shape feature extraction methods are all explored as part of handcrafted features in Sections 2.1 to 2.4. After that, the uses of convolution neural networks (CNN) for image classification and image retrieval are explained in Section 2.5.

2.1 Color Features for CBIR

Color-based image retrieval is an essential CBIR approach. Color features are the most evident and intuitive visual features. It is also a crucial aspect of perception. Color features are more stable and resilient than other image features like texture and shape since they are not affected by translation, rotation, or scale modifications [18]. To produce color features of an image, feature extraction procedures like color histogram [19], color correlogram [20], color auto correlogram [21, 22] inter-channel voting between hue and saturation [23] can be applied to the image. Color histogram mainly provides the color frequency information in a particular color model. To calculate the color histogram of a color image first it must be separated into different color components. The feature vector is then created by obtaining a distinct histogram for each color component and concatenating the resulting histograms. Before calculating the histogram, the pixels might be quantized into various bins for each color component to minimize the length of the feature vector. Color Correlogram is a novel color feature descriptor introduced by Haung et al.[21] This feature not only considers the frequency of each color pixel but also concentrates on the co-occurrence of all pixel pairs within a given distance. Color Autocorrelogram was proposed to decrease the feature vector length of the color correlogram. This color feature focuses solely on the co-occurrence of the same color, instead of considering all color pairs.

2.1.1 Interchannel Voting among Hue, Saturation and Intensity

All color feature extraction methods maintained above concentrate only on an individual color channel to extract color feature descriptor. Raju et al. [24] proposed a new color feature named inter-channel voting among hue, saturation, and intensity components of an HSI color image.

This method explores the interrelationship among three components of a color image. To perform inter-channel voting between saturation and intensity channels, both channels are quantized into different bins and then added with the quantized intensity value to the respective saturation bin and vice versa. This process is shown in Figure 2.1. Due to the non-commutative characteristics of inter-channel voting, the feature vector produced by inter-channel voting between hue and saturation is different from the feature vector generated by inter-channel voting between saturation and hue. The process of inter-channel voting is applied to hue and saturation, hue and intensity, and intensity and saturation components of an image separately which creates a total of 6 different feature vectors. To create the final feature vector, all six feature vectors are concatenated. Feature vector creation for '*saturation & intensity*' and '*saturation & hue*' are shown in equation 2.1 to equation 2.4 For hue and intensity components the same type of equations can be applied.

//For SATURATION//

$$Range_S = \frac{\max\left(S\left(i,j\right)\right) - \min\left(S\left(i,j\right)\right)}{S_{level}} \forall i, j \in S \tag{2.1}$$

where S_{level} is the number of quantization levels for *Saturation*.

$$S_{new}\left(i,j\right) = \begin{cases} S_{level} - 1, & S\left(i,j\right) = \max\left(S\left(i,j\right)\right) \\ \left\lfloor \dfrac{S\left(i,j\right)}{Range_S} \right\rfloor, & else \end{cases} \tag{2.2}$$

$$Bin_{SI}\left(S_{new}\left(i,j\right)\right) = Bin_{SI}\left(S_{new}\left(i,j\right)\right) + I\left(i,j\right), \forall i, j \in S \tag{2.3}$$

$$Bin_{SH}\left(S_{new}\left(i,j\right)\right) = Bin_{SH}\left(S_{new}\left(i,j\right)\right) + H\left(i,j\right), \forall i, j \in S \tag{2.4}$$

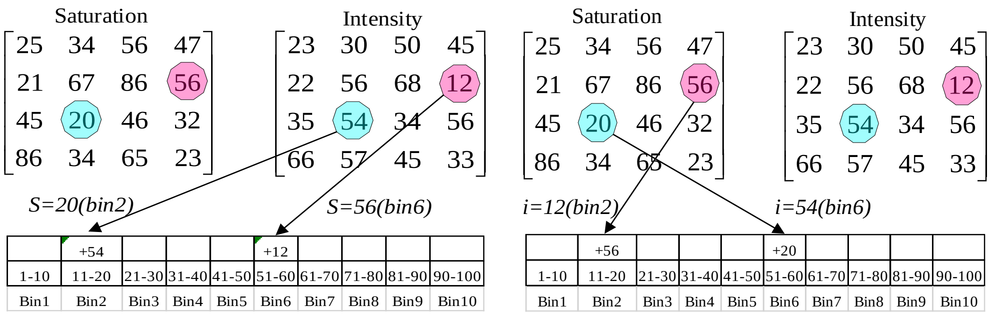


Figure 2.1 Inter-channel voting. The First 1-D array is the result of the process between ‘Saturation & Intensity’ while the second one is the result of the process between ‘Intensity & Saturation’ components.

2.2 Texture Features for CBIR

Despite the lack of a universally agreed definition, the term ‘Texture’ can be defined as significant variation in intensity between nearby pixels. Cross and Jain [25] suggested that texture is a stochastic, possibly periodic, and two-dimensional image field. A texture might be smooth or rough, soft or firm, coarse or fine, matt or glossy, and so on. Textures are classified into two types: tactile textures and visual textures. The instant physical sensation of a surface is referred to as tactile texture. The visual texture of an image refers to the visual impression that textures give to the human viewer, which is related to local spatial variations of simple stimuli such as color, direction, and intensity. Hence, the thesis is solely concerned with visual textures, the term 'texture' will be used strictly to refer to 'visual texture' unless otherwise specified.

For texture feature extraction, Local Binary Pattern (LBP) [26], Uniform Local Binary Pattern (ULBP) [26], Centrally Symmetric Local Binary Pattern (CS_LBP) [27], Local Extrema Patterns (LEP) [28], Diagonally Symmetric Pattern (DSP) is a texture feature that was proposed in [23]. Local Derivative Patterns (LDP) [29], Local Tetra Patterns (LTrP) [30] are used by different researchers. Based on Local Tetra Patterns, Local Octa Patterns (LOtP) and Local Hexadeca Patterns (LHdP) are proposed in [31]. One more texture feature descriptor is Grey Level Co-occurrence Matrix (GLCM), which reveals knowledge about pixel pair co-occurrence of the image is also used.

2.2.1 Local Binary Pattern and Uniform Local Binary Pattern

Ojala et al. [26] proposed one widely used texture feature named Local Binary Pattern (LBP). While calculating the local binary pattern, the central pixel of a 3×3 window is considered as a threshold for all other eight neighbor pixels. The LBP value of the center pixel can be calculated using equation 2.5. Figure 2.2 shows a 3×3 window of an image, where I_c is the center pixel of the window and I_1 to I_8 are eight neighbors of I_c.

$$LBP_{P,R}\left(I_c\right)=\sum_{i=1}^{P} f_1\left(I_i-I_c\right)\times 2^{i-1} \qquad (2.5)$$

$$f_1(x)=\begin{cases}0, & x<0\\ 1, & x\geq 0\end{cases}$$

where the number of neighbors is represented by P and the radius of the neighborhood can be expressed as R and $f_1(x)$ is the step function used to consider I_c as a threshold. The feature vector can be obtained from the LBP image by using equation 2.6. Here, the length of the feature vector is the number of distinct values in the LBP image. The process of calculating LBP for a central pixel (45) is shown with an example in Figure 2.3, which results in LBP (45) to 201. To extract rotation invariant texture features, some extended version of LBPs is also proposed like rotational completed LBPs [32], Multichannel Decoded LBPs [33], etc. various versions of LBP are described in [34].

I_1	I_2	I_3
I_8	I_c	I_4
I_7	I_6	I_5

Figure 2.2. 8-Neighborhood of pixels around I_c.

$$H(L)\big|_{LBP}=\sum_{m=1}^{M}\sum_{n=1}^{N} 1-\left\lceil \frac{\left|LBP(m,n)-L\right|}{L_{\max}} \right\rceil ; L\in\left[0,2^{P}-1\right] \qquad (2.6)$$

84	60	32
76	45	40
22	11	95

1	1	0
1	🟩	0
0	0	1

128	64	32
1		16
2	4	8

Figure 2.3 LBP calculation (a) Original Image (b) Resultant binary numbers for (a). (c) Decimal weights for the corresponding locations.

Uniform Local Binary Pattern (ULBP)

To reduce the feature vector length and also because most of the features in many of the image datasets are covered with 59 bins as given in [26], ULBP is used. In ULBP, the number of bins as well as the length of the feature vector is reduced from 256 to 59 based on the number of transitions in the binary patterns. The decimal numbers are given in Figure 2.4, have the number of transitions ≤2. All the remaining decimal numbers in the range of 0 to 255, the number of transitions is 4, 6 or 8. So these decimal numbers with the number of transitions ≤2 are considered the same as those values resulting in 58 bins since we have 58 such decimal numbers. The remaining numbers are considered as the 59th bin. The entire process of ULBP is given in equations 2.7, 2.8 and 2.9. From Figure 2.3, the LBP(45) is 201. If we apply ULBP on this, as '201' is not in the list of numbers in Figure 2.4, 201 is considered to be in the 59th bin of ULBP.

0	255	1	2	3	4	6	7	8	12	14	15	16	24	28	30
31	32	48	56	60	62	63	64	96	112	120	124	126	127	128	129
131	135	143	159	191	192	193	195	199	207	223	224	225	227	231	239
240	241	243	247	248	249	251	252	253	254	All other values in range of 0-255					

Figure 2.4. Uniform Local Binary Patterns: 59-bins.

$$UP(I_c)=\sum_{j=0}^{P-1}\left|Bin(I_c, j\%P)-Bin\left(I_c,(j+1)\%P\right)\right| \qquad (2.7)$$

$$ULBP(I_c)=\begin{cases} LBP_{P,R}(I_c), & UP(I_c)\le 2 \\ y, & else \end{cases} \qquad (2.8)$$

where, $y\in Z^{+}-\{i:0\le i\le 255\}$

$$Bin(I_c,x)=\left\lfloor \frac{I_c}{2^x} \right\rfloor \%2 \qquad (2.9)$$

2.2.2 Diagonally Symmetric Pattern

Diagonally Symmetric Pattern (DSP) is a texture feature that was proposed by Bhunia et al. [23]. In this method also, the authors considered the 3×3 window of an image, and the relationship between diagonally neighboring pixels is considered to get the pattern. The principal diagonal and the counter diagonal are two diagonals for a 3×3 window of the image. Consider the center pixel as I_c and the surrounding pixels as I_i where i={1,2,3,…..,8}. Here Q_i is defined as $Q_i = I_{i+1} - I_c$ where i = {0,1,……,7}. To get DSP, we consider the values in Q_i shown in Figure 2.5. The principal diagonal pattern is obtained by considering (Q_7, Q_5), (Q_0, Q_4) and (Q_1, Q_3). Similarly, the counter diagonal pattern is obtained by considering (Q_3, Q_5), (Q_2, Q_6) and (Q_1, Q_7). If both (Q_m, Q_n) are of the same sign then the resultant bit will be 1, otherwise it will be 0. DSP is calculated by using equation 2.10 to equation 2.12. Figure 2.6.

shows the process of calculating DSP for the central pixel (55), which results in DSP (55) as 52.

$$DSP\_CD\left(I_c\right)=\sum_{i=0}^{2} f_1\left(Q_{3-i},Q_{5+i}\right)\times 2^{i} \qquad (2.10)$$

$$DSP\_\mathrm{P}D\left(I_c\right)=\sum_{i=0}^{2} f_1\left(Q_{(7+i)\%8},Q_{5-i}\right)\times 2^{i+3} \qquad (2.11)$$

$$Q_c=DSP\left(I_c\right)=DSP_{PD\left(I_c\right)}+DSP_{CD\left(I_c\right)} \qquad (2.12)$$

where,

$$f_1\left(x,y\right)=\begin{cases}1, & x\times y\geq 0\\ 0, & else\end{cases}$$

Q_0	Q_1	Q_2
Q_7	Q_c	Q_3
Q_6	Q_5	Q_4

Figure 2.5 Representation of Q_i. Q_c is the central pixel of the 3×3 image window

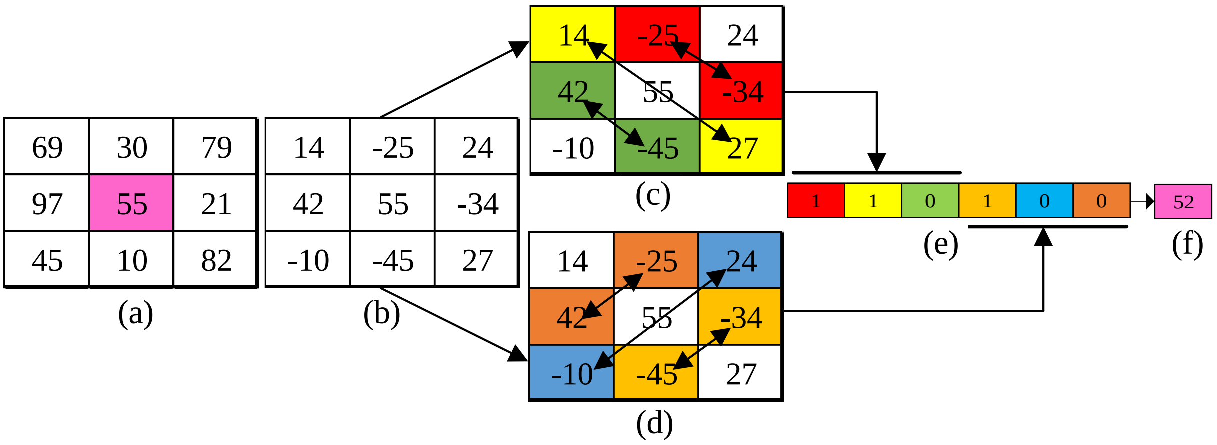


Figure 2.6 DSP calculation (a) Original Image (b) The Qi representation of (a) (c) Participants of principal diagonal pattern calculation (d) Participants of counter diagonal pattern calculation (e)Resultant diagonal symmetric pattern (f) Resultant DSP value of center pixel 55.

2.2.3 Grey Level Co-occurrence Matrix

GLCM can be considered as one of the global texture features. It belongs to the second-order statistics, hence collects information regarding a pair of pixels. It is a matrix, built using two parameters: distance from the pixel taken into consideration and the angles in degrees belonging to the set {0^0, 45^0, 90^0, 135^0}. The process to compute the GLCM is given in Algorithm 2.1.

Algorithm 2.1: *Grey Level Co-occurrence Matrix Feature Extraction*

1. Quantization is applied on a gray level image (I) to get I_q [35].

2. Calculate GLCM, which is a 2D matrix of size ((Lmax +1)×(Lmax +1)) where L_{max} is the maximum possible gray value. Creation of GLCM matrix is as follows:

   i. I_q can be considered as an input image.

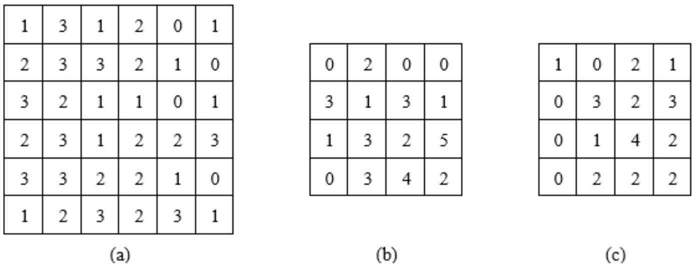
   ii. Each element of the GLCM, GLCM(i,j) can be expressed as the number of joint occurrences of intensity values i and j at a specified distance *d* and orientation θ.

3. Each element of GLCM must be divided by the summation of all elements for the normalization of the GLCM.

The whole process is given in equation 2.13.

$$GLCM_{(\Delta x,\Delta y)}(i,j)=\sum_{m=1}^{M}\sum_{n=1}^{N}1-\left[\frac{1}{2}\left(\frac{\left|I_q(m,n)-i\right|}{L_{\max}}\right)+\frac{1}{2}\left(\frac{\left|I_q(m+\Delta x,n+\Delta y)-j\right|}{L_{\max}}\right)\right] \qquad (2.13)$$

where (Δx, Δy) can be expressed as an offset in x and y directions, respectively. If the value of (Δx, Δy) is (1,0), then the above algorithm will create a GLCM having distance 1 in x-direction and 0 in y-direction which can be expressed as distance 1 and orientation 0^0. Resultant GLCM for the original image (Figure 2.7(a)) is given in Figure 2.7(b). If we have a GLCM having distance value 1 and orientation value 45^0, then the value of (Δx, Δy) will be (1,-1). The resultant matrix is given in Figure 2.7(c).



1	3	1	2	0	1
2	3	3	2	1	0
3	2	1	1	0	1
2	3	1	2	2	3
3	3	2	2	1	0
1	2	3	2	3	1

(a)

0	2	0	0
3	1	3	1
1	3	2	5
0	3	4	2

(b)

1	0	2	1
0	3	2	3
0	1	4	2
0	2	2	2

(c)

Figure 2.7 GLCM Calculation example. (a) Original Image. (b)GLCM having (Δx, Δy) as (1,0) (c) GLCM having (Δx, Δy) as (1,-1).

2.3 Bag-of-Visual-Word Features for CBIR

However, none of the strategies listed above are very effective for large image datasets. The Bag-of-Visual-Word (BoVW) [36, 37] technique is used to tackle this problem. Unlike the LBP-related feature extraction methods, BoVW uses the Scale-Invariant Feature Transform (SIFT) and speeded up robust features (SURF) methods to extract point feature descriptors. The codebook is then formed by applying a clustering method to the retrieved spatial information. Even though this strategy produces promising results in a wide range of datasets, it has the disadvantage of being extremely computationally intensive. Chen et al. [38] proposed a unique image representation approach that employs the Gaussian mixture model (GMM) to give spatial weighting to visual words and applies this method to improve CBIR performance. Firstly, the visual tokens are extracted from the image data set and then clustered into a lexicon of visual words. The spatial composition of an image is then represented as a mixture of n Gaussians, and the image is also divided into n sections. The spatial weighting method works by weighting visual words based on the likelihood of each visual word belonging to one of the n sections in the image. Distance is calculated using the cosine similarity of spatially weighted visual word vectors. Bouachir et al.[39] proposed a novel BoVW-based CBIR technique, in which an image is viewed as a vector of weights. Each of these weights corresponds to the importance of a visual word in the image and is calculated using the weighting scheme of choice. Instead of merely applying the known weighting algorithms from the text retrieval domain, this research developed a unique approach tailored exclusively for images. The proposed weighting technique is based on a fuzzy model that accounts for the basic difference between textual and visual words. In [40] Zhu et al. have proposed a novel BoVW based CBIR method that captures query-specific weights for visual words in the query image. Guo et al. [41, 42] suggested a unique CBIR methodology based on the notion, in which a halftoningbased BTC method called DDBTC was utilized to produce an image feature vector, followed by LBG clustering.

2.4 Shape Features for CBIR

Shape information often focuses on perceptually important and effective shape aspects that are either based on shape boundary information or based on border plus inside content. Moments, curvature, shape signature, signature histogram, shape matrix, spectral features, shape invariants, and other features are all designed. These shape features are compared to performances according to how well they make it possible to find related shapes in a given database [43].

Gradients [44], Fourier descriptor [45], Wavelet Fourier descriptor [45], angular pattern, and binary angular pattern [46], Convex Hull [47], etc. can be used for extracting shape information. Marr and Nishihara [48] and Braddy [49] have deeply discussed the representation and sets of criteria for the purpose of shape evaluation. Soffer and Samet [50] proposed a technique for graphical query specification. This technique enables the formulation of complicated pictorial inquiries that include spatial restrictions between query image objects that are specified symbolic query images as well as contextual constraints that specify the number of objects that should be present in the destination image. Shape features, such as eccentricity, circularity, and rectangularity, are used to represent specified symbolic query images. Folkers and Samet [51] extended this to allow queries on images with spatial extents such as rectangles, polygons, ellipses, and B-splines.

2.4.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is an efficient way of shape feature extraction [52]. As we know for the image retrieval color, texture, and shape features can be used, if we add the shape information of the image to the color and textures features, we can get a better CBIR system. The entire process to obtain the HOG features is given in Algorithm 2.2.

Algorithm 2.2: *Feature Extraction using Histogram of Oriented Gradients*

---

1. Take an image as input.
2. Compute the vertical(g_y) and horizontal(g_x) gradient over the image by using [-1,0,1] and $[-1,0,1]^T$ filters respectively.
3. Obtain magnitude(M) and direction(θ) for each pixel by using the equations 2.14 and 2.15.

4. Consider a cell size from this image as shown in Figure 2.8
5. Obtain the feature vector by considering M and θ of a cell using equations 2.16 and 2.17.
6. Choose a block size from the original image as shown in Figure 2.8
7. Obtain the feature vector for each cell of block considered.
8. Concatenate the feature vectors of the whole block.
9. Normalize the feature vector obtained in step-8.
10. Repeat Step-5 to Step-9 for each block.
11. By considering the stride of 1, do the same process for all the blocks in the image.
12. Concatenate all these features vectors to get the feature vector for the whole image.

$$M(i,j) = \sqrt{{g_x}^2 + {g_y}^2}$$

(2.14)

$$\theta(i,j) = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

(2.15)

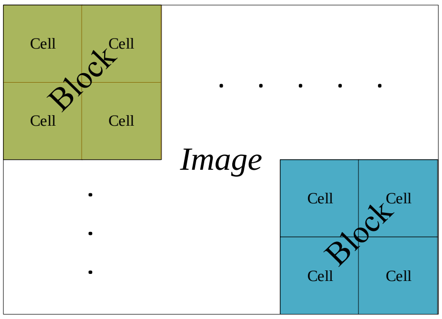


Figure 2.8 Representation of Cell, Block for HOG features calculation. The cell size is 8×8 pixels. One block is constructed using 2×2 cells

$$H\left(\left\lceil \frac{\theta(i,j)}{20} \right\rceil \bmod 9 \times 20\right) = H\left(\left\lceil \frac{\theta(i,j)}{20} \right\rceil \bmod 9 \times 20\right) + \left(M(i,j) \times f_1\right) \qquad (2.16)$$

$$H\left(\left\lfloor \frac{\theta(i,j)}{20} \right\rfloor \bmod 9 \times 20\right) = H\left(\left\lfloor \frac{\theta(i,j)}{20} \right\rfloor \bmod 9 \times 20\right) + \left(M(i,j) \times f_2\right) \qquad (2.17)$$

where f_1=(θ(i,j) mod 20) / 20 and f_2=1- f_1.

The process of calculating the HOG for an image, shown in Figure 2.9, of size 128×64, is explained in this section. As the first step, the cell size is considered as 8×8, the contents of this are shown in Figure 2.10. The result of magnitude (M) and direction (θ) for each pixel in Figure 2.10 is shown in Figure 2.11. Now, we compare the respective magnitude (M) and direction (θ) to get a 1×9 vector (A). The vector A is indexed as 0^0, 20^0, 40^0,..., 160^0. While selecting an index from the vector, the value of the direction cell is taken into account. After the selection of an index, its value is selected from the corresponding magnitude cell. Consider the value in the red circle of the direction cell which is shown in Figure 2.11. As the value 135.95 is not having its place in the vector A, rather it lies between indices 120 and 140. Therefore, the corresponding value is taken from magnitude cell which is 332.12 and split in between 120 and 140 i.e., 67.25 in location 120 and 264.86 in location 140 of vector A. Similarly, considering the value in the blue circle of the direction cell, the value in the magnitude cell 236.35, split in between 0 and 20 i.e., 91.23 in location 0 and 145.11 in location 20 of vector A. This process continues by updating the location values and we get the final result as shown in Figure 2.12.

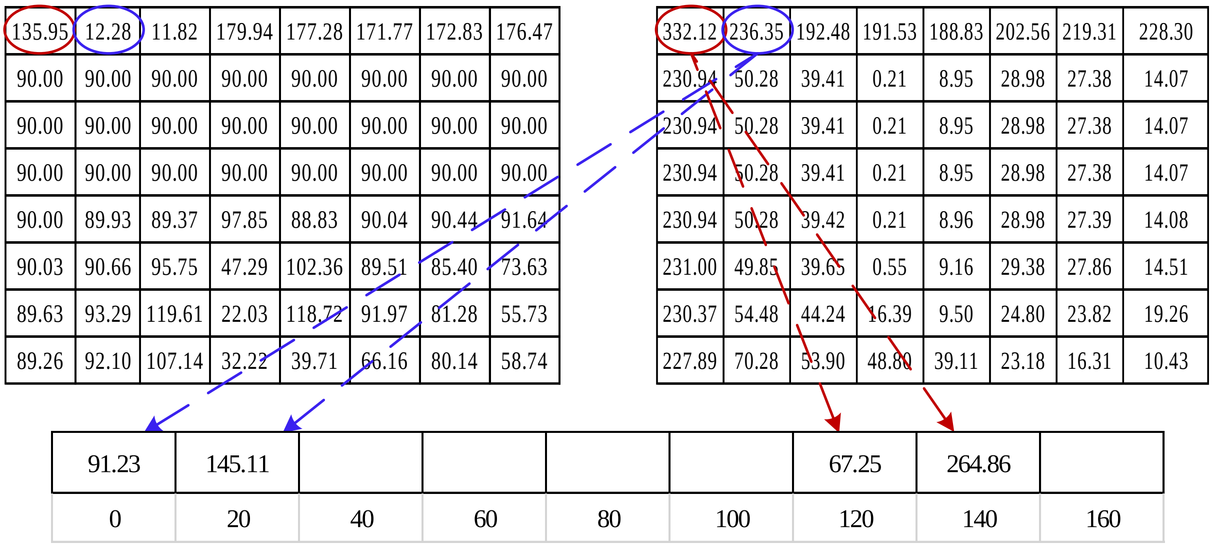
Normalization of the feature vector for a block:

The normalization process is done by considering the values in the vector. For example, if the vector is of size 1×4: [4, 6, 11, 9]. To normalize it we need to calculate the l2- norm for this vector, which is $\sqrt{4^2 + 6^2 + 11^2 + 9^2} = 15.93$. Then divide each of the vector values with this l2-norm: $[\frac{4}{15.93}, \frac{6}{15.93}, \frac{11}{15.93}, \frac{9}{15.93}]$ i.e. equal to [0.25, 0.37, 0.69, 0.56]. This process when applied on the block of size 2×2 cells results in a normalized feature vector of size 1×36 vector. For an image of size 128×64, when considered the cell size as 8×8 and block size as 2×2 with stride 1, results in 7×15 = 105 blocks. As mentioned before one block will have a 1×36 size vector, for 105 blocks, the resultant feature vector size will be 105×36 = 3780.

Figure 2.9 A sample image. The image is cropped to the size 128×64.

239	231	188	192	189	200	218	228
239	231	188	192	189	200	218	228
239	231	188	192	189	200	218	228
239	231	188	192	189	200	218	228
239	231	188	192	189	200	218	228
239	231	188	192	189	200	218	228
239	230	184	192	191	200	216	224
237	228	167	177	193	201	214	217

Figure 2.10. A cell. The cell is of size 8×8 pixel taken from the sample image.



135.95	12.28	11.82	179.94	177.28	171.77	172.83	176.47
90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00
90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00
90.00	90.00	90.00	90.00	90.00	90.00	90.00	90.00
90.00	89.93	89.37	97.85	88.83	90.04	90.44	91.64
90.03	90.66	95.75	47.29	102.36	89.51	85.40	73.63
89.63	93.29	119.61	22.03	118.72	91.97	81.28	55.73
89.26	92.10	107.14	32.22	39.71	66.16	80.14	58.74

332.12	236.35	192.48	191.53	188.83	202.56	219.31	228.30
230.94	50.28	39.41	0.21	8.95	28.98	27.38	14.07
230.94	50.28	39.41	0.21	8.95	28.98	27.38	14.07
230.94	50.28	39.41	0.21	8.95	28.98	27.38	14.07
230.94	50.28	39.42	0.21	8.96	28.98	27.39	14.08
231.00	49.85	39.65	0.55	9.16	29.38	27.86	14.51
230.37	54.48	44.24	16.39	9.50	24.80	23.82	19.26
227.89	70.28	53.90	48.80	39.11	23.18	16.31	10.43

91.23	145.11					67.25	264.86	
0	20	40	60	80	100	120	140	160

Figure 2.11 Process of updating the cell's feature vector. The first and second matrix represent the direction and magnitude of the cell respectively.

971.93	293.13	75.15	45.78	1336.34	1351.52	139.94	264.78	228.56

Figure 2.12 The resultant feature vector.

2.5 CNN Methods for CBIR

Convolution Neural Network (CNN) allows complicated design architectures to be programmed in the same way that a human brain is, and to interpret data at several stages of transformation and representation. The success of CNN based approaches explores a promising solution for various computer vision fields like classification and retrieval. CNN uses convolution layers to extract hierarchical feature representation [53]. The output of each layer is known as a feature map, obtained by convolving weight matrix over receptive fields. The weight matrix or the filter is vital in identifying features. Apart from the convolution layers, a pooling layer is used, conventionally after every convolutional layer. It downsamples the feature map by applying techniques such as max-pool, min-pool and L2-norm to yield more robust feature maps [54, 55]. Another layer is added to introduce non-linearity in the system [54, 55]. This can be done by the ReLu layer which simply changes all negative activations to zero. This gives substantial speedup as opposed to tanh and sigmoid nonlinear functions. In the end, fully connected layers are placed which maps the input from the preceding layer to an N-dimensional vector. N denotes the number of classes in the dataset. This vector contains the probability of the classes present. Different CNN models are used in the field of CBIR. Various Pre-trained CNN like AlexNet [56], VGG[57], GoogleNet[58], ResNet[59], Inceptionv3[60], Xception[61], SqueezeNet[62], MobileNetv2[63] and Darknet-53[64] can be used in CBIR.

AlexNet: AlexNet secured the first position in ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2012. It is mainly used for image classification problems. ImageNet dataset was used to train AlexNet. The total number of parameters and neurons of AlexNet is 60 million and 6,50,000 respectively. Five to six days were required to train AlexNet on the ImageNet dataset using two GTX 580 3GB GPUs.

VGG: VGG Net achieved an error rate of 7.3% in ILSVRC 2014. VGG has two popular implementations using 16 and 19 layers. When compared to AlexNet, the filter size is reduced here so that the number of parameters can also be reduced. ReLu layer is used after each convolution layer to make the system non-linear. It was trained for three weeks on four NVIDIA Titan Black GPUs.

GoogleNet: This was proposed in 2015 by Szegedy et al [58]. In this CNN model, the inception concept was introduced. The details of this CNN model and how it is used in CBIR are discussed in Chapter 4.

ResNet: Residual Network being a 152-layered network architecture gives a remarkable performance in classification, localization, and detection problems. ResNet won ILSVRC 2015 with 3.6% error rate. An 8 GB GPU machine was used for 14 to 21 days to train ResNet. The degradation problem is resolved with the help of a deep residual learning framework. There are different variations of ResNet available having the number of layers 18, 34, 50, 101 and 152.

Inceptionv3: This is a 42-layer deep network trained on the ImageNet dataset. Several techniques are employed in this architecture to improve the performance. Some of them are factorized convolutions, smaller receptive fields, grid size reduction and symmetric convolutions. For training, a distributed system was employed using an NVIDIA Kepler GPU for 100 epochs with batch size of 32.

Xception: This is a 71-layer deep network also trained on the ImageNet dataset. It can classify up to 1000 object categories. Like ResNet, this also follows a directed acyclic graph structure. The feature extraction module is formed by 36 convolution layers. It was trained on 60 NVIDIA K80 GPUs. Xception was trained on the ImageNet dataset.

SqueezeNet: SqueezeNet is an 18-layer deep network which has reduced the model size considerably without affecting the accuracy. SqueezeNet uses fire modules which squeeze and expand networks. The model also comes with simple bypass and complex bypass variation. Due to the smaller size of the CNN, it was able to be trained on FPGAs.

MobileNetv2: It is a light weight model with 53 layers. It was depthwise convolution and pointwise convolution. It showed good performance in classification, detection, and segmentation tasks. It was trained on 16 GPU asynchronous workers.

But among all these pre-trained CNN models DarkNet-53 [64] is showing the best accuracy in the field of CBIR.

2.5.1 DarkNet-53

Darknet-53, introduced in the year 2018, is used as the basic feature extraction module in real-time object detection network YOLOv3[64]. The purpose of Darknet-53 is the characteristic extraction of input images [65, 66]. The basic structure of Darknet-53 is shown in Table 2.1. The key idea of this network can be represented as the combination of the basic feature extraction framework of YOLOv2[66] and Residual module [67]. The smallest component of Darknet-53 consists of successive convolution layers of kernel size 1×1 and 3×3. Batch Normalization (BN) layer [68] and the LeakyRelu layer are used after every convolution layer. The basic structure of Darknet-53 contains five repetitive modules where the smallest component is repeated one, two, eight, eight, and four times respectively. The function of different layers is discussed below:

1) Convolution Layers: Output of the convolution layer is calculated using equation 2.18:

$$F_j^L = \sum_{i \in FV_j} F_i^{L-1} * C_{ij}^L + B_j^L \tag{2.18}$$

Where FV_j represents the feature vector of the input image. ‘ C_{ij}^L ’ is the i^{th} element of the j^{th} convolution kernel in layer L.A*B represents the convolution operation between A and B. F_j^L represents j^{th} feature map in layer L which is generated by convolution operation performed between the input image and different convolution kernels.

2)Batch Normalization (BN) Layer: This layer is used for normalizing the output to the same distribution based on the eigenvalues of the same batch. This layer is used after the convolutional layer to speed up network convergence and to avoid the over-fitting problem. The output calculation of the BN layer can be expressed using equation 2.19

$$F_{out} = \frac{s\left(F_{conv} - \mu\right)}{\sqrt{\partial^2 + cons}} + d \qquad (2.19)$$

where *'s'* is the scaling factor, *'μ'* is the mean of all input values, '∂' is the variance of all inputs, *'cons'* represents a constant value, and *'d'* is the offset. *'F_{conv}'* is the output of the convolution layer and *'F_{out}'* is the final output value of the BN layer.

3) *LeakyRelu (LR) Layer:* In the layer, an activation function is used which is a combination of a linear unit with leakage correction. This layer is used to increase the nonlinearity of the network. This activation function is given in equation 2.20

$$Y_i = \begin{cases} F_i & if \quad F_i \geq 0 \\ \dfrac{F_i}{p_i} & else \end{cases} \qquad (2.20)$$

Where *'F_i'* is the input value, *'Y_i'* is the output of the activation function and *'p_i'* is a fixed parameter that lies in the interval (1, +∞)

Table 2.1. Darknet-53 architecture. Total 5 repetitive blocks are there. Each contains two convolution layers and one Residual layer.

# Repeatation	Type	# of filters	Size/Stride	Output Size
1	Convolution	32	3×3/1	256×256
1	Convolution	64	3×3/2	128×128
1	Convolution	32	1×1/1	
	Convolution	64	3×3/1	
	Residual			128×128
1	Convolution	128	3×3/2	64×64
2	Convolution	64	1×1/1	
	Convolution	128	3×3/1	
	Residual			64×64
1	Convolution	256	3×3/2	32×32
8	Convolution	128	1×1/1	
	Convolution	256	3×3/1	
	Residual			32×32
1	Convolution	512	3×3/2	16×16
8	Convolution	256	1×1/1	
	Convolution	512	3×3/1	
	Residual			16×16
1	Convolution	1024	3×3/2	8×8
4	Convolution	512	1×1/1	
	Convolution	1024	3×3/1	
	Residual			8×8
	Avg_Pool			1×1
	Connected		1000	
	Softmax			

2.5.2 CBIR using CNN Approaches

Many literatures were proposed where CNN based features are used in the CBIR framework and their performance is compared with that of handcraft features [69, 70]. Sezavar et al. [71] proposed a resilient CBIR method that improves CBIR performance by combining a convolutional neural network and sparse representation. Image features are extracted using CNN and sparse representation in this method is used to improve retrieval speed and accuracy. As a pre-trained CNN model, AlexNet is considered to extract high level image features. After feature extraction, sparse representation is employed to obtain the relevant images, rather than calculating the Euclidian distance between the query feature vector and all feature vectors of the related image dataset, used by most of the CBIR methods. This reduces the retrieval time of images efficiently. Saritha et al. [72] proposed a new CBIR method where a deep belief network (DBN) method of deep learning is used to extract the features and classification. One new CBIR framework was proposed by Mustafic et al. [73] where the deep CNN model and transfer learning method are used to train the similarity measures between the query image and dataset images. VGG19 deep neural network is used to train distance function. Ramanjaneyulu et al. [74] proposed a CNN based CBIR method where feature extracted from VGG-16 is used to retrieve relevant images. Messina et al [75] proposed a novel CBIR method known as relational content-based image retrieval (R-CBIR), in which a two-stage relation network module (2S-RN) is used. This network is trained on the R-VQA task for extracting non-aggregated visual features. After that authors proposed aggregated visual features relation network (AVF-RN), which extracts more efficient features as it can learn aggregation inside the network. One novel Semantic-Based Image Retrieval was proposed by Song et al. [76] where two different loss functions are used to give more importance to the spatial distribution of image features. The first loss function is used to combine two types of loss: softmax and center loss. The purpose of this loss function is to make the feature vector more efficient so that it can separate images with different content and can identify the same class images. The second loss function can be expressed as a modified centre loss that simultaneously deals with distance between extracted CNN features and feature center of the same class & far distance between image feature and different class feature centres. Zheng et al. [77] presented a survey work based on CBIR methods that use SIFT and CNN features individually and as a combination of both features. Swati et al. [78] proposed one new CBIR framework where the VGG19-based image feature vector is used for image retrieval and closed-form metric learning is used as a distance function. In this literature, the authors proposed a block-wise fine-tuning strategy for transfer learning to improve the performance of their proposed method. Cai et al.

[79] proposed a new Medical Image Retrieval framework based on CNN and hash coding. In this method, the Siamese network is used and in each branch of this network, to extract features, CNN followed by a hash function, is used. Zheng et al. [80] proposed an innovative CBIR method that has two streams that manage two different tasks simultaneously. The mainstream extracts distinct visual features that mainly concentrate on semantic attributes of an image. On the other side, the auxiliary stream is used to assist the mainstream to consider salient image content for more efficient image feature extraction. One new CNN-based CBIR model was proposed by Bhandi et al. [81] where features extracted from VGG-16 and ResNet-50 are combined form the final feature vector. Ozaydın et al. [82] proposed a new feature vector for retrieving images by combining features extracted from VGG-16 and SIFT. Tzelepi et al. [83] proposed one new deep learning based image retrieval method where BVLC Reference CaffeNet is used for feature extraction. Rao et al. [84] proposed a new relevance feedback CBIR framework where authors have used a pseudo-label strategy and an improved active learning selection method.

Desai et al. [85] introduced a CBIR structure where a pre-trained CNN model, VGG-16, is used for feature extraction, and SVM is used for image classification. Simran et al. [86] proposed a CBIR structure where Deep CNN models are used to extract features, along with Principal Component Analysis (PCA) as part of feature reduction. In [87] Dubey explores a comparative study between traditional and CNN-based methods for CBIR. This research categorizes various state-of-art CBIR methods and the performance of those methods is also compared. Rao et al. [88] introduced an efficient CBIR model where instead of one CNN model, two CNNs, VGG-16 and VGG-19, are used for feature extraction. A Hash function-based CBIR method is proposed by Christy et al. in [89]. Tagging of images is performed using CNN models to identify the objects in an easy and efficient manner.

2.5.3 Feature Fusion based CBIR Methods

Yet, in the CNN module, it is not guaranteed that the features from the highest level can always achieve the best performance. To overcome this problem, effective feature fusion methods need to be proposed. In [90], Keisham et al. [91] proposed a Deep Search and Rescue-based CBIR model. Noise removal as a pre-processing step, multiple feature extraction methods, weighed feature fusion, and clustering using optimization algorithms are all included in this model. In a Region of Attention (ROA) based CBIR approach is proposed by Pradhan et. al where color and texture features from ROA portion and Background information from the non-ROA region are fused to calculate the final feature vector. Color Correlation histogram and DWT are used to extract color and multi-directional texture features of ROA image on the other hand Dissimilarity feature mapping is used to gather background information from the non-ROA region of the image. In this method VGG-16 is used as a classifier. Alrahhal et al. [92] proposed the WNAHVF feature fusion-based CBIR framework, which uses a combination of AlexNet and Handcrafted Visual Features to extract features from the query image. Bag of Features (BOF) and Local Neighbor Pattern (LNP) are used to produce a handcrafted image. These three features are normalized using Euclidean Norm [93] after feature extraction. Following normalization, a weighted concatenation is carried out, with AlexNet, BOF, and LNP features assigned weights of 1,0.3, and 0.1, respectively. Li et al. [94] proposed an unsupervised subspace-based multi-view fusion method for CBIR. In this paper, multi-view fusion refers to combining multi-view features for effective image representation. The multi-view features come from multiple visual features. Since these features are used to describe a single image from different visual perspectives, they are treated as multi-view visual representations of the image. Yan et al. [95] have proposed an image representation model, complementary CNN, and SIFT (CCS), to fuse CNN and SIFT in a multi-level and complementary way. As CNN features GoogleNet is used in this work. The output of the Pooling layer of GoogleNet which is of length 1024 is used in this paper. These features are fused with SIFT features to create the final feature vector of the proposed CCS model. Devulapalli et al. [96] proposed an efficient CBIR method by combining handcrafted features and deep features. In this paper, the pre-trained GoogleNet is used to extract deep learning features. The image was decomposed into coarse and detailed scales using the curvelet transform. Haralick features were calculated in four directions from detail coefficients and combined with CNN features to calculate the final feature vector. Liu et al. [97] proposed one innovative CBIR framework which combines both high- and low-level features for image retrieval. Features extracted from pre-trained GoogleNet is used as high-level features. Authors have proposed two-level codebooks in the DDBTC feature extraction method to reduce feature vector length and used the new feature as a low-level feature vector.

2.5.4 Modified CNN based CBIR Methods

Though CNN models are giving Good retrieval accuracy, their still scope for improvement in CNN architecture. In recent day research, Modified versions of CNN are proposed. Sakarani proposed a modified version of AlexNet in [98]. In this proposed CNN structure authors have used the LeakyReLU layer as the activation layer instead of the ReLu layer used in traditional

AlexNet. In addition, the authors replaced the Fully-Connected (FC) layer of size 1×512 in place of the FC layer of 1 × 4096 which significantly reduces the number of parameters and the overall training time of the proposed CNN. In [99] Hussian et al. proposed a modified CNN model for CBIR, the MaxNet model, which is formed by stacking the revised inception module hierarchically. After each inception, features extracted from various pipelines in the inception module are combined to maximize feature values. A 4-dimensional image is used as input in the proposed CNN. All RGB input images are converted to grey-level images, which are then added to the original image to obtain a 4-dimensional image. Putzu et al. [100] proposed a novel CBIR system based on transfer learning from a CNN trained on a large image database. Then, to narrow the semantic gap, the pre-trained CNN is further improved by utilizing Relevance Feedback (RF) provided by the user. Following user feedback, the authors modified and then re-trained the CNN using the labelled collection of relevant and non-related images. The CNN model which is proposed based on this strategy is named as RFNet. The study also includes various ways for utilizing the updated CNN to return a new set of images that are expected to be relevant to the user's needs.

