

**UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS**

**Image Texture Analysis and Feature Extraction using Multi-Scale  
Decomposition and Supervised Learning**

THÈSE PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU PROGRAMME DE DOCTORAT EN SCIENCES ET TECHNOLOGIES DE  
L'INFORMATION

PAR

**ASAL ROUHAFZAY**

19 Avril 2023

## **Jury d'évaluation**

Président du Jury : Dr. Michael Korwin Pawlowski

Membre du Jury externe : Dr. Abdelhamid Mammeri

Membre du Jury interne : Dr. Ana-Maria Cretu

Directeur de recherche : Dr. Nadia Baaziz

Codirecteur de recherche : Dr. Mohand Saïd Allili

## **Remerciements**

I owe a huge debt of gratitude to my dear supervisors Professor Dr. Nadia Baaziz and Professor Dr. Mohand Saïd Allili for constantly encouraging my research and for being supportive whenever I needed help. Their guidance has been priceless. Many thanks to my beloved parents Lili Razeghi and Dr. Farhad Rouhafzay for their unconditional love and wise counsel. I am blessed to be your daughter. My appreciation also goes to the best sisters in the word, Dr. Ghazal Rouhafzay and Lael Rouhafzay for your love, support and kindness.

# Contents

<b>Jury d'évaluation .....</b>	<b>2</b>
<b>Remerciements .....</b>	<b>3</b>
<b>Contents .....</b>	<b>4</b>
<b>List of Figures.....</b>	<b>7</b>
<b>List of Tables .....</b>	<b>12</b>
<b>Abstract.....</b>	<b>15</b>
<b>Résumé .....</b>	<b>16</b>
<b>CHAPTER 1. Introduction.....</b>	<b>17</b>
1. Motivation.....	17
2. Objectives and contributions .....	19
3. Document outline.....	20
<b>CHAPTER 2. Multi-scale image transforms .....</b>	<b>21</b>
1. The Laplacian pyramid.....	21
2. The discrete wavelet transform .....	22
3. The Gabor transform .....	24
4. The standard contourlet transform (SCT) .....	28
5. The non-subsampled contourlet transform (NSCT).....	30
6. The redundant contourlet transform (RCT) .....	31
<b>CHAPTER 3. Texture discrimination and machine learning.....</b>	<b>37</b>
1. Feature extraction for texture discrimination .....	38
2. Supervised machine learning: training and prediction .....	41
2.1. Accuracy and cross-validation .....	42
2.2. Support Vector Machine (SVM) .....	43
2.3. K-Nearest Neighbors (KNN) .....	44

3. Content-based image retrieval .....	45
3.1. <i>Feature extraction and indexing</i> .....	46
3.2. <i>Similarity measurement</i> .....	47
3.3. <i>Retrieval efficiency</i> .....	48
3.4. <i>Machine learning in CBIR</i> .....	50
4. Region of interest (RoI) detection .....	53
5. Deep learning.....	54
<b>CHAPTER 4. Methodology .....</b>	<b>57</b>
1. The RCT-Plus representation of grayscale textures .....	58
2. Statistical modeling of the RCT-Plus .....	62
2.1. <i>General Gaussian density modeling (GGD)</i> .....	62
2.2. <i>Energy-based modeling</i> .....	64
2.3. <i>Joint color texture modeling</i> .....	65
2.4. <i>Gaussian Mixture Modeling (GMM)</i> .....	66
3. A CBIR system based on color-texture discrimination.....	72
3.1. <i>The proposed approach for CBIR</i> .....	72
3.2. <i>Integrating supervised learning in CBIR</i> .....	74
4. Abnormality detection in color texture images .....	78
4.1. <i>The labeling of local features</i> .....	79
4.2. <i>The offline stage</i> .....	81
4.3. <i>The online stage</i> .....	83
4.4. <i>Application to segmentation in dermoscopic images</i> .....	86
<b>CHAPTER 5. Experiments and results .....</b>	<b>88</b>
1. Data description and evaluation criteria .....	88
1.1. <i>The VisTex-40 dataset</i> .....	89
1.2. <i>The Kylberg-27 dataset</i> .....	90
1.3. <i>The Kylberg-28 dataset</i> .....	91
1.4. <i>The Stex dataset</i> .....	92
1.5. <i>The ISIC-32 dataset</i> .....	93

1.6. <i>The ISIC-42 dataset</i> .....	95
1.7. <i>Evaluation criteria</i> .....	97
2. Texture feature extraction methods .....	98
2.1. <i>The multi-scale decomposition in feature extraction methods</i> .....	98
2.2. <i>RCT-Plus in comparison to other transforms</i> .....	103
2.3. <i>Color texture feature extraction methods in the CBIR framework</i>	107
3. Integrating a learning approach into the CBIR framework.....	112
3.1. <i>ML algorithms selection</i> .....	112
3.2. <i>ML-CBIR results, comparison and discussion</i> .....	113
4. Abnormality detection experiments and results .....	119
4.1. <i>Abnormality detection choice of parameters and methods</i> .....	119
4.2. <i>Block-based detection</i> .....	120
4.3. <i>Texture segmentation and comparison with a deep learning approach</i> .....	126
<b>CHAPTER 6. Conclusion and future work</b> .....	<b>131</b>
<b>References</b> .....	<b>134</b>
Annex I .....	147
Annex II .....	149
Annex III.....	150
Annex IV.....	152
Annex V .....	155

## List of Figures

Figure 2.1 A block diagram for creating a Laplacian pyramid [3].....	22
Figure 2.2 Process of decomposing an image into two-level DWT.....	23
Figure 2.3 Gabor kernels corresponding to 3 scale levels and 4 orientations per level.....	25
Figure 2.4 Gabor decomposition block diagram. ....	26
Figure 2.5 Original Tile1 image [7].....	27
Figure 2.6 Gabor sub-bands corresponding to filtered Tile1 image using multi-scale Gabor kernels.....	27
Figure 2.7 The two stages of an SCT decomposition scheme: the multi-scale stage (LP) implementing a Laplacian pyramid and the directional filter bank stage (DFB).....	29
Figure 2.8 Frequency partition for a two-level SCT decomposition with 8 directional sub-bands at scale level 1 and 4 directional sub-bands at scale level 2.....	29
Figure 2.9 NSCT decomposition scheme. An example of the performed frequency partition is shown for each stage (NSLP and NSDFB). ....	30
Figure 2.10 RCT decomposition diagram (3 scale levels, 4 directions). Separable Pseudo-Gaussian filters are used in the Redundant Laplacian (RLP) stage (with $a=2, 4$ and $8$ ).....	32
Figure 2.11 A block diagram of the RCT decomposition illustrating an example of frequency partition by the RLP stage and the DFB stage.....	33
Figure 2.12 Original Tile image [7].....	34
Figure 2.13 RCT sub-bands of Tile image for 3 scale levels and 4 directions. ....	34
Figure 2.14 Original Brick Image [7].....	35
Figure 2.15 RCT sub-bands of Brick image for 3 scales and 4 directions per scale level.....	35
Figure 2.16 Original Tile image. (128×128) [7].....	36
Figure 2.17 An example of a redundant Laplacian pyramid (RLP) decomposition (3 scale levels).....	36

Figure 3.1 Support Vector Machine (SVM) schema in a two-dimensional space [46].	44
Figure 3.2 KNN classification process [47].	45
Figure 3.3 CBIR schema, online and offline phases.	46
Figure 3.4 Illustration of <i>Precision</i> and <i>Recall</i> metrics [48].	49
Figure 3.5 <i>Recall</i> and <i>Precision</i> metrics are related [48].	49
Figure 3.6 CBIR scheme with a classification approach.	52
Figure 4.1 Block diagram of RCT-Plus decomposition (with 3 scale levels).	59
Figure 4.2 Frequency response of the 2-D pseudo-Gaussian filter for $a=2$ .	60
Figure 4.3 Frequency response of the 2-D pseudo-Gaussian filter for $a=4$ .	60
Figure 4.4 Frequency response of the 2-D pseudo-Gaussian filter for $a=8$ .	60
Figure 4.5 A block diagram of the RCT-Plus decomposition illustrating an example of frequency partition by the RLP stage and the DFB stage.	61
Figure 4.6 RCT-Plus decomposition of a $N \times N$ grayscale texture image. The number of scale levels is $L=3$ . The number of directional sub-bands at each scale level $l$ is indicated by $D_l$ , thus $[D_1, D_2, D_3] = [4, 8, 4]$ . The total number of RCT-Plus coefficients is $4 \times N \times N$ and the redundancy factor is 4.	61
Figure 4.7 Three different color texture images in RGB format from the VisTex and Stex datasets [7] [25]. The second row illustrates the corresponding Gaussian mixture models with $k=2$ , and the third row illustrates their corresponding RGB histogram.	69
Figure 4.8 Three different color texture blocks in skin images from the ISIC dataset [40]. The second row illustrates the corresponding Gaussian mixture models with $k=2$ , and the third row illustrates their corresponding RGB histogram.	69
Figure 4.9 GGD and GMM fitting examples applied on RCT-Plus approximation sub-band of the red color image component. The images are selected from VisTex and Stex datasets [7] [25].	70
Figure 4.10 GGD fitting examples applied on an RCT-Plus decomposition with ( $L=3$ and $D=[8\ 8\ 8]$ ). The image is selected from ISIC dataset [40].	71
Figure 4.11 ML-CBIR scheme with KNN/SVM classifiers.	75
Figure 4.12 ML-CBIR using concatenated color texture features.	77
Figure 4.13 ML-CBIR using RCT-Plus grayscale texture features concatenated to color texture features which are extracted from the RCT-Plus approximation sub-bands using GM modeling.	77



Figure 4.14 ML-CBIR using RCT-Plus grayscale texture features and a color texture feature using GM modeling of the whole image. .... 78

Figure 4.15 Examples of dermoscopic color images from ISIC dataset [40] (on the left side). The corresponding grayscale images with *lesion* border representation using the *Labeling method 2* (on the right side). .... 81

Figure 4.16 Abnormality detection scheme using RCT-Plus texture features concatenated with GMM color features and KNN/SVM binary classifiers. 85

Figure 4.17 Abnormality segmentation framework using Graph-Cut. .... 86

Figure 5.1 Texture images selected from the VisTex dataset [7]. From left to right and top to bottom: Bark0, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Buildings9, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric14, Fabric15, Fabric17, Fabric18, Flowers5, Food0, Food5, Food8, Grass1, Leaves8, Leaves10, Leaves11, Leaves12, Leaves16, Metal0, Metal2, Misc2, Sand0, Stone1, Stone4, Terrain10, Tile1, Tile4, Tile7, Water5, Wood1, and Wood2. .... 89

Figure 5.2 RGB VisTex dataset with 40 classes [7]. Each image, indicates a sample from 16 images for each class. From left to right and top to bottom: Bark0, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Buildings9, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric14, Fabric15, Fabric17, Fabric18, Flowers5, Food0, Food5, Food8, Grass1, Leaves8, Leaves10, Leaves11, Leaves12, Leaves16, Metal0, Metal2, Misc2, Sand0, Stone1, Stone4, Terrain10, Tile1, Tile4, Tile7, Water5, Wood1, and Wood2. .... 90

Figure 5.3 Kylberg-27 texture dataset [37]. From left to right and top to bottom: blanket1, canvas, ceiling1, ceiling2, cushion1, floor1, floor2, grass1, lentils1, linseeds1, oatmeal1, pearlsugar1, rice1, rice2, rug1, sand1, scarf1, scarf2, screen1, seat1, seat2, sesameseed1, stone1, stone2, stone3, and stoneslab1. .... 91

Figure 5.4 Kylberg-28 texture dataset [37]. from left to right and top to bottom: blanket1, blanket2, canvas, ceiling1, ceiling2, cushion1, floor1, floor2, grass1, lentils1, linseeds1, oatmeal1, pearlsugar1, rice1, rice2, rug1, sand1, scarf1, scarf2, screen1, seat1, seat2, sesameseed1, stone1, stone2, stone3, and stoneslab1. .... 92

Figure 5.5 A sample of 63 classes of the Salzburg Texture Image Dataset (STex) [25]. .... 93

Figure 5.6 Selected dermoscopic images (ISIC-32) from the ISIC dataset [40] and their corresponding ground truth segmentation maps. .... 94

Figure 5.7 Selected 42 dermoscopic images (ISIC-42) from the ISIC dataset [40] and their corresponding ground truth segmentation maps.....	96
Figure 5.8 Average retrieval rates ( $AR\%$ ) according to the number of top matches considered, $TopN$ . The compared CBIR methods (GGD1, GGD2, GGD2-ED, E1 and E2) are applied on an RCT-Plus decomposition with 3 scale levels and 8 directional sub-bands per scale level ( $L=3, D= [8, 8, 8]$ ). .....	101
Figure 5.9 Average retrieval rates ( $AR\%$ ) of 640 queries according to the number of top matches considered, $TopN$ . Two retrieval methods are compared: GGD1 using a 3-level RCT-Plus with $D= [4, 4, 4]$ and GGD1 using a 3-level RCT-Plus with $D= [8, 8, 8]$ .....	102
Figure 5.10 Average retrieval rates ( $AR\%$ ) of 640 queries according to the number of top matches considered, $TopN$ . The GGD1 method is applied on 3-level contourlet decomposition variants (SCT, NSCT, RCT and RCT-Plus). .....	105
Figure 5.11 Examples of Top16 retrieval results from the VisTex-40 dataset (640 images). The query image is Food8.02. ....	106
Figure 5.12 Examples of KNN-CBIR retrieval using GGD2 on VisTex-40. For each submitted query (left column), the predicted class membership (right column) is false. ....	115
Figure 5.13 Examples of abnormality detection results mapped onto 4 original test images from ISIC-32. For each image, the red contour represents the original block-based ground truth, and the white contour shows the resulting abnormality detection map ( <i>healthy/lesion</i> ). ....	123
Figure 5.14 Examples of abnormality detection results mapped onto four test images from the ISIC-42 dataset. The rows show: a) Texture <i>border/non-border</i> detection maps; b) Color <i>healthy/lesion</i> detection maps; c) Conciliation between texture and color detection maps and d) a comparison between the achieved detection results and the block-based ground truths (red contours). ....	125
Figure 5.15 Examples of image segmentation results obtained on ISIC-32 dataset. The first column a) illustrates five ground truths and their superimposition on the corresponding original images. The columns b)-e) show segmentation results obtained by using Grabcut, U-net_50, U-net_250, and U-net_500, respectively. ....	129
Figure 5.16 Examples of image segmentation results obtained on ISIC-42 dataset. The first column a) illustrates five ground truths and their	

superimposition on the corresponding original images. The columns b)-e) show segmentation results obtained by using Grabcut, U-net\_50, U-net\_250, and U-net\_500 U-net, respectively. .... 130

Figure 0.1 Original Zone Plate Image. .... 147

Figure 0.2 Redundant contourlet sub-bands of zone plate image for 3 scale levels and 4 directions..... 147

Figure 0.3 RCT-Plus decomposition of a  $256 \times 256$  zone plate image. The number of scale levels is  $L=3$ . The number of directional sub-bands at each scale level  $l$  is indicated by  $D_l$ , thus  $[D_1, D_2, D_3] = [4, 8, 8]$ . The redundancy factor is 4..... 148

Figure 0.4 Examples of Top16 retrieval results from the VisTex-40 dataset (640 images). The query image is Stone.0004.15. .... 149

Figure 0.5 SVM ..... 150

Figure 0.6 Examples of primary abnormal detection results mapped onto original test images from ISIC-32 dataset. The considered classifier is KNN with  $K=7$  and KLD distance, operating on the feature datasets a) *Dataset-0%*, b) *Dataset-30%*, and c) *Dataset-40%*. .... 158

## List of Tables

Table 4.1 Illustration of the estimated parameters for GM model with $k=2$ . .....	67
Table 5.1 Average retrieval rates ( $AR\%$ ) according to the number of top matches considered, $TopN$ . The compared CBIR schemes (E1, E2, GGD1, GGD2 and GGD2-ED) operate on RCT-Plus decomposition with L ranging from 1 to 3 and 8 directional sub-bands per scale level. ....	100
Table 5.2 Average retrieval rates ( $AR\%$ ) in the Top16 images. Retrieval methods using RCT-Plus with various combinations of scale levels and directions are compared. The number of directional sub-bands at each scale level $l$ is indicated by $D_l$ .....	102
Table 5.3 Average retrieval rates ( $AR\%$ ) in the Top16 images. CBIR schemes based on contourlet variants (SCT, RCT, NSCT and RCT-Plus) are compared to DWT and Gabor CBIR schemes. ....	104
Table 5.4 Color texture feature extraction methods: associated models and similarity metrics. ....	108
Table 5.5 Average retrieval rates ( $AR\%$ ) in the Top16 images. The compared CBIR schemes operate on RCT-Plus decomposition with $D=[8, 8, 8]$ and $L=3$ , and GMM modeling is performed with $k=2$ . ....	110
Table 5.6 Comparison of the proposed color texture GGD1+GM method to various state of art methods in terms of feature vector length and CBIR average retrieval rates $AR$ (%) in the Top16 images. ....	111
Table 5.7 Comparison of the proposed grayscale texture methods, GGD1+GMapp and GGD2+GMapp, to the existing methods LBP-ED and LDPVBP (Local directional peak valley binary pattern [35]) in terms of feature vector length and CBIR average retrieval rates $AR(\%)$ on Grayscale image datasets. ....	111
Table 5.8 <i>Accuracy</i> measures (%) of compared ML algorithms on the whole VisTex-40 dataset. The learning phase is held with 5-fold cross-validation. Trained features are derived from GGD1, GGD2, and E2 methods using RCT-Plus.....	113
Table 5.9 Performance comparison of ML-CBIR schemes in terms of average retrieval rates ( $AR\%$ ), number of false predictions and classifier <i>Accuracy</i>	

corresponding to a learning phase with 5-fold and 10-fold cross-validations. The considered ML algorithms are KNN and SVM and texture dataset is VisTex-40 [7]...... 114

Table 5.10 Performance comparison of ML-CBIR schemes vs. conventional CBIR schemes in terms of average retrieval rates ( $AR\%$ ) over 640 image queries from VisTex-40 dataset. .... 115

Table 5.11 Average retrieval rates ( $AR\%$ ) of ML-CBIR using fine (K=1) KNN. .... 117

Table 5.12 Average retrieval rates ( $AR\%$ ) ML-CBIR using SVM. .... 118

Table 5.13 Comparison of the proposed grayscale texture GGD1+GMapp method to various state-of-the-art methods in terms of CBIR average retrieval rates  $AR\%$  in the Top16 images for VisTex-40 dataset. .... 118

Table 5.14 A sample of block-based abnormality detection results corresponding to 4 test images from ISIC-32 dataset. RCT-Pus, GGD and GMM ( $k=2$ ) modeling of texture and color texture blocks are used. .... 123

Table 5.15 A comparison between block-based abnormality detection results using: a) texture features, b) color texture features and c) joint color texture features from RGB and grayscale images in the ISIC-42 dataset. .... 124

Table 5.16 A comparison between block-based abnormality detection results using: a) texture features, b) color texture features and c) joint color texture features from YCbCr and grayscale images in the ISIC-42 dataset. .... 124

Table 5.17 Comparison of segmentation results between U-net deep learning and Grabcut methods applied on ISIC-32 dataset. .... 127

Table 5.18 Comparison of segmentation results between U-net deep learning and Grabcut methods applied on ISIC-42 dataset. .... 127

Table 0.1 Average retrieval rates ( $AR\%$ ) according to the number of top matches considered,  $TopN$ . The compared CBIR methods (E1, E2, GGD1, GGD2 and GGD2-ED) operate on NSCT decompositions with  $D=8$  and  $L$  ranging from 1 to 3. .... 152

Table 0.2 Average retrieval rates ( $AR\%$ ) according to the number of top matches considered. The compared CBIR schemes operate on RCT-Plus decomposition with  $D= [8, 8, 8]$  and  $L=3$ , and GMM modeling is performed with  $k=2$ . .... 153

Table 0.3 ML-CBIR using fine (K=1) KNN. .... 154

Table 0.4 ML-CBIR using SVM. In GGD1, GGD2, E2 and LBP linear kernel and in GGD1+GMapp and GGD1+GM the KLD kernel are applied. .... 154

Table 0.5 A sample of abnormality detection results, *Precision*, *Recall* and *Accuracy*, corresponding to various selected feature datasets (*Dataset-T %*) from ISIC-32. The KNN classification algorithm is based on either KLD or ED distance. The factor K is equal to 7..... 157

Table 0.6 A sample of abnormality detection results, *Precision*, *Recall* and *Accuracy*, corresponding to various selected feature datasets (*Dataset-T %*) from ISIC-32. The considered classification algorithm is SVM with linear kernel and ED distance. .... 157

## Abstract

Texture analysis is an active research area in image processing and computer vision. Analyzing images with powerful feature extraction methods can lead to the successful design and implementation of machine intelligence applications such as content-based image retrieval, image classification, object detection, image segmentation, face recognition, abnormality detection, etc. In this thesis, we address the issue of texture analysis and discrimination with a new methodology based on parametric statistical modeling of multi-scale image representations. A novel multi-scale image decomposition, named RCT-Plus, is proposed. It is a variant of the contourlet transform that is redundant, rich in directional information, and applicable to grayscale and color texture images. We also propose a hybrid approach for modeling texture data in the multi-scale space by a combination of suitable parametric statistical models such as Generalized Gaussian Distribution (GGD) and multivariate Gaussian Mixture Model (GMM). This approach along with adapted similarity metrics resulted in the development of new feature extraction methods that capture relevant texture information, provide highly compact features, allow for a joint exploitation of texture and color texture features and enhance texture discrimination in applications such as content-based image retrieval (CBIR) in texture datasets and abnormality detection in dermoscopic images of human skin tissue. Furthermore, supervised machine learning algorithms (KNN and SVM) are integrated into the processing system as key techniques of feature learning and multi-class classification to infer texture types on the extracted features and achieve improved performance in terms of texture discrimination. Various experimental setups are conducted using six well-known texture datasets. We successfully increased the image retrieval rate up to 97.10% for the Stex dataset while the size of the feature vector is reduced to only 67 elements. In the case of abnormality detection, moving from grayscale texture features to joint color texture features improved the *Precision* of detection by up to 21% in the ISIC-42 dataset. A comparison with state-of-the-art methods, including deep learning, showed that our proposed texture feature extraction methodology yields more successful results.

## Résumé

L'analyse de texture est un domaine de recherche actif en traitement d'images et en vision par ordinateur. L'analyse d'images avec de puissantes méthodes d'extraction de primitives peut conduire à la conception et à la mise en œuvre d'applications d'intelligence artificielle réussies telles que la récupération d'images basée sur le contenu, la classification d'images, la détection d'objets, la segmentation d'images, la reconnaissance faciale, la détection d'anomalies, etc. Dans cette thèse, nous abordons la problématique d'analyse et de discrimination des textures avec une nouvelle méthodologie basée sur la modélisation statistique paramétrique des représentations multi-échelles d'images. Une nouvelle décomposition multi-échelles d'images, nommée RCT-Plus, est proposée. C'est une variante de la transformée en contourlettes qui est redondante, riche en informations directionnelles et applicable aux images de texture en niveaux de gris et en couleur. Nous proposons également une approche hybride pour modéliser les données de texture dans l'espace multi-échelles par une combinaison de modèles statistiques paramétriques appropriés tels que la distribution gaussienne généralisée (GGD) et le mélange de gaussiennes multivariées (GMM). Cette approche, associée à des métriques de similarité adaptées, a abouti au développement de nouvelles méthodes d'extraction de primitives qui capturent des informations de texture pertinentes, fournissent des primitives très compactes, permettent une exploitation conjointe des primitives de texture et de texture en couleur et améliorent la discrimination de texture dans des applications telles que la recherche d'images basée sur le contenu (CBIR) dans des bases de données de textures et la détection d'anomalies dans des images dermoscopiques de tissus cutanés humains. De plus, des algorithmes d'apprentissage automatique supervisé (KNN et SVM) sont intégrés au système de traitement en tant que techniques clés d'apprentissage de primitives et de classification multi-classes pour prédire le type de texture des primitives extraites et obtenir des performances améliorées en termes de discrimination de texture. Diverses expérimentations sont menées en utilisant six bases de données de texture bien connues. Nous avons réussi à augmenter le taux de récupération d'images jusqu'à 97,10 % pour la base de données Stex, tandis que la taille des primitives est réduite à seulement 67 éléments. Dans le cas de la détection d'anomalies, l'exploitation conjointe de primitives de texture en niveaux de gris et des primitives de textures en couleur a permis d'améliorer la *Précision* de la détection d'un taux de 21 % pour la base de données ISIC-42. Une comparaison avec des méthodes de l'état de l'art, y compris l'apprentissage profond, a montré que la méthodologie proposée donne de meilleurs résultats.



# CHAPTER 1. Introduction

## 1. Motivation

In images from the real world, grayscale and color textures are omnipresent (as in grasslands, tiles, brick walls, fabrics, biological tissues, etc.) and tend to reveal many image properties such as granularity, smoothness, coarseness, periodicity, geometric structure and orientation [1]. Therefore, special attention is given to texture analysis and texture features have gained more significance in many image processing, computer vision and machine intelligence applications. Indeed, relevant information from image textures emerges through feature extraction to provide useful and precise image characterization (image signatures).

Nowadays, with the great revolution in cloud data storage resources as well as the enhancement of the quality of image acquisition devices, the size of digital image datasets is increasing very fast. Efficient texture and color texture analysis, feature extraction methods and indexation are required for different domains, including medicine, industry, architecture, remote sensing, fashion, crime prevention, publishing, etc. For this aim, a huge research effort continues to be invested in developing methods for texture and color texture feature extraction that promote the quality of different applications relying on texture discrimination such as image retrieval, image classification, object detection, image segmentation, face recognition, abnormality detection, etc.

Previous works have reported the achievement of remarkable outcomes due to the development of a variety of texture and color texture feature extraction methods operating on multi-scale image representations in the spatial-frequency domain (e.g. wavelets). This probably was motivated by two core reasons: a) the human visual system adequacy to the spatial-frequency representation of image signals, and b) the inherent nature of texture patterns in terms of the presence of edges, with changes in scales and directions in repetitive structures [1].

On another side, artificial intelligence has witnessed enormous progress in recent decades and has found its way into many different fields including image analysis and processing. Consequently, taking advantage of artificial

intelligence, and more specifically machine learning (ML) paradigms, is a possible approach to enhance the performance of image processing and computer vision applications based on texture analysis and feature extraction. This work considers two main applications: content-based image retrieval and abnormality detection in medical skin lesion images.

Searching images in large visual media datasets is becoming frequent, but remains a challenging task of data retrieval with good accuracy and performance. Content-based image retrieval (CBIR) methodology refers to the automatic recovery of images from a dataset using similarity measurement criteria between the query characteristics and various low-level visual features (e.g., color, shape, texture) from the image indexing process [2].

It is important to mention that the success of content-based image retrieval is strongly related to the choice of an efficient similarity metric and the development of feature extraction methods that achieve powerful characteristic discrimination while providing feature representation with reduced dimensionality. Hence, such a system can be trained on previously labeled images and then used to classify new dataset images.

The study and investigation of medical skin lesion images to detect abnormalities is, on the one hand, a time-consuming task calling for high level of precision. On the other hand, a variety of subtle details in medical images may remain invisible to human eye. Computer-aided approaches can be advantageously used to help physicians analyze medical images. Detecting tumors, micro-calcification, vascular anomalies and skin lesions in different parts of the body are some examples where such computational approaches have been engaged to detect abnormalities. Nowadays, a variety of image feature extraction techniques are available, however feature extraction from medical images require further exploration due to the high level of sensitivity associated with medical images. We mainly focus on dermoscopic images of human skin lesions. Once appropriate texture and color texture features are extracted from skin texture images, either supervised or unsupervised machine learning techniques can be exploited to localize regions of interest and detect abnormality based on the extracted features.

## 2. Objectives and contributions

In this thesis, we address the issue of texture analysis and discrimination with a new methodology based on parametric statistical modeling of multi-scale image representations. The goal is to develop new methods for the extraction of powerful discriminating features from grayscale and color texture images and improve texture analysis. First, we develop a new variant of the contourlet transform as a contribution to the domain of multi-scale image representation. Augmented directional selectivity, redundancy, Gaussian filtering, directional filtering and applicability to grayscale and multi-channel color spaces are among the ingredients of this new multi-scale decomposition which allows the exposure of texture structures through low-frequency sub-bands and high-frequency directional sub-bands.

Second, a hybrid approach for modeling texture and color texture data in the multi-scale space by a combination of suitable parametric statistical models such as Generalized Gaussian Distribution (GGD) and multivariate Gaussian Mixture Model (GMM) is another contribution to this work. This approach along with adapted similarity metrics are meant to capture relevant information, provide highly compact features, allow for a joint exploitation of texture and color texture features and enhance texture discrimination in image processing and computer vision applications such as 1) content-based image retrieval in image texture datasets and 2) detection of abnormal tissues on a medical skin lesion image dataset.

Third, we focus on coupling texture features and color texture feature extraction methods with supervised machine learning algorithms and analyze the benefits of feature learning and multi-class classification in both CBIR and abnormality detection schemes. A trained KNN or SVM learning model is used to predict the class label of a given query, speed up the labeling process and make it more accurate compared to traditional computations for similarity measurement, ranking of images or detection of the region of interest.

### 3. Document outline

The remainder of this document is organized as follows. Chapter 2 describes the main characteristics of various types of multi-scale image decompositions such as wavelets, standard contourlets, non-subsampled contourlets and redundant contourlets. Chapter 3 presents some well-known works dealing with feature extraction for texture discrimination, the working principles of supervised machine learning (ML) algorithms for classification problems and content-based image retrieval schemes. Further details and state-of-the-art about CBIR with ML approach, region of interest (RoI) detection and deep learning are also provided. Chapter 4 is devoted to the proposed methodology for feature extraction and the proposed schemes for content-based image retrieval, abnormality detection and segmentation in medical skin lesion images. Redundant contourlet decomposition of texture images, parametric statistical modeling like GGD and multivariate GMM, KLD-based similarity metrics and also supervised machine learning for feature classification are among the main components of the proposed frameworks. Chapter 5 presents the evaluation and validation of the proposed methodology for texture feature extraction and its incorporation into content-based image retrieval and abnormality detection schemes. We describe the conducted experimental studies on six popular datasets followed by evaluation criteria and provide the main experimental results with discussions and drawn conclusions. Finally, a summary of contributions, concluding remarks and future work are highlighted in chapter 6.

## CHAPTER 2. Multi-scale image transforms

Multi-scale image transforms are spatial-frequency representations resulting from the decomposition of the image into a set of sub-images or sub-bands corresponding to localized frequency partitions of the image spectrum. Laplacian pyramids, discrete wavelets, Gabor transforms and contourlets are examples of these image representations where various characteristics are exhibited such as multiple scales, frequency selectivity, multiple directional orientations, spatial redundancy and shift invariance. Multi-scale image transforms rely primarily on the use of two processing tools: appropriately designed linear filters and sub-sampling operators.

The redundancy factor is determined by the ratio of the total number of transform coefficients to the number of original image samples. In the case of critical sub-sampling (maximum possible sub-sampling rate) the redundancy factor is one.

### 1. The Laplacian pyramid

In the classic Laplacian pyramid decomposition [3], a half-band low-pass filter is applied on the input image to yield a coarse version called image approximation at scale level 1. Since image approximation is a low-resolution signal, critical sub-sampling with rates 1:2 horizontally and 1:2 vertically is applied to reduce its size. By subtracting from the input image its approximation (previously interpolated) the difference information (image details) is captured as a Laplacian sub-band at scale level 1 ( $LP_1$ ).

The whole process is repeated  $L$  times on successive image approximations in order to form a Laplacian pyramid having  $L$  levels in addition to the last image approximation at scale level  $L$  (see Figure 2.1).

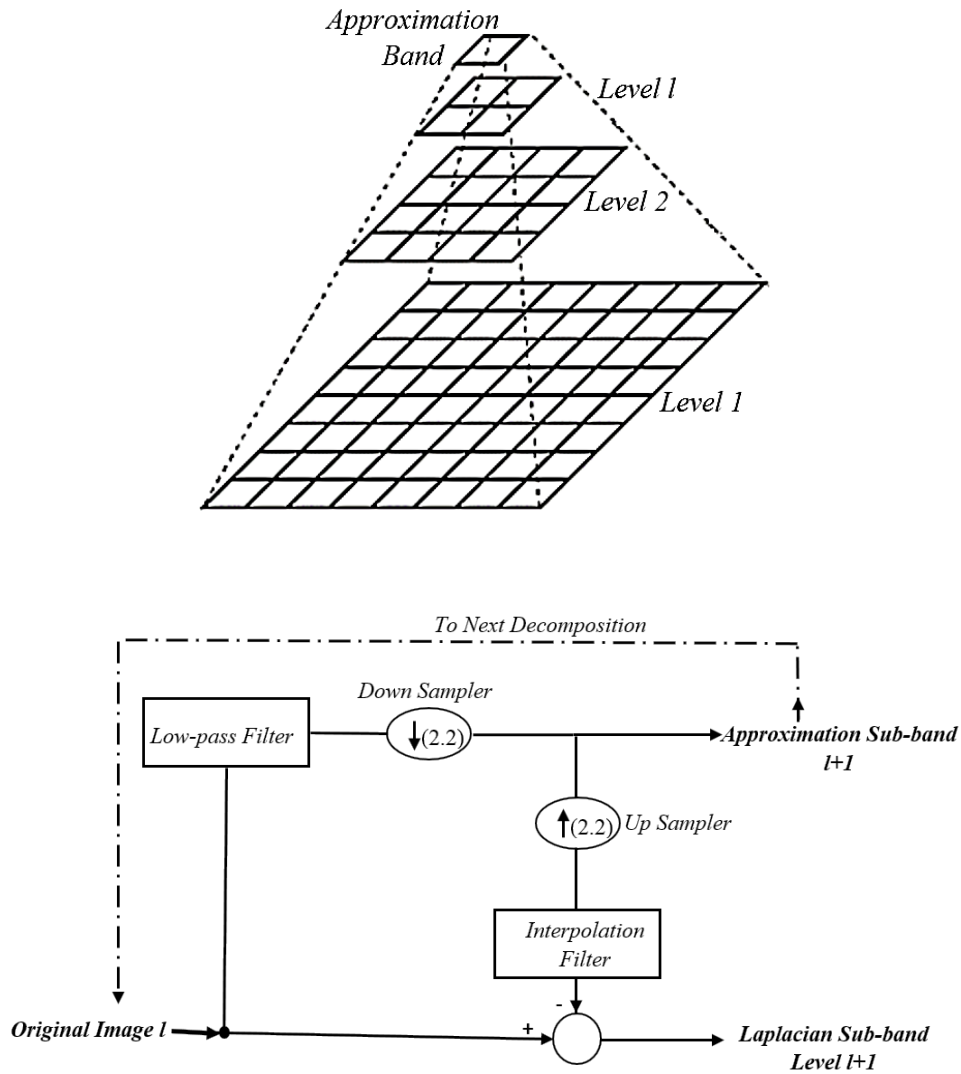


Figure 2.1 A block diagram for creating a Laplacian pyramid [3].

## 2. The discrete wavelet transform

The two-dimensional discrete wavelet transform (DWT) is an efficient tool for analyzing images in a multiresolution framework and capturing localized details of images in the spatial and frequency domains. For its implementation in decomposition mode, the DWT follows a recursive algorithm and uses low-pass and high-pass linear filtering with critical sub-sampling operations. The principle of this algorithm is to divide the input image into four sub-images at each iteration: three sub-bands named HH(high-high), HL(high-low) and LH(low-high), holding the oriented details of the image and a fourth sub-band

which contains the most important information (the low-frequency sub-band or the approximation image (LL)). The approximation sub-band will serve as an input image to the next scale level of decomposition which means it will be used to create the four sub-bands of the next iteration. Thus, this algorithm applies iterative linear filtering. The sub-bands (HH, HL, and LH) are sparse sub-images showing image details in horizontal vertical and diagonal orientations.

The DWT provides a highly compact image representation, i.e., the transformation is orthogonal, and the applied sub-sampling rates (1:2 horizontal and 1:2 vertical) result in a total number of wavelet coefficients which equals the original image size. Therefore, the redundancy factor of the DWT decomposition is always 1.

Although DWT is one of the most popular transformation in image compression field, it is limited in capturing relevant information due to the lack of shift invariance (translation sensitivity), the poor frequency selectivity (only three sub-bands at each decomposition level) and the low directionality (only the horizontal, vertical and diagonal orientations are provided). Figure 2.2 illustrates the DWT decomposition process with two scale levels resulting into seven wavelet sub-bands.

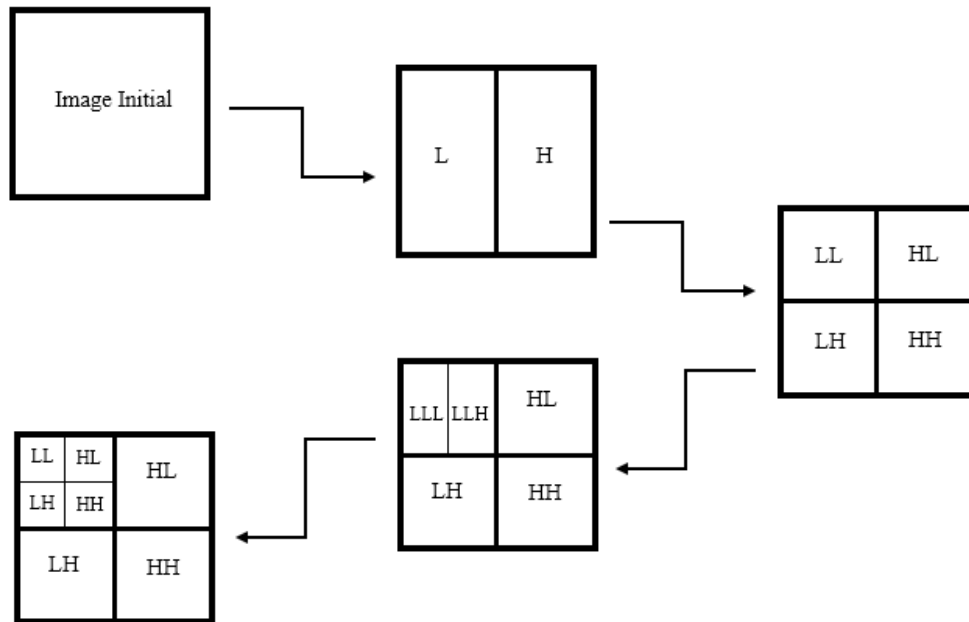


Figure 2.2 Process of decomposing an image into two-level DWT.

### 3. The Gabor transform

Gabor transform is widely used in image processing with the purpose of texture analysis and feature extraction. This transform is in fact the Fourier transform applied to a windowed version of the signal by a Gaussian which will result in representing the signal in time and frequency domain simultaneously. As the bandwidth is inversely proportional to the time duration, there is always uncertainty between the time and the frequency domain and Gabor function is proved to achieve the lowest uncertainty value [4] [5]. Based on the fact that Gabor transform is selective to the image orientation, sub-band decomposition is applied with four (4) different orientations. When Gabor filters are developed with three different scale factors, this leads to the generation of twelve Gabor sub-bands (corresponding to three scale levels and four orientations). Since no sub-sampling is applied, each resulting Gabor sub-band is oversampled and has the same size as the original image. Therefore, Gabor pyramids form a redundant multi-scale representation of image data [6] and the redundancy factor is equal to the total number of directional sub-bands. Figure 2.3 depicts the kernels used to create Gabor sub-bands.



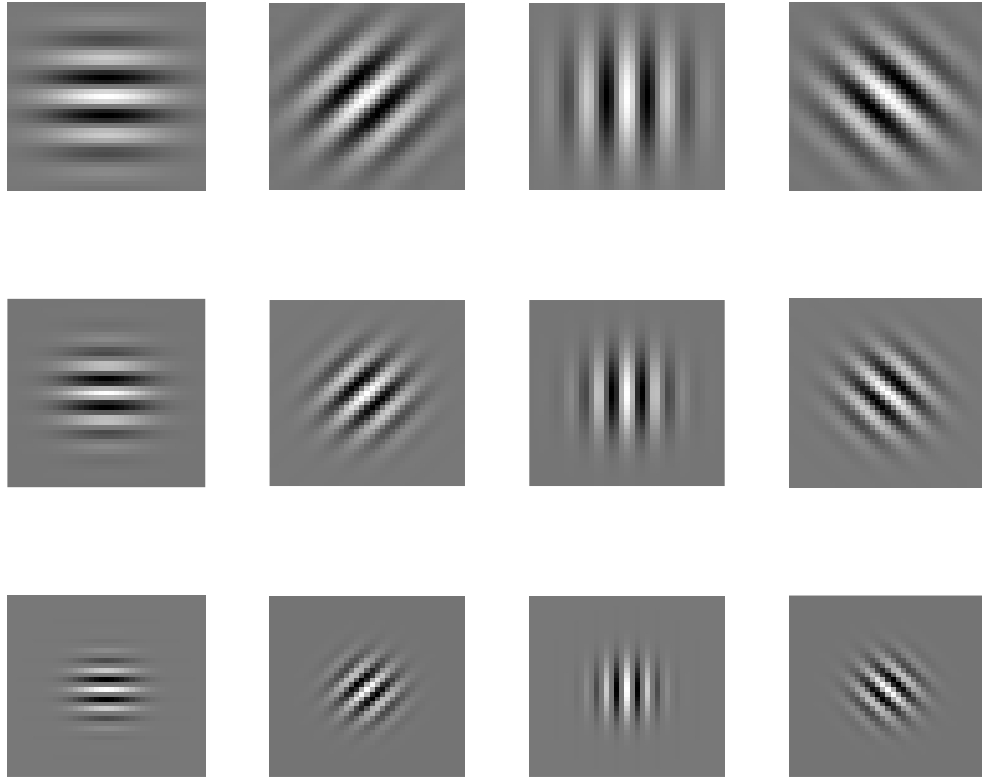


Figure 2.3 Gabor kernels corresponding to 3 scale levels and 4 orientations per level.

