



รายงานการวิจัยฉบับสมบูรณ์

ระบบจัดเก็บและสื่อสารภาพทางการแพทย์และ
ระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ
Picture Archiving and Communication System and
Content-Based Medical Image Retrieval System

นายเทอดศักดิ์ ลีวหาทอง

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณแผ่นดิน ประจำปีงบประมาณ พ.ศ. 2557

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



รายงานการวิจัยฉบับสมบูรณ์

ระบบจัดเก็บและสื่อสารภาพทางการแพทย์และ
ระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ
Picture Archiving and Communication System and
Content-Based Medical Image Retrieval System

นายเทอดศักดิ์ ลีวหาทอง

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณแผ่นดิน ประจำปีงบประมาณ พ.ศ. 2557

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อโครงการ (ภาษาไทย) ระบบจัดเก็บและสื่อสารภาพทางการแพทย์และระบบค้นหาภาพทาง
การแพทย์ด้วยรายละเอียดของภาพ

แหล่งเงิน งบประมาณแผ่นดิน ประจำปีงบประมาณ พ.ศ. 2557 ตามมติคณะรัฐมนตรี

ประจำปีงบประมาณ 2557 จำนวนเงินที่ได้รับการสนับสนุน 800,000 บาท

ระยะเวลาทำการวิจัย 1 ปี ตั้งแต่ 1 ตุลาคม 2556 ถึง 30 กันยายน 2557

ชื่อ-สกุล หัวหน้าโครงการ และผู้ร่วมโครงการวิจัย พร้อมระบุ หน่วยงานต้นสังกัด

นายเทอดศักดิ์ ลีวาททอง

บทคัดย่อ

ภาพทางการแพทย์เช่นภาพ Computed Tomography (CT), Magnetic Resonance Image (MRI) และ X-Ray เป็นเครื่องมือที่สำคัญในการวินิจฉัยโรคต่างๆ ในปัจจุบันภาพทางการแพทย์เหล่านี้ถูกสร้างขึ้นเป็นจำนวนมากเนื่องจากเครื่องสแกนภาพทางการแพทย์มีจำนวนมากขึ้นและราคาถูกลง นอกจากนี้เครื่องสแกนภาพรุ่นใหม่ ๆ มีการสนับสนุนเทคโนโลยีภาพดิจิทัลและการสื่อสารทางการแพทย์ (Digital Imaging and Communications in Medicine, DICOM) เพื่อให้แพทย์สามารถรักษาผู้ป่วยได้อย่างมีประสิทธิภาพมากยิ่งขึ้น งานวิจัยนี้จึงพัฒนาระบบการจัดเก็บและสื่อสารภาพทางการแพทย์ และการค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ เพื่อลดปัญหาการซ้ำซ้อนและสิ้นเปลืองเนื้อที่ในการจัดเก็บภาพ เพิ่มประสิทธิภาพในการสื่อสารภาพระหว่างเครื่องสแกนภาพชนิดต่างๆภายในโรงพยาบาล และช่วยให้แพทย์สามารถค้นหาภาพที่ต้องการเช่นภาพของผู้ป่วยอื่นๆ ที่เคยได้รับการรักษาและมีอาการของโรคคล้ายกับผู้ป่วยที่กำลังรักษาอยู่ เพื่อเป็นข้อมูลในการตัดสินใจ วินิจฉัยโรคของผู้ป่วย

คำสำคัญ : ระบบจัดเก็บและสื่อสารภาพทางการแพทย์, ภาพดิจิทัลและการสื่อสารภาพทางการแพทย์, ระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ, เวกเตอร์ของลักษณะเฉพาะทาง Texture, Weighted Multidimensional Generalization of Wald-Wolfowitz Runs Test

Research Title: Picture Archiving and Communication System and Content-Based Medical Image Retrieval System

Researcher: Mr. Thurdsak Leauhatong

Faculty: Faculty of Engineering

Department: Department of

Electronics

ABSTRACT

Medical images such as Computed Tomography (CT), Magnetic Resonance Image (MRI), and X-Ray are important tools for diagnosing many diseases. Nowadays, a number of medical images are created because the number of X-ray, CT and MRI scanners have been increasing and become cheaper. Moreover, new scanners support Digital Imaging and Communications in Medicine (DICOM) to let physicians treat patients more effectively. This research developed a Picture Archiving and Communication System and Content-Based Medical Image Retrieval System in order to reduce the problems of redundancy and storage of saving pictures, increase the efficiency of communicating between scanners in hospital and help physicians to retrieve the query image such as pictures of other patients who have been treated and had similar symptom to the new patient in order to decide, diagnosis and select the best method to treat the patient.

Keywords : Picture Archiving and Communication System, Digital Imaging and Communications in Medicine, Content-Based Medical Image Retrieval, Feature Vector of Texture, Weighted Multidimensional Generalization of Wald-Wolfowitz Runs Test

กิตติกรรมประกาศ

งานวิจัยฉบับนี้สำเร็จลุล่วงไปด้วยดีเนื่องจากผู้วิจัยได้รับความช่วยเหลือ คุณเลเอาใจใส่จากหลายๆฝ่าย โดยเฉพาะอย่างยิ่ง ศาสตราจารย์ ดร. Shozo Kondo และรองศาสตราจารย์ ดร. มนัส สังวรศิลป์ ในความช่วยเหลือและให้คำแนะนำเกี่ยวกับงานทางด้านการประมวลผลภาพ ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาของอาจารย์ทั้งสองท่านนี้เป็นอย่างยิ่ง และขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้ขอขอบคณนักศึกษา สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ ที่ได้ให้ความร่วมมือในการดำเนินการทดลอง

นอกจากนี้ผู้วิจัยยังได้รับกำลังใจจากคุณพ่อ คุณแม่ และเพื่อนๆ ตลอดจนบุคคลต่างๆที่ทำให้กำลังใจและความช่วยเหลืออีกมาก ที่ผู้วิจัยไม่สามารถกล่าวนามได้หมดในที่นี้ ผู้วิจัยรู้สึกซาบซึ้งในความกรุณาและความปรารถนาดีของทุกท่านเป็นอย่างยิ่ง จึงกราบขอบพระคุณและขอบคุณไว้ใน โอกาสนี้

การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จากแหล่งทุน งบประมาณแผ่นดิน ตามมติคณะรัฐมนตรี ประจำปีงบประมาณ พ.ศ. 2557

เทอดศักดิ์ ลีวาททอง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญภาพ.....	ญ
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย.....	1
1.2 วัตถุประสงค์ของโครงการวิจัย.....	2
1.3 ขอบเขตของโครงการวิจัย.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ระบบ Picture Archives and Communication System (PACS).....	4
2.1.1 หน้าที่การทำงานของระบบ PACS.....	4
2.1.2 ประโยชน์ที่จะได้รับจากระบบ PACS.....	5
2.1.3 องค์ประกอบของระบบ PACS.....	5
2.1.4 ขั้นตอนการทำงานร่วมกันระหว่างระบบ PACS, HIS, และเครื่องสแกนภาพทาง การแพทย์.....	6
2.1.5 ระบบเน็ตเวิร์คของระบบ PACS.....	7
2.1.6 DICOM Viewer.....	8
2.2 CBMIRS (Content-Based Medical Image Retrieval System).....	11
2.2.1 ปัญหาของระบบค้นหาภาพโดยรายละเอียดของภาพ.....	11
2.2.1.1 การทำความเข้าใจรายละเอียดภาพของมนุษย์.....	11
2.2.1.2 คุณลักษณะและการวัดความเหมือนของภาพ.....	11
2.2.1.3 ความเชื่อมโยงระหว่างคุณลักษณะพื้นฐานของภาพ และแนวความคิดขั้นสูง.....	12
2.2.1.4 ระบบติดต่อผู้ใช้.....	13
2.2.1.5 ตัวชี้ในมิติขั้นสูง.....	13
2.2.1.6 การประเมินประสิทธิภาพและตัวทดสอบมาตรฐาน.....	13

สารบัญ (ต่อ)

	หน้า
2.2.2 คุณลักษณะของภาพ.....	14
2.2.2.1 สี.....	14
2.2.2.2 พื้นผิว.....	16
2.2.2.3 รูปร่าง.....	18
2.2.3 การวัดความเหมือน.....	20
2.2.3.1 Minkowski-Form Distance.....	20
2.2.3.2 Quadratic Form (QF) Distance.....	21
2.2.3.3 Mahalanobis Distance.....	21
2.2.3.4 Kullback-Leibler (KL) Divergence และ Jeffrey-Divergence (JD).....	22
2.2.3.5 χ^2 Statistic.....	22
2.2.3.6 Earth's Mover Distance (EMD).....	22
2.2.4 การประเมินประสิทธิภาพ.....	23
2.3 ระบบการค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ CBMIRS.....	23
(Content-Based Medical Image Retrieval System)	
บทที่ 3 ทฤษฎีพื้นฐาน	25
3.1 ทฤษฎีการแปลง Wavelet.....	25
3.1.1 นิยามของ Subspace และ Scaling Function.....	25
3.1.2 ปริภูมิรายละเอียดและฟังก์ชัน Wavelet.....	26
3.1.3 Orthogonal Base Wavelet.....	27
3.1.4 การแปลง Wavelet แบบเร็ว.....	28
3.2 ทฤษฎีกราฟและต้นไม้แผ่ทั่วที่น้อยที่สุด.....	29
3.3 ทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz.....	32
3.3.1 นิยามของทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz.....	32
3.3.2 การแจกแจงการเรียงสับเปลี่ยนของทดสอบรันแบบหลายมิติของ Wald.....	35
และ Wolfowitz	
บทที่ 4 การวัดความเหมือนโดยใช้ทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz	38
4.1 การคำนวณคุณลักษณะทางพื้นผิวโดยใช้การแปลง Wavelet.....	39
4.2 Vector Quantization และการแบ่งคลัสเตอร์แบบ k-Means.....	40

สารบัญ (ต่อ)

	หน้า
4.2.1 Vector Quantization, Integral ของ Bennett และฟังก์ชันความหนาแน่นของจุด	40
4.2.2 วิธีการแบ่งคลัสเตอร์แบบ k-Means	44
4.3 นิยามของวิธีทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid	46
แบบมีน้ำหนัก	
บทที่ 5 ระบบค้นหารูปภาพโดยใช้รายละเอียดของภาพ	49
5.1 การทำงานของโปรแกรม VqKmeanCalculate	50
5.2 การทำงานของโปรแกรม VqRetrieval	50
5.3 การทำงานของ Web Page ที่ใช้ในการเก็บภาพลงในฐานข้อมูล	51
5.4 การทำงานของ Web Page ที่ใช้ในการค้นหาภาพในฐานข้อมูล	52
5.5 การทำงานของ Web Page ที่ใช้ในการค้นหาภาพในฐานข้อมูล	52
บทที่ 6 ผลการทดลอง	54
6.1 การทดลองระบบจัดเก็บและสื่อสารภาพทางการแพทย์	54
6.1.1 การทำงานของโปรแกรม DICOM Viewer	54
6.1.2 การทำงานของเครื่องมือของโปรแกรม DICOM Viewer	56
6.1.2.1 การปรับ Window/Level	56
6.1.2.2 การปรับ Alpha Channel	56
6.1.2.3 การใช้แว่นขยาย	57
6.1.2.4 การย่อ/ขยายภาพ	57
6.1.2.5 การเลื่อนภาพ	58
6.1.2.6 การวัดขนาด	58
6.1.2.7 การสร้างวงรี	59
6.1.2.8 การสร้างกรอบสี่เหลี่ยม	59
6.1.2.9 การเขียนคำอธิบาย	60
6.1.2.10 การสร้างลูกศร	60
6.1.2.11 การวัดขนาดมุม	61
6.1.2.12 การทำ Highlight	61
6.1.2.13 การสร้างกรอบเลือกแบบสี่เหลี่ยม	62
6.1.2.14 การสร้างกรอบเลือกแบบอิสระ	62

สารบัญ (ต่อ)

	หน้า
6.1.2.15 การลดขนาดกรอบเลือกแบบอิสระ.....	62
6.1.2.16 การสร้างกรอบเลือกแบบวงรี.....	63
6.1.2.17 การสร้างกรอบเลือกแบบหลายเหลี่ยม.....	63
6.1.2.18 การสร้างกรอบเลือกแบบ Freehand.....	63
6.1.2.19 การสร้างกรอบเลือกที่มีค่าความสว่างตามที่กำหนด.....	64
6.2 การทดลองระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ.....	65
6.2.1 ฐานข้อมูลภาพ.....	65
6.2.2 ขั้นตอนการทดลอง.....	69
6.2.3 ผลการทดลอง.....	72
บทที่ 7 สรุปผลและวิจารณ์.....	74
บทที่ 8 โปรแกรม.....	75
8.1 โปรแกรม PACS.....	75
8.2 โปรแกรม VqKmeanCalculate.....	82
8.2.1 VQ_KmeanKDTree.h.....	82
8.2.2 VQ_KmeanKDTree.cpp.....	83
8.2.3 kmeanCluster.h.....	89
8.2.4 VQ_KmeanKDTree.cpp.....	89
8.3 โปรแกรม VqRetrieval.....	92
8.3.1 yamlWriteRead.h.....	92
8.3.2 yamlWriteRead.cpp.....	92
8.3.3 VqRetrieval.cpp.....	93
8.4 โปรแกรม Web Page สำหรับเก็บภาพลงในฐานข้อมูล.....	99
8.4.1 DBCommand.php.....	99
8.4.2 Php.php.....	104
8.5 โปรแกรม Web Page สำหรับค้นหาภาพในฐานข้อมูล.....	112
8.5.1 SearchInner.php.....	112
8.5.1 Retrieval.php.....	113
บรรณานุกรม/เอกสารอ้างอิง.....	119

สารบัญ (ต่อ)

	หน้า
สรุปค่าใช้จ่ายในการดำเนินโครงการวิจัย.....	125
ประวัตินักวิจัย.....	126

สารบัญตาราง

ตารางที่	หน้า
2.1 ตัวอย่างรูป X-Ray ทางกรแพทย์ที่ใช้ระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ.....	24
6.1 กลุ่มของภาพที่ใช้ในการทดลอง.....	65
6.2 ผลการทดลองการวัดค่า Precision เกลี่ย.....	72

สารบัญภาพ

ภาพที่	หน้า
2.1 ผังการติดต่อระหว่างระบบ PACS, HIS, และเครื่องสแกนภาพทางการแพทย์.....	7
2.2 ผังระบบเน็ตเวิร์กของระบบ PACS.....	8
2.3 DICOM Viewer.....	8
2.4 การปรับ Window Level.....	9
2.5 แวนขยาย.....	10
2.6 การขยายหรือลดภาพ.....	10
2.7 การเลือกพื้นที่ที่ต้องการ.....	11
3.1 ผังการแปลง Wavelet 1 มิติแบบเร็ว.....	29
3.2 ผังการแปลง Wavelet 2 มิติแบบเร็ว.....	29
3.3 ตัวอย่างของเครื่องมือที่ใช้ในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดที่นิยามโดย Prim.....	30
3.4 ตัวอย่างการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดโดยใช้วิธีของ Prim.....	31
3.5 ตัวอย่างของทดสอบรันแบบ 2 มิติของ Wald และ Wolfowitz.....	33
3.6 ตัวอย่างการทดสอบรันของ Wald และ Wolfowitz ที่เงื่อนไขแตกต่างกัน.....	34
3.7 การทดลองหาค่า $\Pr[R = k]$	36
4.1 ผังการคำนวณ Φ^B	40
4.2 ตัวอย่างของต้นไม้แผ่ทั่วที่น้อยที่สุดของ $\{a_1, \dots, a_8, b_1, \dots, b_8\}$	46
4.3 ตัวอย่างทดสอบรันในบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$	47
5.1 ระบบค้นหารูปภาพโดยรายละเอียดของภาพ.....	49
5.2 Flowchart การทำงานของโปรแกรม VqKmeanCalculate.....	50
5.3 Flowchart การทำงานของโปรแกรม VqRetrieval.....	51
5.4 Web page สำหรับป้อนภาพลงในฐานข้อมูล.....	52
5.5 Web page สำหรับสร้างฐานข้อมูลใหม่.....	52
5.6 Web page สำหรับป้อนภาพที่ต้องการค้นหา.....	53
5.7 Web Page สำหรับแสดงภาพผลลัพธ์ของการค้นหา.....	53
6.1 DICOM Viewer.....	54
6.2 การแสดงภาพของ DICOM Viewer.....	55
6.3 Toolbar ของ DICOM Viewer.....	55
6.4 การทดลองปรับ Window/Level.....	56