Chapter 3

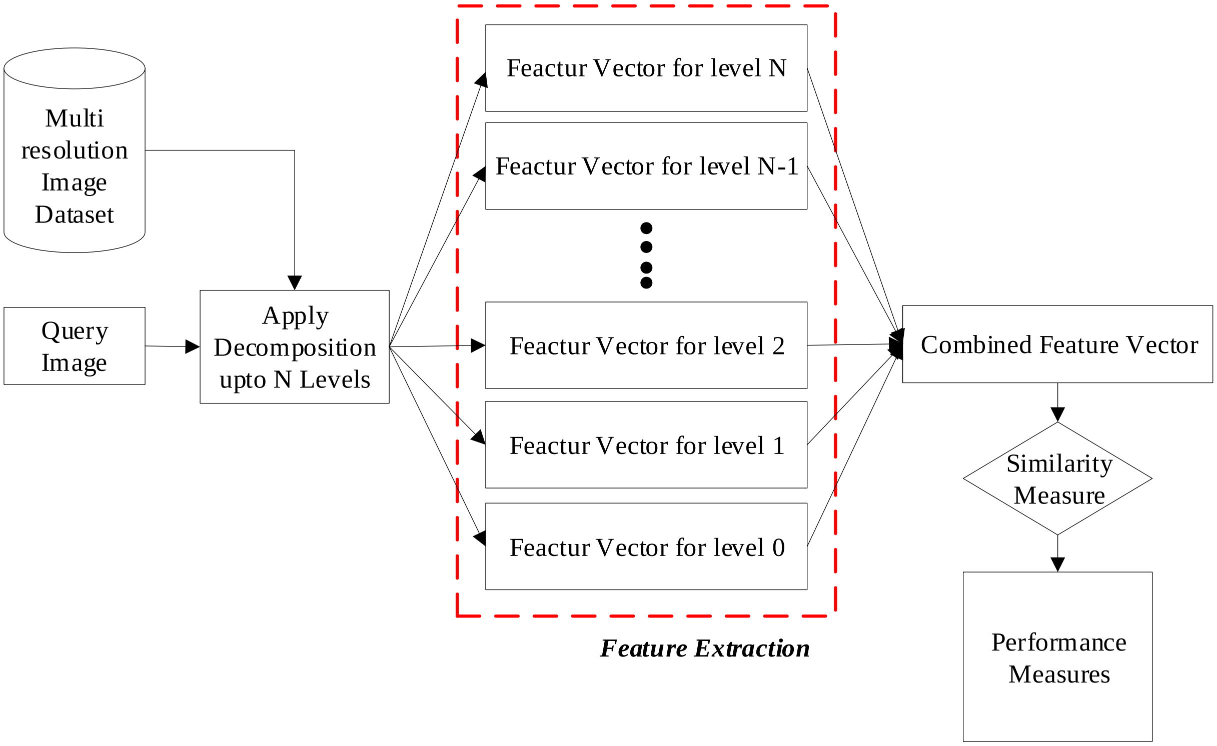
A Resolution Independent Feature Extraction Method for Content Based Image Retrieval

In this chapter, a novel content based image retrieval method is proposed, which is effective for both single and multiresolution datasets. Most of the methods proposed for multiresolution image retrieval apply a decomposition technique up to the same level for all images in the dataset irrespective of their size, which causes a significant loss of information. In this chapter, the Haar wavelet transform is applied to the input images as a decomposition method, but the level is determined based on the size of the image. To extract features, the proposed CBIR approach uses both color and texture features. For color feature, inter-channel voting between hue, saturation, and intensity component of an image is used, as the inter-relationship between color and intensity component is independent of the dimension of the image. Gray Level Co-occurrence Matrix (GLCM) on Diagonally Symmetric Pattern is computed to get the texture features of an image.

3.1 Framework for Multiresolution CBIR

This chapter proposed a CBIR framework that follows the traditional structure of multiresolution CBIR. An illustrative representation of such a CBIR framework is shown in Figure 3.1. It contains the following steps:

1. The decomposition technique is applied to the input image and all images of the image dataset for multiresolution analysis of the image.
2. Compute the feature vector of the query image and all images of the image dataset at each level of decomposition.
3. Combine all features obtained from different levels of resolution to get the final feature vector.
4. Compare the feature vector of the query image to all the images of the dataset. This comparison can be made using a distance measure.
5. Performance measures are calculated based on the similarity between the query image and dataset images to evaluate the proposed method.


Multi resolution Image Dataset
Query Image
Apply Decomposition upto N Levels
Feactur Vector for level N
Feactur Vector for level N-1
Feactur Vector for level 2
Feactur Vector for level 1
Feactur Vector for level 0
Feature Extraction
Combined Feature Vector
Similarity Measure
Performance Measures

. Figure 3.1 General Framework for Multiresolution CBIR.

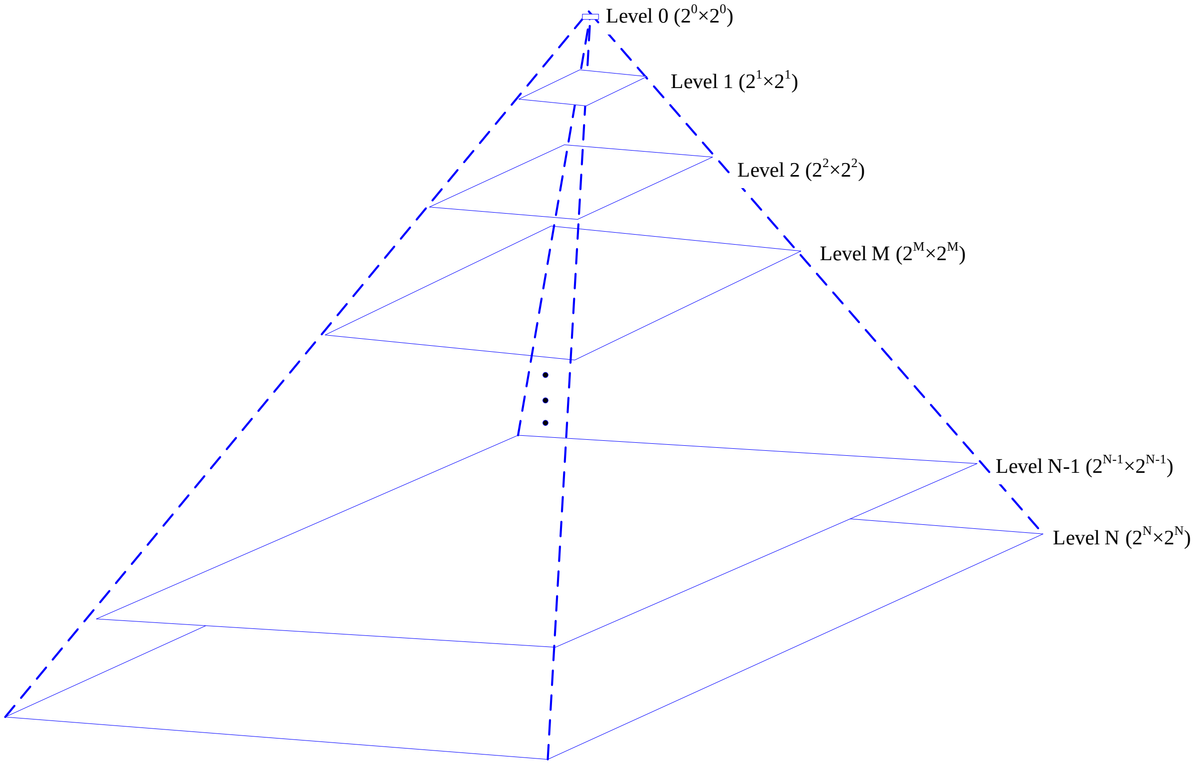
3.2 Decomposition Methods

The framework in Figure 3.1 suggests the first step of multiresolution CBIR is to apply the decomposition technique on both query images and all images of the image dataset. The decomposition technique is applied to images to obtain a different level of resolution of an image, and processing is done on each resolution for efficient multiresolution analysis of that image. The decomposition of an image into multiple-scale can be done using two methods:1. Image Pyramid and 2. Wavelet Transform.

3.2.1 Image Pyramid

An image Pyramid can be defined as a powerful but simple structural representation of an image at more than one resolution [101]. The structural representation of an image pyramid is shown in Figure 3.2. The bottom-most level of an image pyramid is nothing but the highest resolution representation of an image, whereas the topmost level is the lowest resolution representation of an image. Moving from base to apex of an image pyramid a decreasing trend of an image in terms of resolution is observed. If the original size has a size of $2^N \times 2^N$ then a fully populated image pyramid will have a total N+1 level, where the base is considered level N containing an image with a size of $2^N \times 2^N$ and the apex is level 0 containing a single-pixel ($2^0 \times 2^0$).

Any of the intermediate level *n* contains *n* level approximation of output image having size $2^n \times 2^n$, where $0 \leq n \leq N$. But as images having very few pixels do not contain much information, most of the image pyramid reduce their size to the M+1 level, where the base includes an image of size $2^N \times 2^N$ and the apex comprises an image of size $2^{N-M} \times 2^{N-M}$ and 0<M<N.

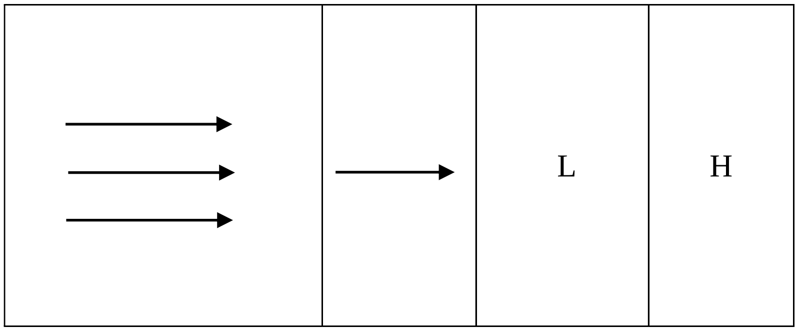


Level 0 ($2^0 \times 2^0$)
Level 1 ($2^1 \times 2^1$)
Level 2 ($2^2 \times 2^2$)
Level M ($2^M \times 2^M$)
Level N-1 ($2^{N-1} \times 2^{N-1}$)
Level N ($2^N \times 2^N$)

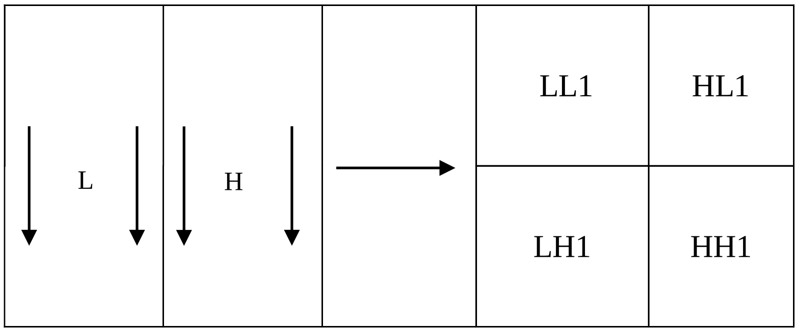
Figure 3.2 The Pyramidal Structure of an Image.

3.2.2 Wavelet Transforms

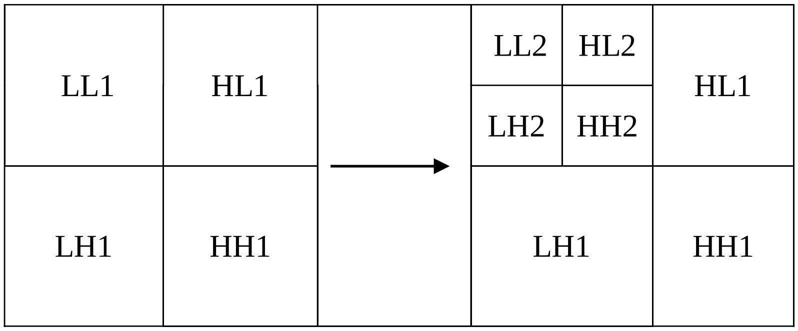
In the present day's scenario, wavelets play a vital role in various fields of application like Astronomy, Magnetic Resonance Imaging, Subband Coding, Signal and Image Processing, Computer and Human Vision, Earthquake Prediction, Optics, Radar, Music, Acoustics, Turbulence, Nuclear Engineering, Data Mining, Neurophysiology and various Mathematical operation like Partial Differential Equations (PDE), etc. In wavelet transform [102,103,104], an analysis filter bank is used to analyze an image signal. At each level of decomposition, the analysis filter contains one low pass filter and one high pass filter. While passing through these filters, an image single is divided into two bands. Among them, a low pass filter is commensurate to an averaging operation and is used to extract coarse information from an image signal. At the same time, a high pass filter is commensurate to a differencing operation and is used for extracting detailed features of an image signal [102, 105, 106]. A 2D wavelet transform can be achieved by applying two 1D wavelets transform separately. At first low pass and high pass filters are applied to the image along *x*-dimension. Coefficients obtained from the low pass filter are kept on the left portion of a matrix (known as the L part), whereas coefficients obtained from the high pass filter are kept on the right portion of a matrix (known as the H part) as shown in Figure 3.3(a). After that same low and high pass, filters are applied along the y-dimension of each sub-image (L and H part) which decompose the original image into four components after 1st level of wavelet decomposition, named as Low-Low (LL) band, High-Low (HL) band, Low-High (LH) band and High-High (HH) band as shown in Figure 3.3(b). The result of the 2nd level of wavelet decomposition is shown in Figure 3.3(c), where analyzing filter is applied only on the LL band of 1st level decomposition. This process will be repeated for a further level of decomposition. This kind of decomposition is known as the 'Pyramidal decomposition' of an image. The result of the 2nd level of decomposition of the Gray level Lena image is shown in Figure 3.4. The same process can be applied to all the three channels of RGB color to get the colored version of the Wavelet transform, shown in Figure 3.5. The different types of wavelets are Haar [105, 106, 107], Daubechies[108], Coiflet, Symlet, Morlet, Meyar, Gaussian, Shan, etc.



(a)

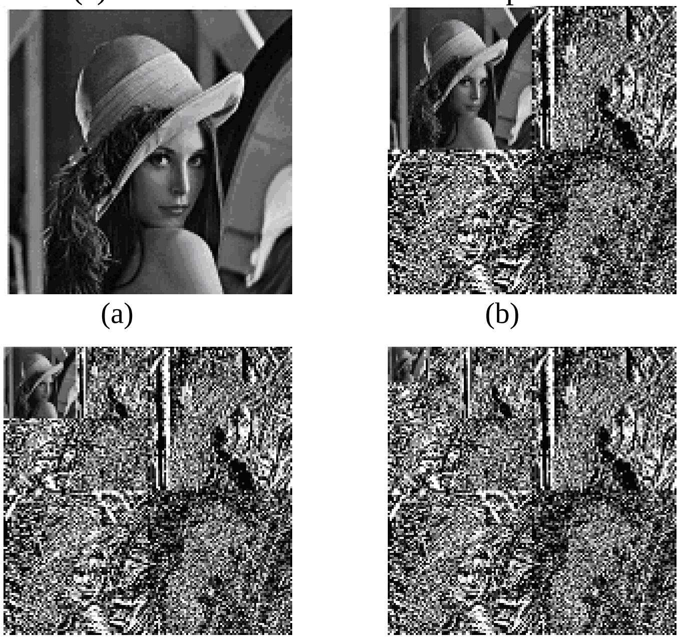


(b)



(c)

Figure 3.3 (a) Horizontal Wavelet Decomposition. (b) Vertical Wavelet Decomposition. (c) Second Level Wavelet Decomposition.



(a) (b)

(c) (d)

Figure 3.4 (a) Original Gray Lena Image. (b) Level-1 Gray Wavelet Transformed Lena Image. (c) Level-2 Gray Wavelet Transformed Lena Image. (d) Level-3 Gray Wavelet Transformed Lena Image.

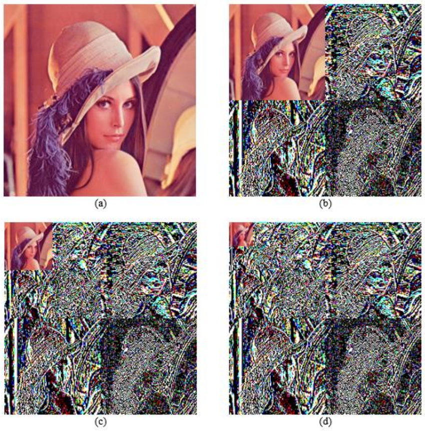


Figure 3.5 (a) Original Color Lena Image (b) Level-1 Color Wavelet Transformed Lena Image. (c) Level-2 Color Wavelet Transformed Lena Image. (d) Level-3 Color Wavelet Transformed Lena Image.

Haar Wavelet Transform

The first known wavelet transform is the Haar wavelet transform, proposed by Alfred Haar in 1909. Haar has used *averaging and differencing* method for the forward wavelet transform and the *summing and differencing* method for inverse transformation. The whole process of forward transformation can be obtained by using equation 3.1.

$$N = W^T \times M \times W \qquad (3.1)$$

where M is the original matrix, 'W' is the transformation matrix and 'N' is the wavelet transformed matrix. Similarly, the inverse transformation can be obtained by using equation 3.2

$$M = \left(W^T\right)^{-1} \times N \times W^{-1} \qquad (3.2)$$

For example, if the given image is of size 8×8, the transformation matrix (W) is defined as given below.

$$\begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

For $n \times n$ images, the transformation matrix can be obtained accordingly.

The above-mentioned Wavelet transforms are known as a linear wavelet transform. However, nonlinear extensions on wavelet transforms are also possible [109, 110, 111]. Sweldens proposed a novel tool, the lifting scheme, which is very useful for constructing nonlinear wavelet transform [112, 113, 114]. The different nonlinear wavelet transforms (i) Morphological Haar Erosion Wavelet (ii) Morphological Haar Dilation Wavelet (MHDW) (iii) Morphological Binary Wavelet (MBW).

3.3 Methodology

All the existing multiresolution retrieval methods apply decomposition techniques up to the same level on all the images of a multiresolution dataset. As the multiresolution image dataset contains images with different dimensions, applying the decomposition technique on images irrespective of their size results in a substantial loss of information. In this work, to reduce information loss, the Haar wavelet transforms on the input images are applied for N times, resulting N+1 level of decomposition of an input image, where N is defined as equation 3.3. N is named **'Level Count,'** which depends on the ratio between the size of the input image and the minimum image size in the image dataset. If the value of N is more significant than zero, then the LL component of the image, obtained after applying the Haar wavelet on one level, can be considered the input image for the next level.

$$N = f(K) = \begin{cases} 0, & K < 0 \\ K, & K \geq 0 \end{cases} \tag{3.3}$$

where,

$$K = \left\lfloor \log_2 \sqrt{\frac{size \quad of \quad input \quad image}{\min imum \quad image \quad size \quad of \quad dataset}} \right\rfloor - 1$$

In multiresolution datasets, the size of the images is not the same for all the images, so the features which depend on the image size are not suitable. Inter-relationship among the different color channels can be considered dimension independent characteristics of an image. To explore the inter-relationship between color and intensity components, inter-channel voting between hue, saturation, and intensity components [24], is used as a color feature. To reduce feature vector length, Hue, Saturation, and Intensity channels are quantized into 72, 20, and 32 bins respectively. As symmetricity among diagonally located pixel pairs of an n×n image window can be considered a resolution independent texture feature, DSP is used to extract texture features [23]. To get the spatial relation among pixel values in the DSP matrix, GLCM having distance parameter 1 and direction parameter 0^o is constructed from the DSP matrix. To reduce the feature vector length, we quantize it into 16 bins, resulting in a 16×16 matrix. The final feature vector is the concatenation of color and texture features followed by L2 normalization. This feature extraction method is applied on each level of decomposition of the input image, resulting in total N+1 feature vectors. The final feature vector is calculated by a weighted average of these N+1 feature vectors. The entire process is given in Algorithm-3.1, shown in Figure 3.6.

***Algorithm-3.1**: Algorithm for the Proposed Resolution Independent CBIR.*

---

1. Take an RGB image as input.
2. Apply Haar wavelet on input image N times, resulting in N+1 level of decomposition of the input image, where *N* is defined in equation 3.3.
3. At each level of decomposition, apply Step 4 to Step 10 on the LL component of the previous level image.
4. Convert RGB image to HSI image.
5. Apply inter channel voting between the Hue, Saturation, and Intensity component of the image.
6. Apply DSP on the Intensity part of the HSI color space.
7. Quantize DSP into 16 levels on which Grey Level Co-occurrence Matrix is computed.
8. Convert 16×16 GLCM into a vector form.

9. The final feature vector of the respective level of decomposition is the concatenation of vectors obtained from Step5 and Step 8.
10. L2 norm is applied to the concatenated feature vector obtained in Step 9.
11. The final feature vector of the input image is calculated using equation 3.4.

$$FV = \sum_{i=1}^{N+1} \frac{2 \times i}{(N+1)(N+2)} \times FV_i \qquad (3.4)$$

where FV is the final feature vector of the input image and FVi is the feature vector of the ith level of decomposition.

12. Use similarity measures to compare the features of the query image (input image) to all the images in the dataset.
13. Construct RANK MATRIX based on results obtained in Step 12.
14. All performance measures are calculated based on the rank matrix.

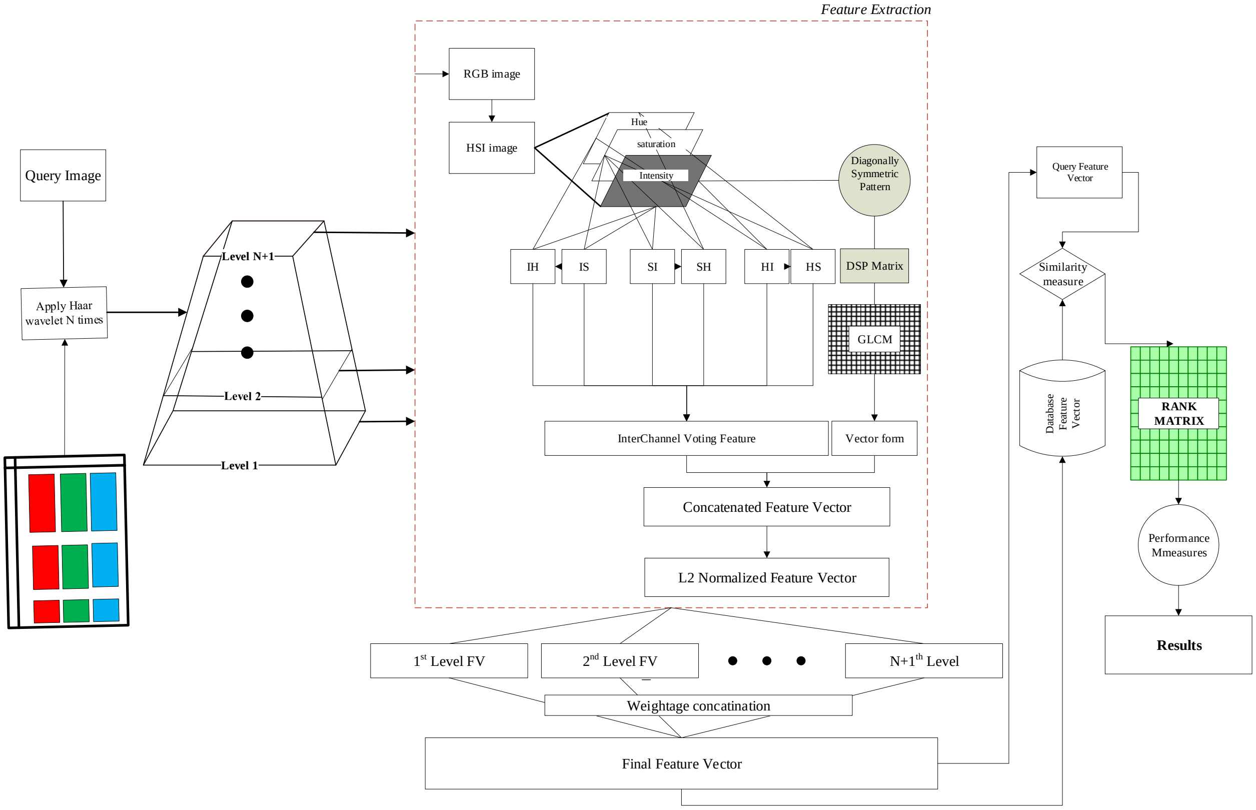


Figure 3.6 Block Diagram of the Proposed Resolution Independent CBIR System.

3.4 Experimental Results and Discussions

To evaluate the performance of the proposed method, *six* benchmark image datasets of color images and their multiresolution version have been used. Results obtained from the proposed method have been compared with nine existing techniques shown in Table 3.1. Precision, Recall, ANMRR, TMRE, and F-Measure are used for performance comparison.

Table 3.1 Different Methods used for Evaluating the Performance of the Proposed Resolution Independent CBIR System.

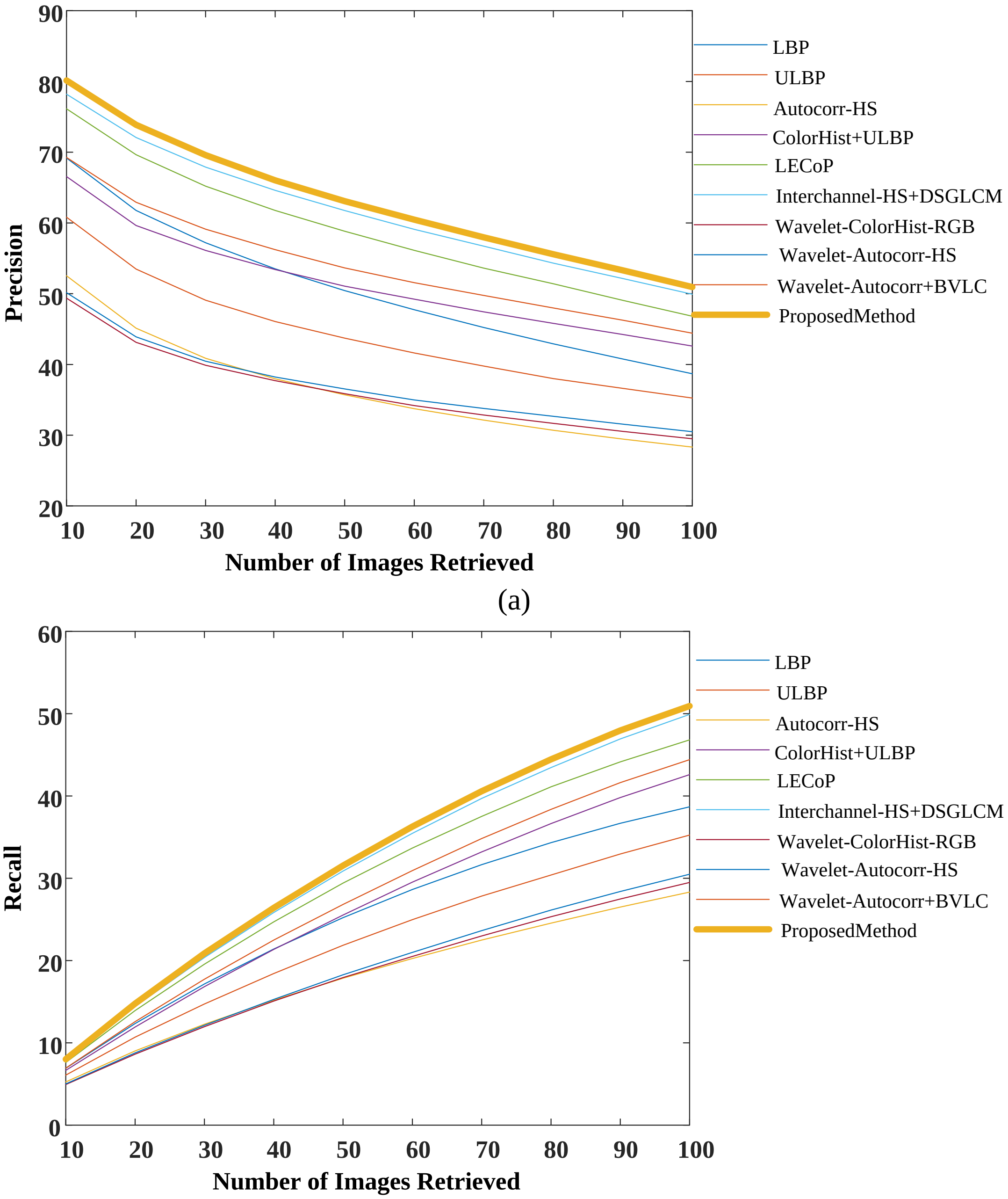
‘Texture Only’ as Feature Vector	LBP[26]
	ULBP[26]
‘Color Only’ as Feature Vector	Autocorr_HS[21]
‘Color + Texture’ as Feature Vector	ColorHist_HSI+ULBP[24]
	LECoP[28]
	Interchannel_HS+DSGLCM[23]
‘Color Only’ as Feature Vectorwith Wavelets	Wavelet_ColorHist_RGB[115]
	Wavelet_Autocorr_HS[116]
‘Color + Texture’ as Feature Vector with Wavelets	Wavelet_Autocorr+BVLC[22]
	Proposed Method

Image Dataset-1 (Corel-1K and Multiresolution Version)

The five performance measure values for both Corel-1K and Multiresolution Corel-1K image datasets are shown in Table 3.2. In the case of the single resolution Corel-1K dataset, all five performance measures APR, ARR, F-Measure, ANMRR, and TMRE for the proposed method are improved by 1.95%, 1.02%, and 0.87%, 1.08%, and 0.45% respectively concerning best among other methods given in Table 3.1. Whereas in the case of the multiresolution version of the Corel-1K dataset, the proposed method shows improvement of 6.60%, 25.66%, 14.93%, 31.48%, and 20.05%, respectively, for the same performance measures concerning best among other methods for multiresolution Corel-1K given in Table 3.1. Precision, Recall graphs for both Corel-1K and Multiresolution Corel-1K image datasets are shown in Figure 3.7.

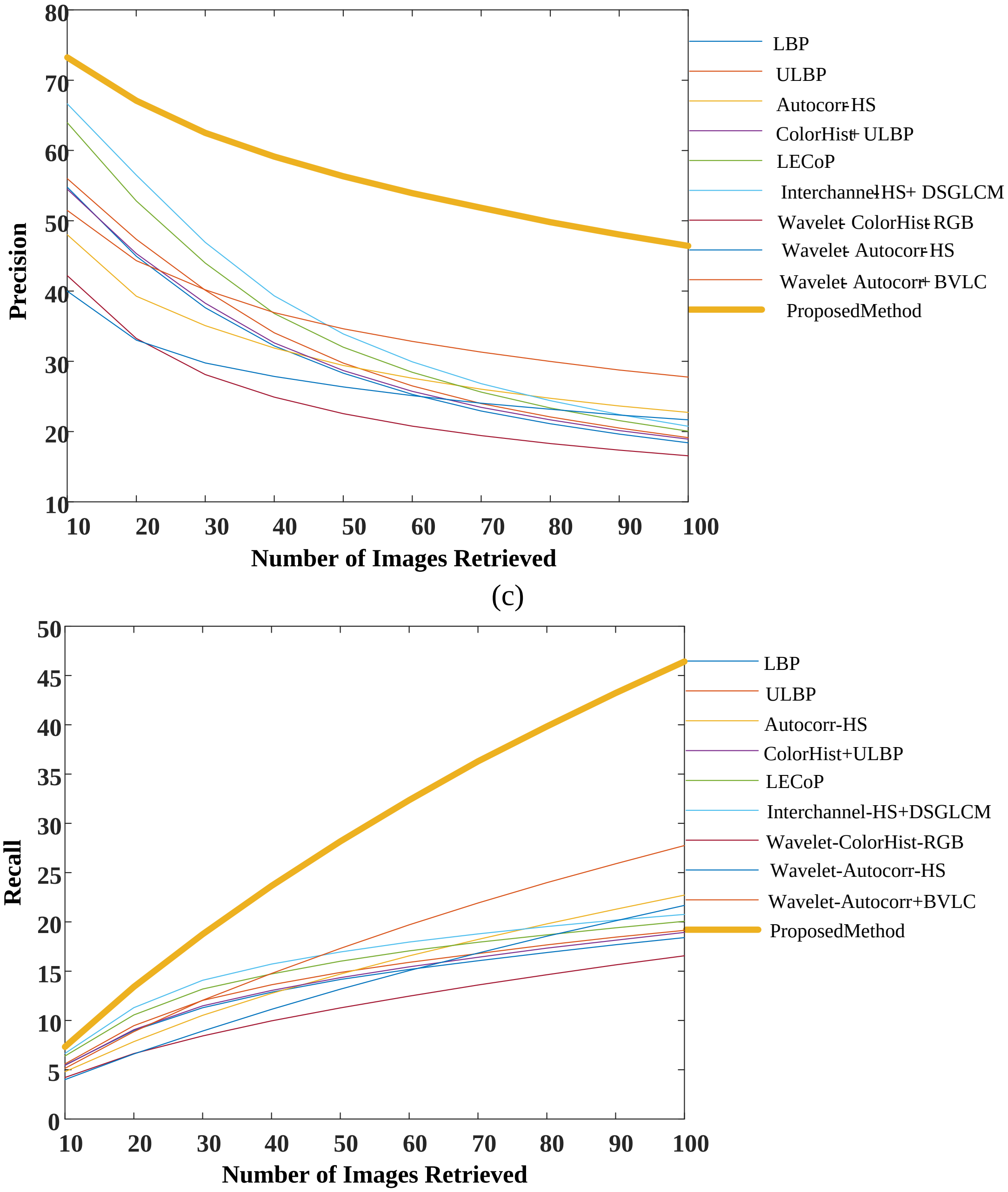
Image Dataset-2 (Corel-5K and Multiresolution Version)

The five performance measure values for both Corel-5K and Multiresolution Corel-5K image datasets are shown in Table 3.3. By observing the results, in the case of the single resolution Corel-5K dataset, the results of four performance measures APR, ARR, F-Measure, and AMNRR for the proposed method are improved by a minimum of 1.46%, 1.02%, 0.87%, and 0.27% respectively. For TMRE, the proposed method lags by less than one percent from the best among other techniques. Whereas in the case of the multiresolution version of Corel-5K proposed method shows a minimum improvement of 3.05%, 7.45%, 4.81%, 9.90%, and 9.02%, respectively, for the same performance measures.



(a)

(b)



(c)

(d)

Figure 3.7 (a) Precision Graph for Corel-1K. (b) Recall Graph for Corel-1K. (c) Precision Graph for Multiresolution Corel-1K. (d) Recall Graph for Multiresolution Corel-1K.

Table 3.2 Performance Measures for Core-1K and Multiresolution Corel-1K.

	Corel 1K					**Corel-1K Multiresolution**				
Method Name	APR	ARR	F-Measure	ANMRR	TMRE	APR	ARR	F-Measure	ANMRR	TMRE
LBP[26]	69.18	38.69	31.228	0.5208	8.03	54.76	18.41	17.326	0.7684	0.08
ULBP[26]	69.26	44.42	33.929	0.4574	7.49	56.00	19.14	18.162	0.7624	0.07
Autocorr_HS[21]	52.54	28.32	22.423	0.6278	9.45	48.00	22.72	18.510	0.6981	0.09
ColorHist_HSI+ULBP[24]	66.56	42.60	32.365	0.4755	7.42	54.48	18.94	17.637	0.7645	0.07
LECoP[28]	76.14	46.82	36.691	0.4301	7.59	63.95	20.07	19.560	0.7548	0.08
Interchannel_HS+DSGLCM[23]	78.20	49.92	38.632	0.3988	7.29	66.64	20.76	20.594	0.7473	0.07
Wavelet_ColorHist_RGB[115]	49.37	29.49	22.649	0.6129	8.63	42.19	16.57	14.190	0.7844	0.09
Wavelet_Autocorr_HS[116]	50.15	30.50	23.212	0.6016	8.79	39.99	21.68	16.762	0.7082	0.09
Wavelet_Autocorr+BVLC[22]	60.82	35.25	27.470	0.5443	8.85	51.48	27.75	21.896	0.6322	0.09
Proposed Method	**80.15**	**50.94**	**39.502**	**0.3880**	**7.25**	**73.24**	**46.42**	**35.523**	**0.4365**	**0.07**

Table 3.3 Performance Measures for Core-5K and Multiresolution Corel-5K.

	Corel 5K					**Corel-5K Multiresolution**				
Method Name	APR	ARR	F-Measure	ANMRR	TMRE	APR	ARR	F-Measure	ANMRR	TMRE
LBP[26]	46.19	19.85	16.667	0.7446	41.62	28.20	8.91	8.054	0.886	48.15
ULBP[26]	46.94	21.10	17.584	0.7276	38.87	25.02	8.75	7.564	0.884	48.22
Autocorr_HS[21]	35.33	14.91	12.509	0.8065	44.67	25.64	8.94	7.727	0.882	48.24
ColorHist_HSI+ULBP[24]	44.96	20.61	17.055	0.7358	40.23	32.09	9.85	9.080	0.875	49.00
LECoP[28]	58.34	27.18	22.782	0.6603	37.58	39.95	11.25	10.790	0.859	48.65
Interchannel_HS+DSGLCM[23]	61.14	29.67	24.628	0.6296	37.44	43.87	12.78	12.323	0.841	47.81
Wavelet_ColorHist_RGB[115]	31.12	12.83	10.669	0.8317	45.15	22.94	7.09	6.331	0.907	48.88
Wavelet_Autocorr_HS[116]	31.13	13.00	10.722	0.8275	45.56	21.88	7.93	6.670	0.894	48.40
Wavelet_Autocorr+BVLC[22]	38.67	16.61	13.774	0.7821	43.05	29.10	10.75	9.157	0.858	47.98
Proposed Method	**62.60**	**30.23**	**25.250**	**0.6269**	**36.55**	**46.92**	**20.23**	**17.133**	**0.742**	**43.39**

Image Dataset-3 (Corel-10K and Multiresolution Version)

The five performance measure values for both Corel-10K and Multiresolution Corel-10K image datasets are shown in Table 3.4. By analyzing the results given in Table 3.4, in the case of the single resolution Corel-10K dataset, by using the proposed method, 1.68%, 0.43%, 0.52%, 0.27%, and 0.02% improvement is obtained for APR, ARR, F-Measure, ANMRR, and TMRE respectively concerning best among other methods. In the case of the multiresolution version of the Corel-10K dataset, the performance measures are improved by 1.43%, 4.10%, 2.61%, 5.40%, and 5.51% compared to the best other methods for multiresolution Corel-10 K given in Table 3.1.

Table 3.4 Performance Measures for Core-10K and Multiresolution Corel-10K.

	Corel 10K					**Corel-10K Multiresolution**				
Method Name	APR	ARR	F-Measure	ANMRR	TMRE	APR	ARR	F-Measure	ANMRR	TMRE
LBP[26]	38.17	15.20	12.969	0.8057	83.20	23.48	6.61	6.158	0.915	96.08
ULBP[26]	38.61	15.65	13.318	0.8001	79.67	26.64	7.42	7.000	0.906	97.56
Autocorr_HS[21]	29.18	11.32	9.628	0.8530	86.74	20.53	6.25	5.571	0.918	95.44
ColorHist_HSI+ULBP[24]	35.92	15.11	12.709	0.8061	80.79	26.14	7.31	6.885	0.907	97.77
LECoP[28]	48.99	20.72	17.685	0.7398	75.65	32.92	8.77	8.505	0.890	97.35
Interchannel_HS+DSGLCM[23]	51.89	22.67	19.201	0.7148	73.58	35.93	9.99	9.645	0.875	95.70
Wavelet_ColorHist_RGB[115]	25.59	9.45	8.069	0.8762	89.56	19.07	5.25	4.826	0.932	97.28
Wavelet_Autocorr_HS[116]	25.54	9.52	8.034	0.8742	89.52	17.89	5.45	4.782	0.927	96.49
Wavelet_Autocorr+BVLC[22]	30.94	12.12	10.247	0.8410	84.74	23.50	7.63	6.706	0.900	94.97
Proposed Method	**53.57**	**23.10**	**19.728**	**0.7121**	**73.57**	**37.36**	**14.09**	**12.257**	**0.821**	**89.99**

Image Dataset-4 (VisTex and Multiresolution Version)

The five performance measure values for both VisTex and Multiresolution VisTex image datasets are shown in Table 3.5. In single resolution VisTex dataset, a combination of inter-channel voting between hue and saturation with DSGLCM is giving the best performance, but in the case of the multiresolution version of VisTex proposed method is performing best while giving a minimum improvement of 0.02%, 17.59%, 12.95%, 18.10% and 26.00% for APR, ARR, F-Measure, ANMRR, and TMRE respectively.

Table 3.5 Performance Measures for VisTex and Multiresolution VisTex.

	VisTex					**VisTex Multiresolution**				
Method Name	APR	ARR	F-Measure	ANMRR	TMRE	APR	ARR	F-Measure	ANMRR	TMRE
LBP[26]	96.45	79.30	63.267	0.1261	4.11	75.55	23.13	26.914	0.723	35.43
ULBP[26]	97.66	83.24	65.485	0.1096	4.23	77.93	23.13	27.253	0.725	36.86
Autocorr_HS[21]	81.91	53.54	46.617	0.3798	15.91	59.69	27.54	27.028	0.657	27.75
ColorHist_HSI+ULBP[24]	95.94	74.32	61.103	0.1713	8.32	74.18	22.72	26.452	0.730	37.37
LECoP[28]	98.40	81.69	65.608	0.1185	4.80	82.73	22.86	27.759	0.725	36.01
Interchannel_HS+DSGLCM[23]	98.45	80.21	65.339	**0.0544**	**2.62**	89.90	24.53	30.053	0.707	34.74
Wavelet_ColorHist_RGB[115]	75.86	48.09	42.298	0.4213	17.29	51.37	17.72	19.453	0.784	37.15
Wavelet_Autocorr_HS[116]	81.60	55.04	47.323	0.3534	14.05	62.19	28.39	27.758	0.644	29.85
Wavelet_Autocorr+BVLC[22]	89.69	65.53	54.654	0.2394	7.95	76.21	40.03	37.800	0.517	24.34
Proposed Method	**99.41**	**89.32**	**68.894**	0.0622	2.84	**89.92**	**57.72**	**50.746**	**0.336**	**14.14**

Image Dataset-5 (STex and Multiresolution Version)

The five performance measure values for both STex and Multiresolution STex image datasets are shown in Table 3.6. As shown in Table 3.6 for the single resolution STex dataset, a combination of inter-channel voting between hue and saturation with DSGLCM is performing best. Still, the proposed method's performance is lagging from the best performing method by less than1%. However, in the case of the multiresolution version of the STex dataset proposed method is showing the best performance by a factor of 7.58%, 21.43%, 15.09%, 24.90%, and 39.82% in the case of APR, ARR, F-measure, ANMRR, and TMRE respectively when compared to best among other methods.

Image Dataset-6 (Color Brodatz and Multiresolution Version)

In the case of the Color Brodatz dataset, the proposed method gives the best performance in both the single and multiresolution versions of this dataset, as shown in Table 3.7. In a single resolution dataset, 0.66%, 2.90%, 1.35%, 1.44%, and 0.60% improvements are obtained for APR, ARR, F-Measure, ANMRR, and TMRE, respectively, concerning best among other methods mentioned in Table 3.1. Whereas in the case of the multiresolution version of Corel-10K same performance measures achieve a minimum improvement of 17.83%, 31.28%, 25.93%, 34.10%, and 22.43%, respectively.

Table 3.6 Performance Measures for STex and Multiresolution STex.

	STex					STex Multiresolution				
Method Name	APR	ARR	F-Measure	ANMRR	TMRE	APR	ARR	F-Measure	ANMRR	TMRE
LBP[26]	76.92	47.22	41.849	0.4526	104.11	48.00	15.22	17.329	0.815	425.98
ULBP[26]	82.54	52.39	45.915	0.4027	86.54	52.24	18.64	18.640	0.804	442.51
Autocorr_HS[21]	68.41	41.44	36.642	0.5020	104.89	41.79	15.50	16.429	0.753	453.73
ColorHist_HSI+ULBP[24]	87.11	59.08	50.683	0.3333	73.53	51.70	16.14	18.558	0.804	444.25
LECoP[28]	91.67	65.80	55.397	0.2657	44.40	64.02	19.52	22.685	0.766	434.83
Interchannel_HS+DSGLCM[23]	93.41	69.20	57.021	0.3326	45.13	68.36	20.53	24.049	0.754	418.05
Wavelet_ColorHist_RGB[115]	50.27	24.75	23.552	0.7004	230.30	34.57	10.45	12.112	0.871	440.55
Wavelet_Autocorr_HS[116]	65.14	36.77	33.445	0.5596	150.02	40.24	14.37	15.476	0.822	393.34
Wavelet_Autocorr+BVLC[22]	71.67	42.59	38.029	0.4917	93.52	45.50	17.84	18.498	0.779	330.00
Proposed Method	**93.44**	**68.53**	**57.245**	**0.2422**	**42.43**	**75.94**	**41.96**	**39.158**	**0.505**	**141.21**

Table 3.7 Performance Measures for Color Brodatz and Multiresolution Color Brodatz.