Figure 2.4 illustrates the Gabor decomposition block diagram. As a first step, the number of scales and orientations is selected, and a Gabor kernel is calculated. Second, a two-dimensional Gabor filter with impulse response  $g_k()$  is generated for each pair of scale and orientation values. Finally, input image is filtered using each  $g_k()$  to yield Gabor sub-bands  $B_k$  respectively.

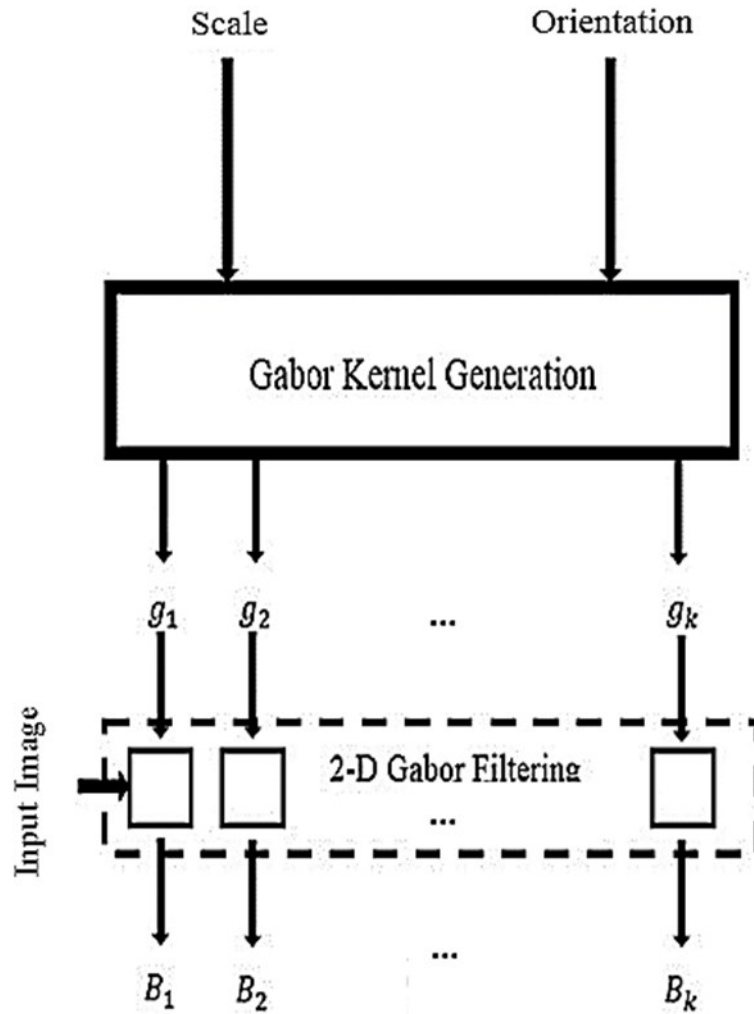


Figure 2.4 Gabor decomposition block diagram.

Figure 2.5 shows a texture image from the VisTex dataset [7] and Figure 2.6 depicts its decomposition into twelve Gabor sub-bands using Gabor kernels (with 3 scale levels and 4 orientations).

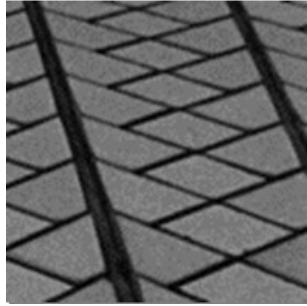


Figure 2.5 Original Tile1 image [7].

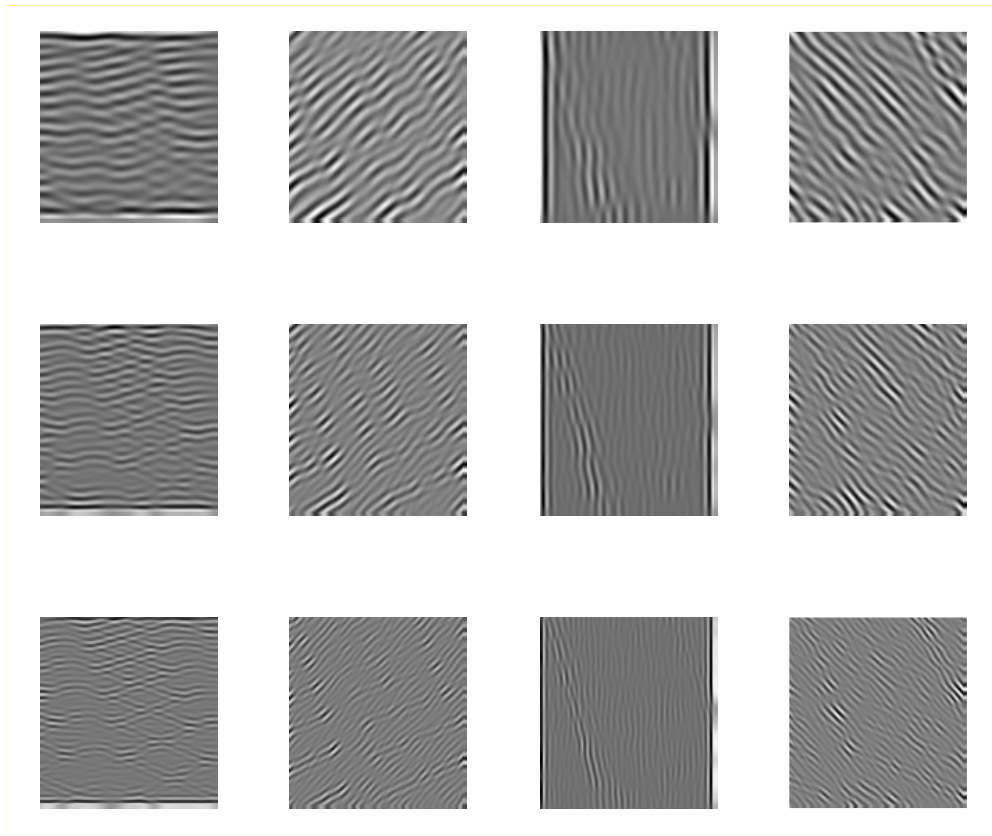


Figure 2.6 Gabor sub-bands corresponding to filtered Tile1 image using multi-scale Gabor kernels.

#### 4. The standard contourlet transform (SCT)

The discrete contourlet transform has been introduced as an enhancement of the two-dimensional discrete wavelet transform [8]. It is designated here as the standard contourlet transform (SCT). This transform is inspired from human vision capabilities and is able to capture the intrinsic geometrical structures and directional information in natural images. Multiple scales, multi-directionality, non-separable two-dimensional filtering and small amount of redundancy are among the ingredients of this geometric transform.

The SCT decomposition is performed with high computational efficiency by combining two distinct stages. In the first stage, a multi-scale decomposition uses a Laplacian pyramid (LP) scheme to transform the image into one coarse version plus a set of Laplacian sub-images [3]. In the second stage, a directional filter bank (DFB) applies iteratively 2-D non-separable filtering and critical sub-sampling to further partition each Laplacian scale level into different and flexible numbers of frequency wedge-shaped sub-bands, thus capturing geometric structures and directional information in natural images. Figure 2.7 illustrates the two stages of the SCT decomposition and Figure 2.8 shows an example of frequency partition.

In comparison to discrete wavelets, the SCT with its extra feature of directionality is almost critically sampled with a small redundancy factor up to  $4/3$  due to the inherent oversampling of the Laplacian pyramid. SCT leads to an efficient representation of smooth object boundaries with a small number of local coefficients in the right directional sub-bands.

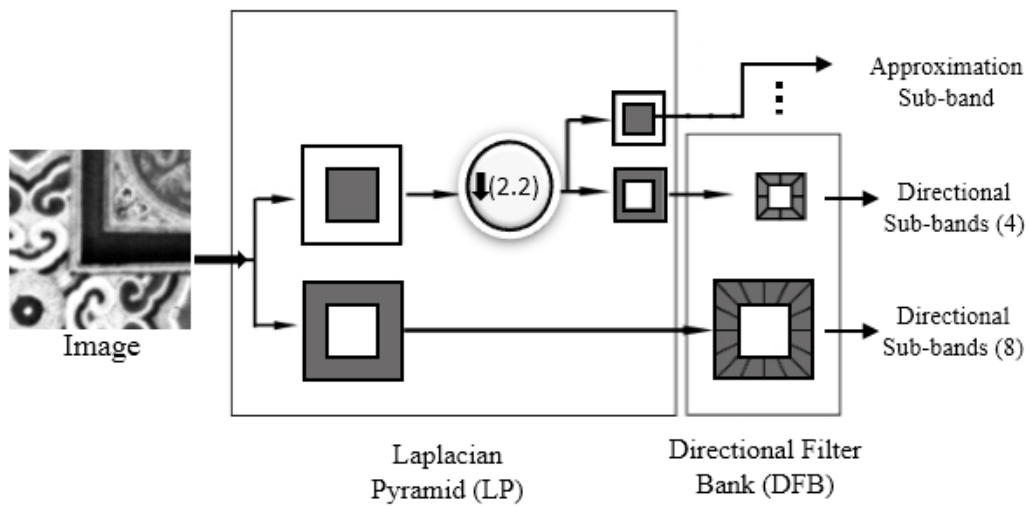


Figure 2.7 The two stages of an SCT decomposition scheme: the multi-scale stage (LP) implementing a Laplacian pyramid and the directional filter bank stage (DFB).

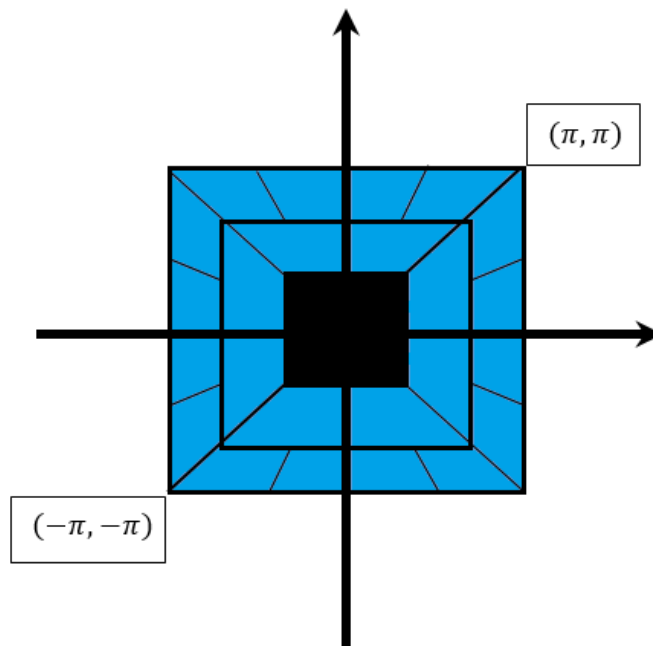


Figure 2.8 Frequency partition for a two-level SCT decomposition with 8 directional sub-bands at scale level 1 and 4 directional sub-bands at scale level 2.

## 5. The non-subsampled contourlet transform (NSCT)

The non-subsampled contourlet transform (NSCT) is a variant of the SCT [9] where all down-sampling operations are discarded from both decomposition stages thus eliminating aliasing problems and allowing for full shift-invariance and substantial redundancy. In other words, the NSCT decomposition scheme produces the same frequency partition as in SCT but all generated sub-images (sub-bands) are of the same size as the input image. As shown in Figure 2.9, the multi-scale stage is supported by a non-subsampled Laplacian pyramid (NSLP) while the directional stage is handled by a non-subsampled directional filter bank (NSDFB). However, its major drawback lies in the rapid increase of computational cost as large number of directional sub-bands are generated.

For example, the NSCT of a  $K \times M$  image with 3 scale levels and 8 directions per level results into one approximation sub-band and 24 directional sub-bands. Therefore, the total number of NSCT coefficients is equal to  $25 \times K \times M$  since each sub-band has a size identical to that of the original image. In general, if the total number of NSCT directional sub-bands is  $D$ , the resulting redundancy factor is  $D+1$ .

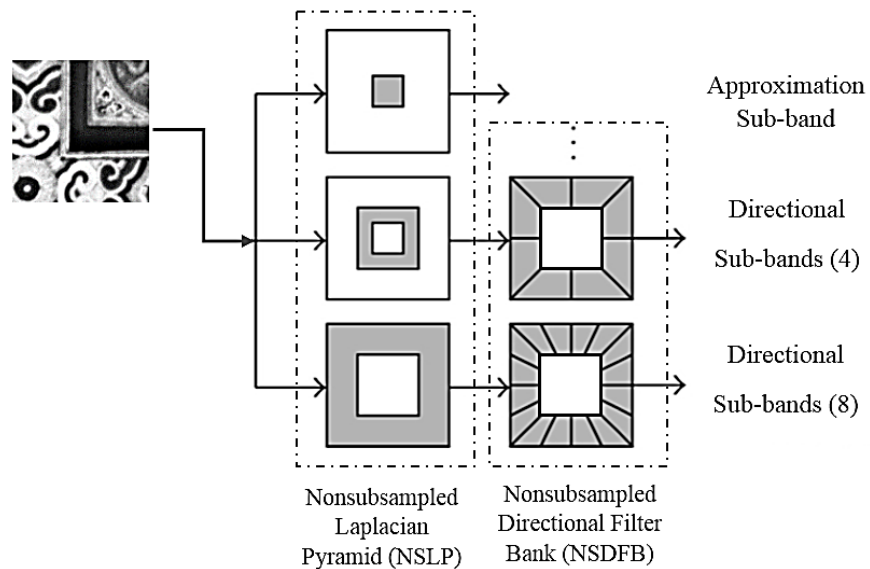


Figure 2.9 NSCT decomposition scheme. An example of the performed frequency partition is shown for each stage (NSLP and NSDFB).

## 6. The redundant contourlet transform (RCT)

The redundant contourlet transform (RCT) has been introduced by N. Baaziz in [10]. The RCT variant aims at increasing the amount of redundancy in the standard contourlet transform (SCT) by removing any sub-sampling operation in the multi-scale image representation. RCT decomposition scheme, like in SCT, takes two main parts: a multi-scale decomposition followed by a directional decomposition. However, main changes made to this scheme are as follows:

1) The multi-scale stage implements a redundant Laplacian pyramid (RLP) using low-pass filtering only. All sub-sampling operations are discarded in order to eliminate aliasing artefacts and provide additional redundancy amount in RLP sub-images.

2) Linear phase low-pass filters with pseudo-Gaussian properties are used to build the redundant Laplacian pyramid. Filter impulse responses  $h_a(n)$  are given by:

$$h_a(n) = e^{-2\left(\frac{n}{a}\right)^2} - e^{-2} \left\{ e^{-2\left(\frac{n-a}{a}\right)^2} + e^{-2\left(\frac{n+a}{a}\right)^2} \right\}, \quad (2.1)$$

where increasing values of the factor  $a$  (with  $a=2, 4, 8, 16\dots$ ) decrease the filter passband.

3) To build a redundant Laplacian pyramid having  $L$  scale levels,  $L$  pseudo-Gaussian filters (with  $a=2^l, l=1\dots L$ ) are used to generate  $L+1$  equal-size sub-images:  $L$  detail sub-images ( $RLP_l$ ) and one low-pass image approximation  $C_L$ . Therefore, the redundancy factor of this decomposition is  $L+1$ . The corresponding decomposition diagram is detailed in Fig. 2.10.

4) The directional decomposition stage, like in SCT, is a directional filter bank (DFB) using two-dimensional linear filtering and critical sub-sampling. However, the number of directional sub-bands is fixed and limited to four ( $D=4$ ) in order to ensure equal size sub-images at each scale level. As shown in Figure 2.10, a DFB with  $D=4$  orientations and  $1:4$  critical sub-sampling is

applied on each of the RLP sub-images to obtain 4 equal-size directional sub-bands  $\{C_{ld}; l=1 \dots L; d=1 \dots D \text{ with } D=4\}$ .

Therefore, additional redundancy and pseudo-Gaussian filtering in the multi-scale stage are the main specific ingredients in this contourlet variant designated as RCT. For a number  $L$  of scale levels, the total number of RCT sub-bands is  $4 \times L + 1$  and the redundancy factor is  $L + 1$ . Figure 2.11 illustrates the frequency partition performed at each stage during the RCT decomposition process. Examples of RCT and RLP decompositions of various images are given in Figures 2.12 to 2.17.

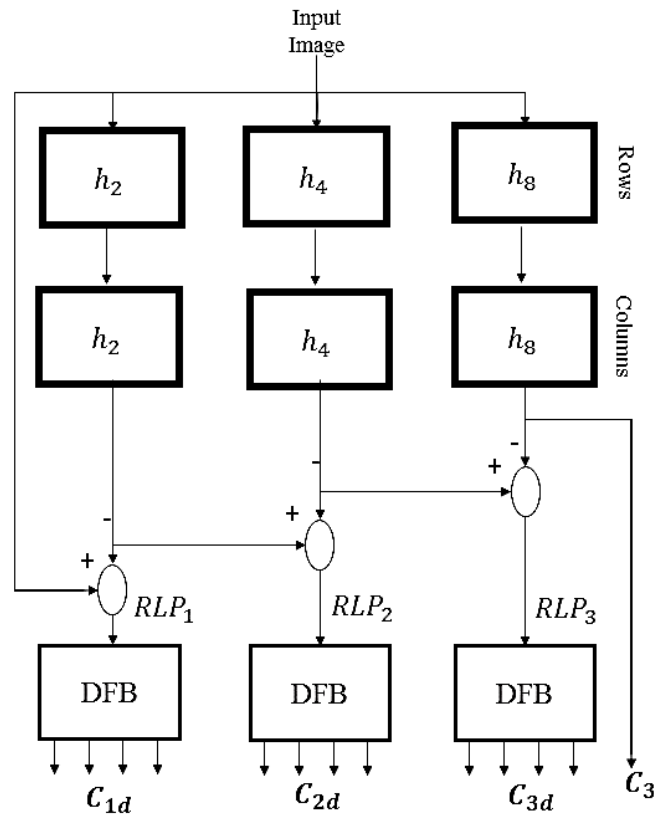


Figure 2.10 RCT decomposition diagram (3 scale levels, 4 directions). Separable Pseudo-Gaussian filters  $h_a$  are used in the Redundant Laplacian (RLP) stage (with  $a=2, 4$  and  $8$ ).



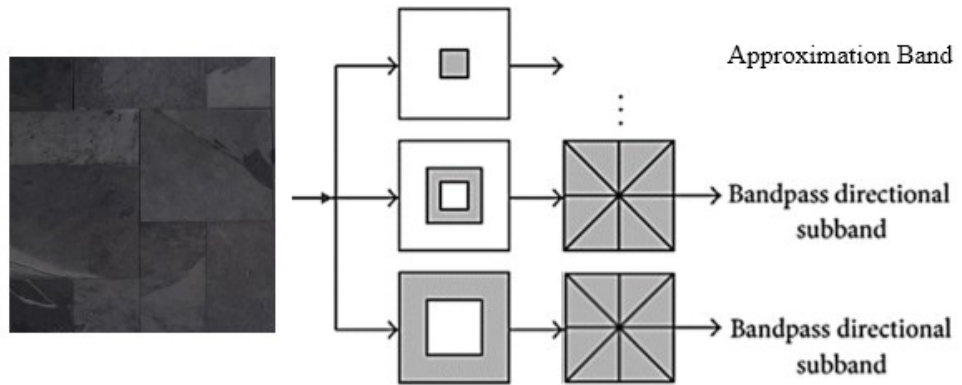


Figure 2.11 A block diagram of the RCT decomposition illustrating an example of frequency partition by the RLP stage and the DFB stage.

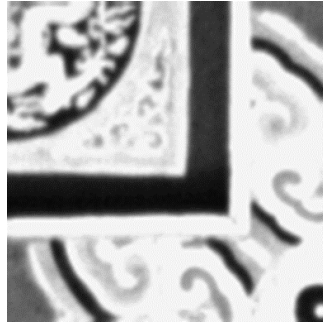


Figure 2.12 Original Tile image [7].

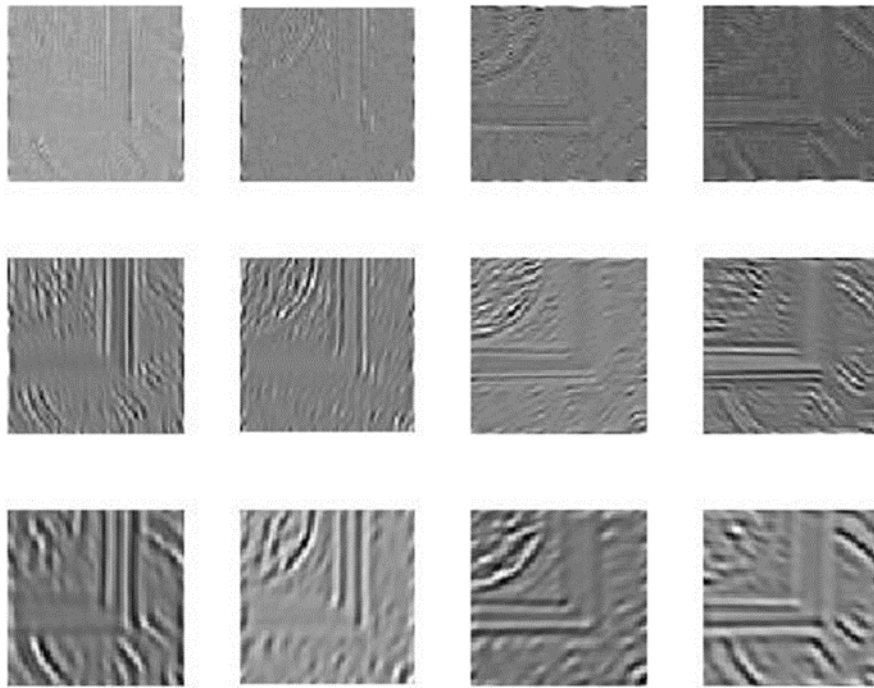


Figure 2.13 RCT sub-bands of Tile image for 3 scale levels and 4 directions.

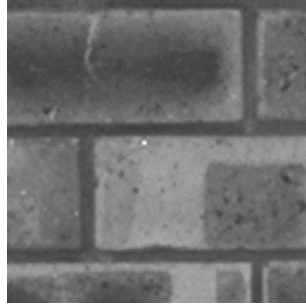


Figure 2.14 Original Brick Image [7].

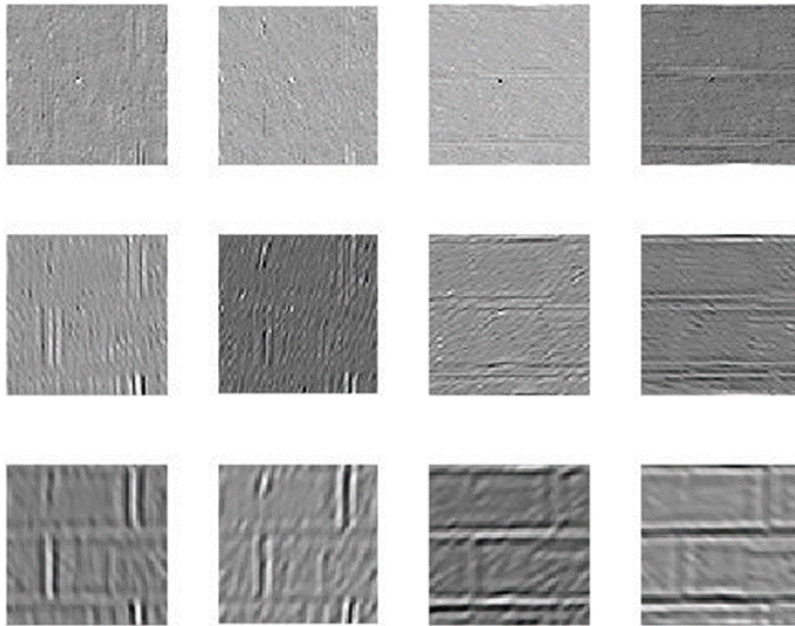


Figure 2.15 RCT sub-bands of Brick image for 3 scales and 4 directions per scale level.

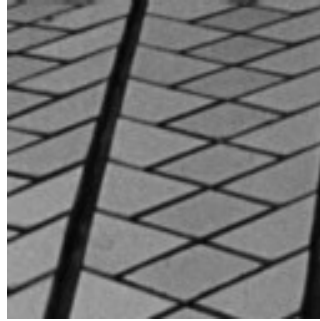


Figure 2.16 Original Tile image. (128×128) [7].

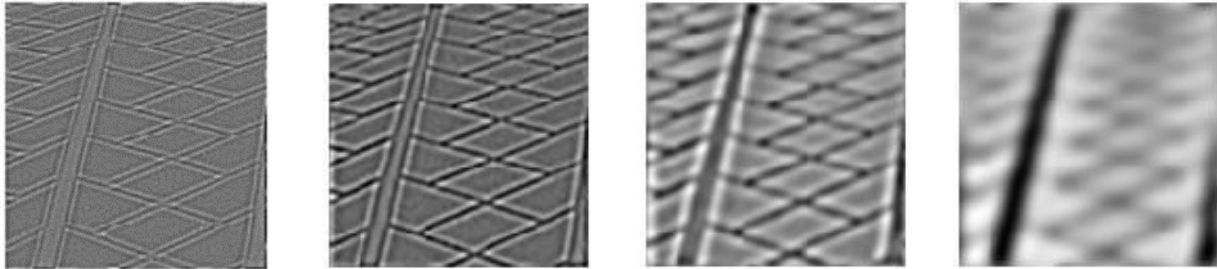


Figure 2.17 An example of a redundant Laplacian pyramid (RLP) decomposition (3 scale levels).

## **CHAPTER 3. Texture discrimination and machine learning**

Texture analysis has been extensively explored in the field of image processing and computer vision. Texture analysis targets revealing distinctive image properties such as granularity, smoothness, coarseness, periodicity, geometric structure, and orientation as a function of the spatial variation in image intensity. These features are visible in the intensity values, or grayscale levels as well as in different color spaces of the image.

We are interested in texture discrimination which consists of extracting valuable texture information from images that could lead to differentiating between various types of images or regions inside one image. Early image processing techniques were mainly based on grayscale level features for texture discrimination. Later on, the inclusion of color-based features provided crucial information that could boost up the performance of texture discrimination.

Nowadays, with the great revolution in cloud data storage resources as well as the enhancement of the quality of image acquisition devices, the size of digital image datasets is increasing very fast. Efficient texture and color texture analysis, feature extraction methods and indexation are required for different domains, including medicine, industry, architecture, remote sensing, fashion, crime prevention, publishing, etc. For this aim, a huge research effort continues to be invested in developing methods for texture and color texture feature extraction that promote the quality of different applications relying on texture discrimination such as defect detection, image retrieval, face recognition, abnormality detection, etc.

While the quality of the extracted features plays a decisive role in discriminating different textures, machine learning techniques can be advantageously integrated into the process to facilitate distinguishing among different textures. Machine learning algorithms are a group of computational techniques aiming to understand and build methods that can learn based on available data. Such techniques can be beneficial to discriminate textures based on their features.

## 1. Feature extraction for texture discrimination

A variety of feature extraction methods for texture discrimination have been explored during the two last decades. These features could be extracted in the spatial domain, frequency domain or spatial-frequency domain of grayscale and color images.

Do et al. [11] proposed a feature extraction method using GGD modeling of the wavelet decomposition sub-bands for image retrieval. Approximation sub-band is not included in the modeling. They achieved a retrieval rate of 74.66% in the grayscale VisTex [7] dataset.

Bharati et al. [12] compared different image texture analysis methods of several different works from state-of-the-art including traditional statistical approaches operating in the spatial domain such as gray level co-occurrence matrix (GLCM) methods, multivariate statistical approaches based on principal components analysis (PCA) and partial least squares (PLS). They further used Gabor and two-dimensional wavelet transform (in spatial-frequency domain) whose sub-bands were modeled by energy approach. They leveraged all these feature extraction methods to classify a set of rolled steel sheets based on their quality grades. For the experiments, a total of 35 images of steel surfaces were used. In the end, the wavelet texture analysis (WTA) method appeared to be the most powerful approach for image texture analysis. In [13] and [14] the authors proposed a new framework using finite Mixtures of Generalized Gaussian distributions MOGG statistical modeling of contourlet sub-bands. They combined low-pass and high-pass coefficients for texture description in grayscale images. This method is applied to image retrieval, defect detection and other applications. The achieved retrieval rate for the grayscale VisTex [7] dataset is 97.41%. Also, they used a binary Bayes classifier to classify the defect-free and defective fabrics in TILDA [15] dataset.

The authors of [16], [17] and [18] have experimented with different spatial domain grayscale feature extraction methods such as GLCM, LBP and local phase quantization (LPQ) using the SVM classifier in CBIR, edge detection and defect detection respectively. All three research works concluded that the GLCM feature extraction method outperforms other feature extractors on medical images and the TILDA [15] dataset.

In 2019, Alrahhhal et al. [19] extracted features from grayscale images by local neighbor pattern (LNP) to propose a new CBIR system using supervised

machine learning techniques. They compared their achieved results with the local tetra pattern (LTrP) method. They used the TDF face dataset [20], Corel 1k [21] and VisTex [7] grayscale datasets and improved the *Recall* between 1.55 to 5.24 percent.

In 2020, Alpaslan et al. [22] proposed a new hybrid LBP feature extractor based on Hessian matrix and attractive center-symmetric LBP. Their approach is called Hess-ACS-LBP. The Hessian matrix provides the directional derivative information of texture regions. Multi-scale Hessian matrix presents the intrinsic local geometry of the texture changes. They extracted texture information from eight grayscale datasets. They adopt KNN classifier considering only one nearest neighbor (fine KNN) to measure the similarity between images and classify them.

In 2021, [23] Khan et al. presented a novel texture feature descriptor named directional magnitude local hexadecimal pattern (DMLHP) and applied it for texture retrieval purposes. They used ensemble subspace discriminant (ESD) for classification and obtained a 98% retrieval rate in VisTex [7] dataset.

Also, for color texture images, many researches have been done to improve the texture analysis. Vasconcelos et al. [24] worked on embedded multiresolution mixtures of Gaussians to model the discrete cosine transform (DCT) coefficients of color images in the frequency domain. Notwithstanding their feature vector sizes were huge (6208 elements), they achieved a 73.70% retrieval rate in Stex [25] and 88.90% retrieval rates on VisTex [7] datasets. In the spatial-frequency domain of color texture images, Li et al. [26] have studied the dependence structure in Gabor wavelets, and used Gaussian copula to model the sub-bands of Gabor wavelets representations. They utilized KLD as the similarity measure between two Copula models for retrieval purposes. As the result, they achieved 76.4% and 66.1% retrieval rates on Stex [25] and VisTex [7] color datasets respectively. Etemad et al. [27], proposed a feature extraction method using a Gaussian Copula to model the dependencies between different sub-bands of the non-subsampled shearlet transform (NSST) and non-Gaussian models are used for marginal modeling of the coefficients. In conclusion, they achieved 80.81% and 69.47% retrieval rates on Stex-small [25] and VisTex-full [7] datasets. In 2022, the authors in [28] proposed a Gabor-GLCM-based color texture feature extraction method. Gabor filtering was performed in three RGB layers to enrich the texture

features in each direction, and a GLCM was used for feature extraction. The statistical values were compressed into eight dimensions by Gaussian weighted mean and PCA methods. They used GRU neural network for training and testing to predict the Oxygen (O<sub>2</sub>) content and Nitrogen oxides (NO<sub>x</sub>) from flame color texture images in industrial boilers. Also, they collected data from 8 hours of boiler operation and the data were divided into two parts: a 6.6 h training set and a 1.6 h test set. Bai et al. [29] presented a K-means-based histogram (KBH) using a combination of color and texture features for image retrieval. Multiresolution feature vectors representing color and texture features are directly generated from the coefficients of discrete wavelet transform (DWT). K-means is exploited to partition the vector space to reduce the number of histogram bins. Afterward, a fusion of z-score normalized Chi-Square distance between KBHs is adopted as the similarity measure. The method was applied successfully with improved retrieval rates in VisTex [7], Alot [30] and Stex [25] color datasets. However, they couldn't succeed in feature compactness and computational costs. In [31], Sugamya et al. combined three low-level feature extraction techniques for image retrieval purposes: color-correlogram for color feature extraction, Gabor wavelet and DWT decomposition modeled by energy for texture feature extraction. They used the SVM classifier to classify the features of a query image and distinguish between relevant and irrelevant images accordingly. This method led to a better performance than the traditional image retrieval method. In [32] Verdoolaege et al. proposed a new color texture feature extraction method in which the coefficients of the DWT transform were modeled by the multivariate power exponential. They applied their method for image retrieval in Stex [25] and VisTex [7] datasets and achieved 71.30% and 91.20% retrieval rates respectively. However, the feature vector in this approach lacks compactness (around 430 elements). Choy et al. [33] presented a statistical wavelet sub-band characterization based on generalized Gamma density in color texture image retrieval and obtained 52.90% and 81% retrieval rates on Stex [25] and VisTex [7] datasets.

As examples of feature extraction methods operating in the spatial domain, Kavitha et al. [34] considered the combination of texture and color features in feature extraction. GLCM is used to extract the texture features and Color histograms of images are used to extract the color features in three color spaces namely RGB, HSV and OPP. This method is presented to classify the



dermoscopic images into melanoma and non-melanoma by using the SVM classifier. The performance of the proposed system is evaluated by the metrics sensitivity and specificity. The experimental results suggested that the texture combined with RGB color space provides better classification *Accuracy*. In 2020, Gupta et al. [35] presented a feature extraction method named local directional peak valley binary pattern (LDPVBP) to use in the CBIR schema. In their approach, each center pixel has been compared to its 24 nearest neighbors by deriving the LBP value of each of the eight nearest neighbors and comparing them with the LBP value of the center pixel which is further used to derive one peak and one valley pattern. The method has been applied to RGB as well as grayscale texture images. They tested their proposed pipeline on five different face and texture datasets namely Brodatz [36], VisTex [7], Kylberg [37], AT&T [38] and Yale B. They achieved an 83.34% retrieval rate in VisTex [7] and 61.45% in the Kylberg [37] dataset. In 2020, Vidya et al. [39] used the ABCD rule, combined with GLCM and HOG for the early detection of skin lesions. They took advantage of a pre-processing step to improve the skin lesion quality and reduce artifacts. Geodesic active contour (GAC) method was used to segment the lesion regions. Then, ABCD scoring method implemented to extract additional features based on symmetry, border, color, and diameter of the lesion. HOG and GLCM were applied to extract textural features. Finally, all extracted features are directly passed to SVM, KNN and Naïve Bayes classifiers to classify skin lesions in their selected ISIC dataset [40].

## **2. Supervised machine learning: training and prediction**

Machine learning (ML) is part of artificial intelligence area which has witnessed an enormous progress in recent decades and has found its way in many different fields including image analysis and image processing. Machine learning algorithms are organized into several categories, such as supervised learning and unsupervised learning. The word supervised here means that there is a supervision on the learning process which depends on a set of training data with known labels (results) [41]. Indeed, a machine learning algorithm is trained on previously known results in order to build a mathematical model that correctly maps input data into output values (labels). The supervised learning process continues until a model with a desired level

of *Accuracy* is achieved. The obtained ML model is saved and can be used later to predict the results for new data (test data) whose output is not known. The prediction procedure varies according to the working principle of the machine learning model considered.

### *2.1. Accuracy and cross-validation*

To evaluate the efficiency of a supervised machine learning algorithm, one popular metric is *Accuracy* which is based on the prediction results obtained from applying the ML model to test data. For example, the prediction results of binary classification are formulated as four measures: true positives  $tp$ , false positives  $fp$ , true negatives  $tn$  and false negatives  $fn$ . The *Accuracy* measure is then given by the following ratio:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \times 100 \quad (3.1)$$

Cross-validation is a statistical approach widely used to determine how well a developed machine learning model can generalize the solution to new data. During the cross-validation the whole training data set is randomly split into a number  $n$  of equal folds. The ML model is then trained on  $n-1$  folds while the remaining fold is kept for testing purposes. This operation is iteratively repeated  $n$  times and each iteration deals with a newly selected testing fold. Consequently, we have  $n$  different values specifying the *Accuracy* of the ML model. The average of the  $n$  obtained measures is then considered as the overall performance of the ML model. This approach gives a more realistic evaluation of the model at expense of higher computational cost.

In this work, we focus on supervised machine learning applied to classification problems. By using ML algorithms, the following benefits are expected:

- Machine learning algorithms can review large capacities of data and discover specific trends and patterns that would not be obvious to humans.

- With machine learning, there is no need to follow a project every step of the way, since it gives the learning skill to the machine, allows it to make predictions.
- ML algorithms gain experience. They keep improving in *Accuracy* and efficiency as they are fed with new data. Accordingly, they can make better decisions.
- ML algorithms are good at handling data that is multi-dimensional and multivariate.

Among known supervised machine learning algorithms for classification [42], [43], [44], one can find decision trees, discriminant analysis, support vector machines (SVM) and K-nearest neighbors (KNN). Decision tree learning algorithm is creating a tree structure (as a predictive model) where each non-leaf node denotes a feature and leaf-nodes signify class labels. The prediction process (decision) goes from observations (features on branches) to conclusions (class labels on the leaves). This algorithm analysis is simply explainable and has low computational complexity, which is an easy and practical method in machine learning. Linear discriminant classifiers and quadratic discriminant classifiers try to determine linear boundaries and quadratic/surface boundaries, respectively, between existing classes. Generally, linear discriminant analysis assumes that data in each class belongs to a specific Gaussian distribution. During the training process the parameters (mean, variance) for the Gaussian distribution associated to each class are estimated. In the following sections, two other machine learning algorithms are presented, namely, SVM and KNN.

## 2.2. Support Vector Machine (SVM)

Support vector machine is a supervised machine learning approach where training data is represented as a set of feature vectors of size  $p$ , viewed as points in a  $p$ -dimensional space. Each feature vector has a class label. During the learning phase, the goal of SVM is to find a set of hyper-planes that separate the training data points into distinct classes according to their labeling [45]. In the case of two classes, the optimal choice corresponds to the hyper-plane that exhibits the largest gap (maximum margin) to the nearest data point of any class (see Figure 3.1). During the classification or prediction phase, a

new data point is simply mapped into the  $p$ -space in order to predict its class membership.

A kernel SVM is a version of SVM used for datasets where data is not separable as it is. Therefore, using a mapping function known as kernel, data points are transformed to a new feature space where the data are separable.

SVM is known for its efficiency in classifying high-dimensional data. SVM has different parameters such as kernel functions (linear, Gaussian, quadratic, cubic ...), Penalty parameter  $C$  and  $\Gamma$  which can be tuned to change the space and so the results. The  $\Gamma$  parameter is the inverse of the standard deviation of the Gaussian kernel, which is used as similarity measure between two points. A small  $\Gamma$  value presents the Gaussian function with a large variance. In this condition, two points can be considered similar even if they are far from each other.

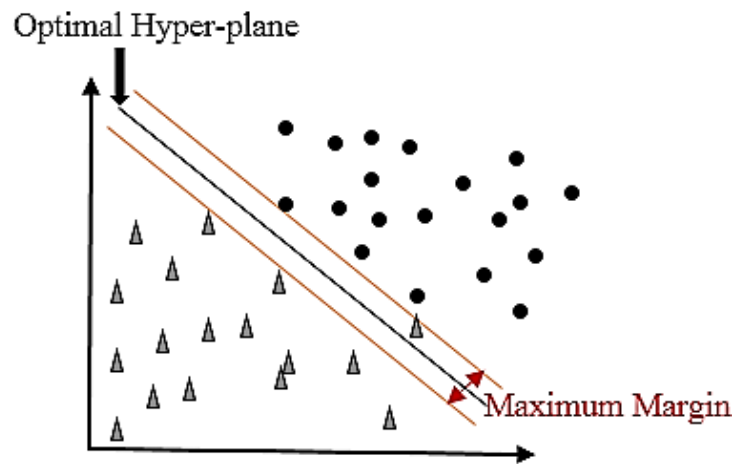


Figure 3.1 Support Vector Machine (SVM) schema in a two-dimensional space [46].

### 2.3. *K-Nearest Neighbors (KNN)*

KNN is a type of supervised learning method which is known to be very efficient in terms of classification capability on small data sets. In a KNN classifier, all training data features and corresponding class labels are required for classifying new input data. KNN, determines the similarity of the features by calculating the distance (such as Euclidean distance) between test data and training data features. Then, class membership is predicted by a voting process between the  $K$  nearest neighbors. A KNN classifier is shown in Figure 3.2

where the yellow point is a new testing data point. If we assume  $K=3$ , by the voting process the new data point is included in class B and if we assume  $K=7$ , the new data is contained in class A. In KNN algorithm, there are mainly two parameters to tune; the value of  $K$  which is the number of the neighbors as explained above and the choice of distance metric (such as Euclidean, Cosine and Manhattan distances).

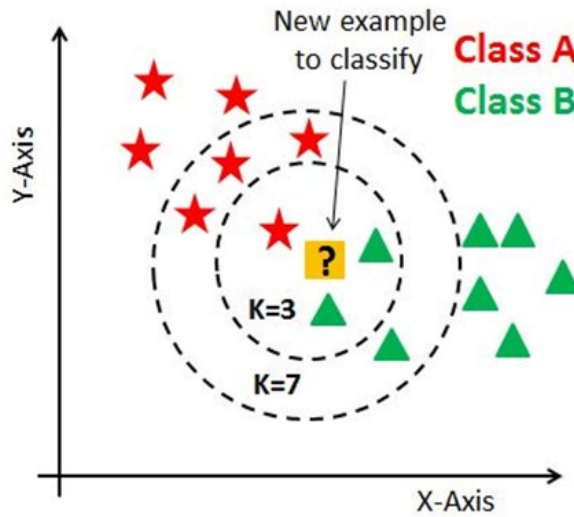


Figure 3.2 KNN classification process [47].

### 3. Content-based image retrieval

Content-based image retrieval (CBIR) methodology refers to the automatic search and recovery of images from a dataset using similarity measurement criteria between the query characteristics and various low-level visual features (color, shape, texture ...) from the image indexing process [2]. It is important to mention that the success of CBIR is strongly related on the choice of an efficient similarity metric and the development of feature extraction methods that achieve powerful characteristic discrimination while providing feature representation with reduced dimensionality.

