GABOR CONVOLUTIONAL NEURAL NETWORK THAT INTEGRATES BOTH ROTATIONAL AND ILLUMINATION INVARIANCE TO IMPROVE THE PERFORMANCE OF CONTENT-BASED IMAGE RETRIEVAL

JUDY MUTHONI GATERI

DOCTOR OF PHILOSOPHY

(Computer Science)

JOMO KENYATTA UNIVERSITY

OF

AGRICULTURE AND TECHNOLOGY

2024

Gabor Convolutional Neural Network that Integrates Both Rotational and Illumination Invariance to Improving the Performance of Content-Based Image Retrieval

Judy Muthoni Gateri

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Science of the Jomo Kenyatta University of Agriculture and Technology

2024

DECLARATION

This thesis is my original work and has not been submitted for a degree in any other university.

Signature…….. Date……

Judy Muthoni Gateri

This thesis has been submitted for examination with our approval as the university supervisor

Signature…….. Date……

Dr. Richard Rimiru, PhD

JKUAT, Kenya

Signature…….. Date……

Dr. Michael Kimwele, PhD

JKUAT, Kenya

DEDICATION

This Thesis is dedicated to my husband Titus, My children Maxwell, Sheerah, and Mark, and My parent Peris Wakari for giving me easy moments during my studies. Special dedication to my late dad John Gateri for believing in me in all things. Special dedication to my siblings Dr. Waithera, Wambui, Kingi, and Njeri for supporting me tirelessly during the entire PhD journey.

ACKNOWLEDGMENT

I thank God for granting me the opportunity to walk the PhD journey.

There are some people without whom this thesis might not have been written, and to whom I am greatly indebted.

I utmost appreciate my supervisors Dr. Richard Rimiru and Dr. Michael Kimwele, whose guidance, and encouragement I will always be grateful for, and for working closely with me on the thesis. You took the time to read, edit, and correct my thesis. God bless you abundantly.

I especially thank the entire faculty members in the School of Computing and Information Technology for their positive feedback during seminar presentations. Dr. Mwalili created time from his busy schedule and guided me when I reached out to him. I thank Mercy, Emily, and Nyamai for the love and encouragement they offered to me.

Evelyn Nasambu always rekindled the dedication and interest in the work when it looked like there was no light at the end of the tunnel. Thank you very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ACRONYMS AND ABBREVIATIONS

CBIR	Content-Based Image Retrieval
RGB	Red, Green, Blue
HSV	Hue Saturation Value
CIELAB	Color Space Specified by the International Commission on Illumination (French Commission internationale de l'éclairage, hence its CIE initialism).
FFBP	Feed Forward Backpropagation
DCNN	Deep Convolutional Neural Network
CNN	Convolutional Neural Network
GCNN	Gabor Convolutional Neural Network
SPP	Spatial Pyramid Pooling
GLCM	Gray Level Cooccurrence Matrix
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features

ABSTRACT

Content-Based Image Retrieval (CBIR) is the main stay of current image retrieval systems where a user submits an image based query which is then used by the system to extract visual features like shape, color or texture from images. For the comparison of rotated images to be more effective these images should be insensitive to illumination changes, occlusion changes, pose changes, and less sensitive to noise. When the image rotates, its angle position changes, altering its original consistency and form and limiting it from generating similar outcomes. Furthermore, the way patterns are arranged in an image can distort its appearance, and lighting conditions can alter an image's brightness and yield different outcomes, making it challenging to find similar images. Color has been utilized previously to improve illumination where images were first transformed from RGB to HSV color before they were fed as input in the network model. Contrast texture features have also been combined with other texture descriptors to compute rotation-invariant representations of textures. Gabor Convolutional Neural network methods have also been utilized in rotation invariance. Gabor Convolutional Neural Networks are built to capture texture and spatial frequency information effectively, but they do not inherently account for intensity or lighting shifts. Gabor filters are sensitive to the absolute intensity of input images, meaning brightness changes significantly alter filter responses. This sensitivity can lead to inconsistent feature extraction under varying orientation and lighting conditions, affecting the GCNN's performance in image retrieval. When applied to RI-GCN, where the first layer of convolutional neural networks was convolved with a Gabor filter, and TI-GCN, where the first and last layers were convolved with a Gabor filter, it has shown good results for rotation invariance. Nevertheless, when the image is retrieved throughout the Networks, it continues to encounter challenges with variations in illumination and rotation. This research investigates the effects of integrating these filters into different CNN layers, early, middle, late, combined configurations, and all layers to assess their impact on rotational and illumination invariance across datasets of varying complexity, specifically CIFAR-10 and ImageNet. Experimental results demonstrate that specific layer configurations optimize performance, with early and middle layers providing fundamental color and texture differentiation for simpler datasets, and deeper layers effectively handling complex features in more challenging datasets. The combined configurations enhance both rotational and illumination robustness, contributing to improved retrieval accuracy, precision, and recall. The findings underscore the importance of adaptive, multi-layer filter integration, offering a promising direction for developing robust and efficient CBIR systems. Our results show the model is applicable in various areas such as medical Imaging to retrieve relevant diagnostic images accurately, regardless of how they were captured, facilitating better comparisons, diagnostic accuracy, and treatment planning.

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Content-based image retrieval (CBIR) is a technique used in computer vision and information retrieval to search and retrieve images from a large database based on their visual content. Typically, "content" in the context of image processing refers to the significant information that an image represents. CBIR, uses the basic visual features of an image such as color, shape, texture, and spatial layout to represent and index the images. The term "content or features" in this context might refer to color, shape, texture, or any other information that can be derived from the image itself (Mufarroha et al., 2020).

Within the computing domain, we can describe an image as a matrix of numbers. These numbers are represented as different light intensities which altogether make up the image. A pixel is the term representing the numbers within the matrix assorted in rows and columns. The pixel values are data dependent and can range from a minimum to a maximum value. For simplicity sake, let’s say that the pixel light intensity values are represented as an 8-bit number, or a range of 0 to 255. The higher the value, the greater the intensity (the pixel appears brighter with a higher intensity). Colored images are overlapping matrices with 3 separate channels: red, green, and blue. In image processing, handling color images requires more processing than grayscale because there are 3 channels to compute on instead of 1. Because images are represented in terms of matrices full of intensity values, we can use computation to process and analyze an image or an array of images (Garcia et al., 2020).

According to Srivastava et al. (2023) refer to "retrieval" as the process of searching for and accessing images from a collection or database based on specific criteria or queries. Image retrieval involves finding images that are relevant to a given task, such as matching a particular visual pattern, identifying objects, or retrieving images similar to a query image. Two popular techniques are text-based image retrieval

(TBIR), which retrieves images based on related textual annotations or metadata, and content-based image retrieval (CBIR), which retrieves images based on their visual content.

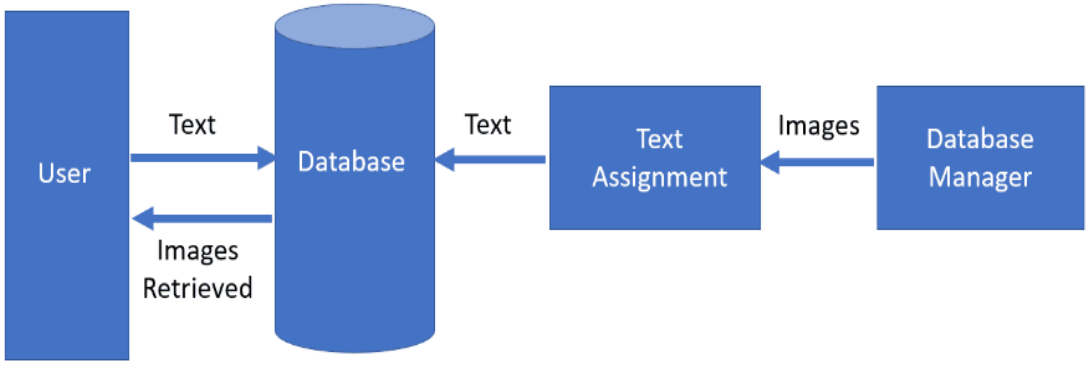

Figure 1.1: Text-Based Image Retrieval System

Figure 1.1 illustrates how Text-based image retrieval also known as traditional image retrieval (TBIR) retrieves images. The text-based image retrieval approach interprets image by text and then uses textual information to retrieve images from a text-based database management system. The greatest advantage of TBIR is that when images are recorded correctly, good search results can be achieved.

This method has several drawbacks; it uses keywords associated with images to retrieve visual information. Keywords due to their subjective natures fail to bridge the semantic gap between the retrieval system and the user demands; consequently, the accuracy of the retrieval system is questioned (Gasmi et al., 2023). The keyword for describing images becomes inadequate in large databases and it is not scalable.

Text-Based Image Retrieval (TBIR) is non-standardized, as different users often use varying keywords for annotation. This process can sometimes be incomplete, and it relies on humans to manually describe each image, making it inconsistent and labor-intensive. Although TBIR is still in use it is cumbersome and labor intensive. Due to these disadvantages, CBIR proved to be more promising and efficient than TBIR. The main advantage of using CBIR system is that the system uses image features

instead of using the image itself. So, CBIR is cheap, fast, and efficient over image search methods.

CBIR, uses the basic visual features of an image such as color, shape, texture and spatial layout to represent and index the images. The term "content or features" in this context Goldwasser (2024) might refer to color, shape, texture, or any other information that can be derived from the image itself. In typical content-based image retrieval systems, the features of every image are extracted and grouped to form the feature vector. All the feature vectors are stored in the feature database. Similarity matching between the feature vector of the query image and the feature vector database is carried out to retrieve similar images. CBIR systems extract features from the raw images themselves and calculate an association measure (similarity or dissimilarity) between a query image and database images based on these features. CBIR is becoming very popular because of the high demand for searching image databases of ever-growing size. Figure 1.2 demonstrates how a content-based image retrieval system retrieves images.

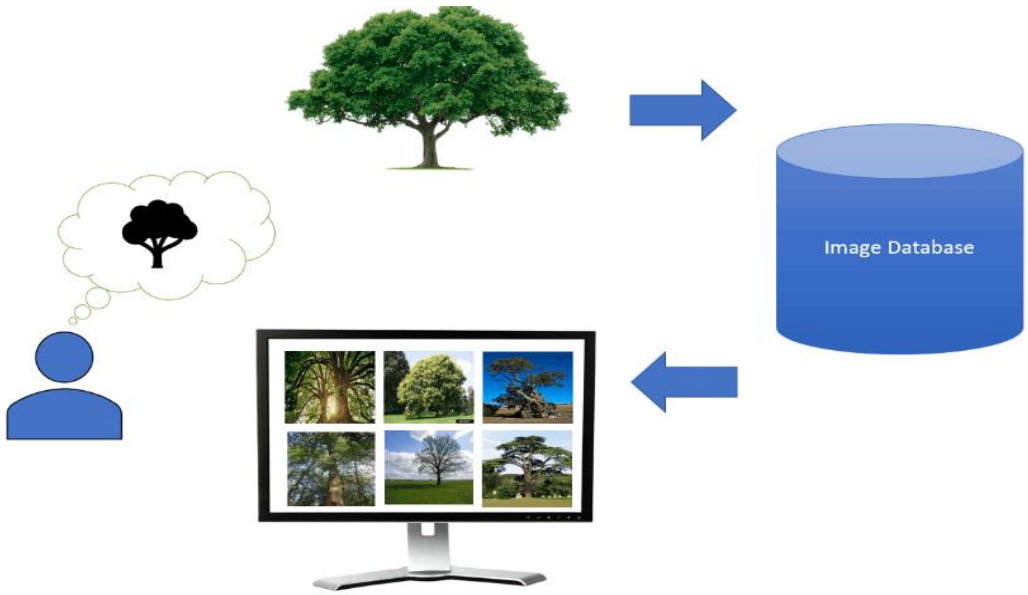

Figure 1.2: Content-Based Image Retrieval System

CBIR basically uses the visual contents of an image. There are two levels for extracting features in Content-Based Image Retrieval (CBIR) (Hameed et al., 2021) High level feature and Low level feature.

Levels for Extracting Features in Content-Based Image Retrieval (CBIR)

To efficiently retrieve images based on their visual content, Content-Based Image Retrieval (CBIR) systems use several levels of feature extraction. There are various layers to the extraction process, each of which focuses on distinct feature types that enhance the system's overall performance.

High-Level Features

High-level features (concepts) are keywords, text, and visual features that interpret images and find similarities between pixels of images (Chen et al., 2021). The high-level feature is also known as the semantic feature. There are three conventional methods for extracting the image semantic feature.

1. First, semantic features can be extracted based on image processing and domain knowledge. There are three fundamental processes: image segmentation, object recognition, and analysis of the object relation.
2. Secondly, Images' semantics features can be extracted through manual tagging or human interaction, but both require arduous human effort.
3. Third, we can extract the semantics from external information, such as file name, URL, image near the text, or metadata information.

Low Level Feature

Low Level feature extraction includes color, texture, shape, spatial information as represented in Figure 1.3.

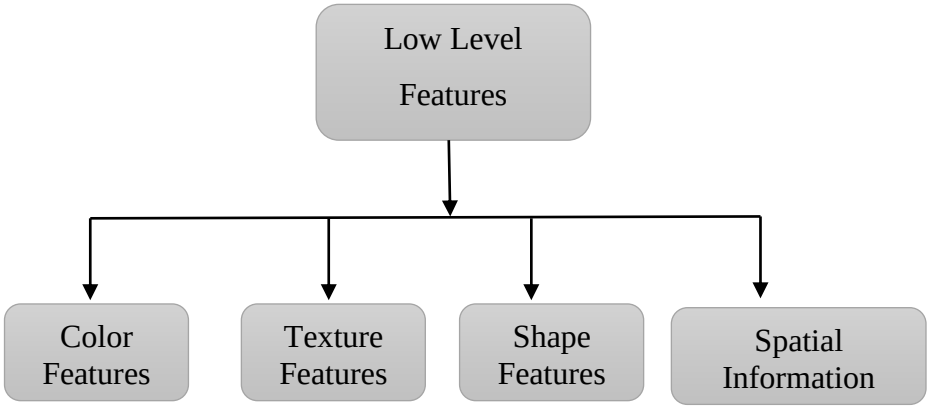

Figure 1.3: Representation of Low Level Features

Color - Color reflects the chromatic attributes of the image (Anowar, 2023). Color is one of the most reliable used low level visual features and is invariant to image size and orientation. Color features are widely used in CBIR systems as they are independent of image size and orientation (Dowerah & Patel, 2024). They are usually extracted from different color spaces, e.g., RGB, HSV, YCbCr, by computing the color histogram, color moments, or dominant colors. Color is the sensation caused by the light as it interacts with our eyes and brain. Color features are the fundamental characteristics of the content of images. Human eyes are sensitive to colors, and color features enable humans to distinguish between objects in the images. Colors are used in image processing because they provide powerful descriptors that can be used to identify and extract objects from a scene. Also, it is more reliable and easy to implement. Color features provide sometimes powerful information about images, and they are very useful for image retrieval (Liu et al., 2024).

Color feature is an essential component for image retrieval. For huge image databases, image retrieval using the color feature is very successful and effective. Although color feature is not a persistent parameter, because it is subjected to many

non-surface characteristics for example, the taking conditions such as illumination and characteristics of the device color are still popular in feature extraction (Zhou et al., 2024).

Texture: It is another important property of images. Various texture representations have been investigated in pattern recognition and computer vision. Texture refers to the surface properties of an object and its relationship to the surrounding environment. According to Tamura et al. (1978), texture is a measure of coarseness, contrast, directionality, line-likeness, regularity, and roughness. The texture can also be seen as a similarity grouping in an image or as natural scenes containing semi-repetitive arrangements of pixels (Chen et al., 2021).

Texture in the realm of image processing gives information about the local spatial arrangement of colors or intensities in a given image. Images that have similar texture properties should therefore have the same spatial arrangements of colors or intensities, but not necessarily the same colors. Because of this, the use of texture-based image indexing and retrieval techniques is quite different than those used strictly for color. Texture consists of some basic primitives, and also describes the structural arrangement of a region and the relationship of the surrounding regions. Texture features can be classified into two categories, firstly spectral features such as (the Gabor filter and discrete wavelet transformation) and secondly, statistical features such as (the word feature, Tamura feature, and gray-level co-occurrence matrix representation).

Shape: Shape features of objects or regions have been used in many content-based image retrieval systems. Compared with color and texture features, shape features are usually described after images have been segmented into regions or objects. Since robust and accurate image segmentation is difficult to achieve, the use of shape features for image retrieval has been limited to special applications where objects or regions are readily available. The state-of-art methods for shape description can be categorized into either boundary-based or region-based methods. A good shape representation feature for an object should be invariant to translation, rotation and

scaling. The major problem in the use of shape is how to represent shape information.

Spatial: Regions or objects with similar color and texture properties can be easily distinguished by imposing spatial constraints. For instance, regions of blue sky and ocean may have similar color histograms, but their spatial locations in images are different. Therefore, the spatial location of regions or the spatial relationship between multiple regions in an image is very useful for searching images.

One of the most indispensable problems in image processing and pattern recognition is determining how to extract effective features from images. The extraction of determining attributes that provide a better description of the images in the database usually follows the process of matching the similarity of the appropriate images. As a result, retrieving images requires comparing previously acquired images that represent the images' important aspects rather than matching them to the whole images. In most cases, we can observe and photograph an object from various angles. This means that the feature must be invariant to various types of spatial deformations caused by imaging geometry, such as two-dimensional translation, rotation, illumination, similarity transform, affine transform, and so on. Figure 1.4 illustrates the process of querying and retrieving images from a database.

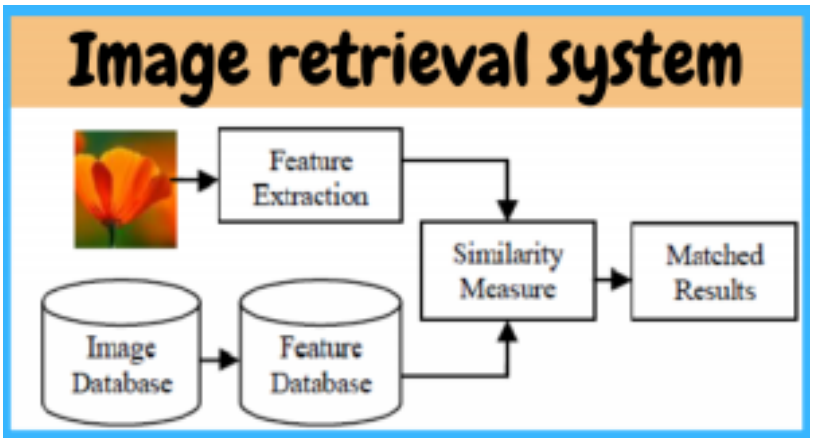

Figure 1.4: Image Retrieval System

One important aspect of image processing is dealing with rotational and illumination invariance. In image processing, rotational invariance refers to a feature or property

of an image processing technique that maintains its original form or consistency despite rotations of the input image. In other words, a rotationally invariant algorithm or feature will produce a similar outcome regardless of the orientation of the image if the image is rotated. Illumination variation, on the other hand, refers to differences in lighting conditions across an image. These differences have the potential to distort or misrepresent the original scene, as well as impair the efficacy of image analysis algorithms.

When an image rotates, the angle position of the image changes, and hence the image loses its original form or consistency and fails to produce similar output. Also, the appearance of the image changes due to distortion of the arrangement of patterns in an image and lighting conditions that affect the brightness of the image and yield different outputs thus making it challenging to assess and extrapolate similar images. For effective retrieval, these images should maintain the same pattern and color at all angles. A solution to the problem is to continuously have images that are insensitive to rotational and illumination invariance. However, many of the models only increase the model's resistance to rotations and illumination without offering any assurances about rotational and illumination invariance (Bagewadi & Veerashetty, 2023). Some Gabor filter variants do handle illumination changes (Veerashetty et al., 2022), and others handle both rotational and illumination invariance (Hong & Guan, 2021).

Traditional Convolutional Neural Networks (CNNs) are adept at extracting spatial and texture features, but they face limitations when applied to datasets with diverse image conditions (Archana & Jeevaraj, 2024). For example, CIFAR-10 and ImageNet, two popular datasets, vary significantly in complexity, with CIFAR-10 generally containing simpler patterns while ImageNet presents a wide range of high-level features and textures. This diversity in dataset complexity underscores the need for enhanced feature extraction techniques that can provide both rotational and illumination invariance, which are critical for consistent retrieval performance.

However, some models combine rotational and illumination invariance, making it difficult to retrieve images effectively when rotating them at different angles while preserving their color and texture. Salas (2019) created a network architecture that

uses CNN layers for classification after preprocessing with Gabor filters to extract rotation-invariant information. Yao and Song (2022) presented a model to learn Gabor-guided deep convolutional features by first rotating each input image to create several rotating image features, which are then fed into a network structure. Zhuang et al. (2022) presented transformation-invariant Gabor convolutional networks (TI-GCNs) by substituting the last layer of traditional convolutional layers with GCL to enhance rotation invariance.

Gabor Convolutional Neural Networks (GCNNs) face challenges in handling illumination changes primarily due to their reliance on fixed Gabor filters and convolutional operations that assume consistent intensity patterns. Changes in illumination alter the intensity distribution of an image, causing the Gabor filters to produce inconsistent responses. This sensitivity degrades the network's ability to extract robust features under varying lighting conditions. Identifying the most effective CNN layer for rotation and illumination invariance is a complex task. To address these challenges, this study proposes a layer-by-layer integration of HSV color space, directional texture filters, and Gabor filters within a CNN architecture. By evaluating each configuration's impact on performance whether applied in early, middle, late, or across all layers. The research aims to determine an optimal strategy for integrating these filters. This adaptive approach intends to enhance CBIR accuracy and reliability across datasets of varying complexities, thereby advancing the robustness and flexibility of image retrieval systems.

By incorporating the Gabor filter is a useful tool for feature extraction in image representation learning because it can extract images that are similar and reliable from all angles (Titrek & Baykan, 2023). The Gabor functions are derived from a sinusoidal plane wave of a specific frequency and orientation that describes the image's spatial frequency information. The purpose of directional texture filters is to extract specific patterns and structures from an image. These local patterns are consistent and frequently hold true when an image rotates. The HSV color space on the other hand, aims to provide a more natural representation of color compared to RGB color space and to divide color information into three independent parts, making it simpler to work with and modify color in an image (Rimiru et al., 2022;

Dhakshayani & Surendiran, 2023). Due to this division, the system can concentrate on the inherent color qualities (Hue), regardless of variations in brightness (Value) or intensity (Saturation). The integration of GCNNs, directional texture filters, and HSV color space forms a powerful model that addresses rotational and illumination invariance, making CBIR systems more reliable and effective across diverse conditions.

1.2 Problem Statement

One important aspect of image processing is dealing with rotational and illumination variation. In image processing, rotational invariance refers to a feature or property of an image processing technique that maintains its original form or consistency despite rotations of the input image. In other words, a rotationally invariance algorithm or feature will produce similar outcome regardless of the orientation of the image if the image is rotated. Illumination variation, on the other hand, refers to differences in lighting conditions across an image. These differences have the potential to distort or misrepresent the original scene, as well as impair the efficacy of image analysis algorithms.

GCNNs are built to capture texture and spatial frequency information effectively, but they do not inherently account for intensity or lighting shifts. Gabor filters are sensitive to the absolute intensity of input images, meaning brightness changes significantly alter filter responses. The reliance on Gabor filters makes them highly effective for capturing texture and orientation, but their lack of adaptation to intensity and illumination changes in different layers limits their robustness. This sensitivity can lead to inconsistent feature extraction under varying orientation and lighting conditions, affecting the GCNN's performance in image retrieval. According, Bagewadi and Veerashetty (2023), the Gabor filter only increase the model's resistance to rotations and illumination without offering any assurances about rotational and illumination invariance. Veerashetty et al. (2022) used Gabor filter to handle illumination changes, Hong and Guan (2021) used Gabor filter to handle both rotational and illumination invariance. Yao and Song (2022) constructed GCNNs that are rotationally invariant in the first layer. Additionally, Zhuang et al. (2022)

presented transformation-invariant Gabor convolutional networks (TI-GCNs) by substituting last layer of traditional convolutional layers with GCL to enhance rotation invariance. These techniques have achieved a lot but still fails to retrieve images effectively when rotating them at different angles while preserving their color and texture.

Identifying the most effective CNN layer for rotation and illumination invariance is a complex task. Applying rotation and illumination invariance within a single layer often falls short, as each layer in a CNN is specialized for certain types of feature extraction, from simple edges in early layers to complex textures and patterns in later layers. To address this, the study proposes an extended Gabor CNN model. The extended Gabor CNN model integrates HSV color, directional texture, and Gabor filter. The research aims to identify how each layer configuration influences retrieval performance of CBIR by testing the impact of each individual layers (early, middle, combined, and late). This approach allows a more detailed analysis of how each layer configuration contributes to image retrieval accuracy, color robustness, and texture orientation invariance, ultimately helping to determine whether a specific configuration or combination of layers offers superior performance across different datasets, such as CIFAR-10 and ImageNet. The HSV color space maintain consistency under lighting changes, Directional texture reinforces rotational invariance and Gabor filter provides rotation invariance regardless of angle changes.

1.3 Research Questions

The study was guided by the following questions:

i. What are the design limitations of Gabor Convolutional Neural Networks (GCNNs) in extracting color and texture for content-based image retrieval?
ii. What design techniques are used by Gabor Convolutional Neural Networks (GCNNs) to ensure rotational invariance for texture features in content-based image retrieval?
iii. What design techniques of Gabor Convolutional Neural Networks (GCNNs) handle illumination invariance for color and texture-based feature?

iv. How can integrating both rotational and illumination invariance in Gabor Convolutional Neural Networks (GCNNs) improve the performance of content-based image retrieval system?

1.4 Objectives

1.4.1 General Objective

The general objective of this study was to design and implement a Gabor Convolutional Neural Networks (GCNN) that integrate both rotational and illumination invariance to improve the performance of color and texture-based CBIR system.

1.4.2 Specific Objectives

The specific objectives of the study are

i. To analyze the design limitations of Gabor Convolutional Neural Networks (GCNNs) in extracting color and texture features.
ii. To analyze the effectiveness of Gabor Convolutional Neural Networks (GCNNs) in handling rotational invariance for texture features and identify their weakness.
iii. To investigate the ability of GCNNs to extract illumination invariance features and identify their weakness.
iv. To design and validate GCNNs prototype that integrate both rotational and illumination invariance to improve CBIR performance for color and texture-based features.

1.5 Practical Implication of Rotation and Illumination Invariance

CBIR provides a solution for many types of image information management systems as follows.

Medical imaging - In medical imaging (e.g., radiology, histology), images may be captured at various angles and under different lighting, depending on the equipment

or conditions during examination. The model would help to retrieve relevant diagnostic images accurately, regardless of how they were captured, facilitating better comparisons, diagnostic accuracy, and treatment planning.

Forensic and security application - In forensic investigations, CBIR can be used to match images (e.g., faces, objects, or scenes) across surveillance footage that may be captured from different angles and under various lighting conditions. The model identifies suspects, objects, or activities from multiple viewpoints and lighting situations, enhancing security and evidence reliability.

Retail and e-commerce - Retailers often rely on visual search technology for product recommendations and similar-item searches, with images submitted by users often taken at different angles and under various lighting conditions. The model recognizes products accurately, providing customers with consistent recommendations and similar-item searches, regardless of how they photograph the items.

1.6 Contribution of the Thesis

As our first contribution, this research introduces an extended Gabor CNN model. The extended model integrates HSV color, directional texture, and Gabor filters into the CNN model. As a second contribution, this research presents the performance evaluation for the extended Gabor CNN model. The evaluation results show that the HSV color is illumination invariance. Lastly, the results show that the directional texture filter and all Gabor CNN layers are rotation invariant.

1.7 Thesis Organization

This thesis is divided into five chapters as follows.

Chapter 2 presents the literature review of feature extraction and retrieval. This involves classifying the extracted features from images into three categories: low, mid, and high-level features. Then the chapter examines color and texture image extraction techniques are examined, followed by similarity matching techniques for CBIR. Finally, rotation and illumination invariance techniques.

Chapter 3 presents the research methodology. This chapter first, describes system development approach followed by conceptual framework and discussion. Secondly, the chapter details the workflow which describes how image features were extracted. Thirdly, chapter describes how the system architecture and design was applied to rotational invariance. Finally, the chapter describes how both rotation and illumination prototype was developed and tested.

Chapter 4 presents the results and discussion of the experiments. The results are analyzed and presented in the form of tables and charts.

Chapter 5 presents the summary of the thesis, the conclusions and gives the directions of future work.

CHAPTER TWO

LITERATURE REVIEW

The chapter presents the literature review. It begins by giving general introduction about images. This is followed by techniques for extracting image features such as machine learning techniques. The chapter ends by presenting the summary of the literature review.

2.1 Introduction

An image is represented by its dimensions (height and width) based on the number of pixels. For instance, if an image is 500 × 400 (width x height), then 200000 pixels make up the entire image. This pixel is a point on the image that assumes a certain shade, opacity, or color. Typically, it appears in Grayscale where gray is an integer with a value of 0 to 255, or totally black or completely white, is referred to as a pixel, RGB (Red, green, and blue) pixels are made up of three integers with values ranging from 0 to 255 or RGBA an expansion of RGB that includes an additional alpha field that symbolizes the opacity of the image. Thus, each pixel of an image must undergo a fixed series of operations during image processing (Mishra et al., 2022). The initial series of actions are carried out pixel-by-pixel by the image processor on the image. The second operation will start once this is finished in full, and so forth. Any pixel in the image can be used to determine the output value of these procedures. Therefore, image processing is a technique for taking a physical image and putting it into digital form so that you may manipulate it and add to it or take information out of it. It is a kind of signal distribution where the input is an image, such as a video frame or a photograph, and the output could be another image or information related to that image (Schütz et al., 2022).

Typical image processing systems treat images as two-dimensional signals and process them using pre-established signal processing techniques. The purpose of image processing is divided into five groups namely visualization, Sharpening and restoration, Retrieval, Recognition, and Pattern recognition. The process of retrieving images from a database is called content-based image retrieval (CBIR). In CBIR, a

user enters a query image to retrieve database images matching the query image. CBIR compares the content of the input image to the database images in order to determine the images that are the most similar. More specifically, CBIR assesses the degree of visual similarity between the query image and the database images in terms of elements including forms, colors, texture, and spatial information (Burger & Burge, 2022).

Nowadays, CBIR (Content-based image retrieval) is a hotspot of digital image processing techniques. CBIR research started in the early 1990s and is likely to continue during the first two decades of the 21st century (Zhou et al., 2024). The growing demands for image retrieval in multimedia fields such as crime prevention, Fashion, graphic design, and biometrics have pushed application developers to search for ways to manage and retrieve images more efficiently. Retrieval focuses on developing new techniques that support effective searching and browsing of large digital image libraries based on derived image features. Manual browsing of the database to search for identical images would be impractical since it takes a lot of time and requires human intervention. A more practical way is to use content-based image retrieval (CBIR) technology. CBIR has provided an automated way to retrieve images based on the content or features of the images themselves. The CBIR system simply extracts the content of the query image and matches it to the contents of the search image (Hadid et al., 2023).

CBIR is defined as a process to find similar images or images in the image database when a query image is given. Given a picture of a car, the system should be able to present all similar images of a car in the database to the users. This is done by extracting the features of the images such as color, texture, and shape (Choe et al., 2022). These image features are used to compare between the query image and images in the database. A similarity algorithm is used to calculate the degree of similarity between those two images. Images in the database which has similar images features to the query image (acquiring the highest similarity measure) is presented to the user.

Accurate retrieval findings are still a challenge and a current study topic. With content-based image retrieval (CBIR), similar images from a big image database are found for a given query image based on how closely their contents match. In CBIR, feature extraction from images is the initial stage. It is possible to use a variety of features, including color, texture, shape, and spatial distribution. Color histograms, Gabor filters, Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), and Convolutional Neural Networks (CNN) are examples of frequently used feature extraction methods. Following retrieval, a similarity measurement is needed to compare the features of the query image with the features of the images in the database. Euclidean distance, Manhattan distance, Cosine similarity, correlation coefficients, indexing, and relevance feedback are examples of common similarity measurements. Relevance feedback is a technique that allows users to provide feedback on the retrieved results to improve subsequent searches. Users can mark images as relevant or irrelevant, and the system uses this feedback to change its retrieval technique. The retrieval system's accuracy can eventually be increased through this iterative approach (Wang et al., 2023). With the rise of deep learning, convolutional neural networks (CNNs) have become a popular choice for CBIR. CNNs can learn complex features directly from the raw image data.