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
6.5 การทดลองปรับ Alpha Channel.....	56
6.6 การทดลองใช้แว่นขยายเพื่อส่องดูเฉพาะบริเวณที่ต้องการ.....	57
6.7 การทดลองย่อ/ขยายภาพ.....	57
6.8 การทดลองเลื่อนภาพ.....	58
6.9 การทดลองวัดขนาดของวัตถุที่สนใจในภาพ.....	58
6.10 การทดลองสร้างวงรีเพื่อล้อมกรอบวัตถุที่สนใจในภาพ.....	59
6.11 การทดลองสร้างกรอบสี่เหลี่ยมรอบวัตถุที่สนใจในภาพ.....	59
6.12 การทดลองเขียนคำอธิบายในภาพ.....	60
6.13 การทดลองเพื่อสร้างลูกศรชี้ไปยังวัตถุที่สนใจในภาพ.....	60
6.14 การทดลองวัดขนาดมุมของวัตถุที่สนใจในภาพ.....	61
6.15 การทดลอง Highlight วัตถุที่สนใจในภาพ.....	61
6.16 การทดลองสร้างกรอบเลือกแบบสี่เหลี่ยม.....	62
6.17 การทดลองสร้างกรอบเลือกแบบอิสระ.....	62
6.18 การทดลองลดขนาดกรอบเลือกแบบอิสระ.....	62
6.19 การทดลองสร้างกรอบเลือกแบบวงรี.....	63
6.20 การทดลองสร้างกรอบเลือกแบบหลายเหลี่ยม.....	63
6.21 การทดลองสร้างกรอบเลือกแบบ Freehand.....	63
6.22 การทดลองสร้างกรอบเลือกที่มีค่าความสว่างตามที่กำหนด.....	64
6.23 ตัวอย่างการค้นหาภาพช่องท้อง.....	70
6.24 ตัวอย่างการค้นหาภาพกระดูกสันหลัง.....	71
6.25 กราฟ Precision เปรียบ.....	72

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

ภาพทางการแพทย์เช่นภาพ Computed Tomography (CT), Magnetic Resonance Image (MRI) และ X-Ray เป็นเครื่องมือสำคัญในการวินิจฉัยโรคต่างๆ ในปัจจุบันภาพทางการแพทย์เหล่านี้ถูกสร้างขึ้นเป็นจำนวนมาก เนื่องจากเครื่องสแกนภาพทางการแพทย์เช่นเครื่อง X-Ray, CT Scan และ MRI มีจำนวนมากขึ้นและมีราคาถูกลง เครื่องสแกนภาพทางการแพทย์ส่วนใหญ่มีการสนับสนุนเทคโนโลยีภาพดิจิทัลและการสื่อสารทางการแพทย์ (Digital Imaging and Communications in Medicine, DICOM) ซึ่งเป็นเทคโนโลยีใหม่ที่ใช้ในการจัดการ, จัดเก็บ, พิมพ์ และส่งภาพทางการแพทย์ระหว่างคอมพิวเตอร์และเครื่องถ่ายภาพทางการแพทย์ ดังนั้นโรงพยาบาลส่วนใหญ่จึงนิยมเก็บไฟล์ภาพทางการแพทย์ในรูปแบบ DICOM

แต่อย่างไรก็ตามก็ยังคงมีปัญหาในเรื่องของการจัดเก็บและบริการภาพทางเหล่านี้ เนื่องจากโรงพยาบาลแต่ละแห่งมักมีเครื่องสแกนภาพอยู่หลายเครื่องและหลายประเภท และมีหลายแผนกภายในโรงพยาบาลที่จำเป็นต้องใช้งานภาพเหล่านี้ ดังนั้นจึงมีความจำเป็นจะต้องมีระบบจัดเก็บและสื่อสารภาพทางการแพทย์ (Picture Archiving and Communication Systems, PACS) ซึ่งเป็นระบบที่ทำหน้าที่เชื่อมโยงเครื่องสแกนภาพทั้งหมดเข้ากับเครือข่ายคอมพิวเตอร์ของโรงพยาบาล เพื่อทำหน้าที่ดังนี้ 1) รับและส่งภาพและข้อมูลผู้ป่วยตามมาตรฐาน DICOM ผ่านระบบเครือข่ายคอมพิวเตอร์ 2) จัดเก็บข้อมูลไว้ในฐานข้อมูล 3) ส่งข้อมูลภาพไปยังแผนกต่างๆของโรงพยาบาลเช่น ห้องตรวจโรค หรือคลินิก เป็นต้น และ 4) บริหารจัดการข้อมูลภาพ

นอกจากนี้ระบบ PACS ยังต้องสามารถเชื่อมโยงกับระบบข้อมูลรังสีวิทยา (Radiology Information System: RIS) และระบบข้อมูลโรงพยาบาล (Hospital Information System: HIS) เพื่อเชื่อมโยงข้อมูลภาพกับข้อมูลอื่น ๆ ของผู้ป่วย เช่น ประวัติผู้ป่วย ประวัติการรักษาและผลการตรวจรักษา เป็นต้น การเชื่อมโยงข้อมูลของระบบทั้ง 3 เข้าด้วยกัน มีส่วนช่วยเพิ่มประสิทธิภาพการรักษาผู้ป่วยในขั้นตอนต่าง ๆ เช่น การวินิจฉัยโรค การตัดสินใจทางคลินิก การให้เหตุผลเชิงประสบการณ์ (Case-based Reasoning : CBR) และการเวชปฏิบัติเชิงประจักษ์ (Evidence-based Medicine : EBM) เป็นต้น ยกตัวอย่างเช่น แพทย์อาจจะค้นหาข้อมูลของผู้ป่วยอื่น ๆ ที่เคยได้รับการรักษาและมีอาการของโรคคล้ายกับผู้ป่วยที่กำลังรักษาอยู่(CBR) เช่น ประวัติการวินิจฉัยโรค ประวัติการรักษา เป็นต้น เพื่อเป็นข้อมูลที่ใช้ในการตัดสินใจ วินิจฉัยโรคของผู้ป่วย และเลือกวิธีการรักษาที่ดีที่สุดสำหรับผู้ป่วย (EBM)

จากตัวอย่างดังกล่าว จะเห็นว่าประสิทธิภาพของการรักษาผู้ป่วยขึ้นกับ ความถูกต้องของ ข้อมูลที่ได้จากการค้นหา และประสิทธิภาพของวิธีการค้นหาข้อมูล วิธีการค้นหาข้อมูลที่นิยมใช้ใน ปัจจุบันคือ การค้นหาโดยใช้ข้อความหรือคำพูด (Text-Based Retrieval) เป็นหลัก เช่นการค้นหา ข้อมูลโดยใช้ชื่อผู้ป่วย, อาการของโรค, และประวัติผู้ป่วยเป็นต้น

การค้นหาโดยใช้ข้อความเพียงอย่างเดียวมีข้อจำกัดต่างๆมากมายเช่น การค้นหาโดยใช้ชื่อ หรือรหัสผู้ป่วย ทำให้แพทย์ต้องสามารถจดจำชื่อของผู้ป่วยที่มีอาการของโรคคล้ายกันได้ ซึ่งเป็นไป ไม่ได้ในทางปฏิบัติ หรือค้นหาตามอาการของโรคเช่น ปวดหัว หน้ามืด หรือ ตัวร้อน เป็นต้น มี ข้อจำกัดคือมีหลายโรคที่มีอาการคล้ายกัน ทำให้ผลลัพธ์ของการค้นหาอาจจะไม่ตรงกับโรคที่ผู้ป่วย เป็น จากปัญหาดังกล่าวทำให้แพทย์สามารถค้นข้อมูลภาพและประวัติการรักษาของผู้ป่วยที่กำลัง รักษาอยู่เพียงเท่านั้น แต่ไม่สามารถนำข้อมูลของผู้ป่วยอื่นๆ มาใช้ประกอบการตัดสินใจและการ รักษาได้

จากความก้าวหน้าของเทคโนโลยีการประมวลผลภาพในปัจจุบัน ทำให้เกิดการพัฒนา ระบบ การค้นหาภาพโดยใช้รายละเอียดของภาพ (Content-Based Image Retrieval System : CBIRS) การ ทำงานของระบบ CBIRS มีดังนี้ เริ่มต้นจากผู้ใช้ป้อนภาพที่ต้องการค้นหาให้กับระบบ จากนั้นระบบ จะนำภาพดังกล่าวไปเปรียบเทียบกับความเหมือนกับภาพที่ถูกเก็บไว้ในฐานข้อมูล การเปรียบเทียบ ความเหมือนที่นิยมใช้ในระบบ CBIRS คือ เปรียบเทียบสี, Texture และรูปร่าง หลังจากนั้นระบบจะ แสดงภาพที่อยู่ในฐานข้อมูลที่ใกล้เคียงกับภาพที่ผู้ใช้ต้องการค้นหา โดยเรียงจากภาพที่เหมือนมาก ที่สุด ไปยังภาพที่เหมือนน้อยที่สุด

การใช้ระบบ CBIRS มาค้นหาภาพทางการแพทย์ มีประโยชน์อย่างมาก เนื่องจากแพทย์ สามารถนำเอาภาพ CT หรือ X-Ray ของผู้ป่วยที่กำลังรักษาไปใช้ค้นหาภาพของผู้ป่วยอื่นๆใน ฐานข้อมูลของระบบ PACS ที่คล้ายกัน จากนั้นแพทย์สามารถเข้าไปดูรายละเอียดการวินิจฉัย ขั้นตอนการรักษา และอื่นๆ เพื่อนำมาประกอบการรักษาผู้ป่วยให้มีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการวิจัย

โครงการวิจัยนี้มีวัตถุประสงค์ที่จะพัฒนาระบบจัดเก็บและสื่อสารภาพทางการแพทย์ (Picture Archiving and Communications System, PACS) และระบบค้นหาภาพทางการแพทย์ด้วย รายละเอียดของภาพ (Content-Based Medical Image Retrieval, CBMIRS) ซึ่งประกอบด้วย คอมพิวเตอร์และซอฟต์แวร์ดังนี้

1. DICOM Server ทำหน้าที่เป็น Database Server เพื่อจัดเก็บไฟล์ DICOM สำหรับผู้ป่วย
2. DICOM Worklist Provider เป็นโปรแกรมสำหรับเชื่อมโยง PACS กับเครื่องสแกนภาพทาง การแพทย์และระบบข้อมูลโรงพยาบาลและระบบข้อมูลรังสีวิทยา เพื่อลดภาระการป้อน ข้อมูลของผู้ป่วย

3. DICOM Directory เป็นซอฟต์แวร์ที่ใช้จัดเก็บข้อมูล DICOM ให้เป็นหมวดหมู่ เพื่อเพิ่มความสะดวกในการอ่านข้อมูลของผู้ป่วย
4. DICOM Viewer เป็น Web Base Application สำหรับแพทย์ใช้ดูภาพถ่ายทางการแพทย์และข้อมูลอื่นๆของผู้ป่วย และมีเครื่องมือสำหรับการวิเคราะห์และวินิจฉัยอาการของโรค
5. Web Application สำหรับใช้ค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพซึ่งสามารถเชื่อมโยงกับฐานข้อมูลของ DICOM Server

1.3 ขอบเขตของโครงการวิจัย

ขอบเขตของโครงการวิจัยนี้มีดังต่อไปนี้

1. พัฒนา DICOM Server ซึ่งประกอบด้วยคอมพิวเตอร์ที่ติดตั้งโปรแกรม MySQL และ Apache สำหรับระบบ PACS และ CBIRS
2. พัฒนา DICOM Worklist Provider เพื่อใช้สื่อสารระหว่างระบบ PACS กับเครื่องสแกนภาพทางการแพทย์และระบบ HIS และ RIS
3. พัฒนา DICOM Directory เพื่อใช้จัดเก็บข้อมูล DICOM ให้เป็นหมวดหมู่ ทำให้เพิ่มความสะดวกในการอ่านข้อมูลของผู้ป่วย
4. พัฒนา DICOM Viewer ซึ่งเป็น Web Base Application ที่ใช้ในการแสดงภาพและข้อมูลของผู้ป่วย และมีเครื่องมือพื้นฐานสำหรับวิเคราะห์และวินิจฉัยภาพเช่น เครื่องมือที่ใช้วัดขนาดและพื้นที่ของอวัยวะที่บาดเจ็บหรือเนื้องอก การปรับปรุงคุณภาพของภาพเพื่อให้สามารถมองภาพได้ชัดเจนขึ้น การย่อหรือขยายภาพ เป็นต้น
5. พัฒนา Web Application สำหรับใช้ในการค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ เพื่อดึงภาพที่ใกล้เคียงกับภาพของผู้ป่วย ที่เก็บไว้ในฐานข้อมูลของ DICOM Server

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบ Picture Archives and Communication System (PACS)

ระบบ PACS เป็นระบบที่ใช้เพื่อการบริหารและจัดเก็บข้อมูลภาพทางการแพทย์ และเป็นตัวกลางระหว่างระบบ Hospital Information System (HIS) และเครื่องสแกนภาพทางการแพทย์ เช่น เครื่อง MRI Scan, CT Scan และ X-Ray เป็นต้น

2.1.1 หน้าที่การทำงานของระบบ PACS

เพื่อให้ระบบ PACS สามารถทำงานร่วมกับระบบ HIS และเครื่องสแกนภาพของโรงพยาบาลได้อย่างมีประสิทธิภาพ และทำให้ผู้ป่วยได้รับบริการที่ดีขึ้นจากโรงพยาบาลทางผู้วิจัยได้ออกแบบให้ระบบ PACS มีหน้าที่การทำงานทั้งหมด 3 อย่างดังต่อไปนี้

1. ระบบ HIS จะส่งข้อมูลของผู้ป่วยที่ต้องการตรวจรักษามาให้กับระบบ PACS เช่น ชื่อผู้ป่วย, ชื่อแพทย์, ชื่อโรงพยาบาล, และชนิดของเครื่องสแกนภาพเช่น MRI หรือ X-Ray เป็นต้น จากนั้นระบบ PACS จะนำข้อมูลดังกล่าวมาสร้างเป็น DICOM Worklist (DMWL) และส่งต่อไปให้กับเครื่องสแกนภาพที่ผู้ป่วยถูกส่งตัวไปตรวจรักษา
2. เนื่องจากในการสแกนผู้ป่วยหนึ่งครั้งจะได้ผลลัพธ์เป็นภาพเป็นจำนวนมาก โดยที่แต่ละภาพจะแยกเก็บไว้ในไฟล์ DICOM แต่ละไฟล์ ภาพเหล่านี้มีความสัมพันธ์ซึ่งกันและกันไม่สามารถแยกออกจากกันได้ยกตัวอย่างเช่น คนไข้ A ถูกส่งตัวไปตรวจที่เครื่อง CT Scan เพื่อศึกษา (Study) Abdomen^3 PhaseLiver MSMC โดยแบ่งออกเป็น 4 อนุกรม (Series) คือ
 - a. AngioRoutine 3.0 B20f ได้ผลลัพธ์เป็นภาพทั้งหมด 441 ภาพ
 - b. LIVER 5.0 B20f ได้ผลลัพธ์เป็นภาพทั้งหมด 70 ภาพ
 - c. Topogram ได้ผลลัพธ์เป็นภาพทั้งหมด 2 ภาพ
 - d. Liver C-5.0 B30f ได้ผลลัพธ์เป็นภาพทั้งหมด 70 ภาพ
 ดังนั้นภาพทั้งหมดที่ได้จากเครื่อง CT Scan จึงมีความสัมพันธ์กันเป็นลำดับชั้น (หรือเรียกว่า Directory) ดังนี้

Patient → Study → Series → Image

ระบบ PACS จะรับไฟล์ DICOM ของภาพทั้งหมดที่ได้จากเครื่อง CT Scan มาสร้างเป็น DICOM Directory และเก็บ DICOM Directory และไฟล์ DICOM ทั้งหมดไว้ในฐานข้อมูลของระบบ PACS

3. เมื่อแพทย์ต้องการดูผลการสแกน แพทย์จะส่งผ่านระบบ HIS มายังระบบ PACS จากนั้นระบบ PACS จะทำการค้นหา DICOM Directory และไฟล์ DICOM ในฐานข้อมูลที่ตรงกับความต้องการของแพทย์ แล้วนำมาแสดงใน DICOM Viewer