In content-based image retrieval systems, the general architecture is organized into two separate phases: the offline phase and the online phase (as shown in

Figure 3.3. The offline phase is mainly dedicated to feature extraction and indexing process of the image dataset. The online phase operates a dataset search to find relevant images to the user query. Similarity measurements (distance metrics) are first calculated between the query features and the image dataset index. Next, The  $N$  most similar images to the query are presented (in a ranked order) to the user as being the most relevant in the sense of the similarity metric considered ( $TopN$  retrieved images).

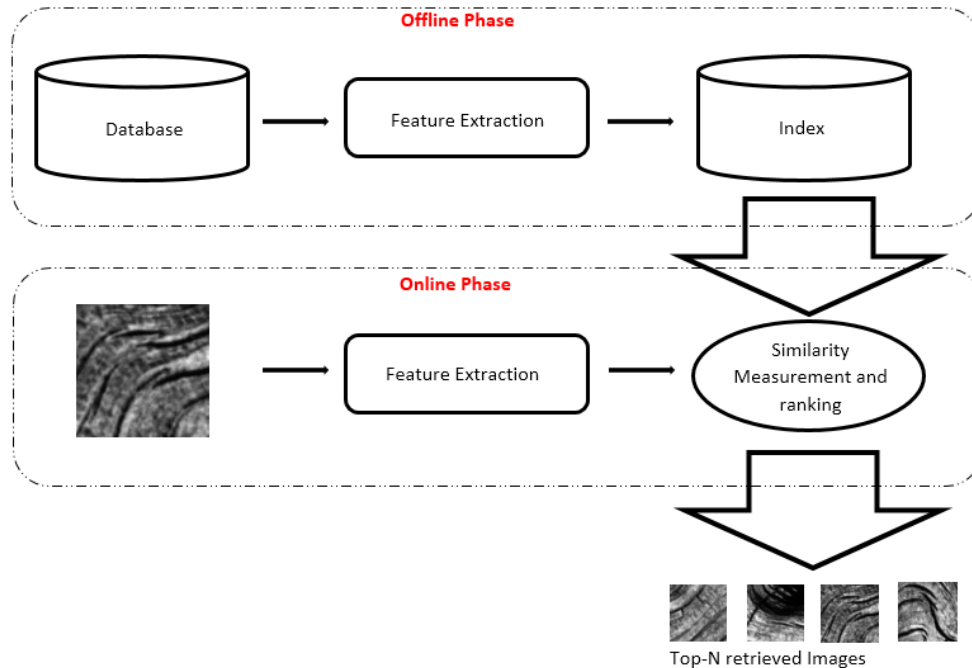


Figure 3.3 CBIR schema, online and offline phases.

### 3.1. Feature extraction and indexing

Feature extraction is a dimensionality decreasing process which starts from an initial set of measured data for human interpretation, supposed to be redundant. Then, it is transformed into a decreased set of features usually called feature vector. The chosen features are anticipated to possess the relevant information from the input data in order to achieve powerful characteristic discrimination while ensuring sparse data representation and compactness. In image data, relevant features are extracted from low-level visual characteristics such as color, shape and texture.

Index is the word generally referred to as an indicator or a point of reference. For a Given image dataset, a visual index is constructed and stored by computing, for each image, its feature vector which will represent the image during a search phase of a CBIR system. Indeed, the idea about indexing formed at this point is that features in one image can be rapidly associated with the most similar features in another image. The indexing process is done offline so as not to penalize the user with often lengthy calculations (offline phase). Due to the numerous publications dedicated to the field of content-based image retrieval, a state of the art is an ambitious undertaking. Many of the proposed techniques are dedicated to improve the quality of extracted image texture characteristics. Multi-scale approaches to feature extraction are discussed in the next chapter.

### *3.2. Similarity measurement*

Between two images, there is always similarity and dissimilarity degree which can be measured based on the components of their corresponding feature vectors. For this purpose, one or more suitable distance metrics (Manhattan, Euclidean, Canberra, Kullback-Leibler ...) are selected. After determining the distance or similarity of each pair of feature components, all measures are added together and turned into a single similarity degree between the two feature vectors considered.

In CBIR frameworks, retrieving similar images to the user query image is done through the calculation of the distance measures between the query image feature vector and the dataset visual index features. The  $N$  smallest distances in a ranked order are then selected as *TopN* matches and corresponding images are retrieved.

### 3.3. Retrieval efficiency

To evaluate CBIR efficiency, the most popular performance metrics are *Recall* and *Precision*. These two metrics are based on an understanding and measure of relevance.

*Recall* is the ratio of the number of relevant images retrieved to the total number of relevant images in the dataset. It is usually expressed as a percentage:

$$Recall = \frac{\text{number of relevant images retrieved}}{\text{total number of relevant images in the database}} \times 100 \quad (3.2)$$

*Precision* is the ratio of the number of relevant images retrieved to the total number of irrelevant and relevant images retrieved (*TopN* matches). It is usually expressed as a percentage:

$$Precision = \frac{\text{number of relevant images retrieved}}{\text{total number of retrieved images from the database}} \times 100 \quad (3.3)$$

Figure 3.4 describes *Recall* and *Precision* in terms of selected elements that are true positives, true negatives, false positives and false negatives. In Figure 3.5, one can see how *Recall* and *Precision* are inversely related. The two lines of the graph may represent the performance of different CBIR systems. While the exact slope of the curve may vary between systems, the general inverse relationship between *Recall* and *Precision* remains.



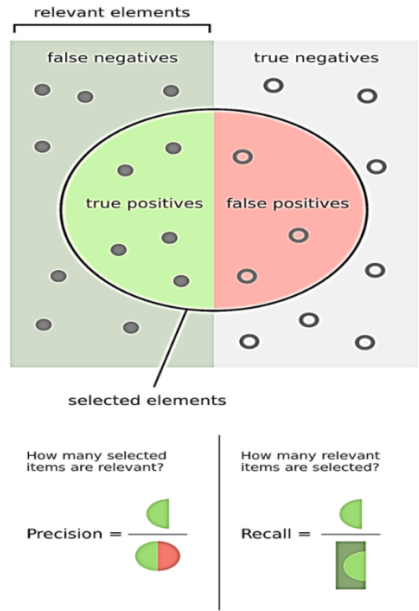


Figure 3.4 Illustration of *Precision* and *Recall* metrics [48].

As recall ↑ precision ↓  
conversely:  
As recall ↓ precision ↑

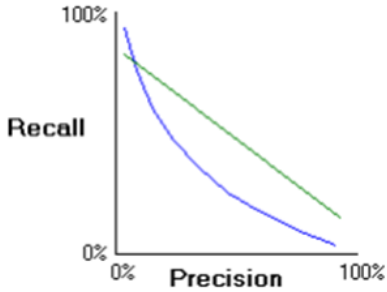


Figure 3.5 *Recall* and *Precision* metrics are related [48].

### 3.4. Machine learning in CBIR

Taking advantage of supervised machine learning algorithms is a possible approach in a CBIR system such that a classification model is trained based on previously labeled image features and then used to classify new input images. Figure 3.6 shows the flowchart of a CBIR system incorporating a supervised machine learning classifier to train the feature dataset (visual index). Training the system iteratively and saving the best model (offline phase) is expected to improve the efficiency of the image retrieval process. Indeed, during the online phase, the trained classifier is used to predict the class membership (class label) of any given query image. Then, all images from the predicted class are retrieved and ranked according to the similarity measure. Finally, the  $N$  first images (in ranked order) are considered as the *TopN* matches.

In 2015, Sandeep et al. [49] developed and put into practice an efficient feature extraction technique. They used various visual features that are extracted by applying a regression model and calculating feature weight by neural network. The next step is training the features with KNN. They applied different distances to the framework such as L1, L2, Manhattan, Chebychev, Cosine, Correlation, Spearman and Relative Deviation. The results are reported as precision and recall graphs according to the content of retrieved images from the datasets.

In 2016, Srinivas et al. [50] proposed a classification method for medical image retrieval using online dictionary learning (ODL) approach on edge-based features. During the learning phase, a dictionary is designed for each class of the training images. Classification consists on finding the sparsest dictionary for a given test image and assigning its associated class. The proposed method gives best classification results in comparison to other image classifiers such as linear discriminant analysis, naïve Bayes or kernel SVM.

In [31], the proposed CBIR system combines three low-level feature extraction techniques: color-correlogram for color feature extraction, wavelet transformation for extracting shape features and Gabor wavelet for texture feature extraction. Further, a support vector machine classifier (SVM) is trained to classify the features of a query image and distinguish between relevant and irrelevant images accordingly. This method gives better

performance than the traditional method of image retrieval. Achieved results are found encouraging in terms of color, shape and texture image classification *Accuracy*.

In [51], Savita et al. have investigated different methods of representing shape and texture in content-based image retrieval with machine learning technology. They have combined five types of image features in a training dataset and these are trained by SVM (support vector machine) algorithm. They combined histogram features, texture features (GLCM features), wavelet features, Gabor features, and statistical features. The Wang dataset [52] containing ten different image classes is used to extract all feature vectors for each image and store them in a new dataset so that SVM can use it for the training process and to classify the query image. By using these feature sets from Wang dataset [52], they reached 97.53% classification *Accuracy*.

In 2017, Kaur et al. [53] proposed a CBIR system based on statistical image features such as skewness, kurtosis and standard deviation, which are extracted from the probability color histogram of dataset images. The training phase is performed on Corel-1000 dataset [21] using one of the three classifiers: artificial neural network, naïve Bayes and neuro-fuzzy network. These retrieval approaches were compared on the basis of *Precision* and error rates. Neuro-fuzzy classifier outperforms other techniques on the experimented dataset.

Ali et al. [54] took advantage of scale-invariant feature transform (SIFT) for feature extraction from images. Subsequently, a neural network is trained using the extracted features. They used the optimization technique BFOA (bacteria foraging optimization algorithm) to reduce the complexity, cost, energy and time consumption. The considered image features are average value, minimum value and maximum value. The training is carried out using a deep neural network algorithm. This trained system when presented with a query image retrieves from the dataset images which are relevant and similar to the query. The results show a considerable improvement in terms of *Accuracy* ranging from 50% to 96% for different image classes in Caltech-101 dataset.

In 2018, Luke Toroitich et al. [55] presented their approach in CBIR using the combination of color, texture features and KNN. Euclidean distance metric is used to determine the nearest objects, thus resulting in the least number of

images retrieved by the system. Color and texture features are used to generate the feature matrices on which the image comparison is made. For KNN algorithm, different K numbers are tested to determine the best value for different classes of images.

In 2019, Al-Qasemy et al. [56] have also used the KNN supervised learning to train statistical features (mean, standard deviation and skewness) which are extracted from either RGB or grayscale image datasets. Image retrieval efficiency is measured by means of precision, recall and f-measure. The proposed feature selection method has made CBIR simpler and have resulted in high precision retrieval. In [19], Alrahha and Supreeti firstly proposed the local neighbor pattern (LNP) method for image feature extraction, which yields a better retrieval recall when compared with other methods such as local binary pattern (LBP). Secondly, the CBIR system is improved in terms of *Accuracy* using supervised machine learning such as SVM and KNN on three different image datasets.

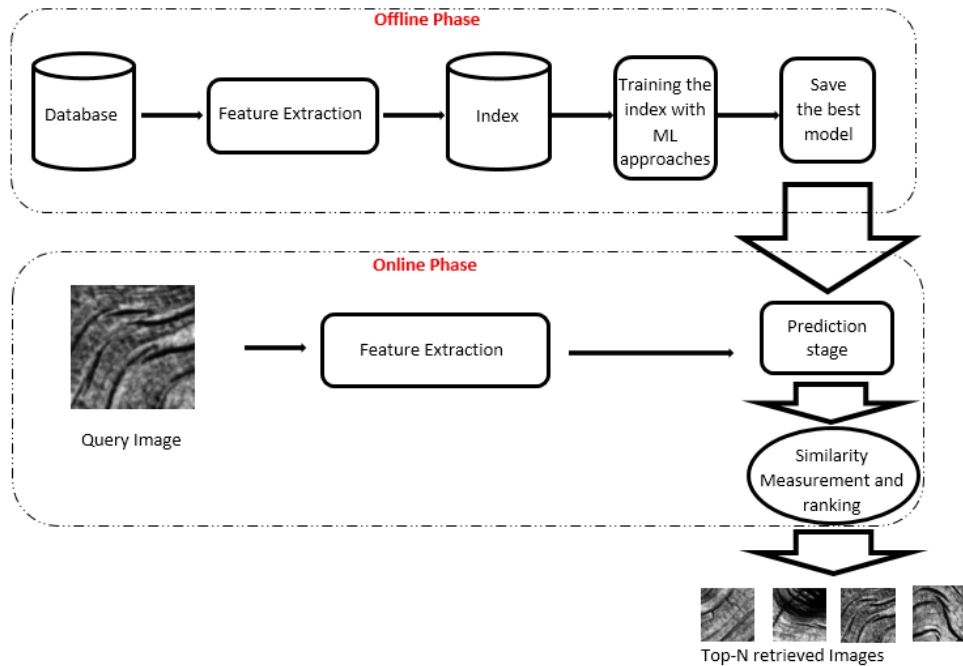


Figure 3.6 CBIR scheme with a classification approach.

#### 4. Region of interest (RoI) detection

In an image, a certain region with a distinctive characteristic with respect to other regions and which is intended to be identified and segmented for a specific purpose, is referred to as a Region of Interest (RoI). In medical images, tumors and abnormal tissues are examples of RoIs.

RoI detection in texture images and more specifically abnormality detection in images of human tissues has attracted a huge research interest in recent years. It includes both segmentation and classification of the abnormal region in a medical image. For this purpose, many existing methods take advantage of texture feature extraction techniques and machine learning solutions.

In 2015, Ketan Mashhale et al. [57] proposed a classification system to recognize normal and abnormal MRI brain images. They employed standard deviation, skewness, kurtosis, entropy, dissimilarity, inverse energy and contrast as feature extraction technique followed by a classifier, such as Support Vector Machine (SVM) or Hybrid Classifier (SVM-KNN) to classify images. They succeeded to improve the results up to an *Accuracy* of 98%. Ning Guo et al. [58] demonstrated the feasibility of performing SVM based lung cancer staging using multiple image features in PET/CT images. They used SUV mean features for PET images, entropy and uniformity features for CT images. Using SVM classifier with a combination of tumor heterogeneity and other effective features had great potential to augment diagnostic *Accuracy*.

In 2017, M.A. Rahman et al. [59] extracted features from images of skin lesion using color plane-histogram quality technique in order to classify them. More specifically, they used color thresholding method cascaded with interactive segmentation tool to segment lesion regions. Color features (RGB, HSV and YCbCr histogram features) and image quality analysis parameters are subsequently extracted from segmented lesion regions. Finally, KNN, SVM and SVM+KNN models were applied for classification.

In 2019, Ranijitha et al. [60] proposed an algorithm to identify bone malignant growth using k-means segmentation and KNN classifier. They evaluated the impact of a variety of feature extraction techniques including auto-correlation, cluster prominence, cluster shade, sum average, sum entropy, sum of square, variance and sum variance on the dataset. They further used the K-means

algorithm for segmentation and K-nearest neighbor for classification. They succeeded to achieve an *Accuracy* rate as high as 98.14% in their own selected images.

Tajkia Saima Chy et al. [61] performed a comparative analysis using classification with KNN, SVM and ELM (extreme learning machine) algorithms in order to detect sickle cell anemia in medical images. They used two types of features namely geometrical features and statistical features. In geometrical features, metric value and elongation are applied. In the statistical part, they took advantage of mean, standard deviation, variance, skewness and kurtosis calculations. Comparing the results in terms of precision, sensitivity, specificity, *Accuracy*, f-score and computation time, they concluded that ELM outperforms other techniques.

## 5. Deep learning

Despite the overall success of using classical feature extractors such as wavelet decomposition followed by conventional classifiers as showed in section 3.4 of this chapter, there exist a list of factors encouraging the researchers to exploit deep learning based approaches and more specifically convolutional Neural Networks (CNNs) to develop a reliable solution.

The complex nature of medical images calls for a powerful tool for feature extraction, in other words there exist numerous subtle details in a medical image that might not be captured by a conventional feature extractor which are usually sensitive to specific type of features. Deep learning approaches basically train an architecture which learns extracting the appropriate features from images and thus achieve a higher performance.

The fast grow of Graphical Processing Units (GPU) as well as cloud computation platforms has eased training a deep architecture on large datasets of images.

Nowadays, pre-trained deep learning-based architectures over enormous datasets of images are available which can be adapted as the start point of training an abnormality detector. These architectures can be leveraged through transfer-learning to create a desired network by excessively reducing the computational cost.

In 2018 Xi et al. [62], [63] proposed a computer aided detection approach for classifying and localizing calcifications and masses in mammogram images. To improve on conventional approaches, they applied (CNN) for automatic feature learning and classifier building. They leveraged deep CNNs' hierarchical feature extraction capabilities through transfer learning. By using CBIS-DDMS dataset to evaluate their methods, the experimental results indicate that VGGNet among AlexNet, GoogleNet and ResNet, receives the best overall *Accuracy* at 92.53% in classifications.

In 2018, Khachnaoui et al [64] analyzed the computer-assisted diagnosis (CAD) system based on deep learning and focused on thyroid diagnose. They summarized the recent ultrasound thyroid CAD systems based on the deep Learning which used on different datasets.

Ronneberger et al. [65] show results for cell segmentation in light microscopy images from the ISBI cell tracking challenge 2015 [66]. They proposed U-net convolutional neural network on two datasets (PhC-U373 and DIC-HeLa). In training phase, 30 images (512x512 pixels) are collaborated as the input images and their corresponding segmentation maps to train the network with the stochastic gradient descent implementation of Caffe [67]. The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function. The U-net results show higher segmentation results compared to other methods (IMCB-SG, KTH-SE, HOUS-US).

U-net is a Fully Convolutional Network developed for the task of semantic segmentation. The architecture of U-net is composed of two main parts including a decoder module and a decoder module. The encoder module includes a stack of convolutional and pooling layers to compute a rich feature embedding from the input images. Such an embedding provides essential information about the object classes present in the image. The decoder module consists of transposed convolutional layers to up sample the feature map into the original size. This step is necessary to retrieve the location of objects in the original image which is crucial for semantic segmentation as the task aims to determine what pixels in the image belong to a certain object. The main advantage of a transposed convolution for up-sampling compared to regular interpolation techniques is that a transposed convolution consists of learnable

parameters which are adjusted during the training and contain the association information between the input and output.



## CHAPTER 4. Methodology

We are interested in performing color texture analysis by means of developing feature extraction methods that achieve powerful characteristic discrimination of texture images while providing feature representation with reduced dimensionality. The exploitation of multi-scale representations of texture images (such as the contourlet transform) together with parametric statistical modeling of both grayscale and color spaces in image data is the proposed approach for the texture feature extraction process. Consequently, we present new feature extraction methods using a new variant of the contourlet transform (RCT-Plus) which is rich in directional information and applicable to grayscale and multi-channel color spaces. Taking advantage of the Generalized Gaussian distribution (GGD), the multivariate Gaussian Mixture Model (GMM) and the multi-scale RCT-Plus decomposition reveals an improved level of information on texture representation. These choices are motivated by the fact that, on one hand, contourlet transform variants (such as the RCT-Plus) offer augmented and flexible directional frequency selectivity through image scales, which is meant to enhance texture characterization and extract richer directional information in both grayscale and color image textures. On the other hand, a hybrid approach combining GGD and GMM modeling on directional multi-scale image data is meant to be powerful in terms of fitting efficiency, flexibility in dealing with multi-channel data, feature compactness and easy adaptation of similarity metrics between texture features.

Moreover, the color texture features can be extracted according to two strategies: 1) a global region-based feature extraction by considering the entire image and 2) a local region-based feature extraction by considering the image blocks. These two strategies are justified by the fact that, in certain application domains, the global features are not able to capture the main characteristics of the entire image because of the presence of a variety of local textures.

To illustrate texture feature extraction according to these two strategies, two main applications are considered. The first application is an improved content-based image retrieval (CBIR) framework based on multi-scale color texture feature extraction and classification. The second application is a new abnormality detection method on color textures in medical skin lesion images.

In both applications, feature classification by means of supervised machine learning algorithms is a key technique that is incorporated into the processing system in order to achieve better performance in terms of retrieval or detection rates (respectively).

## 1. The RCT-Plus representation of grayscale textures

RCT-Plus is developed in this work as a new variant of the contourlet transform. It is meant to enhance and generalize the redundant contourlet transform (RCT). In addition to redundancy and pseudo-Gaussian filtering features inherited from RCT, the RCT-Plus offers augmented and flexible directional frequency selectivity which is meant to enhance texture characterization and extract richer directional information in the image [68]. RCT-Plus decomposition scheme, like in SCT and RCT, takes two main parts: a multi-scale decomposition followed by a directional filter bank (DFB). However, the main changes made to this scheme are as follows:

1. The multi-scale stage implements a redundant Laplacian pyramid (RLP) using low-pass filtering only. All sub-sampling operations are discarded in order to eliminate aliasing artefacts and provide additional redundancy amount in RLP sub-images.
2. Linear phase low-pass filters with pseudo-Gaussian properties are used to build the redundant Laplacian pyramid. Filter impulse responses  $h_a$  are given in equation (3.1). Increasing values of the factor  $a$  (with  $a=2, 4, 8, 16\dots$ ) decrease the filter passband, i.e., the filter becomes sharper, filtering out higher frequencies. The magnitude of the frequency response for pseudo-Gaussian filters is shown in Figure 4.2, Figure 4.3 and Figure 4.4. One can notice how the filter shape and the filter passband change by modifying the value of  $a$ .
3. To build a redundant Laplacian pyramid having  $L$  scale levels,  $L$  pseudo-Gaussian filters (with  $a=2^l, l=1\dots L$ ) are used to generate  $L+1$  equal-size sub-images:  $L$  detail sub-images ( $RLP_l$ ) and one low-pass image approximation  $C_L$ . Therefore, the redundancy factor of this

decomposition is  $L+1$ . The corresponding decomposition diagram is detailed in Figure 4.1.

- At the DFB stage, all input RLP sub-images are of the same size as the original image since the redundant Laplacian pyramid is not down-sampled. A 2-D filter bank (*pkva* filters) with  $D_l$  orientations and critical sub-sampling is applied on each RLP sub-image at scale level  $l$  to obtain a number  $D_l$  of directional sub-bands  $\{C_{ld}; l = 1 \dots L; d = 1 \dots D_l\}$ , also designated as detail sub-bands. The allowed values for  $D_l$  are in the set  $\{2, 4, 8 \dots\}$  and can vary from one scale level to another of the RLP pyramid. Due to the critical-subsampling at the DFB stage, the total size of the generated directional sub-bands at each scale level  $l$  is equal to the size of the input  $RLP_l$  sub-image. The frequency partition in both RLP and DFB stages are shown in the diagram of Figure 4.5.

When RCT-Plus is generated with four directional sub-bands ( $D_l$  fixed at 4) at each scale level, the resulting decomposition is designated as a redundant contourlet transform (RCT) [10]. In other words, RCT is a specific case of RCT-Plus. Indeed, RCT-Plus and RCT share the properties of Gaussian filtering and redundancy in the multi-scale stage. In addition, RCT-Plus has augmented directional selectivity. The redundancy factor for both RCT-Plus and RCT is  $L+1$ . Figure 4.6 illustrates an example of RCT-Plus decomposition of a grayscale texture image.

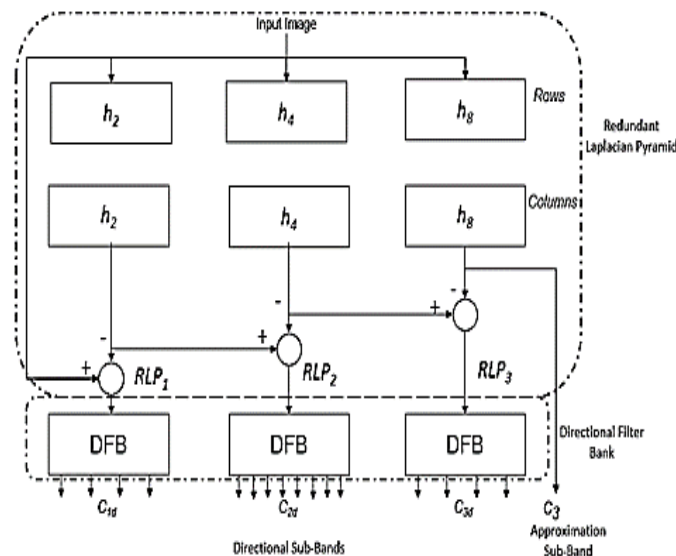


Figure 4.1 Block diagram of RCT-Plus decomposition (with 3 scale levels).

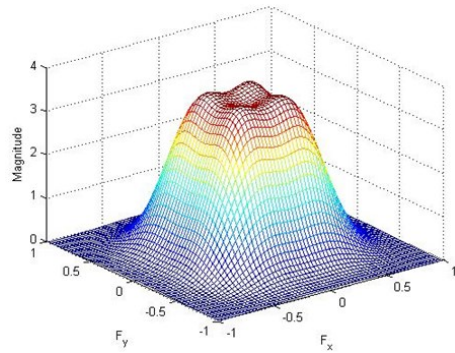


Figure 4.2 Frequency response of the 2-D pseudo-Gaussian filter for  $a=2$ .

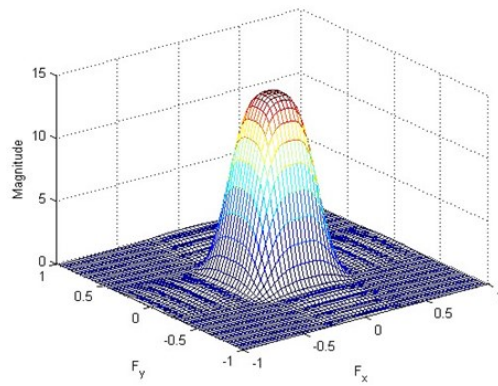


Figure 4.3 Frequency response of the 2-D pseudo-Gaussian filter for  $a=4$ .

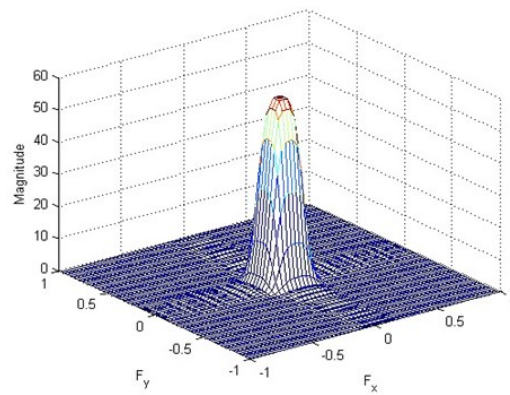


Figure 4.4 Frequency response of the 2-D pseudo-Gaussian filter for  $a=8$ .

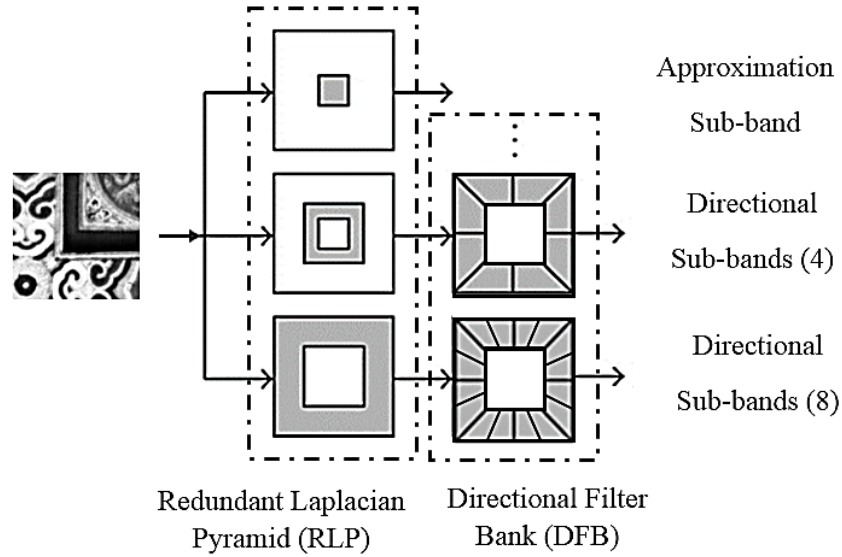


Figure 4.5 A block diagram of the RCT-Plus decomposition illustrating an example of frequency partition by the RLP stage and the DFB stage.

Original Texture Image $N \times N$								
Level 1 $D_1 = 4$	 $N/2 \times N/2$	 $N/2 \times N/2$	 $N/2 \times N/2$	 $N/2 \times N/2$				
Level 2 $D_2 = 8$	 $N/4 \times N/2$	 $N/4 \times N/2$	 $N/4 \times N/2$	 $N/4 \times N/2$	 $N/2 \times N/4$	 $N/2 \times N/4$	 $N/2 \times N/4$	 $N/2 \times N/4$
Level 3 $D_3 = 4$	 $N/2 \times N/2$	 $N/2 \times N/2$	 $N/2 \times N/2$	 $N/2 \times N/2$				
Approximation sub-band	 $N \times N$							

Figure 4.6 RCT-Plus decomposition of a  $N \times N$  grayscale texture image. The number of scale levels is  $L=3$ . The number of directional sub-bands at each scale level  $l$  is indicated by  $D_l$ , thus  $[D_1, D_2, D_3] = [4, 8, 4]$ . The total number of RCT-Plus coefficients is  $4 \times N \times N$  and the redundancy factor is 4.

## 2. Statistical modeling of the RCT-Plus

Generally speaking, parametric statistical modeling can be defined as the process of applying statistical analysis to a dataset to obtain a mathematical representation of the data under specific assumptions. The statistical representation of the data helps better interpret and understand the characteristics of the dataset [69]. Accordingly, many of the texture feature extraction techniques in the literature rely on performing parametric statistical modeling over the entire image or a subset of information in an image. By leveraging parametric statistical modeling techniques, we can compute efficient texture image signatures with low dimensions.

In this section, we take advantage of well-known parametric statistical modeling approaches including General Gaussian Density, multivariate Mixture of Gaussians as well as energy-based modeling. We then apply them to various image data representations such as grayscale, color or sub-band data (as computed by multi-scale image decomposition) to analyze image textures and extract relevant and compact features.

### 2.1. General Gaussian density modeling (GGD)

The multi-scale GGD approach for texture feature extraction consists in characterizing the statistical distribution of sub-band coefficients through multi-scale representations of a grayscale image. According to [11] [10], a good probability density function (PDF) approximation for the marginal density of coefficients at a particular sub-band produced by various types of wavelet transforms may be achieved by adaptively varying two parameters (the scale and the shape) of the generalized Gaussian density function (GGD). In the case of the contourlet image transform, the main assumption is that a single generalized Gaussian density function (GGD) can capture the shape of the data distribution in each contourlet sub-band. The PDF of the generalized Gaussian distribution is defined as:

$$P(x; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})} e^{-(\frac{|x|}{\alpha})^\beta} \quad (4.1)$$

where  $\alpha$  and  $\beta$  are the scale and shape parameters of the GGD respectively, and  $\Gamma$  is the Gamma function defined as:

$$\Gamma(x) = \int_0^{\infty} e^{-t} t^{x-1} dt, \quad x > 0. \quad (4.2)$$

The scale parameter  $\alpha$  controls the dispersion of the PDF while the shape parameter  $\beta$  determines whether it is peaked or flat. This gives the PDF the flexibility to fit the shape of heavy-tailed data.

Given a multi-scale decomposition of an image  $I$ , yielding  $L$  scale levels and  $D_l$  directional (detail) sub-bands  $C_{ld}$  at each scale level, we have modeled the marginal distribution for each sub-band by a GGD function and then a model parameter estimator, namely *Moment Matching* estimator (*MM*) or *Maximum Likelihood* estimator (*ML*) is applied to estimate the scale and the shape parameters ( $\alpha$  and  $\beta$ ) of the fitting GGD. Once  $\alpha$  and  $\beta$  values for each sub-band of an image are estimated, they are concatenated to form the image feature vector  $F_I$ :

$$F_I = \{\alpha(l, d), \beta(l, d); \quad l = 1 \dots L; \quad d = 1 \dots D_l\}. \quad (4.3)$$

To take into account the GGD modeling of the image approximation, one can normalize it (to zero mean and unit variance) and one can consider it as an additional sub-band (with  $l=L+1; D_{L+1}=1$ ), which gives a feature vector  $F_I = \{\alpha(l, d), \beta(l, d); \quad l = 1 \dots L + 1; \quad d = 1 \dots D_l\}$ , thus adding the estimated scale  $\alpha(L+1, 1)$  and shape  $\beta(L+1, 1)$ .

Kullback-Leibler divergence (KLD) and Euclidean distance are two mostly used metrics to evaluate similarity between two GGD feature vectors [11]. Indeed, the KLD is a statistical measure of how different two probability distributions over the same event space are. The KLD between parameters of marginal distributions is used for the similarity measurement among images. The closed form for the KLD between two GGDs ( $P_1$  and  $P_2$ ) is expressed as follows:

$$KLD(P_1, P_2) = \text{Log} \left( \frac{\beta_1 \alpha_2 \Gamma(\frac{1}{\beta_2})}{\beta_2 \alpha_1 \Gamma(\frac{1}{\beta_1})} \right) + \left( \frac{\alpha_1}{\alpha_2} \right)^{\beta_2} \cdot \frac{\Gamma(\frac{\beta_2+1}{\beta_1})}{\Gamma(\frac{1}{\beta_1})} - \frac{1}{\beta_1} \quad (4.4)$$

Equation (4.4) expresses that the divergence between the GGDs,  $P_1$  and  $P_2$ , only depends on  $\alpha_1, \alpha_2, \beta_1$  and  $\beta_2$  parameter values which yields to easily approximate the KLD. Moreover, the KLD of two identical GGDs is zero but KLD is not a symmetric measure since  $KLD(P_1, P_2) \neq KLD(P_2, P_1)$ . Once the divergence between all multi-scale sub-band pairs of two images are calculated, the summation of all KLD values will be considered as the divergence measure between two images.

Consequently, the similarity measurement between two images  $T$  and  $Q$  requires the computation of KLD measures between GGD probability distributions. It is formulated as the overall symmetric divergence  $\delta_{TQ}$  between all sub-band GGDs:

$$\delta_{TQ} = \frac{1}{2} \sum_{l=1}^L \sum_{d=1}^{D_l} KLD(P_T^{ld}, P_Q^{ld}) + \frac{1}{2} \sum_{l=1}^L \sum_{d=1}^{D_l} KLD(P_Q^{ld}, P_T^{ld}) \quad (4.5)$$

where  $P_T^{ld}$  and  $P_Q^{ld}$  represent the statistical models estimated in the two images  $T$  and  $Q$ , for the sub-band of the  $l$ -th scale level and the  $d$ -th direction, respectively. Also, the similarity measure  $\delta_{TQ}$  has positive values with  $\delta_{TQ} = \delta_{QT}$  and  $\delta_{TT} = 0$ .

In case where the GGD modeling extends over the image approximation sub-band, one can consider an additional KLD in (4.5) with  $l=L+1; D_{L+1}=1$ , which measures the divergence between the two GGDs corresponding to the image approximations of  $T$  and  $Q$ . The criterion  $\delta_{TQ}$  is then expressed as:

$$\delta_{TQ} = \frac{1}{2} \sum_{l=1}^{L+1} \sum_{d=1}^{D_l} KLD(P_T^{ld}, P_Q^{ld}) + \frac{1}{2} \sum_{l=1}^{L+1} \sum_{d=1}^{D_l} KLD(P_Q^{ld}, P_T^{ld}) \quad (4.6)$$

Note that Euclidean distance (ED) between the feature vectors  $F_Q$  and  $F_T$  is another alternative for similarity measurement  $\delta_{TQ} = ED(F_T, F_Q)$  between the images  $T$  and  $Q$ .

## 2.2. Energy-based modeling

Multi-scale energy-based approach for texture feature extraction includes calculating energy ( $L^1$  norm,  $L^2$  norm or some combination of both) and characterizing its distribution through multi-scale sub-band images [6], [70]. The energy-based approach assumes that different texture patterns have



different energy distribution in the space-frequency domain. This approach is very appealing due to its low computational complexity involving mainly the calculation of first and second order moments of sub-bands coefficients. Given a multi-scale image representation yielding one image approximation,  $L$  scale levels and  $D_l$  directional sub-bands  $C_{ld}$ . at each level  $l$ , a feature vectors  $E_1$  (or  $E_2$  respectively) is formed as follows:

$$E(l, d) = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M |C_{ld}(i, j)| \quad (4.7)$$

$$E'(l, d) = \sqrt{\frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M [C_{ld}(i, j)]^2} \quad (4.8)$$

$$E_1(l, d) = \left( E(l, d), \sqrt{E(l, d)^2 + E'(l, d)^2} \right) \quad (4.9)$$

$$E_2(l, d) = (E(l, d), E'(l, d)) \quad (4.10)$$

$$E_1 = \{E_1(l, d); \quad l = 1 \dots L + 1; \quad d = 1 \dots D_l\} \quad (4.11)$$

$$E_2 = \{E_2(l, d); \quad l = 1 \dots L + 1; \quad d = 1 \dots D_l\} \quad (4.12)$$

Note that  $M \times N$  indicates the size of the considered sub-band  $C_{ld}$ . The approximation sub-band is denoted by  $l=L+1$  and  $D_{L+1}=1$ ; For similarity measurement, the Euclidean distance is computed between two compared feature vectors.

### 2.3. Joint color texture modeling

In addition to texture characteristics in the grayscale space, color information is a significant part of the physical properties of an image. Therefore, it is commonly exploited to enhance the discriminative power of texture features in characterizing natural images. Computation of a standard histogram on multi-channel color spaces (RGB, YCbCr, HSV, ...) is simple but may result into a huge amount of data which is a drawback to the compactness of the feature vector. By modeling color information with parametric statistical approaches, one can achieve significant improvement in the extraction of relevant and compact color features. Then, considering both texture and color texture features, there are many different strategies to perform their

combination in the multi-scale space (such as the RCT-Plus image representation), which can result in a variety of feature extraction methods.

For instance, a straightforward way consists in applying a multi-scale decomposition and GGD modeling (4.3) to the multi-channel color spaces (RGB, YCbCr, HSV, ...), once to each color channel  $c$  separately, and then concatenating the resulting  $N_c$  feature vectors as follows:

$$F_l = \{\alpha(l, d, c), \beta(l, d, c); l = 1 \dots L + 1; d = 1 \dots D_l; c = 1 \dots N_c\} \quad (4.13)$$

For similarity measurement between two color images  $T$  and  $Q$ , the  $\delta$  criterion defined in (4.6) is first applied to each pair of color channels  $c$  to yield  $\delta_{TQ}^c$ , then, the summation is performed to yield the overall symmetric divergence between  $T$  and  $Q$ .

$$\delta_{TQ} = \sum_{c=1}^{N_c} \delta_{TQ}^c \quad (4.14)$$

In particular, when the data is multimodal and the number of data channels is large (such as in RGB texture images), the multivariate Gaussian mixture modeling and representation is powerful in terms of fitting efficiency, model compactness and flexibility in adding more data channels. In the following section we introduce the basic principles of a multivariate Gaussian Mixture Model (GMM) in the color image space and express an approximated similarity metric based on the Kullback-Leibler divergence.

#### 2.4. Gaussian Mixture Modeling (GMM)

A multivariate random variable  $x = (x_1, x_2, x_3, \dots, x_n)^T$  follows a Gaussian distribution if its probability density function [71] can be expressed as:

$$\mathcal{N}(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right) \quad (4.15)$$

where  $\mu$  is a mean vector that defines its center and  $\Sigma$  is a covariance matrix that defines its width. If the random variable  $x$  is  $n$ -dimensional then  $\mu$  is a  $n$ -dimensional vector and  $\Sigma$  is a  $n \times n$  matrix.

For modeling multimodal data, a Gaussian Mixture Model (GMM) that contains a finite number  $k$  of multivariate Gaussians can be expressed as:

$$P(x) = \sum_{i=1}^k \omega_i \mathcal{N}(x: \mu_i, \Sigma_i) \quad (4.16)$$

where each Gaussian function is identified by  $i \in \{1, \dots, k\}$  and the prior probability  $\omega_i$  of the  $i$ th Gaussian (mixing parameter) must meet the conditions  $\sum_{i=1}^k \omega_i = 1$  and  $0 \leq \omega_i \leq 1$ .

To estimate the parameters of a  $k$ -component GMM that fits a given set of training data  $I$ , the *iterative Expectation-Maximization* (EM) algorithm is usually applied to yield a parameter vector (feature vector)  $F_I$ :

$$F_I = \{\omega_i, \mu_i, \Sigma_i; \quad i = 1 \dots k\} \quad (4.17)$$

For example, if we apply a GM model with  $k=2$  to an RGB image, we will have a  $2 \times 3$  matrix for the mean and a  $3 \times 3 \times 2$  covariance matrix, which is symmetric (See Table 4.1). Also, we have 2 mixing parameters  $\omega_1$  and  $\omega_2$  that add to one. By taking into account the repeating values, one can reduce the covariance matrix to 12 values and keep only one mixing parameters. Therefore, the feature vector size will be 19 (instead of 26 elements).

In the case of a univariate Gaussian model, the GMM with  $k=2$  results in 5 elements feature vector;  $\omega$ ,  $\mu$ , and  $\Sigma$  with 1,  $1 \times 2$ , and  $1 \times 1 \times 2$  elements respectively (see Table 4.1).

Random variable	Number of Gaussians ( $k$ )	Mean ( $\mu_i$ )	Covariance ( $\Sigma_i$ )	Mixing parameters ( $\omega_i$ )
Multivariate (RGB)	$k=2$	R1 G1 B1	$a1 \ b1 \ c1$ $b1 \ d1 \ e1$ $c1 \ e1 \ f1$ Instead of 9, we keep 6 values	$\omega_1$
		R2 G2 B2	$a2 \ b2 \ c2$ $b2 \ d2 \ e2$ $c2 \ e2 \ f2$ Instead of 9, we keep 6 values	$\omega_2$
Univariate	$k=2$	$I1$	$a1$	$\omega_1$
		$I2$	$a2$	$\omega_2$

Table 4.1 Illustration of the estimated parameters for GM model with  $k=2$ .