2.2 Categories of Image Feature Extraction

Image feature extraction refers to the process of identifying and extracting meaningful and distinctive patterns, structures, or characteristics from an image. These features can represent various aspects of the image, such as edges, corners, textures, colors, or shapes. Finding the most pertinent information in the original data and representing it in a lower dimensional space are the fundamental objectives of feature extraction. These extracted features work as a condensed representation of the image, making it possible to handle tasks like image segmentation, object recognition, image retrieval, and classification quickly and effectively. Low-level features, mid-level features, and high-level features are three general categories for image feature extraction approaches (Lu et al., 2023).

These classifications show the various degrees of abstraction and complexity in capturing image information. Figure 2.1 shows the summary of image feature extraction categories.

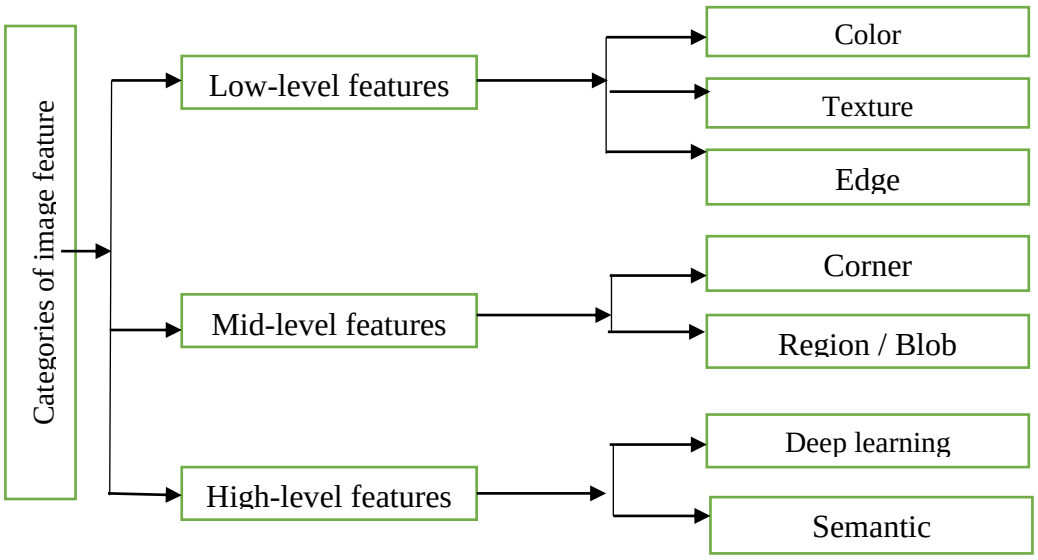

Figure 2.1: Categories of Image Feature Retrieval

2.2.1 Low-Level Features

Low-level features capture basic and primitive visual characteristics of an image, such as color, texture, shape, and spatial. Low-level features that are often used include the following.

Color

In image processing, color is a fundamental attribute that aids in distinguishing objects, defining patterns, and enhancing the perception of details within an image. Colors are represented as numerical values within a color space, with popular models like RGB (Red, Green, Blue), HSV (Hue, Saturation, Value), and Lab used to quantify colors in ways that align with human perception or computational requirements. According to Phuangsaijai et al. (2021) RGB is common in digital displays, representing colors through combinations of red, green, and blue intensities. HSV, on the other hand, separates the color's hue (its type), saturation (its intensity),

and value (its brightness), making it effective for tasks that need color differentiation without being affected by brightness variations. The choice of color model significantly impacts how color information is extracted, analyzed, and manipulated in image processing applications.

Color plays a crucial role in applications such as content-based image retrieval (CBIR), object recognition, and segmentation, where the ability to distinguish one object from another often depends heavily on color consistency (Srivastava et al., 2023). However, real-world variations in lighting conditions, shadows, and reflections can cause colors to appear differently, complicating color-based analysis. For instance, an object photographed under sunlight may look different when captured under artificial light. This shift can lead to reduced accuracy in color-based retrieval or recognition tasks if not addressed properly. Methods like HSV and Lab color spaces help mitigate these issues by decoupling color from brightness, allowing systems to remain more resilient to such changes.

To enhance image retrieval or recognition, advanced processing techniques such as histogram equalization or color constancy adjustments are used to correct and normalize color variations caused by differing illumination (Ebner, 2021). Additionally, using color spaces like HSV enables models to focus on hue while minimizing sensitivity to brightness, which is especially valuable for applications that operate under varied lighting conditions. This approach improves the model's robustness, ensuring more accurate and consistent color-based analysis and retrieval even when images are captured in diverse environments.

Texture

Texture in image processing refers to the repeated patterns, structures, or spatial arrangements of pixels within an image that represent the surface characteristics of objects (Chaki & Dey, 2020). Texture conveys essential information about an object's appearance, such as its roughness, smoothness, or granularity, and is often crucial in distinguishing between different objects or surfaces. Mathematically, texture is characterized by statistical, structural, or spectral features, which capture the variation and distribution of pixel intensities across an image region. Common

methods to analyze texture include techniques like gray-level co-occurrence matrices (GLCM), local binary patterns (LBP), and Gabor filters, each capturing different aspects of texture and aiding in effective feature extraction.

In image processing applications such as content-based image retrieval (CBIR), object recognition, and medical imaging, texture analysis plays a vital role in identifying and classifying objects based on their surface patterns (Raju et al., 2020). For instance, in CBIR, texture features help distinguish between images of similar color but different textures, like differentiating between images of sand and water. Unlike color, which may change due to lighting variations, texture can provide a more stable descriptor for object identity, making it particularly useful in environments where lighting and orientation vary. By focusing on patterns rather than single-pixel values, texture descriptors help models capture the consistent appearance of materials and surfaces across different images.

To achieve robust texture analysis, advanced filtering techniques such as directional filters and Gabor filters are commonly employed. Directional texture filters can capture patterns oriented in specific directions, helping models remain invariant to rotations, while Gabor filters provide spatial and frequency information, identifying patterns across scales and orientations (Ghalati et al., 2021). This ability to capture and quantify multi-scale, multi-directional texture patterns enhances the accuracy of image processing models, particularly in challenging tasks that require distinguishing fine surface details or repeating patterns. With these methods, image processing systems can better interpret complex textures, improving performance in diverse applications, from industrial inspection to medical diagnostics.

Shape

In image processing, shape refers to the geometric outline or silhouette of an object within an image, providing essential information about its structure, form, and boundary. Raj and Balaji (2023), Shapes are fundamental descriptors in image analysis, as they allow for the identification and classification of objects based on their contours rather than relying solely on color or texture. Shape features are typically extracted by analyzing edges, contours, and boundaries, which highlight the

transition points between an object and its background. Techniques such as edge detection, contour extraction, and corner detection are commonly used to capture the defining boundaries of shapes, making shape analysis a core aspect of image segmentation, object recognition, and tracking.

Shape descriptors in image processing vary in complexity and type, ranging from simple geometric properties (like area, perimeter, and circularity) to advanced models like Fourier descriptors and shape contexts (Morgenstern et al., 2021). Simple descriptors help distinguish basic shapes, such as circles, squares, and triangles, based on size or symmetry, while advanced methods allow for the capture of intricate and irregular shapes by encoding both boundary information and relative positions of points along the shape. Fourier descriptors, for example, analyze shape contours in the frequency domain, enabling models to remain invariant to rotation, scaling, and translation. This adaptability makes shape descriptors valuable for applications like object recognition, where the appearance of an object may change due to camera angle, size, or position within the image.

In real-world applications, shape analysis is crucial in areas like medical imaging, robotics, and autonomous driving, where accurate object identification is essential. For instance, in medical imaging, shape-based methods help identify abnormalities, such as tumors, by comparing detected shapes to known anatomical structures (Wijayathunga et al., (2023). In robotics and autonomous systems, shapes allow for object detection and environment mapping, where identifying obstacles or paths relies on accurate boundary and contour recognition. The ability to effectively analyze shapes, combined with other features like color and texture, enhances the robustness of image processing systems, enabling them to perform reliably across diverse applications where understanding the form

Spatial

Spatial information refers to the arrangement and relative positioning of pixels or features within an image, providing context for object location, orientation, and relationships to other elements. Spatial features are crucial because they help maintain the structure and layout of objects in an image, giving insight into both

global and local contexts. For example, spatial relationships between various parts of an object (like the eyes, nose, and mouth on a face) can help systems recognize faces more effectively. By analyzing spatial patterns, image processing algorithms can differentiate between objects that have similar textures or colors but differ in structure or orientation (Caron et al., 2024).

Spatial analysis often involves capturing both low-level and high-level spatial characteristics. Low-level spatial features focus on pixel neighborhoods, edges, and regions, providing basic information about shapes, contours, and boundaries. High-level spatial features consider the arrangement of larger regions or objects, identifying patterns in the overall structure of an image (Yang et al., 2022). Techniques such as spatial filtering, feature mapping, and region-based segmentation are commonly used to capture spatial information. Convolutional Neural Networks (CNNs), for instance, utilize spatial hierarchies by stacking layers that capture increasingly complex spatial patterns, making CNNs effective for tasks like object detection and scene understanding.

Spatial information is especially valuable in applications like remote sensing, medical imaging, and autonomous navigation, where understanding an image's layout and the relationships between elements is crucial. According to, Pore et al. (2023) in medical imaging, spatial patterns can help locate and diagnose abnormalities within specific regions of an organ. In autonomous navigation, spatial relationships allow a vehicle to understand its environment by mapping out objects, roads, and obstacles to itself. The ability to leverage spatial information enhances the robustness and accuracy of image processing models, enabling them to interpret complex scenes and make decisions based on the spatial structure of objects and their surroundings.

2.2.2 Mid-Level Features

Mid-level features capture intermediate-level visual data and are more complicated than low-level features. These characteristics signify more important regions or structures in an image. Mid-level features include Corner features where Corner detection algorithms are utilized to identify distinctive points in an image where the

intensity gradient changes in multiple directions. These points are often used as key points for image matching or object tracking. Others are Blob or region features that represent regions of interest in an image that exhibit certain characteristics, such as scale-invariant key points or maximally stable external regions (MSER) (Ma et al., 2021).

2.2.3 High-Level Features

High-level features are more abstract and record higher-level ideas or semantic information in an image. These features, which reflect intricate visual patterns, are frequently acquired using deep learning models. Convolutional neural networks (CNNs) may learn hierarchical representations of images, where each layer captures ever more abstract and sophisticated characteristics (Sungheetha & Sharma, 2021). Others are semantic features that capture the semantic meaning of an image, such as the presence of specific objects or scenes. They are typically learned through supervised training using annotated datasets (Dey et al., 2021).

The choice of feature extraction technique depends on the specific task, the complexity of the image data, and the level of abstraction required for the analysis. Multiple levels of information can be extracted from an image using a variety of techniques, either in combination or sequentially, providing a richer representation for further analysis or machine learning tasks. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data to perform the desired task. The goal of feature extraction, a crucial stage in the development of any pattern classification, is to extract the pertinent data that distinguishes an image.

2.3 Color Image Extracting Techniques

Color is the most important feature visually recognized by humans and humans tend to distinguish images based mostly on color features. Color is a powerful descriptor that simplifies object identification, and is one of the most frequently used visual features for content-based image retrieval (Kim, 2020). To extract the color features from the content of an image, a proper color space and an effective color descriptor

must to be determined. The purpose of a color space is to facilitate the specification of colors. Each color in the color space is a single point represented in a coordinate system (Chaki & Dey, 2020). Several color spaces, such as RGB, HSV, and CIELAB have been developed for different purposes.

A color space is a method by which we can specify, create and visualize color. As humans, we may define a color by its attributes of brightness, hue, and colorfulness. A computer may describe a color using the amounts of red, green, and blue phosphor emission required to match a color. A color is thus usually specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used (Fang et al., 2022). Other spaces are confusing for the user with parameters with abstract relationships to the perceived color. Some color spaces are tied to a specific piece of equipment (device-dependent) while others are equally valid on whatever device they are used (Maharana et al., 2022).

Commonly used color spaces for image retrieval include RGB, CIE *L*a*b**, CIE *L*u*v**, HSV (or HSL, HSB), and opponent color space. There is no agreement on which is the best. However, one of the desirable characteristics of an appropriate color space for image retrieval is its uniformity. Uniformity means that two color pairs that are equal in similarity distance in color space are perceived as equal by viewers. In other words, the measured proximity among the colors must be directly related to the psychological similarity among them (Alsmadi, 2020). According to Vasan et al. (2020), different color spaces are better for different applications, for example, some equipment has limiting factors that dictate the size and type of color space that can be used. Some color spaces are perceptually linear, i.e., a 10-unit change in stimulus will produce the same change in perception wherever it is applied. Many color spaces, particularly in computer graphics, are not linear in this way. Some color spaces are intuitive to use, i.e., it is easy for the user to navigate within them, and creating desired colors is relatively easy.

Color extraction is susceptible to issues like variable lighting conditions and occlusions, therefore it might not be the most reliable technique. To solve these issues, in addition to the traditional RGB (Red, Green, Blue), alternative color spaces

are employed to extract color information from the image to be recognized, including HSV (Hue, Saturation, Value), HIS (Hue, Intensity, Saturation), and *L*a*b* (CIELAB), among others. Other techniques that solve these issues in addition to color space transformation are color histogram, color normalization, Color Invariant Descriptors, Color Correlogram, and Opponent Color Spaces (Afifi et al., 2021).

2.3.1 Color Histogram Technique

Color histograms are statistical depictions of the distribution of colors found in an image. They give a quantitative account of how frequently various color values appear in an image. The distribution of color values in an image is captured by color histograms. Color histograms make it possible to analyze and manipulate images based on their color features quickly and effectively since they compactly and visually reflect the color distribution (Bhunia et al., 2020). The process of creating a color histogram often involves dividing the color space into a predetermined number of bins or discrete intervals, counting the number of pixels that fall into each bin, and then plotting the results (Bianconi et al., 2020). According to Sethy et al. (2020), the color space used can vary, with common choices being the RGB (Red-Green-Blue) color space or its variants, such as HSV (Hue-Saturation-Value) or LAB color space. Each bin of the histogram represents a range or region of colors, and the value stored in each bin corresponds to the frequency or count of pixels that have colors falling within that range. The color histogram offers useful details on how colors are distributed in an image, including which colors predominate, whether there are color variations or gradients, and how the overall color scheme is composed. Color histogram is utilized for various purposes such as content-based image retrieval, object recognition, image analysis and manipulation, image segmentation, and others.

In content-based image retrieval (CBIR) systems, color histograms, images with similar color distributions can be retrieved by comparing the color histograms of query images with those in a database, providing effective image searching based on color similarity. Color histograms serve as descriptive features for objects in images. By comparing the color histograms of an object in a target image with reference

histograms of known objects, it is possible to recognize and classify objects based on their color characteristics (Vijayan et al., 2023).

Color histograms provide insights into the overall color composition of an image. They can be used for tasks such as color correction, image enhancement, color-based image editing, and artistic image transformations (Gonçalves, 2020). Color histograms provide a concise and informative representation of the color distribution in an image by capturing the frequency or occurrence of different colors and are widely used in various image processing tasks.

One major challenge of color histograms is the choice of binning method. The choice of binning methods, such as equal-width, equal-count, or adaptive binning, for creating a color histogram might affect the representation's accuracy and sensitivity to changes in color distribution. The discrimination power and perceptual similarity represented by the histogram are influenced by the color space used to compute it, such as RGB, HSV, or LAB. Different color spaces may be more suitable for specific applications or types of images (Sarker & Grift, 2023). Xiang et al. (2024) suggest an approach for underwater image improvement that is tailored for histogram segmentation and dual-color space color correction to address the issues of poor contrast, color distortion, and detail loss.

Other challenges are sensitive to lighting, and changes in illumination can have an impact on how well they operate. These difficulties are reduced using methods like histogram normalization and color space transformations. Integrating spatial information with color histograms, such as through spatial color histograms or joint color-texture histograms, improves the representation's discriminatory power and enables more accurate image retrieval, object recognition, and image segmentation (Tajjour et al., 2023).

Moreover, histogram equalization and its variants may not always produce satisfactory results when it comes to handling illumination variation. These techniques can sometimes lead to over-enhancement of certain regions in an image, causing unnatural and unrealistic outcomes. Furthermore, they may not be effective in scenarios where the illumination variation is too severe or uneven across the

image, making it challenging to achieve a balanced correction (Hu et al., 2020). Luo et al. (2023) improved the image with poor light using color histogram.

Although color histograms are reasonably resilient to variations in lighting and insensitive to spatial arrangement, they are not responsive to color order and only consider the total distribution of colors, potentially leading to the loss of crucial spatial information. Moreover, histogram equalization and its variants may not always produce satisfactory results when it comes to handling illumination variation. The techniques can sometimes lead to over-enhancement of certain regions in an image, causing unnatural and unrealistic outcomes. Furthermore, they may not be effective in scenarios where the illumination variation is too severe or uneven across the image, making it challenging to achieve a balanced correction (Hu et al., 2020).

2.3.2 Color Normalization Technique

Color normalization is a fundamental process in image processing that aims to ensure color consistency and remove unwanted variations in images. It plays a crucial role in numerous applications, including medical imaging, remote sensing, computer vision, and various other fields where accurate and reliable analysis of images is required. The following are some of the color normalization techniques.

Histogram Equalization

Histogram equalization is a widely used color normalization technique that adjusts the intensity distribution of an image, enhancing contrast and distributing pixel intensities more evenly. Variants like adaptive histogram equalization can focus on local areas, which is helpful in images with varying illumination. This method can be particularly beneficial in addressing issues related to illumination variance, as it aims to improve the visibility of features in images that may be obscured due to poor lighting conditions (Roy et al., 2024). Color Histogram also expands the dynamic range of pixel values, which can be particularly useful in medical imaging and other applications where detail is crucial. Color Histogram also has drawbacks in that first, while histogram equalization works well for grayscale images, its application to color images can lead to unnatural color representations (color distortion). This

occurs because it does not consider the interdependence of color channels, potentially exaggerating certain colors (e.g., blue) and resulting in unrealistic appearances. Secondly, traditional histogram equalization applies a global transformation across the entire image, which may not be suitable for images with varying local illumination (Global adjustment). This can lead to over-enhancement in some areas while under-enhancing others and finally, to address these limitations, variations such as adaptive histogram equalization (AHE) and contrast-limited adaptive histogram equalization (CLAHE) have been developed (Alternative techniques). These methods apply histogram equalization locally rather than globally, allowing for better preservation of color fidelity and detail in different regions of an image (Vijayalakshmi & Nath, 2023).

Images can be made more aesthetically pleasing by normalizing their colors, and subsequently performed analysis activities can be carried out more precisely and robustly. Different color spaces are commonly used in color normalization to represent and manipulate colors effectively (Swiderska-Chadaj et al., 2020). According to Tosta et al. (2023), RGB is the most widely known and used color space in digital imaging. It represents colors as combinations of red, green, and blue primary colors. Each pixel in an RGB image has three color channels, with values ranging from 0 to 255 (8-bit representation) or 0 to 1 (floating-point representation). RGB is intuitive and closely aligned with the way colors are displayed on electronic devices. However, it is highly sensitive to variations in lighting conditions, making it less suitable for color normalization purposes.

HSV is a color space that separates color information into three components: hue, saturation, and value. Hue represents the dominant color information, saturation determines the purity or intensity of the color, and value represents the brightness or lightness of the color. HSV provides a more intuitive representation of colors, and it is commonly used in color normalization tasks to address variations in lighting conditions. By manipulating the value component, it is possible to adjust the brightness or contrast of an image while preserving the hue and saturation information (Ansari & Singh, 2022).

The CIELAB color space consists of three components L for lightness, A for green-red color channel, and B for blue-yellow color channel. It is designed to be perceptually uniform, meaning that equal distances in LAB space correspond to equal perceptual differences. LAB is often used in color normalization tasks due to its ability to separate color information from luminance (brightness) information. By normalizing the A and B channels, color variations can be effectively mitigated, making it suitable for various applications, including medical imaging and computer vision (Phuangsaijai et al., 2021).

YCbCr is a color space used primarily in digital video and image compression. It separates the color information (chrominance) from the luminance (brightness) information. The Y component represents the luminance information, while the Cb and Cr components represent the blue-difference and red-difference chrominance information, respectively. YCbCr is widely used in color normalization techniques, particularly in applications involving image and video compression, as it allows for efficient compression by allocating more bits to the luminance channel than the chrominance channels (Yang et al., 2020).

Color normalization techniques also face several challenges that impact their robustness and effectiveness, especially in the presence of varying lighting conditions and complex scenes. Images captured under different lighting conditions may exhibit variations in color intensity, brightness, and color temperature. Color normalization methods need to be robust to these variations and ensure consistent colors across different lighting conditions. Images captured in complex scenes often have non-uniform illumination, where different parts of the image are exposed to varying lighting intensities. It becomes challenging to normalize colors accurately when the illumination across the image is not uniform.

2.3.3 Convolutional Neural Networks and Generative Adversarial Networks (GANs)

CNNs have revolutionized the field of computer vision and have been effectively employed for color normalization. They learn hierarchical representations of images by automatically extracting features at different levels of abstraction. There are several techniques to normalize color using CNNs such as transfer learning, and supervised learning.

Generative Adversarial Networks (GANs) consist of a generator and a discriminator network that compete against each other. GANs have been utilized for color normalization tasks by generating synthetic images with normalized colors. The generator network learns to transform images with color variations into color-normalized images, while the discriminator network tries to distinguish between real and generated color-normalized images. GAN-based color normalization methods have several advantages in that they can learn complex mappings between color variations and normalized colors, capture the underlying statistical properties of the color distribution in the training data, and enable the generation of visually appealing and realistic color-normalized images (Alqahtani et al., 2021). CNNs and GANs require large amounts of labeled training data to learn effective color normalization models. However, they have shown remarkable capabilities in capturing complex color relationships and generating high-quality color-normalized images. Continued research and advancements in deep learning architectures are expected to further improve the performance in handling different lighting conditions.

Color normalization techniques face several challenges that impact their robustness and effectiveness, especially in the presence of varying lighting conditions and complex scenes. Some of the major challenges are Changes in lighting conditions pose a significant challenge for color normalization. Images captured under different lighting conditions may exhibit variations in color intensity, brightness, and color temperature. Color normalization methods need to be robust to these variations and ensure consistent colors across different lighting conditions. Images captured in complex scenes often have non-uniform illumination, where different parts of the

image are exposed to varying lighting intensities. It becomes challenging to normalize colors accurately when the illumination across the image is not uniform. Color normalization techniques need to account for local variations in illumination and adapt to different regions within an image. Shadows and highlights can cause significant color variations in images. Shadows tend to introduce a bluish tint, while highlights can appear excessively bright and saturated (Huang et al., 2023).

2.3.4 Hybrid Approaches Combining Multiple Techniques

Hybrid approaches that combine multiple techniques can be effective for improving the performance of color histogram-based color normalization (Basar et al., 2020). By leveraging the strengths of different techniques, these hybrid approaches can address limitations and achieve more accurate and robust color normalization. Some examples of hybrid approaches are as follows.

Histogram Matching followed by Nonlinear Transformation. This hybrid approach involves first applying histogram matching to align the color distribution of an input image with a reference image or a desired target distribution. According to Bottenus et al. (2020) after histogram matching, a nonlinear transformation technique, such as gamma correction or histogram equalization, can be applied to further enhance the color consistency and contrast.

Histogram Equalization with Color Space Transformation. In this approach, histogram equalization is applied to each color channel individually. According to Rao (2020) after equalization, a color space transformation is performed to a color space that better separates color information from luminance, such as LAB or HSV. The transformed image can then undergo additional normalization steps, such as histogram matching or contrast adjustment. Additional hybrid approaches include Adaptive Combination and Ensemble of Data-Driven Models. According to Sarker (2021), the choice of techniques can be based on predefined rules, statistical analysis, or machine learning algorithms that learn the optimal combination for different image conditions.

A comprehensive survey presents an extensive overview of color normalization methods in computational pathology (Liu et al., 2020). It discusses different techniques, including histogram matching, stain separation, and deep learning-based approaches. The study highlights the significance of color normalization in improving the accuracy and reliability of computational pathology analysis. Yang et al. (2020) provide a comprehensive review of color normalization techniques for histopathological images. The study discusses various methods, including histogram-based approaches, color constancy, and machine learning-based techniques. The study also highlights the challenges of color normalization in histopathology and provides insights into the strengths and limitations of different methods. Ghosh and Banerjee (2020) provide a comprehensive review of color normalization methods in medical imaging. It covers various techniques, such as histogram-based methods, statistical approaches, and machine learning-based algorithms. The study discusses the challenges and recent advancements in color normalization and emphasizes the importance of accurate color representation in medical image analysis. Amiri et al. (2020) focus on color normalization techniques for computational pathology image analysis. It presents an overview of various methods, including stain normalization, color augmentation, and deep learning-based approaches. The study discusses the importance of color normalization in improving the performance and reproducibility of computational pathology algorithms. A comprehensive survey of color normalization techniques specifically for whole slide histopathological images (Praveen & Vasu, 2020). The author discusses various methods, including stain separation, color transfer, and deep learning-based approaches, and emphasizes the need for accurate color normalization in whole slide image analysis to ensure consistent and reliable results.

2.3.5 Color Invariant Descriptors

Color invariant descriptors are a fundamental component of image processing techniques, enabling robust and accurate analysis of images under varying lighting conditions and color appearances. According to Lengyel et al. (2021), traditional image descriptors, which solely rely on pixel values, are highly sensitive to changes in illumination, making them less reliable for tasks such as object recognition, image

retrieval, and scene understanding. Color invariant descriptors aim to capture the underlying visual information of an image while minimizing the influence of illumination and color variations. By extracting features that remain stable across different lighting conditions, these descriptors enhance the discriminative power and robustness of image analysis algorithms. They provide a more consistent representation of objects and scenes, facilitating better matching, retrieval, and recognition performance. To extract color properties resistant to changes in light, several color-invariant descriptors have been developed (Kayhan & Fekri-Ershad, 2021). Two examples are Color SIFT, which extends the SIFT descriptor to encode color information, and Color Moment Invariants, which are statistical measurements generated from the color distribution.

2.3.6 Color Correlogram

The spatial correlation between color values at various distances within an image is measured by a color correlogram. Chaki and Dey (2020) color correlograms can capture color correlations while being less sensitive to variations in illumination by determining the distribution of color co-occurrences. Various approaches can be used to increase the robustness of color correlograms to changes in lighting. Shahbaz et al. (2021) improved CBIR by converting color space to CIELAB to improve the consistency of the representation under different lighting conditions.

2.3.7 Opponent Color Spaces

Opponent color spaces divide the color information into perceptually significant channels, including the opponent color space and the normalized opponent color space. These color spaces can be used to extract illumination-invariant color properties since they are made to be more resilient to changes in lighting conditions. Marasco and Vurity (2022) enhance lighting situations by combining multiple color spaces within CNN.

2.3.8 Color Space Transformation

Color space transformation is a way of reducing the impact of illumination fluctuations by converting the color representation from one color space to another. The transformation refers to the conversion of an image from one color space to another. In image processing, different color spaces are used to represent and manipulate color information in images. Each color space has its own set of coordinates and characteristics that determine how colors are represented and perceived (Burger & Burge, 2022). The purpose of color space transformation is multifaceted and depends on the specific requirements of the image processing task.

RGB Color Space

RGB color space is a widely used color space for image display. It is composed of three color components red, green, and blue. These components are called "additive primaries" since a color in RGB space is produced by adding them together. In the RGB color space, each color appears as a three-dimensional point in a subspace of the standard Cartesian coordinate system (Shamey, 2023). Each axis represents one of the three color components (red, green, and blue) from which all colors in the system will be represented. These values are often normalized for convenience so that all values of red, green, and blue fall within the range [0, 1]. One notable drawback of the RGB color space is that it is not perceptually uniform, meaning that the calculated distance in RGB space does not truly correspond to the perceptual color difference. Figure 2.2 shows the RGB color space representation.

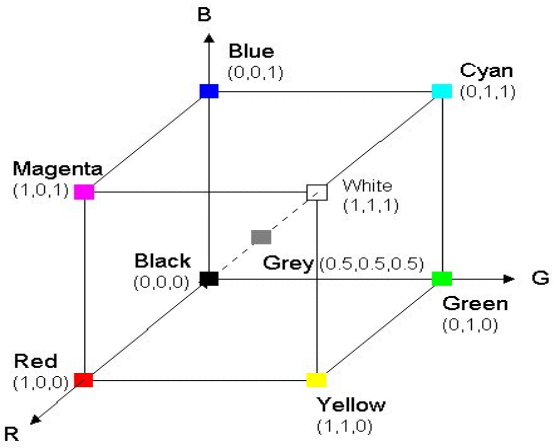

Figure 2.2: RGB Color Space Representation

Among the most frequently employed features in image extraction is color, particularly for separating an image from the intricate natural background. This is because color provides strong, consistent visual features that are less reliant on the image's size.

CMY and CMYK Color Spaces

Like the RGB color model, CMY color space is a subspace of standard three-dimensional Cartesian space, taking the shape of a unit cube. Each axis represents the basic secondary colors cyan, magenta, and yellow. According to Shen et al. (2021) unlike RGB, CMY is a subtractive color model, meaning that where in RGB the origin represents pure black, the origin in CMY represents pure white. In other words, increasing values of the CMY coordinates move toward darker colors whereas increasing values of the RGB coordinates move towards lighter colors. Conversion from RGB to CMY can be done using the simple formula in equation (1).

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (1)$$

Where it has been assumed that all color values have been normalized to the range [0, 1]. This equation reiterates the subtractive nature of the CMY model. Although equal parts of cyan, magenta, and yellow should produce black, it has been found

that in printing applications this leads to muddy results. Thus, in printing applications, a fourth component of true black is added to create the CMYK color model. As with the RGB model, point distances in the CMY space do not truly correspond to perceptual color differences.

HSV and HSL Color Spaces

The HSV (hue, saturation, and value) and HSL (hue, saturation, lightness) color spaces are very different from the previously explored RGB and CMY/K color models in that both systems separate the overall intensity value of a point from its chromaticity (Khalifa et al., 2022). The HSV model, defines a color space in terms of three constituent components: Hue, the color type Range from 0 to 360. Saturation, the "vibrancy" of the color: Ranges from 0 to 100%, and occasionally is called “purity". Value, the brightness of the color: Ranges from 0 to 100%. HSV is cylindrical geometries, with hue, their angular dimension, starting at the red primary at 0°, passing through the green primary at 120° and the blue primary at 240°, and then back to red at 360°.

The HSV space can be visualized in three dimensions as a downward pointing hexacone as illustrated in Figure 2.3 (Chaki & Dey, 2020). The line running down the center of the cone’s vertical axis represents the intensity value V. Hue is represented as the angle relative to the red axis, which resides on the plane perpendicular to the intensity axis. Saturation refers to a point’s perpendicular distance from the intensity axis.

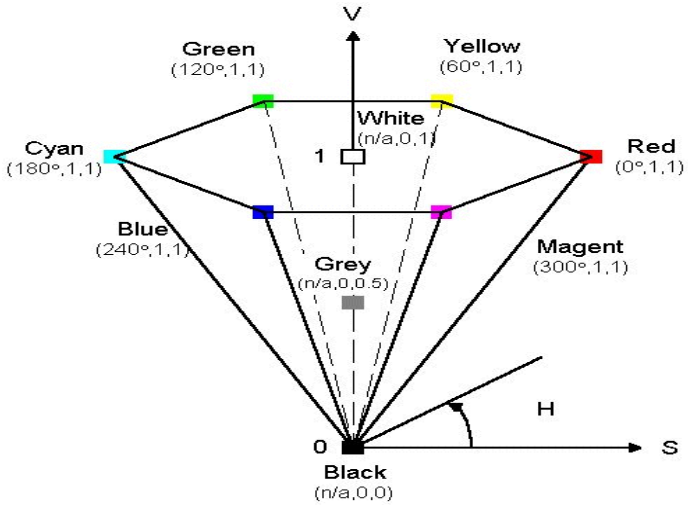

Figure 2.3: Color Hexacone for HSV Representation

The HSL color model is very much similar to the HSV system. A double hexacone, with two apexes at both pure white and pure black rather than just one at pure black, is used to visualize the subspace in three dimensions (Hassan et al., 2023). In HSL, the saturation component always goes from a fully saturated color to the corresponding gray value; whereas in HSV, with V at its maximum, saturation goes from a fully saturated color to white, which may not be considered intuitive to some. Additionally, in HSL the intensity component always spans the entire range from black through the chosen hue to white. In HSV, the intensity component only goes from black to the chosen hue. Because of the separation of chromaticity from intensity in both the HSV and HSL color spaces, it is possible to process images based on intensity only, leaving the original color information untouched. Because of this, HSV and HSL have found widespread use in computer vision research.