	Brodatz					**Brodatz Multiresolution**				
Method Name	APR	ARR	F-Measure	ANMRR	TMRE	APR	ARR	F-Measure	ANMRR	TMRE
LBP[26]	89.29	70.22	54.997	0.2247	13.92	51.51	13.80	15.742	0.831	103.08
ULBP[26]	91.97	74.39	57.630	0.1940	12.36	54.37	14.12	16.311	0.828	105.30
Autocorr_HS[21]	94.47	68.87	55.862	0.2658	34.18	49.91	18.46	18.679	0.775	80.36
ColorHist_HSI+ULBP[24]	92.79	74.08	57.766	0.1897	16.31	60.56	15.69	18.192	0.810	104.96
LECoP[28]	98.98	87.51	65.949	0.0837	8.04	68.96	17.31	20.286	0.790	99.19
Interchannel_HS+DSGLCM[23]	98.91	87.34	65.818	0.0792	6.35	66.39	16.79	19.592	0.796	99.34
Wavelet_ColorHist_RGB[115]	68.69	42.67	36.046	0.4874	40.53	35.39	9.93	11.064	0.877	105.28
Wavelet_Autocorr_HS[116]	88.51	64.84	52.257	0.2723	25.59	47.97	16.67	17.198	0.794	101.92
Wavelet_Autocorr+BVLC[22]	93.24	72.19	57.127	0.1965	14.04	55.70	21.44	21.390	0.736	102.40
Proposed Method	**99.64**	**90.41**	**67.304**	**0.0648**	**6.28**	**86.79**	**52.72**	**47.326**	**0.395**	**55.46**

3.5 Observations

This CBIR approach proposes a novel method for content-based image retrieval by integrating the color and texture information of an image which is efficient for both single and multiresolution image datasets. It combines global and local characteristics to get the features. In the proposed method, the number of decomposition levels is controlled by considering the size of the query image and the image having a minimum dimension in the dataset. As part of the global features, the inter-channel relationship between all possible permutations of H, S, and I are considered. To get the local feature descriptor diagonally symmetric pattern followed by GLCM on this is applied. The feature vector of each decomposition level is formed by fusing the color and texture features. The final feature vector is computed by a weighted combination of feature vectors obtained in various decomposition levels. The proposed method has been evaluated on three natural image datasets (Corel-1K, Corel-5K, and Corel-10K) and three color texture image datasets (VisTex, STex, and Color Brodatz) and their multiresolution versions. Five performance measures: APR, ARR, F-Measure, ANMRR, and TMRE, are calculated to evaluate the proposed method, and these performance methods are compared to nine existing standard methods. In the case of a single resolution dataset in all-natural image datasets and one color texture dataset (Color Brodatz) proposed method is giving the best performance for all performance measures. In the case of all six multiresolution datasets, the proposed method provides a performance improvement of an average of 15.36% compared to the best among the nine existing methods.

Chapter 4

Content Based Image Retrieval using Feature Fusion: Deep Learning and Handcrafted

The method proposed in Chapter-3 uses the combination of texture and color features for CBIR. Instead of using only the handcrafted features for this, the fusion of handcrafted and the features obtained by using different CNNs can further increase the accuracy and is understood from the literature given in Chapter-2. It is because CNNs features are obtained through backpropagation error correction, which produces a better feature representation, thus improving the accuracy of the overall CBIR system. In this chapter, an efficient image retrieval framework is proposed by combining high-level features from GoogleNet using transfer learning and handcraft features extracted from both RGB and HSI color models. A modified version of dot-diffused block truncation coding (DDBTC) [41] is proposed to extract handcraft features in RGB color space. For HSI color space to explore the interrelationship between color and intensity component of an image, interchannel voting between hue, saturation, and intensity component is used as a color feature. Gray Level Co-occurrence Matrix (GLCM) on Diagonally Symmetric Pattern (DSP) is computed to get the texture features on HSI color space. To extract shape features histogram of orientated gradient (HOG) is applied to RGB color space.

4.1 CNN Algorithms for CBIR

From the literature, it is clearly understood that deep learning features for CBIR are showing very high accuracies when compared to handcrafted features. The general outline of CNN based CBIR approach is shown in Figure 4.1. One subset of the image dataset is used to train the CNN model using the transfer learning approach. After that trained CNN model extracts features of all dataset images to create a ‘Feature Dataset’. The same trained CNN model is used to extract the features of the query image. This extracted feature is compared with the ‘Feature Dataset’ using similarity measures, and then the retrieval accuracy is calculated.

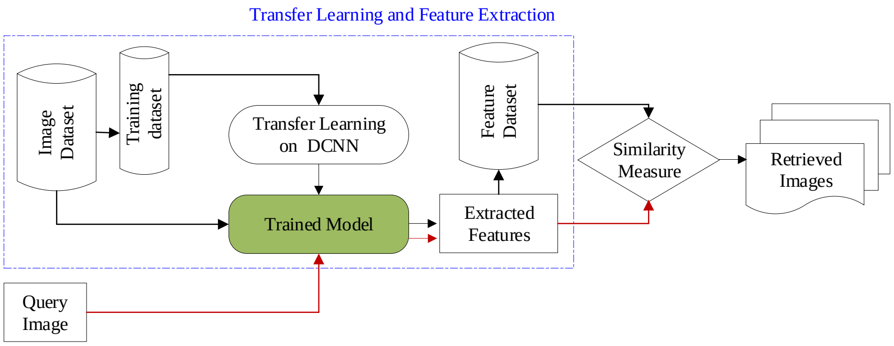


Figure 4.1 General Outline of CNN based CBIR Approach.

4.1.1 Transfer Learning

Transfer learning is used to make a CNN model training process more efficient and less time-consuming. In this approach, assigned weights of a trained CNN model are used rather than training the entire model from scratch [117, 118]. But due to the huge computational complexity of the training process of such CNN models, maximum literature imports the CNN model from already published CNN architecture such as AlexNet, VGG-16, VGG-19, GoogleNet, etc. Canziani et al. presented a review article on the performance of pre-trained CNN models in solving various computer vision problems [119].

4.1.2 Feature Extraction using CNN

A simple CNN is a sequence of layers, and every layer of a CNN transforms one volume of activation to another by passing through certain differentiable functions. There are mainly three types of layers in CNN architectures: Convolutional Layer, Pooling Layer, and Fully-Connected Layer. In most of the research, the last SoftMax layer and Fully-Connected layer, used for calculating probabilities of each class, are removed for feature extraction. The last Average pool layer is considered as the final feature vector of the CNN model. As this vector is the deepest layer of the model, this represents the most learned high-level features. The input images of a CNN based CBIR system are encoded through pre-trained CNN models and got an *n*-dimensional feature vector. The value of ‘*n*’ varies with the selection of deep learning network architecture. The flowchart of this process is shown in Figure 4.2. In this chapter, pre-trained GoogleNet is used for high-level feature extraction.

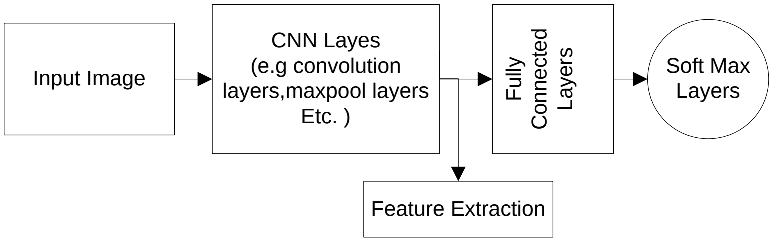


Figure 4.2 Flow Diagram for Feature Extraction from CNN model.

Feature extraction using GoogleNet

GoogleNet is more suitable for high-level feature extraction because of its special inception module, which will give proper memory utilization and less computation complexity.[97] GoogleNet has 22 layers and follows a directed acyclic graph structure. It has an error rate of 6.7%. GoogleNet emphasized that CNNs can be created with layer structures rather than sequentially putting layers. It was trained on high-configuration GPUs in one week.

The feature extraction method for GoogleNet is described below:

1. Images of size 224×224 must be given to GoogleNet as input. So, images of different sizes must be resized to 224×224 as part of pre-processing.
2. After that, input images are fed into multiple numbers of convolution layers.
3. After each convolution layer, Rectified Linear Units (ReLU) is applied. Relu is used as an activation function which increases the non-linearity of the CNN model.
4. To reduce the computational complexity of the network through parameter reduction, the Max-pooling layer is used. This layer is very useful for extracting invariance information of an image.
5. Repeat steps 1 to 4 until the termination condition is satisfied.

4.2 Super-Resolution

One of the CNN based methods of converting low-resolution images to high-resolution images is Super-resolution. This process considers a Single Image Super-Resolution (SISR) method, which aims to recuperate one image having high-resolution from one low-resolution image. But high-frequency image information of high-resolution cannot be reconstructed from low- resolution images, which results in poor conversion of low to high resolution. In addition, it is possible to produce more than one high-resolution image from a single low-resolution image, making SISR more challenging.

4.2.1 Very Deep Super Resolution Network

To perform SISR efficiently, a CNN based architecture was proposed by Kim et al., known as the Very Deep Super Resolution (VDSR) network [120]. It mainly focuses on learning the mapping between high- and low-resolution images. As high- and low-resolution images are differed in high-frequency information but have similar image content, mapping between them plays an important role in the SISR method. A residual learning strategy is introduced by the VDSR network, which helps the network estimate a residual image. In the case of super-resolution, the difference between an upscaled version of low resolution and a high-resolution image is known as a residual image. The bicubic image interpolation technic is used for up sampling to match the size of the low-resolution image to that of a high-resolution image.[121] A residual image carries high-frequency information about an image. The luminance of a color image is used for estimating the residual image. To solve that purpose, YCbCr color space is used. Y component, known as the Luminance component, signifies the brightness of each pixel, whereas the other two components, Cb and Cr, carry color information. To train VDSR, only the Y component is used because human interpretation is more sensitive to brightness change than color details. Let Y_{low} represent the Y component of the upscaled low-resolution image and Y_{high} represents the Y component of the high-resolution image. Then VDSR takes Y_{low} as the input to the network and learns to estimate $Y_{residual}= Y_{high}- Y_{low}$ from a large training dataset. After estimating the residual image, the Y component of high-resolution images, $Y_{recon,}$ can be reconstructed by adding the estimated $Y_{residual}$ to Y_{low} Then, the image is converted to RGB color space using the reconstructed Y_{recon} component. The relation between the size of high- and low-resolution images is known as the scale factor. A higher value of this scale factor makes the SISR method more challenging because it causes more loss of high-frequency information in the high-resolution image. To solve this problem, large receptive field is used by VDSR. Using this method multiple scale factors are used to train the VDSR network using scale augmentation. In addition, images having non-integer scale factors are also accepted by the VDSR network. An example of VDSR based super-resolution method and traditional bi-cubic method is given in Figure 4.3.

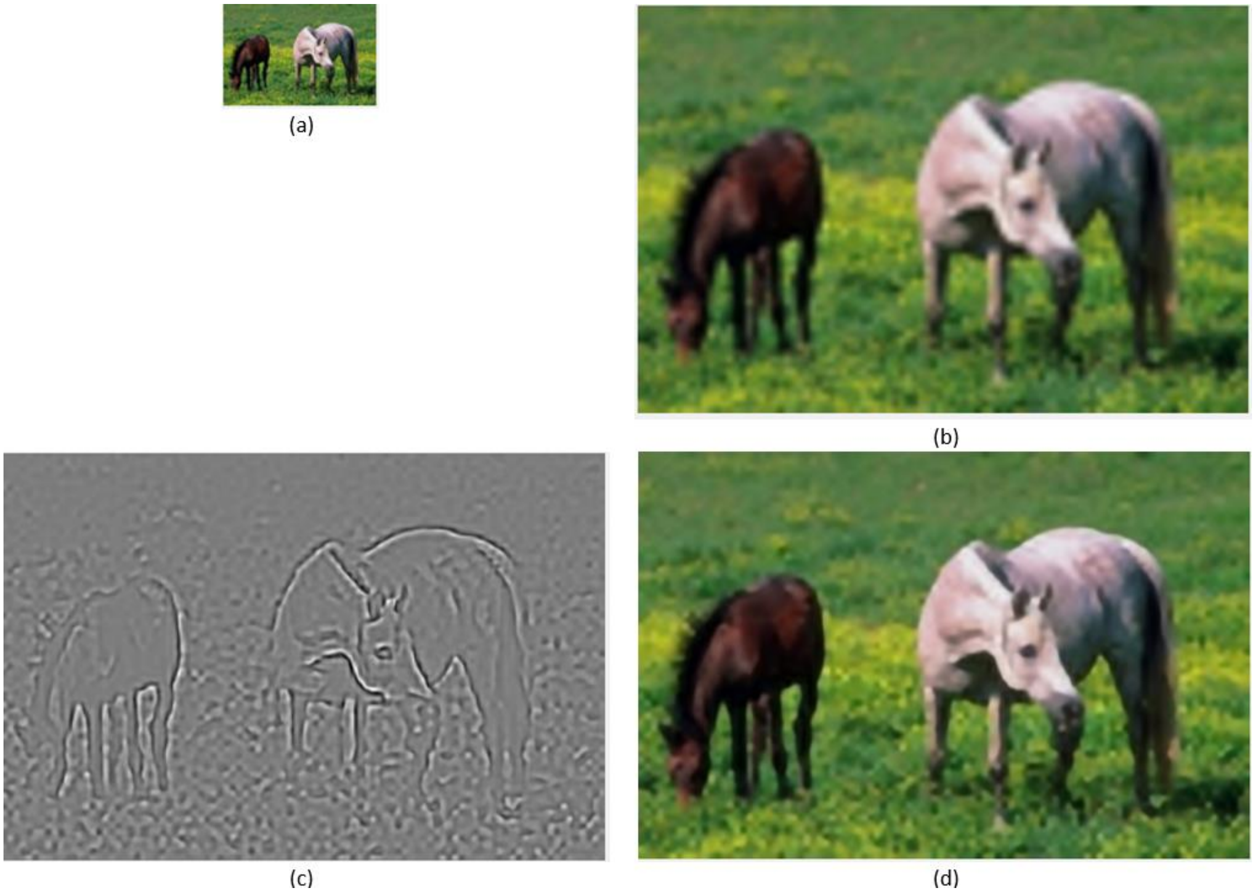


Figure 4.3 (a) Low-Resolution image. (b) High-Resolution Image using Bicubic Interpolation. (c) Residual Image from VDSR. (d) High-Resolution Image using VDSR.

4.3 Dot-Diffused Block Truncation Coding

Dot-Diffused Block Truncation Coding (BTC) was proposed by Delp et al. in the year 1979 which is an efficient method of image compression.[122] In this method, the input image is divided into multiple non-overlapped blocks, and to represent each block, two extreme quantizers (i.e., low and high mean values of each block), and a bitmap image is used. To produce a bitmap image, a thresholding function based on the mean of each block is used. BTC-based CBIR methods have shown promising results in image retrieval problems [123, 124, 125]. Two important feature vectors, known as color quantizers and bitmap, were generated in DDBTC, proposed by Guo et al.[41] Subsequently, three types of codebooks, e.g., maximum color quantizer, minimum color quantizer, and bit map codebooks, are used to form above mentioned feature vectors. In the DDBTC encoding process, each RGB color image of size $P \times Q$ is divided into non-overlapping image blocks of size $p \times q$.

Parallel mechanism can be applied on each image block independently Let *I(i,j)* is denoted as an image block at location *(i,j)*, where $i=1,2,3\ldots,\frac{P}{p}$ and $j=1,2,3\ldots,\frac{Q}{q}$. Let $I_R(a,b)$,$I_G(a,b)$, and $I_B(a,b)$ are the red, green and blue pixel values in image block located at position *(i,j)*, where *a=1,2,3….,p and b=1,2,3…,q*. One minimum quantizer (E_{min}) and one maximum quantizer (E_{max}) for every single image block are generated by the DDBTC encoder. For the image block located at position (i.j) these two quantizers are calculated using equation 4.1 and equation 4.2

$$E_{\min}(i,j)=\{\min I_R(a,b),\min I_G(a,b),\min I_B(a,b)\};\forall a\in\{1,2.....p\}and\forall b\in\{1,2,...q\} \quad (4.1)$$

$$E_{\max}(i,j)=\{\max I_R(a,b),\max I_G(a,b),\max I_B(a,b)\};\forall a\in\{1,2.....p\}and\forall b\in\{1,2,...q\} \quad (4.2)$$

To generate the DDBTC bitmap, the grayscale transformed value for each pixel *(a,b)* in block*(i,j)* is calculated by equation 4.3, which is nothing but taking the average of R, G, and B channels values of pixel position *(a,b)*. The mean value for this image block*(i,j)* can be easily computed using equation 4.4. Finally, the DDBTC bitmap image of the block*(i,j)* can be calculated using equation 4.5

$$G(a,b)=\frac{1}{3}\left[I_R(a,b)+I_G(a,b)+I_B(a,b)\right] \qquad (4.3)$$

$$G(i,j)=\frac{1}{p\times q}\sum_{a=1}^{p}\sum_{b=1}^{q}G(a,b) \qquad (4.4)$$

$$BM(a,b)=\begin{cases}1 & G(a,b)\geq G(i,j)\\ 0 & else\end{cases} \qquad (4.5)$$

4.3.1 Feature Vector Generation from DDBTC

Two feature vectors, named Color Histogram Feature (CHF) and Bit Pattern Feature (BPF), are extracted from DDBTC. The CHF is calculated from minimum and maximum color quantizers whereas the BPF is derived from the bitmap image. Color Histogram Feature (CHF) is a useful and efficient feature vector that mainly concentrates on the contrast and brightness of the color image. As the first step of CHF calculation, indexing operation is performed on maximum and minimum color quantizers based on color codebooks, which are created using some clustering algorithm on the training dataset. Vector Quantization (VQ) is applied to a training set to generate color codebooks *CD_{max}* and *CD_{min}*. LBGVQ algorithm is used to perform clustering operations for generating codebooks [35].

Let $CD_{\min} = \{CD_1^{\min}, CD_2^{\min}, ..., CD_{L_{\min}}^{\min}\}$ and $CD_{\max} = \{CD_1^{\max}, CD_2^{\max}, ..., CD_{L_{\max}}^{\max}\}$ be codebooks generated for minimum and maximum quantizers, respectively, where $CD_i^{\min}$ and $CD_i^{\max}$ are codewords of the codebooks. L_{min} and L_{max} are the length of the minimum and maximum codebook, respectively. The indexing process assigns the closest codeword $CD_k^{\min}$ and $CD_k^{\max}$ using equation 4.6 and equation 4.7

$$IC_{\min}(i,j) = k \quad if \quad \left\| E_{\min}(i,j), CD_k^{\min} \right\|_2^2 < \left\| E_{\min}(i,j), CD_l^{\min} \right\|_2^2, \forall l \in \{1,2...L_{\min}\} and \quad l \neq k \qquad (4.6)$$

$$IC_{\max}(i,j) = k \quad if \quad \left\| E_{\max}(i,j), CD_k^{\max} \right\|_2^2 < \left\| E_{\max}(i,j), CD_l^{\max} \right\|_2^2, \forall l \in \{1,2...L_{\max}\} and \quad l \neq k \qquad (4.7)$$

where $\|A,B\|_2^2$ is defined as Euclidian distance between two vectors A and B. IC_{min} and IC_{max} are two matrices which store the results of the indexing process obtained from equation 4.6 and equation 4.7. After the indexing process, the color feature vectors can be calculated using equations 4.8 and 4.9. The entire process is given in Figure 4.4. An example of CHF feature vector calculation is explained in Figure 4.5

$$CHF_{\min}(k) = \frac{\sum_{i=1}^{\frac{P}{p}} \sum_{j=1}^{\frac{Q}{q}} 1 - \left\lceil \frac{|IC_{\min}(i,j) - k|}{L_{\min}} \right\rceil}{\left(\frac{P}{p}\right) \times \left(\frac{Q}{q}\right)} \qquad (4.8)$$

$$CHF_{\max}(k) = \frac{\sum_{i=1}^{\frac{P}{p}} \sum_{j=1}^{\frac{Q}{q}} 1 - \left\lceil \frac{|IC_{\max}(i,j) - k|}{L_{\max}} \right\rceil}{\left(\frac{P}{p}\right) \times \left(\frac{Q}{q}\right)} \qquad (4.9)$$

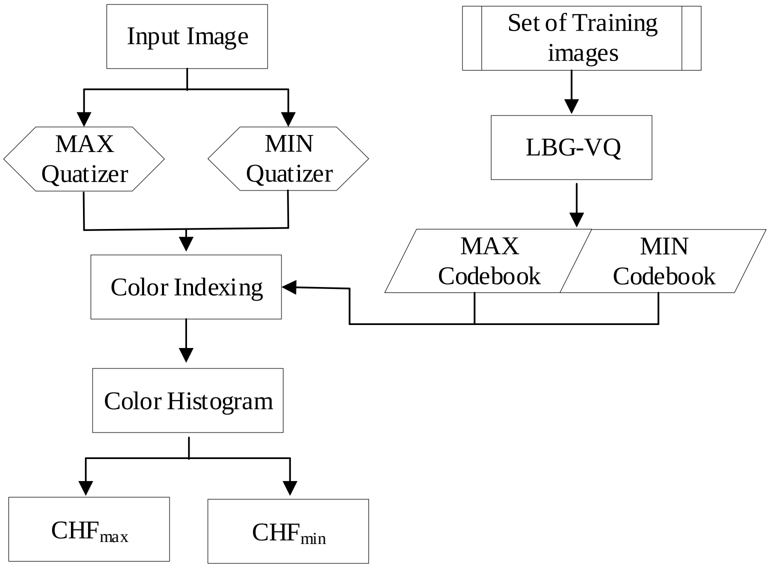
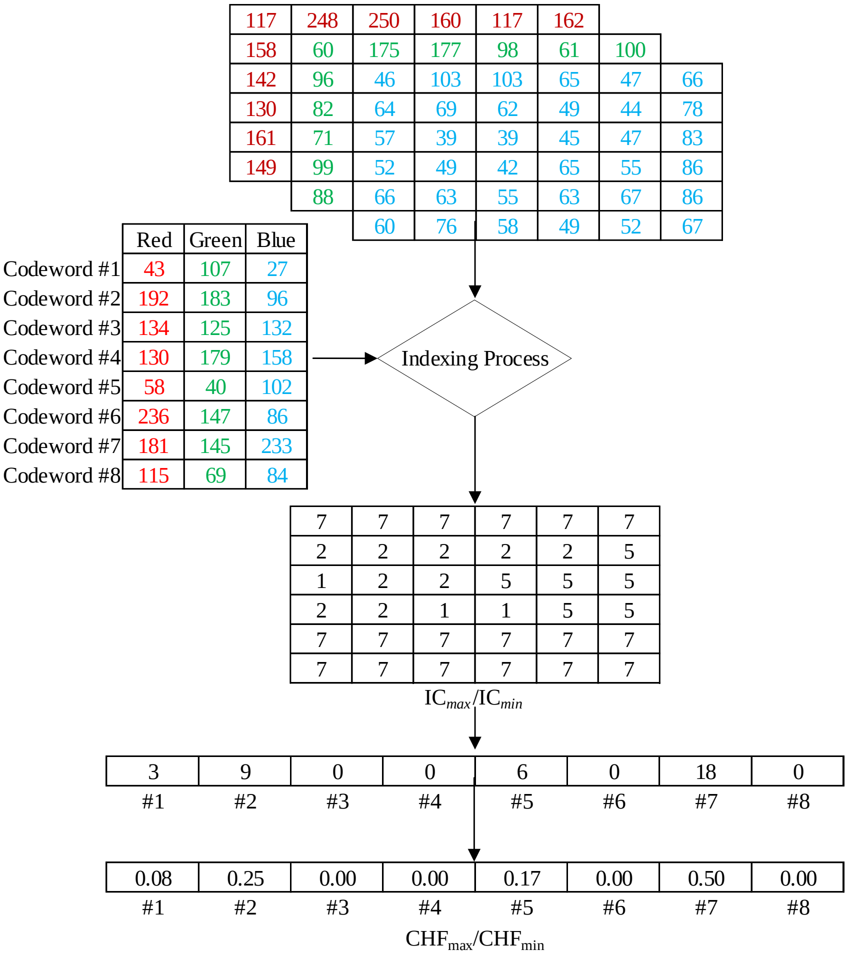


Figure 4.4 Block Diagram of CHF_{max} and CHF_{min} Extraction.



117	248	250	160	117	162		
158	60	175	177	98	61	100	
142	96	46	103	103	65	47	66
130	82	64	69	62	49	44	78
161	71	57	39	39	45	47	83
149	99	52	49	42	65	55	86
	88	66	63	55	63	67	86
		60	76	58	49	52	67

	Red	Green	Blue
Codeword #1	43	107	27
Codeword #2	192	183	96
Codeword #3	134	125	132
Codeword #4	130	179	158
Codeword #5	58	40	102
Codeword #6	236	147	86
Codeword #7	181	145	233
Codeword #8	115	69	84

7	7	7	7	7	7
2	2	2	2	2	5
1	2	2	5	5	5
2	2	1	1	5	5
7	7	7	7	7	7
7	7	7	7	7	7

3	9	0	0	6	0	18	0
#1	#2	#3	#4	#5	#6	#7	#8

0.08	0.25	0.00	0.00	0.17	0.00	0.50	0.00
#1	#2	#3	#4	#5	#6	#7	#8

Figure 4.5 Example of CHF Calculation.

Bit Pattern Feature (BPF): It is another feature that characterizes the edge and visual texture pattern of an image. Let $BP = \{BP_1, BP_2, ..., BP_{L_b}\}$ be the trained bit pattern codebook having L_b codewords. Binary VQ with a soft centroid method is used as the clustering algorithm [126]. To generate the BPF feature vector the same methodology as CHF_{min} and CHF_{max} is applied but to find the closest codeword Hamming distance is used. The indexing process for BPF calculation is given in equation 4.10.

$$BC(i,j) = k \quad if \quad \left\| BM(i,j), BP_k \right\|_1 < \left\| BM(i,j), BP_l \right\|_1, \forall l \in \{1, 2 ... L_b\}\, and \quad l \neq k \qquad (4.10)$$

where $\left\| C, D \right\|_1$ is defined as Hamming distance between two binary vector C and D and *BC* is the matrix that stores the result of the indexing process obtained from equation 4.10. The final BPF feature is calculated based on equation 4.11. The entire process is given in Figure. 4.6. The final feature vector calculation, which is a combination of CHF and BPF is described in Figure 4.7

$$BPF(k) = \frac{\sum_{i=1}^{\frac{P}{p}} \sum_{j=1}^{\frac{Q}{q}} 1 - \left\lceil \frac{\left| BC(i,j) - k \right|}{L_b} \right\rceil}{\left(\frac{P}{p}\right) \times \left(\frac{Q}{q}\right)} \qquad (4.11)$$

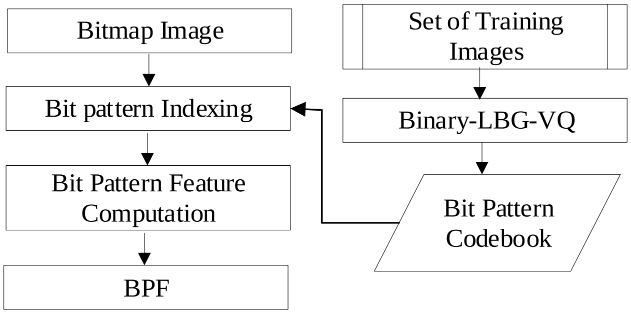


Figure 4.6 Block Diagram of BPF Feature Extraction.

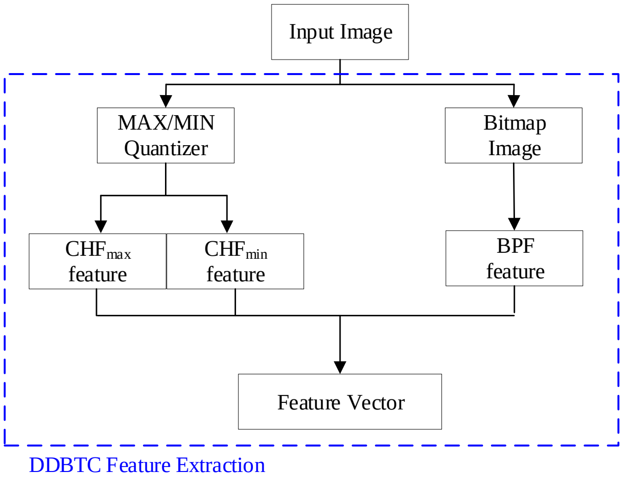


Figure 4.7 DDBTC Feature Extraction.

4.4 Methodology

The feature vector in the proposed method is a synthesis of CNN and handcraft features. When compared to other pre-trained neural networks, GoogleNet includes the inception module and uses less memory while having a lower computational complexity. As a result, the GoogleNet feature is used as the CNN feature in this method. However, because all pre-trained CNNs employ fixed-size input images, scaling the query image is an inescapable pre-processing step for any CNN-based feature extraction approach that results in significant information loss. In this work, all the images are resized using the VDSR network-based single image super-resolution (SISR) approach. Feature vectors extracted from HSI and RGB color space are used as handcraft features. An improved version of DDBTC is proposed, considering a handcrafted feature extracted from RGB color space.

In traditional DDBTC bitmap image is calculated from the grey version of the RGB input image. This grey image was constructed using equation 4.3, which is the equal ratio of R, G, and B components. Since role of each color component is not the same for RGB color image, the above approach of creating grey scale image causes significant loss of information. To solve this luminosity approach of grey conversion is used. In this method, a weighted ratio of three color components, based on their contribution to RGB color image formation, is used for the grey conversion of a color image. In the proposed version of DDBTC grey level conversion is performed using equation 4.12

$$G(x,y) = \left[0.3 \times I_R(x,y) + 0.59 \times I_G(x,y) + 0.11 \times I_B(x,y)\right] \qquad (4.12)$$

In addition to DDBTC features, the HOG feature descriptor is used to compute the shape feature in RGB color space. First of all, the input RGB color image is decomposed into the red, green, and blue components. Then the direction and size of each component are determined. Thereafter the HOG feature extraction method is applied to these two matrices. Inter-channel voting between hue, saturation, and intensity element are used as HSI model color features. This feature vector is very much useful to discover the interdependence between the intensity and color component of the input image. To reduce the length of the feature vector the Hue, Saturation, and Intensity components are placed in barrels of 72, 20 and 32 respectively. DSP is used as a local texture descriptor. GLCM of 16 levels is applied on the DSP matrix resulting in a 16 × 16 matrix. The block diagram of the proposed method is shown in Figure 4.8 and also in Algorithm 4.1

Algorithm 4.1: *Algorithm for the Proposed GoogleNet and Handcraft Feature Fusion CBIR.*

---

1. Take a color image as input.
2. Convert it into 224×224×3 size using the VDSR network-based super-resolution method.
3. GoogleNet is used to extract features from super-resolutioned images.
4. Apply modified DDBTC to extract CHF_{max}, $CHF_{min,}$ and BPF features.
5. Decompose the RGB input image into R, G, and B components and then calculate the magnitude and direction matrix for each channel.
6. For each pixel, the location finds the maximum magnitude and corresponding direction.
7. Obtain the HOG features by using matrixes created in step 6.
8. Convert RGB components of that image into HSI color space
9. Apply inter-channel voting on all possible permutation of Hue, Saturation, and Intensity resulting in six vectors.
10. Concatenate six vectors to generate a color feature vector.
11. Apply DSP on the Intensity part of HSI color space.
12. Quantize DSP into 16 levels on which Gray Level Co-occurrence Matrix is computed.
13. Convert 16 × 16 GLCM into a vector form to generate a texture feature vector.
14. The final feature vector is obtained by fusion of all five feature vectors found in steps 3,4,7,10 and 13.

15. Use similarity measures to compare the features of the query image (input image) to all the images in the dataset.
16. Obtain rank matrix by sorting according to the distance.
17. Apply performance measures on rank matrix to evaluate the proposed method.

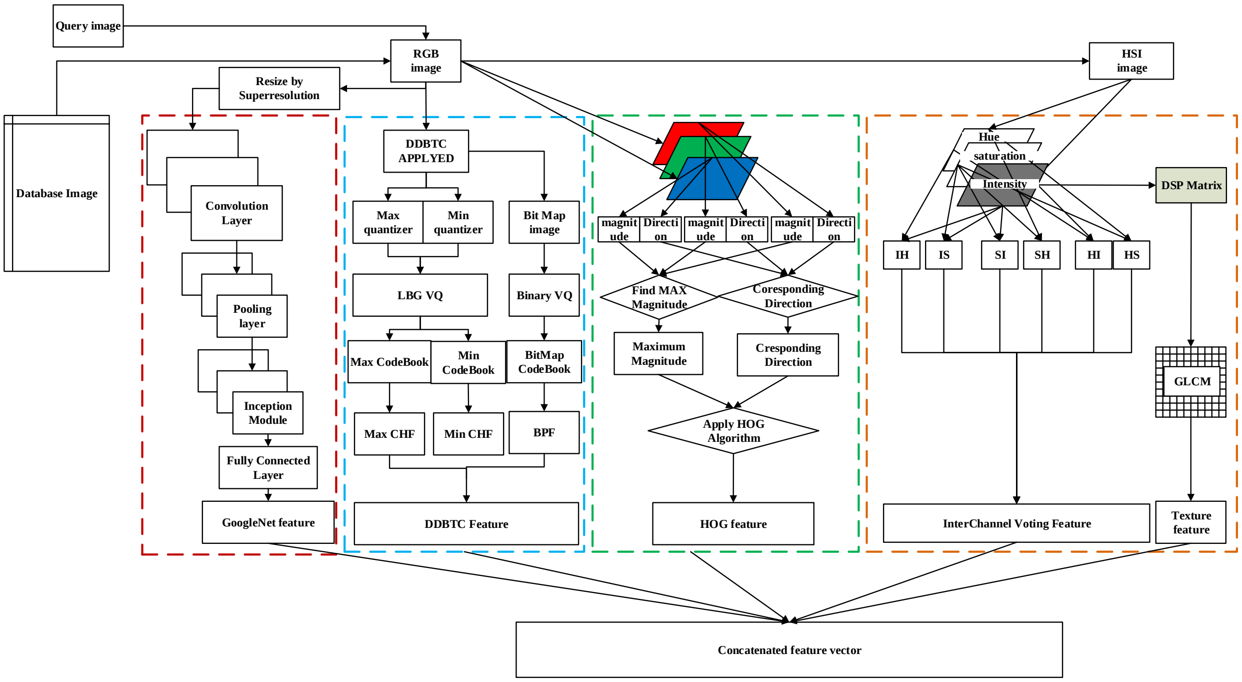


Figure 4.8 Block Diagram of the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

4.5 Experimental Results and Discussions

4.5.1 Experimental Setup

Table 4.1 shows eighteen (17) existing methods used to evaluate the retrieval performance along with the proposed method's results. Out of these 17 existing methods, 13 are handcraft feature CBIR methods, 3 CNN based CBIR methods and 1 fusion based methods. APR [23], ARR[23], F-Measure[23], ANMRR[24] and TMRE[24] are used as parameters of performance analysis. The first three parameters' performance is directly related to their value; in other words, if the value is higher, the method has delivered better outcomes. Whereas for the other two, if the value is lower the method is better.

Table 4.1 Different Methods used for Evaluating the Performance of the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

Type of Method	Method Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP [24]
	ColorHist_HSI+ULBP [24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
	Modifed DDBTC (Proposed)
CNN based Features	AlexNet [56]
	VGG16 [57]
	GoogleNet [58]
Fusion Features	DDBTC+GoogleNet [97]
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)

During the execution procedure, a total of four hypermeters are finetuned. Table 4.2 lists the specifics of various hypermeters. All CNN based CBIR techniques in Table 4.1 are trained using the same experimental setup. On MATLAB R2021a, all CNN models were trained with an NVIDIA GeForce RTX 2070 16GB GPU. In this proposed method, first five images from each class are taken to construct the training image set in the modified DDBTC method. In this method image blocks of size 4×4 is used. In this work, for all dataset length of color and binary codebook for DDBTC is set as Lmax=64, Lmin=64, and Lb=256. Experiential observations have shown that the d1 distance [23] measure provides the best performance as similarity measures between the query image and other dataset images. All the methods given in Table 4.1 are applied on ten different image dataset: Corel-1K, Corel-5K,Corel-10K, VisTex, STex, Color Brodatz, ImageNet-13K, ImageNet-65K, ImageNet-130K, and UKBench. The details are discussed in the remaining part of this section.

Table 4.2. Hyper Parameters used to Train all CNN Models of Table 4.1.

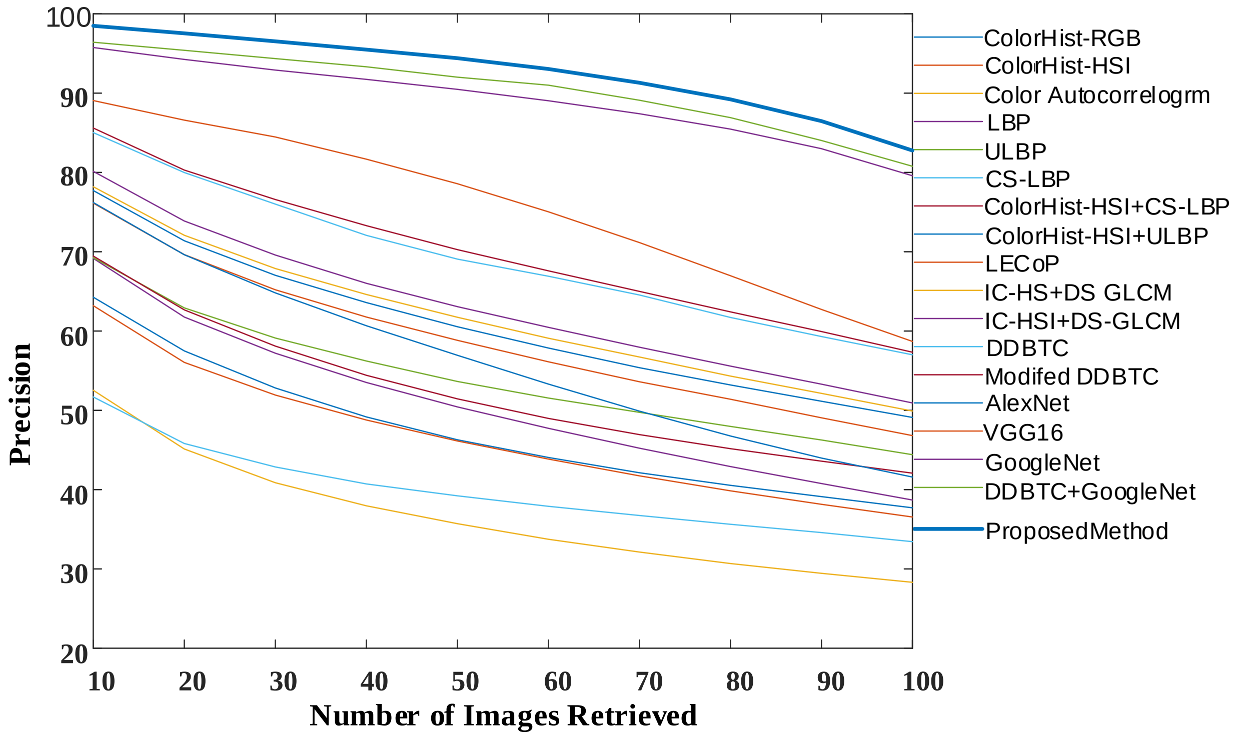
Solver Name	Mini Batch Size	Max Epochs	Initial Learn Rate
SGDM	10	6	10^{-3}

Image Dataset-1 (Corel-1K)

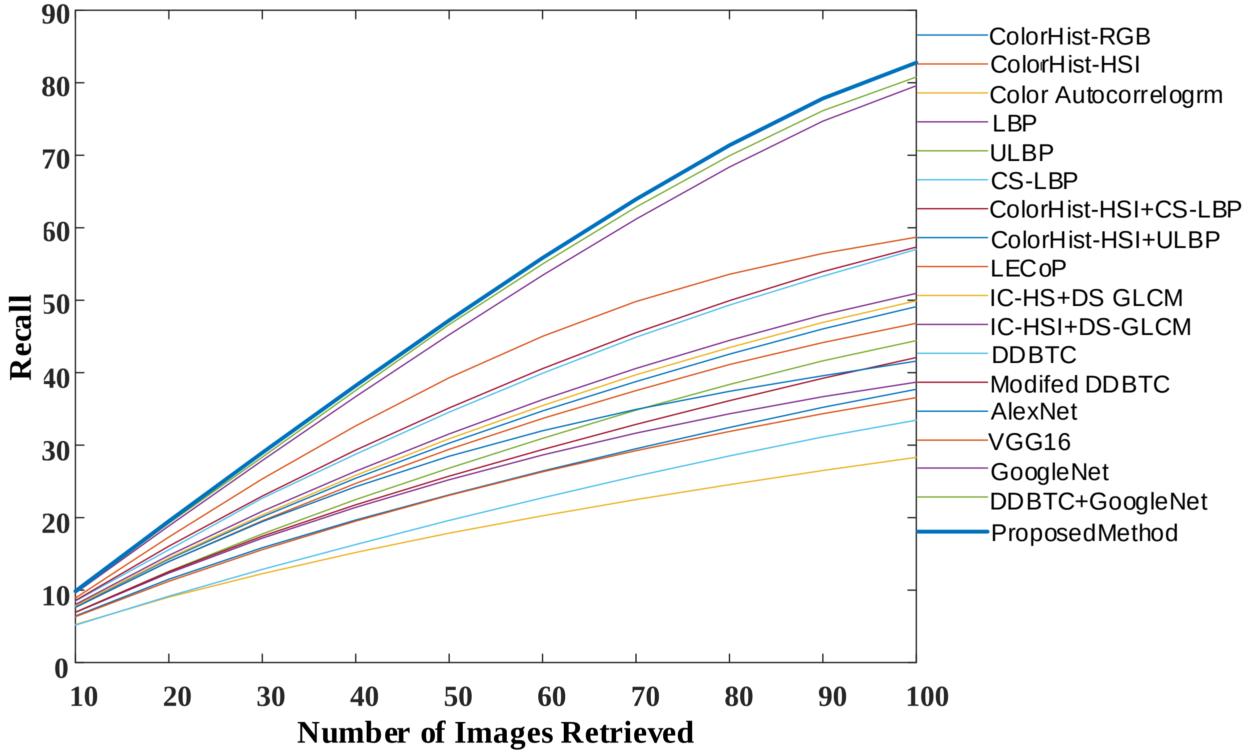
All the methods given in Table 4.1 are experimented on Corel-1K dataset. The five parameters APR, ARR, F-Measure, ANMRR and TMRE for all the eighteen methods are given in Table 4.3. The proposed method is showing a minimum improvement of 2.07%, 1.99%, 1.57%, 1.63%, and 3.13% for the five performance measures respectively. The performance graphs of all the eighteen methods for Corel-1K dataset are shown in Figure 4.9

Table 4.3 Performance Measures for Core-1K Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

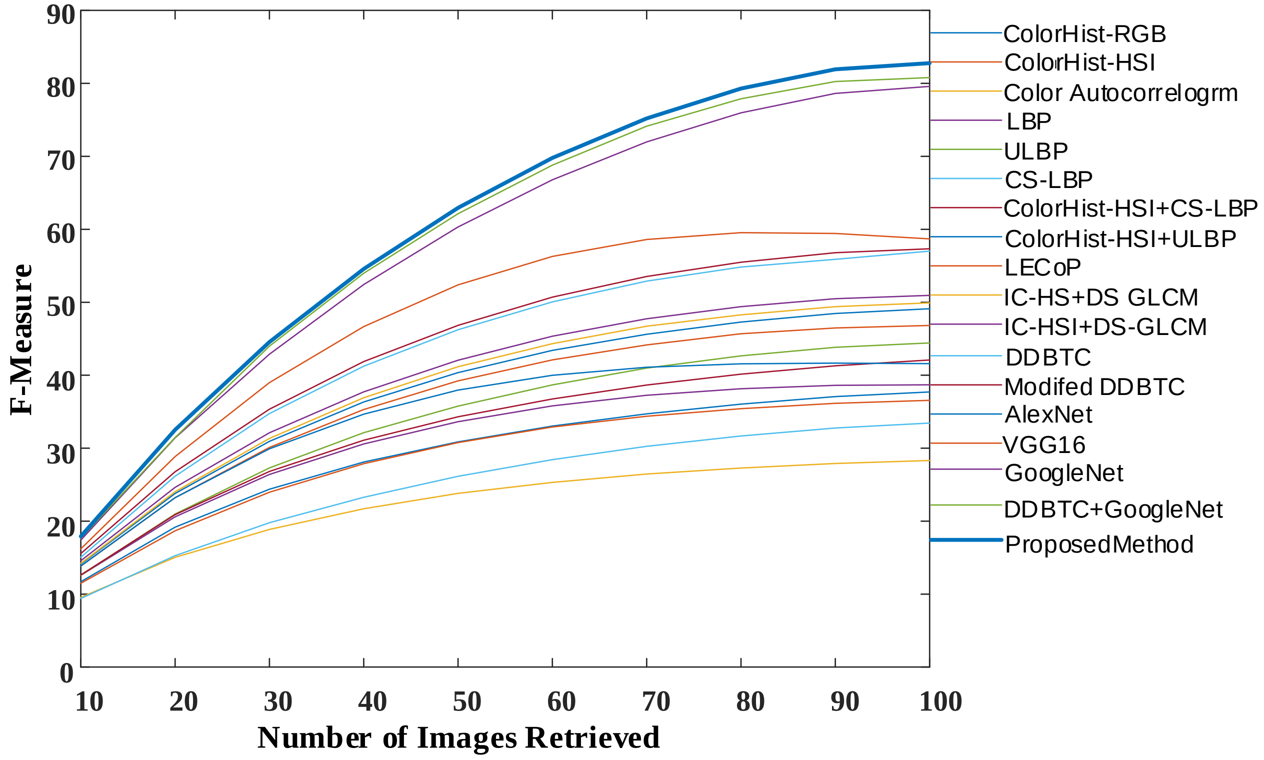
		Corel-1K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	64.28	37.71	29.288	0.5334	8.07
	ColorHist_HSI [19]	63.21	36.56	28.829	0.5335	8.73
	Color Autocorrelogrm [21]	52.54	28.32	22.429	0.6278	9.45
	LBP [26]	69.18	38.69	31.236	0.5208	8.03
	ULBP [26]	69.26	44.42	33.939	0.4574	7.49
	CS_LBP [27]	51.69	33.44	25.046	0.5746	7.82
	ColorHist_HSI+CS_LBP [24]	69.47	42.09	32.468	0.4817	7.63
	ColorHist_HSI+ULBP [24]	77.73	49.11	37.945	0.4077	7.23
	LECoP [28]	76.14	46.82	36.692	0.4301	7.59
	IC_HS+DS_GLCM [23]	78.20	49.92	38.637	0.4020	7.29
	IC_HSI+DS_GLCM [24]	80.15	50.94	39.503	0.3883	7.35
	DDBTC [41]	85.03	57.01	39.726	0.3356	6.85
	Modifed DDBTC (Proposed)	**85.61**	**57.34**	**44.025**	**0.3115**	**6.63**
CNN based Features	AlexNet [56]	76.22	41.59	34.554	0.4912	9.79
	VGG16 [57]	89.08	58.70	47.576	0.3274	5.64
	GoogleNet [58]	95.61	79.43	57.629	0.1201	3.50
Fusion Features	DDBTC+GoogleNet [97]	96.43	80.79	59.112	0.1074	4.71
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG (Prposed Method)	**98.50**	**82.78**	**60.151**	**0.0911**	**3.22**


100
90
80
70
60
50
40
30
20
Precision
10
20
30
40
50
60
70
80
90
100
Number of Images Retrieved
ColorHist-RGB
ColorHist-HSI
Color Autocorrelogrm
LBP
ULBP
CS-LBP
ColorHist-HSI+CS-LBP
ColorHist-HSI+ULBP
LECoP
IC-HS+DS GLCM
IC-HSI+DS-GLCM
DDBTC
Modifed DDBTC
AlexNet
VGG16
GoogleNet
DDBTC+GoogleNet
ProposedMethod

(a)


90
80
70
60
50
40
30
20
10
0
Recall
10
20
30
40
50
60
70
80
90
100
Number of Images Retrieved
ColorHist-RGB
ColorHist-HSI
Color Autocorrelogrm
LBP
ULBP
CS-LBP
ColorHist-HSI+CS-LBP
ColorHist-HSI+ULBP
LECoP
IC-HS+DS GLCM
IC-HSI+DS-GLCM
DDBTC
Modifed DDBTC
AlexNet
VGG16
GoogleNet
DDBTC+GoogleNet
ProposedMethod

(b)



(c)

Figure 4.9 Performance Graphs (a) Precision Graph for Corel-1K. (b) Recall Graph for Corel-1K. (c) F-Measure Graph for Corel-1K.