As a similarity measure between two Gaussian mixture models, a Kullback-Leibler divergence (KLD) is commonly approximated because no closed form expression exists [72] [73]. Using an approximation method based on nearest pairs of Gaussians [74] [75], the symmetrical KLD between two GMM distributions  $P(x) = \sum_{i=1}^{k_1} \omega_{p_i} \mathcal{N}(x: \mu_{p_i}, \Sigma_{p_i})$  and

$Q(x) = \sum_{j=1}^{k_2} \omega_{q_j} \mathcal{N}(x: \mu_{q_j}, \Sigma_{q_j})$  is formulated as follows:

$$\delta_{PQ} = \frac{1}{2} \sum_{i=1}^{k_1} \omega_{p_i} \min_{j=1..k_2} \left\{ KLD(\mathcal{N}_{p_i}, \mathcal{N}_{q_j}) \right\} + \frac{1}{2} \sum_{j=1}^{k_2} \omega_{q_j} \min_{i=1..k_1} \left\{ KLD(\mathcal{N}_{q_j}, \mathcal{N}_{p_i}) \right\} \quad (4.18)$$

where the KLD between two  $n$ -dimensional multivariate Gaussians  $\mathcal{N}(x: \mu_{p_i}, \Sigma_{p_i})$  and  $\mathcal{N}(x: \mu_{q_j}, \Sigma_{q_j})$  has a well-known closed form that is given by:

$$KLD(\mathcal{N}_{p_i}, \mathcal{N}_{q_j}) = \frac{1}{2} \text{Log} \frac{|\Sigma_{q_j}|}{|\Sigma_{p_i}|} + \frac{1}{2} \text{Tr} \left( (\Sigma_{q_j})^{-1} \Sigma_{p_i} \right) + \frac{1}{2} (\mu_{p_i} - \mu_{q_j})^T (\Sigma_{q_j})^{-1} (\mu_{p_i} - \mu_{q_j}) - \frac{n}{2} \quad (4.19)$$

Equations (4.18) and (4.19) express that the similarity measure between two GMMs,  $P$  and  $Q$ , only depends on model parameter values  $\mu$ ,  $\Sigma$  and  $\omega$ . Also,  $T$  is the *Transpose* operator,  $Tr$  is the *Trace* matrix operator and  $|\cdot|$  is the *Determinant*. Moreover, it should be noted that the similarity criterion  $\delta_{PQ}$  as defined in (4.18) is symmetrical ( $\delta_{PQ} = \delta_{QP}$ ), has positive values and equals zero for two identical GMMs ( $\delta_{PP} = 0$ ).

For texture analysis purposes, multi-channel spaces in color texture images (RGB, YCbCr, HSV, ...) are examples of suitable 3-dimensional datasets for GMM fitting. Figure 4.7 and Figure 4.8 illustrate the capacity of the  $k$ -component GM model ( $k=2$ ) to discriminate various RGB textures while providing compact representations.

Figure 4.9 illustrates the comparison between two statistical models (GGD and GMM) applied to the modeling of the RCT-Plus approximation sub-band corresponding to the red color component of RGB images. (univariate random variable). As Figure 4.9 confirms, GMM is a more accurate fit for the approximation sub-bands especially in the case

of multimodal histograms (having more than one peak). Figure 4.10 shows another example of GGD modeling where RCT-Plus is applied to the grayscale version of the image with 3 scale levels of decomposition and 8 directional sub-bands at each level. GGD gives a more appropriate fit for the directional sub-bands.

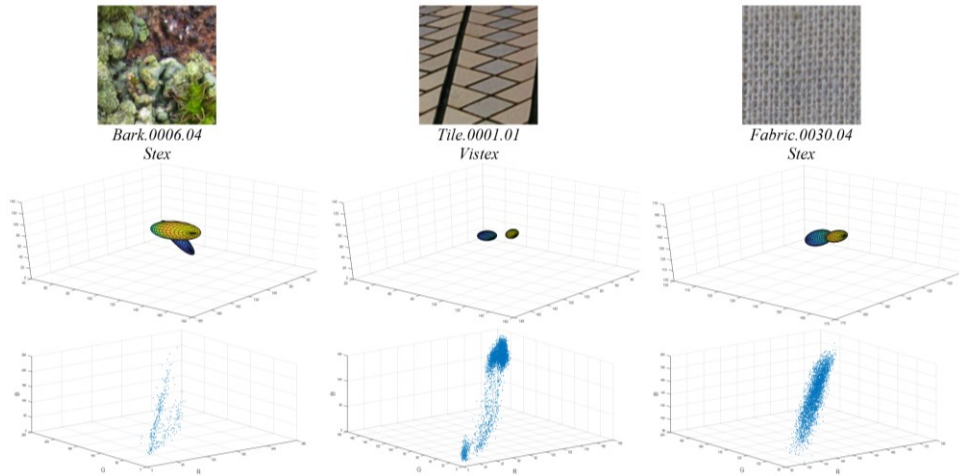


Figure 4.7 Three different color texture images in RGB format from the VisTex and Stex datasets [7] [25]. The second row illustrates the corresponding Gaussian mixture models with  $k=2$ , and the third row illustrates their corresponding RGB histogram.

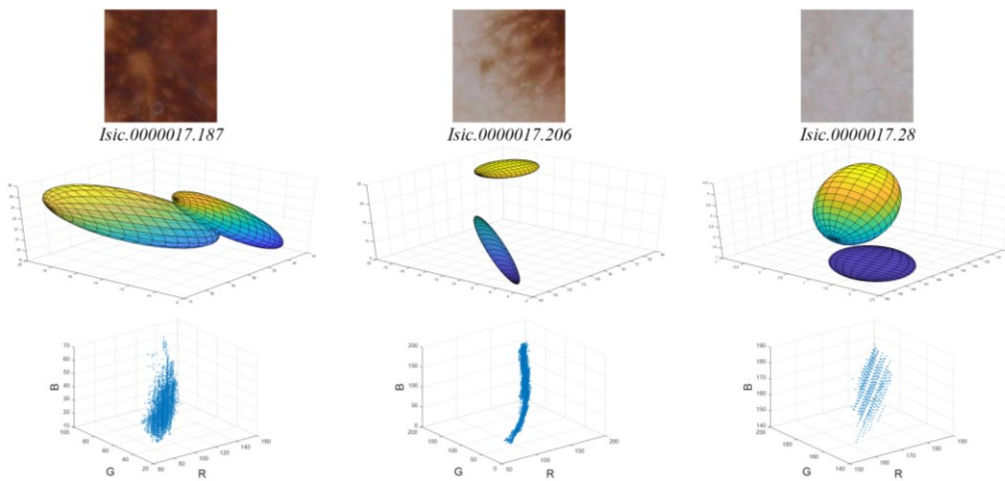


Figure 4.8 Three different color texture blocks in skin images from the ISIC dataset [40]. The second row illustrates the corresponding Gaussian mixture models with  $k=2$ , and the third row illustrates their corresponding RGB histogram.

Original image

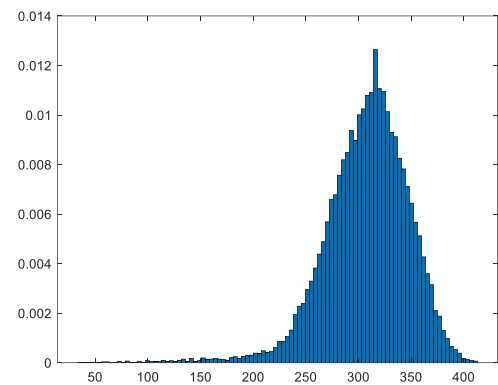
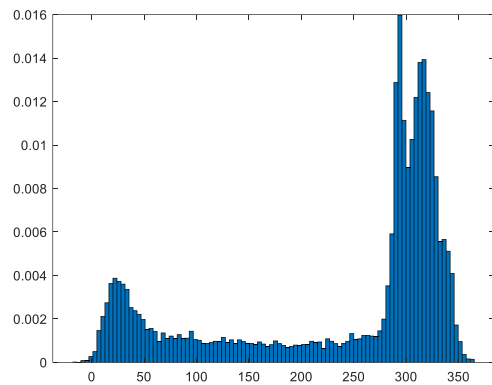


*Tile.0001.05 (VisTex-40)*

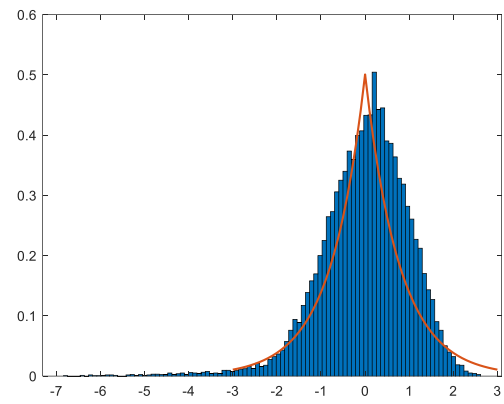
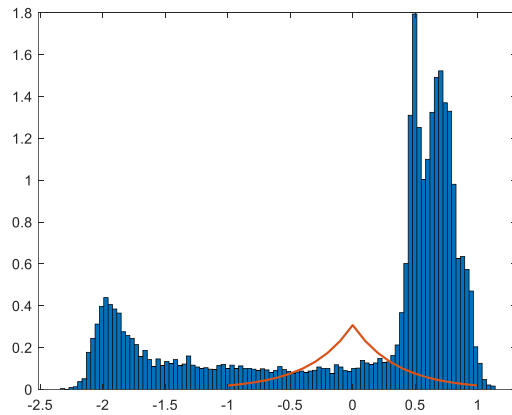


*Stone.0028.11 (Stex)*

Approximation  
sub-band  
histogram



Approximation  
sub-band  
histogram and  
it's fitting  
GGD



Approximation  
sub-band  
histogram and  
it's fitting  
GMM

$k=2$

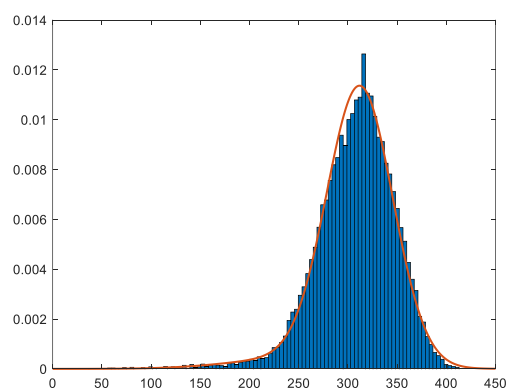
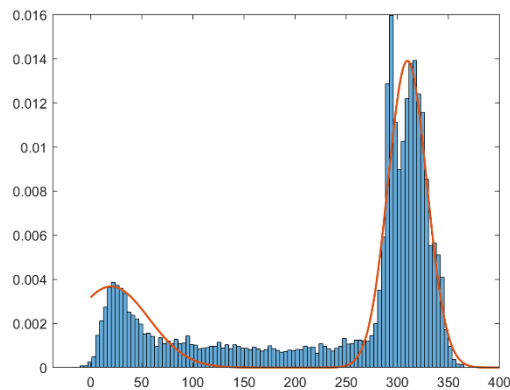


Figure 4.9 GGD and GMM fitting examples applied on RCT-Plus approximation sub-band of the red color image component. The images are selected from VisTex and Stex datasets [7] [25].

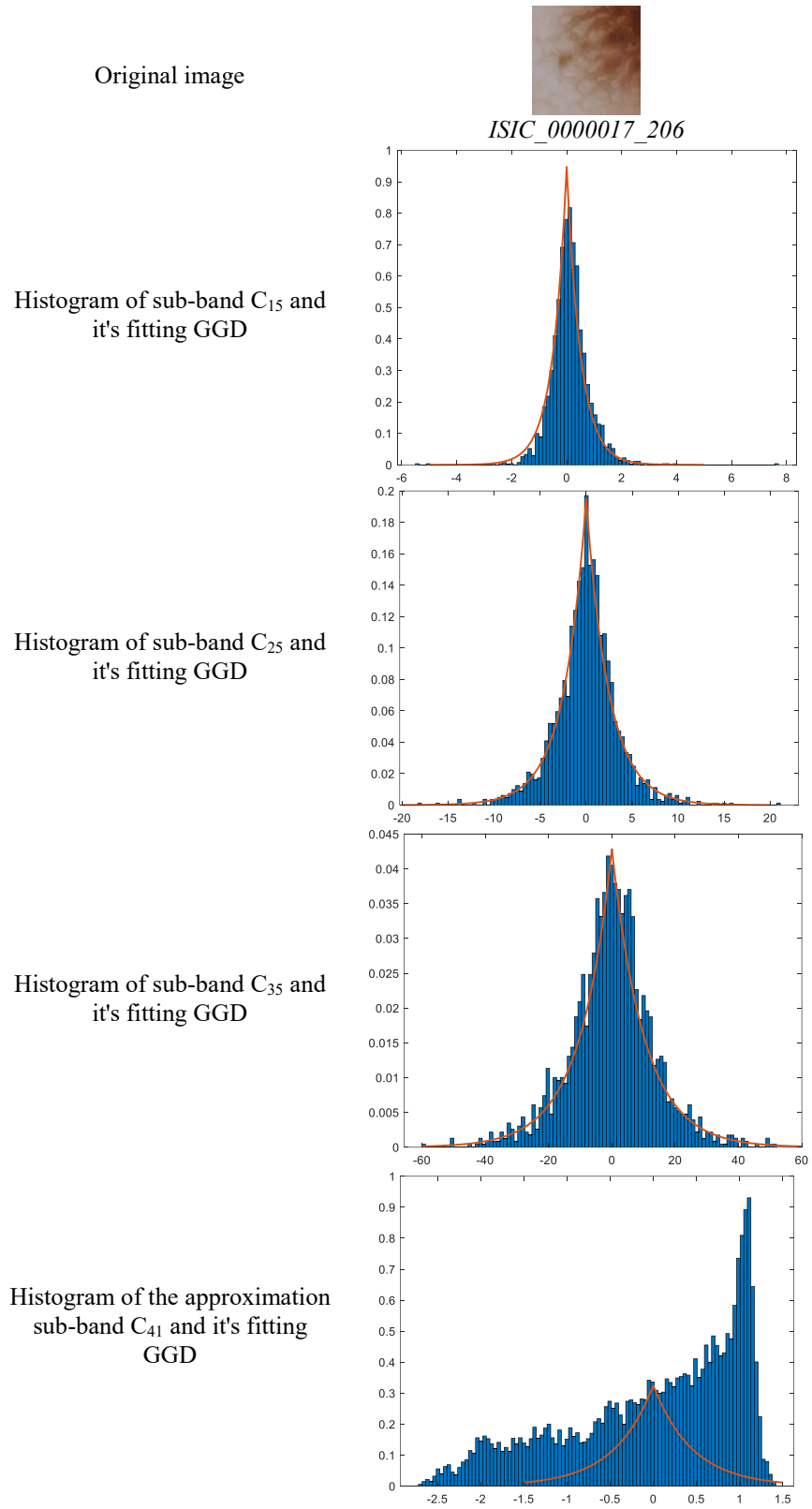


Figure 4.10 GGD fitting examples applied on an RCT-Plus decomposition with ( $L=3$  and  $D=[8 \ 8 \ 8]$ ). The image is selected from ISIC dataset [40].

### 3. A CBIR system based on color-texture discrimination

CBIR for texture images is essential for several applications such as remote sensing, industrial inspection, quality control, medical imaging, to name a few. Usually, the performance of CBIR systems depends largely on the way the images are represented to facilitate their indexation and the type of similarity measurement used to compare images. In fact, a typical image retrieval system will be composed of two main steps: 1) feature extraction and modeling which transforms an image into a compact feature vector, 2) similarity measurement, which is used to compare images and return the closest ones to the query image. However, there are remaining challenges for achieving accurate and computationally efficient color texture retrieval mainly due to the sheer number of texture classes and the variability inside each texture class. In this work, we introduce an improved CBIR framework for grayscale and color texture images. Building on the new proposed methods for texture modeling and feature extraction in Section 2, we propose to extract richer discriminative features from texture images and adapt similarity metric to the parametric statistical modeling of the contourlet representation for grayscale and color textures. Moreover, we introduce a new framework integrating a supervised machine learning approach (ML-CBIR). A query is then first classified to select the best texture class after which the retained class images are ranked to select top ones. This ML-CBIR framework is intended to improve the retrieval score and make it computationally efficient by avoiding the requirement of comparing a query against all images in the dataset.

#### *3.1. The proposed approach for CBIR*

As previously discussed, the CBIR system is mainly made up of image feature extraction as feature vectors followed by similarity measurement between two feature vectors for comparison purposes. Various CBIR schemes are proposed. Each of them combines one feature extraction method (as defined in Section 2) and a suitable similarity measurement from the following:

- 1)  $E_1/E_2$ : Energy-based feature extraction in (4.11) or (4.12) and similarity metric using Euclidean distance.



- 2) GGD2-ED: GGD modeling (4.3) of all image sub-bands estimated by *Maximum Likelihood (ML)* and similarity measurement using Euclidean distance.
- 3) GGD1/GGD2: GGD modeling (4.3) of all image sub-bands estimated by *Moment Matching (MM)* or *Maximum Likelihood (ML)* and similarity metric using the symmetric KLD in (4.6).
- 4) GGD1/GGD2 for color images: GGD modeling (4.13) of all color image sub-bands estimated by either *MM* or *ML* and similarity metric using the symmetric KLD in (4.14).
- 5) GGD1/GGD2+GMapp: GGD modeling (4.13) of the color image detail sub-bands estimated by either *MM* or *ML*, modeling of the approximation sub-bands using GMM (4.17) and the sum of symmetric KLDs in (4.5) and (4.18) as a similarity metric.
- 6) GGD1/GGD2+GM: GGD modeling (4.13) of the grayscale image detail sub-bands estimated by either *MM* or *ML*, modeling of the whole color image using GMM and the sum of the symmetric KLDs in (4.5) and (4.18) as a similarity metric.

It should be noted that the proposed CBIR schemes 4, 5) and 6) are applicable to color texture images (RGB, YCbCr, HSV, ...). Moreover, each feature extraction method can operate on a multi-scale image decomposition such as the RCT-Plus or one of the following contourlet variants:

1. RCT: Redundant Contourlet Transform;
2. SCT: Standard Contourlet Transform;
3. NSCT: Non-Subsampled Contourlet Transform.

In traditional content-based image retrieval, the general architecture is organized into two separate phases: the offline phase and the online phase. During the offline phase, a dataset index (feature dataset) is firstly constructed by extracting a feature vector  $F_I$  from each image in the dataset using one of the proposed CBIR schemes. All calculations are done offline so as not to penalize the user with often lengthy calculations. During the online search phase, a query image  $Q$  is given to the search system which calculates the associated feature vector  $F_Q$  and evaluates the distance (symmetric KLD or ED) between the query feature vector and every feature vector in the dataset index. Next, the  $N$  smallest distances in ascending order are then selected as *TopN* matches and the corresponding images are retrieved from the dataset

and presented to the user as being the most similar images to the query in the sense of the considered metric.

### *3.2. Integrating supervised learning in CBIR*

A supervised machine learning algorithm is incorporated into the CBIR scheme for classification purposes. Therefore, the main changes to the search and retrieval processes in the new ML-CBIR scheme are depicted in Figure 4.11.

During the offline phase, the extracted texture feature vectors from the image dataset are also labeled to indicate their class membership (image category) according to pre-existing texture classes. In addition to feature extraction and feature vector labeling (index), a training procedure is conducted on the labeled feature vector dataset in order to build a classification model with high *Accuracy*. The considered supervised ML algorithms in this work include K-Nearest Neighbors (KNN) and Support Vector Machines (SVM).

During the online phase, the search and retrieval process to any given query image is performed through query image classification. Firstly, the trained classifier is applied to the given query feature vector in order to predict its class membership (class label). Next, all feature vectors from the predicted class are compared to the query feature vector using a similarity metric (distances such as ED or symmetric KLD). The  $N$  smaller similarity indices are selected as the *TopN* matches. The corresponding images are retrieved in a ranked order and returned to the user as the most relevant texture images to the query (*TopN* retrieved images).

It's mentionable that, there is no such thing as the best classifier. It always depends on the context; what kind of data or problem is at hand. For example, it is known that KNN is slow when having a large-scale dataset, since it does not generalize over data in advance, it scans the index dataset each time a prediction is needed.

About the role of kernels in SVM, it is worth mentioning that the SVM algorithm can use a set of functions, namely kernels, to tackle the issue of inseparability in a dataset. The main idea is to take data as input and transform it into a specific feature space where the data points are separable. Linear,

nonlinear, polynomial, and radial basis functions (RBF) are some examples of common SVM kernels.

As previously mentioned, the main objective in SVM is to find a hyper-plane separating the data space while maximizing the margin (minimizing the error band) between the classes. Because the error band in the SVM is independent of the dimensionality of the space, the SVM does not require explicitly determining the mapping function.

Similarly, we exploit the KLD distance as the distance metric for the KNN classifier. Consequently, as mentioned earlier, in such a version of the KNN algorithm, the samples in the test data set are assigned to the class of samples with a minimum KLD value.

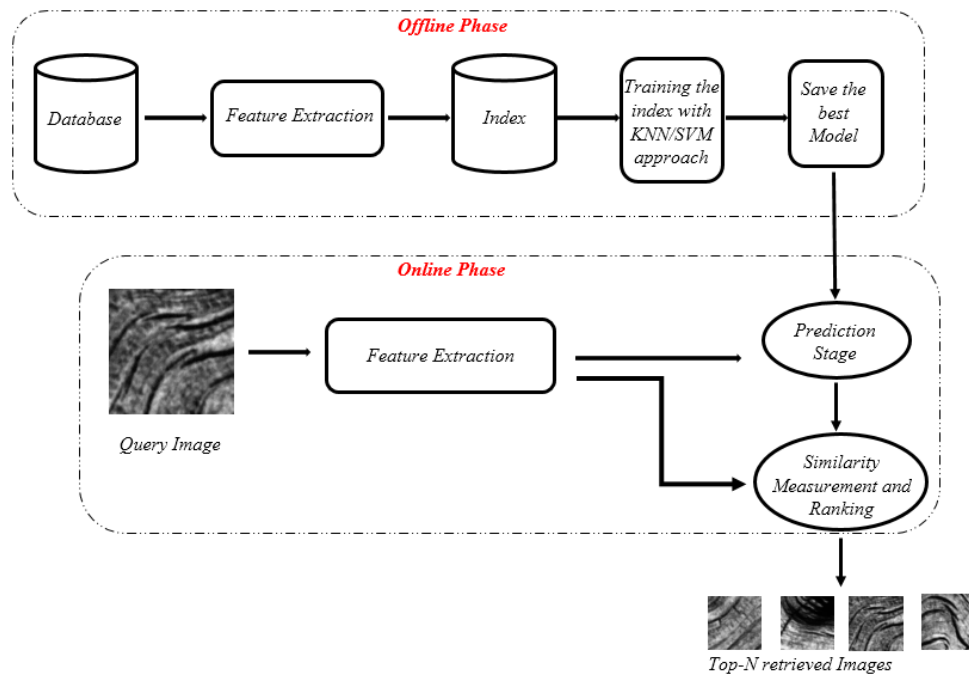


Figure 4.11 ML-CBIR scheme with KNN/SVM classifiers.

In addition to the incorporated ML approach in the proposed framework, the choice of a texture feature extraction method and adapted similarity metric play a key role in the efficiency of the texture retrieval process. Therefore, various ML-CBIR schemes are proposed. Each of them combines one feature extraction method (as defined in Section 2) and a suitable similarity measurement from the following:

- 1) GGD1/GGD2: GGD modeling (4.3) of all image sub-bands estimated by *Moment Matching (MM)* or *Maximum Likelihood (ML)* and similarity metric using the symmetric KLD in (4.6).
- 2) GGD1/GGD2 for color images: GGD modeling (4.13) of all color image sub-bands estimated by either *MM* or *ML* and similarity metric using the symmetric KLD in (4.14).
- 3) GGD1/GGD2+GMapp: GGD modeling (4.13) of the color image detail sub-bands estimated by either *MM* or *ML*, modeling of the approximation sub-bands using GMM (4.17) and the sum of symmetric KLDs in (4.5) and (4.18) as a similarity metric.
- 4) GGD1/GGD2+GM: GGD modeling (4.13) of the grayscale image detail sub-bands estimated by either *MM* or *ML*, modeling of the whole color image using GMM and the sum of the symmetric KLDs in (4.5) and (4.18) as a similarity metric.

The three mentioned methods in 2), 3) and 4) are for color texture retrieval. The corresponding ML-CBIR schemes are depicted in Figure 4.12, Figure 4.13 and Figure 4.14 respectively. The first schema in Figure 4.12 is applicable to multi-channel feature extraction. After separating the color channels, a multi-scale image decomposition (RCT-Plus) followed by a GGD modeling apply to each channel separately and all the extracted features from each color space are concatenated to create the main feature vector.

In the second schema (Figure 4.13), a mix of GGD and GMM modeling is used to extract color texture features. Indeed, after decomposing the image in each color space with RCT-Plus, each detail sub-band is modeled by GGD and the approximation sub-bands are jointly modeled using GMM. At the end, all the extracted features from all channels are concatenated to build the main feature vector.

The third schema-which is shown in Figure 4.14 indicates a compact and rich feature vector operating on both grayscale and color versions of the texture. Once RCT-Plus and GGD modeling are applied on grayscale texture image to extract the texture features, thereupon for color texture features, the GMM modeling applies to the color version of the texture image.

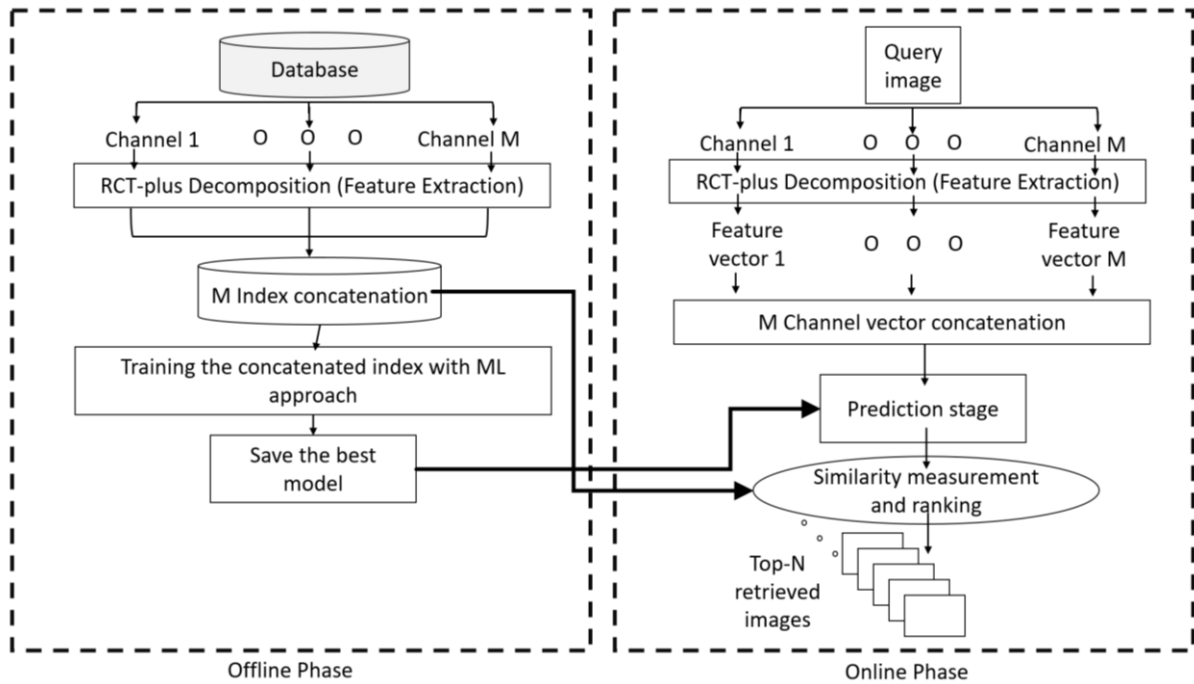


Figure 4.12 ML-CBIR using concatenated color texture features.

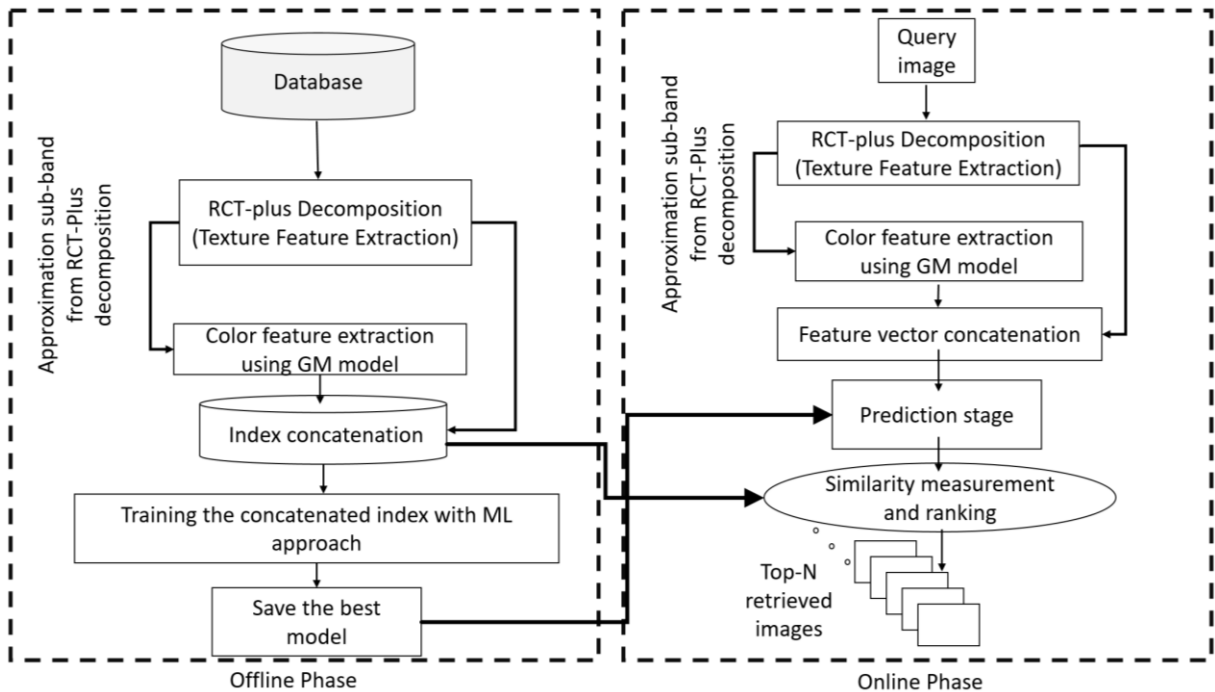


Figure 4.13 ML-CBIR using RCT-Plus grayscale texture features concatenated to color texture features which are extracted from the RCT-Plus approximation sub-bands using GM modeling.

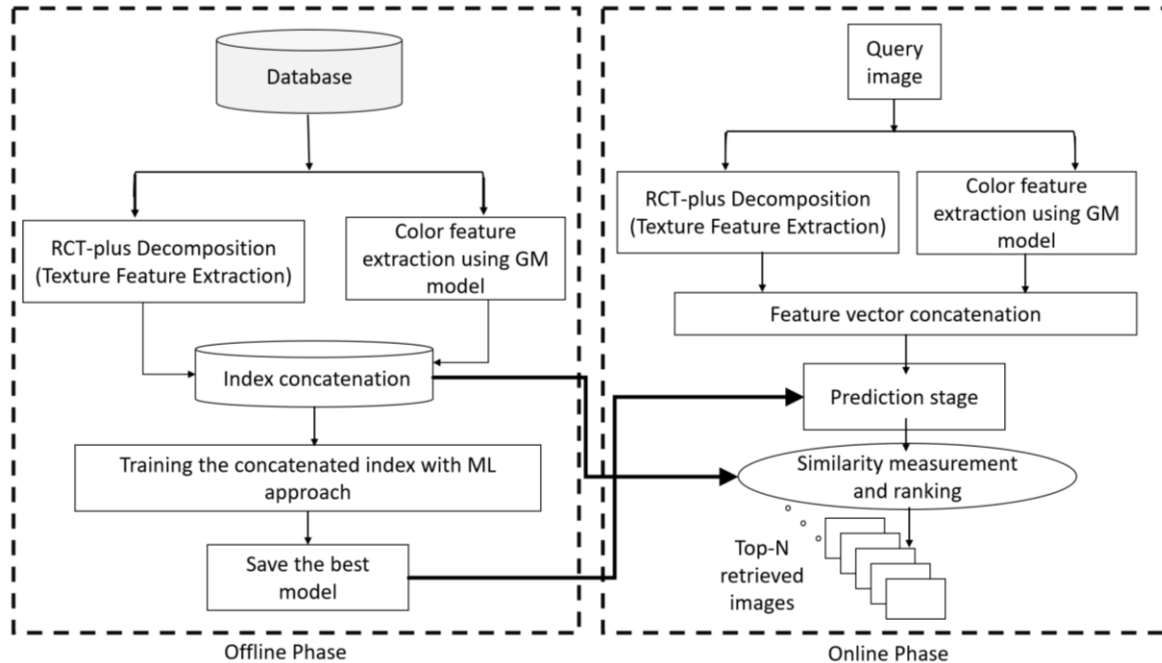


Figure 4.14 ML-CBIR using RCT-Plus grayscale texture features and a color texture feature using GM modeling of the whole image.

#### 4. Abnormality detection in color texture images

Identifying unexpected objects or substances in image datasets that are different from normal data content, refers to abnormality detection. It is obvious that anomaly instances are significantly different from normal instances. To discriminate among normal and abnormal regions in images we need to rely on some low-level image characteristics such as texture or color. By integrating texture and color texture features together, we aim to improve the overall performance of abnormality detection.

In the second part of this research, we apply the proposed methodology for texture and color texture feature extraction to the problem of abnormality detection in medical skin lesion images. We consider melanoma and nevus lesions in dermoscopic images of human skin as abnormal regions (*lesion*) against a background of *healthy* skin. Examples of dermoscopic images are shown in Figure 4.15. Therefore, the proposed approach for abnormality detection is a binary classification of texture and color texture features using supervised machine learning (such as SVM or KNN algorithms) followed by

a conciliation of detection results. This conciliation operation can be seen as a late procedure meant to perform a joint exploitation of texture and color texture features at the result level. Due to the fact that a variety of local textures may exist in the same image (melanoma lesion, healthy skin, skin moles, skin marking, ...) a local region-based strategy is adopted for feature extraction by considering image blocks that undergo RCT-Plus decomposition and parametric statistical modeling (GGD and GMM) as proposed in Section 2.1 and Section 2.4. Moreover, proper class labeling of image blocks is designed in order to enable efficient learning of image features.

It is worth mentioning that, image subdivision into blocks, in the one hand, must be performed with appropriate block size in order to ensure sufficient amount of data for contourlet decomposition and parametric statistical modeling. On the other hand, image subdivision into blocks with appropriate overlapping is a way to enhance the precise localization of the abnormal regions detected.

In Figure 4.16, the shown diagram presents the proposed framework for abnormality detection as a two-stage scheme corresponding to offline and online phases. Its detailed components are described in the following subsections.

#### *4.1. The labeling of local features*

We consider a dermoscopic image of human skin exhibiting a lesion (melanoma or nevus) as a color image containing three different textures: a) the lesion texture (*lesion*), b) the healthy skin texture (*healthy*), and c) the texture in the transition region between the lesion and the healthy skin (*border*). Indeed, the edge of the lesion is very often diffuse in healthy skin, giving rise to a distinct textured region. This texture is particularly well visible in a grayscale image where healthy skin and lesion center textures tend to be smoother (See Figure 4.15). Therefore, it can be used to characterize and detect the border region of the lesion.

Given that each dermoscopic image is accompanied by a ground truth segmentation map that is pixel-based and binary (*lesion/healthy*), each block

resulting from image subdivision is assigned two distinct binary class labels ( $C_{label}$  and  $T_{label}$ ), one for each feature vector being extracted (texture and color texture features respectively). Therefore, two different class labeling methods are devised using two disparate thresholding procedures based on the following rules:

**a) Labeling method 1:** if the ratio of *lesion* pixels in the corresponding ground truth segmentation map is greater than a pre-defined threshold  $T\%$ , the block label is set to 1 (skin *lesion* class) otherwise it is set to the value 0 (*healthy* skin class). The choice of the threshold value  $T\%$  is made empirically. The resulting class label is denoted as  $C_{label}$ .

**b) Labeling method 2:** if the ratio of *lesion* pixels in the corresponding ground truth segmentation map is between two pre-defined thresholds ( $T_1\% - T_2\%$ ), the class label is set to 1 (*border* of the *lesion*) otherwise it is set to the value 0 (*non-border* region). It should be noted that a non-border region corresponds to the *lesion* center or healthy skin. The choice of the threshold values ( $T_1\%$  and  $T_2\%$ ) is made empirically. The resulting class label is denoted as  $T_{label}$ .



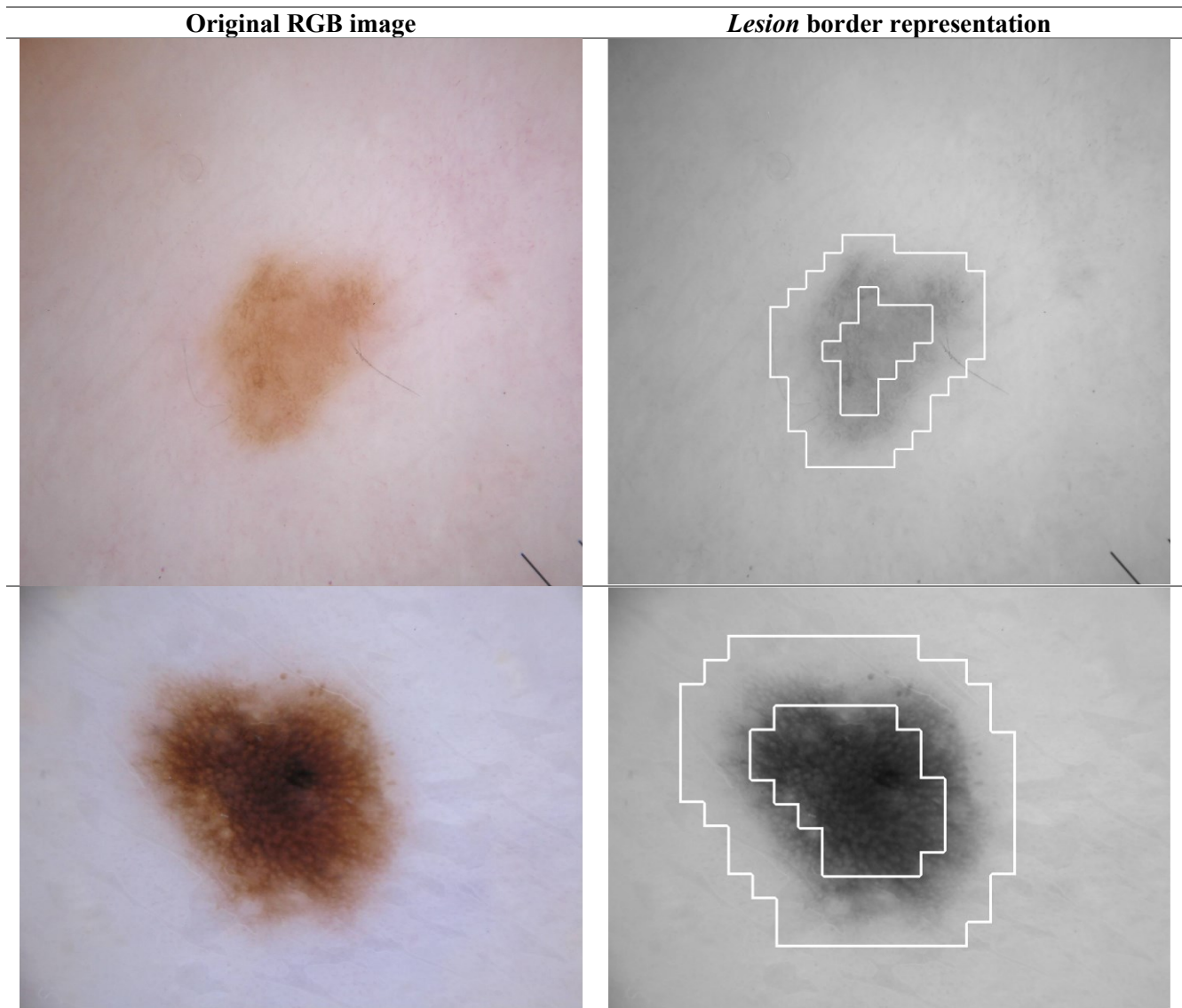


Figure 4.15 Examples of dermoscopic color images from ISIC dataset [40] (on the left side). The corresponding grayscale images with *lesion border representation* using the *Labeling method 2* (on the right side).