The following formula transforms RGB to HSV (Kamiyama & Taguchi, 2021).

$$H = cos^{-1}\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}$$

$$S = 1 - \frac{3}{R+G+B}\big(min(R,G,B)\big) \qquad (2)$$

$$v = \frac{1}{3}(R + G + B)$$

The R, G, B represent red, green, and blue components respectively with values between 0-255. To obtain the value of H from 0^0 to 360^0, the value of S and V from 0 to 1, we execute the following formula H = (H/255*360) mod 360, V = V/255 and S = S/255.

CIE L*a*b* Color Space

The CIELAB color space (also known as CIE *L*a*b** or sometimes abbreviated as simply "*Lab*" color space) is a color space defined by the International Commission on Illumination (CIE) in 1976. Al-Saleem et al. (2020) express color as three values: *L** for the lightness from black (0) to white (100), *a** from green (−) to red (+), and *b** from blue (−) to yellow (+). CIELAB was designed so that the same amount of numerical change in these values corresponds to roughly the same amount of visually perceived change. Because three parameters are measured, the space itself is a three-dimensional real number space, which allows for infinitely many possible colors. In practice, the space is usually mapped onto a three-dimensional integer space for digital representation, and thus the *L**, *a**, and *b** values are usually absolute, with a pre-defined range. The lightness value, *L**, represents the darkest black at *L** = 0, and the brightest white at *L** = 100. The color channels, *a** and *b**, represent true neutral gray values at *a** = 0 and *b** = 0. The* axis represents the green–red component, with green in the negative direction and red in the positive direction. The *b** axis represents the blue–yellow component, with blue in the negative direction and yellow in the positive direction.

The CIE *L*a*b** and CIE *L*u*v** spaces are device independent and considered to be perceptually uniform. They consist of a luminance or lightness component (L) and two chromatic components *a* and *b* or *u* and *v*. CIE *L*a*b** is designed to deal with subtractive colorant mixtures, while CIE *L*u*v** is designed to deal with additive colorant mixtures as demonstrated in Figure 2.4 (Sueeprasan, 2023).

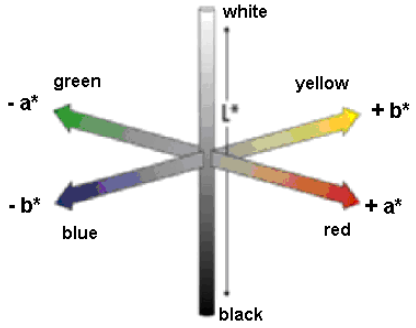

Figure 2.4: CIE LAB Color Space Model

Color Space Properties

The desirable characteristics suitable for color space for image retrieval are the following.

1. Uniformity: The metric proximity between colors indicates the perceived similarity of colors.

2. Completeness: The color space includes all perceptually distinct colors.

3. Compactness: Each color in the color space is perceptually distinct from the other colors.

4. Naturalness: The color space provides for a natural breakdown of colors into the three basic perceptual attributes of color: hue, brightness, and saturation.

According to Tilley (2020), the most important color space property is its uniformity. Uniformity means that two color pairs that are equal in similarity distance in color space are perceived as equal by viewers. In other words, the measured proximity among the colors must be directly related to the psychological similarity among them.

Several models that use RGB, HSV, HSL, and CIELAB color space have been extensively used. CBIR method based on psychophysical and neurobiological characteristics to simulate human visual systems was proposed (Anand et al., 2023). The author showed that human visual characteristics could be presented effectively

and more accurately in the HSV color space than in the *L*a*b** color space. The results of the experiment indicated the mean average precision as 55.89% and 56.57% for Lab and HSV respectively. Hassan and Dhimish (2023), on the other hand, used a straightforward CNN model comprised of two convolutional layers preceded by a max-pooling layer. Then came a dropout layer, two more convolutional layers, one more max pooling layer, one more dropout layer, and finally a dense layer. The model investigated the significance of color spaces for classifying an image. It was evaluated using different databases including CIFAR-10 and CIFAR-100 and produced different performances. Basak et al. (2022) used RGB, HSV, and HSL for sorting strawberries. Nain et al. (2024) used four color spaces, i.e., RGB, HSV, HSL and *L * a * b **, and support vector machine (SVM) algorithms to evaluate the level of ripeness of fruits, and based on the experimental results, it was shown that HSV color feature achieved the best accuracy levels in determining the ripening stages of fruits. Prabhu and Lakshmi (2021) proposed a Mature Tomato Fruit Detection Algorithm based on improved HSV and Watershed Algorithm. Improved HSV transform was used to remove background and detect only red tomatoes. Tamatjita and Sihite (2022) proposed a mask method by use of HSV color space to separate the leaf from the background, where the original image is initially passed to the HSV color space. Chang and Mukai (2022) used HSV color space to extract ripe banana features. Yee et al. (2023) proposed the use of CIELAB to develop an approach to automatically extract dominant colors based on color features that are typically considered by human observers when analysing color schemes.

To reversibly achieve flexible functions that cause hue distortion to the images (Sugimoto & Imaizumi, 2023) proposed an image processing technique for color images. The suggested method reversibly produces brightness increases or decreases, sharpening or smoothing, and contrast enhancement in addition to saturation improvement. An extended approach by Liu and Yang (2021) was also developed to enhance color brightness performance. Their approach precisely specifies the direction to shift the image histogram in each HS sequence to keep the brightness mean constant. The technique maintains brightness while avoiding an overly pronounced contrast increase. Soni et al. (2024) reported combining the hue saturation intensity (HIS) and hue saturation value (HSV) color models ensuring the

preservation of the hue component (H) in both and the suggested outcomes performed better. Hassan and Saud (2023) in their study reported the effectiveness of the HSV color model and logical filter approach in accurately identifying skin pixels in images with a precision of 99.14% and an accuracy of 99.59%. Zhao et al. (2023) used multi-channel information to correct the image's fluctuating illumination in order to get around the issue of illumination changes. To eliminate the intensity variation in the image, they employed the Red, Green, and Blue channels of the RGB color system and the Intensity channel of the HSI color space. The findings of the experiment indicate that the brightness invariant color transformation that has been developed can be used successfully in the retrieval task. To get maximum results of adequate lighting or bright lighting the converted RGB image into a color space with segregated luminance and chromatic components such that the skin color identification procedure only has to use chromatic (Jia et al., 2021).

By transforming the color representation from one color space to another, color space transformation is shown to be a useful technique for mitigating the effects of variations in illumination. The impacts of illumination variations have been reduced using RGB, HSV, and CIELAB color spaces however, the effect of illumination invariance is still a challenge that requires to be addressed.

2.4 Texture Image Extracting Techniques

The texture of an image can be thought of as the spatial variations in pixel brightness intensity in the field of image processing. Some authors proposed to define texture as a measure of coarseness, contrast, directionality, line-likeness, regularity, and roughness (Tamura et al., 1978). The texture can also be seen as a similarity grouping in an image (Kaplan et al., 2020) or as natural scenes containing semi-repetitive arrangements of pixels (Gautam & Singhai, 2024). There are two different types of texture: tactile and optical. We can perceive tactile texture by touching or looking at a surface. When we discuss the shape and content of the image, we are referring to the optical or visual texture. Images that have a particular pattern of texture distribution repeated consecutively throughout them are referred to as textural images in image processing (Sato, 2021). Texture feature extraction in image

processing refers to the process of identifying and representing patterns and structures that characterize the appearance of textures within an image. Texture features capture local variations in pixel intensities, such as fine grain, roughness, or regular patterns that are present in different regions of an image (Jha et al., 2024). The purpose of texture feature extraction is to transform the raw pixel information into meaningful and compact representations that can be used for further analysis and understanding of images. By extracting texture features, an image can be described in a more informative and discriminative manner, allowing algorithms to distinguish between different textures and recognize complex patterns. According to Ma et al. (2023), the commonly used methods for texture feature description are statistical, model-based, and transform-based methods.

2.4.1 Texture Feature Representation

Texture feature representation in image classification involves analyzing and quantifying the surface patterns or structural variations within an image to distinguish between different classes of objects or scenes.

Classification of Texture Feature Extraction Approaches

Texture feature extraction is classified into three major functions namely statistical, model, and filter-based approach. Figure 2.5 shows three major classes of texture feature extraction.

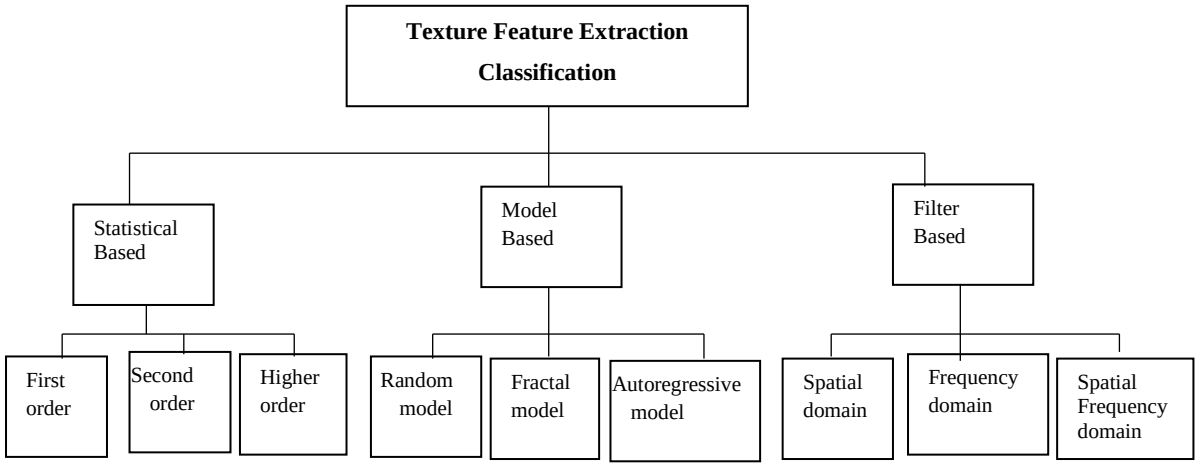

Figure 2.5: Classification of Texture Feature Extraction

2.4.2 Statistical Approach

A statistical approach to texture feature representation in image processing involves analyzing and quantifying the statistical properties of texture patterns within an image or a local region. Instead of directly using raw pixel intensities, statistical texture features provide a more compact and informative representation of textures, capturing characteristics such as homogeneity, contrast, and texture direction (Ramola et al., 2020). Statistics like mean, variance, co-occurrence matrices, or texture energy metrics like entropy are examples of these measurements. These features enable the quantification and separation of textures by capturing the statistical distribution of pixel intensities or texture patterns. These features are widely used in various image analysis tasks, including texture classification, segmentation, and object recognition. The statistical methods can be classified into first order (one pixel), second-order (pair of pixels), and higher-order (three or more pixels) statistics. The first-order statistics estimate properties (e.g., average and variance) of individual pixel values by waiving the spatial interaction between image pixels. The second order and higher-order statistics estimate properties of two or more pixel values occurring at specific locations relative to each other. The most popular second order statistical features for texture analysis are derived from the co-occurrence matrix.

2.4.3 A Model-Based Approach

A model-based approach to texture feature representation in image processing involves using mathematical or statistical models to characterize the underlying structure and statistical properties of textures in an image. According to Shariaty et al. (2022) instead of directly computing local pixel-based features, the methods seek to describe textures using a set of parameters that represent the texture's characteristics and properties. These parameters can then be used to represent and classify textures compactly and efficiently. Model-based approaches are particularly useful when dealing with complex and homogeneous textures that are difficult to capture with simple pixel-based methods.

2.4.4 Transform-Based Approach

A transform-based approach to texture feature representation in image processing involves transforming the pixel intensity values of an image into a different domain to extract meaningful texture features. These transforms can be applied to localized regions or the entire image to analyze the frequency, orientation, or other characteristics of textures. Transform-based methods offer powerful tools for capturing texture patterns at different scales and orientations, allowing for a comprehensive representation of textures. This approach includes three domain techniques namely spatial, frequency, and spatial-frequency domain techniques (Wu et al., 2023).

Spatial domain: The spatial domain has a direct impact on image processing applications. Images are presented as human eyes realize them in this application (Klaib et al., 2021). Spatial domain filters can be classified as the Statistical approach due to a similar grouping in Operation-based methods. Texture features are derived from filtered texture images, including edge frequency, randomness, directivity, linearity, coarseness, and size (Tang et al., 2023).

Frequency domain: The frequency domain is used due to these two reasons, Kernels in the spatial domain are not straightforward, and re-transformation occurs by

multiplying images with the filter into the spatial domain. One of the popular frequency domain methods is Fourier Transforms.

Spatial-frequency domain: There are some disadvantages of Fourier analysis for non-stationary, aperiodic signals (Varanis et al., 2021). Spatial-frequency domain techniques localize image texture simultaneously in spatial and frequency domains. There are spatial-frequency domain techniques for texture feature extraction in image registration such as Wavelet Transforms and Gabor Filters (Keyvanpour et al., 2021).

2.4.5 Structural Approach

Structural texture analysis techniques describe a texture as the composition of well-defined texture elements such as regularly spaced parallel lines. Structural methods define the structural features of an object or image (Chaki & Dey, 2020). Structural features based on topological and geometric properties. Unlike color or texture features, which capture surface details and variations, structural features focus on the shape, layout, and connectivity of elements within an image or between objects. The advantage of the structural method-based feature extraction is that it provides a good symbolic description of the image; however, this feature is more useful for image synthesis than analysis tasks. This method is not appropriate for natural textures because of the variability of micro-texture and macro-texture.

2.4.6 Characteristics of Texture Features

According to Tamura et al. (1978), the author characterized texture features into six features of texture (coarseness, contrast, directionality, line-likeness, regularity, and roughness). Because Tamura texture attributes were developed through research into the spatial and frequency components of textures, they offer a more precise and discriminating representation of textures. Tamura features capture significant visual qualities that are crucial for comprehending and interpreting texture. Tamura features texture approaches by designing texture features that fit the human visual perception. Characteristics of texture features according to Tamura is shown in Figure 2.6 (Tamura et al., 1978).

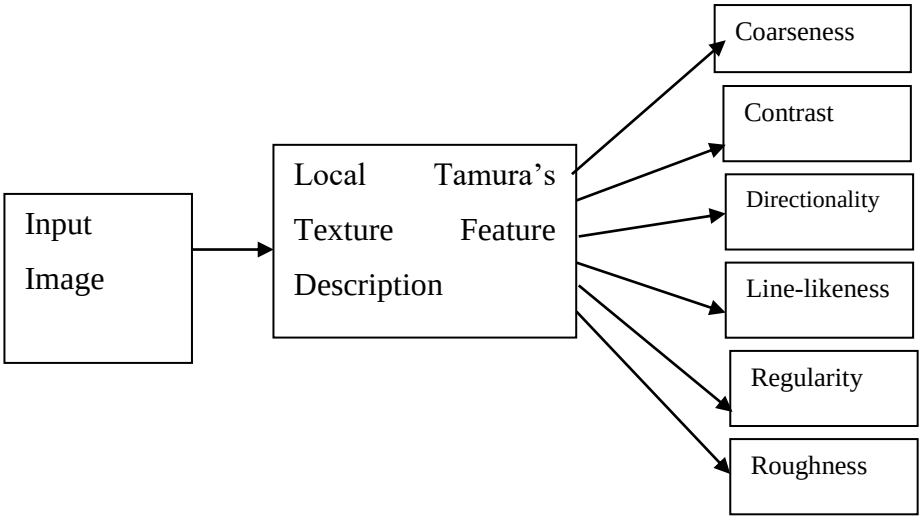

Figure 2.6: Characteristics of Texture Features

Coarseness: The term "coarseness" in image processing describes the estimated or perceived granularity or roughness of a texture in an image. It describes how major features are arranged and sized within the texture.

$$F_{crs} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} S_{best}(i,j) \qquad (3)$$

Contrast: In image processing, contrast of texture refers to the variation in intensity between various areas or features within a texture. It describes how clear or well-defined the texture's patterns are, usually by comparing changes in pixel intensity.

$$F_{con} = \frac{\sigma}{(\alpha_4)^n} \qquad (4)$$

Directionality: In image processing, the term "directionality of texture" describes how components within a texture are oriented or aligned in a preferred way. It describes the prevailing direction of alignment of structures or patterns.

$$F_{dir} = 1 - rn_p \cdot \sum_{p}^{n_p} \sum_{\phi} (\phi - \phi_p)^2 H_D(\phi) \qquad (5)$$

Line-likeness: The degree to which the texture patterns in an image resemble lines or linear structures is referred to as the texture's line-likeness. It is a property that

characterizes the direction and existence of linear units in a texture in texture analysis.

$$F_{lin} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} P_{D,d}(i,j)\cos[(i-j)(2\pi/n)]}{\sum i} = 1^{m}\sum_{j=1}^{n} P_{D,d}(i,j) \quad (6)$$

Regularity: The degree of consistency, predictability, or periodicity of texture patterns within an image is referred to as regularity of texture in image processing. It describes the texture elements' appearance of orderliness or repetition, which is significant for a variety of image analysis tasks.

$$F_{reg} = 1 - r(\sigma_{crs} + \sigma_{con} + \sigma_{dir} + \sigma_{lin}) \quad (7)$$

Roughness: In image processing, the term "roughness of texture" describes how uneven, grainy, or irregular the texture patterns seem in a picture. It explains the texture's degree of richness, diversity, and detail.

$$F_{rgh} = F_{crs} + F_{con} \quad (8)$$

2.4.7 Texture Feature Extraction Techniques

Texture feature extraction techniques involve analyzing and quantifying the surface patterns, spatial arrangements, and frequency characteristics of textures within an image. Texture features capture essential information about the appearance of objects, which helps distinguish them based on their physical surface qualities, like smoothness, roughness, regularity, or randomness. It is widely assumed that a good feature extractor must retain discriminant image components while decreasing intraclass variability (Anand et al., 2023).

GLCM: Gray Level Co-occurrence Matrix

A popular statistical technique for representing texture features is called GLCM. It captures the frequency of certain pixel intensity combinations occurring at specific spatial relationships within an image. These texture features can be relatively robust to rotations, as they describe the general spatial patterns rather than exact pixel

alignments. By nature, GLCM is directionally sensitive to pixel pair comparisons. This indicates that various GLCM values for the same texture pattern result from changing the spatial relationships between pixel pairs when the image is rotated. The generated characteristics may vary significantly as a result of this sensitivity, which could compromise consistency over different rotations (Iqbal et al., 2021).

It calculates the frequency of pixel pairings inside an image or a local area occurring at a given distance and direction with a particular intensity value (Aggarwal, 2022). Many texture properties, including contrast, homogeneity, energy, and correlation, can be obtained from the GLCM.

To calculate the GLCM, the image is first divided into small non-overlapping regions (or pixels) called the reference pixels. Then, the frequency of occurrence of pairs of pixel intensities with specific displacements and orientations is computed within each region (Yang et al., 2023). The result is a symmetric matrix representing the co-occurrence probabilities between pairs of gray levels as demonstrated in pseudo code 1.

Pseudo code 1:

The mathematical formula to compute the GLCM for a given image I with N gray levels (0 to N-1) for a specific displacement vector (dx, dy) is as follows:

1. *Initialize an N x N matrix GLCM, where GLCM(i, j) represents the number of times gray level i co-occurs with gray level j within the specified displacement vector (dx, dy).*
2. *For each pixel in the image (x, y), compute the gray level values I(x, y) and I(x + dx, y + dy) of the pixel and its neighbor displaced by (dx, dy).*
3. *Increment the corresponding GLCM entry: GLCM(I(x, y), I(x + dx, y + dy)) = GLCM(I(x, y), I(x + dx, y + dy)) + 1.*
4. *Repeat steps 2-3 for all pixels in the image.*
5. *Optionally, the GLCM can be normalized to obtain the probability of co-occurrence for each gray-level pair. Divide each entry in the GLCM by the total number of pixel pairs used to construct it.*

After calculating the GLCM, several texture features can be derived from it, such as contrast, energy, homogeneity, correlation, and entropy, among others. These features capture different aspects of texture information and are computed using various mathematical formulas based on the elements of the GLCM.

For example, to compute the contrast feature, the following formula can be used:

$$\text{Contrast} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-j)^2 . GLCM(i,j) \quad (9)$$

Similarly, the energy also known as uniformity or angular second moment is calculated as

$$\text{Energy} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (GLCM(i,j))^2 \quad (10)$$

This directional sensitivity, increases computational cost for multi-directional analysis, limited rotational invariance, and variability in texture features (Barburiceanu et al., 2021).

Histogram of Intensity Values

A simple statistical approach involves computing a histogram of pixel intensities within a local region or the entire image. According to Najjar et al. (2022) histogram provides information about the distribution of pixel intensities, which can be useful in capturing texture characteristics, such as the frequency of occurrence of different intensity levels. Intensity histograms completely disregard spatial information. This means that any spatial patterns or structures in the image are not captured. As a result, two very different images can have identical histograms if their pixel intensity distributions are similar, which limits the histogram's discriminative power for tasks that require spatial context. Combining histograms with other features or using advanced methods helps mitigate these drawbacks as described in pseudo code 2.

Pseudo code 2:

The mathematical formula for calculating the Histogram of Intensity Values for a given image I is as follows:

1. *Define the number of bins B for the histogram. Each bin will correspond to a range of intensity values. For grayscale images, the number of bins is typically set to the number of unique gray levels (0 to N-1), where N is the maximum intensity value in the image.*
2. *Initialize an array or vector H of length B to store the histogram values.*
3. *For each pixel in the image (x, y), compute the intensity value I(x, y).*
4. *Calculate the corresponding bin index b for the intensity value I(x, y) using the following formula:*

$$b = floor\left(\frac{I(x,y)}{N} \times B\right)$$

The floor (.) function rounds the value down to the nearest integer.

5. *Increment the count in the histogram bin corresponding to the index b:*

$$H\,[b] = H\,[b] + 1$$

6. *Repeat steps 3-5 for all pixels in the image.*

After calculating the Histogram of Intensity Values, various texture features can be derived from it. Some common texture features include mean, variance, skewness, kurtosis, and higher-order statistical moments. These features provide information about the distribution and statistical properties of pixel intensities within the region of interest, which can be used to characterize the texture. For example, to compute the mean texture feature the following formula is used (Zhang et al., 2021).

$$\text{Mean} = \frac{1}{N}\sum_{i=0}^{B-1} H[i]\ .i \qquad (11)$$

Where N is the total number of pixels within the region of interest, and H[i] is the count in the i-th bin of the Histogram of Intensity Values. The implementation of the combined pseudo code is quite a challenge.

Local Binary Patterns (LBP)

LBP is a texture descriptor that encodes the relationship between the central pixel and its neighboring pixels by comparing their intensity values. The histogram of LBP patterns within a local region is used as the texture feature representation, providing information about the texture patterns' spatial distribution. Mustaqim et al. (2022) compare the effect of various wavelet transform methods and local binary pattern (LBP) on facial recognition as additional data on the CNN architecture and the pre-trained VGG16 model on the Yale-B facial recognition dataset to extract texture features. The binary pattern is computed using a defined neighborhood (such as 3x3 pixels) in standard LBP. The relationship between the center pixel and its neighbors varies when the image or texture pattern rotates, leading to different LBP codes for the same texture. Rotation invariance is lost in this fixed structure because it fails to take rotations into account (Asma & Brahim, 2022). According to Rasool and Kaur (2021) the binary patterns generated by LBP are highly sensitive to the arrangement of pixels. Even small rotations can significantly alter the LBP code. For instance, a 45-degree rotation can result in a completely different binary pattern, making it difficult to recognize the same texture under different orientations.

Kaur (2023) employs the Rotation-Invariant LBP (RI-LBP) technique, which entails rotating the binary pattern circularly to reach a minimum value. The LBP descriptor becomes rotation-invariant by selecting the shortest binary value after taking into account all possible pattern rotations. It leads to more computing complexity even though it works well.

Fourier Transform (FT)

The Fourier Transform is a widely used transform that converts an image from the spatial domain to the frequency domain. By analyzing the amplitudes and phases of the frequency components, texture features related to periodic structures and

variations can be extracted (Fan et al., 2024). The Fourier Transform is an important image-processing tool that is used to decompose an image into its sine and cosine components. The output of the transformation represents the image in the Fourier or frequency domain, while the input image is the spatial domain equivalent. In the Fourier domain image, each point represents a particular frequency contained in the spatial domain image.

When an image is rotated, its Fourier transform also rotates. This means the complex coefficients of the Fourier transform change, making direct comparisons between the original and rotated images difficult (Hao et al., 2020). According to Jemal et al. (2024) to achieve rotation invariance, the Fourier-Mellin transform combines the Fourier and Mellin transforms. This method helps to separate the effects of rotation by converting the image into log-polar coordinates and then using the Fourier transform. This improves the comparison of rotated images as illustrated in pseudo code 3.

Pseudo code 3:

The mathematical formula for extracting texture features using the Fourier Transform can be summarized as follows

1. *Image Preprocessing: Convert the input image I(x, y) from the spatial domain to the frequency domain using the 2D Fourier Transform:*

$$F(u,v) = F\{I(x,y)\} = \iint_{-\infty}^{\infty} I(x,y)\ .\ e^{-2\pi i\,(ux+vy)}\,dx\,dy$$

   Here, F(u, v) represents the complex-valued Fourier Transform of the image, and (u, v) are the spatial frequency coordinates in the frequency domain.

2. *Compute the Power Spectrum: The power spectrum P(u, v) represents the energy distribution of the image in the frequency domain and is obtained by calculating the squared magnitude of the Fourier Transform:*
3. $P(u,v) = |F(u,v)|^2 = Re\big(F(u,v)\big)^2 + Im\big(F(u,v)\big)^2$

Here, Re(F(u, v)) and Im(F(u, v)) are the real and imaginary components of the Fourier Transform at frequency (u, v).

4. *Extract Texture Features: Texture features can be extracted from the power spectrum using various methods, such as statistical analysis or other frequency-related measurements. Some commonly used features include mean, variance, energy, entropy, and the dominant frequency.*

Since the pseudo-code does not specify the locations of these frequencies in the image, it becomes difficult to comprehend the directional information from the frequency domain and find related images (Veerasingam et al., 2021).

Wavelet Transform (WT)

Wavelet Transform is a multi-resolution transform that decomposes an image into different frequency bands and orientations. This allows for the analysis of textures at various scales, making it useful for capturing both fine and coarse texture details. Arfaoui et al. (2021) a wavelet is a mathematical function used to divide a given function or continuous-time signal into different scale components. Usually, one can assign a frequency range to each scale component. Each scale component can then be studied with a resolution that matches its scale.

The main objective of wavelet transform is to define the powerful wavelet basis functions and find efficient methods for their computation. Fourier methods are not always good tools to recapture the signal or image, particularly if it is highly non-smooth. The wavelet transform is done similarly to Short Term Fourier Transform (STFT) analysis. The signal to be analysed is multiplied with a wavelet function just as it is multiplied with a window function in STFT, and then the transform is computed for each segment generated. Wavelet transforms have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes and for accurately deconstructing and reconstructing finite, non-periodic, and/or non-stationary signals (Akujuobi, 2022).

A wavelet transform is the representation of a function by wavelets. The wavelets are scaled and translated copies (known as "daughter wavelets") of a finite-length or fast-decaying oscillating waveform (known as the "mother wavelet") (Ahmed et al., 2023).

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right), a, b, \in R, a \neq 0 \qquad (12)$$

Where ψ is a wavelet function, a, is a scaling parameter that measures the degree of compression or scale, and b, is a translation parameter that determines the time location of the wavelet.

There have been several texture classifications using transform domain features in the past, such as discrete Fourier transforms, discrete wavelet transforms, and Gabor wavelets. Transform methods analyze the frequency content of the image to determine texture features. Fourier analysis consists of breaking up a signal into sine waves of various frequencies. On the other hand, wavelet analysis breaks up a signal into shifted and scaled versions of the original wavelet (mother wavelet), which refers to the decomposition of a signal into a family of basic functions obtained through the translation and dilation of a special function. The wavelet-based methods, e.g., standard wavelet and Gabor wavelet are the most commonly used techniques to extract the texture vectors as they provide better spatial information (Zitouni et al., 2021). Tabassum et al. (2021) used the Discrete Wavelet Transform (DWT) to obtain the (HH) frequency sub band and then applied the Gabor filter bank at different scales and orientations. Entropy and uniformity were then calculated and stored; this method gives better and more accurate classification results than using any of the DWT or the Gabor filter alone for extracting the features. The Wavelet Transform can be computed using pseudo code 4.

Pseudo code 4:

The general outline of how to extract texture features using the Wavelet Transform is as follows.

1. *Convert the input image I(x, y) from the spatial domain to the wavelet domain using the 2D Wavelet Transform:*

$$W(a,b) = \iint I(x,y) \cdot \psi^*_{a,b}(x,y)dx\ dy$$

Where $W(a,b)$ *is the wavelet coefficient at scale* a *and position* b*, and* $\psi^*_{a,b}(x,y)$ *is the complex conjugate of the wavelet function* $\psi_{a,b}(x,y)$ *at scale* a *and position* b*. The Wavelet Transform produces a set of coefficients at different scales and positions.*

2. *Analyze the wavelet coefficients to extract texture features where several methods can be used such as Wavelet Coefficient Histograms, Energy Distribution, Wavelet Coefficient Statistics.*
3. *Combine the extracted features from different subbands or scales to create a comprehensive texture feature representation.*

The particular texture analysis task at hand as well as the desired balance between fine and coarse texture details will determine the wavelet function to use as well as the decomposition level (number of scales). According to Guo et al. (2022), wavelet filters are designed to capture features in specific directions. Therefore, when an image is rotated, these directional features might not align with the predefined orientations of the wavelet filters, leading to inconsistent feature extraction.

Gabor Transform

The Gabor transform is a mathematical operation that uses Gabor filters to analyze the spatial frequency content of an image. It is a representation of an image in the frequency domain, where the local frequencies and orientations are captured using a set of Gabor filters. The Gabor Transform combines the properties of both the

Fourier Transform and the Gaussian function (Yang et al., 2020). It is particularly effective in analyzing textures with varying frequencies and orientations, as it adapts to different scales and directions.

A Gabor filter is a linear filter that can be defined as a harmonic/sinusoidal function multiplied by a Gaussian function (Dakshayani et al., 2022). These filters can be considered edge, line, and shape detectors. An important characteristic of Gabor filters is that they can be tuned with different frequencies and orientations. The seminal work on Gabor's elementary functions was conducted in 1946 by Dennis Gabor. Gabor based his work on the mechanical wave theory and Heisenberg's uncertainty principle. He proposed the representation of signals as a combination of these elementary functions. Subsequent work has analyzed the specific features obtained using Gabor filters, global Gabor features, and fundamental frequency Gabor features (Zhao et al., 2021).

One of the main advantages of 2-D Gabor filters is their association with a particular location in space. Daugman (2023) has shown that Gabor feature space representation minimizes the joint 2-D uncertainty principle in space and frequency. Moreover, evidence has been presented that the 2D receptive-field profiles of simple cells in the mammalian visual cortex are well described by members of the Gabor 2-D filter family. The advantage of using Gabor filters appears to stand in the ability of such filters to provide some degree of invariance to intensity, translation, and orientation. For these reasons, Gabor filters are finding increasing usage in many applications, such as image enhancement, texture classification and segmentation, image recognition, and motion tracking. Pseudo code 5 illustrates the methods for processing the output produced after convolving the Gabor filters with the input image in order to extract texture features using the Gabor transform (Danlami et al., 2020).