2.1.2 ประโยชน์ที่จะได้รับจากระบบ PACS

1. เนื่องจากระบบ PACS มีการออกแบบให้ใช้งาน DICOM Worklist ร่วมกับระบบ HIS และเครื่องสแกนภาพ จึงทำให้ลดภาระในการป้อนข้อมูลต่างๆของผู้ป่วยให้กับเครื่องสแกนภาพ เนื่องจากข้อมูลส่วนใหญ่จะถูกส่งมาจากระบบ HIS ซึ่งจะลดปัญหาของการป้อนข้อมูลซ้ำซ้อนซึ่งมีโอกาสผิดพลาดได้ง่าย
2. เนื่องจากการสร้างและจัดเก็บ DICOM Directory ร่วมกับไฟล์ DICOM ทุกครั้งที่ผู้ป่วยถูกส่งไปตรวจรักษา และ DICOM Viewer จะเปิดไฟล์ DICOM ร่วมกับ DICOM Directory ทุกครั้งที่แพทย์ต้องการดูผลการตรวจ ดังนั้นแพทย์จึงบริหารจัดการไฟล์ DICOM ของผู้ป่วยซึ่งมีอยู่หลายไฟล์และมีความสัมพันธ์กันได้อย่างมีประสิทธิภาพ
3. ระบบ PACS ได้ถูกออกแบบให้สามารถทำงานร่วมกับระบบ HIS และเครื่องสแกนภาพต่าง ๆ ของโรงพยาบาลได้อย่างมีประสิทธิภาพ แพทย์ไม่จำเป็นต้องออกจากระบบ HIS เพื่อไปใช้งานโปรแกรมอื่นในการติดต่อกับเครื่องสแกนต่างๆ อีกต่อไป

2.1.3 องค์ประกอบของระบบ PACS

ระบบ PACS ประกอบด้วยระบบย่อย 3 ระบบคือ

1. DICOM Server

เป็น Server ที่ทำหน้าที่เก็บ DICOM Directory และ DICOM Files ที่รับมาจากเครื่องสแกนภาพ

2. DICOM Worklist Provider

เป็นโปรแกรมซึ่งมีหน้าที่ทั้งหมด 2 อย่างคือ

- รับข้อมูลของคนป่วยมาจากระบบ HIS เช่น ชื่อผู้ป่วย, ชื่อหมอ, ชื่อโรงพยาบาล, และชนิดของเครื่องสแกนภาพ จากนั้นจึงนำข้อมูลมาสร้าง DICOM Worklist และส่ง DICOM Worklist ไปยังเครื่องสแกนภาพที่ผู้ป่วยจะถูกส่งตัวไปตรวจรักษา

- รับไฟล์ DICOM ทั้งหมดจากเครื่องสแกนภาพมาเพื่อสร้าง DICOM Directory และส่ง DICOM Directory และไฟล์ DICOM ทั้งหมดไปเก็บไว้ในฐานข้อมูลของระบบ

3. DICOM Viewer

เป็นโปรแกรมที่ทำหน้าที่ค้นหา DICOM Directory และไฟล์ DICOM ที่แพทย์ต้องการ และแสดงภาพของไฟล์ DICOM นอกจากนี้ DICOM Viewer ยังประกอบด้วยเครื่องมือพื้นฐานที่ช่วยแพทย์ในการวินิจฉัยภาพจากไฟล์ DICOM

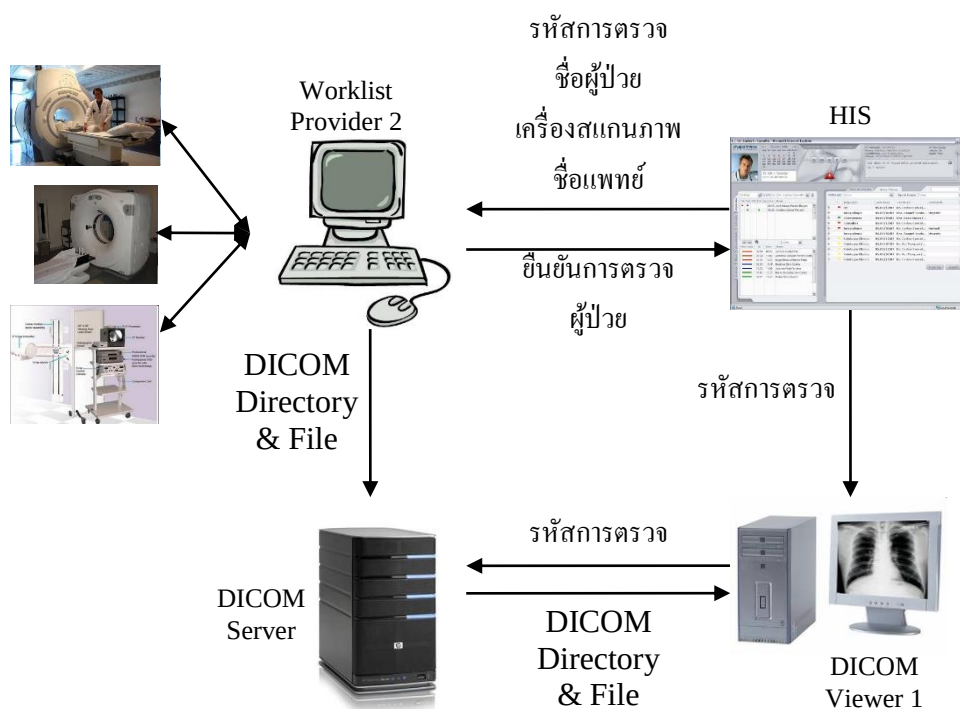
2.1.4 ขั้นตอนการทำงานร่วมกันระหว่างระบบ PACS, HIS, และเครื่องสแกนภาพทางการแพทย์

ภาพที่ 2.1 แสดงผังการติดต่อระหว่างระบบ PACS, HIS, และเครื่องมือทางการแพทย์ ซึ่งมีรายละเอียดการทำงานดังนี้

1. ระบบ HIS ส่งรายละเอียดของผู้ป่วยที่จะถูกส่งไปตรวจรักษาที่เครื่องสแกนภาพทางการแพทย์ เช่นรหัสการตรวจ, ชื่อผู้ป่วย, ชนิดของเครื่องสแกนภาพ, ชื่อแพทย์ เป็นต้น ไปให้กับ DICOM Worklist Provider (บันทึก ระบบ PACS จะใช้รหัสการตรวจที่ถูกส่งจากระบบ HIS ในการอ้างอิง DICOM Directory และไฟล์ DICOM ของผู้ป่วย)
2. DICOM Worklist Provider จะสร้าง Worklist จากข้อมูลที่ถูกส่งมาจากระบบ HIS
3. DICOM Worklist Provider ส่ง Worklist ไปให้กับเครื่องสแกนภาพที่ผู้ป่วยถูกส่งตัวไปตรวจ
4. หลังจากผู้ป่วยถูกตรวจเสร็จแล้ว เครื่องสแกนภาพจะส่งไฟล์ DICOM ทั้งหมดกลับไปให้กับ DICOM Worklist Provider
5. DICOM Worklist Provider จะสร้าง DICOM Directory จากไฟล์ DICOM ที่ได้รับ
6. DICOM Worklist Provider จะส่ง DICOM Directory และไฟล์ DICOM ทั้งหมดไปเก็บไว้ใน DICOM Server และจะส่งข้อความยืนยันการตรวจผู้ป่วยกลับไปให้กับระบบ HIS เพื่อให้ระบบ HIS แจ้งให้กับแพทย์ต่อไป
7. เมื่อแพทย์ต้องการตรวจดูภาพของผู้ป่วย รหัสการตรวจจะถูกส่งจากระบบ HIS ไปให้กับ DICOM Viewer จากนั้น DICOM Viewer จะใช้รหัสการตรวจในการค้นหา

DICOM Directory และไฟล์ DICOM ซึ่งเก็บไว้ใน DICOM Server ที่ตรงกับรหัสการตรวจ

8. DICOM Server จะส่ง DICOM Directory และไฟล์ DICOM ไปให้กับ DICOM Viewer
9. DICOM Viewer จะแสดง DICOM Directory และภาพที่เก็บไว้ในไฟล์ DICOM



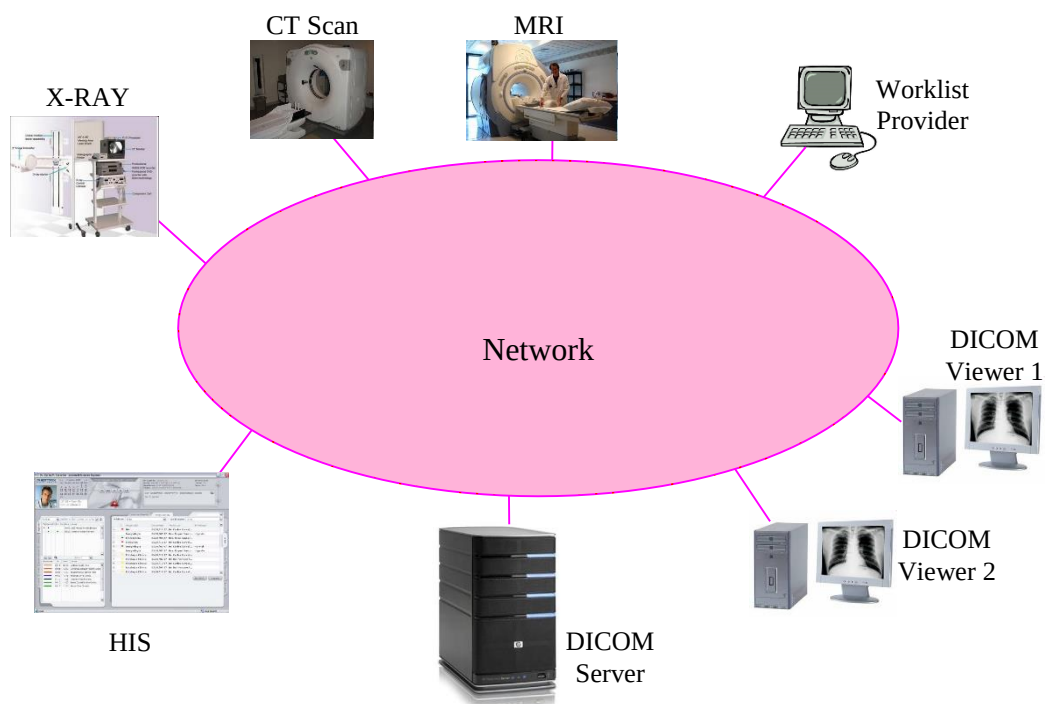
ภาพที่ 2.1 ผังการติดต่อระหว่างระบบ PACS, HIS, และเครื่องสแกนภาพทางการแพทย์

2.1.5 ระบบเน็ตเวิร์คของระบบ PACS

ภาพที่ 2.2 แสดงการเชื่อมต่อระบบ PACS เข้ากับเน็ตเวิร์คของโรงพยาบาล ซึ่งประกอบด้วย DICOM Server 1 เครื่อง DICOM Viewer 2 เครื่องประจำอยู่ที่ห้องแพทย์ DICOM Worklist Provider 1 เครื่องประจำอยู่ที่หน้าห้อง MRI หรือ CT Scan

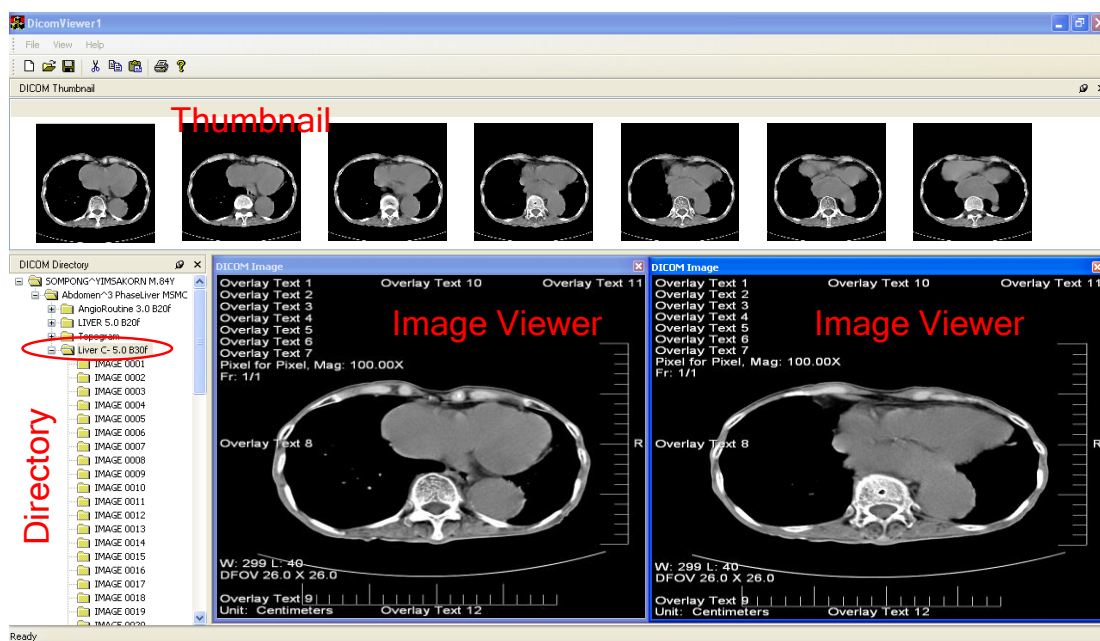
DICOM Worklist Provider ติดต่อกับเครื่องสแกนภาพทางการแพทย์ผ่าน IP Address เช่นเดียวกับโปรแกรม E-Film

ระบบ HIS จะติดต่อกับ DICOM Worklist Provider และ DICOM Viewer ผ่าน Web base



ภาพที่ 2.2 ผังระบบเน็ตเวิร์คของระบบ PACS

2.1.6 DICOM Viewer



ภาพที่ 2.3 DICOM Viewer

เมื่อแพทย์ต้องการดูข้อมูลภาพของผู้ป่วย ระบบ HIS จะส่งรหัสการตรวจของผู้ป่วยมาให้ DICOM Viwer จากนั้น DICOM Viwer จะค้นหา DICOM Directory และไฟล์ DICOM ที่แพทย์ต้องการจาก DICOM Server และนำมาแสดงผล

ภายใน DICOM Viwer ประกอบด้วยหน้าต่างหลัก 3 ชนิดคือ

1. Directory

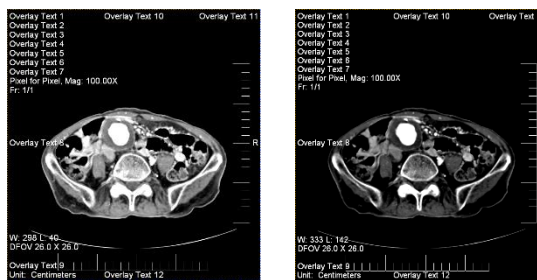
ทำหน้าที่แสดง DICOM Directory แพทย์สามารถที่จะตรวจสอบข้อมูลการตรวจรักษาต่างๆของผู้ป่วยได้เช่น วิธีการศึกษา(จากภาพที่ 2.3 คือ Abdomen^3 PhaseLiver MSMC), ชนิดของอนุกรม (จากภาพที่ 2.3 คือ AngioRoutine 3.0 B20f, LIVER 5.0 B20f, Topogram, และ Liver C-5.0 B30f), และภาพผลลัพธ์

2. Thumbnail

ทำหน้าที่แสดงภาพตัวอย่าง เมื่อแพทย์คลิกเลือกอนุกรมการตรวจรักษาที่ต้องการ เช่นจากภาพที่ 2.3 ถ้าแพทย์เลือกอนุกรม Liver C-5.0 B30f ภาพทั้งหมดของอนุกรม Liver C-5.0 B30f จะถูกแสดงไว้ใน Thumbnail โดยภาพใน Thumbnail จะถูกเรียงลำดับแบบเดียวกับที่แสดงไว้ใน Directory เช่น Image 0001, Image 0002, Image 0003,...