#### 4.2. The offline stage

This stage consists mainly of a supervised learning process. The training data is composed of two sets of labeled feature vectors (*Color index* and *Texture index*). Indeed, starting from a set of image blocks (subdivided images with pixel overlapping) and their corresponding ground truth, feature extraction operates at the block level to derive one feature vector per image block for gray texture and another one for color texture using the methods previously

defined in Sections 4.2. Using the class *labeling methods 1* and *2*, two binary class labels are assigned to each block: the first label denoted as  $C_{label}$  indicates if the block is a *lesion* or a *healthy* skin region (1 or 0 respectively). It is associated to the color texture feature vector. The second class label denoted as  $T_{label}$  indicates if the block is from the *border* of a *lesion* region or from a *non-border* region (1 or 0 respectively). It is associated to the gray texture feature vector. A learning process using either KNN or SVM algorithm is conducted on each of the two sets of labeled feature vectors (*Color index* and *Texture index*) under the following varying parameters: the type of distance metric, the factor  $K$  in KNN and the *kernel* type in SVM. One learning process is applied to the color texture features (with  $C_{label}$ ) and another one is applied to the texture features (with  $T_{label}$ ). These two processes result into two binary classification models  $CL_C$  and  $CL_T$  (respectively) which are saved for detection purposes. The steps of this offline learning phase can be highlighted as follows:

<p><b>Input:</b> Training datasets containing color texture images (RGB, YCbCr, ...) and their corresponding ground truth (<math>C_{dataset}</math> and <math>G_{dataset}</math> respectively).</p> <p><b>Output:</b> Two saved classification models (<math>CL_C</math> and <math>CL_T</math>).</p>
<ol style="list-style-type: none"> <li>1. Subdivide each image in <math>C_{dataset}</math> into overlapping blocks <math>B_c(i)</math>, with <math>i=1...N_B</math>.</li> <li>2. Convert each block <math>B_c(i)</math> into a grayscale block <math>B(i)</math>.</li> <li>3. Subdivide each binary ground truth map in <math>G_{dataset}</math> into overlapping blocks <math>G(i)</math>, with <math>i=1...N_B</math>.</li> </ol> <p style="text-align: center;"><i>// RCT-Plus texture feature extraction</i></p> <ol style="list-style-type: none"> <li>4. Apply RCT-Plus decomposition on each block <math>B(i)</math>, assuming <math>L</math> scale levels and <math>D_l</math> directions at each scale level <math>l</math>.</li> <li>5. Extract the texture feature vector <math>F(i)</math> from image block <math>B(i)</math> by performing a GGD modeling (4.3) on contourlet sub-bands using either <i>ML</i> or <i>MM</i> estimation: <math display="block">F(i) = \{\alpha(l, d), \beta(l, d); \quad l = 1 \dots L; \quad d = 1 \dots D_l\}</math> </li> </ol> <p style="text-align: center;"><i>// GMM color feature extraction</i></p> <ol style="list-style-type: none"> <li>6. Perform on each color block <math>B_c(i)</math>, a <math>k</math>-component GM modeling (4.17) using all available color channels and extract the color feature vector: <math display="block">F_c(i) = \{\omega_j, \mu_j, \Sigma_j; \quad j = 1 \dots k\}</math> </li> </ol> <p style="text-align: center;"><i>// Class labeling of blocks</i></p> <ol style="list-style-type: none"> <li>7. Apply the <i>Labeling method 1</i> to the ground truth <math>G(i)</math> of each block <math>B_c(i)</math> and obtain its binary class label:</li> </ol>

$$C_{label}(i)=\{0("healthy")/1("lesion")\}.$$

Apply the *Labeling method 2* to ground truth  $G(i)$  of each block  $B(i)$  and obtain its binary class label:

$$T_{label}(i)=\{1("border")/0("non - border")\}.$$

*//Index construction*

8. Create the *Color* and *Texture indexes* to include all color and texture feature vectors  $F_c(i)$  and  $F(i)$  and their corresponding class labels  $C_{label}(i)$  and  $T_{label}(i)$  respectively.

*// Classifier training*

9. Use KNN (or SVM) to train a binary classifier once on the *color index* and once on the *Texture index* (from step 8). Obtain the best classification models  $CL_C$  and  $CL_T$  using a cross-validation procedure.
10. Save the two classification models  $CL_T$  and  $CL_C$  (for use in the detection phase).

Considering that feature vector components are statistic GGD or GMM parameters, a new alternative to the Euclidean distance metric is introduced in KNN algorithm. It is based on a symmetric version of the Kullback-Leibler divergence as expressed in Equations (4.5) and (4.18) respectively. Also, we used these symmetric KLDs as kernel functions in SVM for the GGD and GMM statistical features.

### 4.3. The online stage

This stage is meant to detect skin abnormality in each block of the input test image. The trained classifier models  $CL_C$  and  $CL_T$  are used to predict and assign a binary class label 0 (*healthy* for  $CL_C$  and *non-border* for  $CL_T$ ) or 1 (*lesion* for  $CL_C$  and *border* for  $CL_T$ ) to each of the two extracted feature vectors from the considered test image block. Note that each label prediction is accomplished with a probability of confidence denoted as a prediction score ( $T_{score}$  and  $C_{score}$  respectively). Thereupon, obtaining two sets of labels detected by the two different trained classifiers  $CL_C$  and  $CL_T$ , a block-based label conciliation is implemented as a post processing step, in order to achieve the best one label for each block indicating whether it is detected as *lesion* or *healthy* block. The block-based conciliation rules apply as the following:

**Input:** Predicted labels  $C_{label}(i)$  and  $T_{label}(i)$  for test image blocks  $B_c(i)$ .

**Output:** A unique binary  $label(i)$  per test image block  $B_c(i)$ .

1. For each block  $B_c(i)$  of the test image, load the 2 predicted labels  $C_{label}(i)$  and  $T_{label}(i)$ .  
  
*// If the texture label is the same as the color one, the final label is the texture label.*
2. **if**  $T_{label}(i)=0$  and  $C_{label}(i)=0$  **then**  $label(i) = 0$  *// healthy*  
    **else if**  $T_{label}(i)=1$  and  $C_{label}(i)=1$  **then**  $label(i) = 1$  *// lesion*  
  
*// If the texture label is different from the color one,*  
*// the final label is the one having the highest prediction score.*  
    **else if**  $T_{label}(i)=0$  and  $C_{label}(i)=1$  **then**  
        **if**  $T_{score} > C_{score}$  **then**  $label(i) = 0$  *// healthy*  
        **else**  $label(i) = 1$  *// lesion*  
    **endif**  
    **else if**  $T_{label}(i)=1$  and  $C_{label}(i)=0$  **then**  
        **if**  $T_{score} > C_{score}$  **then**  $label(i) = 1$  *// lesion*  
        **else**  $label(i) = 0$  *// healthy*  
    **endif**  
    **endif**
3. Save all labels:  $label(i) = \{0("healthy")/1("lesion")\}$ .

Mosaicking consists of assembling all detection results into a detection map that can be superimposed on the input test image in order to localize and mark the abnormal (*lesion*) skin image areas. The proposed online phase for abnormality detection consists of the following steps:

**Input:** The test color image  $I_c$  (RGB, YCbCr, ...).

**Output:** Color Image with detected abnormal region(s).

1. Subdivide  $I_c$  into  $N_B$  overlapping blocks  $B_c(i)$ , with  $i=1...N_B$
2. Convert each block  $B_c(i)$  into a grayscale block  $B(i)$ .  
  
*// RCT-Plus texture feature extraction and classification*
3. Apply RCT-Plus decomposition on each block  $B(i)$  assuming  $L$  scale levels and  $D_l$  directions at each scale level  $l$ .
4. Extract the texture feature vector  $F(i)$  from image block  $B(i)$  by performing a GGD modeling (4.3) on contourlet sub-bands using either *ML* or *MM* estimation:  
$$F(i) = \{\alpha(l, d), \beta(l, d); \quad l = 1 \dots L; \quad d = 1 \dots D_l\}$$
5. Classify each block  $B(i)$  using  $CL_T$  on  $F(i)$  and predict its label  $T_{label}(i)$ .

- // GMM color feature extraction and classification
6. Perform on each color block  $B_c(i)$  a  $k$ -component GMM modeling (4.17) using all available color channels and extract the color feature vector:
 
$$F_c(i) = \{\omega_j, \mu_j, \Sigma_j; \quad j = 1 \dots k\}$$
  7. Classify each block  $B_c(i)$  using  $CL_C$  on  $F_c(i)$  and predict its label  $C_{label}(i)$ .
  8. Save all labels  $C_{label}(i)$  and  $T_{label}(i)$ .
- // Detection of abnormalities
9. Apply the block-based *Conciliation procedure* to get one final label per block:
 
$$label(i) = \{0("healthy")/1("lesion")\}.$$
  10. Build a *Detection map* (mosaicking) using all labels  $label(i)$ .
  11. Use the *Detection map* to localize the abnormal region on color image  $I_c$ .
  12. Visualize the results.

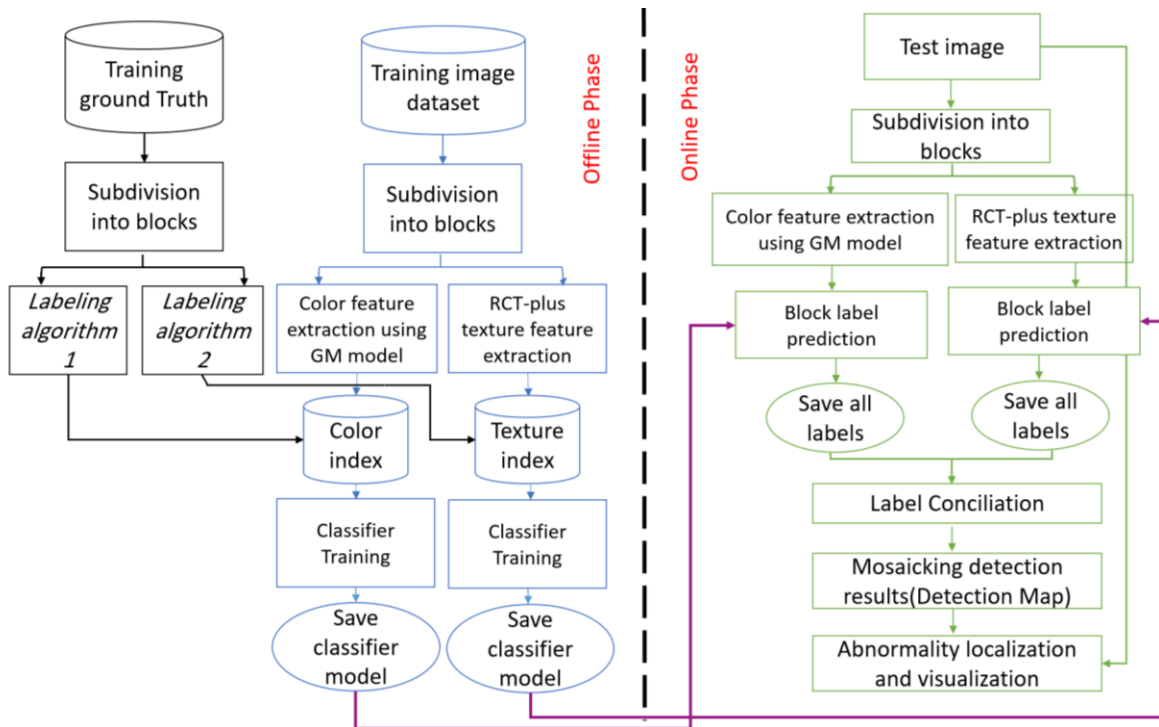


Figure 4.16 Abnormality detection scheme using RCT-Plus texture features concatenated with GMM color features and KNN/SVM binary classifiers.

#### 4.4. Application to segmentation in dermoscopic images

In this section, we propose a segmentation framework to compute accurate (pixel-based) boundaries of the *lesion* against healthy skin tissue in dermoscopic images. For this purpose, the interactive foreground extraction algorithm using Grabcut, is leveraged to segment the *lesion*. Grabcut is an improved version of Graph-Cut (iterated graph-cut) [76] as implemented in the image processing toolbox of Matlab [77]. Also, our proposed abnormality detection method (as previously explained) contributes as a marker to determine the foreground and background masks that are needed to initiate the segmentation process. Figure 4.17 summarizes the proposed segmentation framework.

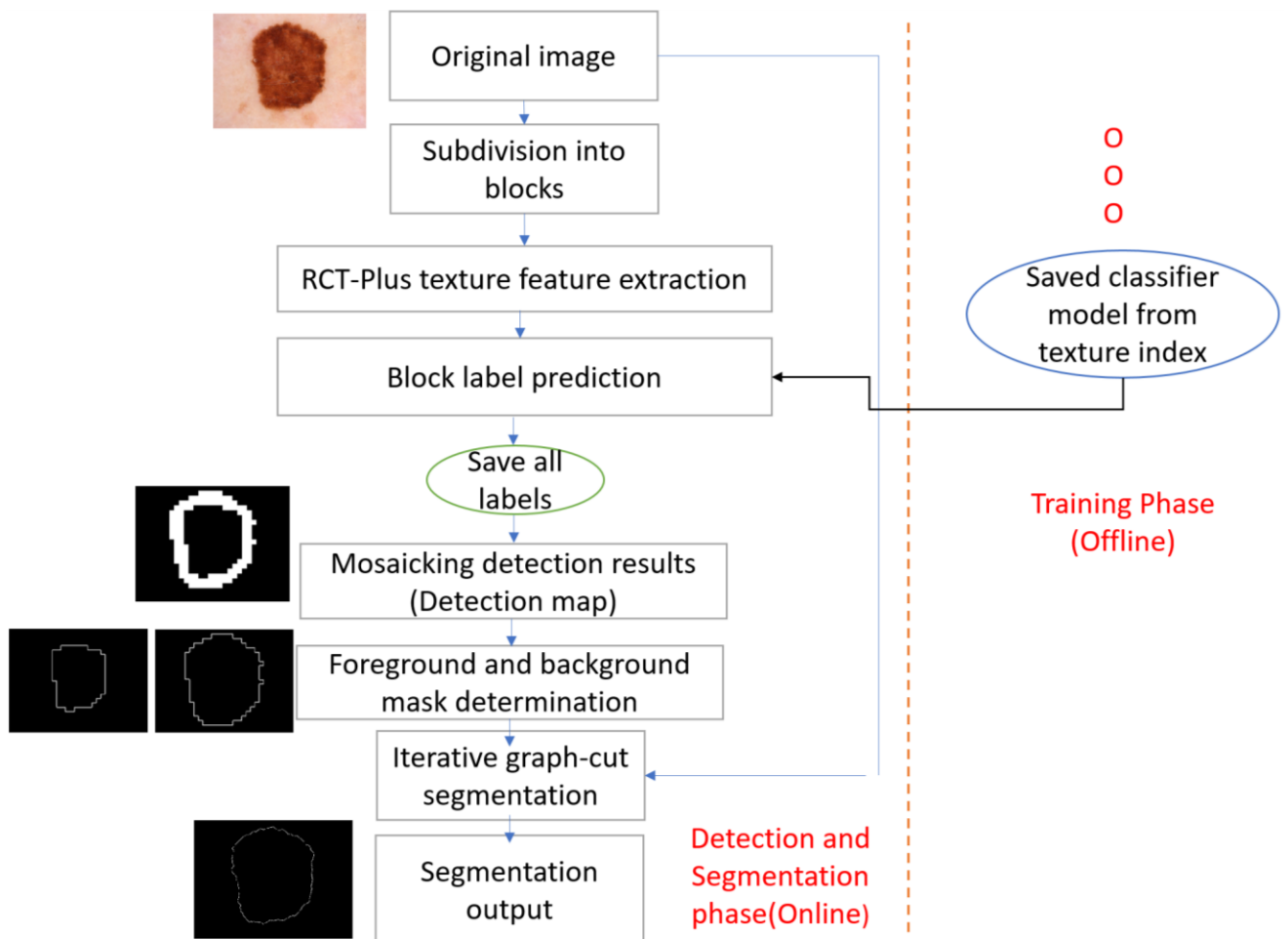


Figure 4.17 Abnormality segmentation framework using Graph-Cut.

Graph-cut segmentation is a technique used in computer vision to segment a region of interest by optimizing the energy function over the segmented region [78]. To do this, a given image is firstly transformed into its graph representation. In graph representation, each image pixel is considered as a node, while two complementary nodes are introduced as Foreground (source) and Background (sink) nodes. The graph edges are defined as connections between all pixel nodes to both the sink and the source nodes while the probability value for each pixel to be a foreground pixel or a background pixel determines the edge weights. Furthermore, all pixel nodes are connected to their neighboring pixels by edges whose weights are determined using an algorithm that would promote the connection of similar pixels. Subsequently, the underlying graph is divided into two graphs by cutting the edges whose removal makes the original graph disconnected while minimizing the sum of the edge weights which are going to be removed. Using different types of edge weighting approaches as well as different optimization techniques, to decide which edges to remove, create the wide range of graph-cut algorithm variants in the literature.

As previously mentioned, in this work we take advantage from the iterative Graph-cut (Grabcut) variant [76].

For this purpose, as illustrated in Figure 4.17, and in compliance with the instructions in section 4.3. The online stage) an input image is firstly subdivided into blocks and then each block is classified using the pre-trained classifier for texture-based features ( $CL_T$ ) followed by mosaicking the detection results (*border/non-border*) into a detection map that can be used as foreground and background masks.

Once the foreground and the background masks are marked, the following steps are iteratively performed by the Grabcut algorithm to find a boundary between the foreground and the background masks [79]:

- 1- Use Gaussian Mixture Model to estimate the color distribution of the foreground and the background.
- 2- Estimate a Markov Random Field for the foreground and background pixel labels.
- 3- Apply a Graph-cut optimization in order to obtain the accurate final segmentation.

## CHAPTER 5. Experiments and results

This chapter is mainly devoted to the evaluation and validation of the proposed methodology for texture feature extraction and its incorporation into content-based image retrieval and abnormality detection schemes. We describe the conducted experimental studies on different popular texture datasets and provide the main experimental results with discussions and drawn conclusions. The implementation of this work is done in the Matlab environment [77]. In section 1, we introduce six color and grayscale image datasets that are utilized in the experimental setups and give the evaluation criteria that are considered for each application domain. Sections 2 and 3 detail the achieved results corresponding to grayscale texture feature extraction and retrieval and show the benefits of using RCT-Plus in comparison to other multi-scale transforms. Thereupon, by extending to joint color texture feature extraction and analyzing the experimental results in conventional CBIR, the most efficient methods are selected and incorporated into ML-CBIR schemes to enhance the texture retrieval performance. In the last section, we evaluate our proposed abnormality detection scheme, applying grayscale and color texture feature extraction methods for block-based detection and segmentation in medical skin lesion images. Detailed results are provided and a comparison to the U-net deep learning approach for segmentation is also performed.

### 1. Data description and evaluation criteria

We used six popular datasets to experiment with our proposed texture and joint color texture feature extraction methods. The first four datasets are applied to content-based image retrieval while the last two datasets are applied to abnormality detection and segmentation applications.



### 1.1. The VisTex-40 dataset

The first well-known texture dataset we used in this work is named MIT Vision Texture (VisTex) dataset to conduct the CBIR experimental work and assess retrieval effectiveness [7]. To simplify the comparison to state-of-the-art, we used the same experimental dataset as in [10]. Thus, from the VisTex dataset, we selected the forty (40) grayscale texture images shown in Figure 5.1. These are texture images of daily life taken from different scenes. We used the original RGB images (shown in Figure 5.2). Also, we studied the converted version of the original RGB images to grayscale and YCbCr format. Each of the forty 512×512-pixel size images were divided into 16 non-overlapping sub-images of size 128×128, constituting a texture class. As a result, a dataset with 640 images organized into 40 texture classes was constructed and designated as VisTex-40.

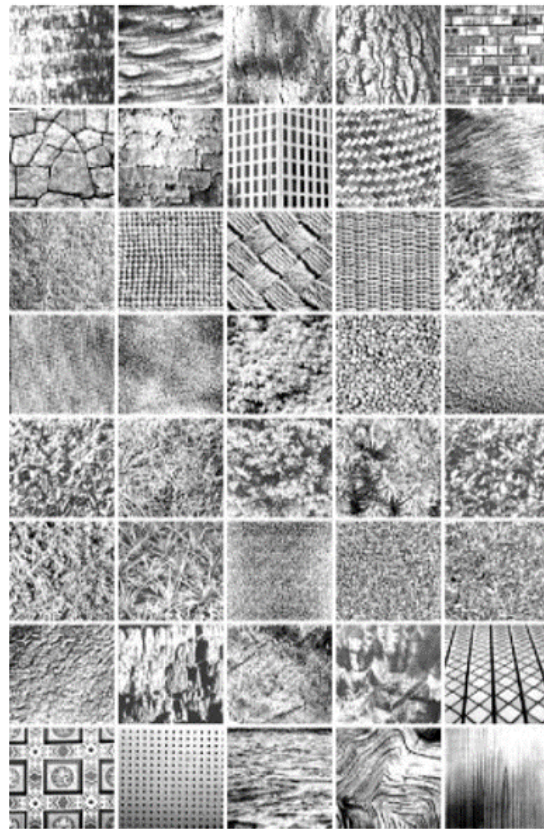


Figure 5.1 Texture images selected from the VisTex dataset [7]. From left to right and top to bottom: Bark0, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Buildings9, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric14, Fabric15, Fabric17, Fabric18, Flowers5, Food0, Food5, Food8, Grass1, Leaves8, Leaves10, Leaves11, Leaves12, Leaves16, Metal0, Metal2, Misc2, Sand0, Stone1, Stone4, Terrain10, Tile1, Tile4, Tile7, Water5, Wood1, and Wood2.

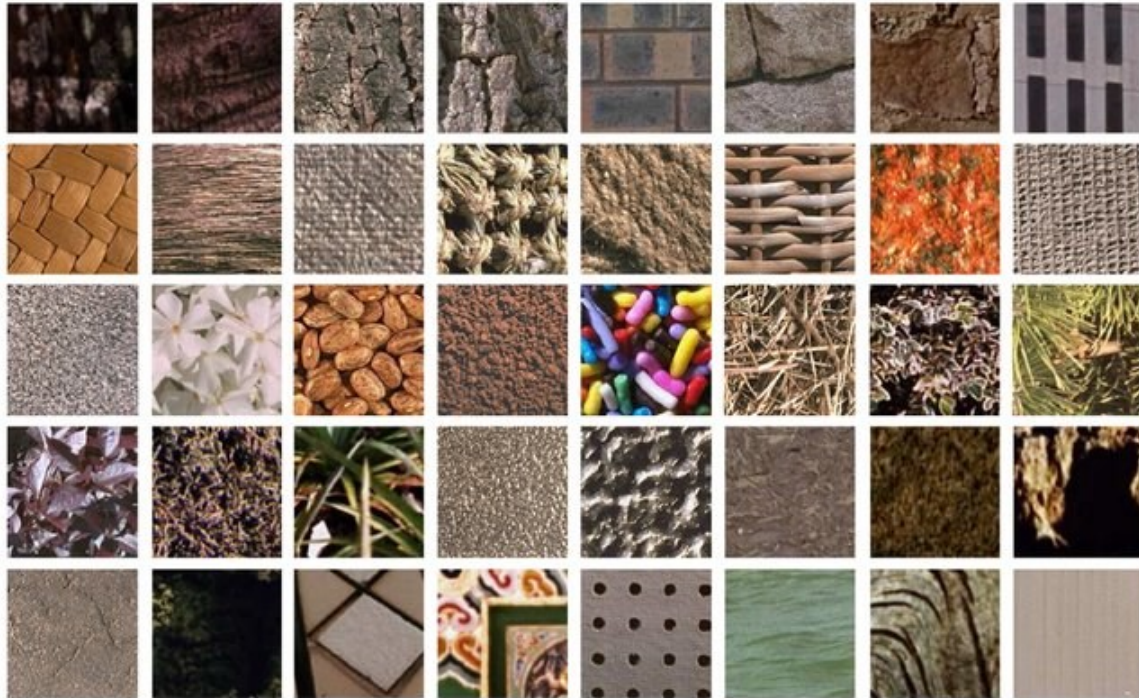


Figure 5.2 RGB VisTex dataset with 40 classes [7]. Each image, indicates a sample from 16 images for each class. From left to right and top to bottom: Bark0, Bark6, Bark8, Bark9, Brick1, Brick4, Brick5, Buildings9, Fabric0, Fabric4, Fabric7, Fabric9, Fabric11, Fabric14, Fabric15, Fabric17, Fabric18, Flowers5, Food0, Food5, Food8, Grass1, Leaves8, Leaves10, Leaves11, Leaves12, Leaves16, Metal0, Metal2, Misc2, Sand0, Stone1, Stone4, Terrain10, Tile1, Tile4, Tile7, Water5, Wood1, and Wood2.

### 1.2. The Kylberg-27 dataset

The original Kylberg texture dataset v. 1.0 created by Gustaf Kylberg contains a collection of 28 texture classes [37]. Each class was imaged from textured surfaces, including fabrics and surfaces of floors, walls, and ceilings in the local surroundings. Textured surfaces were also arranged using articles such as rice grains, sesame seeds, and lentils.

Each class includes 160 unique texture patches without rotations. An alternative Kylberg dataset comes with 12 rotations per original patch, thus creating  $160 \times 12 = 1920$  texture patches per class. The texture patch size is  $576 \times 576$  pixels in grayscale PNG format. All patches are normalized with a mean value of 127 and a standard deviation of 40.

The Kylberg-27 dataset is a subset of the Kylberg texture dataset with 27 distinct texture classes, each class contains 40 grayscale images of size  $512 \times 512$ . As a result, a dataset with 1080 texture images organized into 27 texture classes was constructed for our experimentations and designated as Kylberg-27 (See Figure 5.3).

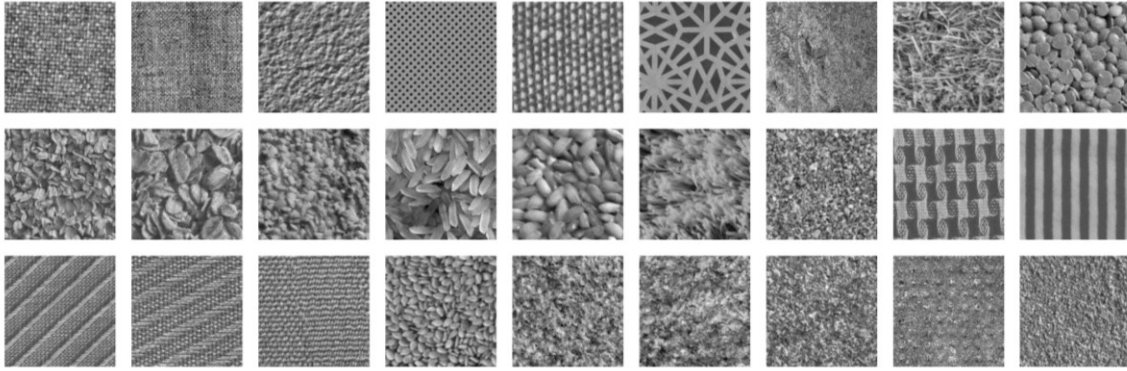


Figure 5.3 Kylberg-27 texture dataset [37]. From left to right and top to bottom: blanket1, canvas, ceiling1, ceiling2, cushion1, floor1, floor2, grass1, lentils1, linseeds1, oatmeal1, pearlsugar1, rice1, rice2, rug1, sand1, scarf1, scarf2, screen1, seat1, seat2, sesameseed1, stone1, stone2, stone3, and stoneslab1.

### *1.3. The Kylberg-28 dataset*

In a broader experience, we built the Kylberg-28 dataset, which contains the whole non-rotated Kylberg texture dataset. It covers 28 distinct texture classes. Each class contains 160 grayscale images cropped to  $512 \times 512$  size images (See Figure 5.4) and creates a dataset of 4480 images.

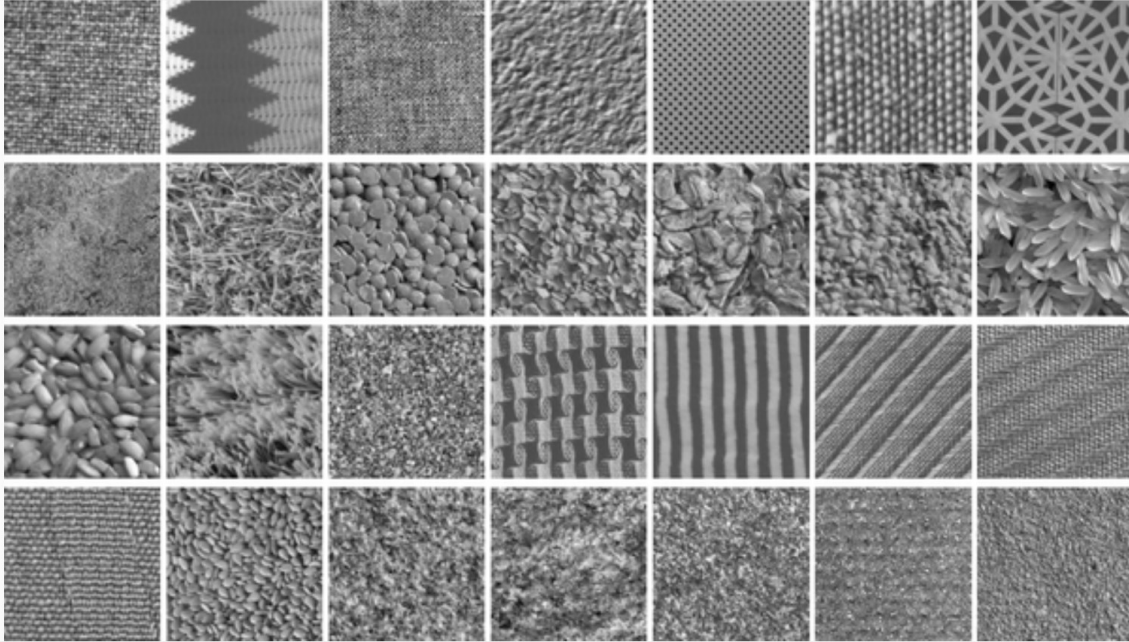


Figure 5.4 Kylberg-28 texture dataset [37]. from left to right and top to bottom: blanket1, blanket2, canvas, ceiling1, ceiling2, cushion1, floor1, floor2, grass1, lentils1, linseeds1, oatmeal1, pearlsugar1, rice1, rice2, rug1, sand1, scarf1, scarf2, screen1, seat1, seat2, sesameseed1, stone1, stone2, stone3, and stoneslab1.

#### 1.4. The Stex dataset

The Salzburg Texture Image Dataset (STex) [25] is an extensive collection of 476 color texture classes captured in Salzburg, Austria. Stex was originally acquired with the aim of use in texture retrieval experiments. With the vast number of classes, it is evident that STex is significantly larger than the VisTex or other popular texture image datasets. Stex contains 7616 distinct color texture images in RGB format, each of size  $128 \times 128$ . A texture class contains 16 color images (see

Figure 5.5).

In this work, two additional versions are derived from the original STex dataset: the converted YCbCr version and the grayscale version.

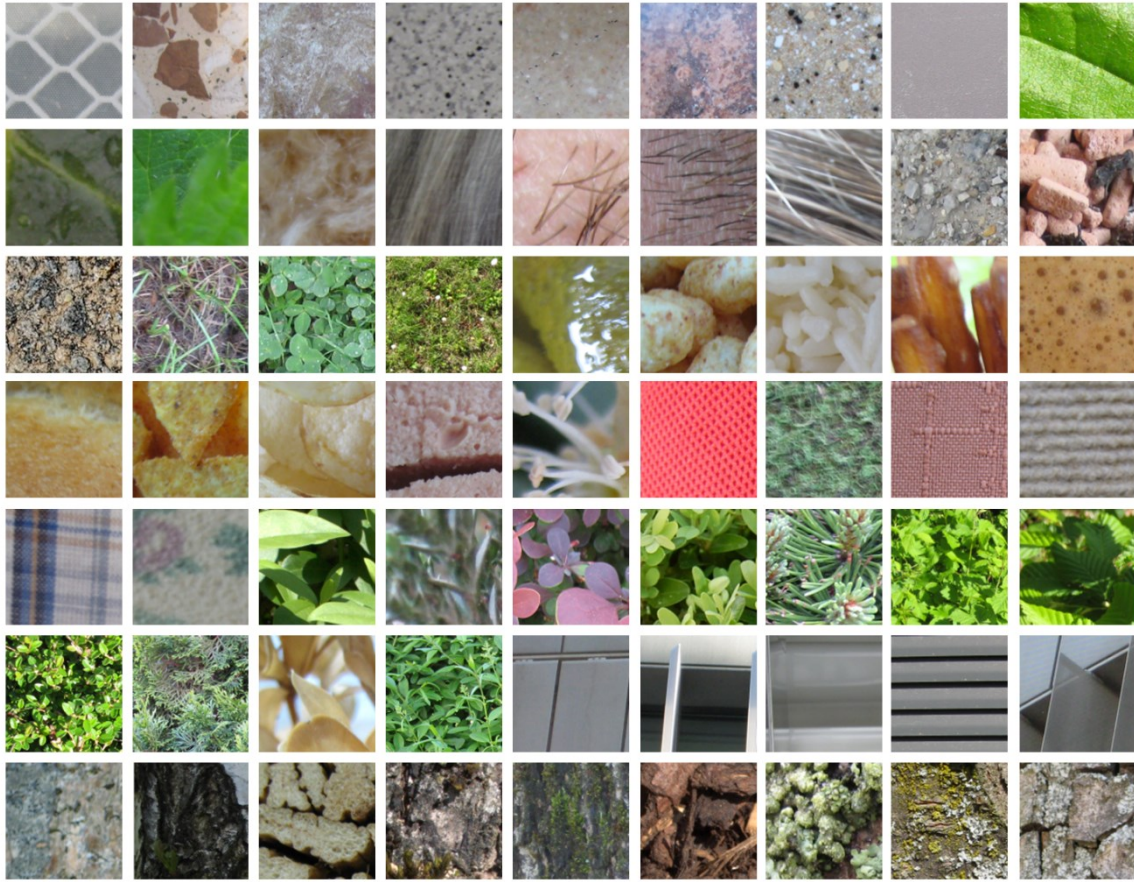


Figure 5.5 A sample of 63 classes of the Salzburg Texture Image Dataset (STex) [25].

### 1.5. The ISIC-32 dataset

The International Skin Imaging Collaboration (ISIC) is an international endeavor to advance melanoma diagnosis, supported by the International Society for Digital Imaging of the Skin (ISDIS) [40]. The ISIC Archive contains the largest available group of quality controlled dermoscopic images of skin lesions.

Currently, the ISIC Archive includes over 13,000 dermoscopic images of melanoma and nevus skin lesions, which have various image sizes and were collected from clinical centers internationally and developed from the variation of devices within each center. In this work, we created two subsets from the ISIC dataset, namely ISIC-32 and ISIC-42.

For ISIC-32, we selected thirty-two dermoscopic RGB images (6 melanoma/malignant and 26 nevus/benign) of the same size (1504×1129) and their corresponding ground truth segmentation maps (as shown in

Figure 5.6). Each of the 32 images has been divided into overlapping 128×128 blocks. The amount of overlap is 64 columns and 64 rows. We have the same image sizes in ISIC-32, so each image is divided into 352 blocks. A similar subdivision is performed on the corresponding ground truth segmentation map. As a result, a dataset having 11264 RGB sub-images and 11264 ground truth segmentation maps was constructed and designated as ISIC-32. Also, we studied in this research the RGB images, the converted version of the original RGB images in YCbCr format and the grayscale version.

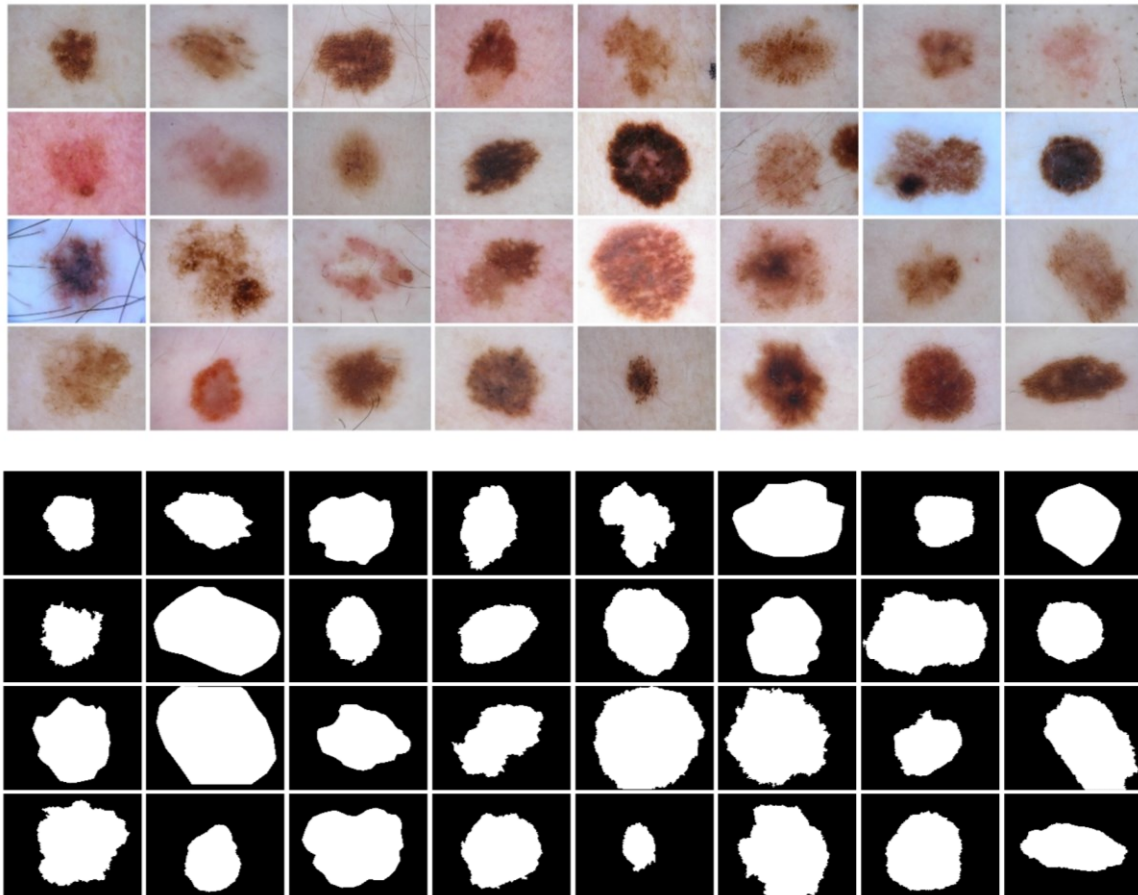


Figure 5.6 Selected dermoscopic images (ISIC-32) from the ISIC dataset [40] and their corresponding ground truth segmentation maps.

### *1.6. The ISIC-42 dataset*

In ISIC-42, we selected forty-two RGB images (7 melanoma/malignant and 35 nevus/benign) of various sizes and their corresponding ground truth segmentation maps (as shown in Figure 5.7). Each of the 42 images has been divided into overlapping  $128 \times 128$  blocks. The amount of overlap is 64 columns and 64 rows. A similar subdivision is performed on the corresponding ground truth segmentation map. As a result, a dataset having 27648 RGB sub-images and 27648 ground truth segmentation maps was constructed and designated as ISIC-42. Furthermore, we studied in this research the RGB images, the converted version of the original RGB images in YCbCr format and the grayscale version.

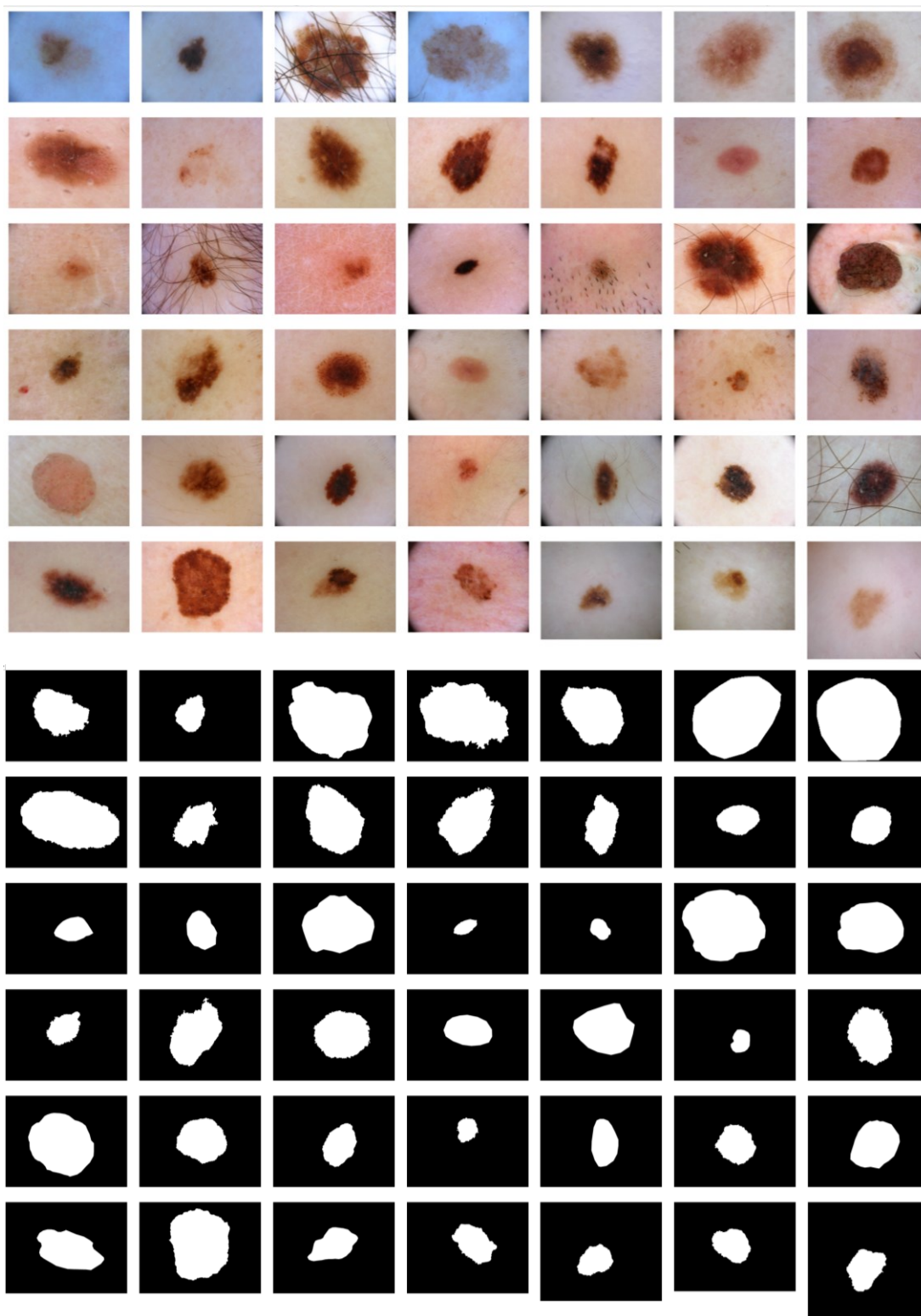


Figure 5.7 Selected 42 dermoscopic images (ISIC-42) from the ISIC dataset [40] and their corresponding ground truth segmentation maps.