Pseudo code 5:

The following are step-by-step process:

1. *Create a bank of Gabor filters with different frequencies* (f) *and orientations* (θ).
2. *Convolve the input grayscale image* $I(x,y)$ *with each Gabor filter* $g(x,y,f,\theta)$ *to obtain the response image* $R(x,y,f,\theta)$ *for each combination of frequency and orientation.*

   $$R\big((x,y,f,\theta)\big) = |I(x,y) \times g(x,y,f,\theta)|$$

   Where $\times$ *represents the convolution operation, and* $|.|$ *the magnitude of the complex result.*

   Compute statistical measures on the filter responses in local neighborhoods. Commonly used statistical features of Mean, Variance, Energy, Entropy, standard deviation, Skewness, Kurtosis

3. *Collect the computed statistical features for all combinations of frequency and orientation to form a feature vector that represents the texture characteristics of the image.*

Transform methods, such as Fourier, Gabor, and wavelet transforms represent an image in space whose coordinate system has an interpretation that is closely related to the characteristics of a texture. Methods based on Fourier transforms have a weakness in spatial localization, so these do not perform well. Gabor filters provide means for better spatial localization, but their usefulness is limited in practice because there is usually no single filter resolution where one can localize a spatial structure in natural textures. These methods involve transforming original images by using filters and calculating the energy of the transformed images. These are based on the process of the whole image which is not good for some applications which are based on one part of the input image.

Muzaffar et al. (2023) use texture information for browsing and retrieval of large image data by use of Gabor wavelet features for texture analysis and provide a comprehensive experimental evaluation. Comparisons with other multiresolution texture features using the Brodatz texture database indicate that the Gabor features provide the best pattern retrieval accuracy.

Bhargava et al. (2020) suggest designing a Gabor filter bank with filter selection incorporated, which results in a smaller filter bank, less computing complexity, and better texture classification performance. The results demonstrated improved performance in texture classification through experiment results on benchmark datasets and a real application. Wang (2023) presents a technique that combines Gabor filter responses with feature extraction techniques to improve texture classification accuracy under different lighting conditions. Lei et al. (2022) propose an illumination-robust texture classification method that utilizes Gabor features along with Gaussian Mixture Models (GMM) to handle variations in lighting conditions. Xue et al. (2020) explore different feature transformations based on Gabor filters to achieve illumination invariance for texture classification tasks. Huang et al. (2023) suggested using quaternion Gabor to extract color texture information while successfully preserving directionality, contrast, and coarseness. Experimental results demonstrate the superiority of the quaternion Gabor method in retaining Tamura texture features compared to traditional Gabor and LBP methods.

2.5 Similarity Matching in CBIR

Image retrieval in image processing refers to the process of searching and retrieving images from a database based on their visual content rather than relying on textual metadata or tags (Tekli, 2022). The main goal of image retrieval is to find images that are visually similar or semantically related to a given query image. This technology allows users to search for images using other images as queries, making it particularly useful for tasks like content-based image search, reverse image search, and image recommendation systems. Users can find images that are comparable to a query image by using image retrieval. When a user has an image but no descriptive tags or keywords, they can utilize this to find images that are comparable to their

own. The system analyzes the visual features of the query image and retrieves images with similar content from the database. Figure 2.7 demonstrates how an image is retrieved from a database.

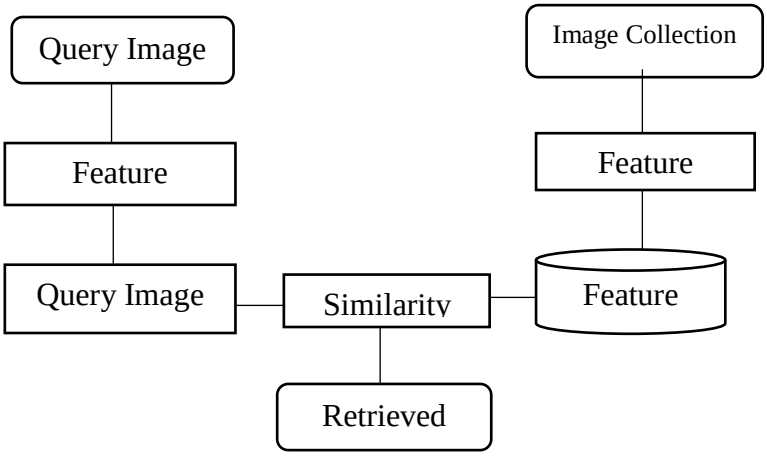

Figure 2.7: Image Retrieval System

One can utilize distance approaches or machine learning techniques to measure similarity to get the correct image following the query.

2.6 Rotational Invariance Techniques

The invariant (I), in image processing, is a characteristic of the image (in this case, a function) that does not change or only slightly changes if the image is transformed (e.g., rotated, scaled, blurred, etc.). Mathematically, a function is said to have rotational invariance (I) if its value does not change when arbitrary rotations are applied to its argument (Ding et al., 2020).

Invariant (I) can be expressed as follows.

$$I(f) \approx I(D\ (f)) \qquad (13)$$

Where f(x,y) is the original image (where x and y is a pixel coordinate of the image and the output of f is the intensity of pixel) and D is the transformer function of the

image (degradation operator). Invariants appear in a variety of forms, and each one has a unique set of transformer functions. We all are familiar with the most fundamental image transformations, such as rotation, scaling, and translation.

In image processing, rotational invariance refers to a feature or property of an image processing technique that maintains its original form or consistency despite rotations of the input image. In other words, a rotationally invariant algorithm or feature will yield the same outcome regardless of the orientation of the image if the image is rotated. In other words, if an image is rotated by a certain angle, a rotationally invariant algorithm would produce the same output as it would for the original image (Zheng et al., 2022). Rotational invariance is desirable in many image processing applications because objects or patterns in images can appear in several orientations, such as object recognition, image classification, or feature extraction. A rotation angle, a rotation point, and an image are the inputs of an image rotation routine. The goal is to achieve the result shown in Figure 2.8 when the image is rotated at any angle.

Figure 2.8: Rotation of Cat Image at 450 Angle

An image rotation model refers to a computational technique used to rotate images by certain angles, typically in computer vision or image processing tasks. Although this model can be applied in a variety of ways and with different methods, the main goal is to change an image's pixels to make it appear to be rotated by a given angle. Figure 2.9 is a Network visualization structure of a rotated image (Dhillon & Verma 2020).

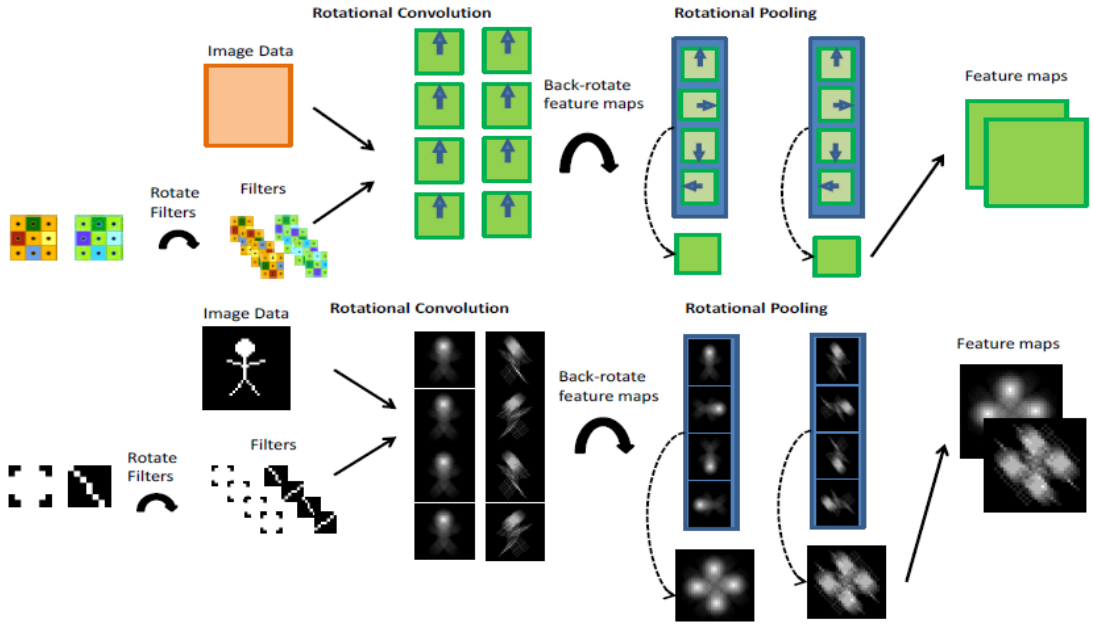

Figure 2.9: Network Visualization Structural of Rotated Image

2.6.1 Techniques for Rotational Invariance in Image Processing

In image processing, achieving rotational invariance means designing techniques that allow a system to recognize objects or patterns regardless of their orientation.

Feature-Based Methods

Extraction of rotationally invariant features from the image is a typical strategy. These features are made to be rotation-independent. According to, Gupta et al. (2021) in object identification tasks, two examples are Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF). Scale-Invariant Feature Transform (SIFT) is an approach for local feature detection and description in computer vision while Speeded-Up Robust Features (SURF) is a patented local feature detector and descriptor.

Scale-Invariant Feature Transform (SIFT)

SIFT has high stability for angle transformation, affine transformation, and noise. SIFT functions in theory like a multi-scale corner detector, with sub-pixel

positioning accuracy and a rotation-invariant feature description affixed to each candidate point as illustrated in pseudo code 6 (Burger & Burge, 2022).

Pseudo code 6:

The following are the primary steps in the calculation of SIFT features:

1. *Extrema detection in a Laplacian-of-Gaussian (LoG) scale space to locate potential interest points.*
2. *Key point refinement by fitting a continuous model to determine precise location and scale.*
3. *Orientation assignment by the dominant orientation of the feature point from the directions of the surrounding image gradients.*
4. *Formation of the feature descriptor by normalizing the local gradient histogram.*

SIFT was used to delete the low contrast extreme value that was used to indicate the image's orientation in the area of the key point, the orientation is canceled and the object is made rotation-invariant. Localization of the input image, namely the size and direction of nearby pixels, is crucial (Selvi & Thilagamani, 2023).

By noting the primary direction of each feature point to ensure rotation invariance, the SIFT operator can handle the rotated images. Utilizing the distance and direction relationship between the feature points and the pixel points in its neighboring range, the direction of the feature points is established to calculate the gradient information. The following formula can be used to get the gradient value of a pixel point (x, y) (Li et al., 2022).

$$m(x,y) = \sqrt{\begin{matrix}(L(x+1,y)-L(x-1,y))^2 \\ (L(x,y+1)-L(x,y-1))^2\end{matrix}}+ \quad (14)$$

Where L represents the spatial scale value

The formula for the calculation of the gradient direction is as follows:

$$\theta(x,y) = tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right) \quad (15)$$

It is necessary to simplify the expression of feature points in the direction after computing the gradient values of numerous pixel points close to the feature point. The major direction of the feature points was defined by the interval value of the biggest statistical value, which was calculated for a total of 36 intervals using 10^0 as the interval. Figure 2.10 shows the main direction-determining process (the figure contains only eight directions for simplification).

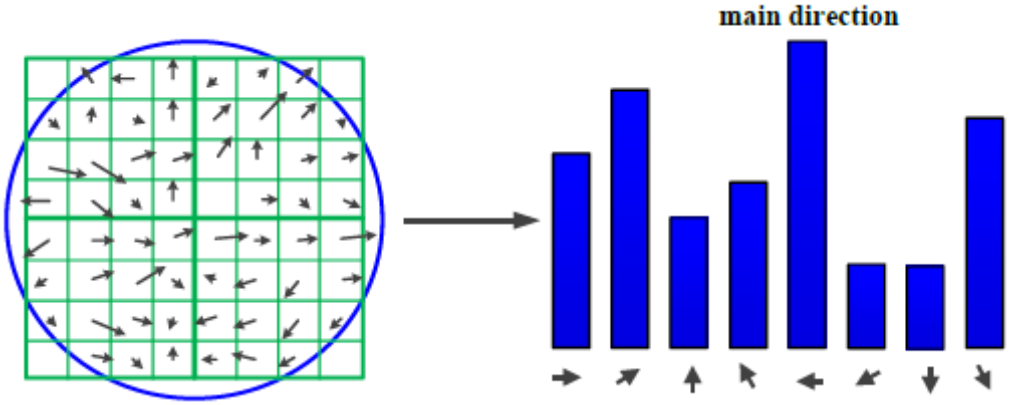

Figure 2.10: Determining the Main Direction

After expressing the direction next is to generate a feature descriptor. The coordinate axis is rotated to maintain the direction of the x-axis while maintaining the rotation invariance of the feature points. 16 small sections, each measuring 4*4, are chosen as the 16*16 sampling area in the feature point neighborhood. The feature points for each pixel in the small region are then described with the feature points' gradient value and direction. Due to an abundance of direction information, the statistics 45° intervals are separated, and each tiny region's histogram distribution is then obtained in eight directions, creating 16 small regions. As an illustration, consider the 8*8 neighborhood, where the production of the feature descriptor is depicted in Figure 2.11.

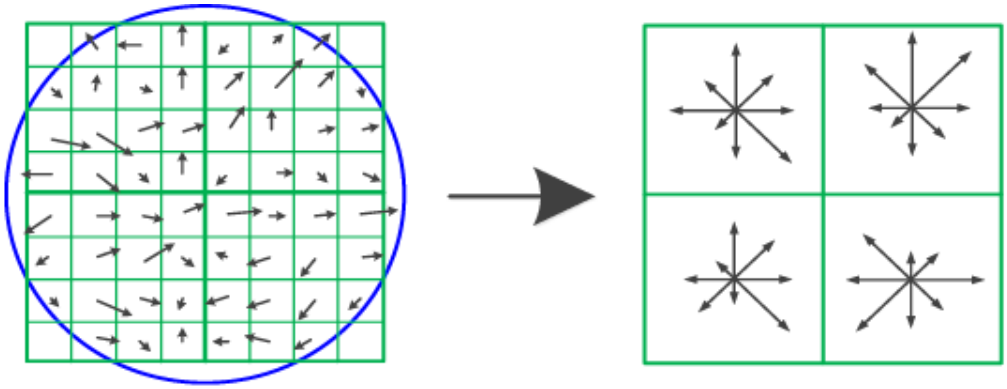

Figure 2.11: The Generation Process of the Feature Descriptor

As the figure shows, the center of the left figure is the feature point. A pixel point is represented by each tiny square, and the direction of the arrow indicates the orientation of the pixel to the feature point. The gradient value is represented by the length of the arrow. The circle's area is a range with a Gaussian weight. The small square on the right shows the eight directions in which statistics are oriented. As a result, the SIFT descriptor is a 128-dimension vector with a total size of 16 * 8 and, after being normalized, can adapt to lighting effects. SIFT, on the other hand, has poor non-rigid transformation stability, high calculation costs, and no capacity to adjust to illumination variance (Hossein-Nejad & Nasri, 2024).

Speeded-Up Robust Features (SURF)

For local, similarity invariant representation and comparison of pictures, the SURF method (Speeded Up Robust Features) is a quick and reliable algorithm. The primary benefit of the SURF technique is the quick computation of operators using box filters, allowing for real-time applications like tracking and object recognition. SURF is known for high calculated speed and high lighting variation adaptability although weak in stability for non-rigid transformations and adaptability to changes in noise and grayscale. SURF is comprised of a feature detector based on a Gaussian second derivative mask, and a feature descriptor that relies on local Haar wavelet responses (Wang et al., 2023). This framework shares many conceptual similarities with the most widely used feature detector in the computer vision community, called the Scale-Invariant Feature Transform (SIFT) (Chen et al., 2021).

Fourier-Based Methods

Utilizing the Fourier Transform to examine the frequency content of an image is an alternative method. Rotational invariance can be attained by studying the frequency components of an image's Fourier Transform, which is rotation-invariant. An illustration of a Fourier-based technique for rotation-invariant shape recognition is the circular harmonic transform (CHT).

Circular Harmonic Transform (CHT)

The Circular Harmonic Transform (CHT) is a mathematical technique used in image processing to analyze and represent rotationally invariant features (Mei et al., 2023). It is a form of complex transform that decomposes an image into a series of components based on its angular or circular characteristics. CHT is particularly useful in applications where rotational invariance is crucial, such as object recognition, texture analysis, and feature extraction in content-based image retrieval (CBIR) systems. One of the primary benefits of CHT is that it naturally provides rotational invariance. This means that the features extracted by CHT remain consistent even if the object or pattern in the image is rotated. This is essential for tasks where the orientation of objects varies, as it allows the system to identify objects independently of their rotation (Andrearczyk et al., 2020). Paoletti et al. (2020) proposed the use of Circular Harmonic Features for rotation-invariant texture classification. The authors extracted circular harmonic coefficients to represent texture patterns, achieving robustness to rotation and scale variations. Almakady et al. (2020) developed circular harmonic filters to efficiently analyze color textures, demonstrating the suitability of CHT for rotation-invariant texture analysis. Anne et al. (2024) incorporated the CHT within the framework of sparse representation for recognizing faces under different illumination and pose conditions.

CHFs are inherently rotation-invariant, and compact representations in the frequency domain, making them suitable for certain texture classification tasks. However, they also come with complexities in parameter tuning, potential loss of spatial information, and variability in performance across different datasets and texture types.

Template Matching

In the process of "template matching," various elements of an image are compared to a reference pattern or template to identify instances of the pattern. The template matching technique can be applied in several orientations to provide rotational invariance, or the template can be represented by rotationally invariant descriptors. Xu et al. (2021) for processing images, template matching is frequently employed. Some of its widely used applications include object positioning, image edge detection, route planning for mobile robots, and image registration methods. Template or area-based approaches and feature-based approaches are the two broad categories for template or image-matching approaches. Feature-based approach extracts key features from the template and the image and then uses methods such as Scale-Invariant Feature Transform (SIFT) or Oriented FAST and Rotated BRIEF (ORB) for matching. These feature-based methods are inherently robust to rotation. Oriented FAST and Rotated BRIEF (ORB) combine the FAST corner detector with the BRIEF descriptor and introduce modifications to achieve rotation invariance. ORB is known for its computational efficiency and good performance in feature-matching tasks (Xie et al., 2022).

Other techniques used by template matching to solve the challenges of rotation invariance are multiple Orientations where multiple templates with different orientations of the object of interest are created. By applying template matching with all the templates, you can find the best match across different orientations. This method works well if the range of possible rotations is limited. Circular Correlation involves circularly shifting the template and the region of interest before performing cross-correlation. This ensures that the correlation operation considers all possible rotations of the template Affine Transformations and others (Wang et al., 2021).

Wu (2023) proposes a multi-template matching approach to handle different orientations of objects in SAR images, improving recognition accuracy. To address deformable object recognition with multiple orientations using hierarchical template matching. Steinmann et al. (2023) propose a cascaded matching strategy to handle different orientations and non-rigid deformations in the target objects. Liu and Yang

(2023) propose a method to generate multiple rotated templates and then perform matching using Contour SIFT features to achieve rotation invariance. Fan et al. (2024) employ a steerable pyramid to generate multiple orientation responses and use them to perform rotation-invariant template matching for texture detection. Yang et al. (2023) divided the templates into multiple circle sectors and performed matching using these sectors to handle various orientations.

Template matching has limitations, including increased computational complexity, sensitivity to template quality, and limited robustness in handling scale and complex object structures. Despite its ability to simplify object detection regardless of orientation, it may not be appropriate for all image-processing tasks. Even though SPP can increase rotational invariance, it can still be difficult to achieve full rotational invariance in complex situations.

Spatial Pyramid Pooling (SPP)

Spatial pyramid pooling is a technique used in computer vision that allows fixed-size feature representations while processing variable-sized input images (Quan et al., 2023). With this method, the image is divided into various scales or areas, and features are computed independently for each scale or region. Rotational invariance can be obtained by mixing features from several scales because they each collect information at a different level of detail. Distinct orientations of an image may have distinct spatial arrangements of information, which makes it challenging for conventional convolutional neural networks (CNNs) to recognize them in multiple orientations when dealing with rotational invariance. SPP can solve this problem, making the model more resistant to object rotations by capturing multi-scale and multi-level spatial information. CNNs can be used with photos of various resolutions because Tan et al. (2021) demonstrated that SPP can handle variable-sized inputs and produce fixed-size feature representations.

Dewi et al. (2023) used SPP to recognize scenes and introduced Deep Spatial Pyramid, which combines SPP with a deep convolutional neural network to gather multi-scale data for scene comprehension. The authors showed how well SPP handled complicated scenarios with various scales and architecture. As a

generalization of SPP, Quach et al. (2024) proposed the idea of Multi-Scale Orderless Pooling (MOP). To improve its ability to handle a wide range of input scales, MOP includes SPP with additional capabilities. The authors showed that by offering higher spatial invariance, MOP can enhance the performance of object detection tasks. Even though SPP can increase rotational invariance, it can still be difficult to achieve full rotational invariance in complex situations. More advanced solutions, such as data augmentation with 3D transformations, employing specialized architecture like capsule networks, or explicitly including other types of rotational equivariance in the network design, may be necessary for extreme rotations or viewpoint shifts.

Convolutional Neural Networks (CNNs)

CNNs can be created to be invariant to specific transformations, including rotation, and have demonstrated considerable effectiveness in image-processing tasks. One common approach is to use pooling layers, such as max pooling, which aggregate information from local regions and are insensitive to small translations and rotations (Kayhan & Gemert, 2020).

In typical CNNs, several modules of convolution and max-pooling layers are stacked on top of each other (Weng & Zhu 2021). They first extract low, then mid, and finally, high-level features of the image before these are fed into a classifier that usually consists of one or several fully connected layers followed by a *softmax*. Convolutions with small filter size and stride (compared to the input feature maps) followed by max-pooling subsequently reduce the feature dimension. At the same time, these feature extraction modules introduce invariance to translations of the objects inside the input image and invariance to smaller distortions and scale changes. Nevertheless, if the object is rotated by medium or large angles, the activations of the filters change in most cases.

CNNs can also learn rotation-invariant features by using methods like data augmentation, which incorporate random rotations during training. According to Achouri and Martin (2021), it is difficult to achieve perfect rotational invariance in all circumstances, and the strategy adopted varies on the particular task and

requirements. Some approaches might increase computational complexity while others might trade off some accuracy. Therefore, the trade-offs and limitations of the application determine the technique to use.

Convolutional neural networks (CNN), have been successfully employed in various computer vision tasks since 2012 and dramatically exceed the majority of manually created image characteristics. Traditional CNN models, on the other hand, are not rotation-invariant, and even slight rotations of an input image can significantly harm their performance. Prior research has demonstrated that the only invariance to image translation that the standard convolutional operation offers is due to local connectivity and spatial parameter sharing (Mumuni & Mumuni, 2021). Utilizing data augmentation to train CNNs is an easy technique to deal with image rotation. Unfortunately, this method wastes computation to learn numerous redundant weights, which not only raises the cost of training time (Li et al., 2021). Many of the first-layer weights can be seen as rotated copies of one another when they are seen in a CNN. Furthermore, figuring out how many rotation versions should be produced for each training image is difficult. Xu et al. (2022) proposed a spatial pyramid pooling network model that generates a fixed-length representation regardless of image size/scale. The model was considered to be one of the team's top models. The choice of structure, however, could either overlook crucial information, resulting in loss of discriminative ability, or capture too much data, producing high-dimensional feature vectors. Paoletti et al. (2020) provide a rotation-equivariant CNN2D model for HSI analysis, with circular harmonic filters (CHFs) in place of conventional convolution kernels. However, the model might not function well on images with intricate or irregular rotational patterns. Performance may suffer if the network's chosen rotational symmetries do not match the rotational transformations in the data well. As a result, the assumptions made on the rotational characteristics of the data have a significant impact on the network's performance.

Currently, the majority of models such as Rotation Invariant Convolutional Neural Network Based on Orientation Pooling and Covariance Pooling (Yao et al., 2022), Gradient-Aligned convolution neural networks (Hao et al., 2022), Rotation invariant Gabor convolutional neural networks for image classification are rotation invariant

since most research has concentrated on rotational changes. However, these models are still having difficulties with shifting lighting conditions (Yao & Song, 2022). The focus of current research has shifted to developing new network architectures to incorporate rotation invariance into conventional CNNs, and several modified models have been designed.

2.6.2 Rotational Invariance Algorithms Incorporated into Conventional CNN

Incorporating rotational invariance algorithms into conventional Convolutional Neural Networks (CNNs) enhances the model's ability to recognize patterns regardless of their orientation. The following are some approaches.

Spatial Transformer Networks (STN)

Spatial Transformer is a new learnable module, which explicitly allows the spatial manipulation of data within the network (Yu et al., 2023). The differentiable module can be added to current convolutional architectures, enabling neural networks to actively modify feature maps' spatial relationships based on the feature map itself without changing the optimization procedure or adding additional training supervision. According to the findings, the model has acquired general warping invariance to translation, scale, and rotation. Figure 2.12 demonstrates how the spatial transformer module works (Xu et al., 2020).

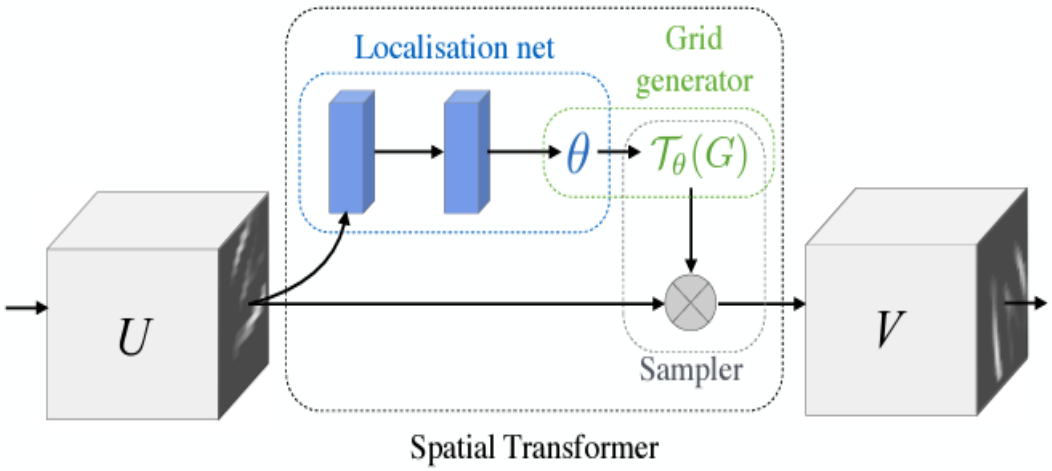

Figure 2.12: The Architecture of a Spatial Transformer Module

A localization network receives the input feature map U and regresses the transformation parameters. The sample grid T(G) is applied to U transforming the normal spatial grid G over V into the warped output feature map V. A spatial transformer is defined by the localization network and sampling mechanism combined. However, the model struggles to capture more complex spatial changes found in some tasks or datasets because it lacks End-to-End Training, which makes the training process more difficult.

Transformation-Invariant Pooling (TI-Pooling)

The model creates transformation-invariant features for convolutional neural networks (Quiroga et al., 2020). For the examined transformation set, parallel siamese architectures are used, and their outputs are subjected to the TI-POOLING operator before the fully connected layers. This architecture limits the duplication in learned features by internally locating the most ideal "canonical" instance of the input image for training. Convolutional neural network features are created using a method influenced by the max pooling operator Gholamalinezhad and Khosravi (2020), and multiple instance learning to be transformation-invariant (Zafar et al., 2022). The original image and its altered counterparts (input instances) are fed into the network along the path of neuron activations, much like augmentation. However, we pool all of the responses and take the maximum number of them rather than considering each occurrence as an independent sample (TI-POOLING operation).

Because of the maximum, the response is unaffected by changes and generates characteristics that are transformation-invariant, which are then spread across the network. Additionally, since it only learns from one instance and that instance already provides the maximum response, this enables more effective data usage. Parallel layers of a Siamese Chicco (2021) network with shared weights and inputs that correspond to various transformations are how this architecture is implemented as described in Figure 2.13.

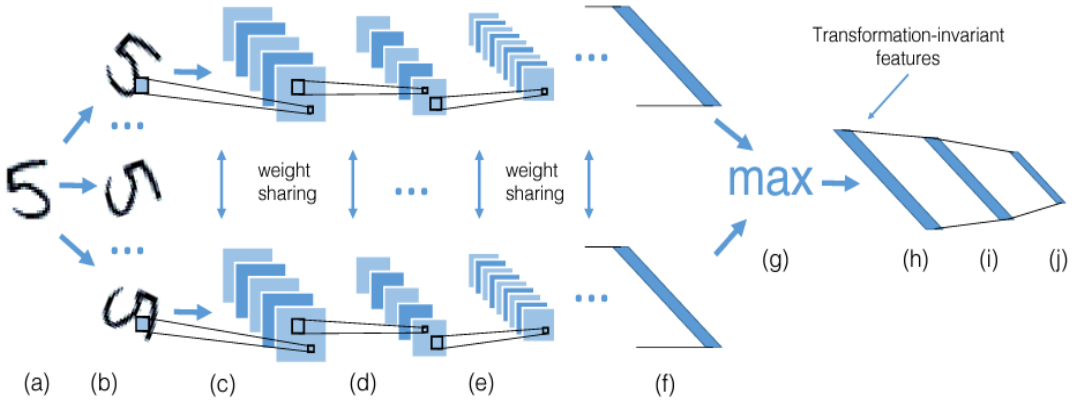

Figure 2.13: Network Topology and Pipeline Description

First, input image x (a) is transformed according to the considered set of transformations Φ to obtain a set of new image instances φ (x), φ ∈ Φ (b). For every transformed image, a parallel instance of a partial Siamese network is initialized, consisting only of convolutional and subsampling layers (two copies are shown at the top and the bottom of the figure). Every instance is then passed through a sequence of convolutional (c, e) and subsampling layers (d), until the vector of scalars is not achieved (e). This vector of scalars is composed of image features f k (φ (x)) learned by the network. Then TI-POOLING (element-wise maximum) (g) is applied to the feature vectors to obtain a vector of transformation-invariant features g k (x) (h). This vector then serves as an input to a fully connected layer (i), possibly with dropout, and further propagates to the network output (j). Because of the weight sharing between parallel Siamese layers, the actual model requires the same amount of memory as just on convolutional neural network. TI-POOLING ensures that the actual training of each feature parameter is performed on the most representative instance φ (x).

The result of the model shows that incorporating the TI-POOLING operator increases the performance over the baselines with a similar number of parameters, and also demonstrates the property of TI-POOLING to find canonical transformations of the input for more efficient data usage. On the other hand, the model usually concentrates on a narrow range of geometric transformations and strongly relies on the presumption that the incoming data mostly shows the required transformations within the designated range (Nirthika et al., 2022).

Group Equivariant Convolutional Networks (G-CNN)

It is a natural generalization of convolutional neural networks that reduces sample complexity by exploiting symmetries (Cohen, 2021). It also employs G-convolutions, a brand-new layer type that offers far more weight sharing than conventional convolution layers. G-convolutions allow the network to express itself more fully without adding more parameters. For discrete groups produced by translations, reflections, and rotations, group convolution layers are simple to create and have little computing overhead. As a result, each vector in the representation space has a pose that may be altered by the components of a certain group of transformations, called G. This extra structure makes it possible for us to model data more effectively (Kawano et al., 2021). Through an operation known as G-convolution, a filter in a G-CNN can match such a feature constellation in every global posture by detecting co-occurrences of features that have the desirable relative position.

A representation space that is related to other representation spaces can get its structure from them. The network or layer Φ that maps one representation to another must be structure-preserving for this to operate (Smets et al., 2023).

$$\Phi\left(T_g x\right) = T'_g \Phi(x), \tag{16}$$

In other words, the same outcome should be obtained by first mapping an input x through and then changing the representation as opposed to first transforming an input x by a transformation g (forming $T_g x$) and then passing it through the learned

map. Equivariance to symmetry transformations constrains the network in a way that can help generalization, as well as enhance statistical effectiveness and aid geometrical reasoning. According to Mo and Zhao (2024), the outcome demonstrates that switching from planar to G-convolutions reliably improves outcomes without further adjustment. However, since these three models solely act on data representations, these approaches have the advantage of taking advantage of traditional CNN implementations. However, they have significant drawbacks in that they can only take into account the input images' global transformations. Another drawback is as opposed to random image rotations, the majority of them are simply invariant to specific rotation angles. Additionally, some techniques need additional trainable parameters to accommodate image rotation. In addition, a number of procedures lack intuitiveness since they rely on complicated mathematical theories and notions. Finally, the majority of researchers continue to train their rotation-invariant CNN models using image data augmentation. Unfortunately, the group representations that are selected for use in the convolutions have a significant impact on how well G-CNNs work (Tegin & Duman, 2023). Additionally, determining the best group representations that effectively enable equivariance and capture the key characteristics of the data can be challenging and may need for specialized knowledge or testing.