Image Dataset-2 (Corel-5K)

The proposed method along with seventeen existing methods given in Table 4.1 are applied to Corel-5K image dataset. The results of five performance parameters for all these methods are shown in Table 4.4. By analyzing the values obtained in Table 4.4, it is observed that the proposed method is showing an improvement of 0.54%, 0.50%, 0.46%, 0.70%, and 3.98% respectively with respect to the best among other methods.

Image Dataset-3 (Corel-10K)

All the methods given in Table 4.1 are applied to Corel-10K image dataset. The results of five performance parameters for all these methods are shown in Table 4.5. It is observed that the proposed method is showing a minimum improvement of 1.81%, 1.14%, 2.60%, 1.70%, and 1.62% respectively.

Table 4.4 Performance Measures for Core-5K Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		Corel-5K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	46.81	20.49	17.356	0.7414	42.22
	ColorHist_HSI [19]	43.26	19.00	15.996	0.7585	42.91
	Color Autocorrelogrm [21]	35.33	14.91	12.509	0.8065	44.67
	LBP [26]	46.19	19.85	16.667	0.7446	41.62
	ULBP [26]	46.94	21.10	17.584	0.7276	38.87
	CS_LBP [27]	32.50	13.40	11.178	0.8241	42.57
	ColorHist_HSI+CS_LBP [24]	50.97	23.07	19.411	0.7089	40.51
	ColorHist_HSI+ULBP [24]	60.43	28.76	23.989	0.6410	37.66
	LECoP [28]	58.34	27.18	22.782	0.6603	37.58
	IC_HS+DS_GLCM [23]	61.14	29.67	24.628	0.6296	36.19
	IC_HSI+DS_GLCM [24]	62.60	30.23	25.250	0.6269	36.55
	DDBTC [41]	62.50	31.13	25.748	0.6136	36.77
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	52.37	24.99	20.946	0.6852	49.21
	VGG16 [57]	80.10	44.21	37.321	0.4740	48.93
	GoogleNet [58]	91.49	61.30	46.179	0.3284	24.52
Fusion Features	DDBTC+GoogleNet [97]	93.71	64.21	50.010	0.2109	23.37
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG (Prposed Method)	**94.26**	**64.71**	**50.371**	**0.2039**	**20.40**

Table 4.5 Performance Measures for Core-10K Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [57]	71.07	36.26	31.031	0.5660	97.80
	GoogleNet [58]	86.52	58.19	45.196	0.3211	43.10
Fusion Features	DDBTC+GoogleNet [97]	88.43	59.82	46.400	0.5035	28.71
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**90.48**	**60.96**	**48.120**	**0.3041**	**27.11**

Image Dataset-4 (VisTex)

The results of five performance parameters for all these eighteen methods for VisTex dataset are shown in Table 4.6. It is observed that the proposed method is showing a minimum improvement of 0.54%, 2.55%, 0.38%, 0.15%, and 1.09% respectively.

Image Dataset-5 (STex)

All the methods given in Table 4.1 are applied to STex image dataset. The results of five performance parameters for all these methods are shown in Table 4.7. It is observed that the proposed method is showing a minimum improvement of 0.91%, 2.91%, 2.17%, 2.21%, and 0.80% respectively.

Image Dataset-6 (Color- Brodatz)

All the methods given in Table 4.1 are applied to Color Brodatz image dataset. The results of five performance parameters for all these methods are shown in Table 4.8. It is observed that the proposed method is showing a minimum improvement of 0.28%, 2.54%, 0.14%, 0.65%, and 1.49% respectively.

Table 4.6 Performance Measures for VisTex Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		VisTex				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	88.91	61.96	52.658	0.2866	13.38
	ColorHist_HSI [19]	90.70	61.58	53.085	0.2995	11.95
	Color Autocorrelogrm [21]	86.64	57.28	49.576	0.3462	15.51
	LBP [26]	97.07	81.00	64.357	0.1216	4.33
	ULBP [26]	98.13	82.75	65.522	0.1081	3.96
	CS_LBP [27]	88.67	66.92	54.663	0.2448	5.74
	ColorHist_HSI+CS_LBP [24]	96.21	74.08	60.938	0.1785	8.58
	ColorHist_HSI+ULBP [24]	98.91	84.55	66.679	0.0951	4.13
	LECoP [28]	98.28	81.63	65.402	0.1254	5.02
	IC_HS+DS_GLCM [23]	98.45	80.21	65.339	0.0544	2.62
	IC_HSI+DS_GLCM [24]	98.67	81.56	65.564	0.1280	5.21
	DDBTC [41]	98.35	88.69	67.987	0.0646	2.68
	Modifed DDBTC (Proposed)	**98.95**	**89.09**	**68.437**	**0.0646**	**2.91**
CNN based Features	AlexNet [56]	86.64	57.20	50.234	0.3478	26.20
	VGG16 [57]	89.77	60.67	53.045	0.3118	4.47
	GoogleNet [58]	98.44	88.14	67.993	0.0700	2.61
Fusion Features	DDBTC+GoogleNet [97]	99.46	96.43	71.858	0.0155	1.54
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**100.00**	**98.97**	**72.138**	**0.0139**	**1.12**

Table 4.7 Performance Measures for STex Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		Stex				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	59.96	31.62	29.480	0.6218	185.87
	ColorHist_HSI [19]	66.66	36.67	33.639	0.5606	115.11
	Color Autocorrelogrm [21]	68.41	41.44	36.642	0.5020	104.89
	LBP [26]	76.92	47.22	41.849	0.4526	104.11
	ULBP [26]	82.54	52.39	45.915	0.4027	86.54
	CS_LBP [27]	62.08	33.65	31.033	0.5987	139.08
	ColorHist_HSI+CS_LBP [24]	74.08	43.51	39.201	0.4928	131.00
	ColorHist_HSI+ULBP [24]	87.11	59.08	50.683	0.3333	73.53
	LECoP [28]	91.67	65.80	55.397	0.2657	44.40
	IC_HS+DS_GLCM [23]	93.51	69.23	57.621	0.2326	38.88
	IC_HSI+DS_GLCM [24]	93.44	68.53	57.245	0.2422	42.43
	DDBTC [41]	94.21	69.94	58.520	0.2411	36.57
	Modifed DDBTC (Proposed)	**95.05**	**70.68**	**59.150**	**0.2011**	**35.92**
CNN based Features	AlexNet [56]	78.10	45.56	41.781	0.4830	39.25
	VGG16 [57]	90.37	60.26	52.874	0.3325	27.75
	GoogleNet [58]	96.36	84.00	63.614	0.1381	23.76
Fusion Features	DDBTC+GoogleNet [97]	97.75	84.04	65.897	0.1055	14.69
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**98.67**	**86.95**	**67.485**	**0.0834**	**10.90**

Table 4.8 Performance Measures for Color Brodatz Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [57]	96.51	76.19	61.121	0.1785	5.75
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
Fusion Features	DDBTC+GoogleNet [97]	99.72	96.44	69.716	0.0204	3.44
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**100.00**	**98.97**	**69.814**	**0.0139**	**1.12**

Image Dataset-7 (ImageNet-13K)

All the methods given in Table 4.1 are applied to ImageNet-13K image dataset. The results of five performance parameters for all these methods are shown in Table 4.9. It is observed that the proposed method is showing a minimum improvement of 1.19%, 1.39%, 1.20%, 2.48%, and 12.47% respectively.

Image Dataset-8 (ImageNet-65K)

All the methods given in Table 4.1 are applied to ImageNet-65K image dataset. The results of five performance parameters for all these methods are shown in Table 4.10. It is observed that the proposed method is showing a minimum improvement of 1.77%, 0.80%, 2.10%, 3.18%, and 8.84% respectively.

Image Dataset-9 (ImageNet-130K)

All the methods given in Table 4.1 are applied to ImageNet-130K image dataset. The results of five performance parameters for all these methods are shown in Table 4.11. It is observed that the proposed method is showing a minimum improvement of 1.10%, 0.90%, 0.45%, 4.78%, and 3.63% respectively

Table 4.9 Performance Measures for ImageNet-13K Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		ImageNet-13K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	47.64	21.86	18.356	0.6414	9.02
	ColorHist_HSI [19]	44.60	21.73	17.926	0.5505	9.13
	Color Autocorrelogrm [21]	36.76	15.98	13.651	0.7065	9.65
	LBP [26]	48.79	21.05	17.967	0.6945	9.03
	ULBP [26]	48.94	22.34	18.348	0.6276	9.06
	CS_LBP [27]	33.50	15.05	12.978	0.7541	9.01
	ColorHist_HSI+CS_LBP [24]	52.87	25.77	21.231	0.6089	8.97
	ColorHist_HSI+ULBP [24]	61.73	29.88	25.399	0.5410	9.18
	LECoP [28]	55.34	28.57	24.078	0.5860	8.76
	IC_HS+DS_GLCM [23]	63.14	31.10	26.096	0.5296	8.99
	IC_HSI+DS_GLCM [24]	65.60	33.96	28.250	0.4927	8.72
	DDBTC [41]	68.50	35.93	31.935	0.4136	8.97
	Modifed DDBTC (Proposed)	**69.05**	**40.65**	**32.619**	**0.3823**	**8.28**
CNN based Features	AlexNet [56]	76.37	39.60	35.795	0.3452	7.85
	VGG16 [57]	83.70	47.82	40.321	0.3040	7.79
	GoogleNet [58]	94.87	69.03	54.179	0.2084	6.95
Fusion Features	DDBTC+GoogleNet [97]	95.79	70.49	55.746	0.1836	6.47
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**96.98**	**71.88**	**56.543**	**0.1588**	**5.35**

Table 4.10 Performance Measures for ImageNet-65K Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		ImageNet-65K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	27.41	11.90	10.628	0.9001	48.10
	ColorHist_HSI [19]	25.57	10.76	9.511	0.9104	48.75
	Color Autocorrelogrm [21]	20.18	8.92	7.766	0.9430	49.67
	LBP [26]	31.17	11.82	10.928	0.9057	47.86
	ULBP [26]	41.01	17.90	14.929	0.8345	45.15
	CS_LBP [27]	19.88	9.19	8.113	0.9245	48.08
	ColorHist_HSI+CS_LBP [24]	34.66	13.05	12.019	0.8523	46.05
	ColorHist_HSI+ULBP [24]	42.78	18.18	16.378	0.8129	42.69
	LECoP [28]	41.85	17.68	15.544	0.8098	45.63
	IC_HS+DS_GLCM [23]	43.06	16.04	16.048	0.7992	44.53
	IC_HSI+DS_GLCM [24]	48.87	20.89	17.786	0.7521	45.11
	DDBTC [41]	49.15	21.84	18.261	0.7425	43.31
	Modifed DDBTC (Proposed)	**49.67**	**22.46**	**18.788**	**0.7325**	**43.99**
CNN based Features	AlexNet [56]	60.67	31.86	16.597	0.5943	39.69
	VGG16 [57]	69.08	36.65	29.116	0.5200	38.93
	GoogleNet [58]	82.98	55.19	44.597	0.3311	35.59
Fusion Features	DDBTC+GoogleNet [97]	83.74	56.12	45.465	0.3188	34.88
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**86.75**	**57.54**	**46.854**	**0.2870**	**30.55**

Table 4.11 Performance Measures for ImageNet-130K Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [57]	66.05	34.95	26.640	0.5400	75.92
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
Fusion Features	DDBTC+GoogleNet [97]	80.98	55.75	41.399	0.3584	70.36
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**82.08**	**56.65**	**42.988**	**0.3067**	**66.77**

Image Dataset-10 (UKBench)

All the methods given in Table 4.1 are applied to UKBench image dataset. The results of five performance parameters for all these methods are shown in Table 4.12. It is observed that the proposed method is showing a minimum improvement of 0.43%, 0.45%, 1.15%, 0.02%, and 0.03% respectively

Table 4.12 Performance Measures for UKBench Dataset using the Proposed GoogleNet and Handcraft Feature Fusion CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [57]	97.06	86.49	75.195	0.0698	8.65
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
Fusion Features	DDBTC+GoogleNet [97]	98.46	94.67	80.323	0.0499	3.47
	GoogleNet+MDDBTC+IC_HSI_GLCM+HOG(Prposed Method)	**99.02**	**95.12**	**81.166**	**0.0398**	**2.94**

4.6 Observations

This chapter suggests a novel CBIR approach based on the fusion of CNN-based features and handcraft features. As CNN features use GoogleNet features. As a pre-processing step for the release of the GoogleNet feature, all input images were converted to 224 × 224 using VDSR network-based technology to reduce data loss during image resizing. Feature vectors from both the HSI and RGB color space are extracted from traditional features. As RGB handcraft feature DDBTC method is used. In DDBTC, a modified weight gain method for gray image conversion is used. HOG feature extraction method is applied on RGB image as shape feature.is used in the

RGB color image. All possible permutations of three components of HSI color space are used to find inter-relationships among these components. The DSP feature is calculated for extracting the local texture features. 16-level GLCM is applied on DSP for extracting the final texture features. We are exploring our proposed approach on ten different types of image databases: Corel-1K, Corel-5K, Corel-10K, VisTex, Stex, Color Brodatz, and three subsets of the ImageNet dataset (ImageNet-13K, ImageNet-65K, and ImageNet-130K), and UKBench. The results show that the proposed approach has achieved a significant performance improvement compared to different traditional and modern state-of-art methods.

Chapter 5

Modified CNN Architectures for Content Based Image Retrieval

5.1 Introduction

Most of the CNN based CBIR methods use existing pre-trained CNN architecture for extracting image features. However, these CNN models have the following shortfalls:

- CNN models that are used in various existing works are suffering from the 'Degradation Problem' because Residual Connection was either not existing or not efficiently used. To overcome this problem 'Residual Connection' must be in a proper way.
- CNN models used in existing CBIR methods have produced features that lack detailed high-level features. Moreover, these CNN models are unable to produce a hierarchical feature representation of the image.
- The conventional 2D convolution method was employed by all the CNN models in the CBIR framework, which is less informative and difficult to fit on a single GPU.
- In many of the existing CNN architectures, after convolution operation, Batch Normalization (BN) method is used to reduce the number of epochs in the training process. But the use of the BN layer makes the training process dependent on batch size. Due to this reason, the batch error will increase for smaller batch sizes.

To address these problems the present chapter proposes a total of five modified CNN models:

- Residual-GoogleNet,
- Cascade-ResNet-50,
- Group Normalized-Inception-DarkNet-53,
- Xception-DarkNet-53, and
- Shuffled-Xception-DarkNet-53.

5.2 Methodology

5.2.1 Residual-GoogleNet

GoogleNet is proposed in the year 2015 by researchers at Google. GoogleNet first introduced the concept of Inception module in CNN architecture [58]. It contains a total of 22 layers with nine Inception Modules. Each Inception model of GoogleNet has one 1×1, one 3×3, one 5×5 Convolution filter, and one 3×3 Max-Pooling layer. In spite of having good performance in CBIR, it still suffers from the 'Degradation Problem.' The residual connection can be used to overcome this problem. The proposed CNN model is a hybrid CNN model which uses both Inception and the concept of Residual operation. We are incorporating residual operation in each Inception module in GoogleNet. The basic block diagram of GoogleNet is shown in Figure 5.1

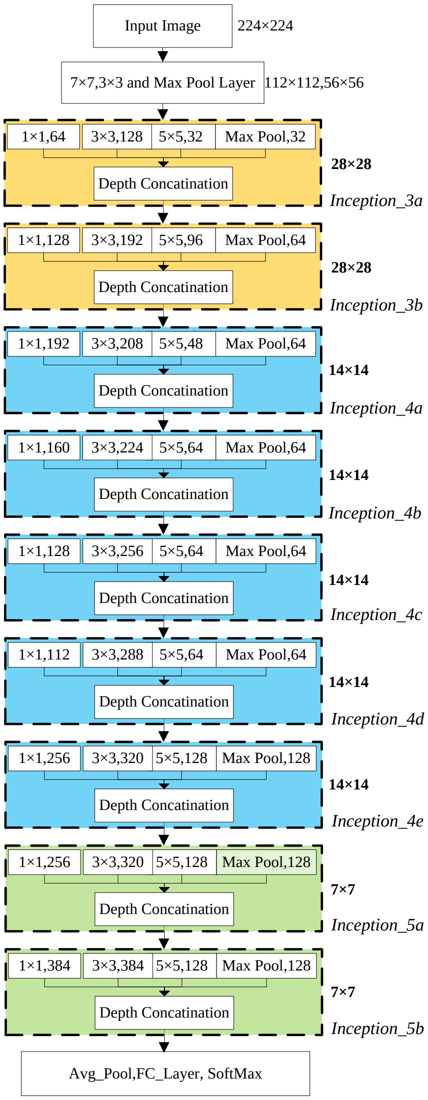


Figure 5.1 Basic Block Diagram of GoogleNet.

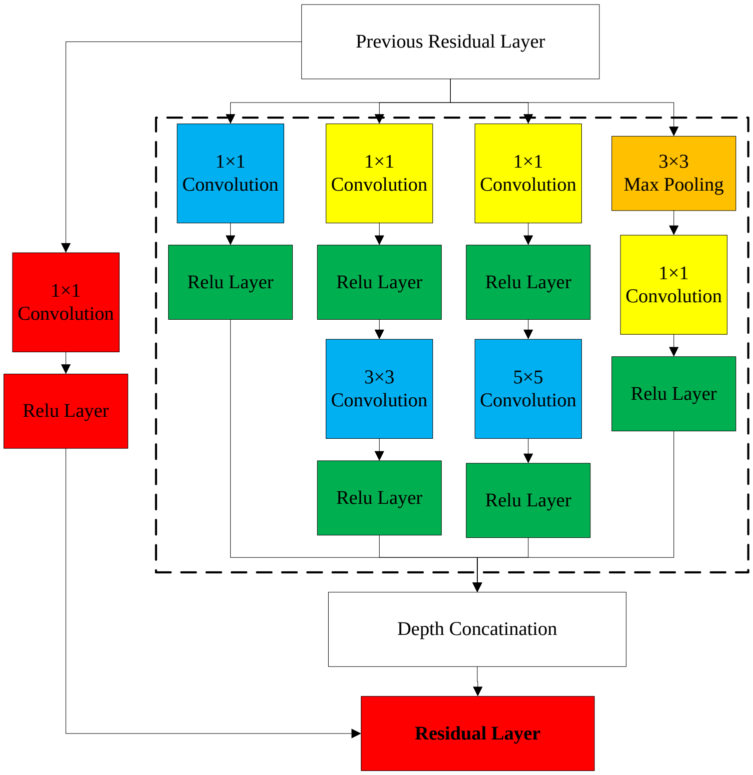


Figure 5.2 Basic Residual Block used in Residual-GoogleNet.

5.2.2 Cascade-ResNet-50

ResNet model was proposed in the year 2015. In which Residual concept was introduced with CNN structure. ResNet50 is the best performing variant of the ResNet model for CBIR which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer [59]. These 48 Convolution layers are divided among 4 blocks which contain 3,4,6,3 repetitive layers related to some specific size. Each repetitive layer contains three Convolution layers. The residual connection is applied between the input and output of each repetitive layer. The Basic architecture of ResNet-50 is shown in Table 5.1.

Proposed Model

As shown in Table 5.1, a total of four repetitive blocks exist in ResNet-50 architecture. Each block is corresponding to one particular output size. In ResNet-50, a Residual connection exists among the layers of the same block. But no connection exists among layers of different blocks. The proposed **Cascade-ResNet-50** established a residual connection among the different output size blocks, which results in the cascade structure of the proposed CNN model. The cascade structure of the proposed model is shown in Figure 5.5. In the Cascade structure, a Residual connection is established among the last layer of each block. In the proposed CNN model, the output of block-1 (56×56) and the output of the last layer of block-2 (28×28) are connected through the Residual layer. Similarly, residual connections are established between the last layers of block-2(28×28), block-3 (14×14), and block-3(14×14), block-4 (7×7). In addition to cascade structure, the initial convolution layer is also modified in the proposed CNN model. The first convolution layer of ResNet-50 has a receptive field of 7×7. A smaller receptive field such as 3×3 in the initial convolution layer retains more information. This 7×7 convolution layer is replaced by three convolution layers each with a 3×3 receptive field. The overall workflow and the detailed skeleton of the proposed model are shown in Figure 5.6 and Figure 5.7.

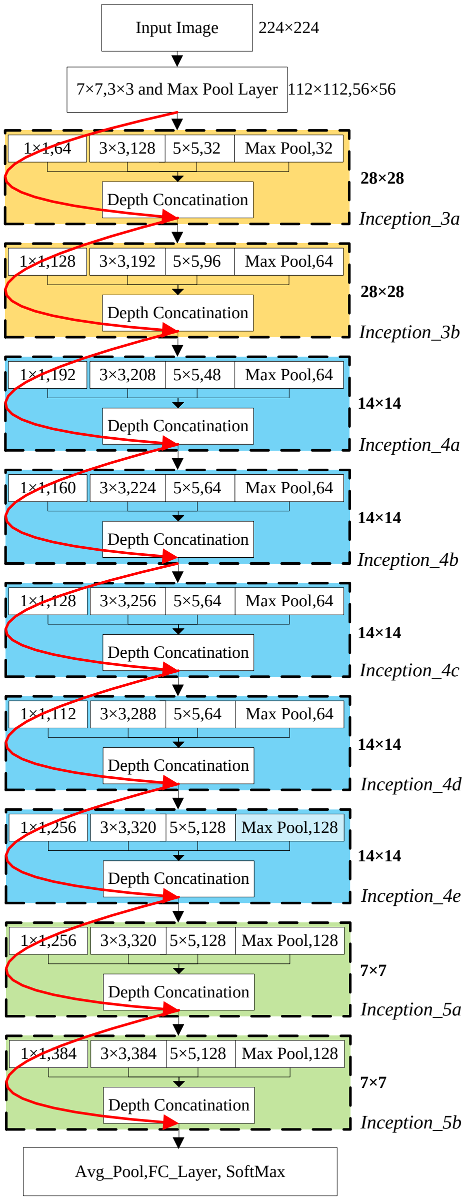


Figure 5.3 Workflow of Proposed Residual-GoogleNet.

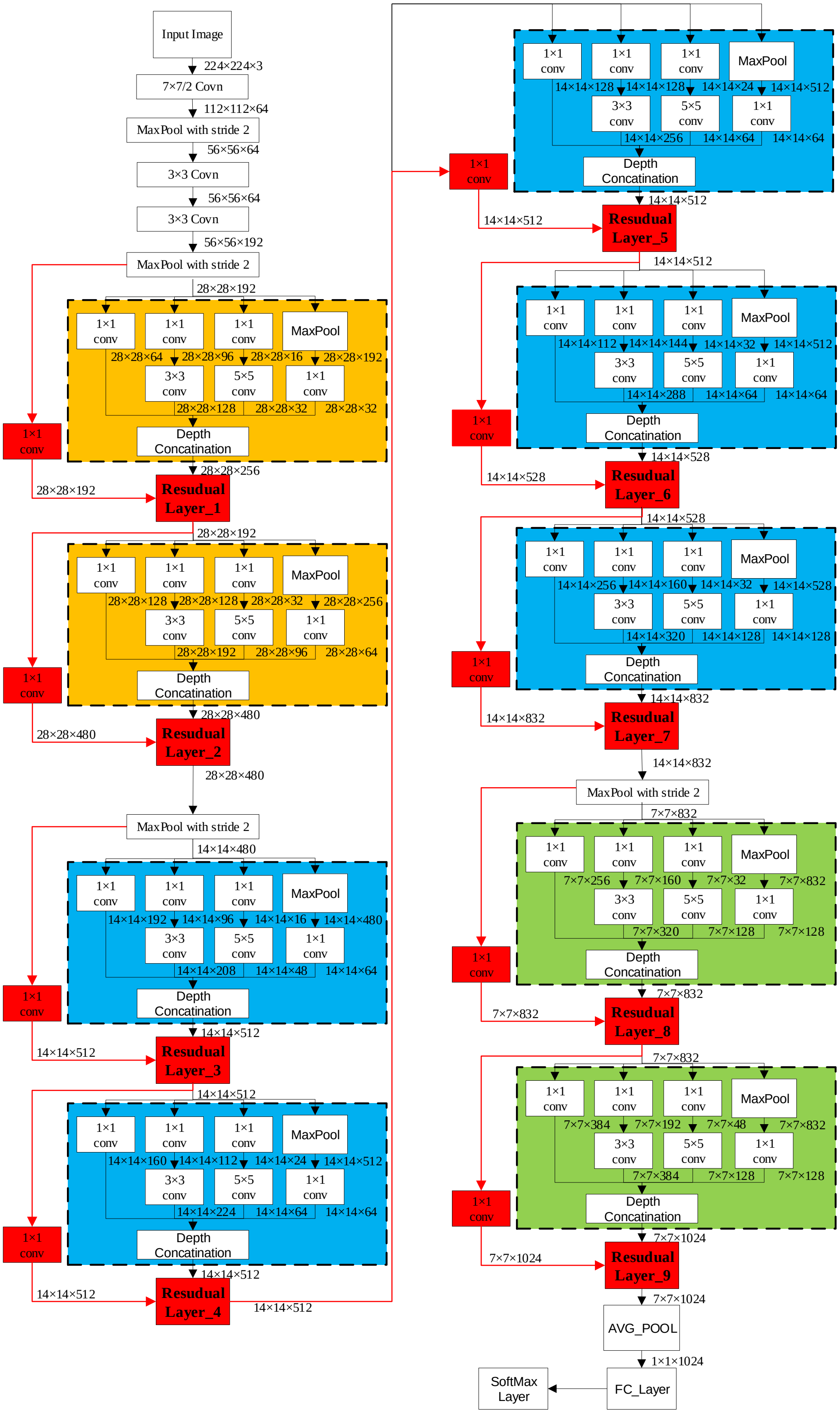


Figure 5.4 Skeleton of Proposed Residual GooleNet.

Table 5.1 ResNet-50 Architecture.

Repeatation	Type	# of filters	Size/Stride	Output Size
1	Convolution	64	**7×7**/2	112×112
1	MaxPool		2	56×56
3	Convolution	64	1×1	**56×56**
	Convolution	64	3×3	
	Convolution	256	1×1	
	Residual			
4	Convolution	128	1×1	**28×28**
	Convolution	128	3×3	
	Convolution	512	1×1	
	Residual			
6	Convolution	256	1×1	**14×14**
	Convolution	256	3×3	
	Convolution	1024	1×1	
	Residual			
3	Convolution	512	1×1	**7×7**
	Convolution	512	3×3	
	Convolution	2048	1×1	
	Residual			
	Avg_Pool			1×1
	Connected		1000	
	Softmax			

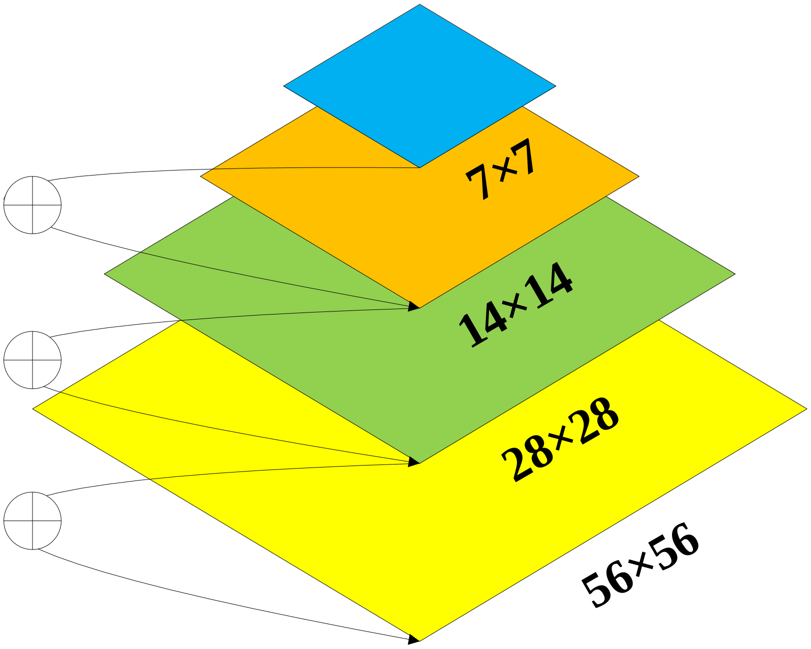


Figure 5.5 Cascade Structure of the Proposed Cascade-ResNet50 Model.

5.2.3 GroupNormalized-Inception-DarkNet-53

DarkNet-53, discussed in Section 2.5.1, gives the best performance among all pre trained CNN methods for CBIR, but has few flaws:

- DarkNet-53 uses only 3×3 filters for feature extraction which cause less informative features.
- After Convolution operation, Batch Normalization (BN) is used. As discussed in Section 5.1, the BN layer exhibits some shortcomings:
    - BN's error increases rapidly when the batch size becomes smaller, caused by inaccurate batch statistics estimation.
    - This limits BN's usage for training larger models and transferring features to computer vision tasks including detection, retrieval, segmentation, and video, which require small batches constrained by memory consumption.

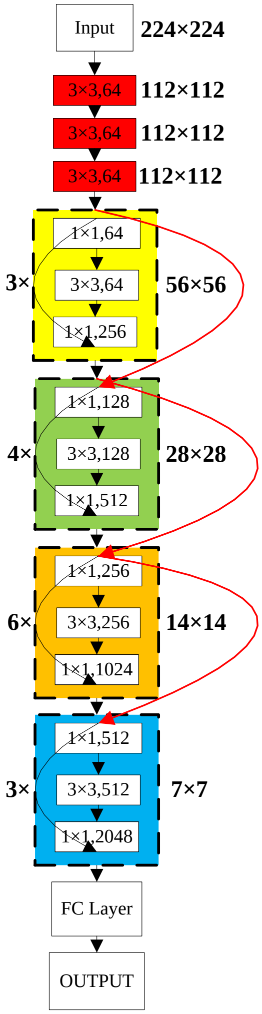


Figure 5.6 Workflow of Proposed Cascade-ResNet-50.

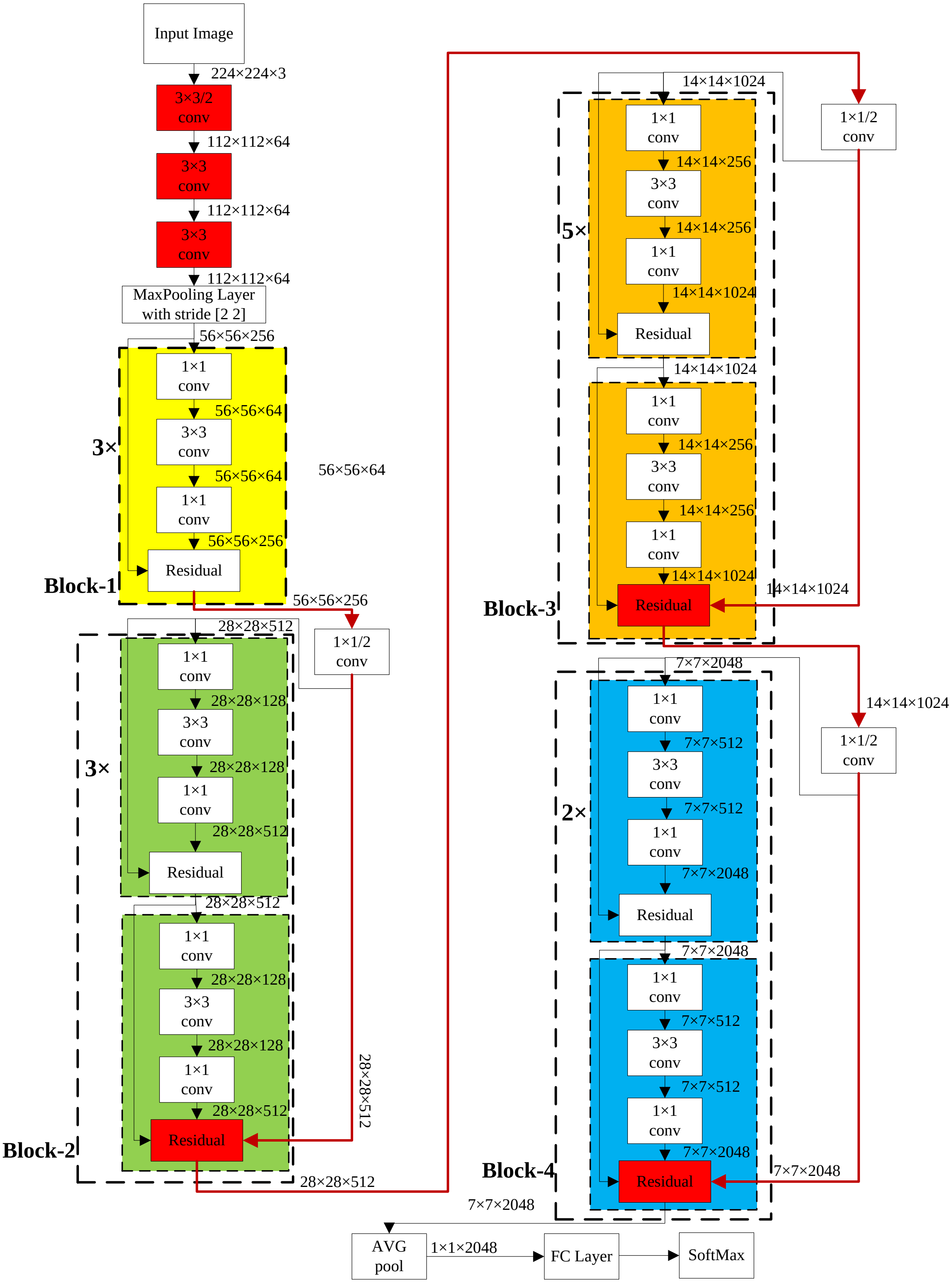


Figure 5.7 Skeleton of Proposed Cascade-ResNet-50.

Group Normalization:

Group Normalization (GN)is an effective substitute for BN [127]. In Group Normalization channels are divided into groups and for each group mean and variance is calculated for the normalization process. The performance of GN does not depend on batch sizes and it has a stable performance accuracy in a wide batch size range. The use of the GN layer, instead of the BN layer, reduces the dependency of a CNN model on batch size and other hardware necessities.

Formulation: The general formulation for all feature normalization methods is shown in using equation 5.1

$$\hat{F}_i = \frac{1}{\sigma_i}\left(F_i - \mu_i\right) \tag{5.1}$$

where, $\hat{F}$ is the normalized feature, F is the extracted features of the previous convolution layer and *i* is an index. For 2-dimensional images, *i* is represented by a 4-dimensional vector (i_N, i_C, i_H, i_W), where *N* is the batch axis, *C* is the channel axis, and *H* spatial height axes and *W* spatial width axes. This vector is indexing the extracted features in (*N*, *C*, *H*, *W*) order. The σ and μ represents stranded deviation and mean which is calculated using equation 5.2 and equation 5.3.

$$\mu_i = \frac{1}{M}\sum_{j \in P_i} F_j \tag{5.2}$$

$$\sigma_i = \sqrt{\frac{1}{M}\sum_{j \in P_i}\left(F_j - \mu_i\right) + \gamma} \tag{5.3}$$

where P_i is pixel set in which mean and standard deviation operation is performed, *M* is the size of the pixel set and γ is a small constant value. Various normalization methods use different approaches to define the pixel set P_i. In BN,Pixel set P_i is defined using equation 5.4

$$P_i = \left\{x \mid x_C = i_C\right\} \tag{5.4}$$

where i_C and x_C are the sub-index of *i* and *x* along the *C* axis. In BN, normalization operation is performed among the pixels having the same channel index. For each channel μ and σ are calculated along the (*N*, *H*, *W*) axes in BN. In the case of GN pixel set P_i is defined using equation 5.5.

$$P_i = \left\{ x \mid x_N = i_N, \left\lfloor \frac{x_C}{C/G} \right\rfloor = \left\lfloor \frac{i_C}{C/G} \right\rfloor \right\} \qquad (5.5)$$

where *G* is the number of groups having a default value of 32, *C/G* is the number of channels per group. The $\left\lfloor \frac{x_C}{C/G} \right\rfloor = \left\lfloor \frac{i_C}{C/G} \right\rfloor$ indicates the indexes *i* and *x* are in the same group of channels. In GN, normalization operation is performed along (H, W) axes and along with a group of *C/G* channels.

Proposed Model

To overcome the drawbacks of DarkNet-53 discussed above this chapter proposed a refined version of DarkNet-53, named **GroupNormalized-Inception-DarkNet-53** (GN-Inception-DarkNet-53). In the proposed model, inception layer is incorporated with the basic structure of Darknet-53. Each inception layer used in the proposed CNN model contains three 1×1 kernels, one 3×3 kernel, and one 5×5 kernel. Among three 1×1 kernels, one is used to extract features, and the remaining two are used to reduce the computational cost of 3×3 and 5×5 kernels. To stabilize and speed up the training process of the proposed network one Group Normalization (GN) layer [127], instead of the BN layer used in Darknet-53, is used after each convolution filter of the proposed inception layer. One LeakyRelu (LR) is applied after the GN layer to give non-linearity characteristics to the proposed model. The output of 1×1,3×3 and 5×5 kernels are depth concatenated to form the final output of the inception layer. The basic structure of the inception layer used in the proposed net is shown in Figure 5.8. In the proposed model total of five inception layers are used. As discussed in Section 2.5.1, Darknet-53 contains a total of five repetitive modules, each module repeats one, two, eight, eight, and four times respectively. In GN-Inception-Darknet-53, the last repetitive layer of each module is replaced by one inception layer which results in a total of five inception layers in the proposed model. The depth of output of the inception layer is reduced by 50% using convolution filters of kernel size 1×1 and then concatenated with the output of the previous inception layer. After that size of the output is reduced by 50% using a convolution layer having kernel size 1×1 with stride 2 and used for depth concatenation with the next inception layer. Whereas the output of the 3×3 kernel of the inception layer is used as input of a residual layer with the output of the previous residual layer. The output of this residual layer is considered the input of the next repetitive module. In the proposed model the overall workflow of the 3rd inception layer is shown in Figure 5.9. Among the five inception layer, the first four inception layer follow the same structure shown in Figure 5.8, whereas the last one contains one extra residual layer which performs an addition operation between the output of the 3×3 convolution layer and the output of the previous residual layer. The basic structure of the last inception layer used in the proposed net is shown in Figure 5.10. The entire Skeleton of GN-Inception- Darknet-53 is shown in Figure 5.11 and the architecture of the proposed network is given in Table 5.2. The output of the average pooling layer of the proposed net, named 'Avg_1' is used as a feature extraction layer, which gives a feature vector of length 1024.

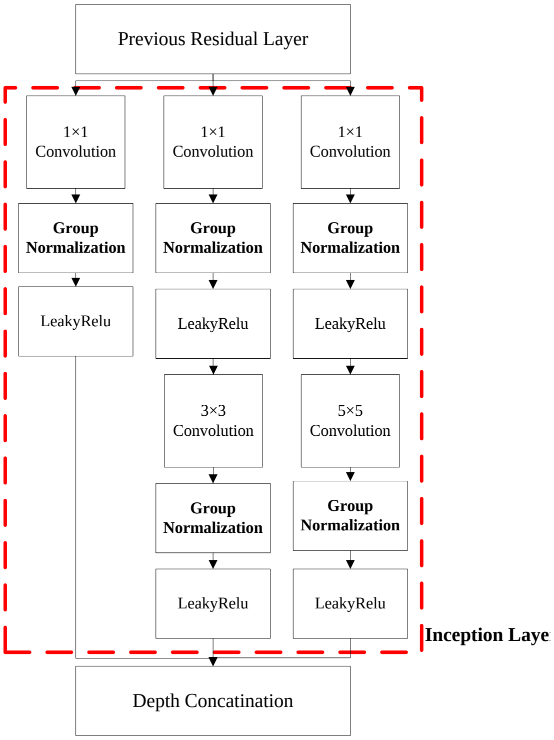


Figure 5.8 Basic Structure of Inception Layer.

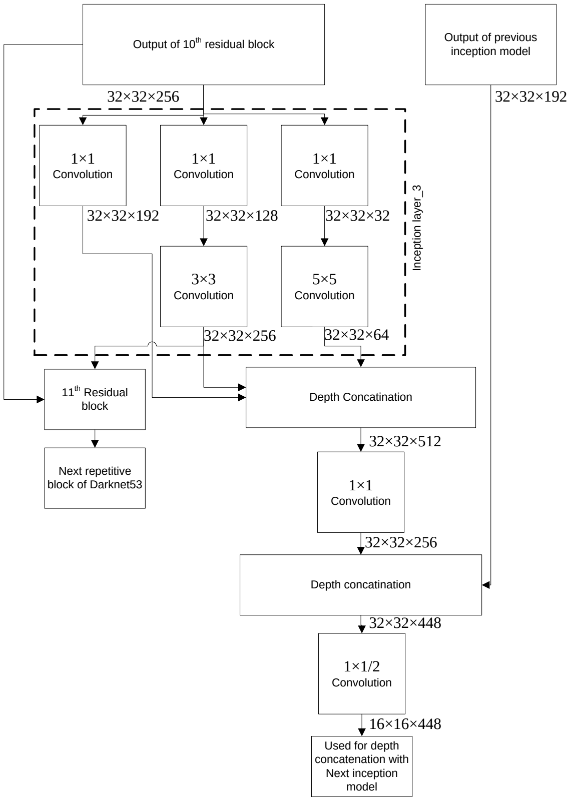


Figure 5.9 Workflow of the Proposed Inception Layer.