3. Image Viewer

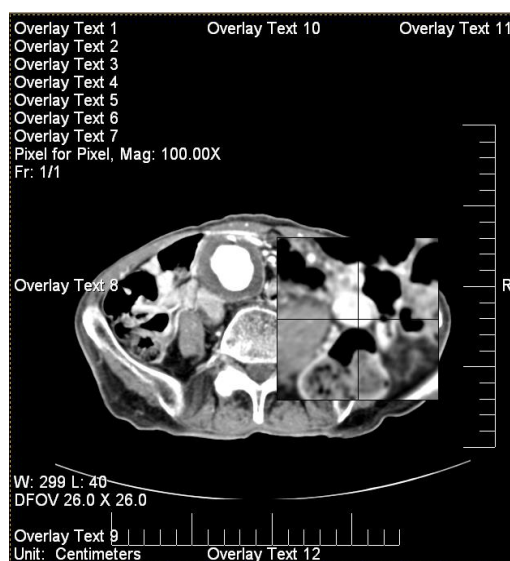
เมื่อแพทย์ดับเบิลคลิกที่ Image ใน Directory หรือภาพที่ต้องการใน Thumbnail ภาพดังกล่าวจะถูกแสดงไว้ใน Image Viwer แพทย์สามารถที่จะแสดงภาพได้พร้อมกันหลายภาพ (จำนวนภาพทั้งหมดที่สามารถเปิดได้ ขึ้นกับขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ที่ใช้เปิด Image Viwer) Image Viwer จะมีเครื่องมือพื้นฐานต่างๆ เพื่ออำนวยความสะดวกให้กับแพทย์เพื่อใช้วิเคราะห์คุณสมบัติของภาพ เช่น เครื่องมือที่ใช้ปรับ Window Level, แวนขยาย, ขยายหรือลดภาพ, เลือกพื้นที่ที่ต้องการ เป็นต้น



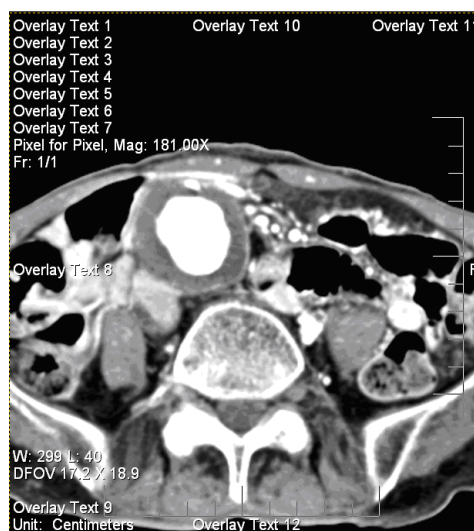
W=298, L=40

W=333, L=142

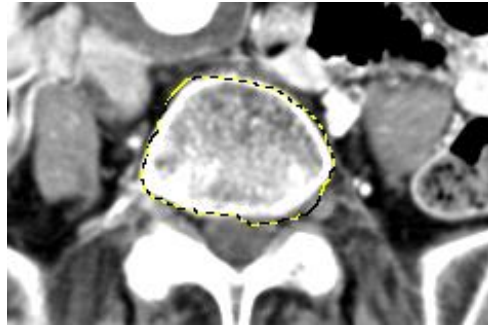
ภาพที่ 2.4 การปรับ Window Level



ภาพที่ 2.5 แวนขยาย



ภาพที่ 2.6 การขยายหรือลดภาพ



ภาพที่ 2.7 การเลือกพื้นที่ที่ต้องการ

2.2 CBMIRS (Content-Based Medical Image Retrieval System)

ถึงแม้ว่าระบบ PACS มีข้อดีมากมาย แต่ก็ยังมีปัญหาที่สำคัญมากอย่างหนึ่งคือ ระบบ PACS สามารถค้นหาข้อมูลภาพทางการแพทย์ได้โดยใช้ข้อความเท่านั้น ตัวอย่างเช่นแพทย์หรือรังสีแพทย์สามารถค้นข้อมูลโดยใช้ชื่อของผู้ป่วย หรือชื่อแพทย์ผู้รักษาเท่านั้น ดังนั้นจึงค้นได้เพียงชื่อของผู้ป่วยที่กำลังรักษา หรือแพทย์อาจจะต้องจำชื่อของผู้ป่วยที่เคยรักษาก่อนหน้านี้ที่มีอาการคล้ายกับผู้ป่วยกำลังรักษา ดังนั้นจึงไม่สามารถนำเอาข้อมูลภาพจำนวนมากที่ถูกเก็บไว้ในฐานข้อมูลของ PACS มาใช้ประโยชน์ได้อย่างมีประสิทธิภาพ

ฉะนั้นจึงเป็นที่ชัดเจนแล้วว่า เพื่อที่ใช้ข้อมูลภาพที่อยู่ในฐานข้อมูลได้อย่างมีประสิทธิภาพสูงสุด จำเป็นจะต้องมีการพัฒนาวิธีการค้นหาภาพโดยใช้คุณสมบัติที่อยู่ภายในภาพ นักวิจัยหลายหลายกลุ่มเช่น Computer Vision, Database Management, Human-Computer Interface และ Information Retrieval จึงได้หันมาพัฒนาระบบค้นหาภาพโดยรายละเอียดของภาพ [1-6]

2.2.1 ปัญหาของระบบค้นหาภาพโดยรายละเอียดของภาพ

ปัญหาของการค้นหาภาพโดยรายละเอียดของมีอยู่หลายอย่างที่จำเป็นจะต้องแก้ไขก่อนที่จะสามารถนำไปใช้งานจริงได้ ดังนี้

2.2.1.1 การทำความเข้าใจรายละเอียดภาพของมนุษย์

เนื่องจากระบบการค้นหาภาพทางการแพทย์เป็นระบบที่พยายามเลียนแบบการทำงานของสมองของมนุษย์ในการค้นหาภาพที่คล้ายกัน ดังนั้นจึงมีความจำเป็นอย่างมากที่จะต้องศึกษาวิธีที่มนุษย์ทำความเข้าใจรายละเอียดของภาพ งานวิจัยในหัวข้อนี้ได้รับความสนใจเป็นอย่างมากในปัจจุบัน โดยมุ่งไปที่การศึกษาวิธีการที่มนุษย์ทำความเข้าใจรายละเอียดของภาพและวิธีการรวมแบบจำลองของวิธีการทำความเข้าใจเข้ากับระบบค้นหาภาพ [7-14]

2.2.1.2 คุณลักษณะและการวัดความเหมือนของภาพ

รายละเอียดของภาพสามารถถูกบรรยายโดยสี, รูปร่าง, พื้นผิว และการจัดวางองค์ประกอบของภาพ ในระบบการค้นหาภาพการบรรยายดังกล่าวสามารถสร้างจากคุณลักษณะ

พื้นฐานของภาพเช่น Color Histogram, Tamura Feature หรือ Fourier Descriptor คุณลักษณะของภาพที่ดีจะต้องมีคุณสมบัติดังนี้ ถ้ามีภาพสองภาพที่คล้ายกัน คุณลักษณะของภาพทั้งสองจะต้องอยู่ใกล้กันด้วย

นอกจากนี้ความเปลี่ยนแปลงต่างๆในการถ่ายภาพยังทำให้เกิดปัญหาในระบบการค้นหภาพ ความเปลี่ยนแปลงของการถ่ายภาพที่มีปัญหามีดังต่อไปนี้ การหมุนวัตถุ อัตราส่วนภาพ การเลื่อนตำแหน่งกล้อง มุมกล้อง ทิศทางของแสง ความเข้มของแสงและเงา ความเข้มของสี การบิดบังวัตถุในภาพ ความสับสนระหว่างวัตถุกับพื้นหลัง วัตถุอื่นๆที่อยู่ในกลุ่มเดียวกัน

ดังนั้นคุณลักษณะของภาพที่ดีจะต้องไม่ขึ้นกับความเปลี่ยนแปลงต่างๆดังกล่าว แต่อย่างไรก็ตามคุณลักษณะที่ดีจะต้องรักษาสมดุลระหว่างการไม่ขึ้นกับการเปลี่ยนแปลงกับการแยกแยะความแตกต่าง เนื่องจากถ้าคุณลักษณะไม่ขึ้นกับความเปลี่ยนแปลงมากเกินไปจะทำให้ไม่สามารถแยกแยะความแตกต่างที่มีความสำคัญได้

นอกจากนี้งานที่สำคัญมากอีกอย่างของระบบการค้นหภาพคือการเปรียบเทียบความเหมือนระหว่างสองภาพ ดังนั้นปัญหาคือวิธีการวัดความเหมือนของภาพสองภาพ ผลลัพธ์ของการวัดความเหมือนจะต้องตรงกับวิธีการวัดความเหมือนของมนุษย์

2.2.1.3 ความเชื่อมโยงระหว่างคุณลักษณะพื้นฐานของภาพและแนวความคิดขั้นสูง

แพทย์มักนิยมใช้แนวความคิดขั้นสูงในการศึกษาภาพถ่ายทางการแพทย์ ตัวอย่างแนวความคิดแบบง่ายๆคือ เนื้อเยื่อ เส้นเลือด กล้ามเนื้อ กระดูก และผิวหนัง เป็นต้น จากแนวคิดแบบง่ายๆ ก็พัฒนาเป็นแนวความคิดที่ซับซ้อนยิ่งขึ้นเช่น กระดูกซี่โครง ปอด หัวใจ ลำไส้ และบาดแผล เป็นต้น

แต่อย่างไรก็ตามเทคนิคในงาน Computer Vision ทั่วไปในปัจจุบันสามารถทำได้แค่ดึงคุณลักษณะพื้นฐานของภาพได้เท่านั้น ในงานบางอย่างเช่นระบบจดจำหน้าและการตรวจลายนิ้วมือที่สามารถเชื่อมโยงระหว่างคุณลักษณะพื้นฐานกับแนวความคิดขั้นสูงได้ แต่ไม่สามารถทำได้ในกรณีทั่วไป มันเป็นการยากมากที่จะเชื่อมโยงคุณลักษณะพื้นฐานกับแนวความคิดขั้นสูง

เพื่อลดช่องว่างในการแปลความหมาย (Semantic Gap) จำเป็นต้องใช้เทคนิคที่รวมคุณลักษณะหลายๆอย่างไว้ด้วยกัน ตัวอย่างเช่นควรจบรวมคุณลักษณะทางสี (Color Feature) เข้ากับคุณลักษณะพื้นผิว (Texture Feature) เพื่อแยกความแตกต่างระหว่างผิวหนังธรรมดาออกจากมะเร็งผิวหนัง หรือรวมคุณลักษณะทางพื้นผิวกับคุณลักษณะทางรูปร่าง (Shape Feature) เพื่อแยกความแตกต่างระหว่างกระดูกซี่โครงและกระดูกข้อมือ เป็นต้น ขึ้นต่อไปจำเป็นต้องใช้การประมวลผลแบบ Off-line และ On-line การประมวลผลแบบ Off-line สามารถทำได้โดยใช้เทคนิค Supervised Learning, Unsupervised Learning หรือรวมทั้งสองวิธี เครื่องมือในการเรียนรู้เหล่านี้คือ Neural network, Genetic Algorithm และ Clustering สำหรับการประมวลผลแบบ On-line จำเป็นต้องใช้

การติดต่อกับผู้ใช้ที่มีความฉลาดและเป็นมิตร มันจะต้องให้ผู้ใช้สามารถประเมินผลลัพธ์ของการค้นหาภาพ เทคนิคดังกล่าวคือ Relevance Feedback

2.2.1.4 ระบบติดต่อผู้ใช้

ความแตกต่างระหว่างระบบจดจำรูปแบบและระบบค้นหาภาพทางการแพทย์คือ แพทย์มีส่วนสำคัญอย่างมากในระบบที่สอง การค้นหาภาพทางการแพทย์เป็นปัญหาที่ไม่ชัดเจน ขึ้นกับโรคของผู้ป่วยแต่ละคน ตัวอย่างเช่นภาพ X-Ray ทรวงอกสามารถใช้ในการวินิจฉัยโรคที่เกี่ยวข้องกับปอด และโรคที่เกี่ยวข้องกับหัวใจได้เช่นกัน ดังนั้นระบบการค้นหาภาพทางการแพทย์จะต้องมีขั้นตอนที่สามารถให้แพทย์มีส่วนร่วมในการตัดสินใจว่าต้องการค้นหาภาพที่เกี่ยวข้องกับโรคปอดหรือโรคหัวใจ

2.2.1.5 ตัวชี้ในมิติขั้นสูง

เพื่อให้ระบบค้นหาภาพสามารถใช้ได้กับฐานข้อมูลขนาดใหญ่ จำเป็นต้องใช้เทคนิคตัวชี้หลายมิติ (Multidimensional Indexing) วัตถุประสงค์ของการใช้ตัวชี้คือเพื่อให้การค้นหาข้อมูลทำได้รวดเร็ว การใช้ตัวชี้หลายมิติมักมีปัญหา 2 อย่างคือ

1) ปัญหามิติขั้นสูง

โดยปกติจำนวนมิติของเวกเตอร์คุณลักษณะจะมีค่าประมาณ 10^2 [15] แต่อย่างไรก็ตามโครงสร้างของตัวชี้ที่ถูกลำเสนอจะสามารถทำงานได้อย่างมีประสิทธิภาพเมื่อจำนวนมิติมีขนาดเล็กมาก [16]

2) การวัดความเหมือนในปริภูมิที่ไม่ใช่แบบยูคลิดียน

เนื่องจากการวัดในปริภูมิแบบยูคลิดียนอาจจะไม่ใช่การวัดที่มีประสิทธิภาพมากที่สุด จึงจำเป็นต้องพัฒนาวิธีการวัดในแบบอื่นๆ เช่น Histogram Intersection, Cosine, Correlation

2.2.1.6 การประเมินประสิทธิภาพและตัวทดสอบมาตรฐาน

การพัฒนาเทคนิคต่างๆ ให้ก้าวหน้าขึ้นจำเป็นจะต้องมีการประเมินประสิทธิภาพที่มีประสิทธิภาพ ตัวอย่างเช่นค่า SNR ใช้ในการประเมินการลดข้อมูล และ Precision และ Recall ใช้ในการประเมินการค้นหาข้อความ วิธีการประเมินที่ดีจะทำให้เทคนิคต่างๆ ถูกพัฒนาไปในทิศทางที่ถูกต้อง ระบบการค้นหาภาพบางระบบในปัจจุบันใช้วิธีการประเมินแบบ Cost/Time [17] ส่วนระบบอื่น ๆ ที่เหลือใช้ วิธีการประเมินแบบ Precision/Recall ที่ยืมมาจากระบบการค้นหาข้อความ

ถึงแม้ว่าวิธีการประเมินเหล่านี้จะเป็นวิธีที่ดี แต่ก็ยังห่างไกลจากความต้องการเหตุผลหลักอันหนึ่งที่ทำให้วิธีการประเมินเหล่านี้ด้อยประสิทธิภาพลงคือปัญหาความคิดเห็นส่วนตัวของการรับรู้รายละเอียดของภาพ นั่นคือความคิดเห็นส่วนตัวทำให้ไม่สามารถประเมินผลลัพธ์ของการค้นหาได้อย่างถูกต้อง

ปัญหาที่สำคัญอีกอย่างคือการหาตัวทดสอบมาตรฐาน ในการลดขนาดภาพเราใช้ภาพเลนา ซึ่งเป็นภาพที่มีความสมดุลระหว่างพื้นผิวหลายแบบ การลดขนาดภาพวีดีโอก็มีภาพวีดีโอที่เหมาะสมสำหรับการทดสอบ ในการค้นหาข้อความก็มีตัวทดสอบมาตรฐาน สำหรับการค้นหาภาพทางการแพทย์ เริ่มมีการรวบรวมฐานข้อมูลภาพที่ใช้เป็นตัวทดสอบมาตรฐาน ตัวทดสอบที่ดีสำหรับการค้นหาภาพทางการแพทย์ จะต้องเป็นฐานข้อมูลภาพที่มีขนาดใหญ่ และต้องรักษาสัมดุลย์ระหว่างรายละเอียดของภาพ เพื่อทดสอบประสิทธิภาพของคุณลักษณะของภาพและประสิทธิภาพโดยรวมของระบบ

2.2.2 คุณลักษณะของภาพ

รายละเอียดของภาพอาจจะประกอบด้วยรายละเอียดทางสายตาและรายละเอียดทางความหมาย รายละเอียดทางสายตาประกอบด้วย สี พื้นผิว รูปร่าง การวางองค์ประกอบ และอื่นๆ รายละเอียดทางความหมายอาจได้จากการกรอกข้อความโดยมนุษย์หรือใช้กระบวนการที่ซับซ้อนซึ่งอยู่บนพื้นฐานของรายละเอียดทางสายตา งานวิจัยนี้จะมุ่งเน้นเฉพาะการบรรยายรายละเอียดทางสายตาเท่านั้น

ในระบบการค้นหาภาพทางการแพทย์ การบรรยายรายละเอียดทางสายตาจะต้องถูกแทนที่ด้วยคุณลักษณะภาพที่เหมาะสม ในหัวข้อนี้จะแนะนำคุณลักษณะภาพอย่างกว้างๆที่ใช้ในการแสดงสี พื้นผิว รูปร่างและองค์ประกอบของภาพ

2.2.2.1 สี

สีเป็นรายละเอียดทางสายตาที่นิยมใช้มากที่สุดในการค้นหาภาพ [18-26] ค่าสีใน 3 มิติทำให้มันมีศักยภาพในการแยกแยะความแตกต่างได้ดีกว่าค่าระดับสีเทาใน 1 มิติของภาพก่อนที่จะเลือกคุณลักษณะของสีที่เหมาะสม จะต้องเลือกปริภูมิสีก่อน