### 1.7. Evaluation criteria

To evaluate the efficiency of a retrieval experiment, every image of the considered test dataset is presented as a query image to the CBIR system. As a matter of fact, the ground truth for each query is known beforehand. Retrieval performance for each query is measured in terms of the retrieval rate ( $RR\%$ ), which is calculated as the percentage of relevant images found among  $N$  retrieved images ( $TopN$  matches). Here, an image is considered relevant if it is part of the ground truth of the query. The  $RR\%$  is in fact a *Recall* measure (2.2) over  $TopN$  retrieved images. All retrieval results presented in this work were obtained by averaging the retrieval rates corresponding to all queries executed ( $AR\%$ ).

To evaluate the efficiency of a supervised machine learning algorithm, one popular metric is *Accuracy* which is based on the prediction results obtained from applying the classifier to test data. The prediction results are formulated as four measures: true positives  $tp$ , false positives  $fp$ , true negatives  $tn$  and false negatives  $fn$ . The *Accuracy* measure is then given by the following ratio:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \quad (5.1)$$

Cross-validation is a statistical approach widely used to determine how well a developed machine learning model can generalize the solution to new data. During the cross-validation, the whole training data set is randomly split into a number  $n$  of equal folds. The classifier model is then trained on  $n-1$  folds while the remaining fold is kept for testing purposes. This operation is iteratively repeated  $n$  times and each iteration deals with a newly selected testing fold. Consequently, we have  $n$  different values specifying the *Accuracy*. The average of the  $n$  obtained measures is then considered as the overall performance of the learned classifier model. This approach gives a more realistic evaluation of the model at expense of higher computational cost.

Also, in order to evaluate the efficiency of the abnormality detection, we firstly compare the predicted labels to the ground truth ones and then calculate the *Precision*, *Recall*, and detection *Accuracy* values under the following formulations:

$$Precision = \frac{tp}{tp + fp} \quad (5.2)$$

$$Recall = \frac{tp}{tp + fn} \quad (5.3)$$

where:

- $tn$  is the number of healthy blocks that are detected as healthy;
- $tp$  is the number of *lesion* blocks that are detected as *lesion*;
- $fp$  is the number of healthy blocks that are detected as *lesion*;
- $fn$  is the number of *lesion* blocks that are detected as healthy.

The *Accuracy* is calculated as in equation (5.1).

## 2. Texture feature extraction methods

In this section, we are verifying and validating the implementations of RCT-Plus, on grayscale textures to test and find the persuaded number of orientations and scales on this transform. Moreover, we compare the results to other multi-scale transformations in CBIR using two metric distances. Then by moving to color texture feature extraction and the cooperation between color and grayscale texture, as we introduced in chapter 4, section 3.1, we can see the improvement in conventional content-based image retrieval.

### 2.1. The multi-scale decomposition in feature extraction methods

For the considered datasets of grayscale texture images, we computed five distinct texture feature extraction methods operating on a multi-scale image representation such as RCT-Plus. Three of them, are based on GGD modeling of image sub-bands and perform similarity measurement using either Kullback-Leibler distance or Euclidean distance. The two remaining approaches involve energy calculations on image sub-bands and Euclidean distance as the similarity metric. Each feature extraction method can operate on the RCT-Plus contourlet transform, with pseudo-Gaussian and *pkva* filters, yielding a flexible number  $D$  of directional sub-bands at each scale level.

Therefore, various CBIR schemes are proposed. Each of them combines RCT-Plus decomposition with sub-band modeling and a suitable similarity measurement from the following:

- 1) GGD1: GGD modeling of each detail sub-band (4.3) and the approximation sub-band using *Moment Matching (MM)* estimation and similarity metric based on the symmetric KLD (4.6). Two parameters (feature vector elements) are estimated for each sub-band.
- 2) GGD2: GGD modeling of each detail sub-band (4.3) and the approximation sub-band using *Maximum Likelihood (ML)* estimation and similarity metric based on the symmetric KLD (4.6). Two parameters (feature vector elements) are estimated for each sub-band.
- 3) GGD2-ED: GGD modeling of each detail sub-band (4.3) and the approximation sub-band using *Maximum Likelihood (ML)* estimation and similarity measurement based on the Euclidean distance. Two parameters (feature vector elements) are estimated for each sub-band.
- 4) E1: Energy-based feature extraction in each image sub-band (4.11) and similarity measurement based on the Euclidean distance. Two energy values (feature vector elements) are calculated for each sub-band.
- 5) E2: Energy-based feature extraction in (4.12) and similarity measurement based on the Euclidean distance. Two energy values (feature vector elements) are calculated for each sub-band.

In a first experiment setup, texture feature extraction for image retrieval is conducted on VisTex-40 dataset using RCT-Plus decomposition. The number of directional sub-bands per scale level is fixed to  $D = 8$  while the number of scale levels is 1, 2, or 3. Feature extraction is applied on all contourlet sub-bands including the approximation sub-band. For each of the CBIR schemes (GGD1, GGD2, GGD2-ED, E1 and E2), retrieval performance in terms of average retrieval rates ( $AR\%$ ) in the  $TopN$  matches are reported in Table 5.1. Some interesting observations may be drawn from these experiments:

- 1) In most test cases, GGD-based methods achieve better retrieval results than energy-based methods and GGD1 (using the *MM* estimation) exhibits the best retrieval rates among the five tested methods.
- 2) There is a substantial improvement in retrieval rates when the number of contourlet decomposition levels is increased from one

to two scales. However, extending contourlet decomposition to three scale levels provides very little improvement in retrieval performance.

<i>TopN</i>	Number of scale levels L	GGD1	GGD2	GGD2-ED	E1	E2
16	1	75.19	72.49	65.75	62.67	63.40
	2	<b>77.43</b>	75.13	71.73	69.23	70.15
	3	77.15	75.24	71.78	70.50	71.72
20	1	80.30	77.52	69.97	67.22	67.96
	2	81.94	78.74	75.62	73.17	74.29
	3	<b>82.03</b>	79.12	75.77	74.67	75.57
40	1	88.43	84.90	81.26	78.06	79.30
	2	88.97	86.18	84.62	83.15	84.43
	3	<b>89.00</b>	86.28	84.54	83.85	84.44
60	1	91.74	88.37	86.76	83.45	84.76
	2	<b>92.72</b>	90.14	89.23	88.48	89.24
	3	92.29	89.62	88.35	88.22	88.48
80	1	93.30	90.28	90.23	86.90	88.30
	2	<b>94.66</b>	92.14	92.09	91.29	91.96
	3	94.30	91.88	91.10	91.25	91.33
100	1	94.53	91.74	92.69	89.43	90.60
	2	<b>95.78</b>	93.39	93.88	93.22	93.59
	3	95.60	93.42	92.82	93.00	92.92

Table 5.1 Average retrieval rates (*AR%*) according to the number of top matches considered, *TopN*. The compared CBIR schemes (E1, E2, GGD1, GGD2 and GGD2-ED) operate on RCT-Plus decomposition with L ranging from 1 to 3 and 8 directional sub-bands per scale level.

Figure 5.8 is another comparison example that clearly shows the superiority of the method GGD1 on RCT-Plus. As it can be seen, by varying the number of top matches considered (*TopN*) from 10 to 100, the GGD1 method operating on RCT-Plus decomposition outperforms all other CBIR methods (namely GGD2, GGD2-ED, E1 and E2) in terms of average retrieval rates (*AR%*). Moreover, almost 96% of relevant images are retrieved when the number of top matches considered is 100 images.

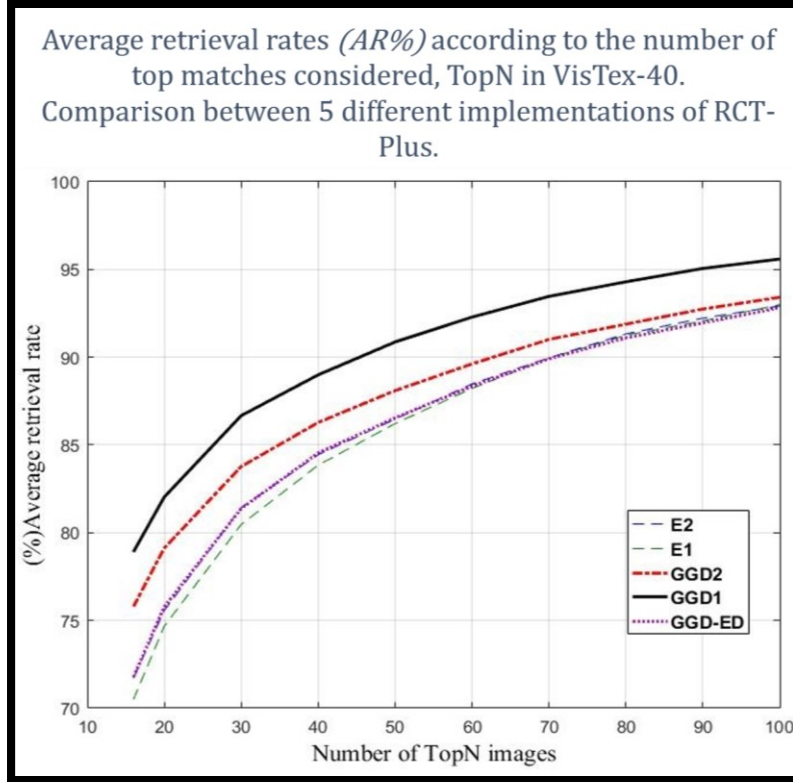


Figure 5.8 Average retrieval rates ( $AR\%$ ) according to the number of top matches considered,  $TopN$ . The compared CBIR methods (GGD1, GGD2, GGD2-ED, E1 and E2) are applied on an RCT-Plus decomposition with 3 scale levels and 8 directional sub-bands per scale level ( $L=3, D=[8, 8, 8]$ ).

A second experiment setup was carried out to study the effect of the number of RCT-Plus directional orientations per scale level ( $D_l$ ) on the retrieval performance. Let's recall that the number of directional sub-bands is variable and flexible in RCT-Plus. Indeed, we can have  $D_l=2, 4, 8, 16, \dots$  directional sub-bands and this number can vary across the scale levels 1 of the same decomposition.

We tested and analyzed many retrieval schemes using RCT-Plus with various combinations of scale levels and directional orientations ( $L$  and  $D_l$  values). A sample of results is shown in Table 5.2. We observe that, for GGD1 method, performance retrieval ( $AR\%$ ) is about 3.5 points higher when the number of directional sub-bands in RCT-Plus increases from 2 to 8. This improvement remains consistent over the number of top matches considered (see Figure 5.9). Moreover, the redundancy factor in this case remains unchanged since it only depends on the number of scale levels of the RCT-Plus decomposition. From this analysis, we came out with the observation that directional

information is important in enhancing texture discrimination and retrieval performance.

Methods	2 scale levels ( $L=2$ )				3 scale levels ( $L=3$ )			
	Directional sub-bands/scale level				Directional sub-bands/scale level			
	$[D_1, D_2]$				$[D_1, D_2, D_3]$			
	[2, 2]	[4, 4]	[4, 8]	[8, 8]	[2, 2, 2]	[4, 4, 4]	[4, 8, 8]	[8, 8, 8]
<b>GGD1</b>	73.54	75.10	75.45	<b>77.43</b>	73.54	74.93	74.99	<b>77.15</b>
<b>GGD2</b>	70.77	68.31	73.47	75.13	71.51	73.20	73.18	75.24
<b>GGD2-ED</b>	63.34	67.09	69.69	70.15	64.96	67.52	69.55	71.78
<b>E1</b>	61.68	66.17	66.99	69.23	65.18	68.44	68.73	70.50
<b>E2</b>	61.56	67.27	67.99	70.15	65.54	69.49	69.76	71.72

Table 5.2 Average retrieval rates ( $AR\%$ ) in the Top16 images. Retrieval methods using RCT-Plus with various combinations of scale levels and directions are compared. The number of directional sub-bands at each scale level  $l$  is indicated by  $D_l$ .

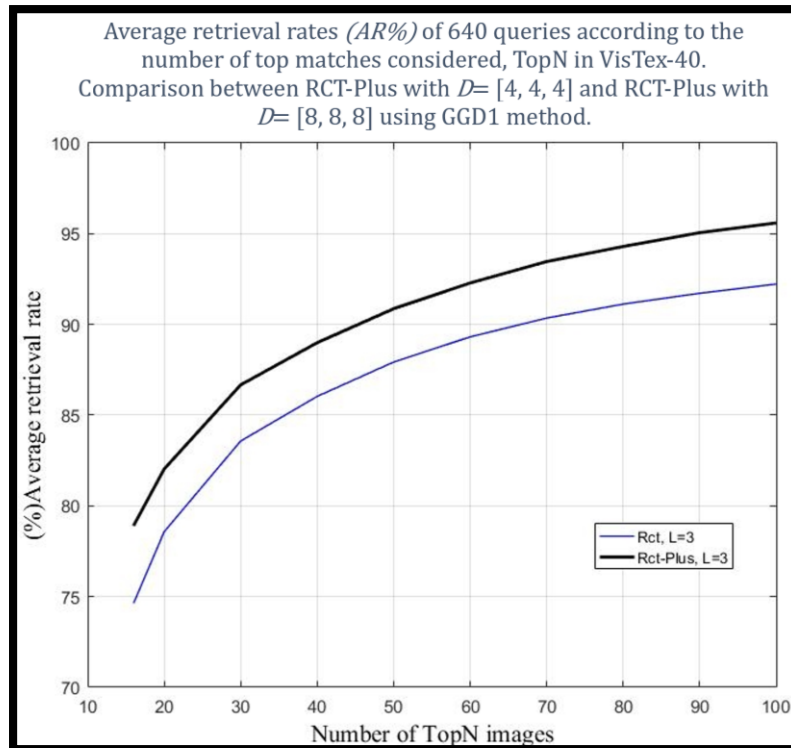


Figure 5.9 Average retrieval rates ( $AR\%$ ) of 640 queries according to the number of top matches considered,  $TopN$ . Two retrieval methods are compared: GGD1 using a 3-level RCT-Plus with  $D=[4, 4, 4]$  and GGD1 using a 3-level RCT-Plus with  $D=[8, 8, 8]$ .

## 2.2. RCT-Plus in comparison to other transforms

A third experiment setup is carried out to compare our contourlet-based CBIR schemes with classical and state-of-the-art methods including discrete wavelet-based CBIR schemes [10] and Gabor-based schemes [6].

Each of the introduced feature extraction method can operate on any of the following multi-scale decompositions including contourlet transform variants:

DWT: Discrete Wavelet Transform, with *db4* filters (Daubechies filters), yielding 3 sub-bands at each scale level.

1. Gabor: Gabor transform, with Gaussian filters, yielding 4 directional sub-bands at each scale level.
2. SCT: Standard Contourlet Transform, with *pkva* filters, yielding a flexible number *D* of directional sub-bands at each scale level.
3. RCT: Redundant Contourlet Transform, with pseudo-Gaussian and *pkva* filters, yielding 4 directional sub-bands at each scale level.
4. NSCT: NonSubsampled Contourlet Transform, with *pkva* filters, yielding a flexible number *D* of directional sub-bands at each scale level.
5. RCT-Plus: Redundant Contourlet Transform, with pseudo-Gaussian and *pkva* filters, yielding a flexible number *D* of directional sub-bands at each scale level.

Therefore, a new CBIR scheme is built for each combination of one of the above-listed multi-scale image decompositions with GGD1, GGD2, GGD2-ED, E1, or E2 feature extraction methods. Let's recall that RCT is a special case of RCT-Plus where the number of directional sub-bands is fixed to 4 at each scale level.

Table 5.3 depicts the best achieved results using GGD1 and energy E2 methods. All transforms have either 2 or 3 scale levels while the number of directional sub-bands at each scale level is 3 in DWT, 4 in Gabor, SCT and RCT, 8 in NSCT and RCT-Plus. It is very clear that 1) RCT delivers much better results than SCT and DWT even if the number of its directional sub-bands is limited to four, thanks to Gaussian filtering and redundancy, 2) there is very little improvement in retrieval performance when going from 2 to 3 levels of decomposition, 3) although the NSCT is a non-subsampled

decomposition and its redundancy factor is much higher than that of the RCT-Plus, feature extraction methods operating on RCT-Plus consistently give superior retrieval results than NSCT-based methods, and 4) feature extraction using RCT-Plus (with  $D_l$  fixed at 8) and GGD1 achieves the best retrieval rates. As additional results, Table 0.1 in Annex IV reports the average retrieval rates ( $AR\%$ ) according to the number of top matches considered (from 16 to 100). We compared CBIR methods (GGD1, GGD2, GGD2-ED, E1 and E2) operating on NSCT decompositions with  $D=8$  and  $L$  ranging from 1 to 3.

In Figure 5.10, more comparison examples clearly show the superiority of the RCT-Plus. As can be seen in Figure 5.10, by varying the number of top matches considered ( $TopN$ ) from 16 to 100, the GGD1 with RCT-Plus method still outperforms the GGD1 with NSCT, RCT and SCT in terms of average retrieval rates ( $AR\%$ ). Moreover, almost 96% of relevant images are retrieved when the  $TopN$  considered is 100 images.

Transform type	2 scale levels ( $L=2$ )		3 scale levels ( $L=3$ )	
	GGD1	E2	GGD1	E2
DWT	70.17	60.30	72.63	60.41
GABOR	63.79	61.76	66.89	64.41
SCT	65.28	60.56	68.29	62.89
RCT	75.10	67.27	74.93	69.49
NSCT	76.91	68.37	75.83	68.84
<b>RCT-Plus</b>	<b>77.43</b>	70.15	<b>77.15</b>	71.72

Table 5.3 Average retrieval rates ( $AR\%$ ) in the Top16 images. CBIR schemes based on contourlet variants (SCT, RCT, NSCT and RCT-Plus) are compared to DWT and Gabor CBIR schemes.



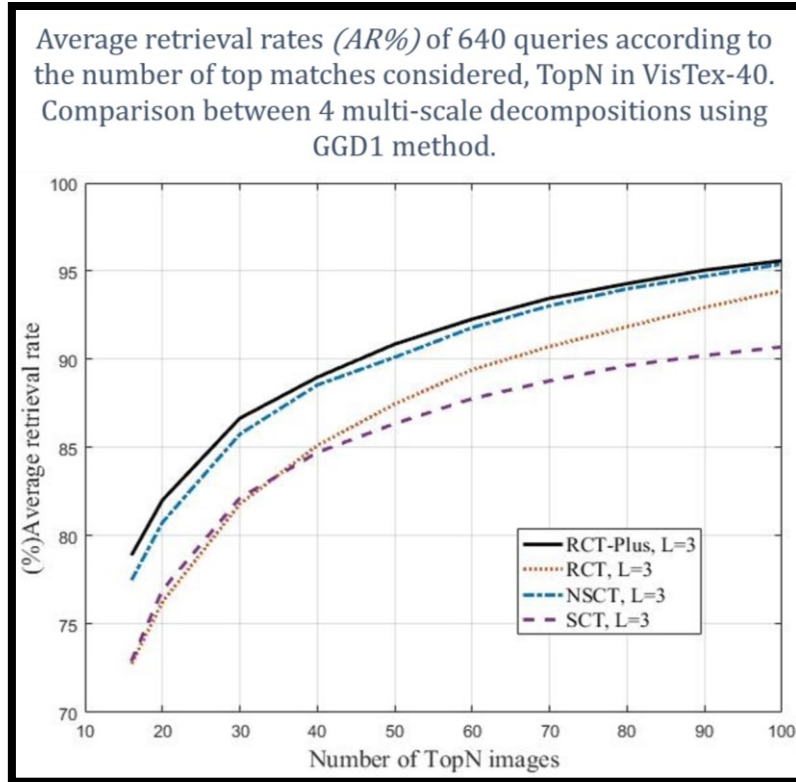


Figure 5.10 Average retrieval rates ( $AR\%$ ) of 640 queries according to the number of top matches considered,  $TopN$ . The GGD1 method is applied on 3-level contourlet decomposition variants (SCT, NSCT, RCT and RCT-Plus).

Examples of retrieved images from VisTex-40 dataset are shown in Figure 5.11. In each retrieval case, the query image is Food8.02 and the feature extraction method is GGD1 operating on one of the contourlet transform variants (SCT, NSCT, RCT or RCT-Plus). For each result, the Top16 retrieved images are ranked in the order of similarity with the query image from left to right, top to bottom. One can notice that RCT-Plus method retrieved perfectly all relevant images to the query Food8.02 while the other methods resulted in some non-relevant images in the Top16 matches.

Thanks to the enhanced properties of Gaussian filtering, redundancy and augmented directional selectivity, the joint exploitation of RCT-Plus with simple statistical modeling such as generalized Gaussian distribution or energy has resulted into powerful methods for discriminating textures and performing content-based image retrieval.



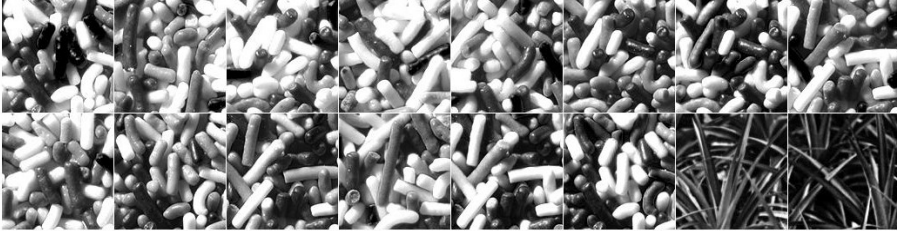


 <p>Query image: Food8.02</p>	
<p>SCT</p>	 <p>Two images from Flower5 and one from Leaves10 are non-relevant to the query.</p>
<p>NSCT</p>	 <p>Two images from Leaves16 are non-relevant to the query image.</p>
<p>RCT</p>	 <p>One image from Tile4 is non-relevant to the query image.</p>
<p>RCT-Plus</p>	 <p>All relevant images to the query are perfectly retrieved.</p>

Figure 5.11 Examples of Top16 retrieval results from the VisTex-40 dataset (640 images). The query image is Food8.02.

### 2.3. Color texture feature extraction methods in the CBIR framework

For datasets of color images (such as RGB and YCbCr color spaces), we involved color texture in the CBIR framework by five different methods of feature extraction (see Table 5.4). Each of them combines RCT-Plus color image decomposition with sub-band modeling and a suitable similarity measurement from the following:

- 1) GGD1 for color images: GGD modeling of color image sub-bands (4.13) using *MM* estimation and similarity metric using the symmetric KLD in (4.14). Two parameters (feature vector elements) are estimated for each sub-band.
- 2) GGD2 for color images: GGD modeling of color image sub-bands (4.13) using *ML* estimation and similarity metric using the symmetric KLD in (4.14). Two parameters (feature vector elements) are estimated for each sub-band.
- 3) GGD1+GMapp: GGD modeling (4.3) of each color channel detail sub-bands using *MM* estimation, modeling of each color approximation sub-band using GMM (4.17) and the sum (over all color channels) of symmetric KLDs in (4.5) and (4.18) as a similarity metric. Two parameters (feature vector elements) are estimated for each detail sub-band and five parameters are estimated for each approximation sub-band.
- 4) GGD2+GMapp: GGD modeling (4.3) of each color channel detail sub-bands using *ML* estimation, modeling of each color approximation sub-band using GMM (4.17) and the sum (over all color channels) of symmetric KLDs in (4.5) and (4.18) as a similarity metric. Two parameters (feature vector elements) are estimated for each detail sub-band and five parameters are estimated for each approximation sub-band.
- 5) GGD1+GM: GGD modeling (4.3) of grayscale image detail sub-bands using *MM* estimation, modeling of the entire color image using GMM and the sum of the symmetric KLDs in (4.5) and (4.18) as a similarity metric. Two parameters (feature vector elements) are estimated for each detail sub-band and nineteen parameters are estimated for the entire color image.

Feature extraction method	GGD modeling	GMM modeling	Similarity metric	Feature vector components
1) GGD1	<i>MM</i> estimation on each color image sub-band (4.13)	none	Symmetric KLD (4.14)	2/sub-band
2) GGD2	<i>ML</i> estimation on each color image sub-band (4.13)	none		2/sub-band
3) GGD1+GMapp	For each color channel, <i>MM</i> estimation on each detail sub-band (4.3)	Approximation sub-band for each color channel (4.17)	Sum of symmetric KLDs (4.5) and (4.18) over all color channels	2/detail sub-band, 5/approximation sub-band
4) GGD2+GMapp	For each color channel, <i>ML</i> estimation on each detail sub-band (4.3)	Approximation sub-band for each color channel (4.17)	Sum of symmetric KLDs (4.5) and (4.18) over all color channel	2/detail sub-band, 5/approximation sub-band
5) GGD1+GM	For grayscale image, <i>MM</i> estimation on each detail sub-band (4.3)	Entire color image	Sum of symmetric KLDs (4.5) and (4.18)	2/detail sub-band, 19 for color image

Table 5.4 Color texture feature extraction methods: associated models and similarity metrics.

Based on the experimental results and conclusions drawn in the previous section 2.1, the energy-based modeling is discarded in favor of GGD and GMM, and the RCT-Plus decomposition is performed using 3 scale levels and 8 directions ( $L=3$ ,  $D=[8, 8, 8]$ ), thus producing 24 detail sub-bands and one approximation sub-band for each color channel from an RGB or YCbCr image. Consequently, the size of the extracted feature vector in methods 1) and 2) is 150 (50 elements per color channel). In methods 3) and 4), 48 parameters are extracted from the detail sub-bands while 5 (see Table 4.1) parameters are from the approximation sub-band of each color channel for a total vector size of 159 elements. In method 5), the feature vector includes 67 elements, 48 of which come from the grayscale detail sub-bands and 19 (See Table 4.1) from the GMM modeling of the entire color image.

It should be noted that the proposed CBIR schemes 1), 2), 3) and 4) are also applicable to grayscale images (considered as single-color channel images) and this allows us to compare the results and show the contribution of the color components in the discrimination of textures.

Table 5.5 illustrates a sample of the achieved average retrieval rates ( $AR\%$ ) according to the number of top matches considered, in two color datasets, namely, VisTex-40 (Top16, 640 queries) and Stex (Top16, 7616 queries). As it appears in Table 5.5, we could conclude the following:

- 1- Joint color texture features improve the retrieval results substantially in comparison to grayscale texture features.
- 2- Using Gaussian Mixtures (GMM) to model the approximation sub-band fits more accurately than the GGD model.
- 3- In terms of retrieval performance, the YCbCr color space always shows a slight improvement in comparison to the RGB color space.
- 4- The GGD1+GM method appears to be the best choice for color texture retrieval among the proposed methods. Indeed, its retrieval rates are the highest ones while its feature vector size (67 elements) is the most compact (159 for GGD1+GMapp and 150 for GGD1 and GGD2).

CBIR schema	Color space	Feature vector size	AR% VisTex-40	AR% Stex
GGD2	Grayscale	50	75.24	51.14
	RGB	150	81.58	58.14
	YCbCr	150	84.36	69.20
GGD1	Grayscale	50	77.15	51.89
	RGB	150	83.43	59.29
	YCbCr	150	85.93	70.67
GGD2+GMapp	Grayscale	53 (48+5)	82.53	53.14
	RGB	159 (48×3+15)	84.91	63.28
	YCbCr	159 (48×3+15)	89.53	76.96
GGD1+GMapp	Grayscale	53 (48+5)	82.77	53.64
	RGB	159 (48×3+15)	85.15	63.72
	YCbCr	159 (48×3+15)	89.83	77.20
GGD1+GM	Gray + RGB	67 (48+19)	91.77	76.85
	Gray + YCbCr	<b>67 (48+19)</b>	<b>92.26</b>	<b>77.63</b>

Table 5.5 Average retrieval rates ( $AR\%$ ) in the Top16 images. The compared CBIR schemes operate on RCT-Plus decomposition with  $D=[8, 8, 8]$  and  $L=3$ , and GMM modeling is performed with  $k=2$ .

Furthermore, we conducted LBP-ED (Local Binary Pattern and Euclidean distance) texture feature extraction with eight neighbors for comparison to our approaches. The LBP function is available in the *Image Processing Toolbox* of Matlab [77].

We also compared many existing methods [11], [32], [33], [24], [80] [81], [35] to our proposed methods, namely, GGD1+GM, GGD1+GMapp and GGD2+GMapp. Table 5.6 presents the color textures feature extraction methods applied on Stex and VisTex-40 datasets. Correspondingly, Table 5.7 presents the grayscale texture feature extraction methods on Stex, VisTex-40, Kylberg-27 and Kylberg-28 datasets. For the Stex which is a challenging dataset, we achieved a retrieval rate of 77.63% using GGD1+GM method, while the best rate achieved by the compared state-of-the-art method is only 73.70%. Also, the corresponding feature vector size increases from 67 to 776 elements respectively. For the Kylberg dataset, we gained between 10 to 25 percentage in retrieval rate compared to LBP-ED and LDPVBP methods.

Consequently, our proposed methods perform very well, in comparison to many existing methods, both in terms of feature vector compactness and average retrieval rates ( $AR\%$ ).

Also, complimentary results from Kylberg-27 and Kylberg-28 are available in Table 0.2.

CBIR schema	Feature vector size	$AR\%$ -Top16 Stex	$AR\%$ -Top16 VisTex-40
DWT/GGD [11], [80]	54	49.30	82.00
Multivar. Power Exp. [32] The Multivariate Power Exponential modeling of DWT coefficients	54	71.30	91.20
DWT/ GGamma [33]	81	52.90	81.00
EMM [24] Embedded multiresolution mixtures of Gaussian modeling of DCT coefficients	776	73.70	88.90
Gaussian Copula GGD [80]	63	65.20	87.50
DTCWT Weibull [81]	108	58.80	84.00
Gaussian Copula Weibull [80]	207	70.60	89.50
Gaussian Copula Gamma [80]	207	69.40	89.10
LBP-ED (RGB)	177	57.00	85.10
LBP-ED (YCbCr)	177	66.92	85.44
<b>GGD1+GM (YCbCr)</b>	<b>67</b>	<b>77.63</b>	<b>92.26</b>

Table 5.6 Comparison of the proposed color texture GGD1+GM method to various state of art methods in terms of feature vector length and CBIR average retrieval rates  $AR\%$  in the Top16 images.

CBIR schema	Feature vector size	$AR\%$ -Top16 VisTex-40	$AR\%$ -Top40 Kylberg-27	$AR\%$ -Top160 Kylberg-28	$AR\%$ -Top16 Stex
LDPVBP [35]	32	71.79	-	39.46	-
LBP-ED	59	75.74	61.24	53.46	50.70
GGD2+GMapp	53	82.53	69.09	63.25	53.14
GGD1+GMapp	<b>53</b>	<b>82.77</b>	<b>70.04</b>	<b>63.89</b>	<b>53.64</b>

Table 5.7 Comparison of the proposed grayscale texture methods, GGD1+GMapp and GGD2+GMapp, to the existing methods LBP-ED and LDPVBP (Local directional peak valley binary pattern [35]) in terms of feature vector length and CBIR average retrieval rates  $AR\%$  on Grayscale image datasets.

### 3. Integrating a learning approach into the CBIR framework

In this section, the image search and retrieval process is improved through a learning-based approach where the images of the dataset are classified using an adapted similarity metric to the statistical modeling of the RCT-Plus transform. A query is then first classified to select the best texture class after which the retained class images are ranked to select the top ones.

#### 3.1. ML algorithms selection

In order to study the performance of a selection of supervised ML algorithms such as KNN, SVM, Decision tree, linear discrimination, and Quadratic discrimination, a learning phase with cross-validation is conducted for each algorithm using:

- VisTex-40 dataset features derived from RCT-Plus transform (with  $L=3$ ,  $D=8$ ) and one of these modeling methods GGD1, GGD2 or E2,
- a choice of parameters for each algorithm such as  $K$  value and Euclidean distance in KNN or kernel type in SVM,
- various data split scenarios in the  $n$ -fold cross-validation procedure.

From the sample of experimental results depicted in Table 5.8, the best obtained *Accuracy* value is 99.69% for KNN (with  $K=1$ ) and 98.40% for linear SVM. In general, KNN and SVM learning models were more accurate than Decision trees and Discriminant analysis models. Moreover, in most cases, KNN and SVM achieved high *Accuracy* levels when using GGD1 and GGD2 feature extraction methods. Therefore, we consider that either KNN or SVM algorithms are a promising approach in terms of classifier *Accuracy* to be incorporated in the ML-CBIR framework. Also, GGD1 and GGD2 are more efficient compared to E2.



ML algorithms	Feature extraction methods using RCT-Plus		
	GGD1	GGD2	E2
KNN (K=1)	<b>99.69</b>	<b>98.00</b>	<b>96.95</b>
KNN (K=10)	88.36	88.92	86.91
SVM (Linear kernel)	<b>98.40</b>	<b>98.36</b>	<b>97.00</b>
SVM (Gaussian kernel)	96.63	96.76	94.42
SVM (Quadratic kernel)	97.33	97.33	95.96
SVM (Cubic kernel)	96.93	96.93	95.80
Decision trees	85.52	85.29	87.83
Quadratic Discrimination	96.12	94.87	91.42
Linear Discrimination	90.33	90.50	87.25

Table 5.8 Accuracy measures (%) of compared ML algorithms on the whole VisTex-40 dataset. The learning phase is held with 5-fold cross-validation. Trained features are derived from GGD1, GGD2, and E2 methods using RCT-Plus.

### 3.2. ML-CBIR results, comparison and discussion

To evaluate the retrieval performance of the proposed ML-CBIR framework using texture discrimination, an experimental setup is conducted by trained KNN and SVM models over 600 images from the grayscale VisTex-40 dataset. The remaining 40 images (one per each texture class) are reserved as test data. The learned image features are based on RCT-Plus image transformation followed by either GGD1 or GGD2 modeling method.

Each image of VisTex-40 is considered as a query image (including the remaining 40 test images) and submitted to the ML-CBIR system. First, the trained classifier is used to predict the class membership (class label) of a given query image. Next, all images from the predicted class are retrieved and ranked according to the similarity measurement using either KLD or Euclidean distance. Finally, the  $N$  first images are displayed as being the  $N$  most similar images to the query ( $TopN$  retrieved images).

During this experimental setup, many other parameters are tuned such as:

- the number of neighbors (K=1, 2, 3, 4, 5, ...) in the KNN algorithm and kernel type in SVM.
- the number of data partitions ( $n$ -fold) in the cross-validation procedure.

The sample of results reported in Table 5.9 show a performance comparison of ML-CBIR schemes in terms of average retrieval rates ( $AR\%$ ) in the Top16

images and classifier *Accuracy* corresponding to a learning phase with 5-fold and 10-fold cross validations. The considered ML algorithms are KNN (with K=1) and SVM (with Linear kernel). One can see that, all of the two feature extraction methods operating on RCT-Plus, namely GGD1 and GGD2, achieved average retrieval rates (*AR%*) that are higher than 99%. Indeed, the number of false class membership predictions are very low (ranging from 0 to 5 over 640 queries). It is also worth mentioning that SVM results are the best and slightly higher than KNN ones.

Figure 5.12 illustrates an example of a KNN-CBIR system which made false class membership predictions to 3 queries among 640. The first column of the table displays the submitted queries to the system while the second column illustrates the wrong predicted class labels. In each case, the Top 16 retrieved images are non-relevant to the query. Therefore, the average retrieval rate (*AR%*) is about 99.5%.

The proposed ML-CBIR system operating on RCT-Plus image features achieved nearly perfect retrieval results on VisTex-40 dataset. It is also more efficient and more discriminative than conventional CBIR. Indeed, KNN-CBIR and SVM-CBIR schemes yield significant improvements in comparison to the conventional CBIR scheme in terms of retrieval rates (see Table 5.10).

<b>ML algorithm: KNN (K =1)</b>						
	5-fold cross-validation			10-fold cross-validation		
	<i>AR%</i>	<i>False predictions</i>	<i>Classifier Accuracy</i>	<i>AR%</i>	<i>False predictions</i>	<i>Classifier Accuracy</i>
GGD1	99.69	2	97.33	99.69	2	98.17
GGD2	99.53	3	97.33	99.84	1	97.67
<b>ML algorithm: SVM (Linear kernel)</b>						
	5-fold cross-validation			10-fold cross-validation		
	<i>AR%</i>	<i>False predictions</i>	<i>Classifier Accuracy</i>	<i>AR%</i>	<i>False predictions</i>	<i>Classifier Accuracy</i>
GGD1	100.00	0	98.50	100.00	0	98.67
GGD2	100.00	0	98.50	100.00	0	98.67

Table 5.9 Performance comparison of ML-CBIR schemes in terms of average retrieval rates (*AR%*), number of false predictions and classifier *Accuracy* corresponding to a learning phase with 5-fold and 10-fold cross-validations. The considered ML algorithms are KNN and SVM and texture dataset is VisTex-40 [7].

Feature extraction using RCT-Plus	<i>AR%</i> (KNN-CBIR)	<i>AR%</i> (Trad. CBIR)	Difference%
GGD1	99.69	77.15	+22.54
GGD2	99.53	75.24	+24.29
	<i>AR%</i> (SVM-CBIR)	<i>AR%</i> (Trad. CBIR)	Difference%
GGD1	100.00	77.15	+22.85
GGD2	100.00	75.24	+24.76

Table 5.10 Performance comparison of ML-CBIR schemes vs. conventional CBIR schemes in terms of average retrieval rates (*AR%*) over 640 image queries from VisTex-40 dataset.

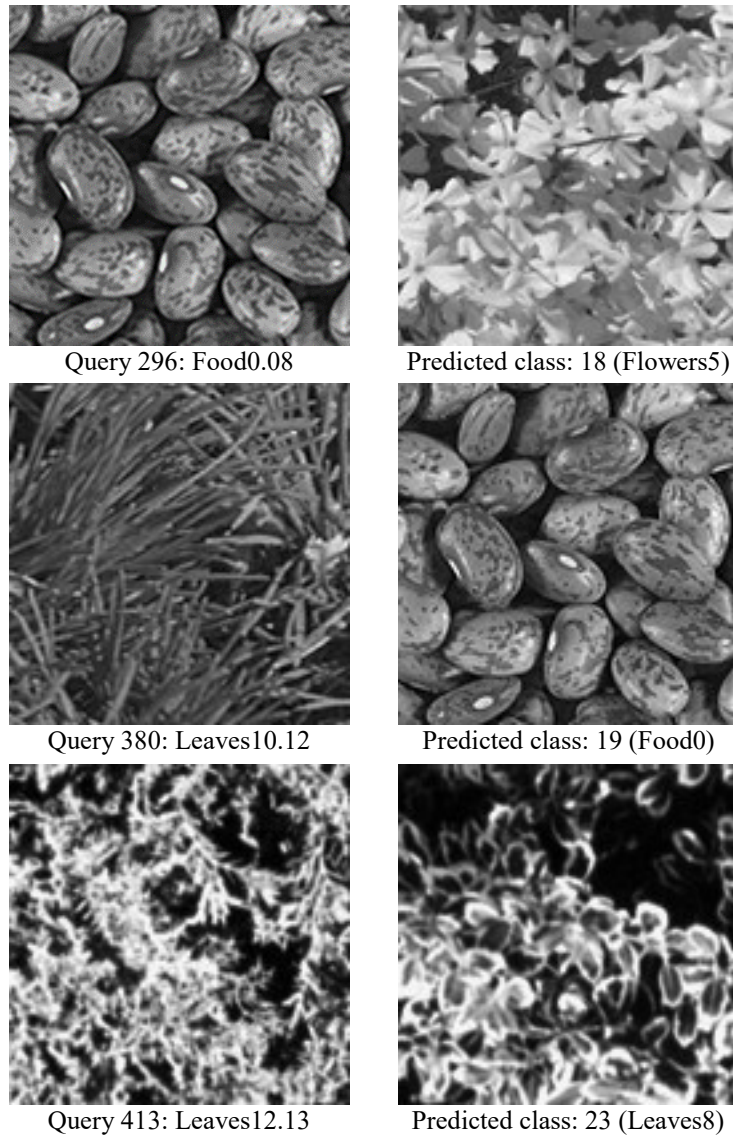


Figure 5.12 Examples of KNN-CBIR retrieval using GGD2 on VisTex-40. For each submitted query (left column), the predicted class membership (right column) is false.

To evaluate and validate the retrieval rate of texture and color texture ML-CBIR each classifier model (KNN and SVM) is learned on a set of training features from four distinguished textural datasets. The learned image features are based on RCT-Plus followed by GGD1 and GGD2 modeling for grayscale images and modeling of either color approximation sub-band or the whole color image using GMM (See Table 5.4). The training and testing data are organized as follows:

VisTex-40 does exist in three versions (RGB, YCbCr, and grayscale). The training set (labeled index) corresponds to 600 images (15 per class) while 40 images are reserved for the online phase test.

Kylberg-27 is only available in grayscale. The training set (labeled index) corresponds to 405 images (15 per class) while 675 images are reserved for the online phase test.

Kylberg-28 is only available in grayscale. The training set (labeled index) corresponds to 1680 images (60 per class) while 2800 images are reserved for the online phase test.

Stex does exist in three versions (RGB, YCbCr, and grayscale): The training set (labeled index) corresponds to 1904 images (4 per class) while 5712 images are reserved for the online phase test.

Each image is considered as a query and submitted to the ML-CBIR system. First, the trained classifier is used to predict the class membership (class label) of a given query image. Next, all images from the predicted class are retrieved and ranked according to the similarity measurement. Finally, the  $N$  first images are displayed as being the  $N$  most similar images to the query (*TopN* retrieved images).

Table 5.11 and Table 5.12 report the performance comparison of ML-CBIR schemes in terms of average retrieval rates ( $AR\%$ ) in 4 texture datasets. Note that the considered ML algorithms are KNN (with  $K=1$ ) and SVM (with linear and KLD kernels) and the training phase has been implemented using 10-fold cross-validations. (See Table 5.9). One can see that:

- All the feature extraction methods operating on RCT-Plus and using both GGD and GMM models, achieved average retrieval rates ( $AR\%$ ) that are higher than other methods such as GGD1 and GGD2.