5.2.4 Xception-DarkNet-53

Grouped Convolution

One of the most crucial layers in every CNN model is the convolution operation. The conventional 2D convolution method was employed by the majority of pre-trained models, which have the following drawbacks:

- To achieve good training and testing performance on different Image datasets, a CNN model having multiple kernels per layer needs to be constructed, resulting in a larger neural network.
- In traditional 2D convolution, each kernel is convolved with the previous layer's full feature maps, resulting in a large number of convolution operations including some redundant convolution.
- Deeper models will be difficult to train since they will be difficult to fit on a single GPU.

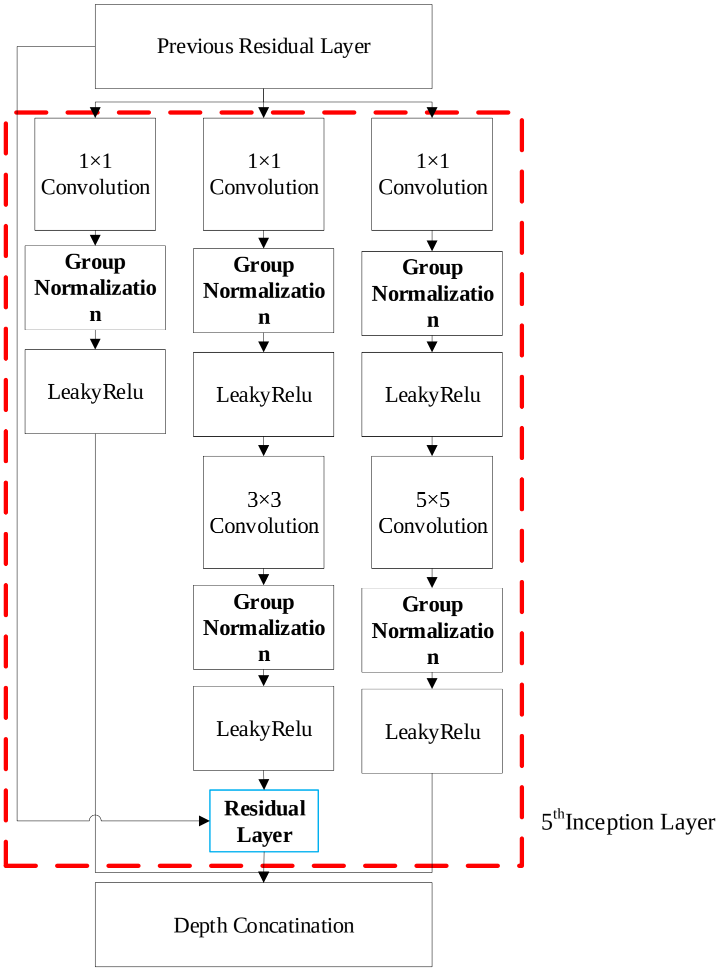


Figure 5.10 Structure of Last Inception Layer.

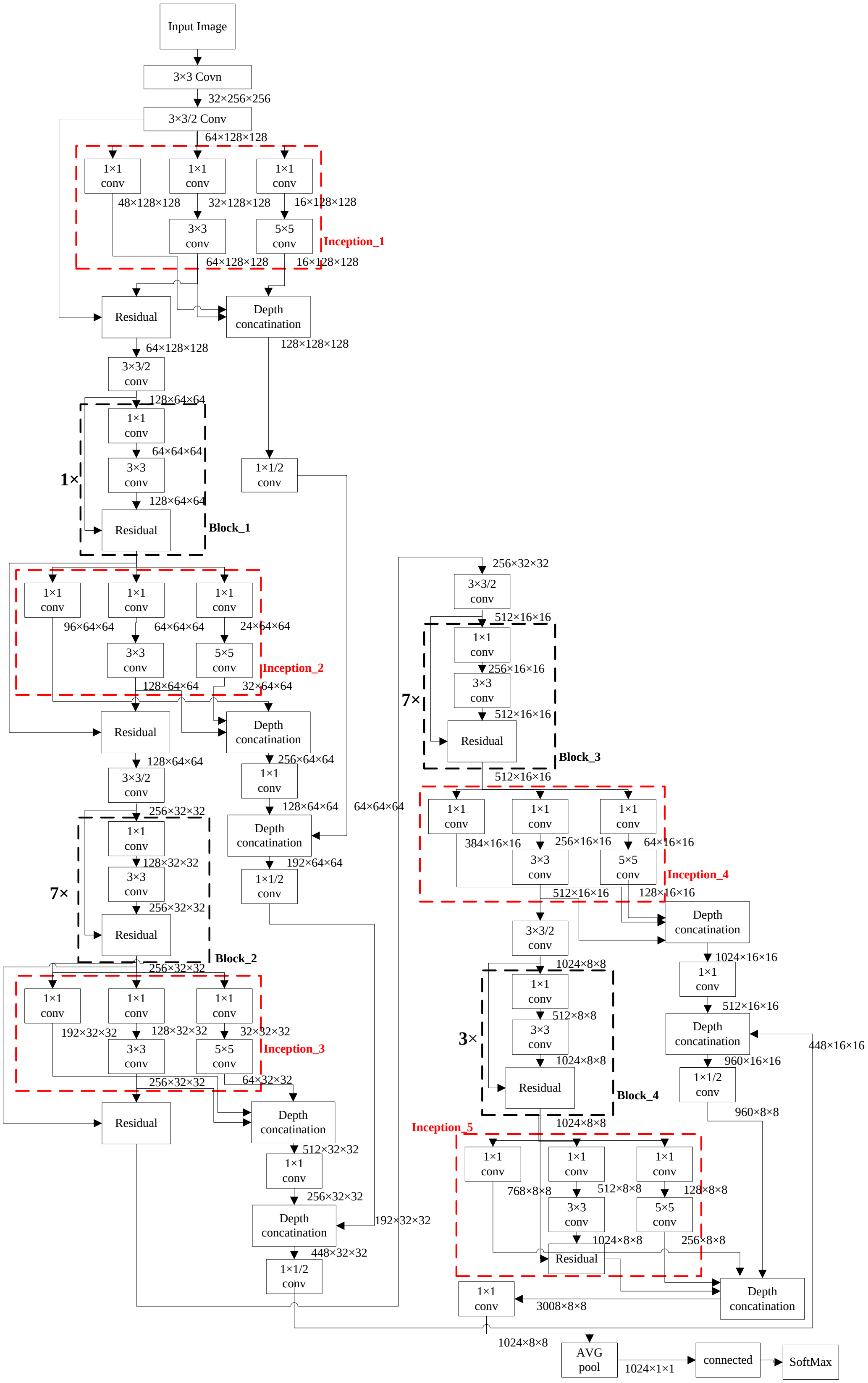


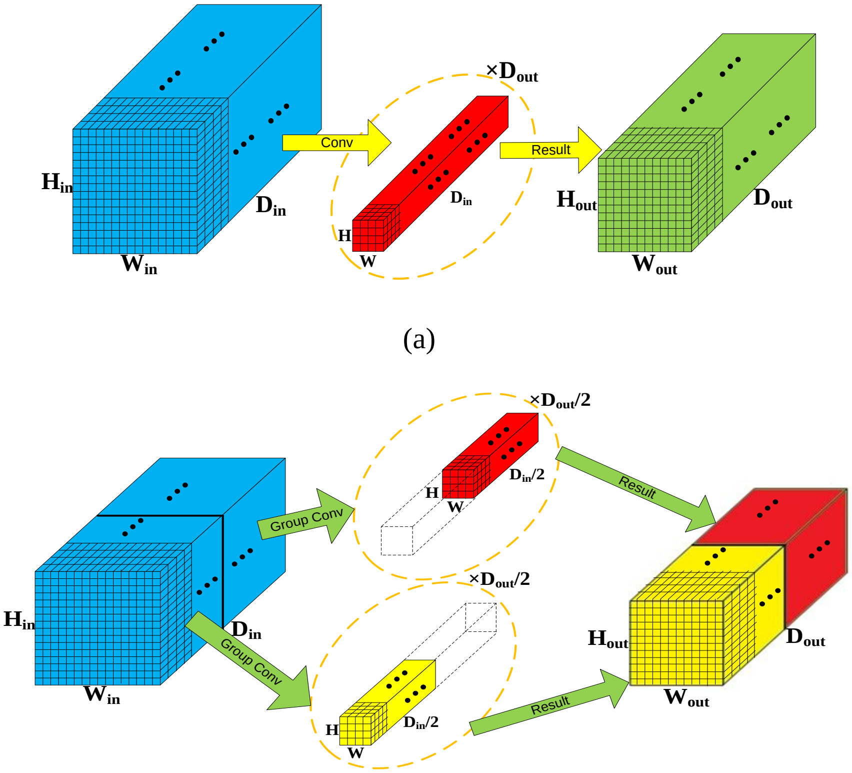
Figure 5.11 Skeleton of Proposed GN-Inception-Darknet-53.

Table 5.2 GN-Inception-Darknet-53 Architecture.

Name	# Reprtation	Type	Inputsize	Outputsize	Stride	#1×1	#3×3 Reduce	#3×3	#5×5 Reduce	#5×5
	1	Convolution	256×256	32×256×256	1			32		
	1	Convolution	32×256×256	64×128×128	2			64		
Inception_1	1	Inception	64×128×128			48	32	64	16	16
		Dept Concatination	3 Inputs	128×128×128						
	1	Convolution	128×128×128	64×64×64	2	64				
	1	Residual	2 Inputs	64×128×128						
	1	Convolution	64×128×128	128×64×64	2			128		
Block_1	1	Convolution	128×64×64		1		64	128		
		Residual	2 Inputs	128×64×64						
Inception_2	1	Inception	128×64×64			96	64	128	24	32
		Dept Concatination	3 Inputs	256×64×64						
	1	Convolution	256×64×64	128×64×64	1	128				
	1	Dept Concatination	2 Inputs	192×64×64						
	1	Convolution	192×64×64	192×32×32	2	192				
	1	Residual	2 Inputs	128×64×64						
	1	Convolution	128×64×64	256×32×32	2			256		
Block_2	7	Convolution	256×32×32		1		128	256		
		Residual	2 Inputs	256×32×32						
Inception_3	1	Inception	256×32×32			192	128	256	32	64
		Dept Concatination	3 inputs	512×32×32						
	1	Convolution	512×32×32	256×32×32	1	256				
	1	Dept Concatination	2 inputs	448×32×32						
	1	Convolution	448×32×32	448×16×16	2	448				
	1	Residual	2 Inputs	256×32×32						
	1	Convolution	256×32×32	512×16×16	2			512		
Block_3	7	Convolution	512×16×16		1		256	512		
		Residual	2 Inputs	512×16×16						
Inception_4	1	Inception	512×16×16			384	256	512	64	128
		Dept Concatination	3 inputs	1024×16×16						
	1	Convolution	1024×16×16	512×16×16	1	512				
	1	Dept Concatination	2 inputs	960×16×16						
	1	Convolution	960×16×16	960×8×8	2	960				
	1	Residual	2 Inputs	512×16×16						
	1	Convolution	512×16×16	1024×8×8	2			1024		
Block_4	3	Convolution	1024×8×8		1		512	1024		
		Residual	2 Inputs	1024×8×8						
Inception_5	1	Inception	1024×8×8			768	512	1024	128	256
		Residual	2 inputs	1024×8×8						
		Dept Concatination	4 inputs	3008×8×8						
	1	Convolution	3008×8×8	1024×8×8	1	1024				
Avg_1	1	Average Pooling	1024×8×8	1024×1×1						
	1	Connected								
	1	SoftMax								

One alternative to traditional 2D convolution is Group Convolution which is first introduced in 2012 [56]. In this method, filters are divided into the required number of groups and each group of filters is convolved with the respective portion of the feature map of the previous layer, not with the entire input layer. The entire process of Group Convolution is shown in Figure 5.12. The following benefits can be achieved using Group Convolution:

1. Using Group Convolution, a wider CNN can be constructed by replicating one modular block of the filter group.
2. The training process has become more efficient. Because the filters are separated into multiple groups, convolution operation for each group can indeed be performed by a different GPU separately. This technique enables parallel model training over many GPUs.
3. The model is more cost-effective, i.e. the number of convolution parameters is less in the case of Group Convolution. As shown in Figure 5.12(a), the total count of parameters is $H \times W \times H_{out} \times W_{out} \times D_{in} \times D_{out}$ in the case of traditional 2D convolution. In the case of Figure. 5.12(b), where Group Convolution has been performed by separating the filters into 2 groups, a total of $(H \times W \times H_{out} \times W_{out} \times D_{in} /2 \times D_{out} /2) \times 2$ parameters are required, which is half of the previous method. In general, if H_{out}, W_{out}, and D_{out} are the height, width, and depth of the output image, D_{in} is the depth of input images, and $H \times W$ is the kernel size then $H \times W \times H_{out} \times W_{out} \times D_{in} \times D_{out}$ is the total parameter count in 2D convolution. In the case of Group Convolution with 'G' filter groups, the required number of parameters is $(h \times w \times H \times W \times D_{in}/G \times D_{out}/G) \times G = (H \times W \times D_{in} \times D_{out})/G$. This observation concludes that concerning 2D convolution, the number of parameters in Group Convolution is reduced by a factor of the number of filter groups used in Group Convolution. So, by increasing the number of filter groups, convolution parameters can be significantly reduced.
4. Group convolutions appear to be learning better data representations. Essentially, the correlation among kernels of various filter groups is typically quite low, implying that each group is learning a distinct data representation.
5. Group Convolution improved the interrelationship across filters in adjacent layers. Filters with strong correlation are trained in a more organized manner in filter groups, resulting in a more effective feature vector.



(a)

(b)

Figure 5.12 (a) Conventional 2D Convolution. (b) Group Convolution with Two Groups.

Proposed Model

In this section, another refined version of DarkNet-53 is proposed, named **‘Xception-DarkNet-53’**. To extract more comprehensive information from the input image, the Xception concept is integrated with the darknet-53 basic framework. Each Xception module employs kernels of sizes 1×1, 3×3, and 5×5. For all three kernel sizes, the 'Group Convolution' technique is used instead of the standard 2D convolution used in Darknet-53. Group Convolution is utilized to make the presented method more informative by enhancing the interconnection among the filters of the neighbouring layers and to develop the effectiveness of the proposed CNN model's training process, as discussed in this section. Each of the 1×1, 3×3, and 5×5 size Group convolution filters is repeated three times in a stacked manner, yielding a total of nine group convolution filters in a single Xception module. To make the training process less batch size-dependent, one Group Normalization (GN) layer is utilized after each group convolution filter, rather than the traditional Batch Normalization operation. After the GN layer, LeakyRelu (LR) is employed to give the proposed model non-linearity properties. The output of the last 1×1, 3×3, and 5×5 group convolution filters are concatenated to generate the Xception module's final output. Figure 5.13 depicts the Xception module's construction in the proposed network.

As shown in Table 2.1, the core structure of Darknet-53 can be distributed into five modules, which are associated with one, two, eight, eight, and four repetitive layers respectively. Each layer of a module produces an output of the same size. The output of the last layer of the respective module contains the final characteristics of that size. In this CNN model, One Xception module is used at the end of each of the five modules, resulting in a total of five Xception modules. The output of the last layer of each module is used as input for the proposed Xception module. The depth of each Xception module output is lowered to the closest power of two using a 1×1 convolution filter, i.e., 64,128,256,512, and 1024 respectively. This output is Depth concatenated with the previous Xception module's output. The output size of this operation is lowered by a factor of 0.5 using one 1×1 convolution layer with stride 2. This output is used for depth concatenation with the next Xception module output. The flowchart of the 4th Xception module is shown in Figure 5.14. For the last Xception module, the depth concatenated output is combined with the output last repetitive layer of the 5th block which creates an output feature of depth 3008. This depth is further reduced to 1024 depth using a convolution filter of size 1×1. The detailed skeleton of Xception-Darknet-53 is given in Figure 5.15. Table 5.3 shows the overall architecture of the proposed CNN model.

5.2.5 Shuffled-Xception-DarkNet-53

Channel-Shuffling

Group convolution considerably reduces computing cost and extracts more detailed features by guaranteeing that each convolution performs only on the relevant input channel group. When many group convolutions are stacked together, one side consequence is that the outputs from a given channel are only derived from a small proportion of the input channels. Two layered group convolution layers are depicted in Figure 5.16(a). It is obvious that a group's outputs are exclusively related to the group's inputs. This feature prevents the information from flowing between channel groups, causing representation to deteriorate. The input and output channels will be fully connected if we allow group convolution to receive input data from multiple groups. To achieve that goal, the channels in each group can be divided into subgroups. After that, for each group in the following layer, subgroups of different groups obtained from the previous group convolution can be used as input. This concept is shown in Figure 5.16(b). A channel shuffle operation [128, 129] can be used to accomplish this in an efficient and elegant manner as shown in Figure 5.16(c). Suppose a group convolutional layer has G groups and each group has N channels, resulting in a total of G×N output channels. Then to perform the channel shuffle process on this layer, first, the output channel of this group convolution layer must be reshaped into the (G, N) dimension. Then the channels are shuffled by taking the transpose of the reshaped output channel. After that, the shuffled channels are flattened and used as the following layer input. It's worth noting that even if the two convolutions have different number of groups, the operation still works. Furthermore, channel shuffle is differentiable, allowing it to be integrated into network architectures for end-to-end training. With successive group convolutional layers and the channel shuffle operation, more powerful architectures may be built.

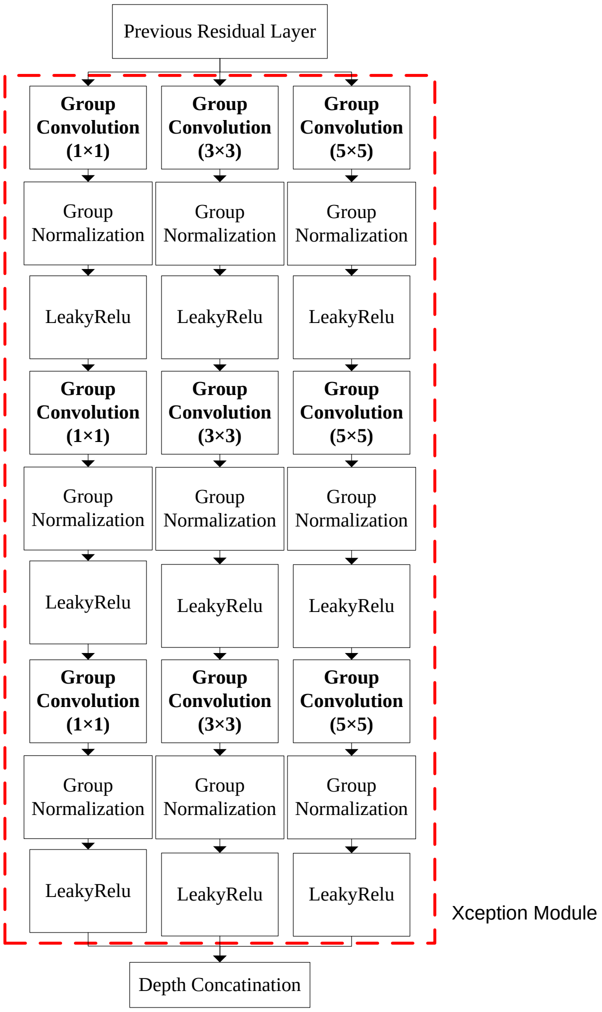
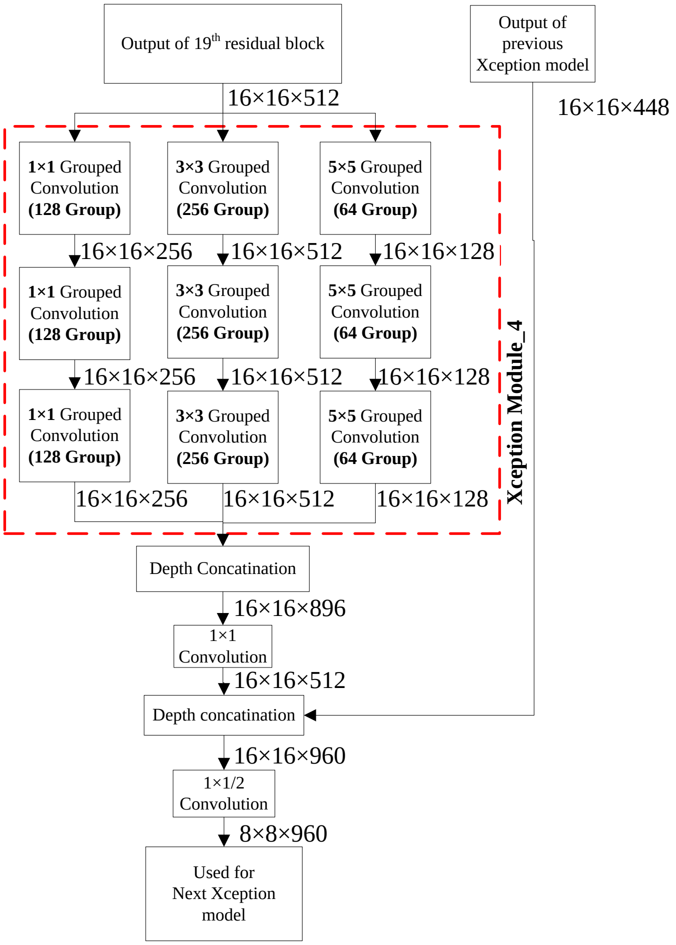


Figure 5.13 Basic Structure of Xception Module.



Output of 19th residual block
Output of previous Xception model
16×16×512
16×16×448
1×1 Grouped Convolution (128 Group)
3×3 Grouped Convolution (256 Group)
5×5 Grouped Convolution (64 Group)
16×16×256
16×16×512
16×16×128
1×1 Grouped Convolution (128 Group)
3×3 Grouped Convolution (256 Group)
5×5 Grouped Convolution (64 Group)
16×16×256
16×16×512
16×16×128
1×1 Grouped Convolution (128 Group)
3×3 Grouped Convolution (256 Group)
5×5 Grouped Convolution (64 Group)
16×16×256
16×16×512
16×16×128
Xception Module_4
Depth Concatination
16×16×896
1×1 Convolution
16×16×512
Depth concatination
16×16×960
1×1/2 Convolution
8×8×960
Used for Next Xception model

Figure 5.14 Workflow of the Proposed Xception-Darknet-53

Proposed model

In this section the third modified version of DarkNet-53 is proposed, named **‘Shuffled-Xception-DarkNet-53’**. In the proposed model, the Xception concept along with channel shuffling operation is integrated with the DarkNet53 basic framework. This proposed CNN model has introduced a total of five additional modules named ‘Shuffled-Xception module’. Each of these modules employs kernels of sizes 1×1, 3×3, and 5×5. In this module also the 'Group Convolution' technique is used for all kernel sizes, instead of the standard 2D convolution. Each of the 1×1, 3×3, and 5×5 size Group Convolution filters is repeated three times in a series connection, yielding a total of nine Group Convolution filters in a single Shuffled-Xception module. As discussed above, the use of two Group Convolution in a series connection causes a loss of information flow among the channels of different groups. Channel Shuffling operation is used between every two stacked Group Convolution of the same size, resulting in a total of six ‘*Channel Shuffle’* layers in each module. The use of successive Group Convolutional layers and the channel shuffling operation produce an efficient and informative CNN architecture. In this proposed module also, instead of the BN layer, the GN layer is used. After the GN layer, LeakyRelu (LR) is employed to give the proposed model non-linearity properties. The output of the last 1×1, 3×3, and 5×5 Group Convolution filters are concatenated to generate the Shuffled-Xception module's final output. Figure 5.17 depicts the Shuffled-Xception module's construction in the proposed network. The overall workflow of the proposed CNN model is the same as Xception-DarkNet-53, discussed in section 5.2.4. The Detailed skeleton of the proposed CNN model is shown in Figure 5.18.

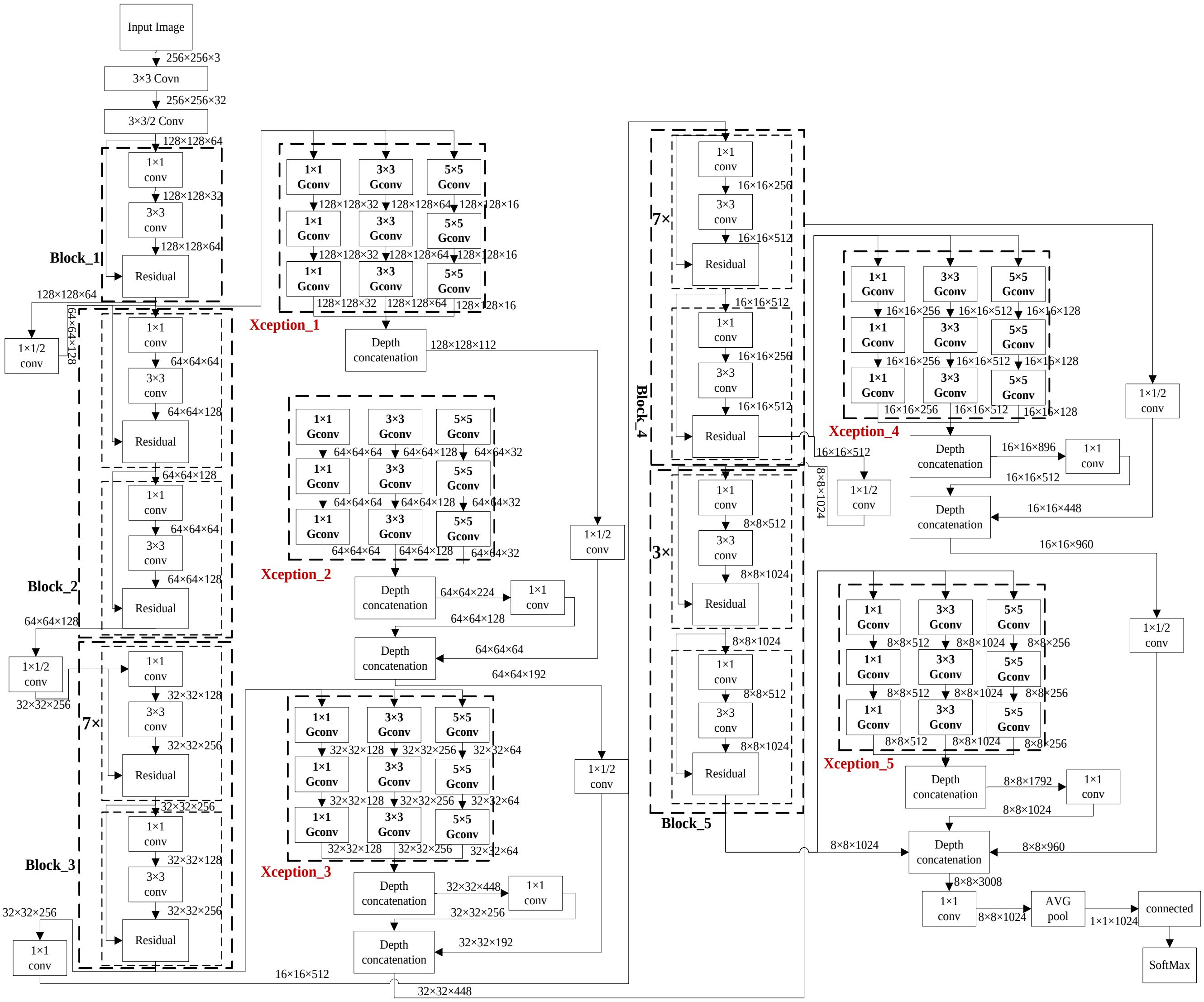


Figure 5.15 Basic Skeleton of the Proposed Xception-Darknet-53.

Table 5.3 Xception Darknet-53 Architecture.

Name	# Reprtation	Type	Inputsize	Outputsize	Stride	#1×1	#3×3 Reduce	#3×3	#5×5 Reduce	#5×5
	1	Convolution	256×256×3	256×256×32	1			32		
	1	Convolution	256×256×32	128×128×64	2			64		
Block_1	1	Convolution	128×128×64		1		32	64		
		Residual	2 Inputs	128×128×64						
Xception_1	1	Group Convolution	128×128×64			32		64		16
		Group Convolution	Depends on Filter			32		64		16
		Group Convolution	Depends on Filter			32		64		16
		Dept Concatination	3 Inputs	128×128×112						
	1	Convolution	128×128×112	64×64×64	2	64				
	1	Convolution	128×128×64	64×64×128	2			128		
Block_2	2	Convolution	64×64×128		1		64	128		
		Residual	2 Inputs	64×64×128						
Xception_2	1	Group Convolution	64×64×128			64		128		32
		Group Convolution	Depends on Filter			64		128		32
		Group Convolution	Depends on Filter			64		128		32
		Dept Concatination	3 Inputs	64×64×224						
	1	Convolution	64×64×224	64×64×128	1	128				
	1	Dept Concatination	2 Inputs	64×64×192						
	1	Convolution	64×64×192	32×32×192	2	192				
	1	Convolution	64×64×128	32×32×256	2			256		
Block_3	8	Convolution	32×32×256		1		128	256		
		Residual	2 Inputs	32×32×256						
Xception_3	1	Group Convolution	32×32×256			128		256		64
		Group Convolution	Depends on Filter			128		256		64
		Group Convolution	Depends on Filter			128		256		64
		Dept Concatination	3 inputs	32×32×448						
	1	Convolution	32×32×448	32×32×256	1	256				
	1	Dept Concatination	2 inputs	32×32×448						
	1	Convolution	32×32×448	16×16×448	2	448				
	1	Convolution	32×32×256	16×16×512	2			512		
Block_4	8	Convolution	16×16×512		1		256	512		
		Residual	2 Inputs	16×16×512						
Xception_4	1	Group Convolution	16×16×512			256		512		128
		Group Convolution	Depends on Filter			256		512		128
		Group Convolution	Depends on Filter			256		512		128
		Dept Concatination	3 inputs	16×16×896						
	1	Convolution	16×16×896	16×16×512	1	512				
	1	Dept Concatination	2 inputs	16×16×960						
	1	Convolution	16×16×960	8×8×960	2	960				
	1	Convolution	16×16×512	8×8×1024	2			1024		
Block_5	3	Convolution	8×8×1024		1		512	1024		
		Residual	2 Inputs	8×8×1024						
Xception_5	1	Group Convolution	8×8×1024			512		1024		256
		Group Convolution	Depends on Filter			512		1024		256
		Group Convolution	Depends on Filter			512		1024		256
		Dept Concatination	3 Inputs	8×8×1792						
		Convolution	8×8×1792	8×8×1024	1	1024				
		Dept Concatination	3 Inputs	8×8×3008						
	1	Convolution	8×8×3008	8×8×1024	1	1024				
Avg_pool	1	avg_pool	8×8×1024	1×1×1024						
	1	Connected								
	1	SoftMax								

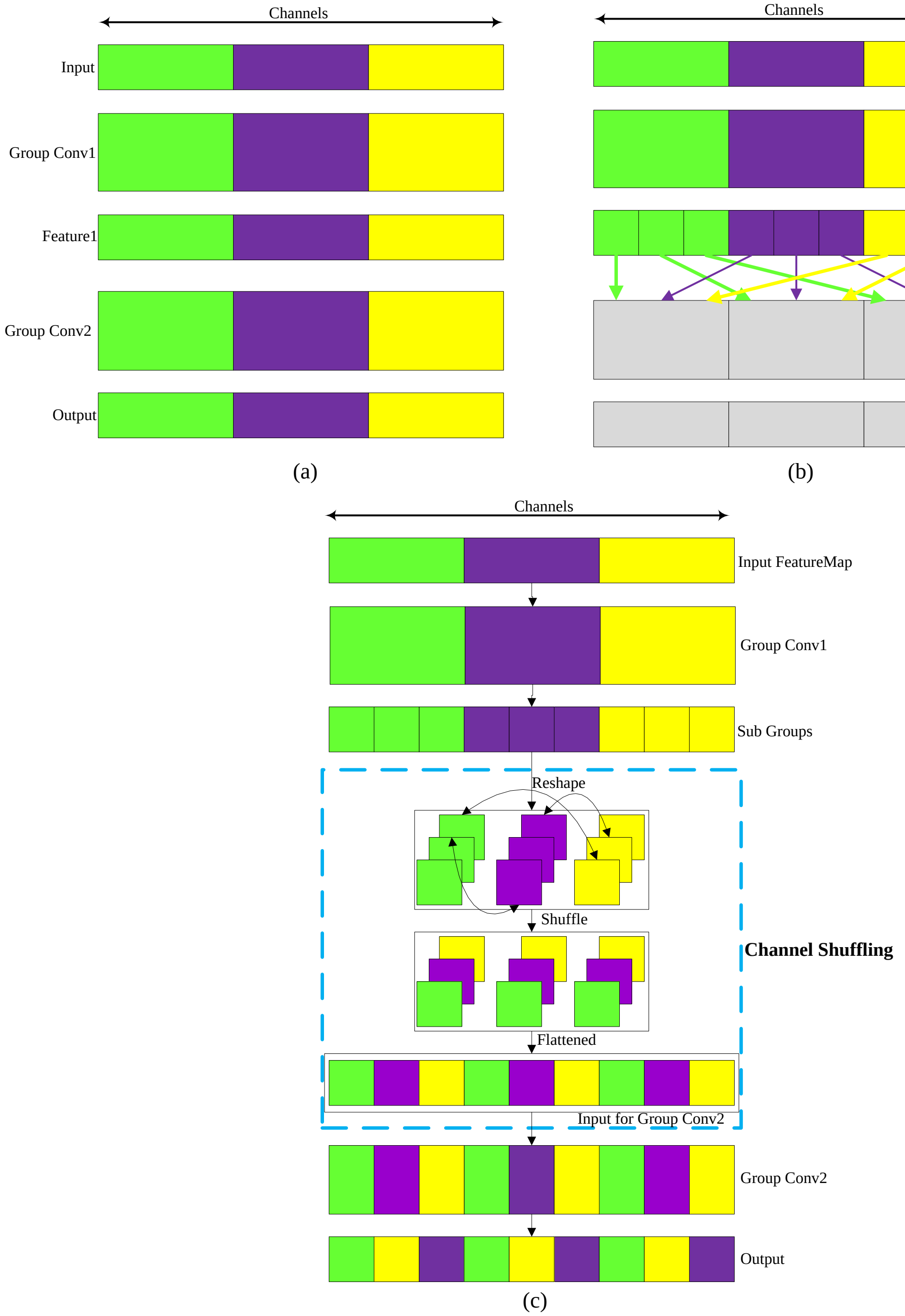


Figure 5.16 (a) Two Stacked Group Convolution Layers with Three groups. No Information Flow Among the Channels (b) Fully Related Input and Output Channels by Subgrouping and using Different Subgroups together as Input (c) Channel Shuffling: A Possible Implementation to (b).

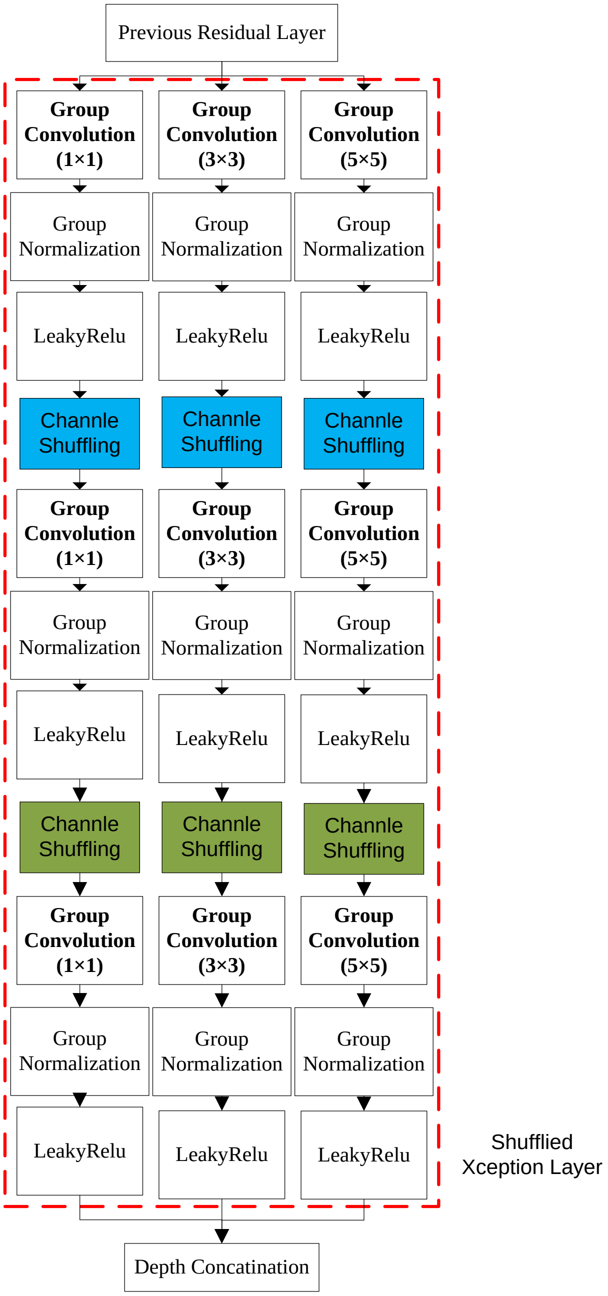


Figure 5.17 Basic Architecture of Shuffled-Xception Module.

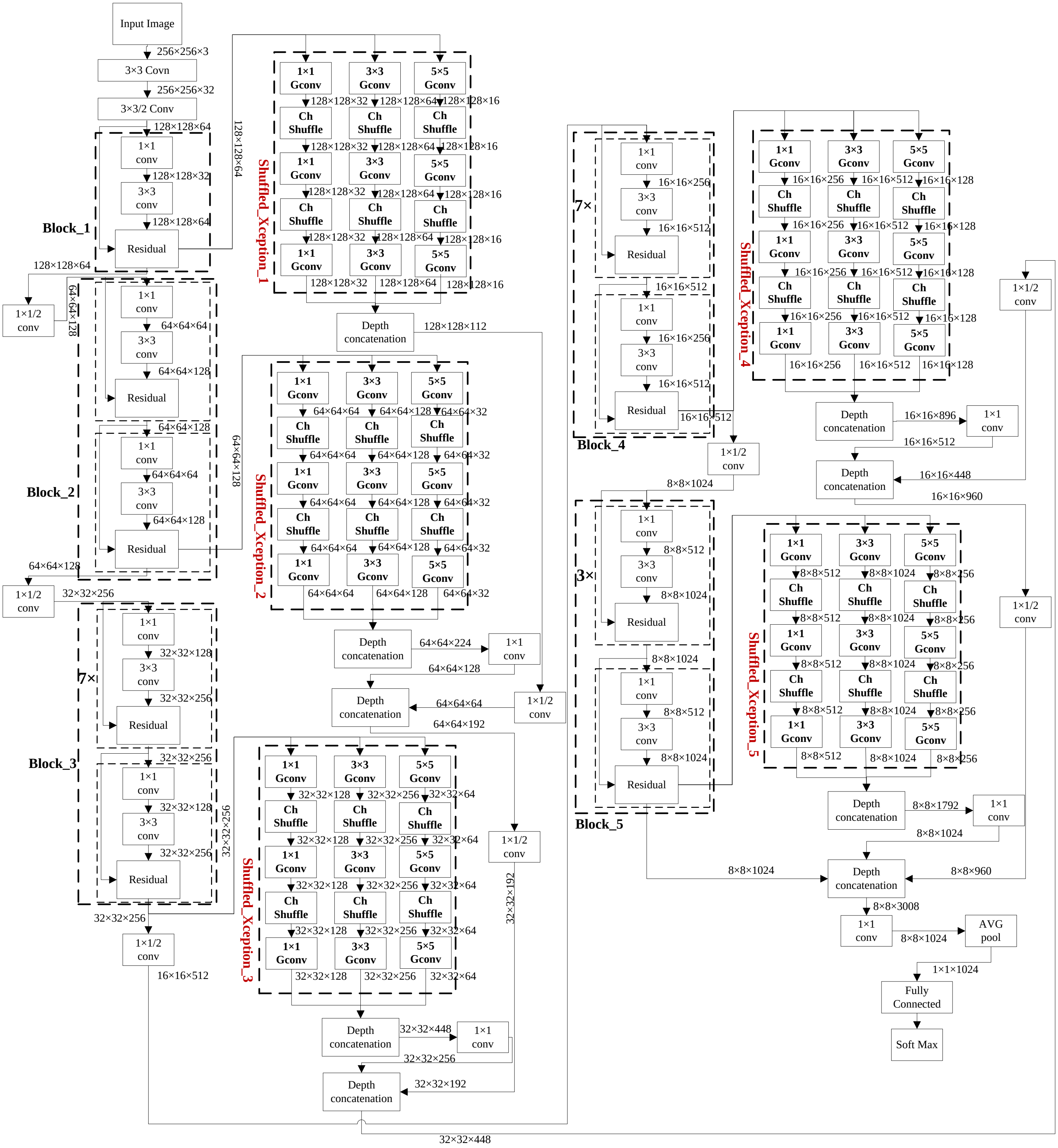


Figure 5.18 Basic Structure of Proposed Shuffled-Xception-DarkNet-53.

5.3 Experimental Results and Discussions

In this chapter also all the ten image datasets which are described and given in Chapter-1 are considered and used to evaluate the five proposed 'refined CNN models' features for CBIR. The ten image datasets are: Corel-1K, Corel-5K, Corel-10K, VisTex, STex, Color Brodatz, ImageNet-13K, ImageNet-65K, ImageNet-130K, and UKBench. The performance of these CNN models is also determined based on the same five parameters used in Chapter-3 and Chapter-4: APR, ARR, F-Measure, ANMRR, and TMRE. All the experimental setup and hyperparameter values are the same as given in section 4.5.1.

5.3.1 Residual-GoogleNet Results

Table 5.4 shows the sixteen different methods that are used to evaluate the performance of the proposed Residual-GoogleNet. Because of the space limitation only four out of ten image datasets: Corel-10K, Color Brodatz , ImageNet-130K and UKBench resultant tables are given, but the statistics of all the ten image databases are discussed here in this Chapter. The consolidated results of all the methods for all ten image datasets are given in Chapter-7.

Table 5.4 Different Methods used for Evaluating the Performance of the Proposed Residual-GoogleNet CBIR System.

Type of Method	Mehod Name
Hand Craft Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP[24]
	ColorHist_HSI+ULBP[24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
CNN based Features	AlexNet[56]
	VGG-16 [57]
	GoogleNet [58]
	Residual-GoogleNet (Proposed)

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 5.4 are executed on the Corel-1K dataset. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated and the For Corel-1K, the proposed method is showing a minimum improvement of 0.92%, 1.34%, 1.88%,0.14%, and 5.17% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 2.05%, 2.84%, 4.05%, 2.57%, and 11.92% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.5. The proposed method is showing a minimum improvement of 3.42%, 2.97%, 5.49%, 2.34%, and 8.53% for the five performance measures respectively.

Table 5.5 Performance Measures for Core-10K Dataset using the Proposed Residual-GoogleNet CBIR System.

		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Craft Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP[24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP[24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
CNN based Features	AlexNet[56]	43.90	19.57	16.597	0.7543	98.38
	VGG-16 [57]	71.07	36.26	31.031	0.5660	97.80
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Residual-GoogleNet (Proposed)	**89.93**	**61.17**	**48.834**	**0.2976**	**24.66**

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 5.4 is executed on the VisTex dataset. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For VisTex image dataset, the proposed method is showing a minimum improvement of 0.13%, 4.76%, 1.75%, 0.13%, and 1.73% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 2.36%, 3.53%, 5.88%, 5.94%, and 2.11% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.6. The proposed method is showing a minimum improvement of 0.11%, 2.29%, 1.95%, 0.63%, and 0.59% for the five performance measures respectively.

Table 5.6 Performance Measures for Color Brodatz Dataset using the Proposed Residual-GoogleNet CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Craft Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP[24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP[24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
CNN based Features	AlexNet[56]	89.47	61.25	51.820	0.3212	10.88
	VGG-16 [57]	96.51	76.19	61.121	0.1785	5.75
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Residual-GoogleNet (Proposed)	**99.76**	**96.50**	**70.191**	**0.0234**	**3.74**

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.41%, 1.99%, 3.33%, 3.38%, and 17.36% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 1.10%, 1.77%, 0.84%, 5.45%, and 4.08% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.7. The proposed method is showing a minimum improvement of 3.93%, 2.35%, 1.94%, 3.83%, and 4.39% for the five performance measures respectively.