แต่ละจุดในภาพสามารถแสดงเป็นจุดในระบบ 3 มิติปริภูมิสี ปริภูมิสีที่นิยมใช้ในการค้นหาภาพคือ RGB, Munsell, CIE $L^*a^*b^*$, CIE $L^*u^*v^*$, HSV, และปริภูมิสีตรงข้าม แต่อย่างไรก็ตามจนถึงตอนนี้ยังคงไม่มีปริภูมิสีใดที่ดีที่สุด แต่ปริภูมิสีที่เหมาะสมที่สุด แต่อย่างไรก็ตามสำหรับระบบการค้นหาภาพปริภูมิของสีที่ดีจะต้องมีคุณสมบัติสม่ำเสมอในการรับรู้ (Perceptually Uniform) [22] ปริภูมิสีที่มีความสม่ำเสมอในการรับรู้หมายความว่า ระยะทางระหว่างสี 2 สีในปริภูมิสีจะต้องมีค่าเท่ากับความเหมือนที่วัดได้โดยการรับรู้ของมนุษย์

ปริภูมิ RGB

เป็นปริภูมิสี ที่ใช้กันอย่างกว้างขวางในการแสดงภาพ มันประกอบด้วยองค์ประกอบของแสงสีแดง เขียวและน้ำเงิน แต่ปริภูมิ RGB เป็นปริภูมิสีที่รู้จักกันโดยทั่วไปว่าไม่มีคุณสมบัติความสม่ำเสมอในการรับรู้

ปริภูมิ CIE $L^*a^*b^*$ และ CIE $L^*u^*v^*$

เป็นปริภูมิสีที่มีความสม่ำเสมอในการรับรู้ มันประกอบด้วยองค์ประกอบทางความสว่าง (Lightness) และองค์ประกอบทางสี (a และ b หรือ u และ v)

ปริภูมิ HSV

เป็นปริภูมิที่นิยมใช้กันอย่างกว้างขวางในงานทางด้าน Computer Graphic องค์ประกอบของมันคือ Hue, Saturation และ Value ค่าของ Hue จะไม่มีการเปลี่ยนแปลงตามความสว่างของแสงและทิศทางของกล้อง ดังนั้นจึงเหมาะสมสำหรับการค้นหารูปภาพ

ปริภูมิสีตรงข้าม

เป็นปริภูมิที่ประกอบด้วยองค์ประกอบ R-G, 2B-R-G, R+G+B ข้อดีของมันคือการแยกข้อมูลความสว่างไว้ในองค์ประกอบที่ 3 และองค์ประกอบที่ 1 และ 2 เป็นองค์ประกอบทางสี ซึ่งสามารถที่จะลดขนาดลงมาได้ เนื่องจากการรับรู้ของมนุษย์มีความไวต่อการเปลี่ยนแปลงของความสว่างมากกว่าการเปลี่ยนแปลงของสี

คุณลักษณะของสีที่นิยมใช้กันมีดังต่อไปนี้

Color Histogram

Color Histogram เป็นวิธีการแสดงรายละเอียดของสีของภาพได้อย่างมีประสิทธิภาพ เราสามารถคำนวณ Color Histogram ของภาพได้อย่างมีประสิทธิภาพและรวดเร็ว นอกจากนี้ มันยังทนทานต่อการเปลี่ยนแปลงทางการเลื่อนภาพ การหมุน และเปลี่ยนแบบซ้ำๆตามสัดส่วนของภาพ การบดบังและมูมมอ

Color Histogram คือการแจกแจงของจุดภาพในแต่ละ Bin ของ Color Space Color Histogram ที่มีจำนวน Bin มากจะทำให้มีศักยภาพสูงในการแยกแยะความแตกต่าง แต่ก็ทำให้สิ้นเปลืองเวลา ในการคำนวณ และไม่เหมาะสำหรับการสร้างตัวชี้ของฐานข้อมูลภาพที่มีประสิทธิภาพ

Color Moment

ค่า Moment อันดับต่างๆของลักษณะการกระจายของสีเช่นค่า Moment อันดับที่ 1 (ค่าเฉลี่ย : Mean) ค่า Moment อันดับที่ 2 (ค่าความแปรปรวน : Variance) และค่า Moment อันดับที่ 3 (ค่าความเบ้ : Skewness) ถูกใช้อย่างประสบความสำเร็จในระบบการค้นหาภาพหลายๆระบบ [27, 28] โดยเฉพาะอย่างยิ่งกับภาพที่มีเฉพาะวัตถุที่ต้องการค้นหาเท่านั้น

Color Moment จะใช้ได้กับปริภูมิสีชนิด $L^*u^*v^*$ และ $L^*a^*b^*$ การใช้ค่า Moment อันดับที่ 3 ร่วมกับ Moment อันดับที่ 1 และ 2 จะทำให้ประสิทธิภาพของการค้นหาภาพดีขึ้น แต่ก็อาจจะทำให้ ประสิทธิภาพของการค้นหาลดลงได้ถ้าภาพมีวัตถุหลายๆชนิดปนอยู่ด้วยกัน

เนื่องจากใช้ Color Moment มีเพียงแค่ 9 ค่าเท่านั้นในการแสดงคุณลักษณะการกระจายของสีในแต่ละแกน ทำให้มีประสิทธิภาพในการค้นหาต่ำ ดังนั้นจึงนิยมใช้ Color Moment ในการค้นหาภาพแบบ หยาบๆ ในเบื้องต้น ก่อนที่จะใช้คุณลักษณะทางสีที่มีความซับซ้อนยิ่งขึ้นเพื่อ ค้นหาภาพให้ละเอียดยิ่งขึ้นต่อไป

Color Coherence Vector

ในบทความที่ [29] ได้เสนอ Color Coherence Vector (CCV) เพื่อใช้รวมคุณสมบัติทางตำแหน่งของภาพและ Color Histogram เข้าไว้ด้วยกัน ฮิสโตแกรมบินแต่ละอันจะถูกแบ่งออกเป็น 2 ชนิดคือ ชนิด Coherent ถ้ามันเป็นสีที่อยู่ในบริเวณของภาพที่มีความสม่ำเสมอทางสี และ ชนิด Incoherent ถ้ามันไม่ได้อยู่ในบริเวณที่มีความสม่ำเสมอทางสี กำหนดให้ α_i เป็นจำนวนของพิกเซลที่มีค่าสีเท่ากับ i ที่อยู่ในบริเวณที่มีความสม่ำเสมอทางสี และ β_i เป็นจำนวนของพิกเซลที่มีค่าสีเท่ากับ i ที่อยู่ในบริเวณที่ไม่มีความสม่ำเสมอทางสี ดังนั้น CCV สามารถเขียนเป็น Vector ได้ดังนี้ $\langle (\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_N, \beta_N) \rangle$ โดยที่ $\langle (\alpha_1 + \beta_1), (\alpha_2 + \beta_2), \dots, (\alpha_N + \beta_N) \rangle$ เป็น Color Histogram ของภาพสี

เนื่องจาก CCV ได้รวมคุณสมบัติทางตำแหน่งของภาพเข้าไปด้วย จึงมีประสิทธิภาพมากกว่า Color Histogram โดยเฉพาะอย่างยิ่งเมื่อใช้กับภาพที่มีบริเวณที่มีความสม่ำเสมอของสีหรือบริเวณที่เป็นพื้นผิว (Texture) เป็นบริเวณกว้าง

2.2.2.2 พื้นผิว (Texture)

พื้นผิวเป็นคุณสมบัติที่สำคัญอย่างหนึ่งของภาพ คุณลักษณะทางพื้นผิวสามารถแบ่งออกเป็น 2 กลุ่มคือ คุณลักษณะทางโครงสร้าง และคุณลักษณะทางสถิติ คุณลักษณะทางโครงสร้างเป็นการระบุรูปแบบของโครงสร้างของพื้นผิวและลักษณะการวางพื้นผิวในภาพ เหมาะสมสำหรับภาพที่มีพื้นผิวแบบสม่ำเสมอ คุณลักษณะทางสถิติเป็นการวิเคราะห์ค่าทางสถิติของค่าความสว่างของพื้นผิว

Tamura Features

Tamura Features [30] ถูกออกแบบมาเพื่อใช้บอก ความหยาบ (Coarseness) ความแตกต่าง (Contrast) ทิศทาง (Directionality) ความเป็นเส้นตรง (Linelikeness) ความสม่ำเสมอ (Regularity) และความขรุขระ (Roughness) ของพื้นผิว

Wold Features

Wold Features [31, 32] ประกอบด้วยค่า Harmonic, Evanescent และ Indeterministic ซึ่งใช้บอก ความเป็นคาบ (Periodicity) ทิศทาง (Directionality) และความสุ่ม (Randomness) ของพื้นผิว ถ้า Harmonic มีค่าสูงแสดงว่าพื้นผิวมีความเป็นคาบค่อนข้างชัดเจน ถ้า

Evanescent มีค่าสูงแสดงว่า พื้นผิวมีทิศทางที่ชัดเจน และถ้า Indeterministic มีค่าสูงแสดงว่า พื้นผิวมีความเป็นโครงสร้างน้อย

Gabor Filter Features

Gabor Filter ถูกใช้อย่างกว้างขวางในการดึงเอาคุณสมบัติของภาพโดยเฉพาะอย่างยิ่งคุณลักษณะทางพื้นผิว [33,34] Gabor Filter เป็นตัวกรองความถี่ที่มีค่าความไม่แน่นอน (Uncertainty) ระหว่างปริภูมิและความถี่น้อยที่สุด และมันมักถูกใช้ในการตรวจจับทิศทางและขนาดของขอบภาพและเส้น

Wavelet Transform Features

Wavelet Transform Features [35, 36] เป็นวิธีการแบบหลายความละเอียด (Multi-Resolution) ที่ใช้วิเคราะห์และแบ่งแยกประเภทของพื้นผิว [37, 38] Wavelet Transform จะแยกองค์ประกอบของสัญญาณโดยใช้ฟังก์ชันพื้นฐาน (Basis Function) $\psi_{mn}(x)$ โดยการเลื่อนและการขยายขนาดของ Mother Wavelet $\psi_{jk}(x)$ นั่นคือ

$$\psi_{jk}(x) = 2^{-j/2} \psi(2^{-j}x - k) \quad (1)$$

โดยที่ j และ k เป็นค่าของความละเอียดและการเลื่อนตามลำดับ สัญญาณ $f(x)$ สามารถเขียนเป็นสมการได้ดังนี้

$$f(x) = \sum_{j,k} c_{jk} \psi_{jk}(x) \quad (2)$$

การคำนวณ Wavelet Transform ของสัญญาณ 2 มิติประกอบด้วย 2 ขั้นตอนคือการกรองความถี่แบบเวียนเกิด (Recursive Filter) และการสุ่มย่อยสัญญาณ (Sub-Sampling) ในแต่ละระดับของความละเอียด สัญญาณจะถูกแยกองค์ประกอบออกเป็น 4 ย่านความถี่ย่อยคือย่านความถี่ LL, LH, HL และ HH โดยที่ L หมายถึงย่านความถี่ต่ำ และ H หมายถึงย่านความถี่สูง มีวิธีการแปลง Wavelet อยู่ 2 วิธีที่ใช้ในการวิเคราะห์พื้นผิวคือ Pyramid-Structured Wavelet Transform (PWT) และ Tree-Structured Wavelet Transform (TWT) PWT เป็นวิธีการแยกองค์ประกอบแบบเวียนเกิดสำหรับย่านความถี่ LL แต่อย่างไรก็ตาม มีพื้นผิวบางประเภทที่ข้อมูลที่สำคัญจะปรากฏอยู่ที่ย่านความถี่กลาง ดังนั้นเพื่อแก้ปัญหานี้ วิธี TWT จะทำการแยกองค์ประกอบย่านความถี่ LH, HL หรือ HH เมื่อจำเป็น

หลังจากแยกองค์ประกอบ Feature Vector ถูกสร้างขึ้นมาโดยใช้ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของลักษณะการกระจายของพลังงานของแต่ละย่านความถี่ย่อยของแต่ละระดับ Feature Vector ของวิธี TWT จะขึ้นกับวิธีการแยกองค์ประกอบของย่านความถี่ย่อยในแต่ละระดับ

เราสามารถสร้าง Tree ที่มีการแยกองค์ประกอบแบบตายตัวได้โดยการแยกองค์ประกอบย่านความถี่ LL, LH และ HL ลงไปเรื่อยๆ ทำให้ได้ Feature Vector ที่มีขนาดเท่ากับ 52×2

1.3.2.3 รูปร่าง (Shape)

คุณลักษณะทางรูปร่างของวัตถุหรือบริเวณถูกใช้ในระบบค้นหาภาพหลายระบบ [39-42] คุณลักษณะทางรูปร่างจะต้องถูกสร้างขึ้นหลังจากที่ภาพถูกแบ่งแยกส่วน (Segment) ออกเป็นบริเวณหรือ วัตถุที่สนใจแล้ว เนื่องจากยังไม่มีวิธีการแบ่งแยกส่วนภาพที่ถูกต้อง (Accurate) และทนทาน (Robust) ดังนั้นคุณลักษณะทางรูปร่างจึงถูกจำกัดให้ใช้ในการค้นหาภาพที่ถูกคัดแยก วัตถุหรือบริเวณเป็นที่เรียบร้อยแล้วเท่านั้น คุณลักษณะทางรูปร่างสามารถแบ่งออกเป็น 2 กลุ่มคือ ตัวบอกรูปร่างที่ขึ้นกับขอบวัตถุ (Boundary Based Shape Descriptor) เช่นวิธี Rectilinear Shape [43], Polygonal Approximation [44], Finite Element Models [45] เป็นต้น และตัวบอกรูปร่างที่ขึ้นกับการแปลง Fourier (Fourier-Based Shape Descriptor) [46-48], หรือตัวบอกรูปร่างที่ขึ้นกับ บริเวณ (Moment ทางสถิติ) [49,50] คุณลักษณะทางรูปร่างที่ดีจะต้องไม่เปลี่ยนแปลงตามการเลื่อน การหมุน และการเปลี่ยนขนาด ในหัวข้อนี้เราจะพูดถึงคุณลักษณะทางรูปร่างบางคุณลักษณะที่ถูก ใช้ทั่วไปในการค้นหาภาพ

Moment Invariants

Moment Invariants เป็นวิธีการแสดงคุณลักษณะทางรูปร่างแบบดั้งเดิม ถ้าวัตถุ R ถูกแสดงโดยภาพไบนารี เราสามารถคำนวณ Moment ศูนย์กลางอันดับที่ $p+q$ ของรูปร่างของ วัตถุ R ได้จาก

$$\mu_{p,q} = \sum_{(x,y) \in R} (x - x_c)^p (y - y_c)^q \quad (3)$$

โดยที่ (x_c, y_c) เป็นจุดศูนย์กลางของวัตถุ Moment ศูนย์กลางนี้สามารถปรับให้เป็นค่ามาตรฐาน เพื่อไม่ให้มีค่าเปลี่ยนแปลงตามขนาดของวัตถุดังนี้

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^\gamma}, \gamma = \frac{p+q+2}{2} \quad (4)$$

เราสามารถคำนวณหาค่า Moment ที่ไม่ขึ้นกับการเลื่อน การหมุน และการเปลี่ยน ขนาดจากสมการที่ 3 ได้ดังนี้

$$\begin{aligned}
\varphi_1 &= \mu_{2,0} + \mu_{0,2} \\
\varphi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\
\varphi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2 \\
\varphi_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2 \\
\varphi_5 &= (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2 \right] \\
&\quad + (\mu_{0,3} - 3\mu_{2,1})(\mu_{0,3} + \mu_{2,1}) \left[(\mu_{0,3} + \mu_{2,1})^2 - 3(\mu_{3,0} + \mu_{1,2})^2 \right] \\
\varphi_6 &= (\mu_{2,0} - \mu_{0,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{0,3} + \mu_{2,1})^2 \right] \\
&\quad + 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{0,3} + \mu_{2,1}) \\
\varphi_7 &= (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{0,3} + \mu_{2,1})^2 \right]
\end{aligned} \tag{5}$$

Turning Angles

เราสามารถแสดงเส้นขอบ (Contour) ของวัตถุ 2 มิติโดยใช้อนุกรมปิดของตำแหน่ง Pixel ของขอบวัตถุ (x_s, y_s) โดยที่ $0 \leq s \leq N-1$ และ N เป็นจำนวน Pixel ทั้งหมดของขอบวัตถุ ฟังก์ชัน Turning หรือ Turning Angle $\theta(s)$ ที่ใช้วัดมุมของเส้นสัมผัสขอบวัตถุตามทิศทางการวนทวนเข็มนาฬิกา เป็นฟังก์ชันของควายาวตามแนวโค้ง (Arc-Length) สามารถคำนวณได้จาก

$$\begin{aligned}
\theta(s) &= \tan^{-1} \left(\frac{y'_s}{x'_s} \right) \\
y'_s &= \frac{dy_s}{ds} \\
x'_s &= \frac{dx_s}{ds}
\end{aligned} \tag{6}$$