- It is also worth mentioning that, in general, using SVM algorithm instead of KNN does not significantly affect the retrieval results on a given dataset.
- Using a YCbCr color dataset yields marginally better retrieval rates compared to RGB and grayscale formats.
- The ML approach for CBIR significantly improved the retrieval rates compared to conventional CBIR schemes. For example, in the YCbCr Stex dataset, using the method GGD1+GM improved the retrieval rate from 77.63% to 97.10%. Also, in the Kylberg-28 which is a grayscale dataset, the retrieval rate  $AR\%$  has increased from 63.89% to 99.04% while applying GGD1+GMapp method.
- This ML-CBIR framework not only enhances the retrieval score but also makes it computationally efficient, thus, avoiding the requirement of comparing a query with all the images of the dataset.

Table 5.13 reports some of the recent states of art compared to our proposed methods for the VisTex-40 dataset. In order to have a fair comparison, we just reported our best grayscale method named GGD1+GMapp. As it appears, our proposed methods achieved higher retrieval rates compared to other state-of-art methods including some deep learning based methods such as CNN-25 [82] and DBN [83]. Complementary results are illustrated in Table 0.3 and Table 0.4 (Annex IV).

ML-CBIR (KNN)	Color Space	VisTex-40 Top16	Kylberg-27 Top40	Kylberg-28 Top160	Stex Top16
GGD1	Grayscale	99.69	97.69	97.32	88.29
	RGB	99.69	-	-	91.78
	YCbCr	99.84	-	-	95.52
GGD2	Grayscale	99.53	97.31	96.90	87.59
	RGB	99.69	-	-	90.93
	YCbCr	99.84	-	-	95.13
GGD1+GMapp	Grayscale	99.64	<b>98.61</b>	<b>98.50</b>	90.61
	RGB	99.84	-	-	93.97
	YCbCr	100.00	-	-	96.09
GGD1+GM	RGB	100.00	-	-	94.24
	YCbCr	100.00	-	-	<b>97.10</b>

Table 5.11 Average retrieval rates ( $AR\%$ ) of ML-CBIR using fine (K=1) KNN.

<b>ML-CBIR (SVM)</b>	<b>Color Space</b>	<b>VisTex-40 Top16</b>	<b>Kylberg-27 Top40</b>	<b>Kylberg-28 Top160</b>	<b>Stex Top16</b>
GGD1	Grayscale	99.69	97.96	98.75	87.64
	RGB	99.84	-	-	90.05
	YCbCr	100.00	-	-	93.21
GGD2	Grayscale	88.69	97.59	98.66	87.24
	RGB	99.84	-	-	89.67
	YCbCr	99.84	-	-	93.04
GGD1+GMapp	Grayscale	99.69	<b>98.70</b>	<b>99.04</b>	89.92
	RGB	100.00	-	-	93.49
	YCbCr	100.00	-	-	95.65
GGD1+GM	RGB	100.00	-	-	94.10
	YCbCr	100.00	-	-	<b>96.99</b>

Table 5.12 Average retrieval rates ( $AR\%$ ) ML-CBIR using SVM.

<b>CBIR schema</b>	<b>Classifier</b>	<b><math>AR\%</math>-Top16 VisTex-40</b>
CNN with 25 layers [82]	CNN-25	95.28
LBP-ED	SVM (Linear)	86.06
LBP-ED	KNN(K=1)	98.59
Directional Magnitude Local Hexadecimal Pattern (DMLHP) [23]	Ensemble Subspace Discriminant (ESD)	98.00
Deep Belief Network- Similarity-Based Indexing DBN- SBI [83]	DBN	98.45
Local Neighbor Pattern (LNP) [19]	SVM (Linear)	98.40
Local Neighbor Pattern (LNP) [19]	KNN(K=1)	99.5
GGD1+GMapp	KNN(K=1)	<b>99.69</b>
GGD1+GMapp	SVM (KLD)	<b>99.69</b>

Table 5.13 Comparison of the proposed grayscale texture GGD1+GMapp method to various state-of-the-art methods in terms of CBIR average retrieval rates  $AR\%$  in the Top16 images for VisTex-40 dataset.

## 4. Abnormality detection experiments and results

This section is dedicated to Abnormality detection experiments and results. Starting with the choice of parameters we concluded from section 3 for texture feature extraction, we validate the block-based approach for abnormality detection (see Sections 4.1-4.4 in Chapter 4). Also, comparing the obtained results with the deep learning approach (U-net) confirms our methodology in the end.

### *4.1. Abnormality detection choice of parameters and methods*

In the previous section, Table 5.11 and Table 5.12 demonstrated that the GGD1+GM method (GGD modeling of grayscale image detail sub-bands, from RCT-Plus, using MM estimation, and the modeling of entire color image using GMM) is a reliable choice for texture feature extraction in CBIR and ML-CBIR applications (or in texture retrieval applications). Consequently, to discriminate among normal and abnormal regions in skin *lesion* images (using KNN/SVM classifiers) we rely on both GGD for texture and GMM for color image characteristics. Also by experimenting with our features on many color spaces (YCbCr, RGB, HIS, and LAB), we chose YCbCr and RGB as suitable spaces for our texture features.

In the context of abnormality detection on image blocks of skin *lesions*, we experimented with different numbers of overlapping pixels between image blocks, color spaces, threshold values in labeling methods, ML algorithms, distance metrics in texture and color texture feature extraction and conciliation processes between grayscale and color label prediction to achieve final block-based abnormality detection.

## 4.2. Block-based detection

We selected two skin lesion datasets namely ISIC-32 and ISIC-42 from the *International Skin Imaging Collaboration* (ISIC) to conduct the experiments and validate our proposed block-based detection method. It is built on two main stages as the following:

- **The offline stage:** Firstly, we selected 28 images from ISIC-32 as a training dataset. Each image is subdivided into overlapping color blocks  $B_c(i)$  and grayscale blocks  $B(i)$ , with  $i=1 \dots N_B$ . The choice of color space could be either YCbCr or RGB. For example, we obtained 9856 color blocks of size  $128 \times 128$  pixels when the number of overlapping pixels is 64. Then we did a similar subdivision on the corresponding binary ground truth maps to create the block-based ground truths  $G(i)$ , with  $i=1 \dots N_B$ .

In order to extract the block-based texture features  $F(i)$ , we rely on GGD1 modeling of RCT-Plus decomposition (with  $D= [8, 8, 8]$  and  $L=3$ ) since this choice of parameters has been shown to be the best over many experiments. Also, we perform a  $k$ -component GMM modeling for each color block  $B_c(i)$  to extract a color texture feature vector  $F_C(i)$ . As defined in Section 4.1 of Chapter 4, (the labeling of local features) assigns two class labels,  $C_{label}$  and  $T_{label}$ , to each block according to *Labeling method 1* and *Labeling method 2* respectively. A threshold  $T\%$  has to be tuned in order to get  $C_{label}$  values (0 for *healthy* / 1 for *lesion*). Also, two thresholds  $T1\%$  and  $T2\%$  have to be tuned to get  $T_{label}$  values (0 for *non-border* / 1 for *border*). One example of thresholding in our experiments is 30% as  $T\%$  in *Labeling method 1*, 20% as  $T1\%$ , and 100% as  $T2\%$  in *Labeling method 2*.

As we have two parallel learning processes for classification purposes, a learning phase with cross-validation is conducted for each training dataset (*Color index* and *Texture index*) using:

- a choice of parameters for each algorithm such as K value and distance metric (based on either ED or KLD) in KNN or kernel type in SVM (Linear, KLD-based);



- various data split scenarios in the  $n$ -fold cross-validation procedure in order to select the best learning models  $CL_T$  and  $CL_C$  corresponding to texture features and color texture features respectively.

An example of parameter selection corresponds to KNN learning with  $K=7$ ,  $n=10$  and a KLD-based metric as defined in Equations (4.5) and (4.19). At the end of the learning process,  $CL_T$  and  $CL_C$  are saved for detection purposes during the online phase.

**The online stage:** The online detection phase is meant to detect skin abnormality in each block of a submitted test image. To evaluate the efficiency of the online phase, we use new test images ( $I_C$ ) from ISIC-32 and ISIC-42, each test image undergoes a block-based subdivision and feature extraction that obey the same parameters as the offline phase (block size, overlap, texture feature method, color texture feature method, etc.). In ISIC-32 all images are the same size while in ISIC-42 the variant size of images is the challenge during subdividing into overlapping blocks. According to the choice of parameters we exemplified in the offline phase, we formed 1408 and 27648 testing blocks regarding the 4 testing images in ISIC-32 and the 42 images in ISIC-42 respectively.

For prediction purposes, the extracted grayscale texture feature is used to classify each test block using the classifier model  $CL_T$  and obtain its label  $T_{label}$  (0 as *non-border* and 1 as *border*) and its corresponding score ( $T_{score}$ ). Similarly, the extracted color texture feature is used to classify each test block using the classifier model  $CL_C$  and obtain its label  $C_{label}$  (0 as *healthy* and 1 as *lesion*) and its corresponding score ( $C_{score}$ ). Then, a block-based label conciliation (See Chapter 4, Section 4.3) is applied in order to derive a single *label* for each block (0 as *healthy* and 1 as *lesion*).

Furthermore, to localize and visualize the abnormality detection results, a binary detection map is constructed by mosaicking a set of blocks tagged by the corresponding predicted class *label* (0 as black and 1 as white). Superimposing the constructed detection map on the original test image  $I_C$  shows the final detected results and determines the abnormal skin image regions.

In order to evaluate the efficiency of the abnormality detection for a given test dataset (ISIC-32, ISIC-42, etc.), we first compare the predicted labels to the corresponding ground truth ones and then calculate the *Precision*, *Recall*, and detection *Accuracy* values (See the above Section 1.7).

Table 5.14 reports the final results for the abnormality detection approach by joining together texture and color texture features in ISIC-32 test images. One can notice that using KLD-based distances instead of ED distances in KNN is more effective in terms of *Accuracy*, *Precision* and *Recall* in the detection results. For example, the *Precision* of detection increases from 72.03% to 85.09% for the RGB ISIC-32 test data. In Figure 5.13, a sample of mapped detection results corresponding to RGB color space and SVM classifier is shown. For each test image, the red contour presents the block-based ground truth while the white contour illustrates the resulting detection map that distinguishes the *lesion* from the *healthy* region. For the four test images, our approach succeeds to achieve an *Accuracy*, *Recall*, and *Precision* of 96.88%, 97.22% and 93.92% respectively.

In order to show the contribution of the joint color texture features to the abnormality detection we compare the obtained result from grayscale texture-based detection, color texture detection and joint color texture detection. Table 5.15 and Table 5.16 show examples of this comparison using RGB and YCbCr color spaces in the ISIC-42 dataset. It is clear that the joint approach improves the *Accuracy* of detection by up to 12% (from 87.55% to 99.68%) and *Precision* by up to 21% (from 75.70% to 97.29%) for SVM classifier and YCbCr color space. Also, in both KNN and SVM, using the joint color texture features gives the best trade-off between *Precision* and *Recall*. Figure 5.14 shows a sample of mapped detection results on ISIC-42 images (corresponding to YCbCr and KNN). The results that are depicted in the row d) confirm that there is a reliable match between the block-based ground truth and the achieved detection.

Overlapping is an approach that can be advantageously used to enhance the localization precision of the detected abnormal regions. For example, we tested 5, 16, 32, and 64-pixel overlapped blocks, and among all the results, the 64-pixel overlap between blocks was chosen for better detection and segmentation. A part of preliminary results is provided in Annex V.

KNN with K=7					
Color Space	Threshold for <i>Labeling Method 1</i> ( $T\%$ ) and distance	Threshold for <i>Labeling Method 2</i> ( $T1\%-T2\%$ ) and distance	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
RGB	30% (KLD)	20%-100% (ED)	90.63	79.85	72.03
RGB	30% (KLD)	20%-100% (KLD)	94.44	100.00	85.09
YCbCr	30% (KLD)	20%-100% (ED)	92.47	97.80	83.70
YCbCr	30% (KLD)	20%-100% (KLD)	<b>94.91</b>	<b>100.00</b>	<b>86.16</b>
SVM with KLD-based kernel					
Color Space	Threshold for <i>Labeling Method 1</i> ( $T\%$ )	Threshold for <i>Labeling Method 2</i> ( $T1\%-T2\%$ )	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
RGB	30%	20%-100%	96.88	97.22	93.92
YCbCr	30%	20%-100%	<b>98.01</b>	<b>97.94</b>	<b>97.28</b>

Table 5.14 A sample of block-based abnormality detection results corresponding to 4 test images from ISIC-32 dataset. RCT-Pus, GGD and GMM ( $k=2$ ) modeling of texture and color texture blocks are used.

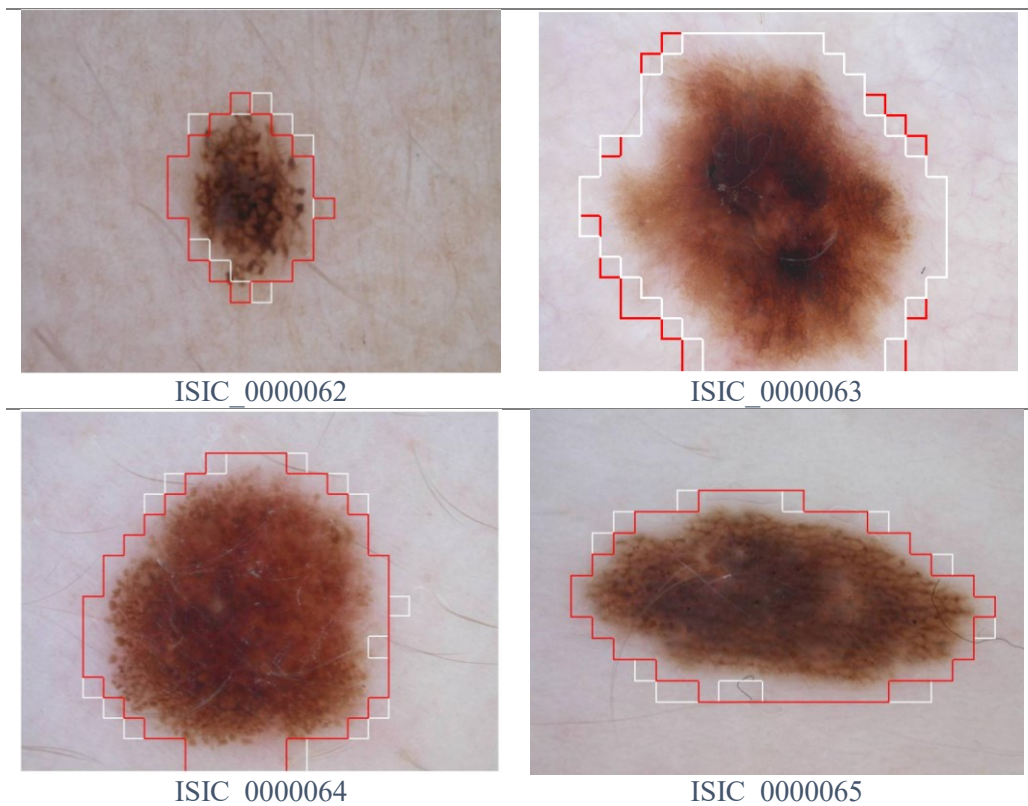


Figure 5.13 Examples of abnormality detection results mapped onto 4 original test images from ISIC-32. For each image, the red contour represents the original block-based ground truth, and the white contour shows the resulting abnormality detection map (*healthy/lesion*).

KNN with K=7 and KLD-based distance					
Color Space	Threshold for <i>Labeling Method 1</i> ( <i>T%</i> )	Threshold for <i>Labeling Method 2</i> ( <i>T1%-T2%</i> )	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
<b>a) Grayscale</b>	-----	20%-100%	97.02	87.69	81.93
<b>b) RGB</b>	30%	-----	95.97	<b>96.48</b>	85.34
<b>c) Grayscale and RGB</b>	30%	20%-100%	<b>98.16</b>	95.24	<b>92.79</b>
SVM with KLD-based kernel					
Color Space	Threshold for <i>Labeling Method 1</i> ( <i>T%</i> )	Threshold for <i>Labeling Method 2</i> ( <i>T1%-T2%</i> )	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
<b>a) Grayscale</b>	-----	20%-100%	87.55	83.69	75.70
<b>b) RGB</b>	30%	-----	95.81	94.19	80.65
<b>c) Grayscale and RGB</b>	30%	20%-100%	<b>99.60</b>	<b>97.91</b>	<b>94.05</b>

Table 5.15 A comparison between block-based abnormality detection results using: a) texture features, b) color texture features and c) joint color texture features from RGB and grayscale images in the ISIC-42 dataset.

KNN with K=7 and KLD-based distance					
Color space	Threshold for <i>Labeling Method 1</i> ( <i>T%</i> )	Threshold for <i>Labeling Method 2</i> ( <i>T1%-T2%</i> )	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
<b>a) Grayscale</b>	-----	20%-100%	97.02	87.69	81.93
<b>b) YCbCr</b>	30%	-----	96.21	<b>96.95</b>	85.94
<b>c) Grayscale and YCbCr</b>	30%	20%-100%	<b>99.16</b>	95.74	<b>93.54</b>
SVM with KLD-based kernel					
Color space	Threshold for <i>Labeling Method 1</i> ( <i>T%</i> )	Threshold for <i>Labeling Method 2</i> ( <i>T1%-T2%</i> )	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>
<b>a) Grayscale</b>	-----	20%-100%	87.55	83.69	75.70
<b>b) YCbCr</b>	30%	-----	96.02	95.72	83.18
<b>c) Grayscale and YCbCr</b>	30%	20%-100%	<b>99.68</b>	<b>98.68</b>	<b>97.29</b>

Table 5.16 A comparison between block-based abnormality detection results using: a) texture features, b) color texture features and c) joint color texture features from YCbCr and grayscale images in the ISIC-42 dataset.

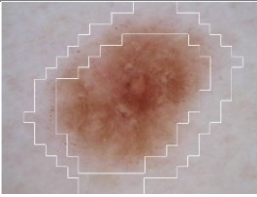
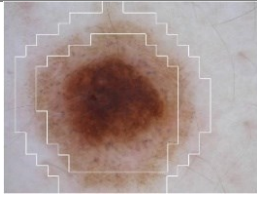
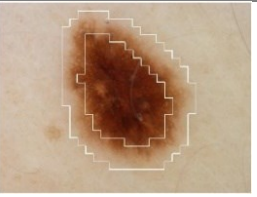
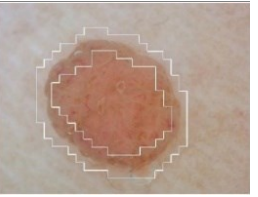

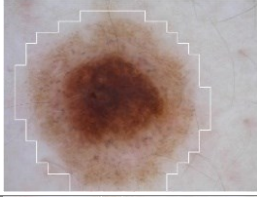
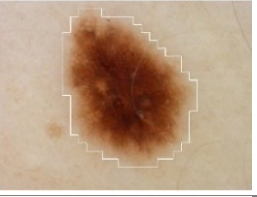


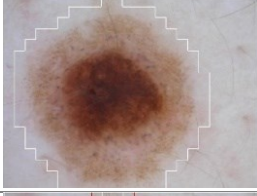
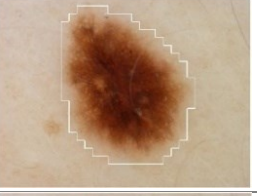

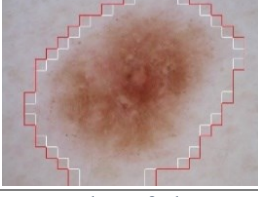
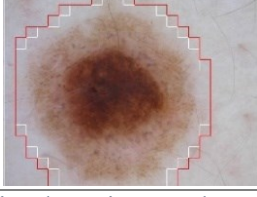
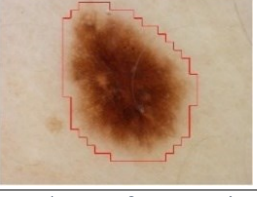
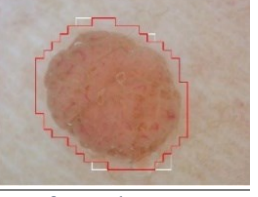
	ISIC_0000058	ISIC_0000060	ISIC_0000082	ISIC_0000121
a) Texture <i>border/non-border</i> detection				
b) Color <i>healthy/lesion</i> detection				
c) Conciliation <i>healthy/lesion</i> detection				
d) Comparison of c) to block-based ground truth				

Figure 5.14 Examples of abnormality detection results mapped onto four test images from the ISIC-42 dataset. The rows show: a) Texture *border/non-border* detection maps; b) Color *healthy/lesion* detection maps; c) Conciliation between texture and color detection maps and d) a comparison between the achieved detection results and the block-based ground truths (red contours).

In addition, the discriminative power of the extracted texture and color texture features, using jointly texture and color through a reasonable labeling method with fair block-based conciliation rules, choosing the right amount of overlap between blocks and accurate ML algorithms are among the variables to improve block-based abnormality detection. From all the achieved results including those presented in previous Tables, the following conclusions can be drawn:

- YCbCr is better than RGB color space for color feature extraction;
- KLD-based distances provide better results than Euclidean distance (ED);
- Using SVM algorithm provides slightly better results than KNN.

### 4.3. Texture segmentation and comparison with a deep learning approach

To perform texture segmentation on dermoscopic images, the interactive foreground extraction algorithm using Grabcut is leveraged to segment the *lesion* against the healthy skin (on a pixel basis). To initiate the segmentation process, we rely on our proposed grayscale RCT-Plus texture detection method. Indeed, we determine foreground and background masks by separating the inner and outer lines from *border/ non-border* abnormality detection map (See Figure 4.17). Then, the texture Grabcut segmentation applies iteratively starting from a random selection of foreground and background points that are marked on the original image.

The Grabcut segmentation results are compared to the deep learning methods (namely U-net\_50, U-net\_250 and U-net\_500) that are trained on various datasets (50 images, 250 images and 500 images respectively). Table 5.17 and Table 5.18 report the segmentation results that are achieved on ISIC-32 and ISIC-42 datasets. For comparison, in Grabcut, the size of training data is 28 images from ISIC-32, while it is, in the U-net methods, about 50, 250 and 500 images from the international ISIC dataset. Our methodology's success appears, especially for small training datasets. Indeed, the *Accuracy* of segmentation that is obtained with U-net\_50 is quite low comparatively to the other methods. Using the Grabcut approach gives the best trade-off between *Precision* and *Recall* of the segmentation. Also, the best achieved *Accuracy* value is up to 99.02% for Grabcut segmentation while the maximum *Accuracy* value achieved with U-net methods is 98.47% (corresponding to U-net\_250 segmentation of ISIC-42 images).

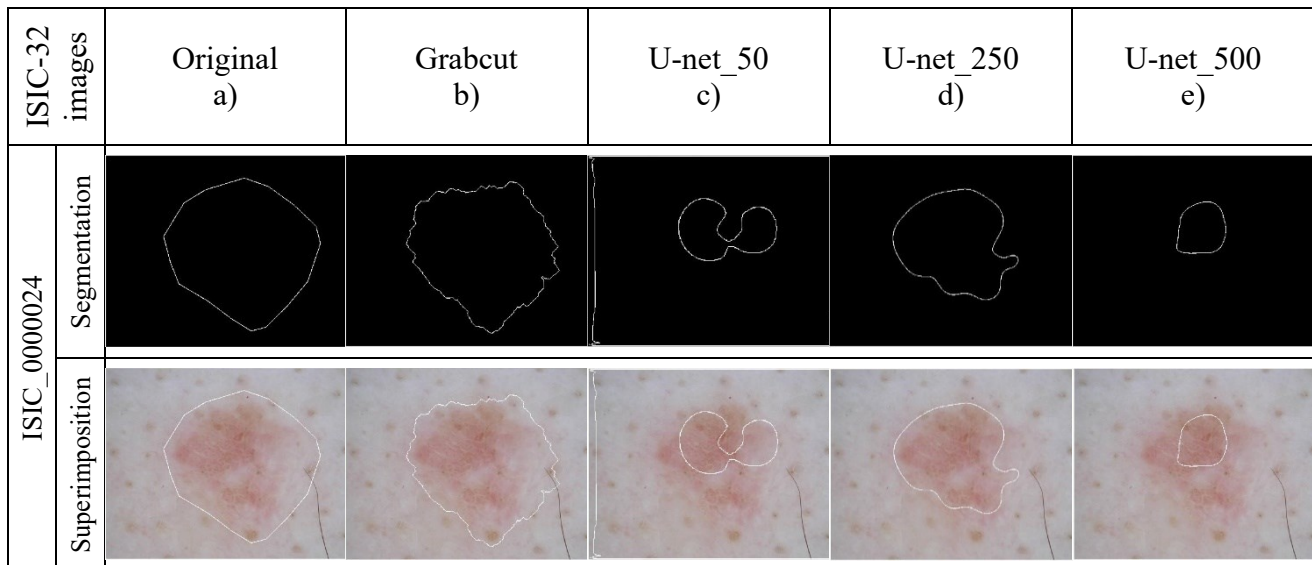
Figure 5.15 and Figure 5.16 compare original images and ground truth to image segmentation examples obtained from Grabcut and U-net methods on ISIC-42 and ISIC-32 datasets. One can notice that, in most examples, the visual quality of segmentation is better when using Grabcut instead of U-net methods.

Segmentation method on ISIC-32	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
U-net_50	90.02	<b>95.51</b>	74.63
U-net_250	96.89	88.69	92.99
U-net_500	95.14	91.38	88.63
Grabcut	<b>98.64</b>	94.43	<b>94.86</b>

Table 5.17 Comparison of segmentation results between U-net deep learning and Grabcut methods applied on ISIC-32 dataset.

Segmentation method on ISIC-42	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
U-net_50	96.09	86.98	83.89
U-net_250	98.47	83.49	93.83
U-net_500	97.70	81.59	92.64
Grabcut	<b>99.02</b>	<b>92.49</b>	<b>94.11</b>

Table 5.18 Comparison of segmentation results between U-net deep learning and Grabcut methods applied on ISIC-42 dataset.



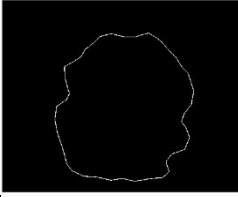

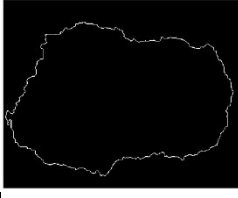
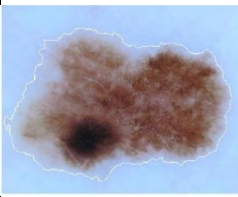
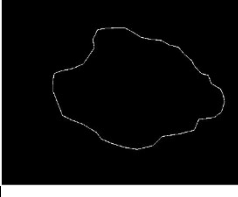
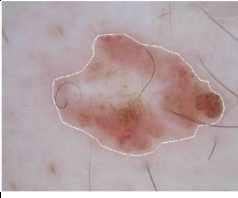
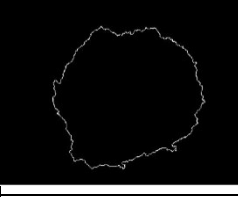

ISIC_0000032	Segmentation	
	Superimposition	
ISIC_0000038	Segmentation	
	Superimposition	
ISIC_0000050	Segmentation	
	Superimposition	
ISIC_0000061	Segmentation	
	Superimposition	



Figure 5.15 Examples of image segmentation results obtained on ISIC-32 dataset. The first column a) illustrates five ground truths and their superimposition on the corresponding original images. The columns b)-e) show segmentation results obtained by using Grabcut, U-net\_50, U-net\_250, and U-net\_500, respectively.

ISIC-42 images		Original a)	Grabcut b)	U-net_50 c)	U-net_250 d)	U-net_500 e)
ISIC_0000040	Segmentation					
	Superimposition					
ISIC_0000046	Segmentation					
	Superimposition					
ISIC_0000081	Segmentation					
	Superimposition					

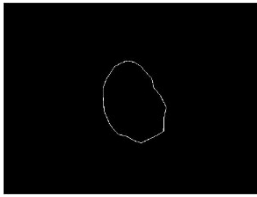
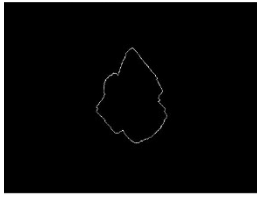
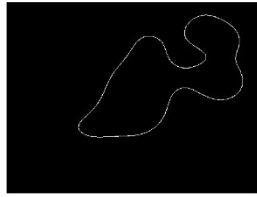
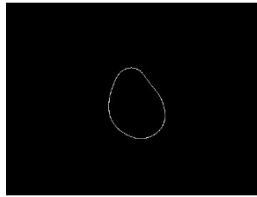
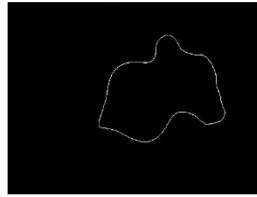





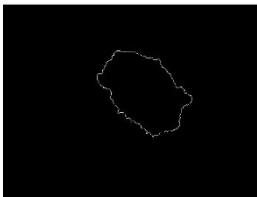
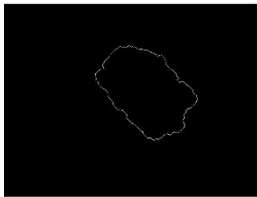
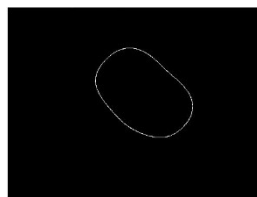
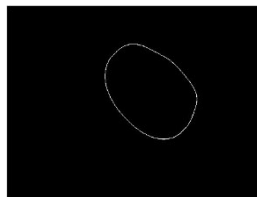
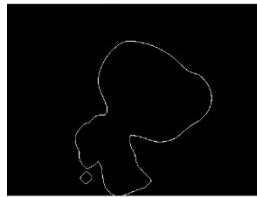





ISIC_0000095	Segmentation					
	Superimposition					
ISIC_0000154	Segmentation					
	Superimposition					

Figure 5.16 Examples of image segmentation results obtained on ISIC-42 dataset. The first column a) illustrates five ground truths and their superimposition on the corresponding original images. The columns b)-e) show segmentation results obtained by using Grabcut, U-net\_50, U-net\_250, and U-net\_500 U-net, respectively.

## CHAPTER 6. Conclusion and future work

In this thesis, we addressed the issue of texture analysis and discrimination with a new methodology based on parametric statistical modeling of multi-scale image representations. We developed novel methods for the extraction of relevant image features that achieved powerful characteristic discrimination in grayscale and color texture images while providing feature descriptors with reduced dimensionality. The major contributions of this work can be summarized as follows:

- A novel multi-scale image representation, named RCT-Plus, is proposed. It is as a variant of the contourlet transform that is redundant, rich in directional information and applicable to grayscale images and multi-channel color images.
- A hybrid statistical modeling approach combining the Generalized Gaussian Distribution (GGD) and the multivariate Gaussian Mixture Model (GMM) is designed to provide highly compact data models of texture images while ensuring accurate shape fitting and flexible adjustment to variability and multimodality in the texture data (color channels, high-frequency detail sub-bands, low-frequency approximation sub-bands). In addition, suitable similarity metrics based on Kullback-Leibler divergence are adapted accordingly.
- A joint exploitation of color and texture features in the multi-scale space is performed either by applying an early concatenation procedure at the feature level or a late conciliation procedure operating at the result level. This approach offers the benefit of extracting relevant information from a much richer data source and results in a variety of enhanced feature extraction methods and more powerful texture discrimination schemes.
- Supervised machine learning algorithms (KNN and SVM) are integrated into the processing system as key techniques of feature learning and multi-class classification to infer texture types on the extracted features and achieve improved performance in terms of texture discrimination.

Furthermore, the methodology we proposed for texture feature extraction, the statistical modeling of texture and color texture on multi-scale image representations, was applied according to the following strategies: global region-based and local region-based in order to respond to the specific needs of various application domains. To illustrate texture feature extraction according to these two strategies, two main applications namely, content-based image retrieval (CBIR) in texture datasets and abnormality detection in medical skin lesion images were considered in this work.

The global region-based texture feature extraction is extensively tested in the framework of conventional CBIR and machine learning CBIR (ML-CBIR) using the well-known grayscale and color datasets VisTex-40 [7], Kylberg-27 [37], Kylberg-28 [37] and Stex [25]. A comparison with state-of-the-art methods, including deep learning, showed that our proposed texture feature extraction methodology yields more successful results. In conventional CBIR, the use of our method GGD1+GM on the challenging YCbCr Stex dataset was very successful as it achieved the highest average retrieval rate ( $AR\%=77.63\%$ ) as well as the most compact feature vector with only 67 elements.

Moving from conventional CBIR to ML-CBIR significantly improved the retrieval results in all experimented datasets. For example, in the YCbCr Stex dataset, the retrieval rate increased from 77.63% to 97.10% for the GGD1+GM method. Also, in the Kylberg-28 which is a grayscale dataset, we improved  $AR\%$  from 63.89% to 99.04% while applying the GGD1+GMapp method. For both conventional CBIR and ML-CBIR, we came to the conclusions that: 1) In terms of retrieval performance, the use of the YCbCr color space has always shown slightly better results compared to the RGB color space; 2) Using GMM to model the color data and RCT-Plus approximation sub-bands fitted more accurately than the GGD model; 3) For both grayscale and color datasets, all the feature extraction methods operating on RCT-Plus and jointly using GGD and GMM models (GGD1+GMapp, GGD2+GMapp and GGD1+GM) achieved average retrieval rates ( $AR\%$ ) that are higher than other methods such as GGD1 and GGD2; 4) The GGD1+GM is the best method among all the proposed ones because it allows powerful characteristic discrimination of color texture images while providing the most compact feature representation.

In abnormality detection and segmentation, we used the local region-based strategy for feature extraction. The proposed GGD1+GM method has been chosen as it showed the best results in comparison to the other proposed methods for color texture feature extraction in CBIR and ML-CBIR applications. We experimented on two selected datasets from the International ISIC dataset [40], namely ISIC-32 and ISIC-42. The joint exploitation of color and texture features which is performed by a conciliation procedure operating on grayscale and color label prediction results, has improved the *Accuracy* of detection by up to 12% and *Precision* by up to 21% for the SVM classifier and YCbCr color space in the ISIC-42 dataset. It also offered the best trade-off between detection *Precision* and *Recall*.

For segmentation purposes, the obtained detection maps were integrated into the Grabcut segmentation application. Then, segmentation results were compared to the following deep learning methods, namely U-net\_50, U-net\_250 and U-net\_500. Deep learning-based classification, detection and segmentation are becoming very popular in the field of medical image processing and they require large datasets to be efficiently trained, while our proposed method is more classic and has been trained only on 28 images (9856 blocks). Also, it has great efficiency on small datasets. The Grabcut segmentation results showed the best trade-off between *Precision* and *Recall* and achieved the best *Accuracy* value of 99.02% in the ISIC-42 dataset.

We published a part of this research in [6], [68] and [84]. In addition, another publication is in preparation. As a future work, we believe that our proposed methodology can be used and extended to address the problem of multi-class classification of skin cancer in dermoscopic images. Indeed, neural network diagnoses of skin tumors are very popular in the field of medical image processing and automatic classification of skin cancer is considered a promising tool for the diagnosis of skin cancer, therefore as future works, we will explore the use of our developed texture feature extraction methods to do lesion classification and determine the type of skin cancer including Melanoma, Nevus, Angiosarcoma, Basal cell carcinoma, Cutaneous B-cell lymphoma, Cutaneous T-cell lymphoma, Dermatofibrosarcoma protuberans, Merkel cell carcinoma, Sebaceous carcinoma or Squamous cell carcinoma [85] [86] [87].

## References

- [1] M. Mirmehdi, X. Xie and J. Suri, Handbook of texture analysis, London: Imperial college press, 2008.
- [2] R. Datta, D. Joshi, J. Li and J. Z. Wang, "Image retrieval: ideas, influences, and trends of the new age," *ACM computing surveys*, vol. 40, no. 2, pp. 1-60, 2008.
- [3] R. C. Gonzalez and R. E. Woods, Digital image processing, 3rd Edition ed., Upper saddle river, NJ: Pearson, 2007.
- [4] L. Shen and L. Bai, "A review on Gabor wavelets for face recognition," *Pattern analysis & applications*, vol. 9, no. 2-3, pp. 273-292, 2006.
- [5] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 18, no. 10, pp. 959 - 971, 1996.
- [6] A. Rouhafzay and N. Baaziz, "Contourlet versus Gabor transform for texture feature extraction and image retrieval," in *15th International conference on signal image technology & internet based systems*, Naples, 2016.
- [7] "Vision texture dataset," MIT, [Online]. Available: <https://vismod.media.mit.edu/vismod/imagery/VisionTexture/Vistex.html>.
- [8] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Trans. on image processing*, vol. 14, no. 12, pp. 2091 - 2106, 2005.
- [9] A. L. Cunha, J. Zhou and M. N. Do, "The nonsubsampling contourlet transform: theory, design, and applications," *IEEE Trans. on image processing*, vol. 15, no. 10, pp. 3089 - 3101, 2006.
- [10] N. Baaziz, "Adaptive watermarking schemes based on a redundant contourlet transform," in *IEEE International conference on image processing*, Genova, 2005.

- [11] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," *IEEE Trans. on image processing*, vol. 11, no. 2, pp. 146-158, 2002.
- [12] M. H. Bharati, J. J. Liu and J. F. MacGregor, "Image texture analysis: methods and comparisons," *Chemometrics and intelligent laboratory systems*, vol. 72, p. 57 – 71, 2004.
- [13] M. S. Allili, N. Baaziz and M. Mejri, "Texture modeling using contourlets and finite mixtures of generalized gaussian distributions and applications," *IEEE Trans. on multimedia*, vol. 16, no. 3, pp. 772-784, 2014.
- [14] D. Yapi, M. S. Allili and N. Baaziz, "Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain," *IEEE Trans. on automation science and engineering*, vol. 15, no. 3, pp. 1014 - 1026, 2018.
- [15] "TILDA dataset for textile defect detection,," [Online]. Available: <http://lmb.informatik.uni-freiburg.de/research/dfg-texture/tilda>. [Accessed 5 Feb 2017].
- [16] M. Rafi Nazari and E. Fatemizadeh, "A CBIR system for human brain magnetic resonance image indexing," *International journal of computer applications*, vol. 7, no. 14, 2010.
- [17] Z. Abbas, M.-u. Rehman, S. Najam and S. Danish Rizvi, "An Efficient gray-Level co-occurrence matrix (GLCM) based approach towards classification of skin lesion," in *Amity international conference on artificial intelligence AICAI*, Dubai, 2019.
- [18] Y. Ben Salem and M. N. Abdelkrim, "Texture classification of fabric defects using machine learning," *International journal of electrical and computer engineering IJECE*, vol. 10, no. 4, p. 4390~4399, 2020.
- [19] M. Alrahhal and . K. Supreethi, "Content-based image retrieval using local patterns and supervised machine learning techniques," in *Amity international conference on artificial intelligence AICAI*, 2019.

- [20] "The dataset of faces Cambridge university computer laboratory," [Online]. Available: <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedataset.html>.
- [21] "Corel 1000 image dataset," University of california at irvine, 01 July 1999. [Online]. Available: [http://wang.ist.psu.edu/docs/related/..](http://wang.ist.psu.edu/docs/related/)
- [22] N. Alpaslan and K. Hanbay, "Multi-resolution intrinsic texture geometry-based local binary pattern for texture classification," *IEEE Access*, vol. 8, pp. 54415-54430, 2020.
- [23] A. Khan, A. Javed, M. T. Mahmood, M. H. Arif khan and I. H. Lee, "Directional magnitude local hexadecimal patterns: A novel texture feature descriptor for content-based image retrieval," *IEEE access*, vol. 9, pp. 135608-135629, 2021.
- [24] N. Vasconcelos and A. Lippman, "A probabilistic architecture for content-based image retrieval," in *IEEE Conference on computer vision and pattern recognition CVPR*, 2000.
- [25] "Salzburg texture image dataset (STex)," Salzburg University, Austria, [Online]. Available: <https://wavelab.at/sources/STex/>.
- [26] C. Li, Y. Huang and L. Zhu, "Color texture image retrieval based on gaussian copula models of gabor wavelets," *Pattern recognition*, vol. 64, p. 118–129, 2017.
- [27] S. Etemad and M. Amirmazlaghani, "Color texture image retrieval based on copula multivariate modeling in the shearlet domain," *Engineering applications on artificial intelligence*, vol. 102, p. 104256, 2021.
- [28] G. Yang, Y. He, X. Li, H. Liu and T. Lan, "Gabor-GLCM-based texture feature extraction using flame image to predict the O2 content and NOx," *ACS omega*, vol. 7, no. 5, p. 3889–3899, 2022.
- [29] C. Bai, J. Zhang, Z. Liu and Z. Wan-Lei, "K-means based histogram using multiresolution feature vectors for color texture dataset



- retrieval," *Springer science+business media*, vol. 74, p. 1469–1488, 2015.
- [30] "Amsterdam library of textures (ALOT)," 2009. [Online]. Available: [https://aloi.science.uva.nl/public\\_alot/](https://aloi.science.uva.nl/public_alot/).
- [31] K. Sugamya, S. Pabboju and A. Babu, "A CBIR classification using support vector machines," in *IEEE International conference on advances in human machine interaction (HMI)*, 2016.
- [32] G. Verdoolaege, S. De Backer and P. Scheunders, "Multiscale colour texture retrieval using the geodesic distance between multivariate generalized Gaussian models," in *15th IEEE International conference on image processing*, San Diego, 2008.
- [33] S. K. Choy and C. S. Tong, "Statistical wavelet subband characterization based on generalized Gamma density and Its application in texture retrieval," *IEEE Trans. on image processing*, vol. 19, no. 2, pp. 281-289, 2010.
- [34] J. C. Kavitha and A. Suruliandi, "Texture and color feature extraction for classification of melanoma using SVM," in *2016 International conference on computing technologies and intelligent data engineering*, Kovilpatti, 2016.
- [35] S. Gupta, P. Roy, D. Dogra and B.-G. Kim, "Retrieval of colour and texture images using local directional peak valley binary pattern," *Pattern anal applic*, p. 1569–1585, 2020.
- [36] "Brodatz album," A photographic album for artists & designers, Dover, New York, NY, USA, 1966. [Online]. Available: <https://www.ux.uis.no/~tranden/brodatz.html>.
- [37] G. Kylberg, "The Kylberg texture dataset v. 1.0, centre for image analysis," [Online]. Available: <http://www.cb.uu.se/~gustaf/texture/>.
- [38] "AT&T dataset of faces," [Online]. Available: <https://www.kaggle.com/datasets/kasikrit/att-dataset-of-faces>.