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.8. The proposed method is showing a minimum improvement of 0.86%, 0.74%, 1.54%,1.52%, and 0.05% for the five performance measures respectively

Table 5.7 Performance Measures for ImageNet-130K Dataset using the Proposed Residual-GoogleNet CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Craft Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP[24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP[24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
CNN based Features	AlexNet[56]	59.76	30.74	15.717	0.6043	78.44
	VGG-16 [57]	66.05	34.95	26.640	0.5400	75.92
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Residual-GoogleNet (Proposed)	**84.10**	**56.46**	**43.977**	**0.3228**	**67.56**

Table 5.8 Performance Measures for UKBench Dataset using the Proposed Residual-GoogleNet CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Craft Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.6169
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.4688
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.2919
	LBP [26]	65.38	40.72	41.780	0.5800	105.4931
	ULBP [26]	67.99	42.19	43.757	0.5082	97.0159
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.6928
	ColorHist_HSI+CS_LBP[24]	68.22	43.78	44.930	0.4400	110.3731
	ColorHist_HSI+ULBP[24]	70.07	44.77	45.741	0.4792	96.0601
	LECoP [28]	89.14	70.10	64.763	0.2338	28.3646
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.0029
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.1722
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.0488
CNN based Features	AlexNet[56]	96.12	85.07	74.340	0.0900	11.8475
	VGG-16 [57]	97.06	86.49	75.195	0.0698	8.6486
	GoogleNet [58]	98.08	93.92	79.436	0.0579	3.8347
	Residual-GoogleNet (Proposed)	**98.94**	**94.66**	**80.562**	**0.0426**	**2.8574**

5.3.2 Cascade-ResNet-50 Results

Table 5.9 shows the name of all seventeen methods that are used to evaluate the performance of the proposed Cascade-ResNet-50. Among them a total of fifteen existing methods have already been mentioned in Table 5.4. The additional existing method in Table 5.9 is ResNet-50.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 5.9 are executed on the Corel-1K dataset. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For Corel-1K, the proposed method is showing a minimum improvement of 2.15%,1.67%,2.25%,1.80% for the first four performance measures respectively. For TMRE GoogleNet is showing the best performance. For Corel-5K image dataset, The proposed method is showing a minimum improvement of 1.63%, 7.65%, 6.06%, 7.21%, and 22.42% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.10. The proposed method is showing a minimum improvement of 3.34%, 6.59%, 5.85%, 6.86%, and 14.27% for the five performance measures respectively.

Table 5.9 Different Methods used for Evaluating the performance of the Proposed Cascade-ResNet-50 CBIR System.

Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP[24]
	ColorHist_HSI+ULBP[24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
CNN based Features	AlexNet[56]
	VGG-16 [57]
	ResNet-50[59]
	GoogleNet [58]
	Cascade-ResNet-50 (Proposed)

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 5.9 is executed on VisTex and STex datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For VisTex, the proposed method is showing a minimum improvement of 0.74%,7.26%,3.84%,3.35%, and 2.30% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 0.34%,4.75%,1.76%,1.85%, and 0.78% for the five performance measures respectively.

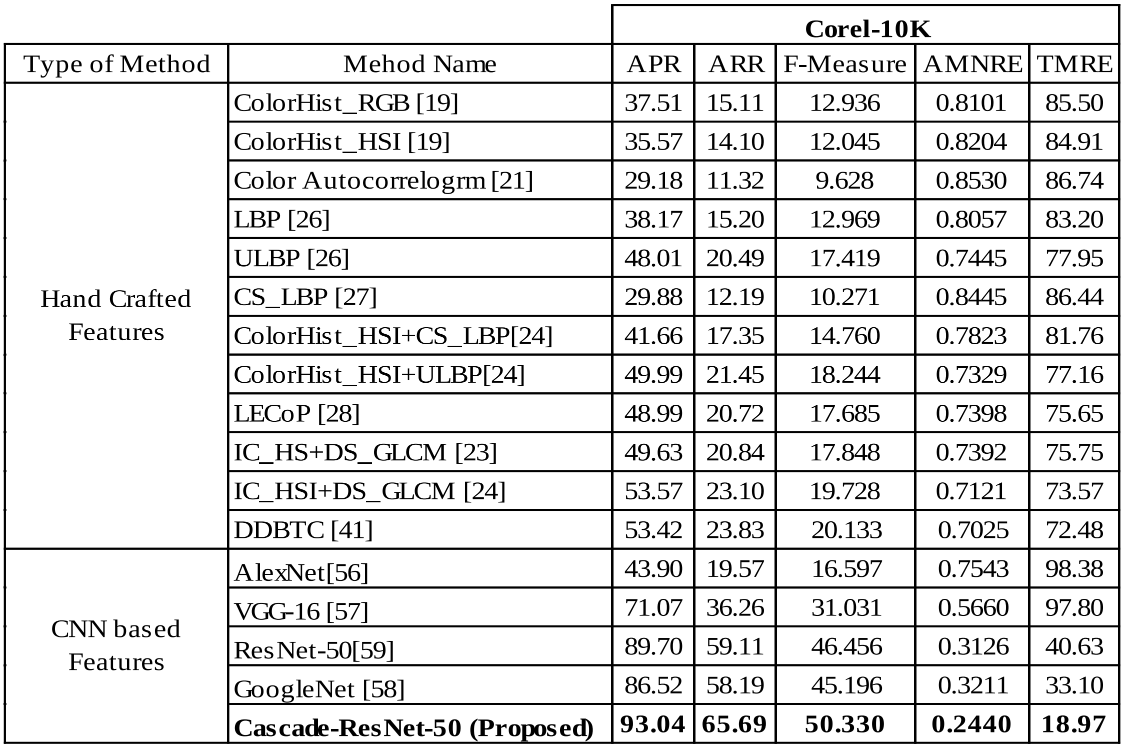
Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.11. The proposed method is showing an improvement of 0.14%, 1.67%, 1.46%, and 0.12%, and 0.29% for the five performance measures respectively.

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For ImageNet-13K, tThe proposed method is showing an enhancement of 1.57%, 3.90%, 5.68%, 4.84%, and 24.03% for the five performance measures respectively. For ImageNet-65K, The proposed method is showing a minimum improvement of 1.98%, 2.59%, 6.19%, 7.80%, and 8.21% for the five performance measures respectively.

Table 5.10 Performance Measures for Core-10K Dataset using the Proposed Cascade-ResNet-50 CBIR System.



		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP[24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP[24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
CNN based Features	AlexNet[56]	43.90	19.57	16.597	0.7543	98.38
	VGG-16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet-50[59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Cascade-ResNet-50 (Proposed)	**93.04**	**65.69**	**50.330**	**0.2440**	**18.97**

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.12. The proposed method is showing a minimum improvement of 0.97%,1.89%, 2.20%, 4.28%, and 6.82% for the five performance measures respectively.

Image Dataset-10 (UKBench)

All the methods given in Table 5.9 is executed on the UKBench dataset. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.13. The proposed method is showing a minimum improvement of 0.46%, 1.63%,1.64%,0.84%, and 0.06% for the five performance measures respectively.

Table 5.11 Performance Measures for Color Brodatz Dataset using the Proposed Cascade-ResNet-50 CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP[24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP[24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
CNN based Features	AlexNet[56]	89.47	61.25	51.820	0.3212	10.88
	VGG-16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet-50[59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Cascade-ResNet-50 (Proposed)	**99.83**	**95.88**	**69.847**	**0.0288**	**2.14**

Table 5.12 Performance Measures for ImageNet-130K Dataset using the Proposed Cascade-ResNet-50 CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP[24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP[24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
CNN based Features	AlexNet[56]	59.76	30.74	15.717	0.6043	78.44
	VGG-16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet-50[59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Cascade-ResNet-50 (Proposed)	**84.95**	**57.98**	**45.103**	**0.2898**	**62.88**

Table 5.13 Performance Measures for UKBench Dataset using the Proposed Cascade-ResNet-50 CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP[24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP[24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
CNN based Features	AlexNet[56]	96.12	85.07	74.340	0.0900	11.85
	VGG-16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet-50[59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Cascade-ResNet-50 (Proposed)	**99.34**	**95.76**	**81.226**	**0.0269**	**2.15**

5.3.3 GN-Inception-DarkNet-53 Results

Table 5.14 shows the name of all Seventeen existing methods that are used to compare the performance of the proposed GN-Inception-DarkNet-53. Among them, the first sixteen methods are already mentioned in Table 5.9. The additional existing method in Table 5.14 is DarkNet-53.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 5.14 are executed on the Corel-1K and Corel-5K image datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For Corel-1K, the proposed method is showing a minimum improvement of 0.27%, 5.09%, 0.92%, 2.48%, and 1.10% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 2.51%,18.42%, 10.43%,14.94%, and 6.06% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.15. The proposed method is showing a minimum improvement of 3.51%, 19.91%, 14.02%, 17.08%, and 28.34% for the five performance measures respectively.

Table 5.14 Different Methods used for Evaluating the Performance of the Proposed GN-Inception-DarkNet-53 CBIR System.

Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP[24]
	ColorHist_HSI+ULBP[24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
CNN based Features	AlexNet[56]
	VGG-16 [57]
	ResNet-50[59]
	GoogleNet [58]
	Darknet-53[64]
	GN-Inception-Darknet-53 (Proposed)

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 5.14 is executed on VisTex and STex datasets. The five parameters APR, ARR, F-Measure, ANMRR and TMRE for all these methods are computed. For VizTex, the proposed method is showing a minimum improvement of 0.52%, 2.74%, 1.27%, 1.49%, and 1.63% for the five performance measures respectively. For STex, The proposed method is showing a minimum improvement of 0.25%, 3.85%, 2.09%, 2.57%, 1.44% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.16. The proposed method is showing a minimum improvement of 0.10%, 1.98%, 0.90%, 1.09%, and 1.25% for the five performance measures respectively.

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods for both ImageNet-13K and ImageNet-65K are evaluated. For ImageNet-13K, the proposed method is showing a minimum improvement of 0.92%,18.00%, 9.24%, 12.62%, and 27.78% for the five performance measures respectively. For ImageNet-65K, The proposed method is showing a minimum improvement of 2.90%, 15.21%, 11.27%, 14.33%, and 20.76% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.17. The proposed method is showing a minimum improvement of 2.61%, 15.38%, 10.15%, 13.94%, and 27.25% for the five performance measures respectively.

Image Dataset-10 (UKBench)

All the methods given in Table 5.14 is executed on UKBench dataset. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.18. The proposed method is showing a minimum improvement of 0.40%, 2.72%, 2.04%, 0.45%, and 0.01% for the five performance measures respectively

Table 5.15 Performance Measures for Core-10K Dataset using the Proposed GN-Inception-DarkNet-53 CBIR System.

		Core-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP[24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP[24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
CNN based Features	AlexNet[56]	43.90	19.57	16.597	0.7543	98.38
	VGG-16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet-50[59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Darknet-53[64]	94.72	71.38	53.762	0.2008	40.81
	GN-Inception-Darknet-53 (Proposed)	**98.23**	**91.29**	**63.047**	**0.0300**	**5.04**

Table 5.16 Performance Measures for Color Brodatz Dataset using the Proposed GN-Inception-DarkNet-53 CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP[24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP[24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
CNN based Features	AlexNet[56]	89.47	61.25	51.820	0.3212	10.88
	VGG-16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet-50[59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Darknet-53[64]	99.81	96.87	69.772	0.0172	2.86
	GN-Inception-Darknet-53 (Proposed)	**99.92**	**98.85**	**70.412**	**0.0063**	**1.08**

Table 5.17 Performance Measures for ImageNet-130K Dataset using the GN-Proposed Inception-DarkNet-53 CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP[24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP[24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
CNN based Features	AlexNet[56]	59.76	30.74	15.717	0.6043	78.44
	VGG-16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet-50[59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Darknet-53[64]	92.60	71.05	53.695	0.2123	60.04
	GN-Inception-Darknet-53 (Proposed)	**95.21**	**86.43**	**60.421**	**0.0729**	**33.06**

Table 5.18 Performance Measures for UKBench Dataset using the Proposed GN-Inception-DarkNet-53 CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP[24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP[24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
CNN based Features	AlexNet[56]	96.12	85.07	74.340	0.0900	11.85
	VGG-16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet-50[59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Darknet-53[64]	99.54	97.06	81.710	0.0156	1.60
	GN-Inception-Darknet-53 (Proposed)	**99.94**	**99.78**	**83.203**	**0.0111**	**1.35**

5.3.4 Xception-DarkNet-53 Results

The results obtained from the proposed Xception-DarkNet-53 is compared with seventeen existing methods given in Table 5.14 for all ten image datasets.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods on both Corel-1K and Corel-5K are evaluated. For Corel-1K, the proposed method is showing a minimum improvement of 0.94%, 9.69%, 4.02%, 4.22%, 19.36% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 3.11%, 24.39%,14.84%, 16.25%, and 9.60% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.19. The proposed method is showing a minimum improvement of 5.08%, 26.48%, 17.94%, 19.34%, and 31.79% for the five performance measures respectively.

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods on the texture datasets: VisTex and STex are computed. For VisTex, the proposed method is showing a minimum improvement of 0.59%, 2.96%, 1.64%,1.64%, and 1.76% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 0.75%, 7.71%,4.86%,4.36%, and 1.91% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.20. The proposed method is showing a minimum improvement of 0.16%,2.62%, 1.22%, 1.42%, and 1.28% for the five performance measures respectively.

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

For ImageNet-13K and ImageNet-65K datasets, results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.14%,19.88%, 10.05%, 12.90%, and 35.84% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 4.21%,17.53%, 12.90%, 17.27%, 28.60% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.21. The proposed method is showing a minimum improvement of 3.16%,15.82%, 10.78%, 14.59%, 37.73% for the five performance measures respectively.

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.22. The proposed method is showing a minimum improvement of 0.43%, 2.82%, 2.12%, 1.51%, and 0.03% for the five performance measures respectively.

Table 5.19 Performance Measures for Core-10K Dataset using the Proposed Xception-DarkNet-53 CBIR System.

		Core-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP[24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP[24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
CNN based Features	AlexNet[56]	43.90	19.57	16.597	0.7543	98.38
	VGG-16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet-50[59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Darknet-53[64]	94.72	71.38	53.762	0.2008	40.81
	Xception-Darknet-53 (Proposed)	**99.81**	**97.86**	**65.644**	**0.0074**	**1.63**

Table 5.20 Performance Measures for Color Brodatz Dataset using the Proposed Xception-DarkNet-53 CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP[24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP[24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
CNN based Features	AlexNet[56]	89.47	61.25	51.820	0.3212	10.88
	VGG-16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet-50[59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Darknet-53[64]	99.81	96.87	69.772	0.0172	2.86
	Xception-Darknet-53 (Proposed)	**99.97**	**99.49**	**70.635**	**0.0030**	**1.05**

Table 5.21 Performance Measures for ImageNet-130K Dataset using the Proposed Xception-DarkNet-53 CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP[24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP[24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
CNN based Features	AlexNet[56]	59.76	30.74	15.717	0.6043	78.44
	VGG-16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet-50[59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Darknet-53[64]	92.60	71.05	53.695	0.2123	60.04
	Xception-Darknet-53 (Proposed)	**95.76**	**86.87**	**60.839**	**0.0664**	**22.69**

Table 5.22 Performance Measures for UKBench Dataset using the Proposed Xception-DarkNet-53 CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP[24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP[24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
CNN based Features	AlexNet[56]	96.12	85.07	74.340	0.0900	11.85
	VGG-16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet-50[59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Darknet-53[64]	99.54	97.06	81.710	0.0156	1.60
	Xception-Darknet-53 (Proposed)	**99.98**	**99.88**	**83.263**	**0.0005**	**1.00**

5.3.5 Shuffled-Xception-DarkNet-53 Results

The performance of the proposed Shuffled-Xception-DarkNet-53 is compared with the same seventeen existing methods given in Table 5.14 for all ten image datasets.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods on Corel-1K and Corel-5K are evaluated. For Corel-1K, the proposed method is showing a minimum improvement of 0.94%, 9.74%, 4.05%, 4.23%, and 19.43% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 3.11%, 24.43%, 14.88%, 16.27%, and 9.84% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.23. The proposed method is showing a minimum improvement of 5.12%, 26.67%, 18.07%, 19.48%, and 31.89% for the five performance measures respectively.

Table 5.23 Performance Measures for Core-10K Dataset using the Proposed Shuffled-Xception-DarkNet-53 CBIR System.

		Core-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP[24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP[24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
CNN based Features	AlexNet[56]	43.90	19.57	16.597	0.7543	98.38
	VGG-16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet-50[59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Darknet-53[64]	94.72	71.38	53.762	0.2008	40.81
	Shuffled Xception-Darknet-53 (Proposed)	**99.84**	**98.04**	**65.729**	**0.0060**	**1.53**

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

The results of the five parameters APR, ARR, F-Measure, ANMRR and TMRE for all these methods for VisTex and STex are evaluated. For VisTex, the proposed method is showing a minimum improvement of 0.68%, 3.31%, 2.05%, 1.74%, and 1.87% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 1.10%, 8.71%, 5.23%, 5.25%, and 2.09% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods for Color Brodatz dataset are given in Table 5.24. The proposed method is showing a minimum improvement of 0.17%, 2.70%, 1.32%, 1.56%, and 1.28% for the five performance measures respectively.

Table 5.24 Performance Measures for Color Brodatz Dataset using the Proposed Shuffled-Xception-DarkNet-53 CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP[24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP[24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
CNN based Features	AlexNet[56]	89.47	61.25	51.820	0.3212	10.88
	VGG-16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet-50[59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Darknet-53[64]	99.81	96.87	69.772	0.0172	2.86
	Shuffled Xception-Darknet-53 (Proposed)	**99.98**	**99.57**	**70.709**	**0.0016**	**1.04**

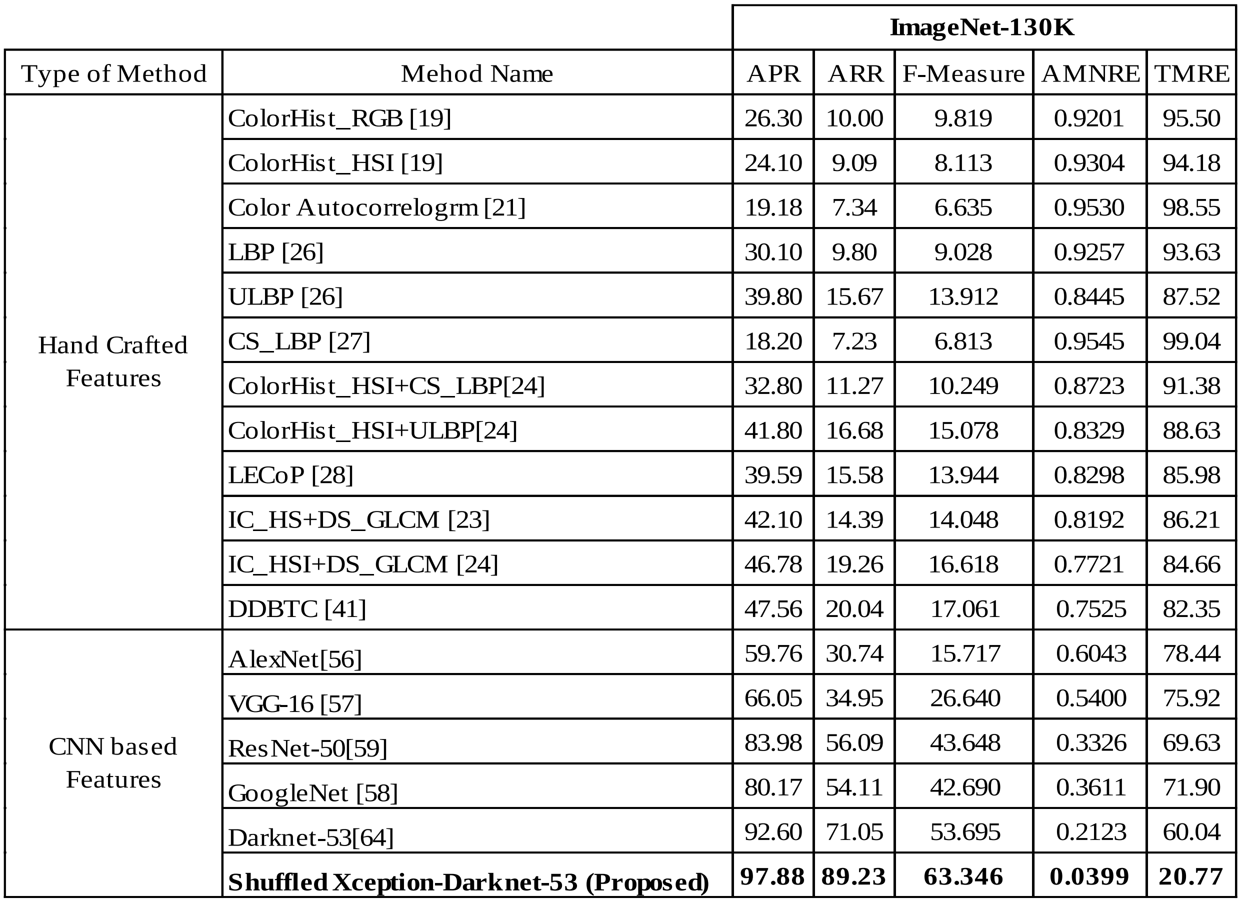
Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods on ImageNet-13K and ImageNet-65K are evaluated. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.23%, 20.41%, 10.36%, 12.95%, and 40.41% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 4.90%, 18.64%, 14.33%, 17.91%, 32.62% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.25. The proposed method is showing a minimum improvement of 5.28%,18.19%, 14.57%,17.24%, and 39.67% for the five performance measures respectively.

Table 5.25 Performance Measures for ImageNet-130K Dataset using the Proposed Shuffled-Xception-DarkNet-53 CBIR System.



		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP[24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP[24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
CNN based Features	AlexNet[56]	59.76	30.74	15.717	0.6043	78.44
	VGG-16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet-50[59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Darknet-53[64]	92.60	71.05	53.695	0.2123	60.04
	Shuffled Xception-Darknet-53 (Proposed)	**97.88**	**89.23**	**63.346**	**0.0399**	**20.77**

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 5.26. The proposed method is showing a minimum improvement of 0.44%, 2.83%, 2.13%, 1.52%, and 0.03% for the five performance measures respectively.

Table 5.26 Performance Measures for UKBench Dataset using the Proposed Shuffled-Xception-DarkNet-53 CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP[24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP[24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
CNN based Features	AlexNet[56]	96.12	85.07	74.340	0.0900	11.85
	VGG-16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet-50[59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Darknet-53[64]	99.54	97.06	81.710	0.0156	1.60
	Shuffled Xception-Darknet-53 (Proposed)	**99.98**	**99.88**	**83.263**	**0.0005**	**1.00**

5.4 Observations

In this chapter a total of five refined CNN models are proposed. First modified CNN model is named as '*Residual GoogleNet*' which is a modified version of GoogleNet. In this proposed model a skip connection is established in each Inception layer in GoogleNet. An improved version of Resnet-50 named '*Cascade-ResNet-50*' is also proposed in this chapter. The proposed model has established a residual connection among the different output size blocks, which results in the cascade structure of the proposed CNN model. This Chapter also proposed three modified version of DarkNet-53, which is the best performing existing CNN model. The first refined version of darknet-53 is '*GN-Inception-DarkNet-53*' in which a total of five proposed inception modules are incorporated with basic structure of DarkNet-53. Each inception layer used in the proposed CNN model contains three 1×1 kernels, one 3×3 kernel, and one 5×5 kernel. After each convolution layer Group Normalization (GN) layer, instead of the BN layer used in DarkNet-53. Another refinement of DarkNet-53 which is proposed in this chapter is ‘*Xception-DarkNet-53*’. Xception concept is incorporated with the basic structure of DarkNet53 which is an extension of Inception concept. In each Xception module, kernels of sizes 1×1, 3×3, and 5×5 are used. But in Xception module instead of the typical 2D convolution used in Darknet-53, the 'Group Convolution' operation is used for all three kernel sizes. Each of the 1×1, 3×3, and 5×5 size Group convolution filters is stacked three times, generating a total of nine group convolution filters in one Xception module. The final modified version of DarkNet-53 which is proposed in this chapter is ‘*Shuffled-Xception-DarkNet-53*’. In this model ‘Channel Shuffle’ operation along with Xception concept is used. In this CNN model also the same Xception module is used but in between every two stacked Group Convolution of the same size, one Channel Shuffling layer is used to maintain the information flow between them. All the refined CNN models are giving better performance than the existing CNN models. Among all the proposed and existing CNN models, Shuffled- Xception-DarkNet-53 is showing the best performance with respect to all the five parameters for all ten datasets.

Chapter 6

Content Based Image Retrieval based on Feature Fusion of Refined CNN Models and Handcraft Features

6.1 Introduction

All the fusion-based methods discussed in literature and Chapter 4, used existing CNN models to extract high-level features. As discussed in Chapter 5 all these existing Pre-trained models are having some technical gaps, refined the existing CNNs are used for CBIR, where it is shown that the results are improved with the refined CNNs features. So it is proved that modified CNN models can be used in place of existing CNN models to extract high level features in turn for CBIR. Shakarami et. al [130] proposed a new feature fusion-based CBIR method where handcrafted features are combined with features extracted from a modified CNN model. In the proposed CBIR framework, features extracted from an improved version of AlexNet are combined with handcraft features. In this modified version of AlexNet, the authors removed the final Fully Connected layer (FC layer) which is of the size 1×4096. In place of these layers, the authors added a new FC layer of dimension 1×64. As a result of this modification, the dimension of the output of the improved AlexNet was reduced to 1×64 from 1×4096. For handcraft texture feature extraction, the LBP feature extraction method is used. To extract shape features HOG is used. Both of these methods are explained in Chapter 2. After this Principal Component Analysis (PCA) is also applied to both LBP and HOG features to reduce the feature vector dimension. Finally, all three features are contaminated to form the final feature vector. Although the authors improved the existing CNN models still the proposed model suffers from the problems given in section 5.1. So more efficient CNN models should be used for fusion-based CBIR methods for extracting deep learning features.

6.2 Methodology

In this chapter, a total of five CBIR methods based on feature fusion of handcrafted and refined CNNs features are proposed. As discussed in chapter 4, both RGB and HSI color models are having their own importance in image creation. Ignoring any of them to extract handcraft features will cause a significant loss of information. So, in this chapter also for all five CBIR methods, handcraft features are extracted from both RGB and HSI color spaces: Modified DDBTC and HOG by considering RGB color space, whereas HSI color space is used to obtain color features by using Interchannel voting among hue, saturation, and intensity components. The same HSI color space was used to obtain the texture features by using DSP followed by GLCM on the intensity component. The detailed discussion about Modified DDBTC is already discussed in Chapter 4 and all the remaining three techniques are discussed in Chapter 2.

To extract the CNN features all the five refined CNN models discussed in chapter-5 are considered. The features extracted from each of these five modified CNN models are concatenated with same handcrafted features used in Chapter 4, which will propose five different feature fusion based State-of-art methods for CBIR. The whole process is shown in the block diagram given in Figure 6.1. The algorithm for this whole process is given in Algorithm 6.1. All the refined CNNs used for feature extraction are discussed briefly in this section.

Residual-GoogleNet: It is an improved version of GoogleNet. As discussed in chapter-4, GoogleNet is performing better than other existing CNN models like: AlexNet, VGG-16 for CBIR. But it suffers from the 'degradation Problem'. In this proposed model a residual connection is established in each Inception layer in GoogleNet to solve this problem.

Cascade-ResNet-50: This proposed model is a refined version of ResNet-50, which is the best performing ResNet version. In the ResNet model the Residual concept is first introduced in CNN structure. The basic structure of ResNet-50 is divided into four different size blocks. In the existing ResNet-50 model residual connection is established inside each block, but there is no residual connection among blocks having different sizes. The proposed model has established a residual connection among the different output size blocks, which results in the cascade structure of the proposed CNN model.

GN-Inception-DarkNet-53: This proposed model is an improved version of DarkNet-53, which is the best performing existing CNN model. In spite of having a better CBIR performance compared to existing CNN models, it uses only a 3×3 size convolution filter for extracting features. In addition to that it uses BN layers after each Convolution which produces more batch error with less batch size. To solve these purposes the proposed model integrates the Inception concept with the basic framework of DarkNet-53. In the proposed GN-Inception-DarkNet-53 structure, each inception layer has three 1×1 kernels, one 3×3 kernel, and one 5×5 kernel. In addition to that, instead of the BN layer used in DarkNet-53, a Group Normalization (GN) layer is added after each convolution layer to make the proposed model's training process independent of batch size. A total of five such inception modules are used in this model.

Xception-DarkNet-53: This is also a modified version of DarkNet-53. In this proposed models, the Xception concept which is an extension of the Inception concept is integrated with the basic structure of DarkNet-53. Like the Inception module, in Xception module also utilized kernels of sizes 1×1, 3×3, and 5×5. But, instead of the conventional 2D convolution used in Darknet-53, the 'Group Convolution' technique is employed in the Xception module for all three kernel sizes. One Xception module produces a total of nine group convolution filters by stacking three times each of the 1×1, 3×3, and 5×5 size Group convolution filters.

Shuffled-Xception-DarkNet-53: This is final refined version of DarkNet-53. In this proposed CNN architecture, the 'Channel Shuffle' operation is integrated with the Xception module. The same Xception module is used in this CNN model as well, but to maintain the information flow between each pair of stacked Group Convolutions of the same size, one Channel Shuffling layer is placed in between them.

Algorithm 6.1: *Algorithm for the Proposed Refined CNN and Handcraft Feature Fusion CBIR.*

---

1. Take a color image as input.
2. Resize this image based on the Refined version of CNN used.
3. Use one of the five refined CNN models to extract high-level features from input images.
4. Extract the handcrafted features from the original color image using Algorithm.1 given in Chapter-4.
5. The final feature vector is obtained by fusion of Features vectors obtained in Step-3 and Step-4.
6. Use similarity measures to compare the features of the query image to all the images in the dataset.
7. Obtain the rank matrix by sorting according to the distance.
8. Apply performance measures on the rank matrix to evaluate the proposed method.

In this algorithm, the resultant size of the resized image in Step-2 depends on the refined version of the CNN model used in Step 3. Table 6.1 shows the required input image size for different CNN models:

Table 6.1 Input Image Size Requirements of Different CNN Models.

CNN Model	Input Size
Residual-GoogleNet	224×224×3
Cascade-ResNet-50	224×224×3
GN-Inception-DarkNet-53	256×256×3
Xception-DarkNet-53	256×256×3
Shuffled-Xception-DarkNet-53	256×256×3

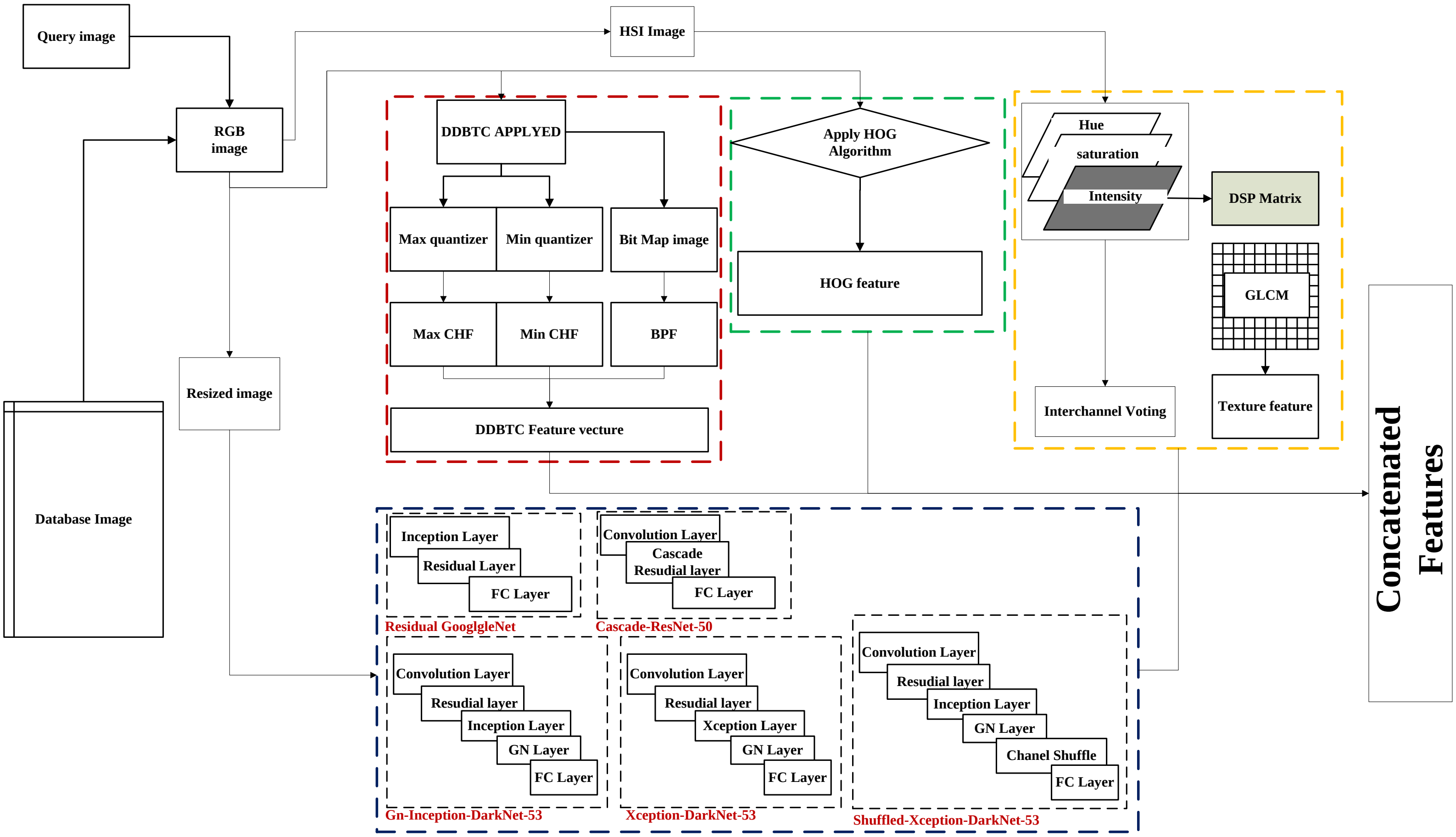


Figure 6.1 Block Diagram of the Proposed Refined CNN and Handcraft Feature Fusion CBIR System.

6.3 Experimental Results and Discussions

In this chapter all the five refined CNN models along with handcrafted features are applied on ten image datasets: Corel-1K, Corel-5K, Corel-10K, VisTex, STex, Color Brodatz, ImageNet-13K, ImageNet-65K, ImageNet-130K, and UKBench. The same five performance parameters used in previous chapters i.e. APR, ARR, F-Measure, ANMRR, and TMRE are used for performance evaluation. All the experimental setup and hyperparameter values for both CNN and handcrafted methods are the same as given in section 4.5.1.

6.3.1 Fusion of Residual-GoogleNet and Handcraft Features

Table 6.2 shows a total of twenty methods along with the proposed method that are used to evaluate the performance of the proposed Residual-GoogleNet. Among them, the first thirteen methods are based on handcraft features. The next five methods are CNN based methods, and the last three methods are feature fusion based CBIR methods.

Because of the space limitation only four out of ten image datasets: Corel-10K, Color Brodatz , ImageNet-130K and UKBench resultant tables are given, but the statistics of all the ten image databases are discussed here in this Chapter. The consolidated results of all the methods for all ten image datasets are given in Chapter-7.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 6.2 are executed on the Corel-1K and Corel-5K datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For Corel-1K, the proposed method is showing a minimum improvement of 2.44%, 2.70%, 2.83%, 4.28%, and 5.27% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 1.44%, 1.20%, 4.17%, 1.63%, and 13.48% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.3. The proposed method is showing a minimum improvement of 3.76%, 4.62%, 7.55%, 6.07%, and 9.23% for the five performance measures respectively.

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 6.2 are executed on the VisTex and STex datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these computed. For VisTex, the proposed method is showing a minimum improvement of 0.09%, 2.69%, 0.64%, 0.51%, and 1.21% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 0.49%, 2.23%, 1.55%, 1.82%, and 0.99% for the five performance measures respectively.

Table 6.2 Different Methods used for Evaluating the Performance of the Proposed Residual-GoogleNet and Handcraft Feature Fusion CBIR System.

Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP [24]
	ColorHist_HSI+ULBP [24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
	Modifed DDBTC (Proposed)
CNN based Features	AlexNet [56]
	VGG16 [57]
	ResNet50 [59]
	GoogleNet [58]
	Residual GoogleNet (Proposed)
Fusion Features	DDBTC+GoogleNet [97]
	Modified AlexNet+HOG+LBP [130]
	Residual GoogleNet+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.4. The proposed method is showing a minimum improvement of 0.28%, 2.65%, 0.84%, 0.96%, and 0.67% for the five performance measures respectively.

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated on ImageNet-13K and ImageNet-65K datasets. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.64%, 1.86%, 2.92%, 3.70% and 15.09% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 1.89%, 1.24%, 2.29%. 6.27%, and 12.71% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.5. The proposed method is showing a minimum improvement of 1.03%,1.27%, 1.99%, 6.48%, and 6.61% for the five performance measures respectively.

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.6. The proposed method is showing a minimum improvement of 0.59%, 0.70%, 2.28%, 1.27%, and 0.06% for the five performance measures respectively

Table 6.3 Performance Measures for Core-10K Dataset using the Proposed Residual-GoogleNet and Handcraft Feature Fusion CBIR System.

		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet50 [59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Residual GoogleNet (Proposed)	**89.93**	**61.17**	**48.834**	**0.2976**	**24.66**
Fusion Features	DDBTC+GoogleNet [97]	88.43	59.82	46.400	0.3035	28.71
	Modified AlexNet+HOG+LBP [130]	88.67	59.00	45.960	0.3288	42.48
	Residual GoogleNet+MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**93.47**	**64.44**	**51.458**	**0.2427**	**19.57**

Table 6.4 Performance Measures for Color Brodatz Dataset using the Proposed Residual-GoogleNet and Handcraft Feature Fusion CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet50 [59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Residual GoogleNet (Proposed)	**99.76**	**96.50**	**70.191**	**0.0234**	**3.74**
Fusion Features	DDBTC+GoogleNet [97]	99.72	96.44	69.716	0.0204	3.44
	Modified AlexNet+HOG+LBP [130]	99.59	92.86	68.373	0.0476	2.76
	Residual GoogleNet+MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**100.00**	**99.09**	**70.309**	**0.0108**	**1.71**

Table 6.5 Performance Measures for ImageNet-130K Dataset using the Proposed Residual-GoogleNet and Handcraft Feature Fusion CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet50 [59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Residual GoogleNet (Proposed)	**84.10**	**56.46**	**43.977**	**0.3228**	**67.56**
Fusion Features	DDBTC+GoogleNet [97]	80.98	55.75	41.399	0.3584	70.36
	Modified AlexNet+HOG+LBP [130]	80.08	55.04	40.874	0.3544	72.89
	Residual GoogleNet+MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**85.01**	**57.36**	**44.965**	**0.2678**	**63.09**

Table 6.6 Performance Measures for UKBench Dataset using the Proposed Residual-GoogleNet and Handcraft Feature Fusion CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet50 [59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Residual GoogleNet (Proposed)	**98.94**	**94.66**	**80.562**	**0.0426**	**2.86**
Fusion Features	DDBTC+GoogleNet [97]	98.46	94.67	80.323	0.0499	3.47
	Modified AlexNet+HOG+LBP [130]	98.59	94.07	79.770	0.0400	3.51
	Residual GoogleNet+MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.48**	**95.37**	**81.988**	**0.0226**	**2.15**

6.3.2 Fusion of Cascade-ResNet-50 and Handcraft Features

Table 6.7 shows different 21 methods that are used to evaluate the performance of the proposed Cascade-ResNet-50.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 6.7 are executed on the Corel-1K and Corel-5K dataset. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluated. For Corel-1K, the proposed method is showing a minimum improvement of 2.83%, 4.57%, 5.70%, 4.89%, and1.51% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 3.06%, 6.03%, 5.87%, 8.52%, and 11.10% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.8. The proposed method is showing a minimum improvement of 6.43%, 8.94%, 11.01%, 15.80%, and 17.10% for the five performance measures respectively.

Table 6.7 Different Methods used for Evaluating the Performance of the Proposed Cascade-ResNet-50 and Handcraft Feature Fusion CBIR System.

Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP [24]
	ColorHist_HSI+ULBP [24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
	Modifed DDBTC (Proposed)
CNN based Features	AlexNet [56]
	VGG16 [57]
	ResNet50 [59]
	GoogleNet [58]
	Casecade Resnet-50 (Proposed)
Fusion Features	DDBTC+GoogleNet [97]
	Modified AlexNet+HOG+LBP [130]
	Casecade Resnet-50+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 6.7 are executed on the VisTex and STex datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For VisTex, the proposed method is showing a minimum improvement of 0.27%, 2.92%, 1.27%, 1.20%, and 1.19% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 0.65%, 8.84%, 3.65%, 5.46%, and 1.62% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.9. The proposed method is showing a minimum improvement of0.27%, 1.22%, 0.41%, 0.69%, 0.01% for the five performance measures respectively.

Table 6.8 Performance Measures for Core-10K Dataset using the Proposed Cascade-ResNet-50 and Handcraft Feature Fusion CBIR System.

		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet50 [59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Casecade Resnet-50 (Proposed)	**93.04**	**65.69**	**50.330**	**0.2440**	**18.97**
Fusion Features	DDBTC+GoogleNet [97]	88.43	59.82	46.400	0.3035	28.71
	Modified AlexNet+HOG+LBP [130]	88.67	59.00	45.960	0.3288	42.48
	Casecade Resnet-50+ MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**96.14**	**68.76**	**53.750**	**0.1455**	**11.78**

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods for ImageNet-13K and ImageNet-65K are computed. For ImageNet-13K, the proposed method is showing a minimum improvement of 2.42%, 3.39%, 4.73%, 5.79%, and 27.04% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 2.56%, 3.77%, 7.20%, 10.61%, and 20.66% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.10. The proposed method is showing a minimum improvement of 2.48%,3.34%, 5.60%, 9.69%, and 12.25% for the five performance measures respectively.

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.11. The proposed method is showing a minimum improvement of 0.77%, 1.90%, 3.17%, 1.96%, and 0.07% for the five performance measures respectively

Table 6.9 Performance Measures for Color Brodatz Dataset using the Proposed Cascade-ResNet-50 and Handcraft Feature Fusion CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet50 [59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Casecade Resnet-50 (Proposed)	**99.83**	**95.88**	**69.847**	**0.0288**	**2.14**
Fusion Features	DDBTC+GoogleNet [97]	99.72	96.44	69.716	0.0204	3.44
	Modified AlexNet+HOG+LBP [130]	99.59	92.86	68.373	0.0476	2.76
	Casecade Resnet-50+ MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**99.99**	**97.65**	**70.007**	**0.0135**	**1.42**

Table 6.10 Performance Measures for ImageNet-130K Dataset using the Proposed Cascade-ResNet-50 and Handcraft Feature Fusion CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet50 [59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Casecade Resnet-50 (Proposed)	**84.95**	**57.98**	**45.103**	**0.2898**	**62.88**
Fusion Features	DDBTC+GoogleNet [97]	80.98	55.75	41.399	0.3584	70.36
	Modified AlexNet+HOG+LBP [130]	80.08	55.04	40.874	0.3544	72.89
	Casecade Resnet-50+ MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**86.47**	**59.43**	**47.355**	**0.2357**	**57.50**

Table 6.11 Performance Measures for UKBench Dataset using the Proposed Cascade-ResNet-50 and Handcraft Feature Fusion CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet50 [59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Casecade Resnet-50 (Proposed)	**99.34**	**95.76**	**81.226**	**0.0269**	**2.15**
Fusion Features	DDBTC+GoogleNet [97]	98.46	94.67	80.323	0.0499	3.47
	Modified AlexNet+HOG+LBP [130]	98.59	94.07	79.770	0.0400	3.51
	Casecade Resnet-50+ MDDBTC+ IC_HSI_GLCM+HOG (Proposed)	**99.65**	**96.58**	**82.640**	**0.0157**	**1.9637**

6.3.3 Fusion of GN-Inception-DarkNet-53 and Handcraft Features

Table 6.12 shows the different 23 methods that are used to evaluate the performance of the proposed GN-Inception-DarkNet-53.