ปัญหาสำคัญของ Turning Angle คือมันเปลี่ยนแปลงตามการหมุนของวัตถุและการเลือกจุดอ้างอิง ถ้าเราเลื่อนจุดอ้างอิงตามไปขอบของวัตถุเป็นจำนวน t ตำแหน่ง จะทำให้ฟังก์ชัน Turning Angle เปลี่ยนเป็น $\theta(s+t)$ และถ้าเราหมุนวัตถุไปด้วยมุม ω จะทำให้ฟังก์ชันใหม่มีค่าเป็น $\theta(s) + \omega$

ดังนั้นการเปรียบเทียบความเหมือนระหว่างวัตถุ A และ B ทำได้โดยหาค่าระยะทางต่ำสุดสำหรับค่า t และ ω ที่เป็นไปได้ทั้งหมด

$$d_p(A, B) = \left(\min_{\omega \in R, t \in [0,1]} \int_0^1 |\theta_A(s+t) - \theta_B(B) + \omega|^p ds \right)^{\frac{1}{p}} \tag{7}$$

Fourier Descriptors

Fourier Descriptor เป็นวิธีที่ใช้แสดงรูปร่างของวัตถุโดยการแปลง Fourier ของอนุกรมปิดของตำแหน่ง Pixel ของวัตถุ (x_s, y_s) โดยที่ $0 \leq s \leq N-1$ และ N เป็นจำนวน Pixel ทั้งหมดของขอบวัตถุ เพื่อให้คุณลักษณะของรูปร่างของทุกวัตถุในฐานข้อมูลมีขนาดเท่ากัน เราอาจจะทำการสุ่มอนุกรมปิดของขอบวัตถุใหม่ให้มีความยาวเท่ากันทั้งหมดเท่ากับ M โดยกำหนดให้ $M = 2^m$ เพื่อให้การคำนวณ Fast Fourier Transform ได้

2.2.3 การวัดความเหมือน

แทนที่จะใช้การเปรียบเทียบโดยตรง ระบบการค้นหภาพทางการแพทย์ใช้วิธีการวัดความเหมือนทางสายตา (Visual Similarity) ระหว่างภาพที่ถูกค้นกับภาพที่อยู่ในฐานข้อมูลภาพ จากนั้นจึงเรียงลำดับภาพในฐานข้อมูลตามค่าความเหมือนที่คำนวณได้ วิธีการวัดความเหมือนของภาพหลายวิธีถูกพัฒนาขึ้นมาเพื่อใช้กับระบบการค้นหารูปภาพ วิธีการวัดที่แตกต่างกันจะทำให้ประสิทธิภาพของระบบการค้นหามีความแตกต่างกันอย่างเห็นได้ชัด ในหัวข้อนี้จะแนะนำวิธีการวัดความเหมือนที่นิยมใช้กันบางวิธี โดยกำหนดให้ $D(I, J)$ เป็นระยะทางระหว่างภาพ I ที่ต้องการค้นหา และภาพ J ที่อยู่ในฐานข้อมูล และ $f_i(I)$ เท่ากับจำนวนจุดภาพที่อยู่ใน Bin ที่ i ของภาพ I

2.2.3.1 Minkowski-Form Distance

Minkowski-Form Distance เหมาะสมสำหรับการคำนวณระยะทางระหว่างภาพ 2 ภาพ ถ้าแต่ละมิติของ Feature Vector ของภาพไม่ขึ้นต่อกันและมีความสำคัญเท่ากัน เราสามารถคำนวณ Minkowski-Form Distance โดย

$$D(I, J) = \left(\sum_i |f_i(I) - f_i(J)|^p \right)^{\frac{1}{p}} \quad (8)$$

ถ้า $p=1, 2$ และ ∞ เราสามารถ $D(I, J)$ แบบย่อๆเป็น L_1, L_2 (หรือเรียกว่า Euclidean Distance) และ L_∞ Minkowski-Form Distance เป็นวิธีการวัดความเหมือนที่นิยมใช้กันมากในระบบการค้นหารูปภาพ ตัวอย่างเช่น ระบบ MARS [13] ใช้ Euclidean Distance ในการวัดความเหมือนระหว่างคุณลักษณะทางพื้นผิว ระบบ Netra [51,52] ใช้ Euclidean Distance ในการวัดความเหมือนของคุณลักษณะทางสีและรูปร่าง และใช้ L_1 ในการวัดคุณลักษณะทางพื้นผิว ระบบ Blobworld [53] ใช้ Euclidean Distance ในการวัดความเหมือน ของคุณลักษณะทางพื้นผิวและรูปร่าง นอกจากนี้ในบทความที่ [54] ใช้ L_∞ ในการคำนวณความเหมือนของคุณลักษณะทางพื้นผิว

Histogram Intersection เป็นกรณีพิเศษของ L_1 ซึ่งถูกพัฒนาโดย Swain และ Ballard [25] เพื่อใช้คำนวณความเหมือนระหว่างภาพ 2 ภาพ Histogram Intersection สามารถคำนวณโดย

$$S(I, J) = \frac{\sum_{i=1}^N \min(f_i(I), f_i(J))}{\sum_{i=1}^N f_i(J)} \quad (9)$$

Histogram Intersection ได้แสดงให้เห็นว่ามันไม่ไวต่อการเปลี่ยนความละเอียดของภาพ, ขนาดของ Histogram, การบิดเบือน, ความลึกและมุมมอง

2.2.3.2 Quadratic Form (QF) Distance

Minkowski Distance กำหนดให้ทุก Bin ของ Histogram มีความสำคัญเท่ากัน โดยไม่สนใจความจริงที่ว่า Bin บางอันมีความสำคัญมากกว่าอันอื่น Quadratic Form Distance ได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหานี้

$$D(I, J) = \sqrt{(F_I - F_J)^T A (F_I - F_J)} \quad (10)$$

โดยที่ $A = [a_{ij}]$ เป็นเมทริกซ์ของความเหมือน และ a_{ij} หมายถึงความเหมือนระหว่าง Bin i และ Bin j F_I และ F_J เป็นเวกเตอร์ของข้อมูลทั้งหมดใน $f_i(I)$ และ $f_i(J)$

QF Distance ถูกใช้ในระบบค้นหภาพที่ใช้ Color Histogram [28] QF Distance ให้ผลลัพธ์การค้นหาคือดีกว่า Euclidean Distance และ Histogram Intersection เนื่องจากมีการนำเอาความเหมือนกันระหว่างแต่ละสีมาพิจารณาด้วย

2.2.3.3 Mahalanobis Distance

Mahalanobis Distance เหมาะสมกับการเปรียบเทียบเมื่อแต่ละมิติของ Feature Vector ไม่เป็นอิสระต่อกันและมีค่าความสำคัญไม่เท่ากัน Mahalanobis Distance สามารถคำนวณได้ดังนี้

$$D(I, J) = \sqrt{(F_I - F_J)^T C^{-1} (F_I - F_J)} \quad (11)$$

โดยที่ C เป็น Covariance Matrix ของ Feature Vector

ในกรณีที่แต่ละมิติของ Feature Vector เป็นอิสระต่อกันเราสามารถเขียนสมการ Mahalanobis Distance ได้ดังนี้

$$D(I, J) = \sum_{i=1}^N \frac{(F_I - F_J)^2}{C_i} \quad (12)$$

2.2.3.4 Kullback-Leibler (KL) Divergence และ Jeffrey-Divergence (JD)

KL Divergence เป็นการวัดค่าความกระชับ (Compact) ของลักษณะการกระจายของคุณลักษณะที่ถูกเข้ารหัสโดย Codebook ของอีกคุณลักษณะหนึ่ง KL Divergence ระหว่างภาพ I และ J สามารถคำนวณได้จาก

$$D(I, J) = \sum_i f_i(I) \log \frac{f_i(I)}{f_i(J)} \quad (13)$$

KL Divergence ถูกใช้ในบทความที่ [8] เพื่อวัดความเหมือนของคุณลักษณะทางพื้นผิว

Jeffrey-Divergence (JD) สามารถคำนวณได้จาก

$$D(I, J) = \sum_i f_i(I) \log \frac{f_i(I)}{\hat{f}_i} + f_i(J) \log \frac{f_i(J)}{\hat{f}_i} \quad (14)$$

โดยที่ $\hat{f}_i = [f_i(I) + f_i(J)] / 2$

2.2.3.5 χ^2 Statistic

χ^2 Statistic เป็นดัชนีทางสถิติที่ใช้วัดว่าลักษณะการกระจายของคุณลักษณะอันหนึ่งไม่เหมือนกับลักษณะการกระจายของคุณลักษณะอีกอันหนึ่ง [58, 59] ซึ่งสามารถคำนวณได้ดังนี้

$$D(I, J) = \sum_i \frac{f_i(I) - \hat{f}_i}{\hat{f}_i} \quad (15)$$

2.2.3.6 Earth's Mover Distance (EMD)

EMD เป็นการวัดความแตกต่างระหว่างลักษณะการกระจาย 2 อัน สามารถคำนวณได้จาก

$$D(I, J) = \frac{\sum_{i,j} f_{ij} d_{ij}}{\sum_{i,j} f_{ij}} \quad (16)$$

โดยที่ d_{ij} เป็นค่าความแตกต่างระหว่างคุณลักษณะที่ i และ j และ $f_{ij} > 0$ เป็นค่าที่ทำให้ $\sum_{i,j} f_{ij} d_{ij}$ มีค่าน้อยที่สุดภายใต้บางเงื่อนไข เราสามารถหาค่า f_{ij} ที่ดีที่สุดโดยใช้การแก้ปัญหาค่าเหมาะที่สุด

2.2.4 การประเมินประสิทธิภาพ

ในงานวิจัยนี้ใช้ค่า Precision ในการประเมินประสิทธิภาพของระบบการค้นหารูปภาพทางการแพทย์ กำหนดให้ q เป็นภาพของผู้ป่วย กำหนดให้ $R(q)$ เป็นเซตของภาพทางการแพทย์ในฐานะข้อมูลที่เกี่ยวข้องกับภาพ q และผลลัพธ์ของการค้นหภาพ q กำหนดให้เป็น $Q(q)$ ค่า Precision คือสัดส่วน ระหว่างจำนวนภาพที่เกี่ยวข้องกับภาพ q กับภาพผลลัพธ์ทั้งหมด

$$precision = \frac{Q(q) \cap R(q)}{|Q(q)|} \quad (17)$$

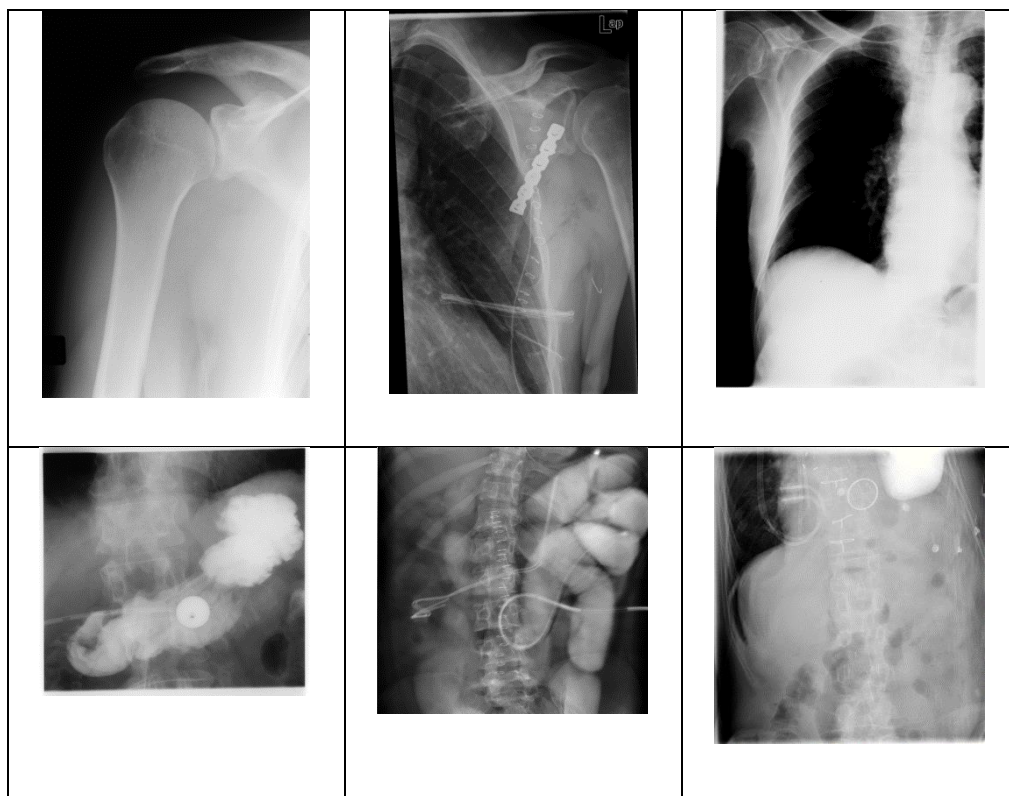
2.3 ระบบการค้นหภาพทางการแพทย์ด้วยรายละเอียดของภาพ CBMIRS (Content-Based Medical Image Retrieval System)

งานวิจัยนี้มีจุดประสงค์ที่จะออกแบบระบบการค้นหภาพ X-Ray ทางทางการแพทย์ที่ใช้ในการวินิจฉัยโรค ตารางที่ 2.1 แสดงตัวอย่างของภาพที่ใช้ในงานวิจัยนี้ จากการที่คุณลักษณะทางรูปร่างยังไม่สามารถนำมาใช้งานได้จริงในทางปฏิบัติเนื่องจากยังไม่มีวิธีในการแบ่งแยกส่วนภาพที่มีประสิทธิภาพดีพอที่จะใช้กับภาพ X-Ray ดังนั้นในงานวิจัยนี้จึงเลือกที่จะใช้คุณลักษณะทางพื้นผิวและได้เสนอวิธีการสร้างคุณลักษณะทางพื้นผิวจากการแปลง Wavelet ที่มีประสิทธิภาพสำหรับใช้ในงานค้นหภาพ

ในบทความที่ [55] Theoharatos และผู้ร่วมวิจัยได้ใช้วิธีการวัดความเหมือนที่นำเสนอโดย Friedman และ Rafsky [56] มาวัดความเหมือนของลักษณะการกระจายของสี วิธีของ Friedman และ Rafsky เป็นการดัดแปลงวิธีทดสอบรันของ Wald และ Wolfowitz [67] ให้สามารถใช้งานได้กับข้อมูลที่เป็นหลายมิติ การทดลองของ Theoharatos และผู้ร่วมงานแสดงให้เห็นว่าวิธีการของ Friedman และ Rafsky ให้ผลลัพธ์ที่ดีกว่าวิธี Histogram Intersection, Kullback-Leibler Divergence, Jeffrey Divergence, χ^2 Statistic และ Earth Mover Distance จากนั้นผู้วิจัยได้เสนอวิธีการวัดความเหมือน [57] ซึ่งได้ปรับปรุงมาจากวิธีของ Friedman และ Rafsky ไว้ 2 วิธีคือ Centroid-Based MWW Runs Test และ Weighted MWW Runs Test จากผลการทดลองผู้วิจัยได้แสดงให้เห็นว่าวิธีการวัดทั้ง 2 วิธีให้ผลลัพธ์ของการค้นหาที่ดีกว่าและใช้เวลาในการคำนวณที่เร็วกว่าวิธีการที่เสนอโดย Theoharatos โดยที่วิธี Weighted MWW Runs Test ให้ผลลัพธ์ที่ดีที่สุด ดังนั้นในงานวิจัยนี้

ผู้วิจัยจึงได้เสนอที่จะใช้วิธี Weighted MWW Runs Test ในการวัดความเหมือนของคุณลักษณะ ทางพื้นผิว

ตารางที่ 2.1 ตัวอย่างภาพ X-Ray ทางการแพทย์ที่ใช้ระบบค้นหาภาพทางการแพทย์ด้วย รายละเอียดของภาพ



บทที่ 3

ทฤษฎีพื้นฐาน

บทนี้จะอธิบายทฤษฎีพื้นฐานที่จำเป็นต้องใช้ในการออกแบบระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ ซึ่งประกอบด้วย ทฤษฎีการแปลง Wavelet, ทฤษฎีกราฟ, ต้นไม้แผ่ทั่วที่น้อยที่สุด (Minimal Spanning Tree) และ ทฤษฎีการทดสอบรูปแบบหลายมิติของ Friedman และ Rafsky

3.1 ทฤษฎีการแปลง Wavelet

พื้นผิว (Texture) เป็นเครื่องมือสำคัญในการวิเคราะห์และการหาเอกลักษณ์ของวัตถุที่อยู่ในภาพ [60] การวิเคราะห์พื้นผิวเป็นองค์ประกอบหลักของการประมวลผลภาพ และเป็นพื้นฐานของการประยุกต์ใช้งานหลายประเภทเช่น ภาพถ่ายดาวเทียม, ตรวจสอบคุณภาพ, และภาพทางการแพทย์ เป็นต้น เร็ว ๆ นี้ตัวกรองความถี่แบบหลายความละเอียด (Multiscale Filter) ได้แสดงให้เห็นถึงศักยภาพในการเป็นตัวบอกลักษณะของพื้นผิว (Texture Description) เนื่องจากตัวกรองความถี่แบบนี้มีค่าพลังงานจำกัดวงสูงสุดทั้งในโดเมนตำแหน่ง (Spatial Domain) และโดเมนของความถี่ [61]