Mo and Zhao (2022) created Rotation-Invariant Coordinate Convolution (RIC-C) that focuses on convolutional layers with strong rotation invariance using a straightforward rotation-invariant coordinate system. A RIC-CNN model was created by swapping out every normal convolutional layer in a CNN for its matching RIC-C. RIC-C is inherently invariant to arbitrary rotations about the input center without the addition of extra trainable parameters and data augmentation. Additionally, the rotation invariance of the RIC-CNN is achieved without the inclusion of any additional learnable parameters, so it has the same amount of parameters as the original CNN. The experimental findings demonstrate that RIC-C is a simple drop-in substitute for conventional convolutions and significantly improves the rotation invariance of CNN models created for various applications.

According to Gao et al. (2022), the Rotation Equivariant Vector Field Networks (RotEqNet) used several rotation instances that is, rotations that occur at various intervals of a uniform standard filter to accomplish convolution. Even though the RotEqNet model is tiny, adding more convolution kernels results in increased memory usage and longer computation times. Visualization of how images were rotated is illustrated in Figure 2.14.

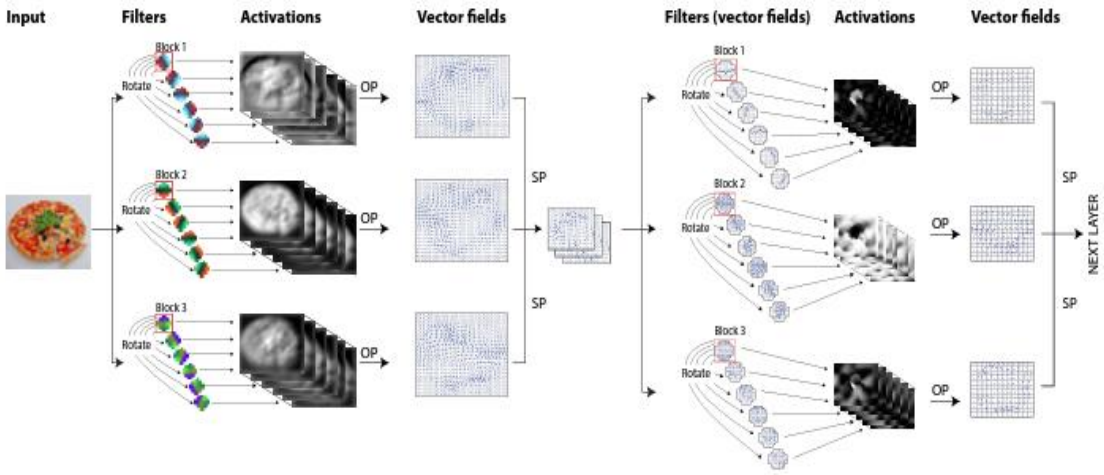

Figure 2.14: An Illustration of RotEqNet's First Two Levels

Each layer replicates the same three canonical filters (shown by red squares) over six orientations. Three vector field maps are produced as the first block's output, and the second block's vector field filters further convolve them. (OP: orientation pooling; SP: spatial pooling). According to Zhung et al. (2022) exploiting cyclic symmetry in convolutional neural networks used parameter sharing to achieve rotation equivariant cyclic symmetry coding in CNNs. Four new operations are introduced: slice, pool, stack, and roll. The procedures construct networks that are equivalent to cyclic rotations and share parameters across various orientations. They can be viewed as layers in a neural network as shown in Figures 2.15 and 2.16. However, the actions (slicing, pooling), amount of feature maps (rolling), or both (stacking) alter the size of the minibatch.

Figure 2.15: The Four Operations that Constitute Framework for Building Rotation Equivariant Neural Networks

Figure 2.16: Schematic Representation of the Effect of the Cyclic Slice, Roll and Pool Operations on the Feature Maps in a CNN

Arrows represent network layers. Each square represents a minibatch of feature maps. The letter 'R' is used to clearly distinguish orientations. Different colors are used to indicate that feature maps are qualitatively different, i.e., they are not rotations of each other. Feature maps in a column are stacked along the batch dimension in practice; feature maps in a row are stacked along the feature dimension.

While some rotations might be easy to handle, achieving robustness to arbitrary rotations (including non-integer degrees) can be difficult and may require extensive tuning and validation. Additionally, designing models that are inherently rotation-invariant is complex and often requires sophisticated mathematical techniques or specialized architectures which involve increased computational costs (Graham et al., 2020). Therefore, current research has shifted to developing new network architectures to incorporate Gabor filters into conventional CNNs to overcome the challenges of rotation variance, and several modified models have been designed.

Recently, Gabor convolutional neural networks have gained popularity due to their potent ability to orient images.

Gabor Convolutional Networks (GCNN)

Other research has demonstrated that Gabor filter features can enhance and complement CNNs (Dhakshayani & Surendiran, 2023). Additionally, to increase accuracy and speed up convergence, researchers modified the architecture by initializing the first layer of CNNs with Gabor filters (Abdullah et al., 2022).

To further employ a small number of convolution kernels with extra prior orientation and scale information for feature extraction, Gabor filters were introduced to modify the grouped separable convolution (GSC) kernels (Zhao et al., 2023). Mehta et al. (2023) suggest using the Gabor representation as an input to an ensemble of deep neural networks that have already undergone training to address the issues of illumination changes, and rotation invariance (which are typically brought on by an unconstrained environment). The availability of several robust and discriminating characteristics created by Gabor filters allows the models to be trained effectively when these Gabor features are used as an input to multiple pre-trained deep neural networks.

Yao and Song (2022) presented a model to learn Gabor-guided deep convolutional features by first twist each input image to create several rotating image features, which are then fed into a weight-sharing Siamese network structure. Then, in order to create a representation of an image that is rotation invariant, computation of the maximum and average feature responses from each instance of the same input image that has been rotated was done. The result demonstrates the efficiency of RIGCN for rotation invariant image classification. By substituting GCLs for traditional convolutional layers, Zhuang et al. (2022) presented transformation-invariant Gabor convolutional networks (TI-GCNs). The recovered Gabor features are fed into a weight-sharing convolution module in their chosen GCL. It is followed by a transformation pooling module to provide some invariance (i.e., taking element-wise maximum). Although it can result in the loss of some discriminative image information and a decrease in the final classification performance, transformation

pooling is frequently used in continuous GCLs. The models listed above primarily emphasize rotational invariance. The study identifies the necessity to address the issue of lighting changes. The response to impulses is defined using a Gaussian function with a sinusoidal wave (Liang et al., 2023).

Complex form of Component

$$g(x.y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x^2+\gamma^2y^2}{2\sigma^2}\right) exp\left(i\left(2\pi\frac{x^1}{\lambda}+\psi\right)\right) \quad (17)$$

Real

$$g(x.y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x^2+\gamma^2y^2}{2\sigma^2}\right) cos\left(i\left(2\pi\frac{x^1}{\lambda}+\psi\right)\right) \quad (18)$$

Imaginary

$$g(x.y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x^2+\gamma^2y^2}{2\sigma^2}\right) sin\left(i\left(2\pi\frac{x^1}{\lambda}+\psi\right)\right) \quad (19)$$

Where

$$x' = x\cos\theta\, y\sin\theta$$

$$y' = -x\sin\theta\, y\cos\theta$$

Here,

λ (Lambda) = wavelength of the sinusoidal factor,

θ (Theta)= orientation of the normal to the parallel stripes of a Gabor function,

ψ = phase offset,

σ = Sigma/ standard deviation of the Gaussian envelope,

γ (Gamma) = spatial aspect ratio

GCNN is a deep convolutionary neural network that uses Gabor-orientation filters (GoFs). A GoF is a manageable filter created using Gabor filter banks to manipulate the learned filters and then generate the improved functionality maps. GCNNs involve fewer learning filters with GoFs thus making them simple to train as well as improving the deep models. Qi et al. (2022) demonstrate how Gabor-orientation filters were generated according to Figure 2.17.

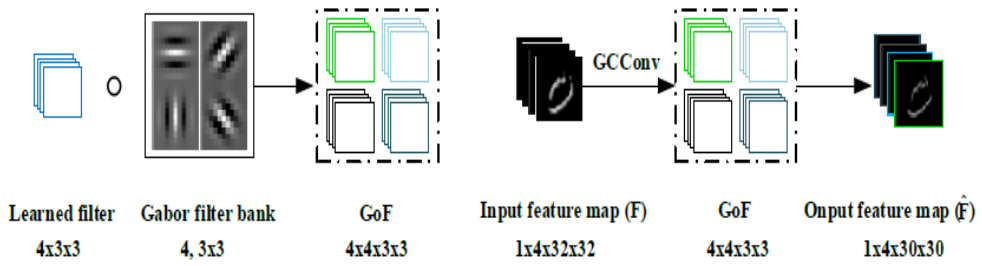

Figure 2.17: Modulation Process of GoFs (left) and GCN Convolution with Four Channels (Right)

GCN convolution

In the Gabor Convolution Network, GoF was used in the production of feature maps that explicitly improved the orientation in deep features as demonstrated in Figure 2.18.

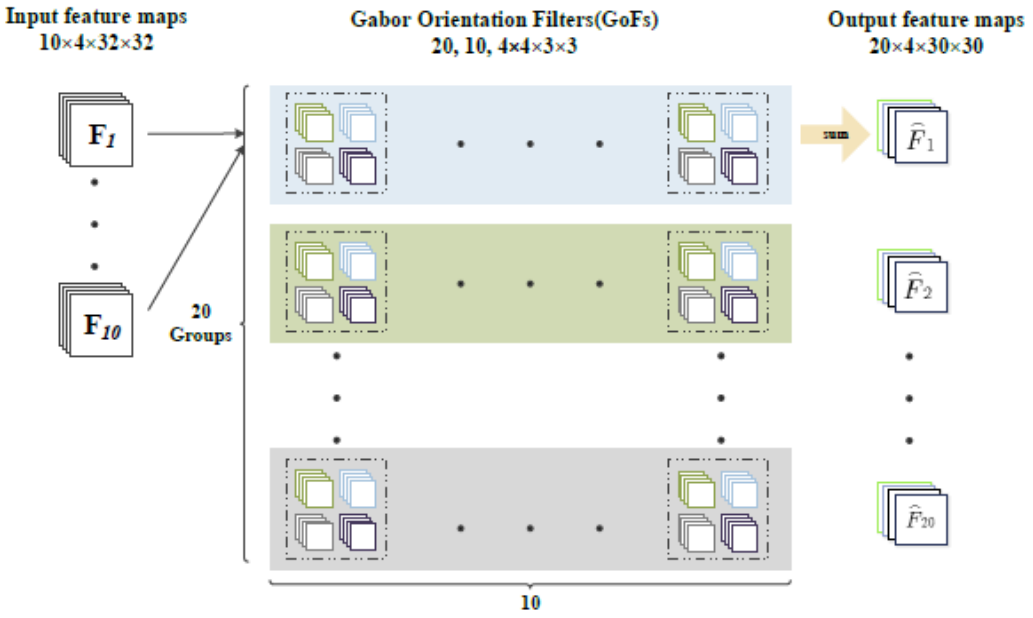

Figure 2.18: Forward Convolution Procedures of Gabor Convolutional Networks with Numerous Feature Maps

Input and output are divided into 10 and 20 function maps. The filters are split into 20 groups and 10 re-constructed filters are contained in each group

In summary, the sub-filter gradient in Gabor Orientation Filters to the respective learned filters is as follows (Uyar et al., 2022).

$$\delta = \frac{\partial L}{\partial C_{i,o}} = \sum_{u=1}^{U} \frac{\partial L}{\partial C_{i,u}} oG(u,v) \qquad (20)$$

Where L represents the loss function. As indicated in the above equation, the Backpropagation process is easily deployed from Gabor kernels. The GCNN's model is only compact and efficient by updating the learned$C_{i,o}$, filters and is also more robust changes in the scale and direction.

Convolutional Gabor Orientation Filters (GoFs)

Gabor filters are of U directions and V scales. To incorporate the steerable properties into the GCNs, the orientation information is encoded in the learned filters, and at the

same time the scale information is embedded into different layers. Due to the orientation and scale information captured by Gabor filters in GoFs, the corresponding convolution features are enhanced.

Before being modulated by Gabor filters, the convolution filters in standard CNNs are learned by the backpropagation (BP) algorithm, which is denoted as learned filters. Let a learned filter be with size $N \times W \times W$, where $W \times W$ is the size of a 2D filter and N refers to a channel.

If the dimensions of the weight per layer in traditional CNNs are expressed as $C_{out} \times C_{in} \times W \times W$, GCNs will represent it as $C_{out} \times C_{in} \times N \times W \times W$, C_{out} and C_{in} represent the channel of output and input feature map respectively.

Where input feature is now integrated Gabor response for each pixel in the modified HSV color space $C_{in} = R(x, y)$.

To keep the channel quantity of the feature map consistent during the forward convolution process, N is chosen to be$U$, which is the number of orientations of the Gabor filters will be used to modulate this learned filter. A GoF is obtained based on a modulated process using U Gabor filters on the learned filters for a given scale V. The details concerning the filter modulation are illustrated in Figure 2.19.

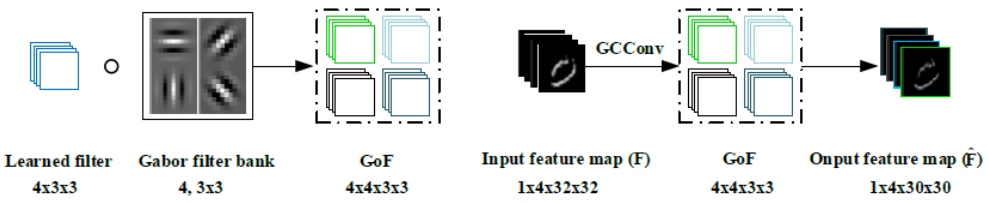

Figure 2.19: Modulation Process of GoFs

Left shows modulation process of GoFs. Right illustrates an example of GCN convolution with 4 channels. In a GoF, the number of channels is set to be the number of Gabor orientations U for implementation convenience.

For the Vth scale, we define:

$$C_{i,u}^{v} = C_{i,o} o G(u,v) \qquad (21)$$

Where $C_{i,o}$ is a learned filter, and o is an element-by-element product operation between $G(u,v)$ and each 2D filter of $C_{i,o}$, $C_{i,u}^{v}$ is the modulated filter of $C_{i,o}$ by the v - scale Gabor filter $G(u,v)$.

Gabor Orientation Filter is defined as:

$$C_i^v = \left(C_{i1}^v, \ldots\ldots C_{i,U}^v\right) \qquad (22)$$

Thus, the ith GoF C_i^v is actually a 3D filter (see Fig. 2, here U = 4). In GoFs, the value increases with increasing layers, which means that the scales of Gabor filters in GoFs are changed based on layers. At each scale, the size of a GoF is $U \times N \times W \times W$. But we only save $N \times W \times W$ sized learned filters, because Gabor filters are given. To simplify the description of the learning process, v is omitted.

GCN convolution

In GCNs, GoFs are used to produce feature maps, which explicitly enhance the scale and orientation information in deep features. An output feature map ($Fmap$) in GCNs is denoted as follows.

$$Fmap = GCconv(F, C_i)$$

$$Fmap = GCconv(R(x,y), C_i)$$

Where C_i is the vth GoF and $R(x,y)$is the input feature map. The channels is obtained by the following convolution:

$$Fmap_{i,k} = \sum_{n=1}^{N} F^n \otimes C_{i,u=k}^{(n)} \qquad (23)$$

Where (n) refers to the nth channel of $R(x, y)$ and $C_{i,u}$, and $Fmap_{i,k}$ is the kth orientation response of $Fmap$. Let the size of the input feature map be$1 \times 4 \times 32 \times 32$. If there are 10 GoFs with 4 Gabor orientations, the size of the output feature map is$10 \times 4 \times 30 \times 30$.

Figure 3 shows the forward convolution process of GCNs when the input feature map is extended to multiple channels ($C_{in} \neq 1$).

The majority of research has focused on Gabor CNN in the first and last layer of standard CNN to address the issue of rotation invariance. The models were resilient to rotations, particularly in the early and last layers where it is important to capture fundamental orientation information. However, it has not been able to address rotation invariance across all layers of standard CNN and the associated challenges, including shifting occlusion, fluctuating illumination conditions, and being insensitive to noise.

2.7 Illumination Invariance

Illumination invariance in the field of image processing is the ability of an image processing algorithm or model to accurately analyze or recognize images despite variations in lighting conditions. According to Ren et al. (2022), an illumination invariant is a function μ of images that is constant on images of an object taken under different illumination conditions. That is, if I_1, I_2 are two images of an object taken under the same viewpoint but different illumination conditions, then $\mu(I_1) = \mu(I_2)$.

Illumination variations can occur due to changes in factors such as the position and intensity of light sources, shadows, and reflections, which can significantly impact the appearance of an image.

Illumination invariance is essential because lighting conditions can vary widely in real-world scenarios, and images captured under different lighting conditions may exhibit significant differences in pixel intensities and contrast (Hu et al., 2024). These variations can pose challenges to various image processing tasks, such as image recognition.

2.7.1 Approaches for Illumination Invariance

Illumination invariance in image processing refers to techniques and approaches designed to make an image analysis system robust to changes in lighting conditions.

Pre-processing

Applying pre-processing techniques to normalize or compensate for illumination variations. This may involve histogram equalization, contrast enhancement, or specific methods designed for illumination normalization.

Image Normalization

Image normalization is a pre-processing method used in image processing to modify an image's brightness and contrast so that it is more suited for subsequent analysis or display. Image normalization aims to improve the visual appeal of the image and make information extraction easier. When we normalise we get two benefits with normalization. First, if data is not normalized, features with larger numerical values dominate features with smaller numerical values and consequently, we will not get contributions from features with smaller values. Second, many learning algorithms behave well with normalized data. This manifests in higher test accuracy for normalized data than with non-normalized data. The purpose of normalization is to bring the image, or other types of signal, into a range that is more familiar or normal to the senses (Badar et al., 2020).

Color Space Transformation

Color space transformation is one of the most widely used methods to achieve illumination invariance. Research shows that color spaces like HSV, LAB, and YCbCr can effectively separate intensity (brightness) from color, thus minimizing the effect of lighting changes on feature extraction. For instance, Marques et al. (2022) demonstrated how certain color spaces could be leveraged to separate illumination and reflectance, allowing object recognition systems to focus on intrinsic colors unaffected by lighting variations. Recent studies have confirmed the effectiveness of the HSV color space for CBIR and facial recognition under variable

lighting conditions, as it enables models to focus on hue and saturation while disregarding changes in brightness Gradient-Based Techniques.

Gradient-Based Techniques

Gradient-based methods have been explored extensively due to their relative robustness to changes in illumination (Wang et al., 2022). Techniques like edge detection and the Histogram of Oriented Gradients (HOG) rely on image gradients, which capture local differences in intensity rather than absolute brightness values. Bhattarai et al. (2023) suggest popularized HOG for object detection, showing it to be effective under diverse lighting because it captures structural and shape-based features instead of relying on color or brightness. Other studies have applied gradient-based methods in face recognition and object detection, highlighting their resilience to illumination changes by focusing on texture and edge features.

Gabor and Directional Filters

Research also demonstrates the use of Gabor filters for illumination invariance, primarily in texture-based feature extraction. Gabor filters, which capture frequency and directional information, are known to be less sensitive to absolute intensity variations and are effective for capturing textured patterns even under fluctuating lighting. Poloni et al. (2021) applied Gabor filters in face recognition to reduce lighting sensitivity, showing that such filters enhance model robustness to shadowing and brightness changes. Studies continue to explore Gabor and similar directional filters as part of hybrid methods, often combining them with other invariance techniques to achieve higher accuracy in CBIR and facial recognition.

Deep Learning Illumination Invariance

Deep learning has revolutionized the field by enabling systems to learn illumination-invariant features directly from data. Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) have been widely studied for their ability to simulate or account for lighting variations (Hu et al., 2021). Wang et al. (2022) demonstrated how data augmentation and synthetic lighting conditions could train

CNNs for robust object detection under various lighting environments. GANs have been used to generate normalized images under uniform illumination, enhancing the illumination robustness of recognition systems. More recent advances include self-attention mechanisms and normalization layers designed to reduce sensitivity to lighting changes, yielding substantial improvements in object detection and retrieval tasks.

Karanwal and Diwakar (2022) proposed the dynamic morphology quotient image (DMQI) pre-processing algorithm based on complex illumination conditions.

Lu et al. (2023) developed a cross-scale and illumination-invariant detection model (CSIM) based on the You Only Look Once (YOLO) architecture to increase the tolerance of large-scale deformations and light changes in outdoor situations. An adaptive cross-scale feature fusion approach to resolve this problem and guarantee the consistency of the built-in feature pyramid was used. Also, the construction of an illumination-invariant chromaticity space on the CSIM model, independent of the corresponding color temperature, to counteract the effects of uneven lighting was used, the model failed to retain the color consistency in the entire image as the image convolved in the architecture. Marlow et al. (2022) created a logarithm of a set of chromaticity coordinates to create an illumination invariant. The algorithm results in an image that is roughly independent of the illuminant at every pixel. Berthier et al., (2021) advanced the model by calibrating the camera to determine the optimal 2D direction to define illumination changes. Barbero-Álvarez et al. (2023) use standardized sRGB color space and sharpening that space, to discover the invariant image with enough accuracy. Rodriguez et al. (2020), suggest that input images may first be considered to be in nonlinear sRGB color space, then linearized to linear-sRGB, and then translated to XYZ. The XYZ curves themselves might then be sharpened in XYZ. Indeed, the model improved the invariant direction. The findings demonstrate that this straightforward approach yields workable results when applied to the elimination of shadows and performs better when establishing an illumination invariant. The method outlined here is advantageously used in place of a more exacting, camera- and image-dependent method for illumination invariant extraction. Koscevic et al. (2020) suggested illumination estimation based on the classification

of light source to make the distribution of image scenes and illuminations less diverse, the input image is classified into images taken in indoor scenes under artificial illumination. Soni et al. (2024) introduce an innovative approach to improve the color quality of underwater images by employing a swift algorithm and a hue-preserving-based mechanism.

The method combines the hue saturation intensity (HIS) and hue saturation value (HSV) color models and applies a noncomplex logarithmic function as a preprocessing stage to remove immoderate pixel values. The technique applies wavelet domain filtering (WDF) and constrained histogram stretching (CHS) methods on HIS and HSV color models, respectively, ensuring the preservation of the hue component (H) in both processes. The suggested approach yields superior outcomes compared to widely recognize underwater enhancement algorithms, as evidenced by both visual and quantitative analyses. The novel method effectively tackles the issues of color variations and low perceptibility in underwater images, leading to successful and dependable image enhancement.

2.8 Thesis Gap

After reviewing all the literature we can say that the color feature is an essential component for image retrieval. For huge image databases, image retrieval using the color feature is very successful and effective. Although the color feature is not a persistent parameter, because it is subjected to many non-surface characteristics such as illumination and, characteristics of the device (Zhou et al., 2021). Color extraction is susceptible to issues like variable lighting conditions and occlusions, therefore it might not be the most reliable technique. To solve these issues, in addition to the traditional RGB (Red, Green, Blue), alternative color spaces are employed to extract color information from the image to be recognized, including HSV (Hue, Saturation, Value), HIS (Hue, Intensity, Saturation), and *L*a*b* (CIELAB), among others.

By transforming the color representation from one color space to another, color space transformation is shown to be a useful technique for mitigating the effects of variations in illumination. Burambekova and Shamoi (2024) noted the impacts of illumination variations have been reduced using RGB, HSV, and CIELAB color

spaces, however, the effects of illumination invariance are still a challenge that require to be addressed.

The HSV color space is commonly applied in CBIR models to separate brightness (value) from hue and saturation, helping maintain color consistency under fluctuating illumination conditions. Studies by Alrahhal and Supreethi (2024) and Ruby et al. (2022) apply HSV transformations within early layers of CNNs, noting improvements in color consistency. However, these studies do not explore how integrating HSV color across multiple layers either early, middle, or late might improve retrieval performance or reduce computational load for more complex datasets. Thus, a gap remains in understanding whether HSV transformations benefit from selective, layer-specific applications within the CNN hierarchy, especially when combined with texture and Gabor filters.

Secondly, texture feature extraction was reviewed. The texture of an image can be thought of as the spatial variations in pixel brightness intensity in the field of image processing. Some authors proposed to define texture as a measure of coarseness, contrast, directionality, line-likeness, regularity, and roughness (Tamura et al., 1978). Directional texture filters are crucial for texture differentiation, capturing patterns and orientations that define an image's surface details. Research by Zhuang et al. (2022) and Kociołek et al. (2022) highlights that applying directional texture filters in CNNs enhances texture-based retrieval. However, these studies often apply these filters in a single layer, typically the middle or late layers, assuming that texture patterns will automatically propagate to other layers. This approach does not examine the layer-by-layer impact of texture filter integration. A selective layer approach may allow for more nuanced texture extraction and improve robustness across varying textures, filling a significant gap in current methodologies.

Gabor filters have demonstrated high efficacy in achieving rotation invariance due to their orientation sensitivity. Srivastava et al. (2022) and Hadid et al. (2023) discuss how Gabor filters enhance CBIR performance by detecting edges and textures that remain consistent under rotational transformations. Most research, however, applies Gabor filters either in a single layer or uniformly across multiple layers, lacking a

systematic analysis of how each layer might contribute differently to rotation invariance. By implementing Gabor filters layer-by-layer in this study, it is possible to assess how each layer's Gabor integration impacts both rotational resilience and computational efficiency, addressing an unexplored dimension in Gabor-based CBIR.

Few studies analyze combined layer configurations, instead often focusing on either early, middle, or late applications of HSV, texture, or Gabor filters. Liu & Yang (2023) noted that applying texture and rotation filters in later layers benefits complex datasets, but they did not examine the potential for combining different configurations (e.g., early-middle or middle-late layers) to balance performance and resource use across different data complexities. Additionally, no existing study has addressed how combining HSV color, texture, and Gabor filters layer-by-layer might yield optimal results, potentially allowing for selective layer engagement based on dataset characteristics (e.g., simpler vs. complex datasets). Thus, a gap remains in understanding the effects of extended GCNN on CBIR performance, particularly to different feature complexities.

The research has proposed to bridge the gap using a layer-by-layer approach in integrating HSV color, directional texture, and Gabor filters offering a new pathway for optimizing CBIR performance. By addressing the unique role of each CNN layer in extracting color, texture, and rotation features, this study fills a gap in existing methodologies, which primarily apply these filters in isolated or uniform layers. Layer-by-layer integration also enables performance benchmarking at each stage, providing a finer understanding of how each feature type contributes to CBIR outcomes and how computational resources are allocated across layers.

CHAPTER THREE

METHODOLOGY

3.1 Introduction

This chapter describes the system's conceptual design followed by developing the new novel extended GCNN model. The extended GCNN model was developed by integrating HSV color, directional texture, and Gabor filters to offer a new pathway for optimizing CBIR performance.

The extended GCNN model is the preferred model due to the following reasons. Gabor filters are optimized for extracting spatial-frequency features, making them ideal for texture and color-based retrieval systems compared to VGG and ResNet models that are not inherently optimized for color and texture (Wang et al., 2021). Secondly, Gabor filters are designed to capture fine-grained texture and edge information more efficiently than traditional CNN that rely on learned filters (Muzaffar et al., 2023).

Figure 3.1 is the general overview of the conceptual framework that illustrates how images were extracted and retrieved. First images were acquired from CIFAR-10 and ImageNet dataset repository followed by color and texture feature extraction. Secondly, the data set is divided into trained and test data sets. Finally, Gabor CNN's layer-by-layer approach was used to retrieve the images.

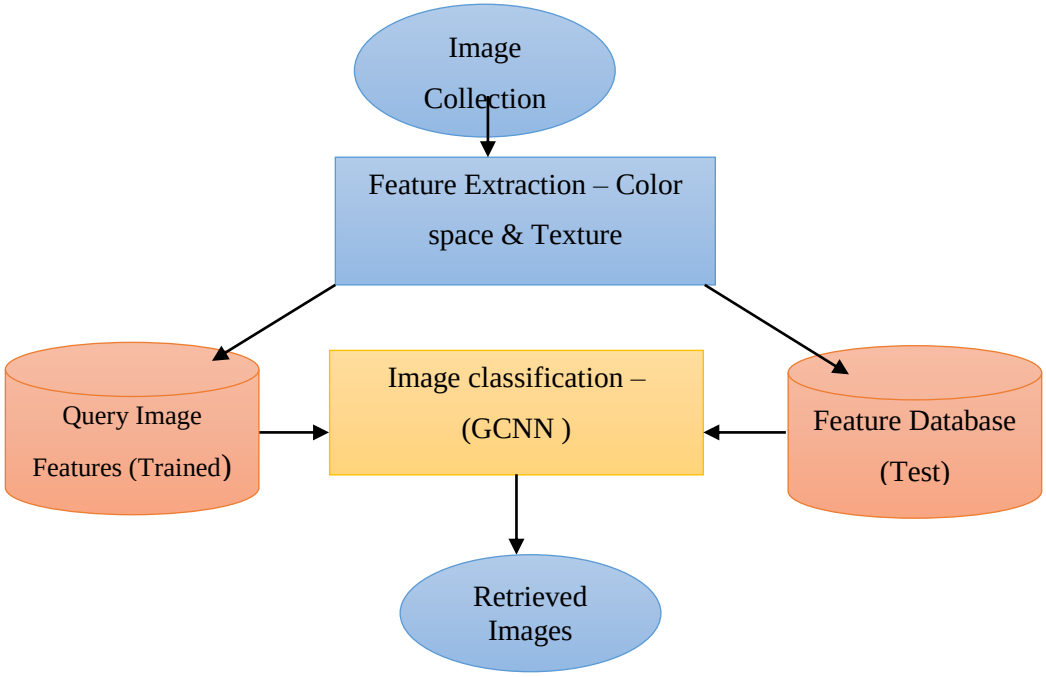

Figure 3.1: Conceptual Framework of CBIR

During the literature review the study reveals color transformation to be a useful technique for mitigating the effects of variations in illumination. Secondly, Texture features capture local variations in pixel intensities, such as fine grain, roughness, or regular patterns that are present in different regions of an image. Thirdly, Gabor CNN has been utilized to handle rotation invariance.

Image Feature Extraction

First, image features were extracted using three different color spaces namely RGB, HSV, and LAB color space. The images were transformed to different color space by converting RGB color space which is the default color of the image to HSV color space using equation (2) and LAB color space. Secondly, the image features were extracted using texture and directional texture filter features using equation (5) through the Gabor filter. The model was measured by accuracy and recall rate metric through conventional convolutional neural network and Gabor convolutional neural network. Thirdly, a layer-by-layer approach was evaluated where Early, Middle, Late, and Combined Layers were evaluated. Figure 3.2 presents a detailed workflow diagram that describe how the layer-by-layer approach in integrating HSV color,

directional texture, and Gabor filters offers a new pathway for optimizing CBIR performance.