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

Initially, all the methods given in Table 6.12 are executed on the Corel-1K and Corel-5K image datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For Corel-1K, the proposed method is showing a minimum improvement of 0.65%, 6.22%, 2.65%, 2.83%, and 6.22% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 2.99%, 21.08%, 13.65%, 15.15%, and 6.85% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.13. The proposed method is showing a minimum improvement of 4.82%, 21.70%, 15.67%, 18.04%, and 24.52% for the five performance measures respectively.

Table 6.12 Different Methods used for Evaluating the Performance of the Proposed GN-Inception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP [24]
	ColorHist_HSI+ULBP [24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
	Modifed DDBTC (Proposed)
CNN based Features	AlexNet [56]
	VGG16 [57]
	ResNet50 [59]
	GoogleNet [58]
	Darknet53 [64]
	GN-Inception-Darknet-53 (Proposed)
Fusion Features	DDBTC+GoogleNet [97]
	Modified AlexNet+HOG+LBP [130]
	Salient-Key+Opponent color [91]
	GN-Inception-Darknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)

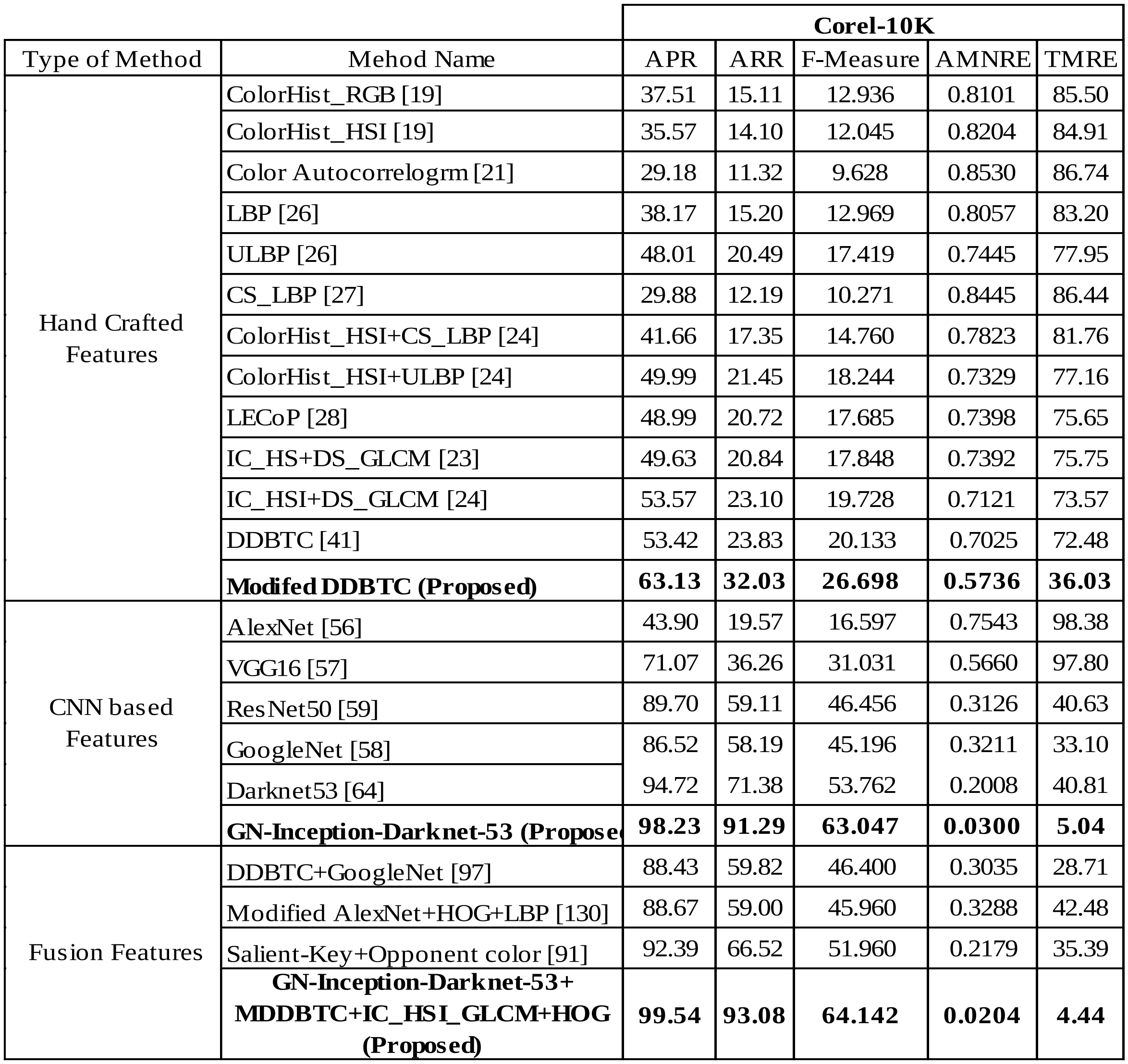
Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 6.12 are executed on the VisTex and STex image datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are evaluate. For VisTex, the proposed method is showing a minimum improvement of 0.38%, 3.15%, 1.38%, 1.40%,and 1.34% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 1.05%, 8.34%, 4.92%, 5.46%, and 2.11% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.14. The proposed method is showing a minimum improvement of 0.19%, 2.78%, 1.40%, 1.62%, and 0.37% for the five performance measures respectively.

Table 6.13 Performance Measures for Core-10K Dataset using the Proposed GN-Inception-DarkNet-53 and Handcraft Feature Fusion CBIR System.



		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet50 [59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Darknet53 [64]	94.72	71.38	53.762	0.2008	40.81
	GN-Inception-Darknet-53 (Propose	**98.23**	**91.29**	**63.047**	**0.0300**	**5.04**
Fusion Features	DDBTC+GoogleNet [97]	88.43	59.82	46.400	0.3035	28.71
	Modified AlexNet+HOG+LBP [130]	88.67	59.00	45.960	0.3288	42.48
	Salient-Key+Opponent color [91]	92.39	66.52	51.960	0.2179	35.39
	GN-Inception-Darknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.54**	**93.08**	**64.142**	**0.0204**	**4.44**

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods on ImageNet-13K and ImageNet-65K are computed. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.04%, 19.84%, 9.92%, 12.89%, and 38.46% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 3.98%, 16.42%, 12.47%, 17.80%, and 24.76% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.15. The proposed method is showing a minimum improvement of 3.88%,17.10%, 12.35%,15.29%, and 30.98% for the five performance measures respectively.

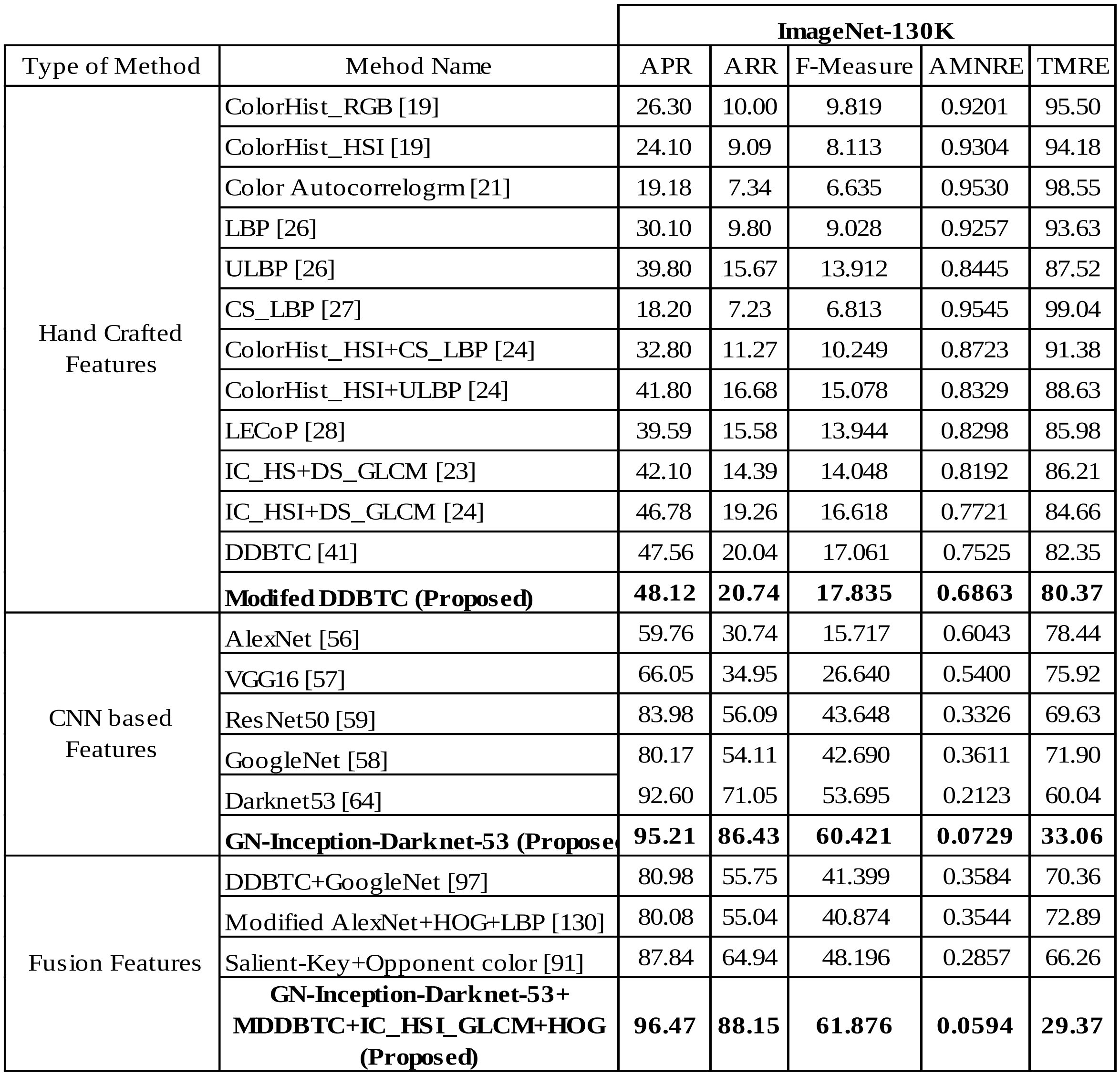
Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.16. The proposed method is showing a minimum improvement of 0.42%,2.79%, 2.07%, 0.79%, and 0.02% for the five performance measures respectively.

Table 6.14 Performance Measures for Color Brodatz Dataset using the Proposed GN-Inception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet50 [59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Darknet53 [64]	99.81	96.87	69.772	0.0172	2.86
	GN-Inception-Darknet-53 (Propose	**99.92**	**98.85**	**70.412**	**0.0063**	**1.08**
Fusion Features	DDBTC+GoogleNet [97]	99.72	96.44	69.716	0.0204	3.44
	Modified AlexNet+HOG+LBP [130]	99.59	92.86	68.373	0.0476	2.76
	Salient-Key+Opponent color [91]	99.74	95.71	69.152	0.0233	2.39
	GN-Inception-Darknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**100.00**	**99.65**	**70.765**	**0.0010**	**1.02**

Table 6.15 Performance Measures for ImageNet-130K Dataset using the Proposed GN-Inception-DarkNet-53 and Handcraft Feature Fusion CBIR System.



		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet50 [59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Darknet53 [64]	92.60	71.05	53.695	0.2123	60.04
	GN-Inception-Darknet-53 (Propose	**95.21**	**86.43**	**60.421**	**0.0729**	**33.06**
Fusion Features	DDBTC+GoogleNet [97]	80.98	55.75	41.399	0.3584	70.36
	Modified AlexNet+HOG+LBP [130]	80.08	55.04	40.874	0.3544	72.89
	Salient-Key+Opponent color [91]	87.84	64.94	48.196	0.2857	66.26
	GN-Inception-Darknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**96.47**	**88.15**	**61.876**	**0.0594**	**29.37**

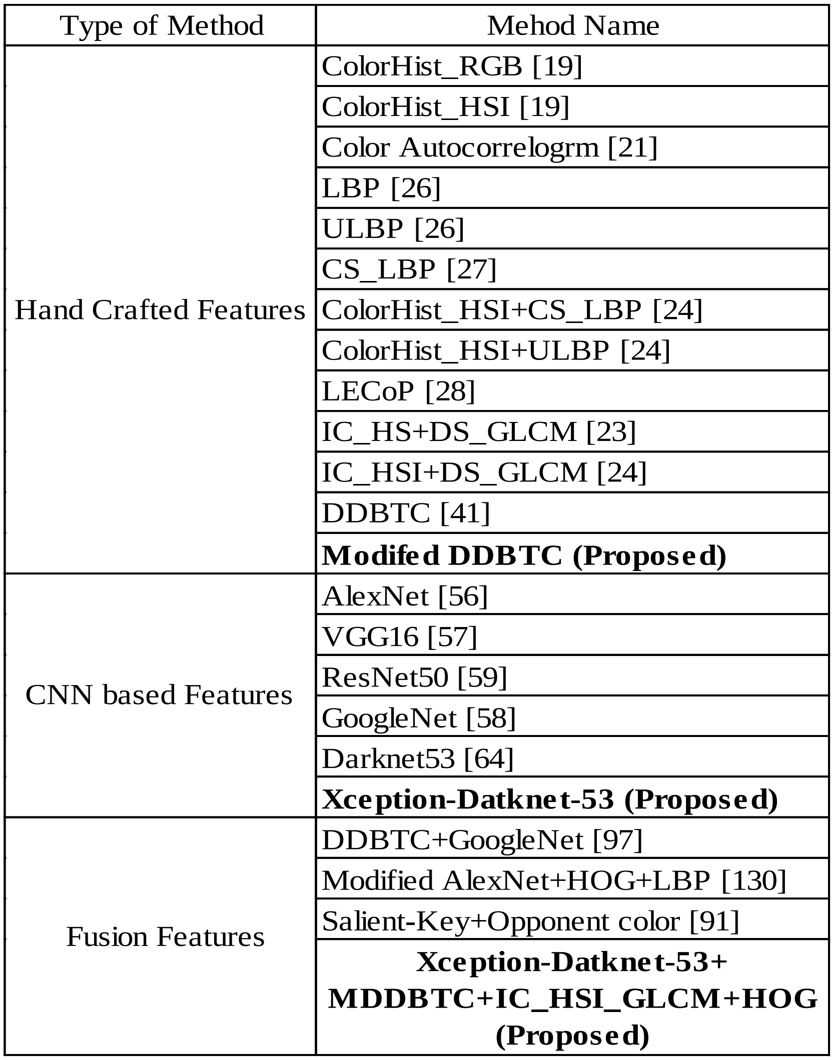
Table 6.16 Performance Measures for UKBench Dataset using the Proposed GN-Inception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet50 [59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Darknet53 [64]	99.54	97.06	81.710	0.0156	1.60
	GN-Inception-Darknet-53 (Propose	**99.94**	**99.78**	**83.203**	**0.0111**	**1.35**
Fusion Features	DDBTC+GoogleNet [97]	98.46	94.67	80.323	0.0499	3.47
	Modified AlexNet+HOG+LBP [130]	98.59	94.07	79.770	0.0400	3.51
	Salient-Key+Opponent color [91]	99.02	95.93	80.860	0.0300	2.58
	GN-Inception-Darknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.96**	**99.85**	**83.225**	**0.0077**	**1.16**

6.3.4 Fusion of Xception-DarkNet-53 and Handcraft Features

Table 6.17 shows the different 23 methods that are used to evaluate the performance of the proposed Xception-DarkNet-53.

Table 6.17 Different Methods used for Evaluating the Performance of the Proposed Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.



Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP [24]
	ColorHist_HSI+ULBP [24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
	Modifed DDBTC (Proposed)
CNN based Features	AlexNet [56]
	VGG16 [57]
	ResNet50 [59]
	GoogleNet [58]
	Darknet53 [64]
	Xception-Datknet-53 (Proposed)
Fusion Features	DDBTC+GoogleNet [97]
	Modified AlexNet+HOG+LBP [130]
	Salient-Key+Opponent color [91]
	Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)

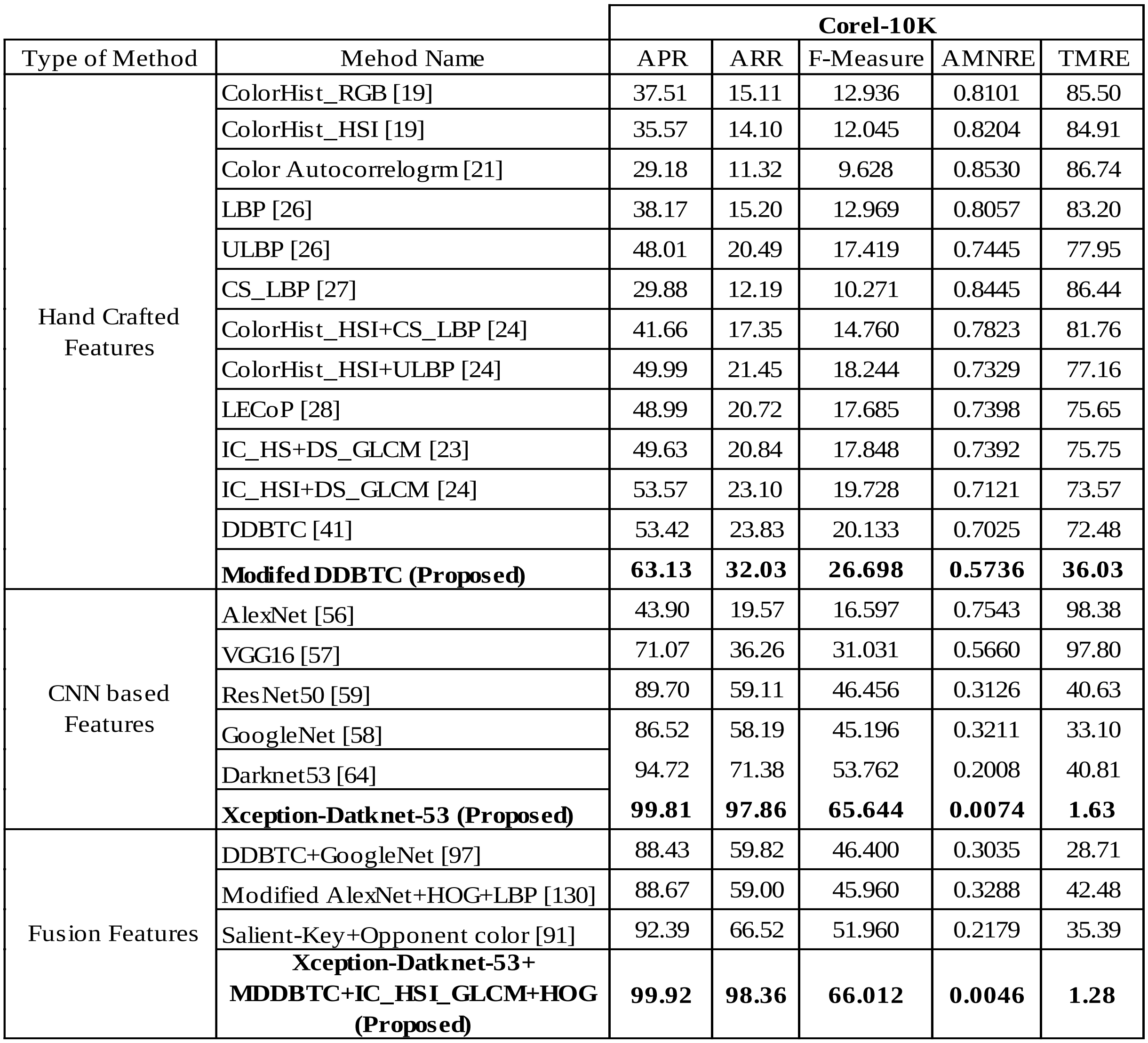
Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 6.17 are executed on the Corel-1K and Corel-5K image datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For Corel-5K, the proposed method is showing a minimum improvement of 0.94%, 9.78%, 4.06%, 4.23%, and 19.40% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 3.11%, 24.81%, 14.99%, 16.28%, and 9.76% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.18. The proposed method is showing a minimum improvement of 5.19%, 26.98%, 18.49%, 19.62%, and 27.71% for the five performance measures respectively.

Table 6.18 Performance Measures for Core-10K Dataset using the Proposed Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.



		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet50 [59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Darknet53 [64]	94.72	71.38	53.762	0.2008	40.81
	Xception-Datknet-53 (Proposed)	**99.81**	**97.86**	**65.644**	**0.0074**	**1.63**
Fusion Features	DDBTC+GoogleNet [97]	88.43	59.82	46.400	0.3035	28.71
	Modified AlexNet+HOG+LBP [130]	88.67	59.00	45.960	0.3288	42.48
	Salient-Key+Opponent color [91]	92.39	66.52	51.960	0.2179	35.39
	Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.92**	**98.36**	**66.012**	**0.0046**	**1.28**

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 6.17 are executed on the VisTex and STex datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For VisTex, the proposed method is showing a minimum improvement of 0.45%, 3.37%, 1.40%, 1.41%, and 1.30% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 1.14%, 8.93%, 5.71%, 5.68%, and 2.20% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.19. The proposed method is showing a minimum improvement of 0.19%, 2.90%, 1.44%, 1.66%, and 0.38% for the five performance measures respectively.

Table 6.19 Performance Measures for Color Brodatz Dataset using the Proposed Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet50 [59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Darknet53 [64]	99.81	96.87	69.772	0.0172	2.86
	Xception-Datknet-53 (Proposed)	**99.97**	**99.49**	**70.635**	**0.0030**	**1.05**
Fusion Features	DDBTC+GoogleNet [97]	99.72	96.44	69.716	0.0204	3.44
	Modified AlexNet+HOG+LBP [130]	99.59	92.86	68.373	0.0476	2.76
	Salient-Key+Opponent color [91]	99.74	95.71	69.152	0.0233	2.39
	Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**100.00**	**99.77**	**70.793**	**0.0006**	**1.02**

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are calculated on ImageNet-13K and ImageNet-65K. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.18%, 20.73%, 10.61%, 13.42%, and 46.36% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 5.28%, 17.84%, 14.11%, 18.91%, and 34.58% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.20. The proposed method is showing a minimum improvement of 4.46%, 18.19%, 13.80%, 17.35%, and 35.96% for the five performance measures respectively.

Table 6.20 Performance Measures for ImageNet-130K Dataset using the Proposed Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet50 [59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Darknet53 [64]	92.60	71.05	53.695	0.2123	60.04
	Xception-Datknet-53 (Proposed)	**95.76**	**86.87**	**60.839**	**0.0664**	**22.69**
Fusion Features	DDBTC+GoogleNet [97]	80.98	55.75	41.399	0.3584	70.36
	Modified AlexNet+HOG+LBP [130]	80.08	55.04	40.874	0.3544	72.89
	Salient-Key+Opponent color [91]	87.84	64.94	48.196	0.2857	66.26
	Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**97.06**	**89.24**	**62.836**	**0.0388**	**24.44**

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.21. The proposed method is showing a minimum improvement of 0.44%, 2.86%, 2.14%, 1.53%, and 0.03% for the five performance measures respectively.

Table 6.21 Performance Measures for UKBench Dataset using the Proposed Xception-DarkNet-53 and Handcraft Featur Fusion CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet50 [59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Darknet53 [64]	99.54	97.06	81.710	0.0156	1.60
	Xception-Datknet-53 (Proposed)	**99.98**	**99.88**	**83.263**	**0.0005**	**1.00**
Fusion Features	DDBTC+GoogleNet [97]	98.46	94.67	80.323	0.0499	3.47
	Modified AlexNet+HOG+LBP [130]	98.59	94.07	79.770	0.0400	3.51
	Salient-Key+Opponent color [91]	99.02	95.93	80.860	0.0300	2.58
	Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.99**	**99.92**	**83.274**	**0.0004**	**1.00**

6.3.5 Fusion of Shaffled-Xception-DarkNet-53 and Handcraft Features

Table 6.22 shows the different twenty three methods that are used to evaluate the performance of the proposed Shaffled-Xception-DarkNet-53.

Table 6.22 Different Methods used for Evaluating the Performance of the Proposed Shuffled-Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

Type of Method	Mehod Name
Hand Crafted Features	ColorHist_RGB [19]
	ColorHist_HSI [19]
	Color Autocorrelogrm [21]
	LBP [26]
	ULBP [26]
	CS_LBP [27]
	ColorHist_HSI+CS_LBP [24]
	ColorHist_HSI+ULBP [24]
	LECoP [28]
	IC_HS+DS_GLCM [23]
	IC_HSI+DS_GLCM [24]
	DDBTC [41]
	Modifed DDBTC (Proposed)
CNN based Features	AlexNet [ch6_10/56]
	VGG16 [ch6_11/57]
	ResNet50 [ch6_12/59]
	GoogleNet [ch6_13/58]
	Darknet53 [ch6_15/64]
	Shuffled Xception-Datknet-53 (Proposed)
Fusion Features	DDBTC+GoogleNet [ch6_14/97]
	Modified AlexNet+HOG+LBP [ch6_1/130]
	Salient-Key+Opponent color [ch_16/91]
	Shuffled Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)

Image Dataset-1 (Corel-1K) and Image Dataset-2 (Corel-5K)

All the methods given in Table 6.22 are executed on the Corel-1K and Corel-5K image datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For Corel-5K, the proposed method is showing a minimum improvement of 0.94%, 9.90%, 4.09%, 4.25%, and 19.54% for the five performance measures respectively. For Corel-5K, the proposed method is showing a minimum improvement of 3.12%, 24.90%, 15.04%, 16.33%, and 9.90% for the five performance measures respectively.

Image Dataset-3 (Corel-10K)

Results of the five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.23. The proposed method is showing a minimum improvement of 5.19%, 27.40%, 18.36%, 19.69%, and 27.82% for the five performance measures respectively.

Table 6.23 Performance Measures for Core-10K Dataset using the Proposed Shuffled-Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [ch6_10/56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [ch6_11/57]	71.07	36.26	31.031	0.5660	97.80
	ResNet50 [ch6_12/59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [ch6_13/58]	86.52	58.19	45.196	0.3211	33.10
	Darknet53 [ch6_15/64]	94.72	71.38	53.762	0.2008	40.81
	Shuffled Xception-Datknet-53 (Proposed)	**99.84**	**98.04**	**65.729**	**0.0060**	**1.53**
Fusion Features	DDBTC+GoogleNet [ch6_14/97]	88.43	59.82	46.400	0.3035	28.71
	Modified AlexNet+HOG+LBP [ch6_1/130]	88.67	59.00	45.960	0.3288	42.48
	Salient-Key+Opponent color [ch_16/91]	92.39	66.52	51.960	0.2179	35.39
	Shuffled Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.91**	**98.78**	**65.921**	**0.0038**	**1.16**

Image Dataset-4 (VisTex) and Image Dataset-5 (STex)

All the methods given in Table 6.22 are executed on the VisTex and STex datasets. The five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are computed. For VisTex, the proposed method is showing a minimum improvement of 0.52%, 3.48%, 1.51%, 1.42%, and 1.35% for the five performance measures respectively. For STex, the proposed method is showing a minimum improvement of 1.20%, 9.35%, 5.84%, 6.07%, and 2.22% for the five performance measures respectively.

Image Dataset-6 (Color Brodatz)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.25. The proposed method is showing a minimum improvement of 0.19%, 2.99%, 1.51%, 1.69%, and 0.38% for the five performance measures respectively.

Table 6.24 Performance Measures for Color Brodatz Dataset using the Proposed Shuffled-Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [ch6_10/56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [ch6_11/57]	96.51	76.19	61.121	0.1785	5.75
	ResNet50 [ch6_12/59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [ch6_13/58]	99.64	94.21	68.812	0.0304	4.40
	Darknet53 [ch6_15/64]	99.81	96.87	69.772	0.0172	2.86
	Shuffled Xception-Datknet-53 (Proposed)	**99.98**	**99.57**	**70.709**	**0.0016**	**1.04**
Fusion Features	DDBTC+GoogleNet [ch6_14/97]	99.72	96.44	69.716	0.0204	3.44
	Modified AlexNet+HOG+LBP [ch6_1/130]	99.59	92.86	68.373	0.0476	2.76
	Salient-Key+Opponent color [ch_16/91]	99.74	95.71	69.152	0.0233	2.39
	Shuffled Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**100.00**	**99.86**	**70.842**	**0.0003**	**1.02**

Image Dataset-7 (ImageNet-13K) and Image Dataset-8 (ImageNet-65K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods on ImageNet-13K and ImageNet-65K are computed. For ImageNet-13K, the proposed method is showing a minimum improvement of 1.27%, 20.94%, 10.70%, 13.63%, and 49.79% for the five performance measures respectively. For ImageNet-65K, the proposed method is showing a minimum improvement of 5.80%, 19.73%, 15.49%, 19.20%, and 38.80% for the five performance measures respectively.

Image Dataset-9 (ImageNet-130K)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.26. The proposed method is showing a minimum improvement of 5.37%, 19.10%, 14.62%, 18.57%, and 39.98% for the five performance measures respectively.

Table 6.25 Performance Measures for ImageNet-130K Dataset using the Proposed Shuffled-Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [ch6_10/56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [ch6_11/57]	66.05	34.95	26.640	0.5400	75.92
	ResNet50 [ch6_12/59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [ch6_13/58]	80.17	54.11	42.690	0.3611	71.90
	Darknet53 [ch6_15/64]	92.60	71.05	53.695	0.2123	60.04
	Shuffled Xception-Datknet-53 (Proposed)	**97.88**	**89.23**	**63.346**	**0.0399**	**20.77**
Fusion Features	DDBTC+GoogleNet [ch6_14/97]	80.98	55.75	41.399	0.3584	70.36
	Modified AlexNet+HOG+LBP [ch6_1/130]	80.08	55.04	40.874	0.3544	72.89
	Salient-Key+Opponent color [ch_16/91]	87.84	64.94	48.196	0.2857	66.26
	Shuffled Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**97.97**	**90.15**	**63.377**	**0.0265**	**20.46**

Image Dataset-10 (UKBench)

Results of five parameters APR, ARR, F-Measure, ANMRR, and TMRE for all these methods are given in Table 6.27. The proposed method is showing a minimum improvement of 0.44%, 2.90%, 2.18%, 1.54%, and 0.03% for the five performance measures respectively.

Table 6.26 Performance Measures for UKBench Dataset using the Proposed Shuffled-Xception-DarkNet-53 and Handcraft Feature Fusion CBIR System.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [ch6_10/56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [ch6_11/57]	97.06	86.49	75.195	0.0698	8.65
	ResNet50 [ch6_12/59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [ch6_13/58]	98.08	93.92	79.430	0.0579	3.83
	Darknet53 [ch6_15/64]	99.54	97.06	81.710	0.0156	1.60
	Shuffled Xception-Datknet-53 (Proposed)	**99.99**	**99.88**	**83.270**	**0.0004**	**1.00**
Fusion Features	DDBTC+GoogleNet [ch6_14/97]	98.46	94.67	80.323	0.0499	3.47
	Modified AlexNet+HOG+LBP [ch6_1/130]	98.59	94.07	79.770	0.0400	3.51
	Salient-Key+Opponent color [ch_16/91]	99.02	95.93	80.860	0.0300	2.58
	Shuffled Xception-Datknet-53+ MDDBTC+IC_HSI_GLCM+HOG (Proposed)	**99.99**	**99.96**	**83.301**	**0.0002**	**1.00**

5.4 Observations

In this chapter a total of five feature fusion based CBIR method are proposed. To extract CNN based features, all five modified CNN models proposed in chapter-5: Residual GoogleNet, Cascade-ResNet-50, GN-Iception-DarkNet-53, Xception-DarkNet-53, and Shuffled-Xception-

DarkNet-53 are used. Each of these CNN features are concatenated with the same handcrafted features used in Chapter 4 which contains both RGB and HSI features. All feature fusion methods have given better performance than corresponding CNN models alone. As ‘shuffled-Xception-DarkNet-53 is best performing CNN model shown in Chapter 5, the respective feature fusion method is showing the best performance among all existing and proposed methods.

Chapter 7

Consolidated Results, Conclusions and Future Scope

7.1 Consolidated Results

To visualize the performance of all the proposed methods, the consolidated results for all the ten image datasets are shown in the Tables 7.1 to Table 7.10.

Table 7.1 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on Corel-1K Image Dataset.

		Corel-1K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	64.28	37.71	29.288	0.5334	8.07
	ColorHist_HSI [19]	63.21	36.56	28.829	0.5335	8.73
	Color Autocorrelogrm [21]	52.54	28.32	22.429	0.6278	9.45
	LBP [26]	69.18	38.69	31.236	0.5208	8.03
	ULBP [26]	69.26	44.42	33.939	0.4574	7.49
	CS_LBP [27]	51.69	33.44	25.046	0.5746	7.82
	ColorHist_HSI+CS_LBP [24]	69.47	42.09	32.468	0.4817	7.63
	ColorHist_HSI+ULBP [24]	77.73	49.11	37.945	0.4077	7.23
	LECoP [28]	76.14	46.82	36.692	0.4301	7.59
	IC_HS+DS_GLCM [23]	78.20	49.92	38.637	0.4020	7.29
	IC_HSI+DS_GLCM [24]	80.15	50.94	39.503	0.3883	7.35
	DDBTC [41]	85.03	57.01	39.726	0.3356	6.85
	Modifed DDBTC (Proposed)	**85.61**	**57.34**	**44.025**	**0.3115**	**6.63**
CNN based Features	AlexNet [56]	76.22	41.59	34.554	0.4912	9.79
	VGG16 [57]	89.08	58.70	47.576	0.3274	5.64
	ResNet50 [59]	94.27	75.99	53.307	0.1989	4.81
	GoogleNet [58]	95.61	79.43	57.629	0.1201	3.50
	Residual GoogleNet (Proposed)	**96.53**	**80.77**	**58.875**	**0.1187**	**3.04**
	Casecade Resnet-50 (Proposed)	**97.76**	**81.10**	**59.117**	**0.1021**	**4.01**
	Darknet53[64]	99.06	89.97	63.521	0.0427	2.77
	GN-Inception-Darknet-53 (Proposed)	**99.33**	**95.05**	**64.133**	**0.0179**	**2.67**
	Xception-Datknet-53 (Proposed)	**100.00**	**99.66**	**66.181**	**0.0005**	**1.03**
	Shuffled Xception-Datknet-53 (Proposed)	**100.00**	**99.70**	**66.204**	**0.0004**	**1.02**
Fusion Features	DDBTC+GoogleNet [97]	96.43	80.79	59.112	0.1074	4.71
	Modified AlexNet+HOG+LBP [130]	93.77	73.92	53.286	0.2178	6.29
	Salient-Key+Opponent color [91]	97.07	84.34	61.088	0.0923	4.08
	GoogleNet+HandCraft (Proposed)	**98.50**	**82.78**	**60.151**	**0.0911**	**3.22**
	Residual GoogleNet+HandCraft (Proposed)	**98.87**	**83.50**	**60.988**	**0.0646**	**3.03**
	Casecade Resnet-50+HandCraft (Proposed)	**99.26**	**85.36**	**62.888**	**0.0585**	**3.36**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.71**	**96.19**	**65.278**	**0.0144**	**2.21**
	Xception-Datknet-53+HandCraft (Proposed)	**100.00**	**99.75**	**66.214**	**0.0004**	**1.02**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**100.00**	**99.87**	**66.232**	**0.0002**	**1.01**

Table 7.2 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on Corel-5K Image Dataset.

		Corel-5K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	46.81	20.49	17.356	0.7414	42.22
	ColorHist_HSI [19]	43.26	19.00	15.996	0.7585	42.91
	Color Autocorrelogrm [21]	35.33	14.91	12.509	0.8065	44.67
	LBP [26]	46.19	19.85	16.667	0.7446	41.62
	ULBP [26]	46.94	21.10	17.584	0.7276	38.87
	CS_LBP [27]	32.50	13.40	11.178	0.8241	42.57
	ColorHist_HSI+CS_LBP [24]	50.97	23.07	19.411	0.7089	40.51
	ColorHist_HSI+ULBP [24]	60.43	28.76	23.989	0.6410	37.66
	LECoP [28]	58.34	27.18	22.782	0.6603	37.58
	IC_HS+DS_GLCM [23]	61.14	29.67	24.628	0.6296	36.19
	IC_HSI+DS_GLCM [24]	62.60	30.23	25.250	0.6269	36.55
	DDBTC [41]	62.50	31.13	25.748	0.6136	36.77
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	52.37	24.99	20.946	0.6852	49.21
	VGG16 [57]	80.10	44.21	37.321	0.4740	48.93
	ResNet50 [59]	93.92	64.38	50.216	0.2518	21.15
	GoogleNet [58]	91.49	61.30	46.179	0.3284	24.52
	Residual GoogleNet (Proposed)	**93.54**	**64.15**	**48.862**	**0.2997**	**18.68**
	Casecade Resnet-50 (Proposed)	**95.55**	**72.04**	**54.229**	**0.1797**	**10.17**
	Darknet53[64]	96.85	74.73	56.202	0.1645	5.94
	GN-Inception-Darknet-53 (Proposed)	**99.36**	**93.15**	**63.110**	**0.0151**	**2.97**
	Xception-Datknet-53 (Proposed)	**99.96**	**99.12**	**66.035**	**0.0020**	**1.23**
	Shuffled Xception-Datknet-53 (Proposed)	**99.96**	**99.16**	**66.056**	**0.0018**	**1.11**
Fusion Features	DDBTC+GoogleNet [97]	93.71	68.21	52.010	0.2109	23.37
	Modified AlexNet+HOG+LBP [130]	92.85	64.87	50.070	0.2835	22.35
	Salient-Key+Opponent color [91]	94.93	71.07	54.840	0.1979	13.39
	GoogleNet+HandCraft (Proposed)	**94.26**	**64.71**	**50.371**	**0.2039**	**20.40**
	Residual GoogleNet+HandCraft (Proposed)	**95.36**	**66.07**	**52.977**	**0.1946**	**14.55**
	Casecade Resnet-50+HandCraft (Proposed)	**96.99**	**74.25**	**55.898**	**0.1257**	**7.93**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.84**	**95.81**	**65.242**	**0.0130**	**2.58**
	Xception-Datknet-53+HandCraft (Proposed)	**99.96**	**99.55**	**66.132**	**0.0018**	**1.15**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**99.97**	**99.63**	**66.168**	**0.0012**	**1.09**

Table 7.3 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on Corel-10K Image Dataset.

		Corel-10K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	37.51	15.11	12.936	0.8101	85.50
	ColorHist_HSI [19]	35.57	14.10	12.045	0.8204	84.91
	Color Autocorrelogrm [21]	29.18	11.32	9.628	0.8530	86.74
	LBP [26]	38.17	15.20	12.969	0.8057	83.20
	ULBP [26]	48.01	20.49	17.419	0.7445	77.95
	CS_LBP [27]	29.88	12.19	10.271	0.8445	86.44
	ColorHist_HSI+CS_LBP [24]	41.66	17.35	14.760	0.7823	81.76
	ColorHist_HSI+ULBP [24]	49.99	21.45	18.244	0.7329	77.16
	LECoP [28]	48.99	20.72	17.685	0.7398	75.65
	IC_HS+DS_GLCM [23]	49.63	20.84	17.848	0.7392	75.75
	IC_HSI+DS_GLCM [24]	53.57	23.10	19.728	0.7121	73.57
	DDBTC [41]	53.42	23.83	20.133	0.7025	72.48
	Modifed DDBTC (Proposed)	**63.13**	**32.03**	**26.698**	**0.5736**	**36.03**
CNN based Features	AlexNet [56]	43.90	19.57	16.597	0.7543	98.38
	VGG16 [57]	71.07	36.26	31.031	0.5660	97.80
	ResNet50 [59]	89.70	59.11	46.456	0.3126	40.63
	GoogleNet [58]	86.52	58.19	45.196	0.3211	33.10
	Residual GoogleNet (Proposed)	**89.93**	**61.17**	**48.834**	**0.2976**	**24.66**
	Casecade Resnet-50 (Proposed)	**93.04**	**65.69**	**50.330**	**0.2440**	**18.97**
	Darknet53[64]	94.72	71.38	53.762	0.2008	40.81
	GN-Inception-Darknet-53 (Proposed)	**98.23**	**91.29**	**63.047**	**0.0300**	**5.04**
	Xception-Datknet-53 (Proposed)	**99.81**	**97.86**	**65.644**	**0.0074**	**1.63**
	Shuffled Xception-Datknet-53 (Proposed)	**99.84**	**98.04**	**65.729**	**0.0060**	**1.53**
Fusion Features	DDBTC+GoogleNet [97]	88.43	59.82	46.400	0.3035	28.71
	Modified AlexNet+HOG+LBP [130]	88.67	59.00	45.960	0.3288	42.48
	Salient-Key+Opponent color [91]	92.39	66.52	51.960	0.2179	35.39
	GoogleNet+HandCraft (Proposed)	**90.48**	**60.96**	**48.120**	**0.3041**	**27.11**
	Residual GoogleNet+HandCraft (Proposed)	**93.47**	**64.44**	**51.458**	**0.2427**	**19.57**
	Casecade Resnet-50+HandCraft (Proposed)	**96.14**	**68.76**	**53.750**	**0.1455**	**11.78**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.54**	**93.08**	**64.142**	**0.0204**	**4.44**
	Xception-Datknet-53+HandCraft (Proposed)	**99.92**	**98.36**	**66.012**	**0.0046**	**1.28**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**99.91**	**98.78**	**65.921**	**0.0038**	**1.16**

Table 7.4 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on VisTex Image Dataset.

		VisTex				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	88.91	61.96	52.658	0.2866	13.38
	ColorHist_HSI [19]	90.70	61.58	53.085	0.2995	11.95
	Color Autocorrelogrm [21]	86.64	57.28	49.576	0.3462	15.51
	LBP [26]	97.07	81.00	64.357	0.1216	4.33
	ULBP [26]	98.13	82.75	65.522	0.1081	3.96
	CS_LBP [27]	88.67	66.92	54.663	0.2448	5.74
	ColorHist_HSI+CS_LBP [24]	96.21	74.08	60.938	0.1785	8.58
	ColorHist_HSI+ULBP [24]	98.91	84.55	66.679	0.0951	4.13
	LECoP [28]	98.28	81.63	65.402	0.1254	5.02
	IC_HS+DS_GLCM [23]	98.45	80.21	65.339	0.0544	2.62
	IC_HSI+DS_GLCM [24]	98.67	81.56	65.564	0.1280	5.21
	DDBTC [41]	98.35	88.69	67.987	0.0646	2.68
	Modifed DDBTC (Proposed)	**98.95**	**89.09**	**68.437**	**0.0646**	**2.91**
CNN based Features	AlexNet [56]	86.64	57.20	50.234	0.3478	26.20
	VGG16 [57]	89.77	60.67	53.045	0.3118	4.47
	ResNet50 [59]	98.74	89.08	68.848	0.0684	2.80
	GoogleNet [58]	98.44	88.14	67.993	0.0700	2.61
	Residual GoogleNet (Proposed)	**99.04**	**93.45**	**69.270**	**0.0530**	**1.93**
	Casecade Resnet-50 (Proposed)	**99.65**	**96.34**	**71.657**	**0.0209**	**1.71**
	Darknet53[64]	99.29	95.72	71.493	0.0228	1.77
	GN-Inception-Darknet-53 (Proposed)	**99.80**	**98.47**	**72.423**	**0.0078**	**1.13**
	Xception-Datknet-53 (Proposed)	**99.88**	**98.68**	**72.689**	**0.0063**	**1.08**
	Shuffled Xception-Datknet-53 (Proposed)	**99.96**	**99.03**	**72.988**	**0.0053**	**1.04**
Fusion Features	DDBTC+GoogleNet [97]	99.46	96.43	71.858	0.0155	1.54
	Modified AlexNet+HOG+LBP [130]	97.96	87.92	67.048	0.0784	2.93
	Salient-Key+Opponent color [91]	99.09	93.72	69.929	0.0376	2.05
	GoogleNet+HandCraft (Proposed)	**100.00**	**98.97**	**72.138**	**0.0139**	**1.12**
	Residual GoogleNet+HandCraft (Proposed)	**99.55**	**99.11**	**72.327**	**0.0104**	**1.07**
	Casecade Resnet-50+HandCraft (Proposed)	**99.73**	**99.35**	**72.788**	**0.0035**	**1.07**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.84**	**99.58**	**72.868**	**0.0015**	**1.02**
	Xception-Datknet-53+HandCraft (Proposed)	**99.91**	**99.80**	**72.880**	**0.0013**	**1.03**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**99.98**	**99.90**	**72.962**	**0.0012**	**1.01**

Table 7.5 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on STex Image Dataset.