การแปลง Wavelet ถูกใช้ในการวิเคราะห์แบบหลายความละเอียดสำหรับตัวบอกลักษณะพื้นผิวเป็นครั้งแรกในบทความที่ [62, 63] การพัฒนาเมื่อเร็ว ๆ นี้ทำให้การแปลง Wavelet เป็นเครื่องมือที่ดีในการใช้วิเคราะห์พื้นผิวให้ได้ผลลัพธ์ที่แม่นยำ

3.1.1 นิยามของ Subspace และ Scaling Function

การแปลง Wavelet มีที่มาจากทฤษฎีการวิเคราะห์แบบหลายความละเอียด (Multiresolution Analysis : MRA) ถ้าเรามีฟังก์ชัน $f \in L_2(\mathbb{R})$ จากทฤษฎีการวิเคราะห์แบบหลายความละเอียดของ $L_2(\mathbb{R})$ จะทำให้เราได้อนุกรมของปริภูมิ $V_j, V_{j+1}, \dots$ ที่เมื่อทำการฉาย (Projection) ฟังก์ชัน f ลงไปในปริภูมิเหล่านี้แล้วจะทำให้ได้ฟังก์ชันการประมาณของฟังก์ชัน f ที่มีความละเอียดมากยิ่งขึ้นเมื่อ $j \rightarrow \infty$

MRA ของ $L_2(\mathbb{R})$ ถูกนิยามว่าเป็นอนุกรมของปริภูมีย่อยแบบปิด $V_j \subset L_2(\mathbb{R})$ โดยที่ $j \in \mathbb{Z}$ ที่มีคุณสมบัติดังต่อไปนี้

1. $\dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots$

2. ปริภูมิ V_j จะต้องมีความสัมพันธ์ดังนี้

$$\bigcup_{j \in \mathbb{Z}} V_j \text{ ต้องหนาแน่น (Dense) ใน } L_2(\mathbb{R}) \text{ และ } \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

3. ถ้า $f(x) \in V_0$ จะทำให้ $f(2^j x) \in V_j$ นั่นคือปริภูมิ V_j ถูกเปลี่ยนความละเอียดโดยมี V_0 ปริภูมิศูนย์กลาง

4. ถ้า $f \in V_0$ จะทำให้ $f(-k) \in V_0$ โดยที่ $k \in \mathbb{Z}$ นั่นคือปริภูมิ V_0 และปริภูมิ V_j ทั้งหมดไม่มีการเปลี่ยนแปลงตามการเลื่อนตำแหน่ง

5. มีฟังก์ชัน $\varphi \in V_0$ ที่ทำให้เซต $\{\varphi(x-k); k \in \mathbb{Z}\}$ เป็น Riesz Basis ในปริภูมิ V_0

เราจะเรียกระดับของ MRA ว่าเป็นหนึ่งในปริภูมีย่อย V_j สำหรับ j ที่ตายตัว เซตของ $\{\varphi_{jk}(x) = 2^{j/2} \varphi(2^j x - k); k \in \mathbb{Z}\}$ ของการขยายขนาดและการเลื่อนของ φ เป็น Riesz Basis ของ V_j เนื่องจาก $\varphi \in V_0 \subset V_1$ เราสามารถเขียน φ ในรูปของการรวมกันเชิงเส้นของเซต $\{\varphi_{1,k}\}$ ดังนี้

$$\varphi(x) = \sum_{k \in \mathbb{Z}} h_k \varphi_{1,k}(x) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \varphi(2x - k) \quad (18)$$

เราเรียกสมการที่ (18) ว่า Two-Scale Equation หรือ Refinement Equation มันเป็นสมการพื้นฐานของ MRA เนื่องจากมันใช้ในการย้ายจากระดับที่มีความละเอียดสูง V_1 ไปยังระดับที่มีความละเอียดต่ำกว่า V_0 ฟังก์ชัน φ เรียกว่าฟังก์ชัน Scaling

จากที่ได้พูดไว้ก่อนหน้านี้ว่าปริภูมิ V_j ถูกใช้ในการประมาณค่าของฟังก์ชัน การประมาณค่าสามารถทำได้โดยการฉายฟังก์ชันอย่างเหมาะสมลงในปริภูมิเหล่านี้ เนื่องจากยูเนียนของ V_j ทั้งหมดจะหนาแน่นอยู่ใน $L_2(\mathbb{R})$ ดังนั้นเราสามารถรับประกันได้ว่าสำหรับทุกฟังก์ชันที่เป็นสมาชิกของ $L_2(\mathbb{R})$ สามารถถูกประมาณค่าได้โดยการฉายดังกล่าว ตัวอย่างเช่นกำหนดให้ปริภูมิ V_j เป็นเซตของ

$$V_j = \left\{ f \in L_2(\mathbb{R}); \forall k \in \mathbb{Z}, f \Big|_{[2^{-j}k, 2^{-j}(k+1)]} = \text{constant} \right\} \quad (19)$$

ดังนั้นฟังก์ชัน Scaling $\varphi(x) = 1_{[0,1)}(x)$ ที่เรียกว่าฟังก์ชัน Haar Scaling สามารถถูกสร้างโดยการเลื่อนและขยายขนาด MRA สำหรับอนุกรมของปริภูมิ $\{V_j, j \in \mathbb{Z}\}$ ซึ่งถูกนิยามในสมการที่ (19) ดู [64, 65]

3.1.2 ปริภูมิรายละเอียดและฟังก์ชัน Wavelet

แทนที่จะพิจารณาปริภูมิ V_j ทั้งหมดที่ซ้อนกัน เราสามารถเพิ่มประสิทธิภาพให้มากขึ้นโดยการพิจารณาเพียงข้อมูลที่สำคัญสำหรับการเปลี่ยนจากปริภูมิ V_j ไปยัง V_{j+1} ดังนั้นให้เราพิจารณาปริภูมิ W_j ที่เป็นปริภูมิ Complement ของ V_j ใน V_{j+1}

$$V_{j+1} = V_j \oplus W_j \quad (20)$$

ปริภูมิ W_j ไม่จำเป็นต้องตั้งฉาก (Orthogonal) กับปริภูมิ V_j แต่มันจะเก็บข้อมูลรายละเอียดที่จำเป็นในการย้ายจากฟังก์ชันการประมาณในปริภูมิที่ j ไปยังฟังก์ชันการประมาณในปริภูมิ $j+1$ ที่มีความละเอียดมากกว่า ดังนั้นถ้าเราใช้สมการที่ (20) เพื่อย้ายฟังก์ชันประมาณค่าที่ปริภูมิความละเอียดต่ำ $j_0 \in \mathbb{Z}$ ไปยัง ปริภูมิที่มีความละเอียดสูงขึ้นเรื่อย ๆ เราจะได้

$$L_2(\mathbb{R}) = \overline{V_{j_0} \oplus \bigoplus_{j=j_0}^{\infty} W_j} \quad (21)$$

โดยกำหนดให้ $W_{j_0-1} := V_{j_0}$ เราเรียกอนุกรม $\{W_j\}_{j \geq j_0-1}$ ว่า Multiscale Decomposition (MSD)

เราเรียก ψ ว่าเป็นฟังก์ชัน Wavelet ถ้าเซตของ $\{\psi(x-k); k \in \mathbb{Z}\}$ เป็น Riesz Basis ของ W_0 เนื่องจาก $W_0 \subset V_1$ ดังนั้นจึงมี Refinement Equation สำหรับ ψ ที่เหมือนกับสมการที่ (18) ดังนี้

$$\psi(x) = \sqrt{2} \sum_k g_k \phi(2x-k) \quad (22)$$

ดังนั้นเซตของฟังก์ชัน Wavelet $\{\psi_{jk} = 2^{j/2} \psi(2^j x - k); k \in \mathbb{Z}\}$ จึงเป็น Riesz Basis ของ $L_2(\mathbb{R})$ หนึ่งในคุณลักษณะที่สำคัญของฟังก์ชัน Wavelet ก็คือมันมีคุณสมบัติ Vanishing Moment ที่ค่าหนึ่ง นั่นคือฟังก์ชัน Wavelet มี Vanishing Moment ที่ N ถ้า $\int \psi(x) x^p dx = 0, p = 0, \dots, N-1$

3.1.3 Orthogonal Base Wavelet

ใน Orthogonal Multiresolution Analysis ปริภูมิ W_j จะต้องเป็น Orthogonal Complement ของ V_j ใน V_{j+1} เราสามารถเขียนสมการของฟังก์ชัน $f(x)$ ได้ดังต่อไปนี้

$$f(x) = P_{j_0}(f) + \sum_{j=j_0}^{\infty} Q_j(f) \quad (23)$$

โดยที่ $P_{j_0}(f)$ เป็นฟังก์ชันการประมาณของ f ในปริภูมิ V_{j_0} และ $Q_j(f)$ เป็นฟังก์ชันรายละเอียดของ f ในปริภูมิ V_j ซึ่งสามารถคำนวณได้จาก

$$P_{j_0}(f) = \sum_k \left(2^{j_0/2} \phi(2^{j_0} x - k) \int f(x) 2^{j_0/2} \phi(2^{j_0} x - k) dx \right) \quad (24)$$

$$Q_j(f) = \sum_k \left(2^{j/2} \psi(2^j x - k) \int f(x) 2^{j/2} \psi(2^j x - k) dx \right) \quad (25)$$

จากสมการที่ (23-25) เราสามารถเขียนฟังก์ชัน $f(x)$ โดยการประมาณค่าของ $f(x)$ ที่ความละเอียด j_0 โดยใช้ฟังก์ชัน $P_{j_0}(f)$ จากนั้นจึงค่อยๆเพิ่มความละเอียดที่ระดับ $j_0+1, j_0+2, \dots$ ไปเรื่อยๆ โดยการบวกฟังก์ชันรายละเอียด $Q_j(f)$ เข้าไปเรื่อย ๆ จนถึงระดับที่เราต้องการ

Orthogonal Wavelet ที่สำคัญคือ Binomial QMF, Coiflet, Haar และ Daubechies Wavelet เป็นต้น

3.1.4 การแปลง Wavelet แบบเร็ว

สมมติว่าเรามีสัญญาณที่ถูกขยายลงในปริภูมิ V_{j+1} ซึ่งสามารถคำนวณได้จาก

$$P_{j+1}(f) = \sum_k s_{j+1,k} \varphi_{j+1,k}(x) \quad (26)$$

โดยที่ $s_{j+1,k} = \int f(x) 2^{(j+1)/2} \varphi(2^{j+1}x - k) dx$ จากสมการ Dual Refinement เราจะได้ว่า

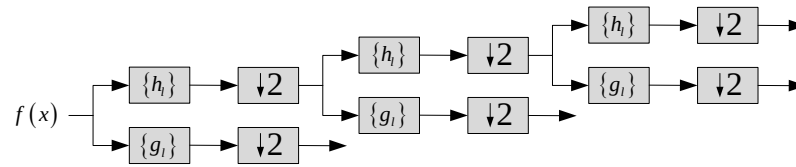
$$\begin{aligned} s_{j,k} &= \int f(x) 2^{j/2} \varphi(2^j x - k) dx \\ &= \int f(x) \left[\sum_l h_l 2^{(j+1)/2} \varphi(2^{j+1}x - 2k - l) \right] dx \\ &= \sum_k h_{l-2k} s_{j+1,l} \end{aligned} \quad (27)$$

โดยที่สัมประสิทธิ์ $s_{j,k}$ ถูกเรียกว่าสัมประสิทธิ์ Scaling เนื่องจากมันเกี่ยวข้องกับฟังก์ชัน Scaling ในทำนองเดียวกัน เราสามารถคำนวณหาสัมประสิทธิ์ Wavelet $d_{j,k}$ ได้จาก

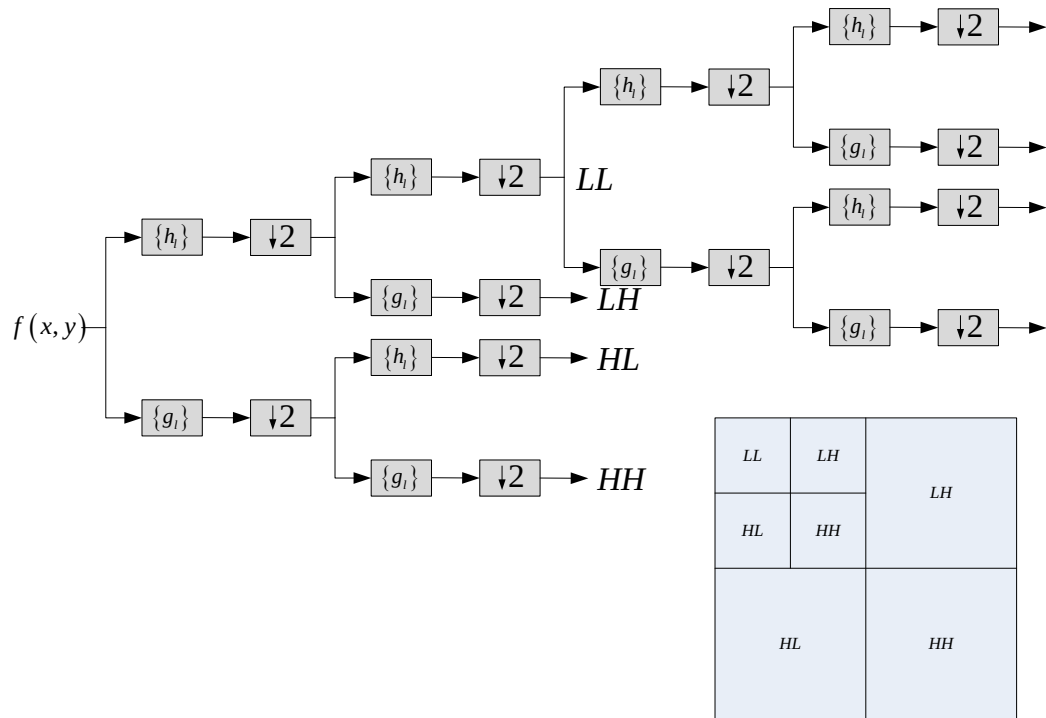
$$d_{j,k} = \int f(x) 2^{j/2} \psi(2^j x - k) dx = \sum_k g_{l-2k} s_{j+1,l} \quad (28)$$

จากสมการที่ (27) และ (28) เราสามารถหาสัมประสิทธิ์ $s_{j,k}$ และ $d_{j,k}$ ได้โดยการกรองความถี่สัมประสิทธิ์ $s_{j+1,l}$ ด้วยตัวกรองความถี่ $\{h_l\}$ และ $\{g_l\}$ และเก็บเอาเฉพาะสัมประสิทธิ์ $s_{j,k}$ และ $d_{j,k}$ ที่มีค่า k เป็นเลขคู่เท่านั้น หรือการทำ Down Sampling

เราสามารถแยกองค์ประกอบของสัญญาณ $f(x)$ เพื่อหาสัมประสิทธิ์ Scaling และ Wavelet ที่ความละเอียด j ใด ๆ โดยการกรองสัญญาณ $f(x)$ ด้วยตัวกรองความถี่ $\{h_l\}$ และ $\{g_l\}$ จากนั้นก็ทำการ Down Sampling และทำแบบเดิมต่อไปเรื่อย ๆ กับสัมประสิทธิ์ที่ได้จาก $\{h_l\}$ ดังแสดงในภาพที่ 3.1



ภาพที่ 3.1 ผังการแปลง Wavelet 1 มิติแบบเร็ว



ภาพที่ 3.2 ผังการแปลง Wavelet 2 มิติแบบเร็ว

เราสามารถประยุกต์การแปลง Wavelet 1 มิติ ให้เป็นการแปลง Wavelet 2 มิติ โดยการแปลง Wavelet 1 มิติในแกน x ก่อนจากนั้นจึงเปลี่ยนไปแปลง Wavelet 1 มิติในแกน y จากนั้นจึงทำแบบเดิมในแกนความถี่ LL ดังแสดงในภาพที่ 3.2

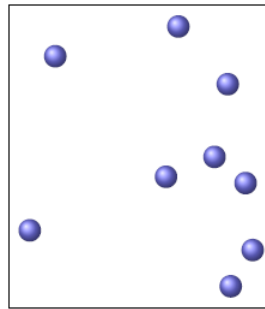
3.2 ทฤษฎีกราฟและต้นไม้แผ่ทั่วที่น้อยที่สุด