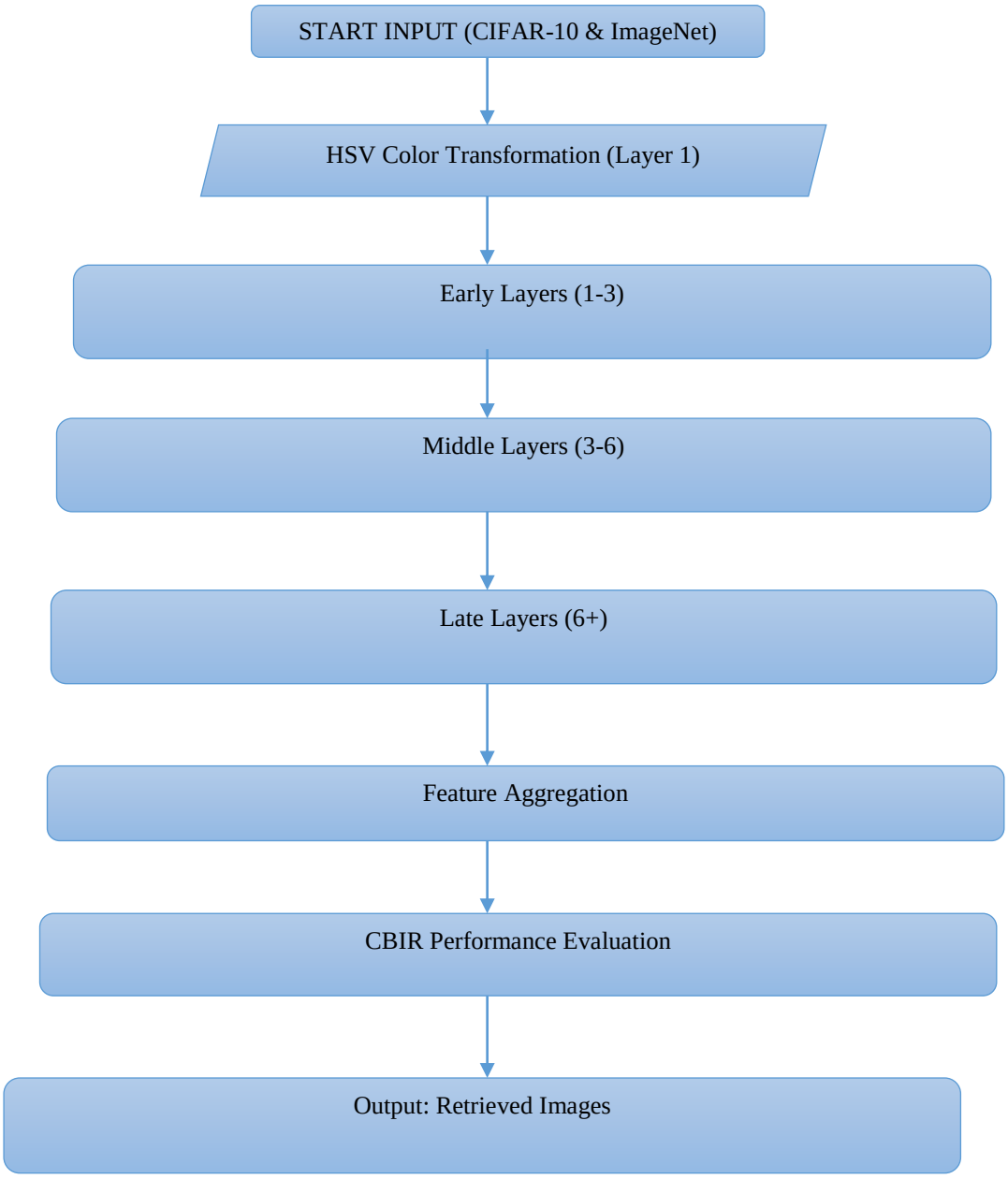

Figure 3.2: Workflow Diagram of Proposed Model

Extended Gabor Convolution Neural Network

1. Start – Input Image
2. The input image is pre-processed (Image resize) and fed into the model pipeline.
3. HSV color Transformation (Early Layer)

   The HSV color space transformation is applied in the first layer. This transformation helps separate hue, saturation, and value for improved illumination invariance.

4. Layer 1-3: Early Layers Processing

   Directional Texture Filter Application to capture basic texture patterns. Apply Gabor filters to capture initial rotation-invariant features.

   Initial color and texture features with enhanced rotation and illumination invariance were achieved.

5. Layer 3-6: Middle Layers Processing

   HSV and Directional Texture Refinement to enhance color and texture feature extraction, capturing mid-level image patterns and details. Gabor Filter for Intermediate Rotation Invariance to add, additional rotation-sensitive filters to capture intermediate level rotational consistency. The process ensured Mid-level color and texture features with refined rotation invariance.

6. Layer 6+: Late Layers Processing

   Complex Texture Extraction to enhance complex texture abstraction to handle high-detail features. Final Gabor Application to capture finer rotation-invariant features. Detailed, high-level features with robust rotation and illumination invariance were achieved.

7. Output Stage

    - Feature Aggregation - Combine extracted features from all layers for CBIR.
    - Performance Evaluation - Assess CBIR performance metrics (Accuracy, Precision, Recall) based on layer configurations.
    - Final Output - Image retrieval results based on layer-by-layer integrated feature extraction.

3.2 Experiments

The experiments were designed to evaluate the proposed content-based image retrieval system's performance in achieving rotational and illumination invariance. The primary goal was to test whether integrating rotational and illumination invariance into CNNs could improve retrieval accuracy for CBIR.

3.2.1 Experiment 1: Feature Extraction Using Color

Color transformation - Convert the input image into different color spaces (e.g., HSV, LAB, RGB) before feature extraction. Each color space can emphasize specific aspects of the image, such as intensity, saturation, or hue, enhancing the overall texture representation. According to Afaq and Rao (2020), the importance of epochs during the training of a neural network has an impact on the model's accuracy. Consequently, it is essential to choose the best epochs possible without under fitting or overfitting. Also, Chi et al. (2022) demonstrated the importance of optimal epochs when performing analysis of the three kinds of neural networks in image retrieval images.

3.2.2 Experiment 2: Feature Extraction Using Texture

Texture features through the aspect of directional texture filters were extracted using the Gabor filter which is integrated into CNN to form GCNN as described in Figure 3.3.

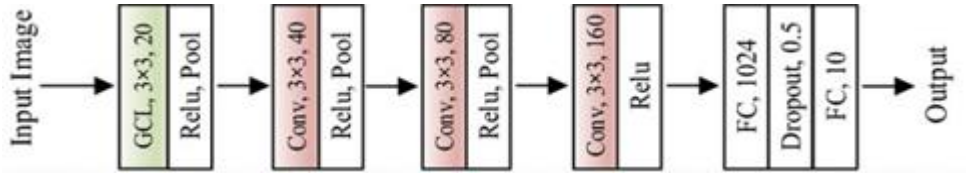

Figure 3.3: The Architecture of a Gabor Convolutional Layer Used in the First Layer of Conventional Convolutional Neural Network

Texture and directional texture filters, especially when integrated across CNN layers, allow for capturing both static and directional textures that enhance CBIR system performance. Gabor filters, in particular, are instrumental in extracting complex textures and orientations, yielding highly detailed and robust feature maps suitable for accurate and reliable image retrieval.

There are four stages in the CNN architecture. Convolution Layer, RELU Layer, and Max Pooling Layer make up the first two phases. The RELU Layer and Convolution Layer make up the third step. In the last stage, there is a layer that is completely connected. Gabor Kernels are used to initialize the Convolution layer's first stage, and CNN's standard design is used for the remaining layers. The first convolution layer is designed based on 16 Gabor filters with uniformly distributed θ (orientation) values. A Gabor kernel of size 3×3 is used as a convolution filter, with Scale (σ) as 4.1, γ as 5.3, and λ as 0.3. The selection of all the values is based on experimental analysis. The first Convolutional layer converts a single input image into 10 output images which are passed to the RELU layer.

The second Convolution Layer and third stage work on kernels initialized with random normal distribution values with a standard deviation of 0.0001. Biases are

initialized with a normal distribution of mean value 1 and standard deviation 0.00001. The learning rate factor of the bias is set to 2. A total of 32 and 64 filters are used in the first and second Convolution layer respectively. Rectified Linear Unit (ReLU) layer performs a threshold operation on each element, where any input value less than zero is set to zero. The ReLU Layer is utilized in the initial three stages. The Max pooling layer performs max-pooling with a kernel size of 2 × 2 that works with a stride of '2' and is used in the first and second Layers.

A fully connected layer is initialized with Normal distribution with a mean of 0 (Zero) and a standard deviation of 0.0001 for the weight values. Bias values are initialized with normal distribution with a mean of 1 and deviation of 0.0001. The learning rate factor of the bias is set to 2. *Softmax* is used as an activation function in the output layer of the fully connected layer.

3.2.3 Experiment 3: Layer-by-Layer Configuration

Experiments on CIFAR-10 and ImageNet dataset with integrating HSV color space, directional texture filters, and Gabor filters applied to different layer configurations (early, middle, combined, last, and all layers) were evaluated. Figure 3.4 demonstrates layers convolved with HSV color space, directional texture filters, and Gabor filters respectively.

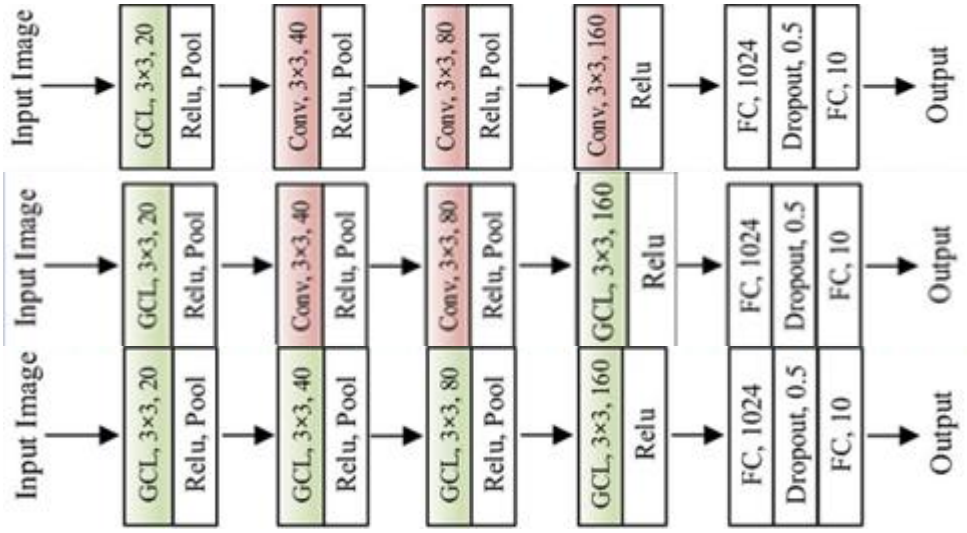

Figure 3.4: Network Structures of Gabor Filter Used in Different Layers of Conventional Neural Network

Convert RGB to HSV image

$I_{rgb(x,y)}$, where (x,y) represents the pixel coordinates.

Convert each RGB pixel to its corresponding HSV values.

$$= H(x,y), S(x,y), V(x,y) \quad (24)$$

Modify the input representation to use modified HSV representation instead of using RGB image to the Gabor Convolutional Network.

Modified input $= I_{hsv(x,y)}$ where

$$I_{hsv(x,y)} = [H(x,y), S(x,y), V(x,y)] \quad (25)$$

Adapt the Gabor filters to the HSV color space.

$$G(x,y,\theta,\lambda,\sigma,\gamma) \quad (26)$$

Where (x,y) represents the filter's spatial coordinates, θ is the orientation, λ is the wavelength, σ is the standard deviation, and γ is the aspect ratio.

Gabor response for each channel is computed as follows.

$$G\_h(x,y) = G(x,y,\theta,\lambda,\sigma,\gamma) \times H(x,y)$$

$$G\_s(x,y) = G(x,y,\theta,\lambda,\sigma,\gamma) \times S(x,y) \quad (27)$$

$$G\_v(x,y) = G(x,y,\theta,\lambda,\sigma,\gamma) \times V(x,y)$$

Where$G_h(x,y)$, $G_s(x,y)$and $G_v(x,y)$are the responses of the hue, saturation, and value channels, respectively.

Convolve each Gabor response with the corresponding modified input channel

$$R\_h(x,y) = I\_hsv(x,y) \otimes G\_h(x,y)$$

$$R\_s(x,y) = I\_hsv(x,y) \otimes G\_s(x,y) \quad (28)$$

$$R\_v(x,y) = I\_hsv(x,y) \otimes G\_v(x,y)$$

Where $\otimes$denotes the convolution operation.

Finally, combine the Gabor responses for each channel to obtain an integrated Gabor response for each pixel in the modified HSV color space.

$$R(x,y) = [R\_h(x,y), R\_s(x,y), R\_v(x,y)] \quad (29)$$

Convolutional Gabor orientation Filters (GoFs)

Where input feature is now integrated Gabor response for each pixel in the modified HSV color space $C_{in} = R(x,y)$. (30)

GCN convolution

In GCNs, GoFs are used to produce feature maps, which explicitly enhance the scale and orientation information in deep features. An output feature map ($Fmap$) in GCNs is denoted in equation 31.

$$Fmap = GCconv(F, C_i)$$

$$Fmap = GCconv(R(x,y), C_i) \quad (31)$$

Where C_i is the vth GoF and $R(x,y)$is the input feature map. The channels are obtained by the following convolution in equation 32.

$$Fmap_{i,k} = \sum_{n=1}^{N} F^n \otimes C_{i,u=k}^{(n)} \quad (32)$$

Where (n) refers to the nth channel of $R(x,y)$ and $C_{i,u}$, and $Fmap_{i,k}$ is the kth orientation response of $Fmap$.

Early Layers – Layers 1-3

This experiment aimed to test how integrating HSV color, directional texture, and Gabor filters could improve the performance of Content-Based Image Retrieval (CBIR) systems in the early layers of a CNN. This setup focuses on testing the retrieval effectiveness and efficiency of early-layer integration, examining how foundational color, texture, and rotation features contribute to CBIR performance across different datasets.

Middle Layers Only – Layers 3-6

This experiment aimed to evaluate the effects of integrating HSV color space, directional texture filters, and Gabor filters exclusively in the middle layers of a CNN, balancing basic feature extraction with higher-level feature abstraction. This configuration is designed to refine Color and Texture Features using HSV and directional filters in middle layers to enhance feature depth. Secondly, to boost

Rotational Invariance Mid-Processing by applying Gabor filters mid-network to capture rotation-invariant patterns at an intermediate stage.

Late Layers – Layers 6+

This experiment aimed to test how integrating HSV color space, directional texture filters, and Gabor filters exclusively in the late layers of a CNN affects CBIR performance. Convert images to HSV color space in the input processing layer to separate brightness from color information and feed them into the Late Layers to reinforce robustness in abstract color differentiation. Apply directional texture filters within the Late Layers, allowing the model to capture orientation. Tune the Gabor filter to capture orientation.

Combined Layers – (Early and Middle, Middle and Late, Early and Late)

Different layers were combined sequentially to evaluate the performance of the Content-Based Image Retrieval (CBIR) system. Each layer represented a specific feature extraction method contributing unique information about the images' color and texture properties. By layering these features, the system was able to capture a more comprehensive representation of the images, allowing for a more accurate and efficient retrieval process. This multi-layered approach was then tested to assess its effectiveness in improving overall CBIR performance.

Available data are typically split into two sets: a training, and a test set. A training set is used to train a network. A test set is ideally used only once at the very end of the project to evaluate the performance of the final model.

The last step is to test the accuracy of the model which is done by compute function. The compute function creates the prediction value. A result variable compares the predicted data with the actual data and creates a confusion matrix with the table function to compare the number of true/false positives and negatives. These were the outcomes after the entire process was completed through matching. The retrieval output image was a relevant image similar to the query image. To compute the retrieval efficiency of relevant images, accuracy, precision, and recall rate were used.

Accuracy: The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier as indicated in equation 33. The accuracy of a classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data (i.e., tuples without class label information). Similarly, the accuracy of a predictor refers to how well a given predictor can guess the value of the predicted attribute for new or previously unseen data (Deng & Zheng, 2021).

$$Accuracy = \frac{Correct\ classifications}{All\ classifications} \tag{33}$$

Precision (specificity) is a measure of the system ability in retrieving only similar images to the query image as denoted in equation 34.

$$precision = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ Number\ of\ images\ retrieved} \tag{34}$$

Recall measures the ability of the system to retrieve all the relevant models, while precision measures the ability of the system to retrieve only the relevant models. The Recall rate which is known as the true positive rate or sensitivity, measures the ability of CBIR systems in terms of the number of similar images retrieved with their similar images in the database as denoted in equation 36.

$$Recall = \frac{Number\ of\ relevant\ images\ retrieved}{Total\ Number\ of\ \ relevant\ images} \tag{36}$$

The number of relevant items retrieved is the number of returned images that are similar to the query image in this case. The total number of items retrieved is the number of images that are returned by the search engine.

3.3 Model Evaluation

The model performance was evaluated with accuracy and loss function for the training, and test datasets. The characteristics derived from confusion matrices were used to compare predicted images.

3.4 Model Implementation Environment

The model was implemented in Google Colab using Python version 3.8, providing an accessible and powerful environment for running complex computations. Google Colab allowed for the use of GPU acceleration, which significantly improved processing speeds, especially for handling image data in the CBIR model. By leveraging Python's extensive libraries, such as OpenCV for image processing, NumPy for numerical computations, and SciPy for scientific computing, the model could efficiently perform feature extraction and similarity matching. This setup facilitated testing, adjustments, and iterative development, ensuring that the CBIR model achieved optimal performance.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the results and discussions, detailing the findings from the CBIR model's performance analysis and evaluating how well the extended GCNN model met the research objectives. Each result is discussed in terms of its impact on retrieval accuracy, precision, and recall, providing insights into the effectiveness of each feature extraction layer. The discussions also include comparisons between this multi-layered approach and other traditional CBIR techniques, highlighting improvements in handling color and texture variations as well as potential limitations observed during testing. This comprehensive analysis offers a clearer understanding of the model's strengths and areas for further refinement, emphasizing its contributions to CBIR optimization.

4.1.1 Experimental Study 1: Epochs Results

Figure 4.1 – Figure 4.6 represent the loss, accuracy, and confusion matrix of feature extraction and classification using different 50, 100, 150, and 200 epochs. Table 4.1 also, represents the summary of different epochs in metric. The result shows that the 150 epochs outperformed the 200 epochs. At 200 epochs, the investigation revealed a decline in performance, indicating that the model had essentially overfitted, or "memorized," the training examples. This overfitting caused the model to struggle with generalization, making it less effective at adapting to new or slightly different input data, which reduced its ability to accurately retrieve relevant images outside the training set. While increasing the number of epochs initially enhanced performance, the rate of improvement significantly diminished after reaching 150 epochs. Beyond this point, further training contributed minimal benefits and even led to a slight decrease in accuracy as the model became more rigid and specific to the training data. Consequently, at 200 epochs, the model's accuracy dropped to 96.4%, prompting the decision to limit training to a maximum of 150 epochs to prevent overfitting and ensure better generalization across varied image inputs.

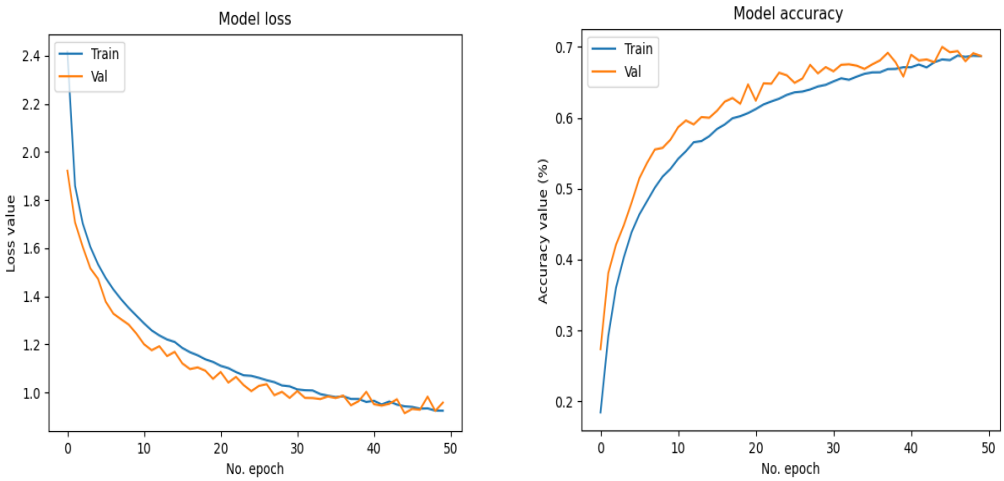

Figure 4.1: Loss and Accuracy Report for 50 Epochs

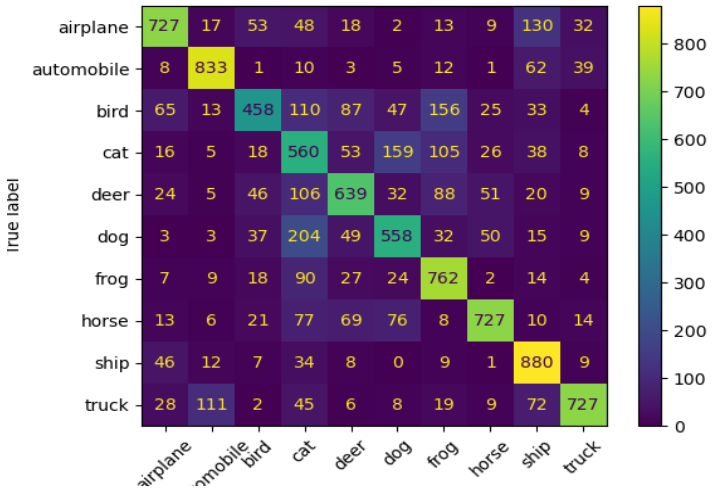

Figure 4.2: Confusion Matrix for 50 Epochs

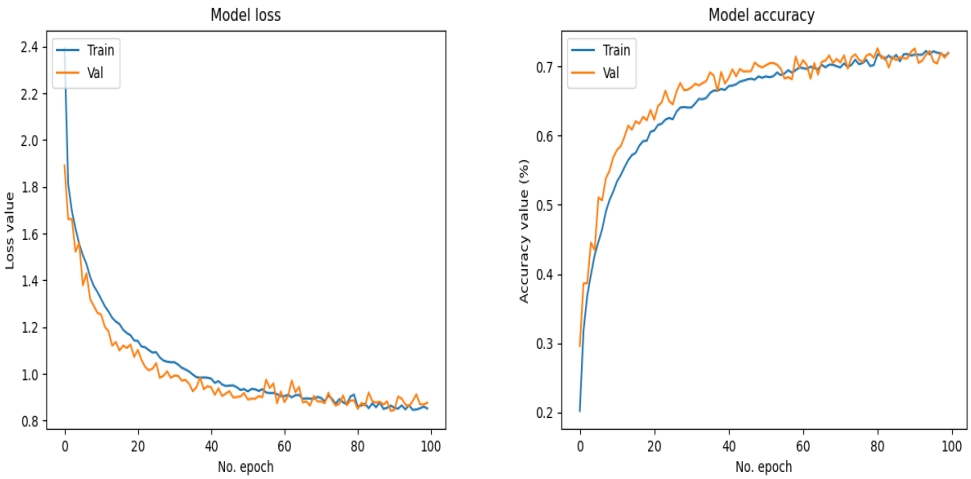

Figure 4.3: Loss and Accuracy Report for 100 Epochs

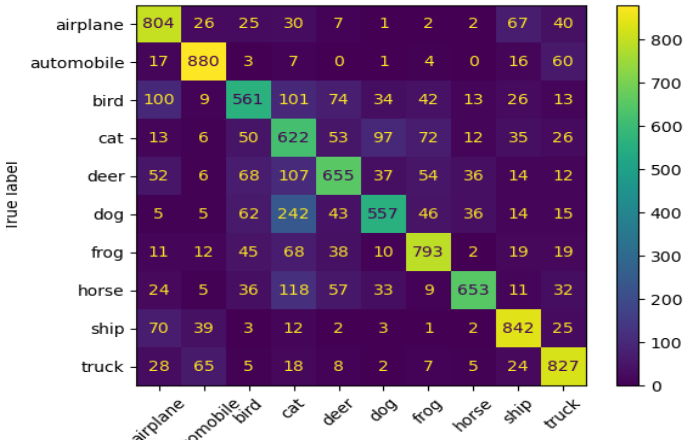

Figure 4.4: Confusion Matrix for 100 Epochs

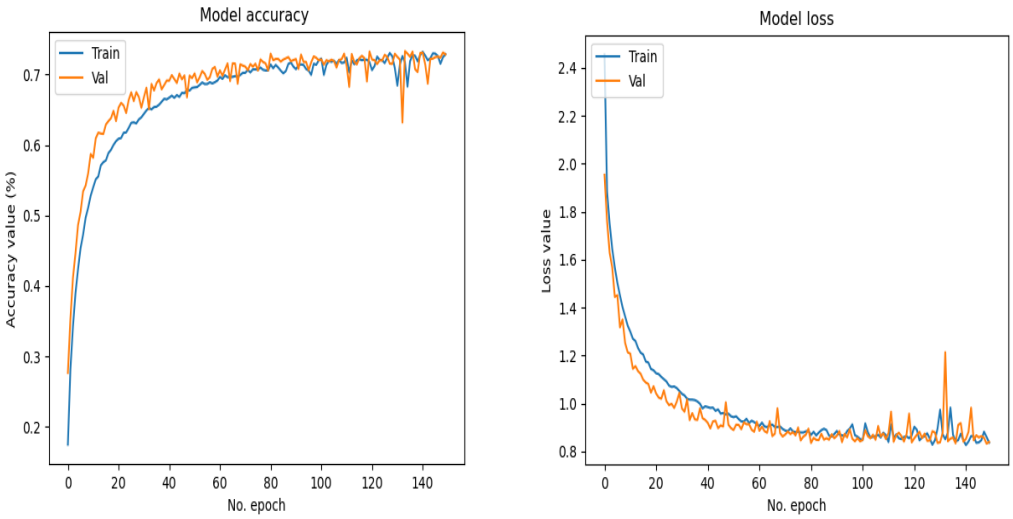

Figure 4.5: Loss and Accuracy Report for 150 Epochs

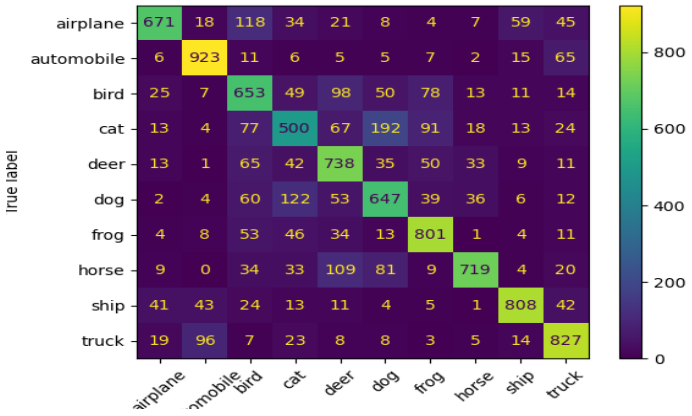

Figure 4.6: Confusion Matrix and Classification for 150 Epochs

Table 4.1 demonstrates a metric summary for different epochs. The accuracy of the CNN model at 50, 100, 150, and 200 epochs, respectively, was 96.3%, 96.5%, 96.6%, and 96.4%. The outcome demonstrates that accuracy and loss are consistent throughout 100 epochs due to the CNN model's stability after multiple iterations.

Table 4.1: Metric Summary for Different Epochs

Epochs	Accuracy	Precision	Recall
50	96.3	96.4	96.4
100	96.5	96.5	96.4
150	96.6	96.5	96.5
200	96.4	96.4	96.4

4.1.2 Experimental Study 1: Feature Extraction Using Different Color Spaces

The results of different color spaces are presented in Table 4.2 and Figure 4.7. The study found that preprocessing images by transforming them into different color spaces yielded different results. Although, the accuracy by itself did not vary too much, on closer inspection with the aid of confusion matrices we could see that, there was not a 100 percent correlation between the results. For color space, HSV yielded the highest accuracy and recall value of 99.41%. The fact that most existing feature extraction techniques assume RGB color space as the optimal color space because it is the default color space, they forget that HSV is very crucial.

Table 4.2: Feature Extraction Using Different Color Spaces

	RGB	HSV	CIELAB
Accuracy	99.39	99.41	99.40
Precision	99.39	99.39	99.39
Recall	99.38	99.41	99.41

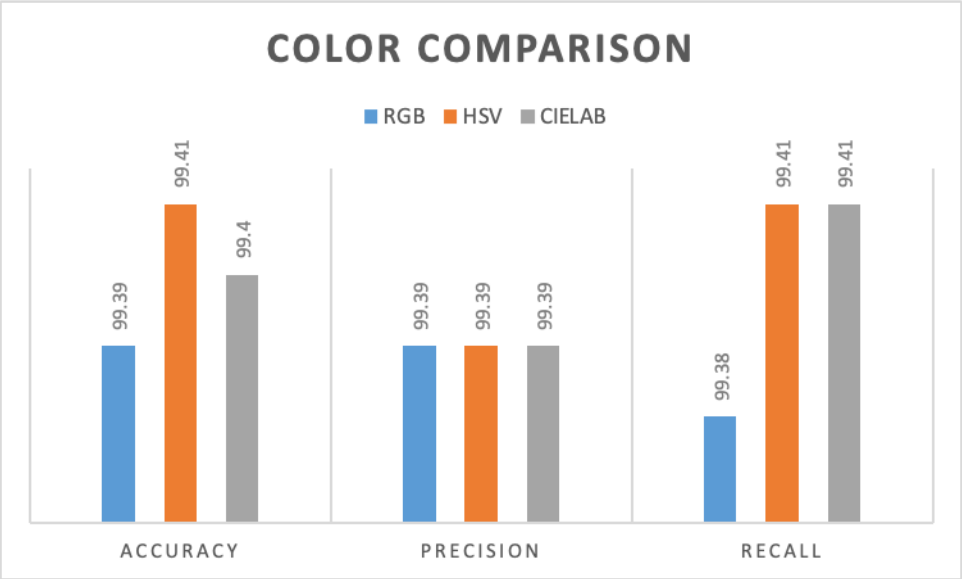

Figure 4.7: Graph of Color Space Feature Extraction Comparison

4.1.3 Experimental Study 2: Gabor Filter Parameters

Table 4.3 illustrates summary of different kernel size. The system is tested with Gabor filters of sizes 3 × 3, 5 × 5, and 7 × 7. The highest efficiency was produced when the kernel size is 3 × 3. When the size of the filter is 3 × 3 the efficiency of the system is 99.68 and when the kernel size is 5 × 5 the efficiency of the system is 98.69 and when the size of the kernel is 7 × 7 the efficiency of the system is 98.46. Therefore, a kernel size of 3 × 3 is selected. The selection of σ, γ, and λ was also carried out based on experimental analysis. It was found that when σ is 4.1, γ is 5.3 and λ is 0.3 the system produces the highest recognition efficiency. Combinations of different values of σ and γ ranging between 1 and 10 are tested for the system and it was found that σ values between 4 and 5 and γ values between 5 to 6 produce the highest recognition rate. Again, testing of the recognition system was repeated with different combinations of σ between 4.1 to 4.9 and γ between 5.1 to 5.9 and the highest efficiency was reported when σ was 4.1 and γ was 5.3. Fixing σ as 4.1 and γ as 5.3 different λ values ranging from 0 to 0.5 were tested and it was found that fixing λ values to 0.3 will increase the efficiency of the system.

Table 4.3: Retrieval Results Using Different Kernel Size

Kernel size	Accuracy	Precision	Recall
3 × 3	99.68	99.46	99.45
5 × 5	98.69	98.45	98.44
7 × 7	98.46	98.12	98.12

Gabor filters exhibit sensitivity to both spatial frequency and orientation, making them suitable for studies focusing on minute details or high-frequency patterns inside images. Smaller kernel sizes are suitable for fine details or high-frequency patterns in the image while larger kernel sizes are suitable for detecting broader patterns or low-frequency components. Therefore, smaller kernel sizes were selected to preserve the image's orientation and scale.

4.1.4 Experimental Study 2: Texture Feature Extraction

Table 4.4 and Figure 4.8 describe the results of texture extraction using texture and directional texture filters. The directional texture filter demonstrated a high accuracy and recall rate of 99.28%. These results have revealed directional filters offer a powerful approach to extracting texture features due to their ability to encode both frequency and orientation information.

Table 4.4: Texture Feature Extraction Results

	Texture	Directional Texture filter
Accuracy	99.23	99.28
Precision	99.25	99.28
Recall	99.24	99.28

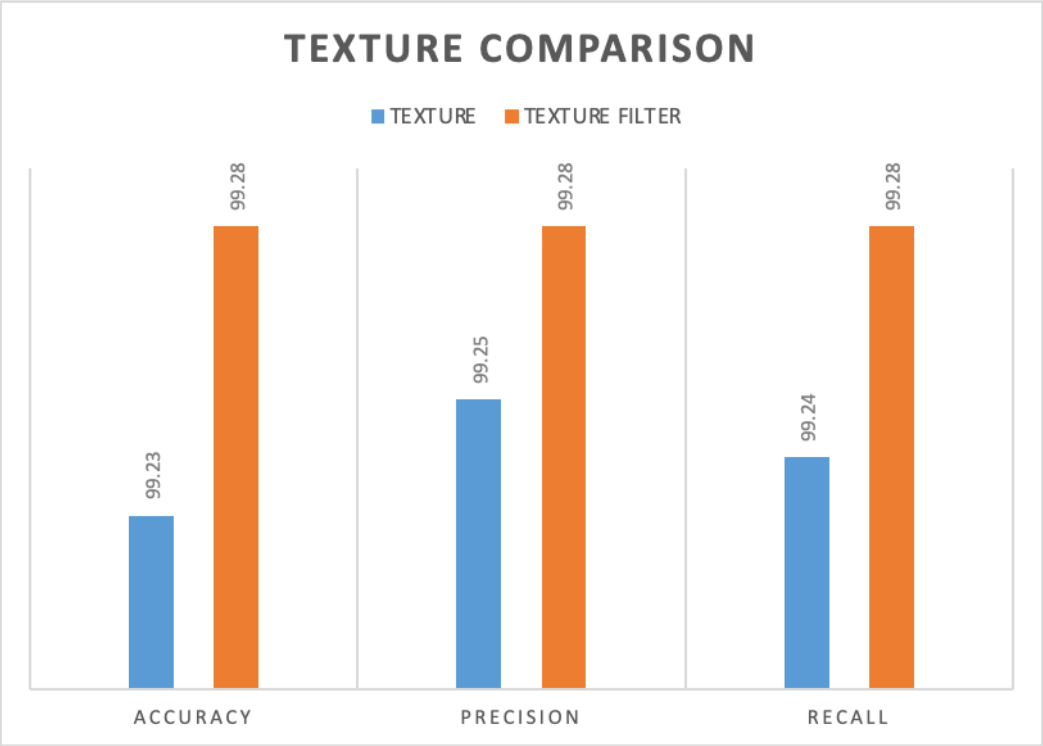

Figure 4.8: Graph for Texture Feature Extraction Comparison

4.2.5 Experimental Study 3: Comparative Analysis of Layer Configurations

Table 4.5 and Figure 4.9 represent the comparative analysis of layer configurations involving evaluating different combinations and sequences of the HSV color, directional texture, and Gabor filter layers CIFAR-10 and ImageNet dataset to determine the optimal structure for enhancing Content-Based Image Retrieval (CBIR) performance. Each configuration was tested to see how the order and presence of these layers influenced the model's ability to capture diverse image features and improve retrieval accuracy.