- [39] M. Vidya and M. V. Karki, "Skin cancer detection using machine learning techniques," in *IEEE International conference on electronics, computing and communication technologies CONECCT*, Bangalore, 2020.
- [40] "ISIC," The international skin imaging collaboration, [Online]. Available: <https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery>.
- [41] N. Ye, *Data Mining, theories, algorithms, and examples*, Boca Raton: CRC Press, 2013.
- [42] S. Rogers and M. Girolami, *First course in machine learning*, second edition, CRC Press., 2016.
- [43] R. E. Neapolitan and X. Jiang , *Artificial intelligence: with an introduction to machine learning*, second edition, CRC Press, 2018.
- [44] A. Géron, *Hands-on machine learning with scikit-learn, keras, and tensor flow*, 2nd edition, O'Reilly media, Inc., 2019.
- [45] Y. Ma and G. Guo, *Support vector machines applications*, Springer, 2014.
- [46] C. C. Rung and H. H. Chung, "Web page classification based on a support vector machine using a weighted vote schema," *Elsevier*, vol. 31, no. 2, pp. 427-435, 2006.
- [47] A. Navlani, "KNN classification using scikit-learn," *Data camp*, 2 August 2018. [Online]. Available: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>.
- [48] "Wikipedia," 13 may 2019. [Online]. Available: [https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall).
- [49] S. Singh and E. Rajput, "Content based image retrieval using SVM, NN and KNN classification," *International journal of advanced research in computer and communication engineering*, vol. 4, no. 6, pp. 549-552, 2015.

- [50] M. Srinivas and K. M. Chalavadi, "Classification of medical images using edge-based features and sparse representation," in *IEEE International conference on acoustics, speech and signal processing ICASSP*, 2016.
- [51] S. Savita, S. Jain and K. K. Paliwal, "A review on CBIR by cascading features & SVM," *International journal of advanced research in computer science*, vol. 8, no. 3, pp. 42-45, 2017.
- [52] "Wang image dataset," [Online]. Available: <http://wang.ist.psu.edu/docs/related/>.
- [53] M. Kaur and S. Dhingra, "Comparative analysis of image classification techniques using statistical features in CBIR systems," in *International conference on IoT in social, mobile, analytics and cloud I-SMAC*, Palladam, 2017.
- [54] A. Ali and S. Sharma, "Content based image retrieval using feature extraction with machine learning," in *International conference on intelligent computing and control systems*, Madurai, 2017.
- [55] L. Toroitich, W. Cheruiyot and K. Ogada, "K-nearest neighbour in image retrieval based on color and texture," *International journal of innovative science, engineering & technology IJASET*, vol. 5, no. 8, 2018.
- [56] F. A. Alqasemi, H. Q. Alabbasi, F. G. Sabeha, A. Alawadhi, S. Khalid and A. Zahary, "Feature selection approach using KNN supervised learning for content-based image retrieval," in *International conference of intelligent computing and engineering ICOICE*, 2019.
- [57] K. Mashhale, H. B. Nandpuru, V. Kapur and L. Kosta, "MRI brain cancer classification using hybrid classifier (SVM-KNN)," in *International conference on industrial instrumentation and control ICIC*, Pune, 2015.
- [58] N. Guo, R. F. Yen, G. El Fakhri and Q. Li, "SVM Based lung cancer diagnosis using multiple features in PET/CT," in *IEEE Nuclear science*

- symposium and medical imaging conference NSS/MIC*, San Diego, 2015.
- [59] M. A. Rahman, M. T. Haque, C. Shahnaz, S. A. Fattah, W. P. Zhu and M. O. Ahmed, "Skin lesions classification based on color plane-histogram-image quality analysis features extracted from digital images," in *IEEE 60th International midwest symposium on circuits and systems MWSCAS*, Boston, 2017.
- [60] M. M. Ranjitha, N. L. Taranath, C. N. Arpitha and C. K. Subbaraya, "Bone cancer detection using K-means segmentation and Knn," in *International conference on advances in information technology ICAIT*, 2019.
- [61] T. S. Chy and M. A. Rahaman, "A comparative analysis by KNN, SVM & ELM classification to detect sickle cell anemia," in *International conference on robotics, electrical and Signal processing techniques (ICREST)*, Dhaka, 2019.
- [62] P. Xi, C. Shu and R. Goubran, "Abnormality detection in mammography using deep convolutional neural networks," in *IEEE International symposium on medical measurements and applications MeMeA*, Rome, Italy, 2018.
- [63] P. Xi, G. Rouhafzay, H. Guan, C. Shu, L. Borgeat and R. Goubran, "Computer aided detection of abnormality in mammography using deep object detectors," in *State of the art in neural networks and their applications*, Elsevier, 2021, pp. 1-18.
- [64] H. Khachnaoui , R. Guetari and N. Khelifa, "A review on deep Learning in thyroid ultrasound computer-assisted diagnosis systems," in *IEEE International conference on image processing, applications and systems IPAS*, Sophia antipolis, France, 2018.
- [65] O. Ronneberger, P. Fischer and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Medical image computing and computer-assisted intervention MICCAI*, vol. 9351, p. 234–241, 2015.

- [66] "IEEE-ISBI international symposium on biomedical imaging," 2015. [Online]. Available: <https://biomedicalimaging.org/2015/program/isbi-challenges/>.
- [67] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on multimedia*, Orlando, 2014.
- [68] A. Rouhafzay and N. Baaziz, "Improving texture image retrieval with redundant contourlet features," in *International conference on computational science and computational intelligence, symposium on signal & image processing CSCI-ISPC*, USA, 2019.
- [69] M. S. Allili, "Wavelet modeling using finite mixtures of generalized Gaussian distributions: application to texture discrimination and retrieval," *IEEE Trans. on image processing*, vol. 21, no. 4, pp. 1452 - 1464, 2012.
- [70] M. Diop, "Transformées redondantes et recherche d'images par la texture dans les bases de données," UQO, Gatineau, Canada, 2016.
- [71] H. Permuter, J. Francos and I. Jermyn, "A study of gaussian mixture models of color and texture features for image classification and segmentation," *Elsevier Ltd*, vol. 39, no. 4, p. 695 – 706, 2006.
- [72] J. R. Hershey and P. A. Olsen, "Approximating the kullback-leibler divergence," in *IEEE International conference on acoustics, speech and signal processing ICASSP '07*, Honolulu, 2007.
- [73] L. Gong, T. Wang, Y. Yu, F. Liu and X. Hu, "A lie group based gaussian mixture model distance measure for multimedia comparison," in *ICIMCS '09 Proceedings of the First international conference on internet multimedia computing and service*, 2009.
- [74] S. Cui and M. Datcu, "Comparison of kullback-Leibler divergence approximation methods between gaussian mixture models for satellite image retrieval," in *IEEE International geoscience and remote sensing symposium IGARSS*, Milan, 2015.

- [75] J. L. Durrieu, J. P. Thiran and F. Kelly, "Lower and upper bounds for approximation of the Kullback-Leibler divergence between gaussian mixture models," in *IEEE International conference on acoustics, speech and signal processing ICASSP*, Kyoto, 2012.
- [76] V. K. a. A. B. Rother. C, "GrabCut - Interactive foreground extraction using iterated graph cuts," *ACM Trans. on graphics SIGGRAPH*, vol. 23, no. 3, p. 309–314, 2004.
- [77] Matlab, "Image processing toolbox," Mathworks, 2022. [Online].
- [78] Y. a. J. M. Boykov, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *In international conference on computer vision*, 2001.
- [79] A. Rosebrock, "OpenCV GrabCut: foreground segmentation and extraction," Pyimagesearch, 2020.
- [80] R. Kwitt, A. Uhl and P. Meerwald-Stadler, "Efficient texture image retrieval using copulas in a bayesian framework," *IEEE Trans. on image processing*, vol. 20, no. 7, pp. 2063 - 2077, 2011.
- [81] R. Kwitt and A. Uhl, "Image similarity measurement by Kullback-Leibler divergences between complex wavelet subband statistics for texture retrieval," in *15th IEEE international conference on image processing*, 2008.
- [82] M. Benco, P. Kamencay, M. Radilova, R. Hudec and M. Sinko, "The comparison of color texture features extraction based on 1D GLCM with deep learning methods," in *IWSSIP*, Rio de Janeiro, 2020.
- [83] S. Bhardwaj, G. Pandove and P. K. Dahiya, "An analysis of two novel and efficient deep learning models for fast and accurate image retrieval," *Jornal of information science and engineering*, vol. 37, pp. 185-201, 2021.
- [84] A. Rouhafzay, N. Baaziz and M. s. Allili, "Texture image retrieval using a classification and contourlet-based features," in *IPCV'21 - The*

*25th Int'l conf on image processing, computer vision, & pattern recognition, Las vegas, 2021.*

- [85] J. A. Camacho-Gutiérrez, S. Solorza-Calderón and J. Álvarez-Borrogo, "Multi-class skin lesion classification using prism- and segmentation-based fractal signatures," *Expert systems with applications*, vol. 197, p. 116671, 2022.
- [86] N. Hameed, A. M. Shabut, M. K. Ghosh and M. A. Hossain, "Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques," *Expert systems with applications*, vol. 141, p. 112961, 2020.
- [87] F. Alenezi, A. Armghan and K. Polat, "Wavelet transform based deep residual neural network and ReLU based extreme learning machine for skin lesion classification," *Expert systems with applications*, vol. 213, p. 119064, 2022.
- [88] "Describable textures dataset (DTD)," 2012. [Online]. Available: <http://www.robots.ox.ac.uk/~vgg/data/dtd/>.
- [89] "Abin24/Textures-dataset," GitHub, Inc, [Online]. Available: <https://github.com/abin24/Textures-Dataset>.
- [90] M. Farooq, H. Zheng, A. Nagabhushana, S. Roy, S. Burkett, M. Barkey, S. Kotru and E. Sazonov, "Damage detection and identification in smart structures using SVM and ANN," in *SPIE Smart structures and materials + nondestructive evaluation and health monitoring*, San Diego, 2012.
- [91] T. Maddess, D. M. Coy, J. Herrington and C. F. Carle, "Learning complex texture discrimination," *Journal of the optical society of america A*, vol. 38, no. 3, pp. 449-455, 2021.
- [92] M.-T. Pham, G. Mercier and L. Bombrun, "Color texture image retrieval based on local extrema features and riemannian distance," *Journal of imaging MDPI*, vol. 43, no. 3, 2017.

- [93] H. F. Atlam, G. Attiya and N. El-fishawy, "Integration of color and texture features in CBIR system," *International journal of computer applications*, vol. 164, no. 3, pp. 0975-8887, 2017.
- [94] S. Sigurdsson, P. A. Philipsen, L. K. Hansen, J. Larsen, M. Gniadecka and H. C. Wulf, "Detection of skin cancer by classification of raman spectra," *IEEE Trans. on biomedical engineering*, vol. 10, no. 51, pp. 1784-1793, 2004.
- [95] G. Rouhafzay, A.-M. Cretu and P. Payeur, "Transfer learning from vision to touch: A hybrid deep convolutional neural network for visuo-tactile 3D object recognition," *MDPI*, 2020.
- [96] J. R. Pomerantz, "Visual discrimination of texture," *Perception and psychophysics*, vol. 24, no. 5, pp. 420-428, 1978.
- [97] M. Zand, S. Doraisamy, A. Abdul Halin and M. R. Mustaffa, "Texture classification and discrimination for region-based image retrieval," *Journal of visual communication and image representation*, vol. 26, pp. 305-316, 2015.
- [98] D. W. Anggara, M. S. M. Rahim, A. W. Ismail, R. Machfiroh, A. Budiman, A. Rahmansyah, Dahliyusmanto and N. A. Atan, "Grayscale image enhancement for enhancing features detection in marker-less augmented reality technology," *Journal of theoretical and applied information technology*, vol. 98, no. 13, pp. 2671-2683, 2020.
- [99] S. Murala, R. P. Maheshwari and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Trans. on image processing*, vol. 21, no. 5, pp. 2874-2886, 2012.
- [100] Y. Liu, D. Zhang, G. Lu and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *The journal of the pattern recognition society*, vol. 40, pp. 262-282, 2007.
- [101] G. Verdoolaege and P. Scheunders, "Geodesics on the manifold of multivariate generalized gaussian distributions with an application to multicomponent texture discrimination," *International journal of computer vision*, vol. 95, p. 265-286, 2011.



- [102] S. Sharma, "Use of artificial intelligence algorithm for content-based image retrieval system," *International journal of advance research ideas and innovations in technology*, vol. 4, pp. 680-684, 2018.
- [103] A. Murugan, S. Anu, H. Nair and K. P. Sanal Kumar, "Detection of skin cancer using SVM random forest and KNN classifiers," *Journal of medical systems springer*, 2019.
- [104] R. S. Mahagaonkar and S. Soma, "A novel texture based skin melanoma detection using color GLCM and CS-LBP feature," *International journal of computer applications*, vol. 171, no. 5, 2017.
- [105] P. H. D. Ojala T, "A comparative study of texture measures with classification based on featured distribution," *Pattern recognition*, vol. 29, no. 1, pp. 51-59, 1996.
- [106] P. D. Wankhade, "A review on aspects of texture analysis of images," *International journal of application or innovation in engineering and management IJAIEM*, vol. 3, no. 10, pp. 229-233, 2014.
- [107] M. Kokare, B. Chatterji and P. Biswas, "A survey on current content based image retrieval methods," *IETE journal of research*, vol. 48, pp. 261-271, 2002.
- [108] N. Fathi, B. Vaseghi and M. Shemshadi, "An overview of content-based image retrieval techniques CBIR," *IOSR journal of computer engineering*, vol. 18, no. 4, pp. 108-116, 2016.
- [109] H. Younis, M. H. Bhatti and M. Azeem, "Classification of skin cancer dermoscopy images using transfer learning," in *International conference on emerging technologies ICET*, Peshawar, 2019.
- [110] M. K. Alsmadi1, "Content-based image retrieval using color, shape and texture," *Arabian journal for science and engineering*, vol. 10, no. 1007, p. 3317–3330, 2020.
- [111] G. J. Tian, Y. Xia, Y. Zhang and D. Feng, "Hybrid genetic and variational expectation-maximization algorithm for Gaussian-mixture-model-based brain MR image segmentation," *IEEE Trans. on*

- information technology in biomedicine*, vol. 15, no. 3, pp. 373-380, 2011.
- [112] P. Pawar and P. P. Belagali, "Image retrieval technique using local binary pattern (LBP)," *International journal of science and research IJSR*, vol. 4, no. 7, pp. 1440-1444, 2015.
- [113] H. Greenspan and A. T. Pinhas, "Medical image categorization and retrieval for RACS using the GMM-KL framework," *IEEE Trans. on information technology in biomedicine*, vol. 11, no. 2, pp. 190-202, 2007.
- [114] X. Cong-Hua, C. Jin-Yi and X. Wen-Bin, "Medical image denoising by generalized Gaussian mixture modeling with edge information," *IET Image processing*, vol. 8, no. 8, pp. 464-476, 2014.
- [115] D. Yapi and M. S. Allili, "Multi-band texture modeling using mixture of multivariate generalized Gaussians distributions," in *26th International conference on pattern recognition ICPR*, Montreal, 2022.
- [116] J. Z. Wang, J. Li and G. Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picture libraries," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 23, no. 9, pp. 947-963, 2001.
- [117] G. Raghuwanshi and V. Tyagi, "Texture image retrieval using hybrid directional extrema pattern," *Springer nature*, vol. 80, pp. 2295-2317, 2020.

## Annex I

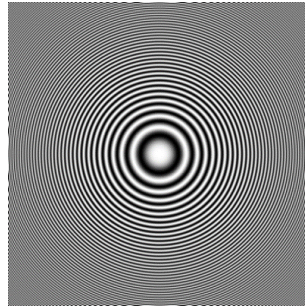


Figure 0.1 Original Zone Plate Image.

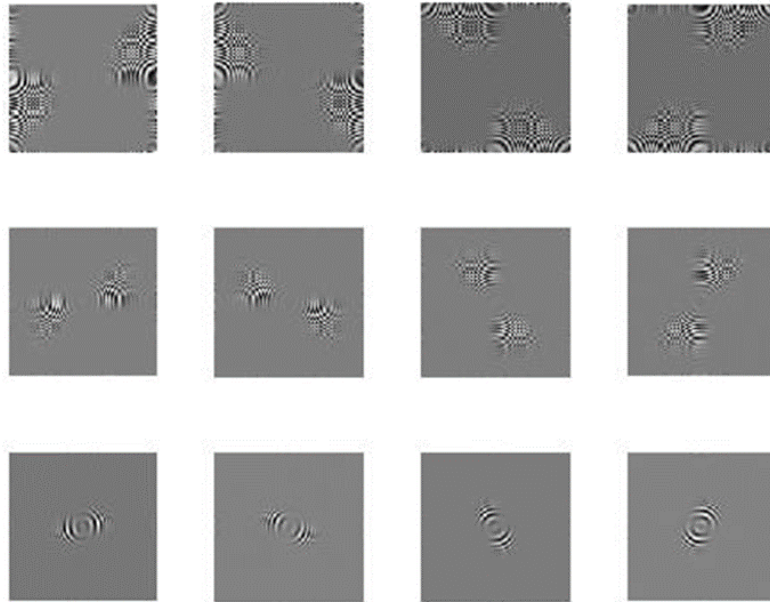


Figure 0.2 Redundant contourlet sub-bands of zone plate image for 3 scale levels and 4 directions.

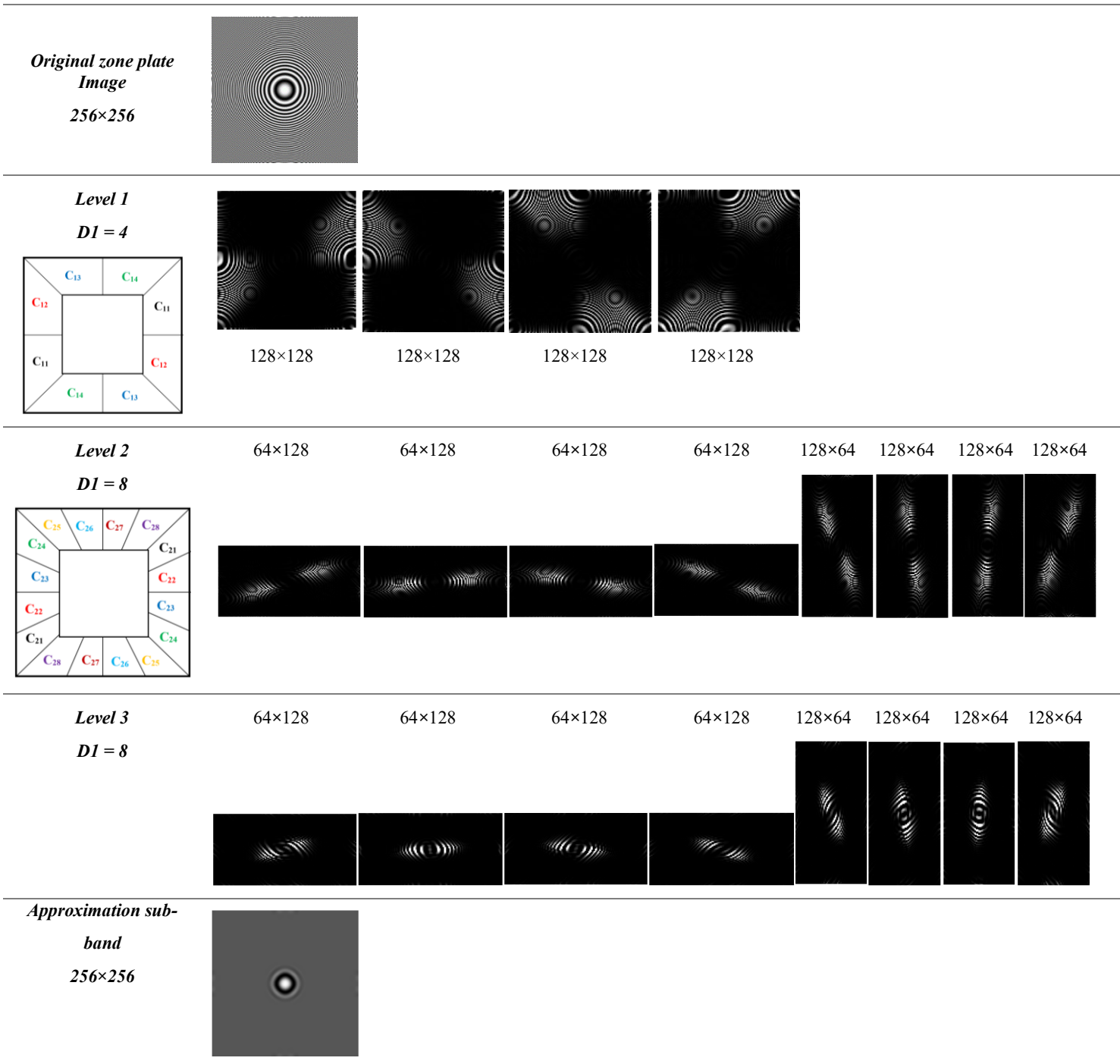


Figure 0.3 RCT-Plus decomposition of a 256×256 zone plate image. The number of scale levels is  $L=3$ . The number of directional sub-bands at each scale level  $l$  is indicated by  $D_l$ , thus  $[D_1, D_2, D_3] = [4, 8, 8]$ . The redundancy factor is 4.

## Annex II

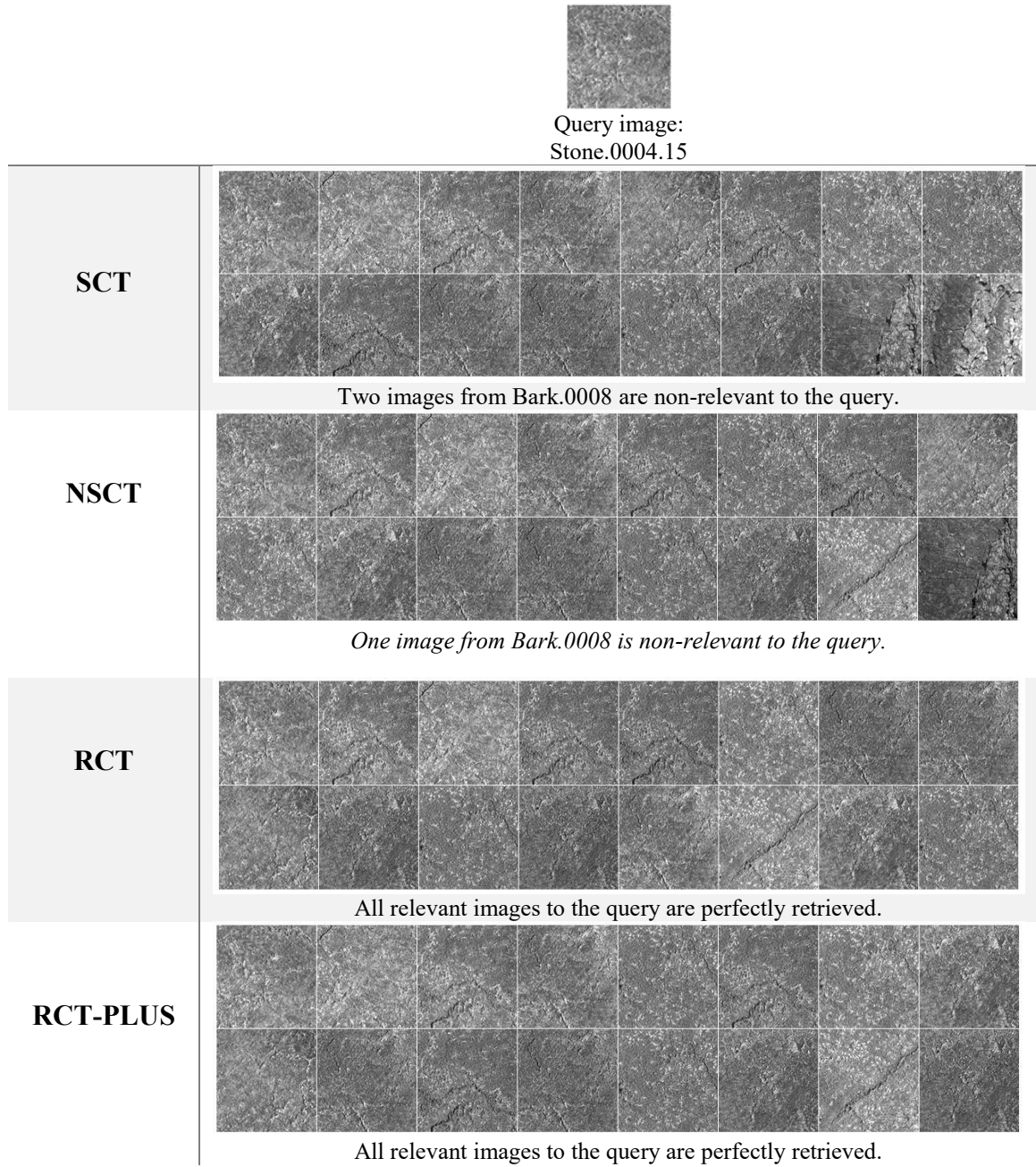


Figure 0.4 Examples of Top16 retrieval results from the VisTex-40 dataset (640 images). The query image is Stone.0004.15.

## Annex III

### Complementary facts about SVM

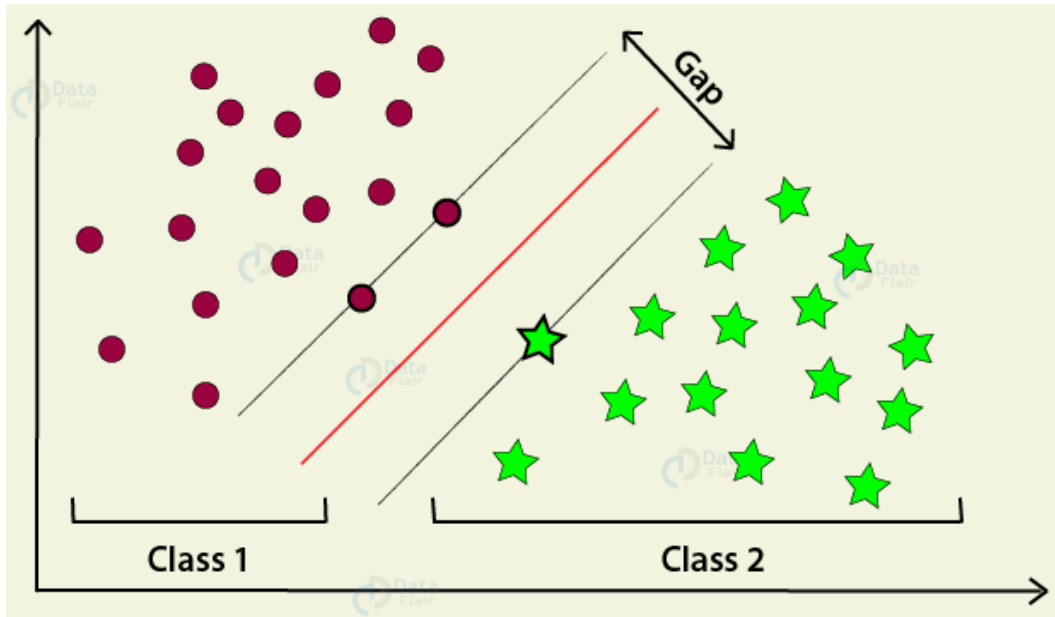


Figure 0.5 SVM

- In ML, the kernel trick, is a method of using a linear classifier to solve a non-linear problem. It transforms linearly inseparable data to linearly separable data. The kernel function is applying on each data to map the original non-linear observations into a higher-dimensional space and make it separable.
- Support vectors are the coordinates of observation, the one green star and two red Circles on maximum margins.
- The kernel SVM happens when the data is not linearly separable, then we need to use a linear or non-linear way to separate the data.
- In linear kernels we map our data to a space which we can separate the classes linearly, but in non-linear kernels we map our data to a space which we need non-linear functions to separate our classes.
- For example, polynomial RBF radial basis function (Gaussians) and sigmoid (neural net activation function) are non-linear kernels.
- Quadratic and Cubic kernels are the special cases of polynomial kernel.

- A simple and natural definition about kernel is a similarity function. Given two elements, the kernel outputs similarity score. The elements can be anything starting from two integers, two real valued vectors everything that the kernel function knows how to compare them.
- Kernel functions are also referred to as Gram matrices. Since the Gram matrix over the elements is a symmetric matrix, it is diagonalizable and its quantities are non-negative. As we created a new kernel function with KLD divergence and convert it to a symmetric gram matrix.
- The penalty parameter of the error which is known as (C) parameter has a role of controlling the trade-off between the decision boundaries to classify the training data properly. In other words, it states that how much the algorithm care about misclassified points. When increasing the value of C, the algorithm sets the margins to classify all the training points correctly, so the classifier becomes more prone to over fitting and cannot generalize on future data.

## Annex IV

<i>TopN</i>	Number of scale levels <i>L</i>	E1	E2	GGD1	GGD2	GGD2-ED
16	1	62.01	62.47	75.30	72.91	69.18
	2	66.81	68.37	<b>76.91</b>	75.41	72.09
	3	66.90	68.84	75.83	75.23	69.55
20	1	66.42	67.14	79.90	76.52	73.30
	2	71.41	72.85	<b>81.55</b>	79.43	75.68
	3	71.61	73.34	80.77	79.79	73.32
40	1	77.78	79.02	87.47	84.27	82.60
	2	81.61	83.03	88.48	86.74	83.98
	3	81.78	82.79	<b>88.55</b>	87.65	82.51
60	1	83.54	84.73	91.04	87.75	87.71
	2	86.84	87.99	<b>92.40</b>	90.86	88.17
	3	86.87	87.42	91.81	91.16	86.36
80	1	87.23	88.45	92.84	89.88	90.52
	2	90.65	91.50	<b>94.45</b>	92.81	91.22
	3	90.19	90.70	94.00	93.35	89.10
100	1	89.67	90.74	94.42	91.63	92.87
	2	92.65	93.35	<b>95.71</b>	94.15	93.14
	3	92.20	92.46	95.40	94.72	91.16

Table 0.1 Average retrieval rates (*AR%*) according to the number of top matches considered, *TopN*. The compared CBIR methods (E1, E2, GGD1, GGD2 and GGD2-ED) operate on NSCT decompositions with  $D=8$  and  $L$  ranging from 1 to 3.



Traditional CBIR	Color Space	VisTex-40	Kylberg-27	Kylberg-28	Stex
E2	Grayscale	71.72	63.77	55.51	44.55
	RGB	77.37	-	-	51.86
	YCBCR	80.42	-	-	66.26
GGD2	Grayscale	75.24	65.79	59.53	51.14
	RGB	81.58	-	-	58.14
	YCBCR	84.36	-	-	69.20
GGD2-ED	Grayscale	71.78	66.60	59.53	45.28
	RGB	7.12	-	-	51.53
	YCBCR	78.37	-	-	63.77
GGD2+GMapp	Grayscale	82.53	69.09	63.25	53.14
	RGB	84.91	-	-	63.28
	YCBCR	89.53	-	-	76.96
GGD1	Grayscale	77.15	64.95	58.48	51.89
	RGB	83.43	-	-	59.29
	YCBCR	85.93	-	-	70.67
GGD1-ED	Grayscale	74.98	66.06	58.48	44.18
	RGB	77.17	-	-	50.19
	YCBCR	78.40	-	-	63.05
GGD1+GMapp	Grayscale	82.77	<b>70.04</b>	<b>63.89</b>	53.64
	RGB	85.15	-	-	63.72
	YCBCR	89.83	-	-	77.20
GGD1+GM	Gray+ RGB	91.77	-	-	76.85
	Gray+ YCBCR	<b>92.26</b>	-	-	<b>77.63</b>
LBP-ED	Grayscale	75.74	61.24	53.46	50.70
	RGB	85.10	-	-	57.00
	YCBCR	85.44	-	-	66.92

Table 0.2 Average retrieval rates ( $AR\%$ ) according to the number of top matches considered. The compared CBIR schemes operate on RCT-Plus decomposition with  $D= [8, 8, 8]$  and  $L=3$ , and GMM modeling is performed with  $k=2$ .

ML-CBIR (KNN)	Color Space	VisTex-40	Kylberg-27	Kylberg-28	Stex
E2	Grayscale	99.22	93.43	92.37	86.06
	RGB	99.37	-	-	90.77
	YCbCr	99.84	-	-	96.17
GGD1	Grayscale	99.69	97.69	97.32	88.29
	RGB	99.69	-	-	91.78
	YCbCr	99.84	-	-	95.52
GGD2	Grayscale	99.53	97.31	96.90	87.59
	RGB	99.69	-	-	90.93
	YCbCr	99.84	-	-	95.13
GGD1+GMapp	Grayscale	99.69	<b>98.61</b>	<b>98.50</b>	90.61
	RGB	99.84	-	-	93.97
	YCbCr	100.00	-	-	96.09
GGD1+GM	RGB	100.00	-	-	94.24
	YCbCr	100.00	-	-	<b>97.10</b>
LBP-ED	Grayscale	98.59	93.48	91.67	85.77
	RGB	99.22	-	-	90.03
	YCbCr	99.37	-	-	93.95

Table 0.3 ML-CBIR using fine (K=1) KNN.

ML-CBIR (SVM)	Color Space	VisTex-40	Kylberg-27	Kylberg-28	Stex
E2	Grayscale	99.69	93.43	96.38	87.13
	RGB	99.69	-	-	89.89
	YCbCr	99.84	-	-	93.29
GGD1	Grayscale	99.69	97.96	98.75	87.64
	RGB	99.84	-	-	90.05
	YCbCr	100.00	-	-	93.21
GGD2	Grayscale	88.69	97.59	98.66	87.24
	RGB	99.84	-	-	89.67
	YCbCr	99.84	-	-	93.04
GGD1+GMapp	Grayscale	99.69	<b>98.70</b>	<b>99.04</b>	89.92
	RGB	100.00	-	-	93.49
	YCbCr	100.00	-	-	95.65
GGD1+GM	RGB	100.00	-	-	94.10
	YCbCr	100.00	-	-	<b>96.99</b>
LBP-ED	Grayscale	86.06	93.43	96.29	86.66
	RGB	95.50	-	-	88.31
	YCbCr	97.32	-	-	93.11

Table 0.4 ML-CBIR using SVM. In GGD1, GGD2, E2 and LBP linear kernel and in GGD1+GMapp and GGD1+GM the KLD kernel are applied.

## Annex V

In proposal, we used grayscale component of images and 4-pixel overlapping subdivisions into blocks. A set of experiments were conducted on the ISIC-32 dataset. On the one hand, feature extraction based on RCT-Plus and GGD modeling was applied to each sub-image yielding a set of feature vectors. As we experienced and examined many retrieval schemes using RCT-Plus with various combinations of scale levels and directional orientations, we chose 3 scale levels and 8 directions at each scale level for the subsequent analysis. Thus, one image decomposition results into 24 contourlet sub-bands. Applying GGD modeling and estimating  $\alpha$  and  $\beta$  parameters on each sub-band yield a feature vector with 48 components.

On the other hand, a class label was assigned to each sub-image of the ISIC-32 dataset according to the following rule: if the ratio of lesion pixels in the corresponding ground truth segmentation map was greater than a pre-defined threshold  $T\%$ , the class label was set to 2 (unhealthy skin) otherwise it was set to the value 1 (healthy skin). The choice of the threshold value  $T\%$  was made empirically. Typically, test values range from 0% to 90% using a step of 10.

Consequently, for each new threshold  $T\%$ , a new ground truth labeling set was defined and assigned to the feature vectors to form a new feature dataset (*Dataset- $T\%$* ).

For the learning process of KNN and SVM algorithms (offline phase), the selected feature dataset (*Dataset- $T\%$* ) was split into train data (3024 feature vectors and labels) and test data (432 feature vectors and labels) corresponding to 28 and 4 original ISIC images respectively. Both KNN and SVM algorithms were considered with either Euclidean (ED) or KLD distances. Moreover, the cross-validation approach was applied during the learning phase with a 10-fold data split for both SVM and KNN.

The online detection phase is meant to detect skin abnormality in each sub-image of the test data. For this purpose, the trained model classifier was used to predict and assign a class label 1 (healthy skin) or 2 (unhealthy skin) for each input feature vector from the test data.

In order to evaluate the effectiveness of the abnormality detection, we firstly compare the predicted labels to the ground truth ones and calculate the

*Precision, Recall, and detection Accuracy* metrics under the following varying parameters: the factor K (in KNN algorithm), the kernel type in SVM, the type of distance metric in either KNN or SVM and the selected dataset *Dataset-T%*.

Table 0.5 displays a sample of obtained results for KNN classifier. In Table 0.6 more detection results are reported by considering the SVM classifier with linear kernel and Euclidean distance ED. The whole detection map was constructed by mosaicking a set of blocks (sub-images) tagged by the predicted class labels. Superimposing the constructed detection map on the original test image shows the segmentation results and determines the abnormal skin image regions. Figure 0.6 presents examples of mapped results on original test images using KNN with image features *Dataset-0%*, *Dataset-30%*, and *Dataset-40%*. It appears the results from *Dataset-30%* show higher recall in KNN.

<b><i>Dataset-T%</i></b>	<b><i>Distance</i></b>	<b><i>K</i></b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>Accuracy</i></b>
0%	ED	7	90.10	93.70	93.30
	KLD		87.40	92.20	92.30
10%	ED	7	83.30	94.80	92.00
	KLD		87.57	97.37	94.21
20%	ED	7	76.54	95.80	89.00
	KLD		81.44	97.64	92.00
30%	ED	7	78.40	93.70	89.35
	KLD		79.60	98.43	91.20
40%	ED	7	74.00	91.34	88,00
	KLD		74.25	97.64	89.35
50%	ED	7	70.34	90.00	87.04
	KLD		71.34	91.80	87.30
60%	ED	7	62.83	87.40	85.88
	KLD		68.00	95.28	87.04
70%	ED	7	55.20	92.40	80.00
	KLD		63.53	96.23	85.64
80%	ED	7	55.32	82.70	83.33
	KLD		56.40	92.13	82.64
90%	ED	7	55.71	92.22	83.10
	KLD		55.00	93.33	82.70

Table 0.5 A sample of abnormality detection results, *Precision*, *Recall* and *Accuracy*, corresponding to various selected feature datasets (*Dataset-T %*) from ISIC-32. The KNN classification algorithm is based on either KLD or ED distance. The factor K is equal to 7.

<b><i>Dataset-T%</i></b>	<b><i>Precision</i></b>	<b><i>Recall</i></b>	<b><i>Accuracy</i></b>
0%	87.10	89.10	90.30
10%	82.74	91.45	90.30
20%	77.11	89.51	87.73
30%	74.23	88.32	86.57
40%	71,00	86.61	85.65

Table 0.6 A sample of abnormality detection results, *Precision*, *Recall* and *Accuracy*, corresponding to various selected feature datasets (*Dataset-T %*) from ISIC-32. The considered classification algorithm is SVM with linear kernel and ED distance.

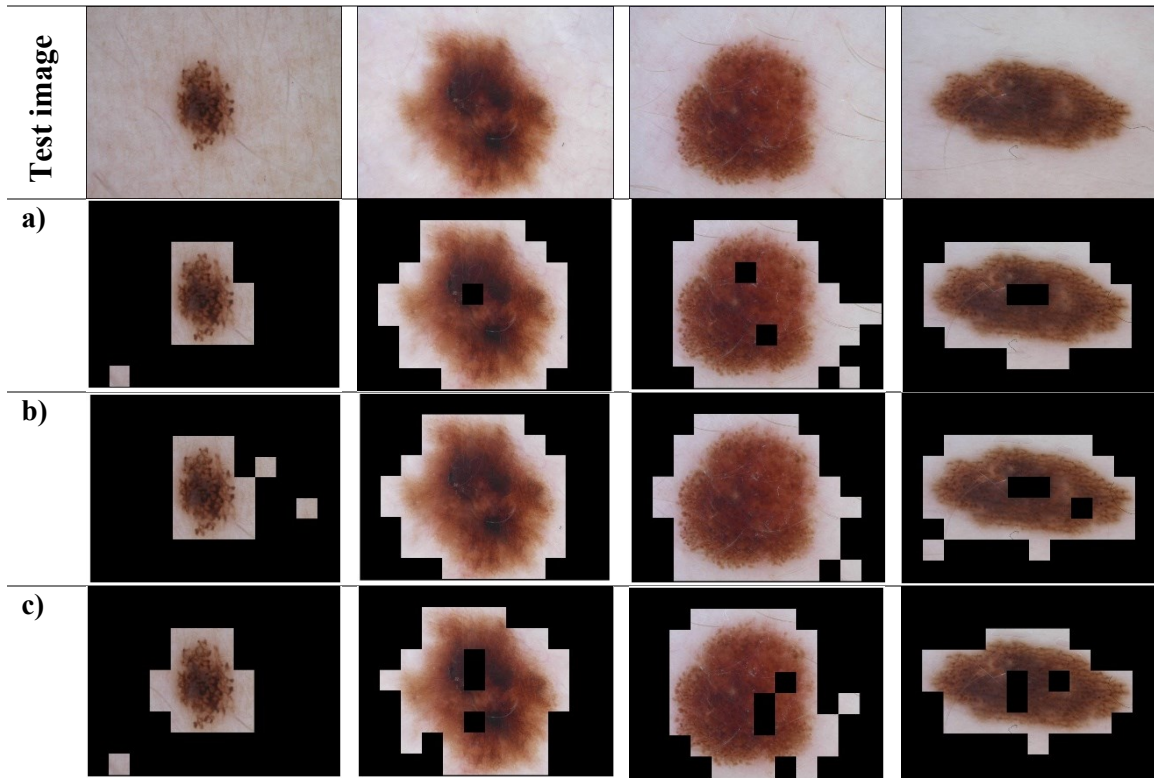


Figure 0.6 Examples of primary abnormal detection results mapped onto original test images from ISIC-32 dataset. The considered classifier is KNN with  $K=7$  and KLD distance, operating on the feature datasets a) *Dataset-0%*, b) *Dataset-30%*, and c) *Dataset-40%*.