		Stex				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	59.96	31.62	29.480	0.6218	185.87
	ColorHist_HSI [19]	66.66	36.67	33.639	0.5606	115.11
	Color Autocorrelogrm [21]	68.41	41.44	36.642	0.5020	104.89
	LBP [26]	76.92	47.22	41.849	0.4526	104.11
	ULBP [26]	82.54	52.39	45.915	0.4027	86.54
	CS_LBP [27]	62.08	33.65	31.033	0.5987	139.08
	ColorHist_HSI+CS_LBP [24]	74.08	43.51	39.201	0.4928	131.00
	ColorHist_HSI+ULBP [24]	87.11	59.08	50.683	0.3333	73.53
	LECoP [28]	91.67	65.80	55.397	0.2657	44.40
	IC_HS+DS_GLCM [23]	93.51	69.23	57.621	0.2326	38.88
	IC_HSI+DS_GLCM [24]	93.44	68.53	57.245	0.2422	42.43
	DDBTC [41]	94.21	69.94	58.520	0.2411	36.57
	Modifed DDBTC (Proposed)	**95.05**	**70.68**	**59.150**	**0.2011**	**35.92**
CNN based Features	AlexNet [56]	78.10	45.56	41.781	0.4830	39.25
	VGG16 [57]	90.37	60.26	52.874	0.3325	27.75
	ResNet50 [59]	98.58	87.14	67.654	0.0801	14.04
	GoogleNet [58]	96.36	84.00	63.614	0.1381	23.76
	Residual GoogleNet (Proposed)	**98.72**	**87.52**	**67.908**	**0.0787**	**13.76**
	Casecade Resnet-50 (Proposed)	**98.92**	**91.89**	**68.943**	**0.0615**	**10.32**
	Darknet53[64]	98.73	89.17	68.522	0.0659	12.04
	GN-Inception-Darknet-53 (Proposed)	**98.98**	**93.01**	**70.049**	**0.0402**	**5.19**
	Xception-Datknet-53 (Proposed)	**99.48**	**96.88**	**72.077**	**0.0223**	**2.99**
	Shuffled Xception-Datknet-53 (Proposed)	**99.82**	**97.88**	**72.346**	**0.0135**	**2.12**
Fusion Features	DDBTC+GoogleNet [97]	97.75	84.04	65.897	0.1055	14.69
	Modified AlexNet+HOG+LBP [130]	95.76	82.85	62.840	0.1807	25.33
	Salient-Key+Opponent color [91]	97.73	88.06	67.953	0.0732	13.72
	GoogleNet+HandCraft (Proposed)	**98.67**	**86.95**	**67.485**	**0.0834**	**10.90**
	Residual GoogleNet+HandCraft (Proposed)	**99.07**	**89.37**	**68.789**	**0.0619**	**9.33**
	Casecade Resnet-50+HandCraft (Proposed)	**99.23**	**95.98**	**70.323**	**0.0254**	**6.35**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.78**	**97.51**	**72.117**	**0.0113**	**2.02**
	Xception-Datknet-53+HandCraft (Proposed)	**99.87**	**98.09**	**72.696**	**0.0091**	**1.61**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**99.92**	**98.51**	**72.788**	**0.0052**	**1.48**

Table 7.6 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on Color Brodatz Image Dataset.

		Color Brodatz				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	80.24	47.10	41.164	0.4493	35.72
	ColorHist_HSI [19]	97.28	77.25	61.027	0.1709	18.96
	Color Autocorrelogrm [21]	94.47	68.87	55.862	0.2658	34.18
	LBP [26]	89.29	70.22	54.997	0.2247	13.92
	ULBP [26]	91.97	74.39	57.630	0.1940	12.36
	CS_LBP [27]	81.48	56.36	46.057	0.3515	19.94
	ColorHist_HSI+CS_LBP [24]	87.94	62.09	50.771	0.2983	24.14
	ColorHist_HSI+ULBP [24]	94.28	77.07	59.472	0.1710	13.30
	LECoP [28]	98.98	87.51	65.949	0.0837	8.04
	IC_HS+DS_GLCM [23]	98.91	87.34	65.818	0.0792	6.35
	IC_HSI+DS_GLCM [24]	99.64	90.41	67.304	0.0648	6.28
	DDBTC [41]	99.65	93.01	68.090	0.0300	5.41
	Modifed DDBTC (Proposed)	**99.66**	**93.48**	**68.502**	**0.0327**	**4.81**
CNN based Features	AlexNet [56]	89.47	61.25	51.820	0.3212	10.88
	VGG16 [57]	96.51	76.19	61.121	0.1785	5.75
	ResNet50 [59]	99.69	93.86	68.724	0.0378	2.46
	GoogleNet [58]	99.64	94.21	68.812	0.0304	4.40
	Residual GoogleNet (Proposed)	**99.76**	**96.50**	**70.191**	**0.0234**	**3.74**
	Casecade Resnet-50 (Proposed)	**99.83**	**95.88**	**69.847**	**0.0288**	**2.14**
	Darknet53[64]	99.81	96.87	69.772	0.0172	2.86
	GN-Inception-Darknet-53 (Proposed)	**99.92**	**98.85**	**70.412**	**0.0063**	**1.08**
	Xception-Datknet-53 (Proposed)	**99.97**	**99.49**	**70.635**	**0.0030**	**1.05**
	Shuffled Xception-Datknet-53 (Proposed)	**99.98**	**99.57**	**70.709**	**0.0016**	**1.04**
Fusion Features	DDBTC+GoogleNet [97]	99.72	96.44	69.716	0.0204	3.44
	Modified AlexNet+HOG+LBP [130]	99.59	92.86	68.373	0.0476	2.76
	Salient-Key+Opponent color [91]	99.74	95.71	69.152	0.0233	2.39
	GoogleNet+HandCraft (Proposed)	**100.00**	**98.97**	**69.814**	**0.0139**	**1.12**
	Residual GoogleNet+HandCraft (Proposed)	**100.00**	**99.09**	**70.309**	**0.0108**	**1.71**
	Casecade Resnet-50+HandCraft (Proposed)	**99.99**	**97.65**	**70.007**	**0.0135**	**1.42**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**100.00**	**99.65**	**70.765**	**0.0010**	**1.02**
	Xception-Datknet-53+HandCraft (Proposed)	**100.00**	**99.77**	**70.793**	**0.0006**	**1.02**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**100.00**	**99.86**	**70.842**	**0.0003**	**1.02**

Table 7.7 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on ImageNet-13K Image Dataset.

		ImageNet-13K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	47.64	21.86	18.356	0.6414	9.02
	ColorHist_HSI [19]	44.60	21.73	17.926	0.5505	9.13
	Color Autocorrelogrm [21]	36.76	15.98	13.651	0.7065	9.65
	LBP [26]	48.79	21.05	17.967	0.6945	9.03
	ULBP [26]	48.94	22.34	18.348	0.6276	9.06
	CS_LBP [27]	33.50	15.05	12.978	0.7541	9.01
	ColorHist_HSI+CS_LBP [24]	52.87	25.77	21.231	0.6089	8.97
	ColorHist_HSI+ULBP [24]	61.73	29.88	25.399	0.5410	9.18
	LECoP [28]	55.34	28.57	24.078	0.5860	8.76
	IC_HS+DS_GLCM [23]	63.14	31.10	26.096	0.5296	8.99
	IC_HSI+DS_GLCM [24]	65.60	33.96	28.250	0.4927	8.72
	DDBTC [41]	68.50	35.93	31.935	0.4136	8.97
	Modifed DDBTC (Proposed)	**69.05**	**40.65**	**32.619**	**0.3823**	**8.28**
CNN based Features	AlexNet [56]	76.37	39.60	35.795	0.3452	7.85
	VGG16 [57]	83.70	47.82	40.321	0.3040	7.79
	ResNet50 [59]	95.92	68.98	54.216	0.1918	7.14
	GoogleNet [58]	94.87	69.03	54.179	0.2084	6.95
	Residual GoogleNet (Proposed)	**96.28**	**71.02**	**56.388**	**0.1746**	**5.39**
	Casecade Resnet-50 (Proposed)	**97.50**	**72.93**	**57.977**	**0.1435**	**4.79**
	Darknet53[64]	98.61	78.03	59.072	0.1399	6.16
	GN-Inception-Darknet-53 (Proposed)	**99.53**	**96.03**	**65.193**	**0.0137**	**3.66**
	Xception-Datknet-53 (Proposed)	**99.75**	**97.91**	**65.729**	**0.0109**	**2.93**
	Shuffled Xception-Datknet-53 (Proposed)	**99.84**	**98.45**	**65.936**	**0.0103**	**2.52**
Fusion Features	DDBTC+GoogleNet [97]	95.79	70.49	55.746	0.1836	6.47
	Modified AlexNet+HOG+LBP [130]	93.86	67.06	53.970	0.2100	7.33
	Salient-Key+Opponent color [91]	96.05	72.41	55.020	0.1800	6.59
	GoogleNet+HandCraft (Proposed)	**96.98**	**71.88**	**56.543**	**0.1588**	**5.35**
	Residual GoogleNet+HandCraft (Proposed)	**97.57**	**72.35**	**57.679**	**0.1466**	**5.11**
	Casecade Resnet-50+HandCraft (Proposed)	**98.35**	**73.88**	**58.877**	**0.1257**	**4.03**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.65**	**97.88**	**65.646**	**0.0110**	**2.70**
	Xception-Datknet-53+HandCraft (Proposed)	**99.79**	**98.77**	**66.099**	**0.0057**	**1.99**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**99.88**	**98.98**	**66.159**	**0.0036**	**1.68**

Table 7.8 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on ImageNet-65K Image Dataset.

		ImageNet-65K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	27.41	11.90	10.628	0.9001	48.10
	ColorHist_HSI [19]	25.57	10.76	9.511	0.9104	48.75
	Color Autocorrelogrm [21]	20.18	8.92	7.766	0.9430	49.67
	LBP [26]	31.17	11.82	10.928	0.9057	47.86
	ULBP [26]	41.01	17.90	14.929	0.8345	45.15
	CS_LBP [27]	19.88	9.19	8.113	0.9245	48.08
	ColorHist_HSI+CS_LBP [24]	34.66	13.05	12.019	0.8523	46.05
	ColorHist_HSI+ULBP [24]	42.78	18.18	16.378	0.8129	42.69
	LECoP [28]	41.85	17.68	15.544	0.8098	45.63
	IC_HS+DS_GLCM [23]	43.06	16.04	16.048	0.7992	44.53
	IC_HSI+DS_GLCM [24]	48.87	20.89	17.786	0.7521	45.11
	DDBTC [41]	49.15	21.84	18.261	0.7425	43.31
	Modifed DDBTC (Proposed)	**49.67**	**22.46**	**18.788**	**0.7325**	**43.99**
CNN based Features	AlexNet [56]	60.67	31.86	16.597	0.5943	39.69
	VGG16 [57]	69.08	36.65	29.116	0.5200	38.93
	ResNet50 [59]	85.98	57.11	45.556	0.3126	36.25
	GoogleNet [58]	82.98	55.19	44.597	0.3311	35.59
	Residual GoogleNet (Proposed)	**84.08**	**56.97**	**45.153**	**0.2766**	**33.59**
	Casecade Resnet-50 (Proposed)	**87.97**	**59.70**	**49.654**	**0.2346**	**31.57**
	Darknet53[64]	93.18	73.23	54.387	0.2026	30.89
	GN-Inception-Darknet-53 (Proposed)	**96.08**	**88.44**	**61.853**	**0.0593**	**20.72**
	Xception-Datknet-53 (Proposed)	**97.39**	**90.77**	**62.935**	**0.0299**	**16.88**
	Shuffled Xception-Datknet-53 (Proposed)	**98.08**	**91.88**	**63.877**	**0.0235**	**14.91**
Fusion Features	DDBTC+GoogleNet [97]	83.74	56.12	45.465	0.3188	34.88
	Modified AlexNet+HOG+LBP [130]	84.98	56.74	45.095	0.3223	37.85
	Salient-Key+Opponent color [91]	88.97	66.54	49.693	0.2635	34.78
	GoogleNet+HandCraft (Proposed)	**86.75**	**57.54**	**46.854**	**0.2870**	**30.55**
	Residual GoogleNet+HandCraft (Proposed)	**87.88**	**58.35**	**47.077**	**0.2499**	**28.65**
	Casecade Resnet-50+HandCraft (Proposed)	**88.54**	**60.88**	**50.323**	**0.2065**	**24.75**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**97.16**	**89.65**	**62.647**	**0.0246**	**18.76**
	Xception-Datknet-53+HandCraft (Proposed)	**98.46**	**91.08**	**63.733**	**0.0135**	**13.95**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**98.98**	**92.97**	**64.646**	**0.0105**	**11.88**

Table 7.9 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on ImageNet-130K Image Dataset.

		ImageNet-130K				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	26.30	10.00	9.819	0.9201	95.50
	ColorHist_HSI [19]	24.10	9.09	8.113	0.9304	94.18
	Color Autocorrelogrm [21]	19.18	7.34	6.635	0.9530	98.55
	LBP [26]	30.10	9.80	9.028	0.9257	93.63
	ULBP [26]	39.80	15.67	13.912	0.8445	87.52
	CS_LBP [27]	18.20	7.23	6.813	0.9545	99.04
	ColorHist_HSI+CS_LBP [24]	32.80	11.27	10.249	0.8723	91.38
	ColorHist_HSI+ULBP [24]	41.80	16.68	15.078	0.8329	88.63
	LECoP [28]	39.59	15.58	13.944	0.8298	85.98
	IC_HS+DS_GLCM [23]	42.10	14.39	14.048	0.8192	86.21
	IC_HSI+DS_GLCM [24]	46.78	19.26	16.618	0.7721	84.66
	DDBTC [41]	47.56	20.04	17.061	0.7525	82.35
	Modifed DDBTC (Proposed)	**48.12**	**20.74**	**17.835**	**0.6863**	**80.37**
CNN based Features	AlexNet [56]	59.76	30.74	15.717	0.6043	78.44
	VGG16 [57]	66.05	34.95	26.640	0.5400	75.92
	ResNet50 [59]	83.98	56.09	43.648	0.3326	69.63
	GoogleNet [58]	80.17	54.11	42.690	0.3611	71.90
	Residual GoogleNet (Proposed)	**84.10**	**56.46**	**43.977**	**0.3228**	**67.56**
	Casecade Resnet-50 (Proposed)	**84.95**	**57.98**	**45.103**	**0.2898**	**62.88**
	Darknet53[64]	92.60	71.05	53.695	0.2123	60.04
	GN-Inception-Darknet-53 (Proposed)	**95.21**	**86.43**	**60.421**	**0.0729**	**33.06**
	Xception-Datknet-53 (Proposed)	**95.76**	**86.87**	**60.839**	**0.0664**	**22.69**
	Shuffled Xception-Datknet-53 (Proposed)	**97.88**	**89.23**	**63.346**	**0.0399**	**20.77**
Fusion Features	DDBTC+GoogleNet [97]	80.98	55.75	41.399	0.3584	70.36
	Modified AlexNet+HOG+LBP [130]	80.08	55.04	40.874	0.3544	72.89
	Salient-Key+Opponent color [91]	87.84	64.94	48.196	0.2857	66.26
	GoogleNet+HandCraft (Proposed)	**82.08**	**56.65**	**42.988**	**0.3067**	**66.77**
	Residual GoogleNet+HandCraft (Proposed)	**85.01**	**57.36**	**44.965**	**0.2678**	**63.09**
	Casecade Resnet-50+HandCraft (Proposed)	**86.47**	**59.43**	**47.355**	**0.2357**	**57.50**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**96.47**	**88.15**	**61.876**	**0.0594**	**29.37**
	Xception-Datknet-53+HandCraft (Proposed)	**97.06**	**89.24**	**62.836**	**0.0388**	**24.44**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**97.97**	**90.15**	**63.377**	**0.0265**	**20.46**

Table 7.10 Performance Measure for all the Proposed Methods Compared with Different Existing Methods on UKBench Image Dataset.

		UKBench				
Type of Method	Mehod Name	APR	ARR	F-Measure	AMNRE	TMRE
Hand Crafted Features	ColorHist_RGB [19]	83.22	63.76	58.980	0.3200	54.62
	ColorHist_HSI [19]	85.70	64.76	60.944	0.2810	43.47
	Color Autocorrelogrm [21]	58.25	33.06	35.870	0.6100	160.29
	LBP [26]	65.38	40.72	41.780	0.5800	105.49
	ULBP [26]	67.99	42.19	43.757	0.5082	97.02
	CS_LBP [27]	59.41	34.26	36.931	0.5937	145.69
	ColorHist_HSI+CS_LBP [24]	68.22	43.78	44.930	0.4400	110.37
	ColorHist_HSI+ULBP [24]	70.07	44.77	45.741	0.4792	96.06
	LECoP [28]	89.14	70.10	64.763	0.2338	28.36
	IC_HS+DS_GLCM [23]	89.46	70.99	65.313	0.2261	23.00
	IC_HSI+DS_GLCM [24]	93.42	79.13	70.707	0.1553	17.17
	DDBTC [41]	94.76	81.09	71.870	0.1445	16.05
	Modifed DDBTC (Proposed)	**95.36**	**81.85**	**72.363**	**0.1377**	**14.33**
CNN based Features	AlexNet [56]	96.12	85.07	74.340	0.0900	11.85
	VGG16 [57]	97.06	86.49	75.195	0.0698	8.65
	ResNet50 [59]	98.89	94.13	80.030	0.0353	3.27
	GoogleNet [58]	98.08	93.92	79.430	0.0579	3.83
	Residual GoogleNet (Proposed)	**98.94**	**94.66**	**80.562**	**0.0426**	**2.86**
	Casecade Resnet-50 (Proposed)	**99.34**	**95.76**	**81.226**	**0.0269**	**2.15**
	Darknet53[64]	99.54	97.06	81.710	0.0156	1.60
	GN-Inception-Darknet-53 (Proposed)	**99.94**	**99.78**	**83.203**	**0.0111**	**1.35**
	Xception-Datknet-53 (Proposed)	**99.98**	**99.88**	**83.263**	**0.0005**	**1.00**
	Shuffled Xception-Datknet-53 (Proposed)	**99.99**	**99.88**	**83.270**	**0.0004**	**1.00**
Fusion Features	DDBTC+GoogleNet [97]	98.46	94.67	80.323	0.0499	3.47
	Modified AlexNet+HOG+LBP [130]	98.59	94.07	79.770	0.0400	3.51
	Salient-Key+Opponent color [91]	99.02	95.93	80.860	0.0300	2.58
	GoogleNet+HandCraft (Proposed)	**99.02**	**95.12**	**81.166**	**0.0398**	**2.94**
	Residual GoogleNet+HandCraft (Proposed)	**99.48**	**95.37**	**81.988**	**0.0226**	**2.15**
	Casecade Resnet-50+HandCraft (Proposed)	**99.65**	**96.58**	**82.640**	**0.0157**	**1.96**
	GN-Inception-Darknet-53+HandCraft (Proposed)	**99.96**	**99.85**	**83.225**	**0.0077**	**1.16**
	Xception-Datknet-53+HandCraft (Proposed)	**99.99**	**99.92**	**83.274**	**0.0004**	**1.00**
	Shuffled Xception-Datknet-53+HandCraft (Proposed)	**99.99**	**99.96**	**83.301**	**0.0002**	**1.00**

7.2 Conclusions and Future Scope

The conclusions of the dissertation and future directions are outlined in this section.

Conclusions:

The content based image retrieval problem is addressed in this work by focusing on the feature extraction part in the process. To extract a detailed feature representation, four methods are proposed.

Resolution Independent CBIR method is proposed using Handcrafted features. As a decomposition method, Haar wavelet is used. The level of the Haar wavelet is determined based on the size of the input image. Interchannel voting among Hue, Saturation, and Intensity components is used as a color feature. DSP followed by GLCM is used as Texture Features. The proposed method is applied to Six benchmark datasets and their multiresolution version as well. Best performance is achieved in both Single and Multi Resolution image Datasets.

A Feature-Fusion based CBIR framework is proposed in which CNN model GoogleNet is used. As Handcrafted features, both RGB (Modified DDBCT and HOG) and HSI color (InterChannel Voting among H, S, and I and DSP) models are used. The proposed method is evaluated on ten datasets (Corel-1K, Corel-5K, Corel-10K, VisTex, Stex, Color Brodatz, ImageNet-13K, ImageNet-65K, ImageNet-130K , and UKBench) and has shown substantial improvement.

A modified version of GoogleNet is proposed: Residual GoogleNet. A skip connection is established in each Inception layer in GoogleNet. A modified version of Resnet-50 named Cascade-ResNet-50 is proposed. The proposed model has established a residual connection among the different output size blocks, which results in the cascade structure of the proposed CNN model. GN-Inception-DarkNet-53: A refined version of DarkNet-53 is proposed where to extract more detailed features than DarkNet-53 of input image inception module is incorporated with the basic structure of DarkNet53. After each convolution layer Group Normalization (GN) layer, instead of the BN layer used in DarkNet-53. Another refinement is: Xception-DarkNet-53. Xception concept is incorporated with the basic structure of DarkNet53 which is an extension of the Inception concept. Instead of a traditional 2D convolution operation, ‘Grouped Convolution’ is performed on all the filters. The last one is Shuffled- Xception-DarkNet-53. In this model the Xception concept along with ‘Channel Shuffle’ is used.

All the proposed refined CNN features are fused with Handcrafted features from both RGB and HSI color space and performance is measured. All feature fusion methods have given better performance than corresponding CNN models.

To summarize the work, the Content based image retrieval problem is tackled with incremental improvement starting from Handcrafted features to deep learning CNN features. In this work the following are proposed and evaluated to achieve the given objectives.

- One new Handcrafted feature for CBIR which is effective for both single and multiresolution datasets.
- One Feature Fusion Method which uses Handcrafted and GoogleNet features.
- Modified CNN Models: Residual GoogleNet, Cascade ResNet-50, GN-Inception-DarkNet-53, Xception-DarkNet-53, and Shuffled-Xception-DarkNet-53 are propose and used for CBIR.
- Five efficient CBIR methods which fuses the modified CNN features and Handcrafted features.

Future Scope:

- The Content based Image Retrieval is limited to images in this work. It will be extended to videos so that it can be utilized in real world applications.
- Region Of Interest (ROI) concept can be incorporated In the field of CBIR.
- As the retrieval method involves computational overhead, it will be shifted to a distributed environment to speed up the process.
- On the other hand, lightweight retrieval methods can be proposed, to enable real time application without the requirement of a GPU.

Publications

1. **Debanjan Pathak** and U.S.N.Raju, “Content-based image retrieval using Group Normalized-Inception-DarkNet-53”, International Journal of Multimedia Information Retrieval, Vol. 10, No.5,Springer, https://doi.org/10.1007/s13735-021-00215-4 (SCI) Journal: Accepted-July, 2021, online published).
2. **Debanjan Pathak** and U.S.N. Raju, “Content-based image retrieval using feature-fusion of GroupNormalized-Inception-DarkNet-53 features and Handcrafted features”, Optik, Vol. 246, Elsevier, https://doi.org/10.1016/j.ijleo.2021.167754, (SCI Journal: Accepted-July, 2021, online published).
3. **Debanjan Pathak** and U.S.N.Raju, “Content-based image retrieval for super-resolutioned images using feature fusion: Deep learning and Handcrafted”, Concurrency and Computation: Practice and Experience, John Wiley & Sons https://doi.org/10.1002/cpe.6851 (SCI Journal: Accepted-December, 2021, online published).
4. **Debanjan Pathak** and U.S.N.Raju, “Resolution Independent Image Retrieval using Global and Local Features”, International Journal of Pattern Recognition and Artificial Intelligence. (SCI Journal: Under Review-September, 2021).
5. **Debanjan Pathak** and U.S.N.Raju, “Xception-DarkNet-53: An efficient Deep Learning model for Content-Based Image Retrieval” Journal of Visual Communication and Image Representation, (SCI Journal: With Editor-May, 2022).
6. **Debanjan Pathak** and U.S.N.Raju, “Shuffled-Xception-DarkNet-53: A Content-Based Image Retrieval Model Based on Deep Learning Algorithm”, Computers and Electrical Engineering,(SCI Journal: Under Review-May, 2022).

References

1. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D. and Steele, D., 1995. Query by image and video content: The QBIC system. computer, 28(9), pp.23-32.
2. Wang, S., 2001. A robust CBIR approach using local color histograms.
3. Carson, C., Belongie, S., Greenspan, H. and Malik, J., 2002. Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Transactions on pattern analysis and machine intelligence, 24(8), pp.1026-1038.
4. Anguera, X., Xu, J. and Oliver, N., 2008, October. Multimodal photo annotation and retrieval on a mobile phone. In Proceedings of the 1st ACM international conference on Multimedia information retrieval (pp. 188-194).
5. Meena, S.M. and Shetty, S.S., 2013, August. Trace transform based identifier for speech based image retrieval on mobile phones. In 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 967-972). IEEE.
6. Choi, Y. and Rasmussen, E.M., 2002. Users' relevance criteria in image retrieval in American history. Information processing & management, 38(5), pp.695-726.
7. Markkula, M., Tico, M., Sepponen, B., Nirkkonen, K. and Sormunen, E., 2001. A test collection for the evaluation of content-based image retrieval algorithms—a user and task-based approach. Information Retrieval, 4(3), pp.275-293.
8. Goodrum, A. and Spink, A., 2001. Image searching on the Excite Web search engine. Information Processing & Management, 37(2), pp.295-311.
9. Tsai, C.F., 2012. Bag-of-words representation in image annotation: A review. International Scholarly Research Notices, 2012.
10. Zhang, D., Islam, M.M. and Lu, G., 2012. A review on automatic image annotation techniques. Pattern Recognition, 45(1), pp.346-362.
11. Corel1k image dataset, http://wang.ist.psu.edu/docs/related/
12. Corel10k image dataset, Guang-Hai Liu, http://www.ci.gxnu.edu.cn/cbir/Dataset.aspx
13. VisTex dataset, Alex Sandy Pentland, http://vismod.media.mit.edu/pub/VisTex/
14. Salzburg Texture Image Database, http://www.wavelab.at/sources/STex/
15. Multiband Texture dataset http://multibandtexture.recherche.usherbrooke.ca/

16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), pp.211-252.

18. Yue, J., Li, Z., Liu, L. and Fu, Z., 2011. Content-based image retrieval using color and texture fused features. Mathematical and Computer Modelling, 54(3-4), pp.1121-1127.

19. Singha, M. and Hemachandran, K., 2012. Content based image retrieval using color and texture. Signal & Image Processing, 3(1), p.39.

20. Huang, J., Kumar, S.R. and Mitra, M., 1997, November. Combining supervised learning with color correlograms for content-based image retrieval. In Proceedings of the fifth ACM international conference on Multimedia (pp. 325-334).

21. Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J. and Zabih, R., 1997, June. Image indexing using color correlograms. In Proceedings of IEEE computer society conference on Computer Vision and Pattern Recognition (pp. 762-768). IEEE.

22. Chun, Y.D., Kim, N.C. and Jang, I.H., 2008. Content-based image retrieval using multiresolution color and texture features. IEEE Transactions on multimedia, 10(6), pp.1073-1084.

23. Bhunia, A.K., Bhattacharyya, A., Banerjee, P., Roy, P.P. and Murala, S., 2020. A novel feature descriptor for image retrieval by combining modified color histogram and diagonally symmetric co-occurrence texture pattern. Pattern Analysis and Applications, 23(2), pp.703-723.

24. Kanaparthi, S.K., Raju, U.S.N., Shanmukhi, P., Aneesha, G.K. and Rahman, M.E.U., 2020. Image retrieval by integrating global correlation of color and intensity histograms with local texture features. Multimedia Tools and Applications, 79(47), pp.34875-34911.

25. Cross, G.R. and Jain, A.K., 1983. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, (1), pp.25-39.

26. Ojala, T., Pietikainen, M. and Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 24(7), pp.971-987.

27. Heikkilä, M., Pietikäinen, M. and Schmid, C., 2006. Description of interest regions with center-symmetric local binary patterns. In Computer vision, graphics and image processing (pp. 58-69). Springer, Berlin, Heidelberg.

28. Verma, M., Raman, B. and Murala, S., 2015. Local extrema co-occurrence pattern for color and texture image retrieval. Neurocomputing, 165, pp.255-269.

29. Zhang, B., Gao, Y., Zhao, S. and Liu, J., 2009. Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. IEEE transactions on image processing, 19(2), pp.533-544.

30. Murala, S., Maheshwari, R.P. and Balasubramanian, R., 2012. Local tetra patterns: a new feature descriptor for content-based image retrieval. IEEE transactions on image processing, 21(5), pp.2874-2886.

31. Raju, U.S.N., Suresh Kumar, K., Haran, P., Boppana, R.S. and Kumar, N., 2020. Content-based image retrieval using local texture features in distributed environment. International Journal of Wavelets, Multiresolution and Information Processing, 18(01), p.1941001.

32. Guo, Z., Zhang, L. and Zhang, D., 2010. A completed modeling of local binary pattern operator for texture classification. IEEE transactions on image processing, 19(6), pp.1657-1663.

33. Dubey, S.R., Singh, S.K. and Singh, R.K., 2016. Multichannel decoded local binary patterns for content-based image retrieval. IEEE transactions on image processing, 25(9), pp.4018-4032

34. Brahnam, S., Jain, L.C., Nanni, L. and Lumini, A. eds., 2014. Local binary patterns: new variants and applications (Vol. 506). Springer Berlin Heidelberg.

35. Linde, Y., Buzo, A. and Gray, R., 1980. An algorithm for vector quantizer design. IEEE Transactions on communications, 28(1), pp.84-95.

36. Sivic, J. and Zisserman, A., 2003, October. Video Google: A text retrieval approach to object matching in videos. In Computer Vision, IEEE International Conference on (Vol. 3, pp. 1470-1470). IEEE Computer Society.

37. Elsayad, I., Martinet, J., Urruty, T. and Djeraba, C., 2010, June. A new spatial weighting scheme for bag-of-visual-words. In 2010 International Workshop on Content Based Multimedia Indexing (CBMI) (pp. 1-6). IEEE.

38. Chen, X., Hu, X. and Shen, X., 2009, April. Spatial weighting for bag-of-visual-words and its application in content-based image retrieval. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 867-874). Springer, Berlin, Heidelberg.

39. Bouachir, W., Kardouchi, M. and Belacel, N., 2009, November. Improving bag of visual words image retrieval: A fuzzy weighting scheme for efficient indexation. In 2009 Fifth International Conference on Signal Image Technology and Internet Based Systems (pp. 215-220). IEEE.

40. Zhu, L., Jin, H., Zheng, R. and Feng, X., 2014. Weighting scheme for image retrieval based on bag-of-visual-words. IET Image Processing, 8(9), pp.509-518.

41. Guo, J.M., Prasetyo, H. and Wang, N.J., 2015. Effective image retrieval system using dot-diffused block truncation coding features. IEEE Transactions on Multimedia, 17(9), pp.1576-1590.

42. Guo, J.M. and Liu, Y.F., 2013. Improved block truncation coding using optimized dot diffusion. IEEE Transactions on Image Processing, 23(3), pp.1269-1275.

43. Zhang, D. and Lu, G., 2005. Study and evaluation of different Fourier methods for image retrieval. Image and vision computing, 23(1), pp.33-49.

44. Hu, R., Barnard, M. and Collomosse, J., 2010, September. Gradient field descriptor for sketch based retrieval and localization. In 2010 IEEE International conference on image processing (pp. 1025-1028). IEEE.

45. Hu, R.X., Jia, W., Ling, H., Zhao, Y. and Gui, J., 2013. Angular pattern and binary angular pattern for shape retrieval. IEEE Transactions on Image Processing, 23(3), pp.1118-1127.

46. Osowski, S., 2002. Fourier and wavelet descriptors for shape recognition using neural networks—a comparative study. Pattern Recognition, 35(9), pp.1949-1957.

47. Mathew, S.P., Balas, V.E. and Zachariah, K.P., 2015. A content-based image retrieval system based on convex hull geometry. Acta Polytechnica Hungarica, 12(1), pp.103-116.

48. Marr, D. and Nishihara, H.K., 1978. Representation and recognition of the spatial organization of three-dimensional shapes. Proceedings of the Royal Society of London. Series B. Biological Sciences, 200(1140), pp.269-294.

49. Brady, M., 1983. Criteria for representations of shape. In Human and machine vision (pp. 39-84). Academic Press.

50. Soffer, A. and Samet, H., 1998. Pictorial query specification for browsing through spatially referenced image databases. Journal of Visual Languages & Computing, 9(6), pp.567-596.

51. Folkers, A. and Samet, H., 2002, August. Content-based image retrieval using Fourier descriptors on a logo database. In 2002 International Conference on Pattern Recognition (Vol. 3, pp. 521-524). IEEE.

52. Dalal, N. and Triggs, B., 2005, June. Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.

53. Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), pp.3212-3232.

54. Oquab, M., Bottou, L., Laptev, I. and Sivic, J., 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 685-694).

55. Oquab, M., Bottou, L., Laptev, I. and Sivic, J., 2014. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1717-1724).

56. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

57. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

58. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

59. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

60. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

61. Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

62. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J. and Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

63. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520).

64. Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

65. Liu, B., Wang, S., ZHAO, J. and LI, M., 2019. Ship tracking and recognition based on Darknet network and YOLOv3 algorithm. Journal of Computer Applications, 39(6), p.1663.

66. Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

67. Ciregan, D., Meier, U. and Schmidhuber, J., 2012, June. Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3642-3649). IEEE.

68. Ioffe, S. and Szegedy, C., 2015, June. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.

69. Alluri, L.A.K.S.H.M.E.E.L.A.V.A.N.Y.A. and Dendukuri, H.E.M.A.L.A.T.H.A., 2020. An efficient system for cbir using deep learning convolutional neural networks. Int J Recent Dev Sci Technol, 4(1), pp.160-167.

70. Tarawneh, A.S., Celik, C., Hassanat, A.B. and Chetverikov, D., 2020. Detailed investigation of deep features with sparse representation and dimensionality reduction in cbir: A comparative study. Intelligent Data Analysis, 24(1), pp.47-68.

71. Sezavar, A., Farsi, H. and Mohamadzadeh, S., 2019. Content-based image retrieval by combining convolutional neural networks and sparse representation. Multimedia Tools and Applications, 78(15), pp.20895-20912.

72. Saritha, R.R., Paul, V. and Kumar, P.G., 2019. Content based image retrieval using deep learning process. Cluster Computing, 22(2), pp.4187-4200.

73. Mustafic, F., Prazina, I. and Ljubovic, V., 2019, October. A new method for improving content-based image retrieval using deep learning. In 2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT) (pp. 1-4). IEEE.

74. Ramanjaneyulu, K., Swamy, K.V. and Rao, C.S., 2018, November. Novel CBIR system using CNN architecture. In 2018 3rd International conference on inventive computation technologies (ICICT) (pp. 379-383). IEEE.

75. Messina, N., Amato, G., Carrara, F., Falchi, F. and Gennaro, C., 2020. Learning visual features for relational CBIR. International Journal of Multimedia Information Retrieval, 9(2), pp.113-124.

76. Song, K., Li, F., Long, F., Wang, J. and Ling, Q., 2018. Discriminative deep feature learning for semantic-based image retrieval. IEEE Access, 6, pp.44268-44280.

77. Zheng, L., Yang, Y. and Tian, Q., 2017. SIFT meets CNN: A decade survey of instance retrieval. IEEE transactions on pattern analysis and machine intelligence, 40(5), pp.1224-1244.

78. Swati, Z.N.K., Zhao, Q., Kabir, M., Ali, F., Ali, Z., Ahmed, S. and Lu, J., 2019. Content-based brain tumor retrieval for MR images using transfer learning. IEEE Access, 7, pp.17809-17822.

79. Cai, Y., Li, Y., Qiu, C., Ma, J. and Gao, X., 2019. Medical image retrieval based on convolutional neural network and supervised hashing. IEEE access, 7, pp.51877-51885.

80. Wei, S., Liao, L., Li, J., Zheng, Q., Yang, F. and Zhao, Y., 2019. Saliency inside: learning attentive CNNs for content-based image retrieval. IEEE Transactions on image processing, 28(9), pp.4580-4593.

81. Bhandi, V. and Devi, K.S., 2019, March. Image retrieval by fusion of features from pre-trained deep convolution neural networks. In 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE) (pp. 35-40). IEEE.

82. Özaydın, U., Georgiou, T. and Lew, M., 2019, September. A comparison of cnn and classic features for image retrieval. In 2019 International Conference on Content-Based Multimedia Indexing (CBMI) (pp. 1-4). IEEE.

83. Tzelepi, M. and Tefas, A., 2018. Deep convolutional learning for content based image retrieval. Neurocomputing, 275, pp.2467-2478.

84. Rao, Y., Liu, W., Fan, B., Song, J. and Yang, Y., 2018. A novel relevance feedback method for CBIR. World Wide Web, 21(6), pp.1505-1522.

85. Desai, P., Pujari, J., Sujatha, C., Kamble, A. and Kambli, A., 2021. Hybrid Approach for Content-Based Image Retrieval using VGG16 Layered Architecture and SVM: An Application of Deep Learning. SN Computer Science, 2(3), pp.1-9.

86. Simran, A., Kumar, P.S. and Bachu, S., 2021, March. Content Based Image Retrieval Using Deep Learning Convolutional Neural Network. In IOP Conference Series: Materials Science and Engineering (Vol. 1084, No. 1, p. 012026). IOP Publishing.

87. Dubey, S.R., 2021. A decade survey of content based image retrieval using deep learning. IEEE Transactions on Circuits and Systems for Video Technology, 32(5), pp.2687-2704.

88. Kopparthi, S. and Nynalasetti, K.K.R., 2020. Content based Image Retrieval using Deep Learning Technique with Distance Measures. In Science (pp. 251-261). Technology & Human Values.

89. Jeya Christy, A. and Dhanalakshmi, K., 2022. Content-Based Image Recognition and Tagging by Deep Learning Methods. Wireless Personal Communications, 123(1), pp.813-838.

90. Keisham, N. and Neelima, A., 2022. Efficient content-based image retrieval using deep search and rescue algorithm. Soft Computing, 26(4), pp.1597-1616.

91. Pradhan, J., Pal, A.K., Banka, H. and Dansena, P., 2021. Fusion of region based extracted features for instance-and class-based CBIR applications. Applied Soft Computing, 102, p.107063.
92. Alrahhal, M. and Supreethi, K.P., 2020. Multimedia image retrieval system by combining CNN with handcraft features in three different similarity measures. International Journal of Computer Vision and Image Processing (IJCVIP), 10(1), pp.1-23.
93. Dong, L., Lin, Z., Liang, Y., He, L., Zhang, N., Chen, Q., Cao, X. and Izquierdo, E., 2016. A hierarchical distributed processing framework for big image data. IEEE Transactions on Big Data, 2(4), pp.297-309.
94. Li, J., Yang, B., Yang, W., Sun, C. and Xu, J., 2021. Subspace-based multi-view fusion for instance-level image retrieval. The Visual Computer, 37(3), pp.619-633.
95. Yan, K., Wang, Y., Liang, D., Huang, T. and Tian, Y., 2016, October. Cnn vs. sift for image retrieval: Alternative or complementary?. In Proceedings of the 24th ACM international conference on Multimedia (pp. 407-411).
96. Devulapalli, S. and Krishnan, R., 2021. Remote sensing image retrieval by integrating automated deep feature extraction and handcrafted features using curvelet transform. Journal of Applied Remote Sensing, 15(1), p.016504.
97. Liu, P., Guo, J.M., Wu, C.Y. and Cai, D., 2017. Fusion of deep learning and compressed domain features for content-based image retrieval. IEEE Transactions on Image Processing, 26(12), pp.5706-5717.
98. Shakarami, A., Menhaj, M.B. and Tarrah, H., 2021. Diagnosing COVID-19 disease using an efficient CAD system. Optik, 241, p.167199.
99. Hussain, S., Zia, M.A. and Arshad, W., 2021. Additive deep feature optimization for semantic image retrieval. Expert Systems with Applications, 170, p.114545.
100. Putzu, L., Piras, L. and Giacinto, G., 2020. Convolutional neural networks for relevance feedback in content based image retrieval. Multimedia Tools and Applications, 79(37), pp.26995-27021.
101. Burt, P.J. and Adelson, E.H., 1987. The Laplacian pyramid as a compact image code. In Readings in computer vision (pp. 671-679). Morgan Kaufmann.
102. Daubechies, I., 1992. Ten lectures on wavelets. Society for industrial and applied mathematics.
103. Goswami, J.C. and Chan, A.K., 2011. Fundamentals of wavelets: theory, algorithms, and applications. John Wiley & Sons.
104. Frazier, M.W., 2006. An introduction to wavelets through linear algebra. Springer Science & Business Media.

105. Gonzalez, R.C., Woods, R.E. and Eddins, S.L., 2004. Digital image processing using matlab, Pearson Education India.

106. Soman, K.P., 2010. Insight into wavelets: from theory to practice. PHI Learning Pvt. Ltd..

107. Prasad, L. and Iyengar, S.S., 2020. Wavelet analysis with applications to image processing. CRC press.

108. Soman, K.P., 2010. Insight into wavelets: from theory to practice. PHI Learning Pvt. Ltd..

109. Claypoole, R.L., Baraniuk, R.G. and Nowak, R.D., 1998, May. Adaptive wavelet transforms via lifting. In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181) (Vol. 3, pp. 1513-1516). IEEE.

110. Claypoole, R.L., Baraniuk, R.G. and Nowak, R.D., 1998, October. Lifting construction of non-linear wavelet transforms. In Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis (Cat. No. 98TH8380) (pp. 49-52). IEEE.

111. Claypoole,R. L., Baraniuk, R. G. and Nowak, R. D., 1999.Adaptive wavelet transforms via lifting. Technical Report 9304, Department of Electrical and Computer Engineering, Rice University, Houston, Texas.

112. Sweldens, W., 1995, September. Lifting scheme: a new philosophy in biorthogonal wavelet constructions. In Wavelet applications in signal and image processing III (Vol. 2569, pp. 68-79). SPIE.

113. Sweldens, W., 1996. The lifting scheme: A custom-design construction of biorthogonal wavelets. Applied and computational harmonic analysis, 3(2), pp.186-200.

114. Sweldens, W., 1998. The lifting scheme: A construction of second generation wavelets. SIAM journal on mathematical analysis, 29(2), pp.511-546.

115. Murala, S., Gonde, A.B. and Maheshwari, R.P., 2009, March. Color and texture features for image indexing and retrieval. In 2009 IEEE International Advance Computing Conference (pp. 1411-1416). IEEE.

116. Skulsujirapa, P., Aramvith, S. and Siddhichai, S., 2004, July. Development of digital image retrieval technique using autocorrelogram and wavelet based texture. In The 2004 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS'04. (Vol. 1, pp. I-273). IEEE.

117. Torrey, L. and Shavlik, J., Transfer learning, in Handbook of Research on Machine Learning Applications and Trends. Algorithms, pp.242-264.

118. Marcelino, P., 2018. Transfer learning from pre-trained models. Towards Data Science, 10, p.23.

119. Canziani, A., Paszke, A. and Culurciello, E., 2016. An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678.

120. Kim, J., Lee, J.K. and Lee, K.M., 2016. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1646-1654).

121. Keys, R., 1981. Cubic convolution interpolation for digital image processing. IEEE transactions on acoustics, speech, and signal processing, 29(6), pp.1153-1160.

122. Delp, E. and Mitchell, O., 1979. Image compression using block truncation coding. IEEE transactions on Communications, 27(9), pp.1335-1342.

123. Qiu, G., 2003. Color image indexing using BTC. IEEE transactions on image processing, 12(1), pp.93-101.

124. Yu, F.X., Luo, H. and Lu, Z.M., 2011. Colour image retrieval using pattern co-occurrence matrices based on BTC and VQ. Electronics letters, 47(2), pp.100-101.

125. Silakari, S., Motwani, M. and Maheshwari, M., 2009. Color image clustering using block truncation algorithm. arXiv preprint arXiv:0910.1849.

126. Fränti, P. and Kaukoranta, T., 1999. Binary vector quantizer design using soft centroids. Signal Processing: Image Communication, 14(9), pp.677-681.

127. Wu, Y. and He, K., 2018. Group normalization. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

128. Zhang, X., Zhou, X., Lin, M. and Sun, J., 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6848-6856).

129. Kumawat, S., Kanojia, G. and Raman, S., 2021. ShuffleBlock: Shuffle to regularize deep convolutional neural networks. arXiv preprint arXiv:2106.09358.

130. Shakarami, A. and Tarrah, H., 2020. An efficient image descriptor for image classification and CBIR. Optik, 214, p.164833.