Table 4.5: Comparative Analysis of Layer Configurations

Layer Configuration	Dataset	Accuracy	Precision	Recall %
Early Layers Layers 1-3	CIFAR 10	99.38	99.37	99.38
	ImageNet	99.35	99.34	99.33
Middle Layers Layers 3-6	CIFAR 10	99.45	99.44	99.43
	ImageNet	99.42	99.40	99.41
Late Layers Layers 6+	CIFAR 10	99.50	99.49	99.48
	ImageNet	99.55	99.54	99.53
Combined Early and Middle Layers	CIFAR 10	99.47	99.46	99.45
	ImageNet	99.48	99.47	99.46
Combined Middle & Late Layers	CIFAR 10	99.52	99.51	99.50
	ImageNet	99.56	55.55	99.54
Combined Early & Late Layers	CIFAR 10	99.49	99.48	99.47
	ImageNet	99.51	99.50	99.49
All Layers	CIFAR 10	99.61	99.60	99.59
	ImageNet	99.63	99.62	99.61

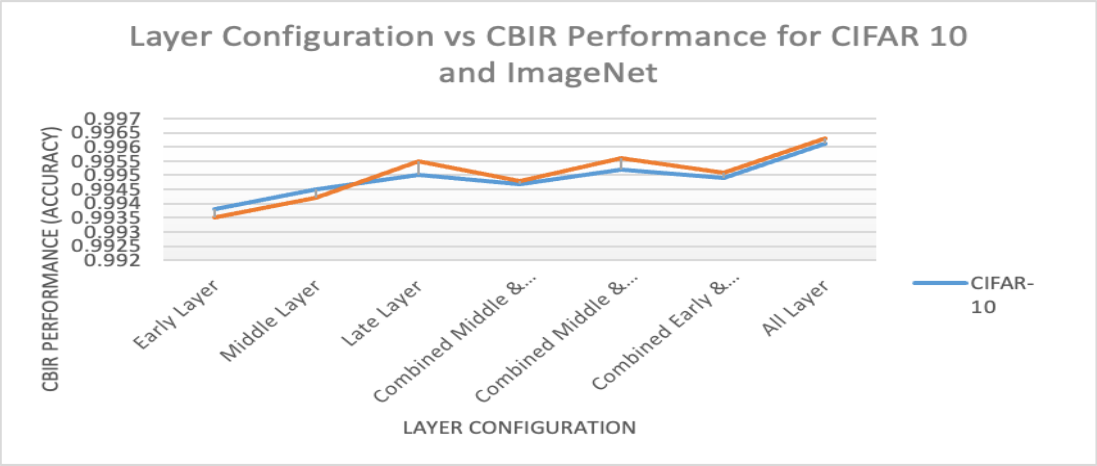

Figure 4.9: Layer Configurations vs CBIR Performance for CIFAR-10 and ImageNet

Table 4.6 and Figure 4.10 represent time complexity to assess whether the model is efficient and how it scales with larger datasets. The graph shows that complexity is not exponential, suggesting that further investigations could be valuable as part of future work.

Table 4.6: Layers Configuration vs Time Taken for CIFAR-10 and ImageNet

Layer Configuration	Time Taken (Seconds) (CIFAR-10)	Time Taken (Seconds) (ImageNet)
Early Layers (1-3)	12.3	20.8
Middle Layers (3-6)	18.4	30.5
Late Layers (6+)	24.9	40.7
Combined Early & Middle Layers	20.1	35.3
Combined Middle & Late Layers	27.5	48.2
Combined Early & Late Layers	23.8	43.9
All Layers	35.6	60.4

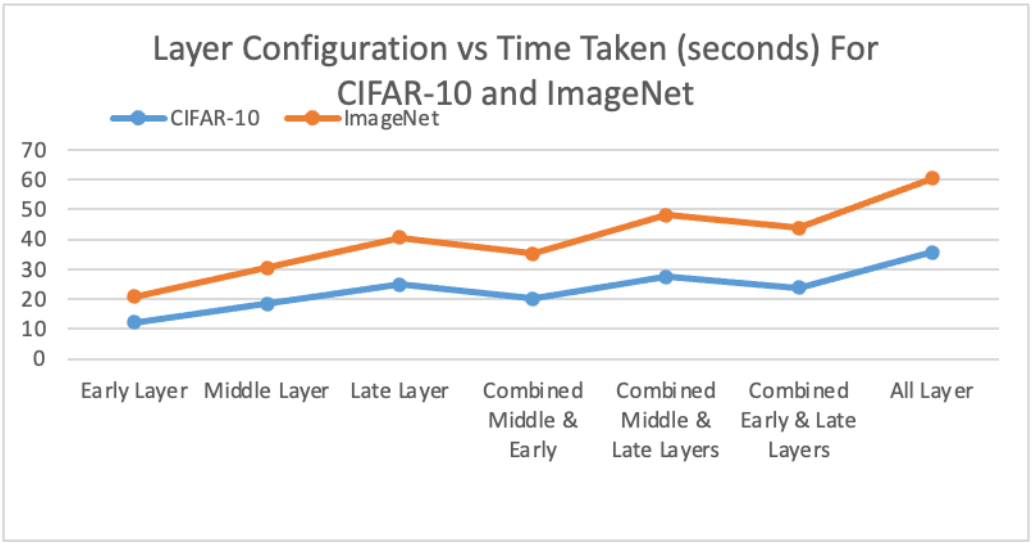

Figure 4.10: Layer Configurations vs Time Taken for CIFAR-10 and ImageNet

In the *Early Layers* (Layers 1-3), applying HSV color space, directional texture, and Gabor filters demonstrated significant improvements in illumination invariance and initial texture capture, leading to strong performance on CIFAR-10. However, early integration struggled to meet the demands of ImageNet's more complex patterns, achieving modest rotation and texture differentiation without fully capturing high-level features.

Middle Layers (Layers 3-6) further refined color separation and enhanced texture capture, improving robustness for texture-rich datasets like ImageNet. Middle layers offered increased resilience to moderate rotations and lighting variations, boosting retrieval rates for both datasets. Nonetheless, the configuration was still limited for intricate rotations and fine-grained textures required by complex images, especially on ImageNet.

The *Combined Early and Middle Layers* configuration (Layers 1-6) provided a balanced approach by integrating foundational and refined feature extraction capabilities. This combination improved color, texture, and moderate rotational invariance for CIFAR-10, outperforming either early or middle layer integration alone. For ImageNet, this combination captured a broader range of textures and lighting variations but faced limitations with extensive rotations and high-level abstractions.

Late Layers (Layers 6+) integration offered the most precise feature abstraction, handling complex rotations, illumination variance, and fine-grained textures effectively. This approach yielded the highest accuracy and recall for both datasets, making it the optimal configuration for high precision tasks. However, the deeper layers also increased computational demands, highlighting a trade-off between performance and efficiency.

Finally, applying all filters across *All Layers* resulted in the highest precision and recall rates across configurations, particularly beneficial for complex datasets like ImageNet. This comprehensive integration allowed the model to handle intricate variations in color, texture, and orientation. While effective, the all-layer approach was computationally intensive, suggesting it is best suited for high-stakes retrieval

tasks where resource availability supports robust feature extraction across the network.

Table 4.7 concisely captures the qualitative findings and the discussion points for CIFAR-10 and ImageNet, offering insights into how each layer configuration impacts the CBIR performance.

Table 4.7: Comparative Analysis of Layer Configurations

Configuration - Layer Position	CIFAR-10	ImageNet	Strengths	Limitations
Early Layers - Layers 1-3	Shows solid initial feature extraction with strong color and texture differentiation.	Benefits from early edge and color capture but lacks high-level feature extraction for complex textures, limiting performance in multi-object, multi-texture recognition.	Captures basic color and texture features; enhances initial illumination invariance; stable under minor rotation	Limited feature depth affects complex textures and rotated objects, especially on complex datasets
Middle Layers - Layers 3-6	Benefits from enhanced rotation and texture robustness, yielding higher retrieval rates.	Gains better pattern recognition but still faces limitations with complex rotations and high-detail images, affecting consistency in precision and recall.	Strengthens color separation and mid-level texture; improves rotation invariance and robustness	Still struggles with complex rotations and lighting variations, particularly for intricate ImageNet images
Combined Early and Middle Layers - Layers 1 - 3 & 3 -6	Achieves a balanced performance with improved texture recognition and moderate rotation variance handling.	Achieves higher precision, handling moderate illumination and texture changes better but still faces challenges with high-level feature complexity.	Combines foundational and refined features, yielding better color and texture invariance	Lacks high-level abstractions needed for more extensive rotations and lighting adjustments
Late Layers - Layers 6+	Achieves optimal fine-grained detail and illumination invariance, yielding the highest performance metrics in precision and recall.	Shows the best retrieval results, robustly handling diverse textures and orientations, though the computational demand is higher.	High precision and recall for rotation and illumination invariance; effective fine-grained texture capture	High computational cost and resource-intensive
All Layers - Layers 1+	Benefits from robust color and moderate rotation handling but at a high computational cost.	achieves the most comprehensive performance, effectively handling illumination and complex texture variations, suitable for high-demand applications through computationally intensive	Comprehensive feature extraction across all layers, maximizing retrieval performance on complex datasets	Computationally intensive, requiring significant resources

Previous works e.g., Li et al. (2021); Singh et al. (2024)), demonstrated that applying HSV color transformation in early CNN layers helps stabilize color consistency under varying lighting. Similar to these studies, our results confirmed that applying HSV in the early layers improves illumination invariance, especially in simpler datasets like CIFAR-10, with a significant boost in precision and recall. However, unlike their findings, we observed that combining HSV applications in both early and middle layers did not yield additional improvements, suggesting that illumination invariance is largely influenced by the initial stages of feature extraction.

Directional texture filters have been widely used for texture feature extraction in CBIR, as seen in the works of Pradhan et al. (2022) and Srivastava et al. (2023). These studies emphasize the importance of integrating directional texture in the middle or later layers for more detailed pattern recognition. Our findings, particularly on the ImageNet dataset, align with these studies, as middle-layer integration enhanced texture detail extraction. Additionally, integrating directional filters in the late layers achieved higher precision in complex image retrieval for ImageNet. However, in contrast to some prior work, our experiments indicate that early layer directional filtering alone may not be sufficient for high texture robustness.

Gabor filters have been extensively studied for their rotation-invariant properties in CBIR. Zhu et al. (2020) concluded that Gabor filters are effective when used across multiple layers. Our study, however, highlights that early and middle-layer integration of Gabor filters achieved better rotation invariance for CIFAR-10 without the computational burden observed with all-layer integration. Specifically, the middle and late-layer combination proved optimal for ImageNet, supporting findings that deeper-layer Gabor filtering is crucial for complex datasets with high rotational variation.

While individual layers (early, middle, or late) have been explored independently in prior research, few studies evaluated combined layer configurations. Our findings indicate that combining early and middle layers effectively balances performance and computational efficiency, especially for datasets like CIFAR-10. Similarly, combining middle and late layers for ImageNet aligns with Wang et al. (2023), who

observed that mid-to-late integration maximizes feature extraction and dataset adaptability. This combined approach yielded competitive results in recall and precision while optimizing computational resources, providing a more flexible solution for mixed-complexity CBIR applications.

In summary, this study expands on existing work by systematically analysing layer configurations and validating that combining specific layers yields distinct advantages across dataset complexities. The integration of HSV color space, directional texture filters, and Gabor filters across targeted layer combinations demonstrated performance gains that are both dataset-sensitive and computationally efficient, providing a comprehensive approach to CBIR that is adaptable to both simple and complex image datasets.

CHAPTER FIVE

CONCLUSION AND FURTHER WORK

5.1 Introduction

This chapter provides an overview of the key findings of this research on improving content-based image retrieval systems through rotational and illumination invariance. It also outlines potential directions for future studies to build on this work.

Research questions revisited

There were four research questions as presented in chapter one. In this section, we revisit each research question as follows.

i. What are the design limitations of Gabor Convolutional Neural Networks (GCNNs) in extracting color and texture for content-based image retrieval?

The study examined the design limitations of Gabor Convolutional Neural Networks (GCNNs) in extracting color and texture features for content-based image retrieval (CBIR). HSV color space has been demonstrated to be illumination invariance. This is due to its value component, which separates color information from brightness and makes it more resistant to changes in lighting conditions thus retrieving images that are consistence regardless of lighting condition changes. More details are listed in chapter two.

ii. What design techniques are used by Gabor Convolutional Neural Networks (GCNNs) to ensure rotational invariance for texture features in content-based image retrieval?

Gabor Convolutional Neural Networks (GCNNs) employ a directional filter technique to ensure rotational invariance for texture features in content-based image retrieval (CBIR). By integrating these filters into convolutional layers, GCNNs extract rotation-independent texture features effectively. The research observed directional texture filters capture the texture patterns and structures present in an

image which remain relatively stable and unchanged due to orientation. More details are found in chapter two.

iii. What design techniques of Gabor Convolutional Neural Networks (GCNNs) handle illumination invariance for color and texture-based features?

Integrating HSV color space, directional texture filters, and Gabor filters at different CNN layer configurations handles illumination invariance for color and texture-based features. More details are listed in chapter three.

Finally, chapter four reveals integrating both rotational and illumination invariance in Gabor Convolutional Neural Networks (GCNNs) improves the performance of a content-based image retrieval system. Early layers improve basic color and texture differentiation but struggle with complex patterns, while middle layers enhance texture robustness and moderate rotation handling. The combined early and middle layer configuration balances foundational and refined feature extraction, enhancing performance without extreme computational demands. Late layers provide the highest retrieval accuracy, capturing fine-grained details and showing strong resilience to complex rotations and illumination changes, especially effective for complex datasets like ImageNet. Although applying these filters across all layers maximizes color, texture, and rotational invariance, it comes at a high computational cost, making layer-specific integration a promising approach for optimizing CBIR performance based on dataset complexity. More details are listed in chapter four.

In conclusion, the study demonstrated that integrating HSV color, texture filter, and GCNN in all layers offers both rotation and illumination invariance and a robust CBIR performance.

5.2 Future Work

In this study rotation and illumination invariance were investigated to improve the performance of CBIR. However, we would recommend further research on experimenting with occlusion, and complexity to improve the performance.

The phenomenon known as "occlusion" occurs when an image or a portion of an image is entirely or partially obscured by another image or the surrounding environment. Occlusion poses challenges in image retrieval because it obscures parts of the scene as the image rotates, hence leading to incomplete or misleading images.

The Extended GCNN model demonstrates that its complexity is not exponential. However, we recommend further investigations as part of future work.

REFERENCES

Abdullah, S., Zamani, M., & Demosthenous, A. (2022). A compact CNN-based speech enhancement with adaptive filter design using gabor function and region-aware convolution. *IEEE Access, 10*, 130657-130671.

Achouri, K., & Martin, O. J. (2021). Fundamental properties and classification of polarization converting bianisotropic metasurfaces. *IEEE Transactions on Antennas and Propagation*, *69*(9), 5653-5663.

Afaq, S., & Rao, S. (2020). Significance of epochs on training a neural network. *Int. J. Sci. Technol. Res, 9*(06), 485-488.

Afifi, M., Brubaker, M. A., & Brown, M. S. (2021). Histogan: Controlling colors of gan-generated and Real images via color histograms. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7941-7950).

Aggarwal, A. K. (2022). Learning texture features from glcm for classification of brain tumor mri images using random forest classifier. *Transactions on Signal Processing*, *18*, 60-63.

Ahmed, R., Mehmood, A., Rahman, M. M. U., & Dobre, O. A. (2023). A Deep Learning and Fast Wavelet Transform-Based Hybrid Approach for Denoising of PPG Signals. *IEEE sensors letters*, *7*(7), 1-4.

Akujuobi, C. M. (2022). *Wavelets and wavelet transform systems and their applications*. Berlin/Heidelberg, Germany: Springer International Publishing.

Almakady, Y., Mahmoodi, S., Conway, J., & Bennett, M. (2020). Rotation invariant features based on three dimensional Gaussian Markov random fields for volumetric texture classification. *Computer Vision and Image Understanding*, *194*, 102931.

Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2021). Applications of generative adversarial networks (gans): An updated review. *Archives of Computational Methods in Engineering*, *28*, 525-552.

Alrahhal, M., & Supreethi, K. P. (2024). Enhancing image retrieval accuracy through multi-resolution HSV-LNP feature fusion and modified K-NN relevance feedback. *International Journal of Information Technology*, 1-15.

Al-saleem, R. M., Al-Hilali, B. M., & Abboud, I. K. (2020). Mathematical representation of color spaces and its role in communication systems. *Journal of Applied Mathematics*, *2020*(1), 4640175.

Alsmadi, M. K. (2020). Content-based image retrieval using color, shape and texture descriptors and features. *Arabian Journal for Science and Engineering*, *45*(4), 3317-3330.

Amiri, A., Arabi, H., & Kermani, M. (2020). A comprehensive survey on color normalization techniques for image analysis in computational pathology. *Journal of Medical Signals and Sensors*, *10*(3), 163-172.

Anand, J., Aasish, C., Narayanan, S. S., & Ahmed, R. A. (2023). *Drones for Disaster Response and Management. In Internet of Drones (pp. 177-200*). New York: CRC Press.

Andrearczyk, V., Fageot, J., Oreiller, V., Montet, X., & Depeursinge, A. (2020). Local rotation invariance in 3D CNNs. *Medical image analysis*, *65*, 101756.

Anne, M. E., Brindha, M., & Sivakumaran, N. (2024, June). Dual Variational Generation Framework for Enhanced Cross-Modal Face Recognition. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

Anowar Hossain, M. (2023). Simulation of chromatic and achromatic assessments for camouflage textiles and combat background. *The Journal of Defense Modeling and Simulation*, *20*(3), 317-332.

Ansari, M. A., & Singh, D. K. (2022). Significance of color spaces and their selection for image processing: a survey. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 15*(7), 946-956.

Archana, R., & Jeevaraj, P. E. (2024). Deep learning models for digital image processing: a review. *Artificial Intelligence Review, 57*(1), 11.

Arfaoui, S., Mabrouk, A. B., & Cattani, C. (2021). *Wavelet Analysis: Basic Concepts and Applications.* New York: Chapman and hall/CRC.

Asma, Z., & Brahim, N. (2022). Local Binary Pattern Regrouping for Rotation Invariant Texture Classification. *Journal of Information Technology Research (JITR), 15*(1), 1-15.

Badar, M., Haris, M., & Fatima, A. (2020). Application of deep learning for retinal image analysis: A review. *Computer Science Review, 35,* 100203.

Bagewadi, S., & Veerashetty, S. (2023). Enhancing Texture Classification through GLCM and LBP Feature Extraction with Modified PCA Dimensionality Reduction. In 2023 4th IEEE Global Conference for Advancement in Technology (GCAT) (pp. 1-8). IEEE.

Bansal, M., Kumar, M., & Kumar, M. (2021). 2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors. *Multimedia Tools and Applications, 80*(12), 18839-18857.

Barbero-Álvarez, M. A., Rodrigo, J. A., & Menéndez, J. M. (2023). Minimum error adaptive RGB calibration in a context of colorimetric uncertainty for cultural heritage preservation. *Computer Vision and Image Understanding, 237,* 103835.

Barburiceanu, S., Terebes, R., & Meza, S. (2021). 3D texture feature extraction and classification using GLCM and LBP-based descriptors. *Applied Sciences, 11*(5), 2332.

Basak, J. K., Madhavi, B. G. K., Paudel, B., Kim, N. E., & Kim, H. T. (2022). Prediction of total soluble solids and pH of strawberry fruits using RGB, HSV and HSL colour spaces and machine learning models. *Foods, 11*(14), 2086.

Basar, S., Ali, M., Ochoa-Ruiz, G., Zareei, M., Waheed, A., & Adnan, A. (2020). Unsupervised color image segmentation: A case of RGB histogram based K-means clustering initialization. *Plos one, 15*(10), e0240015.

Berthier, M., Garcin, V., Prencipe, N., & Provenzi, E. (2021). The relativity of color perception. *Journal of Mathematical Psychology, 103*, 102562.

Bhargava, N., Bhargava, R., Rathore, P. S., & Kumar, A. (2020). Texture recognition using gabor filter for extracting feature vectors with the regression mining algorithm. *International Journal of Risk and Contingency Management (IJRCM), 9*(3), 31-44.

Bhattarai, B., Subedi, R., Gaire, R. R., Vazquez, E., & Stoyanov, D. (2023). Histogram of oriented gradients meet deep learning: A novel multi-task deep network for 2D surgical image semantic segmentation. *Medical Image Analysis, 85*, 102747.

Bhunia, A. K., Bhattacharyya, A., Banerjee, P., Roy, P. P., & 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*, 703-723.

Bianconi, F., Kather, J. N., & Reyes-Aldasoro, C. C. (2020). Experimental assessment of color deconvolution and color normalization for automated classification of histology images stained with hematoxylin and eosin. *Cancers, 12*(11), 3337.

Bottenus, N., Byram, B. C., & Hyun, D. (2020). Histogram matching for visual ultrasound image comparison. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control, 68*(5), 1487-1495.

Burambekova, A., & Shamoi, P. (2024). Comparative analysis of color models for human perception and visual color difference. *arXiv preprint arXiv*:2406.19520.

Burger, W., & Burge, M. J. (2022). *Scale-invariant feature transform (SIFT). In Digital Image Processing: An Algorithmic Introduction (pp. 709-763*). Cham: Springer International Publishing.

Burger, W., & Burge, M. J. (2022). *Digital image processing: An algorithmic introduction*. Switzerland: Springer Nature.

Caron, M., Houlsby, N., & Schmid, C. (2024). *Location-aware self-supervised transformers for semantic segmentation.* In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 117-127).

Chaki, J., & Dey, N. (2020). *Image color feature extraction techniques: fundamentals and applications.* Singapore: Springer Nature.

Chaki, J., & Dey, N. (2020). Texture feature extraction techniques for image recognition. Springer Singapore: Springer.

Chang, Y., & Mukai, N. (2022). Color feature based dominant color extraction. *IEEE Access, 10,* 93055-93061.

Chen, L., Rottensteiner, F., & Heipke, C. (2021). Feature detection and description for image matching: from hand-crafted design to deep learning. *Geo-spatial Information Science, 24*(1), 58-74.

Chen, Y., Liu, L., Tao, J., Chen, X., Xia, R., Zhang, Q., ... & Xie, J. (2021). The image annotation algorithm using convolutional features from intermediate layer of deep learning. *Multimedia Tools and Applications, 80,* 4237-4261.

Chen, Y., Liu, L., Tao, J., Xia, R., Zhang, Q., Yang, K., ... & Chen, X. (2021). The improved image inpainting algorithm via encoder and similarity constraint. *The Visual Computer, 37,* 1691-1705.

Chi, J., Liu, Y., Wang, V., & Yan, J. (2022). Performance Analysis of Three kinds of Neural Networks in the Classification of Mask Images. In Journal of Physics: Conference Series, 2181(1), 012032). IOP Publishing.

Chicco, D. (2021). Siamese neural networks: An overview. *Artificial neural networks*, 73-94.

Choe, J., Hwang, H. J., Seo, J. B., Lee, S. M., Yun, J., Kim, M. J. ... & Kim, B. (2022). Content-based image retrieval by using deep learning for interstitial lung disease diagnosis with chest CT. *Radiology*, *302*(1), 187-197.

Cohen, T. (2021). *Equivariant convolutional networks*, Unpublished PhD dissertation, Amsterdam: Universiteit van Amsterdam

Dakshayani, V., Locharla, G. R., Pławiak, P., Datti, V., & Karri, C. (2022). Design of Gabor Filter Based Image Denoising Hardware Model. *Electronics 2022*, 11, 1063.

Danlami, M., Jamel, S., Ramli, S. N., & Azahari, S. R. M. (2020). Comparing the legendre wavelet filter and the Gabor wavelet filter for feature extraction based on Iris recognition system. In 2020 IEEE 6th International Conference on Optimization and Applications (ICOA) (pp. 1-6). IEEE.

Daugman, J. (2023). Understanding Biometric Entropy and Iris Capacity: Avoiding Identity Collisions on National Scales. *arXiv preprint arXiv:*2308.03189.

Deng, W., & Zheng, L. (2021). Are labels always necessary for classifier accuracy evaluation?. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 15069-15078).

Dewi, C., Chen, R. C., Yu, H., & Jiang, X. (2023). Robust detection method for improving small traffic sign recognition based on spatial pyramid pooling. *Journal of Ambient Intelligence and Humanized Computing, 14*(7), 8135-8152.

Dey, A. U., Ghosh, S. K., Valveny, E., & Harit, G. (2021). Beyond visual semantics: Exploring the role of scene text in image understanding. *Pattern Recognition Letters*, *149*, 164-171.

Dhakshayani, J., & Surendiran, B. (2023). GF-CNN: An Enhanced Deep Learning Model with Gabor Filters for Maize Disease Classification. *SN Computer Science*, *4*(5), 538.

Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, 9(2), 85-112.

Ding, X., Wang, Z., Hu, G., Liu, J., Zhang, K., Li, H., ... & Qiu, C. W. (2020). Metasurface holographic image projection based on mathematical properties of Fourier transform. *PhotoniX*, *1*, 1-12.

Dowerah, R., & Patel, S. (2024). Comparative analysis of color histogram and LBP in CBIR systems. *Multimedia Tools and Applications*, *83*(5), 12467-12486.

Ebner, M. (2021). *Color constancy. In Computer Vision: A Reference Guide* (pp. 168-175). Cham: Springer International Publishing.

Fan, S., Zhang, P., Meng, Y., Liu, H., Luo, Z., & Jin, S. (2024). Ratcheting damage evaluation based on critically refracted longitudinal wave combined with recurrence quantification analysis. *Measurement*, 115970.

Fan, Z., Pi, Y., Wang, M., Kang, Y., & Tan, K. (2024). GLS–MIFT: A modality invariant feature transform with global-to-local searching. *Information Fusion*, *105*, 102252.

Fang, M. H., Bao, Z., Huang, W. T., & Liu, R. S. (2022). Evolutionary generation of phosphor materials and their progress in future applications for light-emitting diodes. *Chemical Reviews*, *122*(13), 11474-11513.

Gao, L., Du, Y., Li, H., & Lin, G. (2022). RotEqNet: Rotation-equivariant network for fluid systems with symmetric high-order tensors. *Journal of Computational Physics, 461*, 111205.

Garcia, H., Correa, C. V., & Arguello, H. (2020). Optimized sensing matrix for single pixel multi-resolution compressive spectral imaging. *IEEE Transactions on Image Processing, 29*, 4243-4253.

Gasmi, K., Aouadi, H., & Torjmen, M. (2023). Link-Driven Study to Enhance Text-Based Image Retrieval: Implicit Links vs. Explicit Links. *IEEE Access, 11*, 90526- 90537

Gautam, S., & Singhai, J. (2024, February). Enhanced Water-body Texture Extraction from GF-2 Satellite Imagery Using Transfer Learning Approach with Gray Level Difference Matrix. In 2024 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) (pp. 1-6). IEEE.

Ghalati, M. K., Nunes, A., Ferreira, H., Serranho, P., & Bernardes, R. (2021). Texture analysis and its applications in biomedical imaging: A survey. *IEEE Reviews in Biomedical Engineering, 15*, 222-246.

Gholamalinezhad, H., & Khosravi, H. (2020). Pooling methods in deep neural networks, a review. *arXiv preprint arXiv:2009*.07485.

Ghosh, S., & Banerjee, A. (2020). A comprehensive review on color normalization in medical imaging. *Journal of Medical Imaging and Health Informatics, 10*(6), 1305-1319.

Goldwasser, S. (2024). Imagining as a skillful mental action. *Synthese, 204*(2), 38.

Gonçalves Dias Diniz, P. H. (2020). Chemometrics-assisted color histogram-based analytical systems. *Journal of Chemometrics, 34*(12), e3242.

Graham, S., Epstein, D., & Rajpoot, N. (2020). Dense steerable filter cnns for exploiting rotational symmetry in histology images. *IEEE Transactions on Medical Imaging*, *39*(12), 4124-4136.

Guo, T., Zhang, T., Lim, E., Lopez-Benitez, M., Ma, F., & Yu, L. (2022). A review of wavelet analysis and its applications: Challenges and opportunities. *IEEE Access*, *10*, 58869-58903.

Gupta, S., Thakur, K., & Kumar, M. (2021). 2D-human face recognition using SIFT and SURF descriptors of face's feature regions. *The Visual Computer*, *37*(3), 447-456.

Hadid, M. H., Hussein, Q. M., Al-Qaysi, Z. T., Ahmed, M. A., & Salih, M. M. (2023). An overview of content-based image retrieval methods and techniques. *Iraqi Journal for Computer Science and Mathematics*, *4*(3), 66-78.

Hameed, I. M., Abdulhussain, S. H., & Mahmmod, B. M. (2021). Content-based image retrieval: A review of recent trends. *Cogent Engineering*, *8*(1), 1927469.

Hao, Q., Tang, M., Cao, J., Rizvi, S., Cui, H., & Wu, Z. (2020). Study on Rotation and Scaling Invariance of Retina-Like Imaging. *IEEE Photonics Journal*, *12*(5), 1-10.

Hao, Y., Hu, P., Li, S., Udupa, J. K., Tong, Y., & Li, H. (2022). Gradient-Aligned convolution neural network. *Pattern Recognition*, *122*, 108354.

Hassan, E. K., & Saud, J. H. (2023, February). HSV color model and logical filter for human skin detection. *In AIP Conference Proceedings* (Vol. 2457, No. 1). AIP Publishing.

Hassan, M. F., Adam, T., Rajagopal, H., & Paramesran, R. (2023). A hue preserving uniform illumination image enhancement via triangle similarity criterion in HSI color space. *The Visual Computer*, *39*(12), 6755-6766.

Hassan, S., & Dhimish, M. (2023). Dual spin max pooling convolutional neural network for solar cell crack detection. *Scientific reports*, *13*(1), 11099.

Hong, T. P., & Guan, L. (2021). A scale and rotational invariant key-point detector based on sparse coding. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *12*(3), 1-19.

Hossein-Nejad, Z., & Nasri, M. (2024). A new hybrid strategy in medical image registration based on graph transformation matching and mean-based RANSAC algorithms. *Multimedia Tools and Applications*, 1-28.

Hu, C. H., Yu, J., Wu, F., Zhang, Y., Jing, X. Y., Lu, X. B., & Liu, P. (2021). Face illumination recovery for the deep learning feature under severe illumination variations. *Pattern Recognition*, *111*, 107724.

Hu, X., Wu, G., Xie, Z., & Liu, F. (2020). Position-based intensity change model for illumination variation in digital image correlation. *Optical Engineering*, *59*(2), 024102-024102.

Hu, Z., Qi, W., Ding, K., Liu, G., & Zhao, Y. (2024). An adaptive lighting indoor vSLAM with limited on-device resources. *IEEE Internet of Things Journal*, *11*(17), 28863 – 28875.

Huang, C., Li, J., & Gao, G. (2023). Review of quaternion-based color image processing methods. *Mathematics*, *11*(9), 2056.

Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., & Shao, L. (2023). Normalization techniques in training dnns: Methodology, analysis and application. *IEEE transactions on pattern analysis and machine intelligence*, *45*(8), 10173-10196.

IIqbal, N., Mumtaz, R., Shafi, U., & Zaidi, S. M. H. (2021). Gray level co-occurrence matrix (GLCM) texture based crop classification using low altitude remote sensing platforms. *PeerJ Computer Science*, *7*, e536.

Jemal, M. A., Sallami, M. M., & Ghorbel, F. (2024). Robust watermarking method based on the analytical Clifford Fourier Mellin transform. *Multimedia Tools and Applications, 83*(9), 25901-25922.

Jha, K., Srivastava, S., & Jain, A. (2024). A novel texture based approach for facial liveness detection and authentication using deep learning classifier. *International Journal of Computational and Experimental Science and Engineering, 10*(3).

Jia, X., Zhu, C., Li, M., Tang, W., & Zhou, W. (2021). LLVIP: A visible-infrared paired dataset for low-light vision. In Proceedings of the IEEE/CVF *international conference on computer vision* (pp. 3496-3504).

Kamiyama, M., & Taguchi, A. (2021). Color conversion formula with saturation correction from HSI color space to RGB color space. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 104*(7), 1000-1005.

Kaplan, K., Kaya, Y., Kuncan, M., Minaz, M. R., & Ertunç, H. M. (2020). An improved feature extraction method using texture analysis with LBP for bearing fault diagnosis. *Applied Soft Computing, 87*, 106019.

Karanwal, S., & Diwakar, M. (2022). Improved ELBP descriptors for face recognition. *International Journal of Computational Science and Engineering, 25*(2), 198-210.

Kaur, M. (2023). A Comparative Analysis of Local Binary Pattern (LBP) Variants for Image Tamper Detection. Retrieved from https://www.researchsquare.com/article/rs-3608580/v1

Kawano, M., Kumagai, W., Sannai, A., Iwasawa, Y., & Matsuo, Y. (2021). Group equivariant conditional neural processes. *arXiv preprint arXiv:2102*.08759.

Kayhan, N., & Fekri-Ershad, S. (2021). Content based image retrieval based on weighted fusion of texture and color features derived from modified local

binary patterns and local neighborhood difference patterns. *Multimedia Tools and Applications, 80*(21), 32763-32790.

Kayhan, O. S., & Gemert, J. C. V. (2020). On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In Proceedings of the IEEE/CVF *Conference on Computer Vision and Pattern Recognition* (pp. 14274-14285).

Keyvanpour, M. R., Vahidian, S., & Mirzakhani, Z. (2021). An analytical review of texture feature extraction approaches. *International Journal of Computer Applications in Technology, 65*(2), 118-133.

Khalifa, N. E., Loey, M., & Mirjalili, S. (2022). A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial Intelligence Review, 55*(3), 2351-2377.

Kim, W., Kanezaki, A., & Tanaka, M. (2020). Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing, 29*, 8055-8068.

Klaib, A. F., Alsrehin, N. O., Melhem, W. Y., Bashtawi, H. O., & Magableh, A. A. (2021). Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies. *Expert Systems with Applications, 166*, 114037.

Kociołek, M., Kozłowski, M., & Cardone, A. (2022). A convolutional neural networks-based approach for texture directionality detection. *Sensors*, *22*(2), 562.

Koscevic, K., Subasic, M., & Loncaric, S. (2020). Deep learning-based illumination estimation using light source classification. *IEEE Access, 8*, 84239-84247.

Lei, Y., Phung, S. L., Bouzerdoum, A., Le, H. T., & Luu, K. (2022). Pedestrian lane detection for assistive navigation of vision-impaired people: Survey and experimental evaluation. *IEEE Access, 10*, 101071-101089.

Lengyel, A., Garg, S., Milford, M., & van Gemert, J. C. (2021). Zero-shot day-night domain adaptation with a physics prior. In Proceedings of the IEEE/CVF *International Conference on Computer Vision* (pp. 4399-4409).

Li, J., Xu, W., Shi, P., Zhang, Y., & Hu, Q. (2022). LNIFT: Locally normalized image for rotation invariant multimodal feature matching. *IEEE Transactions on Geoscience and Remote Sensing, 60*, 1-14.

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems, 33*(12), 6999-7019.

Liang, L., Liu, C., & Wang, J. (2023). Periodicity measure of cyclo-stationary impulses based on low sparsity of Gini index and its application to bearing diagnosis. *ISA transactions, 138*, 611-627.

Liu, G. H., & Yang, J. Y. (2021). Deep-seated features histogram: a novel image retrieval method. *Pattern Recognition, 116*, 107926.

Liu, G. H., & Yang, J. Y. (2023). Exploiting deep textures for image retrieval. *International Journal of Machine Learning and Cybernetics, 14*(2), 483-494.

Liu, G. H., Li, Z. Y., Yang, J. Y., & Zhang, D. (2024). Exploiting sublimated deep features for image retrieval. *Pattern Recognition, 147*, 110076.

Liu, Y., Chen, X., Wu, S., & Liu, X. (2020). A comprehensive survey of color normalization in computational pathology. *Frontiers in Bioengineering and Biotechnology, 8*, 612.

Liu, Y., Gong, X., Chen, J., Chen, S., & Yang, Y. (2020). Rotation-invariant siamese network for low-altitude remote-sensing image registration. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13*, 5746-5758.

Lu, T., Ding, K., Fu, W., Li, S., & Guo, A. (2023). Coupled adversarial learning for fusion classification of hyperspectral and LiDAR data. *Information Fusion, 93*, 118-131.

Lu, Y. F., Gao, J. W., Yu, Q., Li, Y., Lv, Y. S., & Qiao, H. (2023). A Cross-Scale and Illumination Invariance-Based Model for Robust Object Detection in Traffic Surveillance Scenarios. *IEEE Transactions on Intelligent Transportation Systems, 24*(7), 6989-6999..

Luo, S., Zeng, J., & Liu, J. (2023, May). Research on Traffic Signal Light Recognition Method in Complex Scene. In 2023 IEEE 3rd *International Conference on Information Technology, Big Data and Artificial Intelligence* (ICIBA) (Vol. 3, pp. 899-903). IEEE.

Ma, J., Jiang, X., Fan, A., Jiang, J., & Yan, J. (2021). Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision, 129*(1), 23-79.

Ma, S., Gao, J., Wang, R., Chang, J., Mao, Q., Huang, Z., & Jia, C. (2023). Overview of intelligent video coding: from model-based to learning-based approaches. *Visual Intelligence, 1*(1), 15.

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings, 3*(1), 91-99.

Marasco, E., & Vurity, A. (2022). Late deep fusion of color spaces to enhance finger photo presentation attack detection in smartphones. *Applied Sciences, 12*(22), 11409.

Marlow, P. J., Gegenfurtner, K. R., & Anderson, B. L. (2022). The role of color in the perception of three-dimensional shape. *Current Biology, 32*(6), 1387-1394.

Marques, B. A. D., Clua, E. W. G., Montenegro, A. A., & Vasconcelos, C. N. (2022). Spatially and color consistent environment lighting estimation using deep neural networks for mixed reality. *Computers & Graphics, 102*, 257-268.

Mehta, R., Ujjwal, G., Shilpa, S. J., Vityazev, S., & Singh, K. K. (2023). *Rotation invariant 2D ear recognition using gabor filters and ensemble of pre-trained deep convolutional neural network model.* In 2023 25th International Conference on Digital Signal Processing and its Applications (DSPA) (pp. 1-6). IEEE.

Mei, S., Jiang, R., Ma, M., & Song, C. (2023). Rotation-invariant feature learning via convolutional neural network with cyclic polar coordinates convolutional layer. *IEEE Transactions on Geoscience and Remote Sensing*, *61*, 1-13.

Mishra, R. K., Urolagin, S., Jothi, J. A. A., & Gaur, P. (2022). Deep hybrid learning for facial expression binary classifications and predictions. *Image and Vision Computing*, *128*, 104573.

Mo, H., & Zhao, G. (2022). RIC-CNN: rotation-invariant coordinate convolutional neural network. *arXiv preprint arXiv:2211*.11812.

Mo, H., & Zhao, G. (2024). Sorting Convolution Operation for Achieving Rotational Invariance. *IEEE Signal Processing Letters*. *31*, 1199-1203.

Morgenstern, Y., Hartmann, F., Schmidt, F., Tiedemann, H., Prokott, E., Maiello, G., & Fleming, R. W. (2021). An image-computable model of human visual shape similarity. *PLoS computational biology*, *17*(6), e1008981.

Mufarroha, F. A., Anamisa, D. R., & Hapsani, A. G. (2020). Content based image retrieval using two color feature extraction. *In Journal of physics: conference series*, *1569*(3), 032072.

Mumuni, A., & Mumuni, F. (2021). CNN architectures for geometric transformation-invariant feature representation in computer vision: a review. *SN Computer Science*, *2*(5), 340.

Mustaqim, T., Tsaniya, H., Adhiyaksa, F. A., & Suciati, N. (2022, August). Wavelet Transformation and Local Binary Pattern for Data Augmentation in Deep

Learning-based Face Recognition. In 2022 10th *International Conference on Information and Communication Technology* (ICoICT) (pp. 362-367). IEEE.

Muzaffar, A. W., Riaz, F., Abuain, T., Abu-Ain, W. A. K., Hussain, F., Farooq, M. U., & Azad, M. A. (2023). Gabor contrast patterns: A novel framework to extract features from texture images. *IEEE Access*, *11*, 60324-60334.

Nain, S., Mittal, N., & Hanmandlu, M. (2024). CNN-based plant disease recognition using colour space models. *International Journal of Image and Data Fusion*, 1-14.

Najjar, F. H., Khudhair, K. T., Khaleq, A. H. A., Kadhim, O. N., Abedi, F., & Al-Kharsan, I. H. (2022, May). Histogram features extraction for edge detection approach. In 2022 5th International Conference on Engineering Technology and its Applications (IICETA) (pp. 373-378). IEEE.

Nirthika, R., Manivannan, S., Ramanan, A., & Wang, R. (2022). Pooling in convolutional neural networks for medical image analysis: a survey and an empirical study. *Neural Computing and Applications*, *34*(7), 5321-5347.

Paoletti, M. E., Haut, J. M., Roy, S. K., & Hendrix, E. M. (2020). Rotation equivariant convolutional neural networks for hyperspectral image classification. *IEEE Access*, *8*, 179575-179591.

Phuangsaijai, N., Jakmunee, J., & Kittiwachana, S. (2021). Investigation into the predictive performance of colorimetric sensor strips using RGB, CMYK, HSV, and CIELAB coupled with various data preprocessing methods: a case study on an analysis of water quality parameters. *Journal of Analytical Science and Technology*, *12*, 1-16.

Phuangsaijai, N., Jakmunee, J., & Kittiwachana, S. (2021). Investigation into the predictive performance of colorimetric sensor strips using RGB, CMYK, HSV, and CIELAB coupled with various data preprocessing methods: a case study on an analysis of water quality parameters. *Journal of Analytical Science and Technology*, *12*, 1-16.

Poloni, K. M., de Oliveira, I. A. D., Tam, R., Ferrari, R. J., & Alzheimer's Disease Neuroimaging Initiative. (2021). Brain MR image classification for Alzheimer's disease diagnosis using structural hippocampal asymmetrical attributes from directional 3-D log-Gabor filter responses. *Neurocomputing, 419*, 126-135.

Pore, A., Li, Z., Dall'Alba, D., Hernansanz, A., De Momi, E., Menciassi, A., ... & Vander Poorten, E. (2023). Autonomous navigation for robot-assisted intraluminal and endovascular procedures: A systematic review. *IEEE Transactions on Robotics, 39*(4), 2529-2548.

Prabhu, A., & Lakshmi, S. (2021, May). Identification and yield estimation of mature fruits using modified watershed algorithm. In 2021 2nd International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.

Pradhan, J., Pal, A. K., & Banka, H. (2022). A CBIR system based on saliency driven local image features and multi orientation texture features. *Journal of Visual Communication and Image Representation, 83*, 103396.

Praveen, P., & Vasu, S. (2020). A comprehensive survey on color normalization in whole slide histopathological images. *International Journal of Biomedical Imaging, 2020.*

Qi, F., Yang, X., & Xu, C. (2020). Emotion knowledge driven video highlight detection. *IEEE Transactions on Multimedia, 23*, 3999-4013.

Quach, M. D., Vo, D. M., & Pham, H. A. (2024). MS-MixVPR: Multi-scale Feature Mixing Approach for Long-Term Place Recognition. *SN Computer Science, 5*(6), 656.

Quan, Y., Zhang, D., Zhang, L., & Tang, J. (2023). Centralized feature pyramid for object detection. *IEEE Transactions on Image Processing. 62*, 1-14.

Quiroga, F., Ronchetti, F., Lanzarini, L., & Bariviera, A. F. (2020). Revisiting data augmentation for rotational invariance in convolutional neural networks.

In Modelling and Simulation in Management Sciences: Proceedings of the International Conference on Modelling and Simulation in Management Sciences (MS-18) (pp. 127-141). Springer International Publishing.

Raj, E. F. I., & Balaji, M. (2023). *Shape Feature Extraction Techniques for Computer Vision Applications.* In Smart Computer Vision (pp. 81-102). Cham: Springer International Publishing.

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

Ramola, A., Shakya, A. K., & Van Pham, D. (2020). Study of statistical methods for texture analysis and their modern evolutions. *Engineering Reports*, *2*(4), e12149.

Rao, B. S. (2020). Dynamic histogram equalization for contrast enhancement for digital images. *Applied Soft Computing*, *89*, 106114.

Rasool, M., & Kaur, A. (2021). A Novel Rotation Invariant Descriptor for Texture Classification with Local Binary Patterns. *In Soft Computing and Signal Processing: Proceedings* of 3rd ICSCSP 2020, Volume 1 (pp. 385-396). Springer Singapore.

Ren, Z., Fang, F., Yan, N., & Wu, Y. (2022). State of the art in defect detection based on machine vision. *International Journal of Precision Engineering and Manufacturing-Green Technology*, *9*(2), 661-691.

Rimiru, R. M., Gateri, J., & Kimwele, M. W. (2022). GaborNet: investigating the importance of color space, scale and orientation for image classification. *PeerJ Computer Science*, *8*, e890.

Rodríguez, R. G., Vazquez-Corral, J., & Bertalmío, M. (2020). Color matching images with unknown non-linear encodings. *IEEE Transactions on Image Processing*, *29*, 4435-4444.

Roy, S., Bhalla, K., & Patel, R. (2024). Mathematical analysis of histogram equalization techniques for medical image enhancement: a tutorial from the perspective of data loss. *Multimedia Tools and Applications*, *83*(5), 14363-14392.

Ruby, A. U., Chaithanya, B. N., TJ, S. J., Darandale, S., Kerenalli, S., & Patil, R. (2022). An effective feature descriptor method to classify plant leaf diseases using eXtreme Gradient Boost. *Journal of Integrated Science and Technology*, *10*(1), 43-52.

Salas, R. R. (2021). Reduced CNN Model for Rotation-Invariant Classification (Doctoral dissertation, Université Gustave Eiffel).

Sarker, A., & Grift, T. E. (2023). Monitoring Postharvest Color Changes and Damage Progression of Cucumbers Using Machine Vision. *Journal of Food Research*, *12*(2), 37-50.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, *2*(3), 160.

Sato, T. (2021). TXI: Texture and color enhancement imaging for endoscopic image enhancement. *Journal of Healthcare Engineering*, *2021*(1), 5518948.

Schütz, M., Kerbl, B., & Wimmer, M. (2021). Rendering point clouds with compute shaders and vertex order optimization. *In Computer Graphics Forum*, *40*(4), 115-126.

Selvi, A., & Thilagamani, S. (2023). Scale Invariant Feature Transform with Crow Optimization for Breast Cancer Detection. *Intelligent Automation & Soft Computing*, *36*(3).

Sethy, P.K., Barpanda, N.K., Rath, A.K., & Behera, S.K. (2020). Image processing techniques for diagnosing rice plant disease: a survey. *Procedia Computer Science, 167*, 516-530.

Shahbaz, M., Trabelsi, N., Tiwari, A. K., Abakah, E. J. A., & Jiao, Z. (2021). Relationship between green investments, energy markets, and stock markets in the aftermath of the global financial crisis. *Energy Economics, 104*, 105655.

Shamey, R. (Ed.). (2023). *Encyclopedia of color science and technology*. Cham: Springer International Publishing.

Shariaty, F., Orooji, M., Velichko, E.N., & Zavjalov, S.V. (2022). Texture appearance model, a new model-based segmentation paradigm, application on the segmentation of lung nodule in the CT scan of the chest. *Computers in biology and medicine, 140*, 105086.

Shen, C., Rawls, H. R., & Esquivel-Upshaw, J. F. (Eds.). (2021). *Phillips' Science of Dental Materials E-Book: Phillips' Science of Dental Materials E-Book*. India: Elsevier Health Sciences.

Singh, P. K., Mani, H., Maurya, M. K., & Kumar, M. (2024, June). A Novel Approach for Crop Disease Detection using MCCNNs and Decorrelating Color Spaces. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.

Smets, B. M., Portegies, J., Bekkers, E. J., & Duits, R. (2023). PDE-based group equivariant convolutional neural networks. *Journal of Mathematical Imaging and Vision*, *65*(1), 209-239.

Soni, V., Sharma, A., & Rajpurohit, J. (2024). A Swift Algorithm and Hue Preserving Based Mechanism for Underwater Image Colour Enhancement. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(1), 11-27.

Srivastava, D., Singh, S. S., Rajitha, B., Verma, M., Kaur, M., & Lee, H. N. (2023). Content-based Image Retrieval: A Survey on Local and Global Features Selection, Extraction, Representation and Evaluation Parameters. *IEEE Access. 11,* 95410-95431, doi: 10.1109/ACCESS.2023.3308911.

Srivastava, D., Singh, S. S., Rajitha, B., Verma, M., Kaur, M., & Lee, H. N. (2023). Content-based Image Retrieval: A Survey on Local and Global Features Selection, Extraction, Representation and Evaluation Parameters. IEEE Access.

Steinmann, R., Seydoux, L., Beaucé, E., & Campillo, M. (2022). Hierarchical exploration of continuous seismograms with unsupervised learning. *Journal of Geophysical Research: Solid Earth*, *127*(1), e2021JB022455.

Sueeprasan, S. (2023). *CIE Guidelines for Mixed Mode Illumination: Summary and Related Work*. In Encyclopedia of Color Science and Technology (pp. 198-201). Cham: Springer International Publishing.

Sugimoto, Y., & Imaizumi, S. (2023). Reversible image processing for color images with flexible control. *Applied Sciences*, *13*(4), 2297.

Sungheetha, A., & Sharma, R. (2021). Design an early detection and classification for diabetic retinopathy by deep feature extraction based convolution neural network. *Journal of Trends in Computer Science and Smart technology (TCSST)*, *3*(02), 81-94.

Swiderska-Chadaj, Z., de Bel, T., Blanchet, L., Baidoshvili, A., Vossen, D., van der Laak, J., & Litjens, G. (2020). Impact of rescanning and normalization on convolutional neural network performance in multi-center, whole-slide classification of prostate cancer. *Scientific Reports*, *10*(1), 14398.

Tabassum, F., Islam, M. I., & Amin, M. R. (2021). Comparison of filter banks of DWT in recovery of image using one dimensional signal vector. *Journal of King Saud University-Computer and Information Sciences*, *33*(5), 542-551.

Tajjour, S., Garg, S., Chandel, S. S., & Sharma, D. (2023). A novel hybrid artificial neural network technique for the early skin cancer diagnosis using color space conversions of original images. *International Journal of Imaging Systems and Technology*, *33*(1), 276-286.

Tamatjita, E. N., & Sihite, R. D. (2022). Banana Ripeness Classification using HSV Colour Space and Nearest Centroid Classifier. *Information Engineering Express*, *8*(1).

Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics*, *8*(6), 460-473.

Tan, Y. S., Lim, K. M., Tee, C., Lee, C. P., & Low, C. Y. (2021). Convolutional neural network with spatial pyramid pooling for hand gesture recognition. *Neural Computing and Applications*, *33*, 5339-5351.

Tang, B., Chen, L., Sun, W., & Lin, Z. K. (2023). Review of surface defect detection of steel products based on machine vision. *IET Image Processing*, *17*(2), 303-322.

Tegin, B., & Duman, T. M. (2023, December). Transformation-Invariant Over-the-Air Combining for Multi-Sensor Wireless Inference. In GLOBECOM 2023-2023 *IEEE Global Communications Conference* (pp. 2937-2942). IEEE.

Tekli, J. (2022). An overview of cluster-based image search result organization: background, techniques, and ongoing challenges. *Knowledge and Information Systems*, *64*(3), 589-642.

Tilley, R. J. (2020). *Colour and the optical properties of materials*. New York: John Wiley & Sons.

Titrek, F., & Baykan, Ö. K. (2023). Finger Vein Recognition Based on Multi-Features Fusion. *Traitement du Signal*, *40*(1), 101.

Tosta, T. A. A., Freitas, A. D., de Faria, P. R., Neves, L. A., Martins, A. S., & do Nascimento, M. Z. (2023). A stain color normalization with robust dictionary learning for breast cancer histological images processing. *Biomedical Signal Processing and Control, 85*, 104978.

Uyar, K., Taşdemir, Ş., Ulker, E., Unlukal, N., & Solmaz, M. (2022). Improving efficiency in convolutional neural networks with 3D image filters. *Biomedical Signal Processing and Control, 74*, 103563.

Varanis, M., Silva, A. L., Balthazar, J. M., & Pederiva, R. (2021). A tutorial review on time-frequency analysis of non-stationary vibration signals with nonlinear dynamics applications. *Brazilian Journal of Physics, 51*, 859-877.

Vasan, D., Alazab, M., Wassan, S., Naeem, H., Safaei, B., & Zheng, Q. (2020). IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture. *Computer Networks, 171*, 107138.

Veerashetty, S., Virupakshappa, & Ambika. (2022). Face recognition with illumination, scale and rotation invariance using multiblock LTP-GLCM descriptor and adaptive ANN. *International Journal of System Assurance Engineering and Management*, 1-14.

Veerasingam, S., Ranjani, M., Venkatachalapathy, R., Bagaev, A., Mukhanov, V., Litvinyuk, D., ... & Vethamony, P. (2021). Contributions of Fourier transform infrared spectroscopy in microplastic pollution research: A review. *Critical Reviews in Environmental Science and Technology, 51*(22), 2681-2743.

Vijayalakshmi, D., & Nath, M. K. (2023). A strategic approach towards contrast enhancement by two-dimensional histogram equalization based on total variational decomposition. *Multimedia Tools and Applications*, *82*(13), 19247-19274.

Vijayan, T., Sangeetha, M., Kumaravel, A., & Karthik, B. (2023). Feature selection for simple color histogram filter based on retinal fundus images for diabetic retinopathy recognition. *IETE Journal of Research*, *69*(2), 987-994.

Wang, L., Yang, N., & Wei, F. (2023). Query2doc: Query expansion with large language models. *arXiv preprint arXiv:2303*.07678.

Wang, M., Ao, W. K., Bownjohn, J., & Xu, F. (2022). A novel gradient-based matching via voting technique for vision-based structural displacement measurement. *Mechanical Systems and Signal Processing, 171*, 108951.

Wang, P., Li, Y., & Vasconcelos, N. (2021). Rethinking and improving the robustness of image style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 124-133).

Wang, S., Wang, H., Zhou, Y., Liu, J., Dai, P., Du, X., & Wahab, M. A. (2021). Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching. *Measurement, 169*, 108362.

Wang, W. (2023). Expression Recognition Based on Gabor Filter. *Science and Technology of Engineering, Chemistry and Environmental Protection, 1*(4).

Wang, Y., Huang, Q., Jiang, C., Liu, J., Shang, M., & Miao, Z. (2023). Video stabilization: A comprehensive survey. *Neurocomputing, 516*, 205-230.

Wang, Y., Wan, R., Yang, W., Li, H., Chau, L. P., & Kot, A. (2022, June). Low-light image enhancement with normalizing flow. *In Proceedings of the AAAI conference on artificial intelligence, 36*(3), 2604-2612.

Weng, W., & Zhu, X. (2021). INet: convolutional networks for biomedical image segmentation. *Ieee Access, 9*, 16591-16603.

Wijayathunga, L., Rassau, A., & Chai, D. (2023). Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: A review. *Applied Sciences, 13*(17), 9877.

Wu, K. (2023). Creating panoramic images using ORB feature detection and RANSAC-based image alignment. *Advances in Computer and Communication, 4*(4), 220-224.

Wu, Z., Liu, W., Li, J., Xu, C., & Huang, D. (2023). SFHN: spatial-frequency domain hybrid network for image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology, 33*(11), 6459-6473.

Xiang, D., He, D., Gao, P., Wang, H., Zhai, C., Qu, Q., ... & Zhong, J. (2024). Dual-color space color correction and histogram segmentation optimized strategy for underwater image enhancement. *Earth Science Informatics*, 1-19.

Xie, Y., Wang, Q., Chang, Y., & Zhang, X. (2022). Fast target recognition based on improved ORB feature. *Applied Sciences, 12*(2), 786.

Xu, C., Makihara, Y., Li, X., Yagi, Y., & Lu, J. (2020). Cross-view gait recognition using pairwise spatial transformer networks. *IEEE Transactions on Circuits and Systems for Video Technology, 31*(1), 260-274.

Xu, F., Wang, H., Sun, X., & Fu, X. (2022). Refined marine object detector with attention-based spatial pyramid pooling networks and bidirectional feature fusion strategy. *Neural Computing and Applications, 34*(17), 14881-14894.

Xu, Y., Zhang, J., & Brownjohn, J. (2021). An accurate and distraction-free vision-based structural displacement measurement method integrating Siamese network based tracker and correlation-based template matching. *Measurement, 179*, 109506.

Xue, D., Lei, T., Jia, X., Wang, X., Chen, T., & Nandi, A. K. (2020). Unsupervised change detection using multiscale and multiresolution Gaussian-mixture-model guided by saliency enhancement. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14*, 1796-1809.

Yang, D., Peng, B., Al-Huda, Z., Malik, A., & Zhai, D. (2022). An overview of edge and object contour detection. *Neurocomputing, 488*, 470-493.

Yang, J., Yee, P. L., Khan, A. A., Karamti, H., Eldin, E. T., Aldweesh, A., ... & Omar, A. (2023). Intelligent lung cancer MRI prediction analysis based on cluster prominence and posterior probabilities utilizing intelligent Bayesian methods on extracted gray-level co-occurrence (GLCM) features. *Digital health*, *9*, 20552076231172632.

Yang, J., Zhang, J., Xu, H., & Meng, L. (2020). Color normalization for histopathological images: Review and analysis. *Journal of Healthcare Engineering*, *2020*.

Yang, T., Li, R., Liang, N., Li, J., Yang, Y., Huang, Q., ... & Zhang, H. (2020). The application of key feature extraction algorithm based on Gabor wavelet transformation in the diagnosis of lumbar intervertebral disc degenerative changes. *PLoS One*, *15*(2), e0227894.

Yang, Y. G., Zou, L., Zhou, Y. H., & Shi, W. M. (2020). Visually meaningful encryption for color images by using Qi hyper-chaotic system and singular value decomposition in YCbCr color space. *Optik*, *213*, 164422.

Yang, Z., Huangfu, H., Leng, L., Zhang, B., Teoh, A. B. J., & Zhang, Y. (2023). Comprehensive competition mechanism in palmprint recognition. *IEEE Transactions on Information Forensics and Security*. *18*, 5160-5170.

Yao, X., & Song, T. (2022). Rotation invariant Gabor convolutional neural network for image classification. *Pattern Recognition Letters*, *162*, 22-30.

Yao, X., Song, T., Zeng, J., & Xie, Y. (2022). Rotation Invariant Convolutional Neural Network Based on Orientation Pooling and Covariance Pooling. In The *International Conference on Image, Vision and Intelligent Systems* (ICIVIS 2021) (pp. 433-443). Singapore: Springer Nature Singapore.

Yee, A. L. K., Razali, M., Ismail, M. A. M., Yusoff, I. N., Nagendran, S. K., Zainal, Z., ... & Yokota, Y. (2023). Preliminary analysis of rock mass weathering grade using image analysis of CIELAB color space with the validation of Schmidt

hammer: A case study. Physics and Chemistry of the Earth, Parts a/b/c, 129, 103291.

Yu, X., Yang, Q., Zhou, Y., Cai, L. Y., Gao, R., Lee, H. H., ... & Tang, Y. (2023). Unest: local spatial representation learning with hierarchical transformer for efficient medical segmentation. *Medical Image Analysis*, *90*, 102939.

Yuan, Y., Wang, L. N., Zhong, G., Gao, W., Jiao, W., Dong, J., ... & Xiang, W. (2022). Adaptive Gabor convolutional networks. *Pattern Recognition*, *124*, 108495.

Zafar, A., Aamir, M., Mohd Nawi, N., Arshad, A., Riaz, S., Alruban, A., ... & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, *12*(17), 8643.

Zhang, P., Fan, W., Chen, Y., Feng, J., & Sareh, P. (2022). Structural symmetry recognition in planar structures using convolutional neural networks. *Engineering Structures*, *260*, 114227.

Zhang, Z., Yu, L., Liang, X., Zhao, W., & Xing, L. (2021). TransCT: dual-path transformer for low dose computed tomography. In *Medical Image Computing and Computer Assisted Intervention*–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VI 24 (pp. 55-64). Springer International Publishing.

Zhao, R., Wang, Z., Guo, W., & Zhang, C. (2023). Multi-scene image enhancement based on multi-channel illumination estimation. *Expert Systems with Applications*, *226*, 120271.

Zhao, X., Tao, R., Li, W., Philips, W., & Liao, W. (2021). Fractional Gabor convolutional network for multisource remote sensing data classification. *IEEE Transactions on Geoscience and Remote Sensing*, *60*, 1-18.

Zhao, Z., Xu, X., Li, J., Li, S., & Plaza, A. (2023). Gabor-Modulated Grouped Separable Convolutional Network for Hyperspectral Image

Classification. *IEEE Transactions on Geoscience and Remote Sensing. 61*, 1-17, doi: 10.1109/TGRS.2023.3301183.

Zheng, X., Sun, H., Lu, X., & Xie, W. (2022). Rotation-invariant attention network for hyperspectral image classification. *IEEE Transactions on Image Processing*, *31*, 4251-4265.

Zhou, F., Zhao, H., & Nie, Z. (2021, January). Safety helmet detection based on YOLOv5. In 2021 IEEE International conference on power electronics, computer applications (ICPECA) (pp. 6-11). IEEE.

Zhou, J., Gai, Q., Zhang, D., Lam, K. M., Zhang, W., & Fu, X. (2024). IACC: cross-illumination awareness and color correction for underwater images under mixed natural and artificial lighting. *IEEE Transactions on Geoscience and Remote Sensing*, *62*, 1-15.

Zhou, Y., Zhang, Z., Wang, X., Sheng, Q., & Zhao, R. (2024). Multimodal archive resources organization based on deep learning: a prospective framework. *Aslib Journal of Information Management.* https://doi.org/10.1108/AJIM-07-2023-0239

Zhu, L., Chen, T., Yin, J., See, S., & Liu, J. (2023). Learning gabor texture features for fine-grained recognition. In Proceedings of the IEEE/CVF *International Conference on Computer Vision* (pp. 1621-1631).

Zhuang, L., Da, F., Gai, S., & Li, M. (2020). Transformation-invariant Gabor convolutional networks. *Signal, Image and Video Processing*, *14*, 1413-1420.

Zitouni, A., Benkouider, F., Chouireb, F., & Belkheiri, M. (2021). Comparison Between Gabor Filters and Wavelets Transform for Classification of Textured Images. In Proceedings of the 4th *International Conference on Electrical Engineering and Control Applications:* ICEECA 2019, 17–19 December 2019, Constantine, Algeria (pp. 1021-1031). Springer Singapore.

APPENDICES

Appendix I: Publications

1. Gateri, J., Rimiru, R. M., & Kimwele, M. (2023). Rotational Invariance Using Gabor Convolution Neural Network and Color Space for Image Processing. *International Journal of Ambient Computing and Intelligence (IJACI), 14*(1), 1-11.
2. Rimiru, R. M., Gateri, J., & Kimwele, M. W. (2022). GaborNet: investigating the importance of color space, scale and orientation for image classification. *PeerJ Computer Science, 8*, e890.