กำหนดให้ $V = \{v_i | v_i \in \mathbf{R}^d\}_{i=1}^n$ โดยที่ $d \geq 2$ เป็นเซตของจุดในปริภูมิเวกเตอร์หลายมิติ กราฟของเซต V คือคู่ $G = (V, E)$ ของเซตโดยที่ $E \subseteq \{(v_i, v_j) | v_i, v_j \in V\}_{i,j=1}^n$ สมาชิกของ V และ E เรียกว่า Vertex และ Edge ตามลำดับ Vertex v_i และ v_j จะถูกเรียกว่า Adjacent หรือ Neighbor ซึ่งกันและกันถ้า Edge (v_i, v_j) เป็นสมาชิกของ G Degree D_i ของ v_i คือจำนวน Edge ที่ต่อกับ v_i กราฟย่อย (Subgraph) ของ $G = (V, E)$ คือกราฟ $H = (V', E')$ โดยที่ $V' \subset V$ และ $E' \subset E$ ทางเดิน (Walk) ที่มีความยาวเท่ากับ k ของกราฟ G คืออนุกรมไม่ว่างของการสลับ Vertex

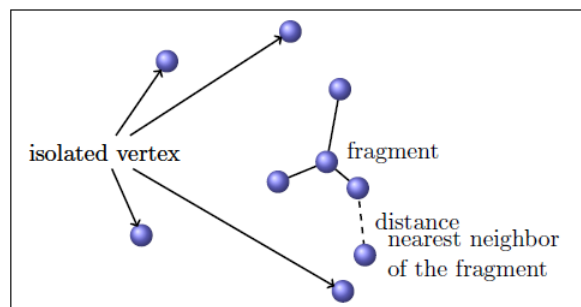
และ Edge $v_0e_0v_1e_1\cdots e_{k-1}v_k$ ของ G โดยที่ $e_i=(v_i,v_{i+1})$ สำหรับทุกๆ $i < k$ เส้นทาง (Path) ของ G คือทางเดินที่ไม่มี Vertex ซ้ำกัน เส้นทาง $v_0e_0v_1e_1\cdots e_{k-1}v_k$ เป็นรอบ (Cycle) ถ้า $v_0 = v_k$ กราฟไม่ว่า G เป็นกราฟเชื่อมต่อ (Connected) หมายถึงกราฟที่ Vertex ทุกคู่ของมันถูกเชื่อมด้วยเส้นทางใดเส้นทางหนึ่งของ G ต้นไม้ (Tree) คือกราฟเชื่อมต่อที่ไม่มีรอบ ต้นไม้แผ่ทั่วที่ (Spanning Tree) หมายถึงต้นไม้ที่ประกอบด้วย Vertex ทุกอันของ G และต้นไม้แผ่ทั่วที่น้อยที่สุด (Minimal Spanning Tree) หมายถึงต้นไม้แผ่ทั่วที่ที่ผลรวมของความยาวของ Edge ของมันมีค่าน้อยที่สุด

$$\sum_{e \in T} |e| = \min_{T'} \left\{ \sum_{e \in T'} |e| \right\} \quad (29)$$

โดยที่ T' คือต้นไม้แผ่ทั่วใดๆของ V และ $|e| = \|v_i - v_j\|$ เป็นความยาวของ $e = (v_i, v_j)$



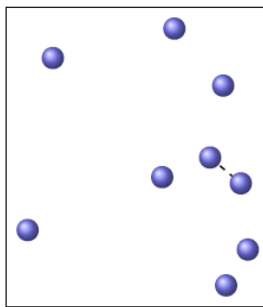
ก. Vertex ของต้นไม้แผ่ทั่วที่น้อยที่สุด



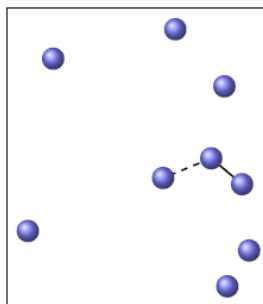
ข. Isolated Vertex, Fragment, ระยะทาง, และ Nearest Neighbor ของ Fragment

ภาพที่ 3.3 ตัวอย่างของเครื่องมือที่ใช้ในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดที่นิยามโดย

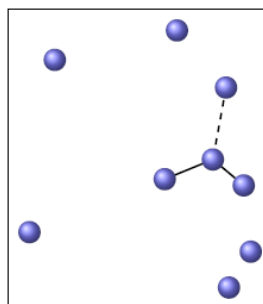
Prim



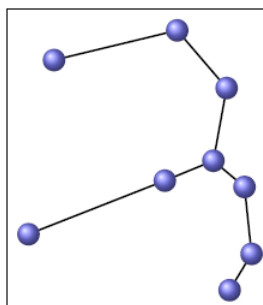
ก. Isolated Vertex เริ่มต้น และ Nearest Neighbor ของมัน



ข. Fragment เริ่มต้นและ Nearest Neighbor ของมัน



ค. Fragment ที่สองและ Nearest Neighbor ของมัน



ง. ต้นไม้แผ่ทั่วที่น้อยที่สุด

ภาพที่ 3.4 ตัวอย่างการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดโดยใช้วิธีของ Prim

Prim ได้เสนอวิธีในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดจากเซต V และได้นิยามเครื่องมือต่างๆที่ใช้ในการสร้างดังต่อไปนี้ Isolated Vertex หมายถึง Vertex ที่ไม่ได้เชื่อมต่อกับ Vertex อื่นๆใน

ระหว่างกระบวนการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุด Fragment หมายถึงกราฟเชื่อมต่อที่สร้างจากเซตของ V ระยะทาง (distance) จาก Vertex ถึง Fragment หมายถึงระยะทางที่สั้นที่สุดจาก Vertex ที่กำลังพิจารณาไปยัง Vertex ใดๆใน Fragment และ Nearest Neighbor หมายถึง Vertex ที่อยู่ใกล้ Fragment มากที่สุด ตัวอย่างของเครื่องมือในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดถูกแสดงไว้ในภาพที่ 3.3

วิธีการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดมีดังนี้

- เลือก Isolated Vertex เริ่มต้นและค้นหา Nearest Neighbor ของมันเพื่อสร้าง Fragment เริ่มต้น

- สำหรับ $i = 1, \dots, N - 2$

เชื่อมต่อ Fragment ที่ i กับ Nearest Neighbor ของมัน

ตัวอย่างการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดโดยใช้วิธีของ Prim ถูกแสดงไว้ในภาพที่ 3.4

3.3 ทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz

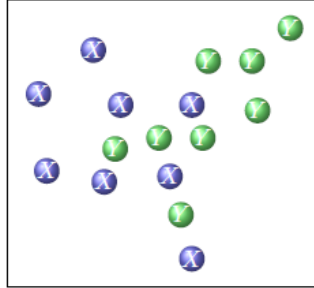
มีการใช้งานการทดสอบรันของ Wald และ Wolfowitz ในงานวิจัยทางด้านสถิติเป็นเวลานานก่อนที่จะถูกประยุกต์ในการค้นหารูปภาพโดยรายละเอียดของภาพ Wald และ Wolfowitz ได้พัฒนาทดสอบรันของ พวกเขาในปี 1940 เพื่อใช้เปรียบเทียบความเหมือนระหว่างเซตของจุด 2 เซต [67] ผลลัพธ์ของการทดสอบรันอยู่ในรูปของค่าความน่าจะเป็นที่เซต ของจุดทั้ง 2 เซตถูกสร้างมาจากการแจกแจงแบบเดียวกัน จากนั้นในปี 1979 Friedman และ Rafsky ได้พัฒนาทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz (Multidimensional Generalization of Wald and Wolfowitz Runs Test) [56] ทฤษฎีของการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz อยู่บนพื้นฐานของการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุด (Minimal Spanning Tree) ทฤษฎีของการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz เป็นดังนี้

3.3.1 นิยามของการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz

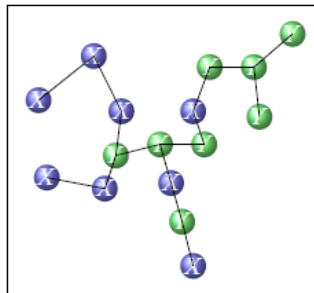
กำหนดให้ $X_1, X_2, \dots, X_m$ เป็นจุดที่เป็นอิสระต่อกันและกันและมีการแจกแจงที่เหมือนกันในปริภูมิ $\mathbf{R}^d$ โดยมีฟังก์ชันการแจกแจงเป็น f และ $Y_1, Y_2, \dots, Y_n$ เป็นจุดที่เป็นอิสระต่อกันและกันและมีการแจกแจงที่เหมือนกันในปริภูมิ $\mathbf{R}^d$ โดยมีฟังก์ชันการแจกแจงเป็น g Friedman และ Rafsky ได้พัฒนาทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz เพื่อใช้ทดสอบสมมติฐานหลัก $H_0 : f = g$ ซึ่งมีทฤษฎีดังนี้

กำหนดให้ T เป็นต้นไม้แผ่ทั่วที่น้อยที่สุดของเซตยูเนียนของเซต $\mathbf{X} = \{X_i\}_{i=1}^m$ และ $\mathbf{Y} = \{Y_i\}_{i=1}^n$ เราเรียก Edge ที่เชื่อมต่อ Vertex ที่ได้จากเซตที่ต่างกันว่า Inter-Set Edge Friedman และ Rafsky นิยามค่าผลลัพธ์ R ของทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz คือจำนวนต้นไม้ที่แยกจากกันที่ได้จากการลบ Inter-Set Edge ออกทั้งหมด

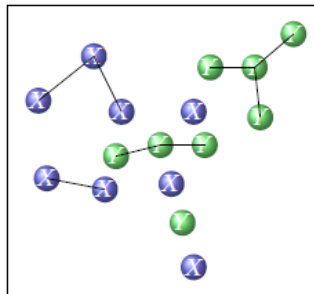
เนื่องจาก T เป็นต้นไม้ที่ไม่มีรอบ ดังนั้นการลบ Edge ออกจากต้นไม้ 1 ต้นจะทำให้ต้นไม้ถูกแยกจากกันเป็น 2 ต้น (ดูทฤษฎีใน [66]) ดังนั้นจำนวนต้นไม้ที่แยกจากกันจะเท่ากับจำนวน Inter-Set Edge + 1 เราสามารถคำนวณค่า R ได้จาก



ก. Vertex ของยูเนียนเซต $X \cup Y$



ข. กราฟต้นไม้แผ่ทั่วที่น้อยที่สุด



ค. ต้นไม้ที่แยกจากกันจำนวน 9 ต้นที่ได้จากการลบ Inter-Set Edge

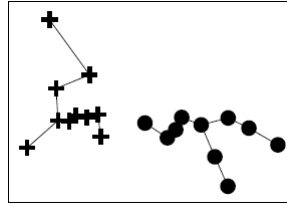
ภาพที่ 3.5 ตัวอย่างของทดสอบรันแบบ 2 มิติของ Wald และ Wolfowitz

$$R = \sum_{i=1}^{m+n-1} z_i + 1 \quad (30)$$

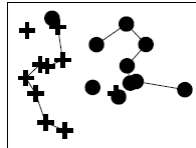
โดยที่ $z_i = 1$ ถ้า Edge ที่ i เป็น Inter-Set Edge และ $z_i = 0$ ถ้าไม่ใช่

ตัวอย่างของทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ถูกแสดงไว้ในภาพที่ 3.5 ภาพที่ 3.5 ก. แสดง Vertex ของยูเนียนเซต $X \cup Y$ โดยที่ $\bullet$ หมายถึง Vertex ที่มาจากเซต X และ

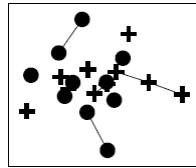
● หมายถึง Vertex ที่มาจากเซต **Y** ภาพที่ 3.5 ข. แสดงต้นไม้แผ่ทั่วที่น้อยที่สุดของ $X \cup Y$ Inter-Set Edge คือ Edge ที่เชื่อมระหว่าง ● และ ● ซึ่งมีทั้งหมด 7 Edge และภาพที่ 3.5 ค. แสดงต้นไม้แผ่ทั่วที่น้อยที่สุดจากกันที่ได้จากการลบ Inter-Set Edge ทั้งหมด ซึ่งจะมีจำนวนต้นไม้แผ่ทั่วจากกันทั้งหมด 8 ต้น ดังนั้นผลลัพธ์ของทดสอบรันของ Wald และ Wolfowitz คือ $R = 8$



ก. ทดสอบรันของ $f = N((0.0, 0.0), 1.0)$ และ $g = N((2.5, 0.0), 1.0)$



ข. ทดสอบรันของ $f = N((0.0, 0.0), 1.0)$ และ $g = N((1.4, 0.0), 1.0)$



ค. ทดสอบรันของ $f = N((0.0, 0.0), 1.0)$ และ $g = N((0.0, 0.0), 1.0)$

ภาพที่ 3.6 ตัวอย่างการทดสอบรันของ Wald และ Wolfowitz ที่มีเงื่อนไขแตกต่างกัน

จากวิธีการของ Prim เราสามารถพิจารณาได้ว่า Inter-Set Edge คือ Edge ของการเชื่อมต่อจุดจากเซต **X** และเซต **Y** ที่ใกล้กันที่สุด ดังนั้นถ้า R มีค่าสูงก็หมายความว่าจำนวนจุดในเซต **X** และเซต **Y** ที่ใกล้กันอยู่มาก ดังนั้นเราสามารถพิจารณาได้ว่าฟังก์ชันการแจกแจง f เหมือนกับฟังก์ชันการแจกแจง g ในทางตรงข้ามถ้า R มีค่าต่ำก็หมายความว่าจำนวนจุดในเซต **X** และเซต **Y** ที่ใกล้กันอยู่น้อย ดังนั้นเราสามารถพิจารณาได้ว่าฟังก์ชันการแจกแจง f ไม่เหมือนกับฟังก์ชันการแจกแจง g ภาพที่ 3.6 แสดงตัวอย่างของ ทดสอบรันของ Wald และ Wolfowitz ที่มีเงื่อนไขต่างกัน 3 เงื่อนไข กำหนดให้ $N(\mu, \sigma)$ เป็นฟังก์ชันการแจกแจงปกติ โดยที่ μ และ σ เป็นค่าเฉลี่ยและค่าความแปรปรวนของการแจกแจง ภาพที่ 3.6 ก. กำหนดให้ $f = N((0.0, 0.0), 1.0)$ และ $g = N((2.5, 0.0), 1.0)$ ฟังก์ชัน f แตกต่างจากฟังก์ชัน g มาก ค่า $R = 2$ ภาพที่ 3.6 ข. กำหนดให้ $f = N((0.0, 0.0), 1.0)$ และ $g = N((1.4, 0.0), 1.0)$ ฟังก์ชัน f คล้ายเหมือนฟังก์ชัน g มาก ค่า $R = 7$ และภาพที่ 3.6 ค. กำหนดให้ $f = N((0.0, 0.0), 1.0)$ และ $g = N((0.0, 0.0), 1.0)$

ฟังก์ชัน $f = g$ ค่า $R=15$ จากการทดลองนี้แสดงให้เห็นว่ายิ่งฟังก์ชันการแจกแจง f เหมือนกับ g เท่าไหร่ยิ่งทำให้ R มีค่าสูงขึ้น

Friedman และ Rafsky คาดเดาว่าสมมติฐานหลัก H_0 จะถูกต้องยิ่งขึ้นถ้า R มีค่าสูงขึ้น

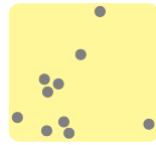
3.3.2 การแจกแจงการเรียงสับเปลี่ยนของทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz

ฟังก์ชันการแจกแจงการเรียงสับเปลี่ยนของทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz เป็นพฤติกรรมที่มีประโยชน์มากและนำมาใช้ในการพัฒนาวิธีการวัดความเหมือนที่ใช้ในงานวิจัยฉบับนี้

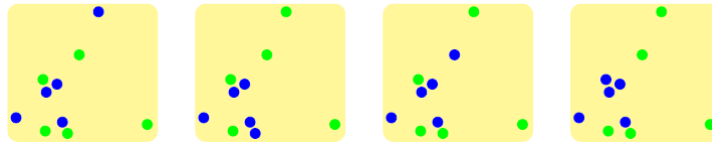
ในการคำนวณค่า R จำเป็นต้องใช้ข้อมูลตำแหน่งของจุดในเซต X และ Y ทั้งหมดเพื่อสร้างต้นไม้แผ่ทั่วที่น้อยที่สุด ถ้าเราทราบแต่เพียงจำนวนของจุดในเซต X และ Y เท่ากับ m และ n ตามลำดับ แต่ไม่ทราบว่าแต่ละจุดอยู่ที่ตำแหน่งใดบ้าง ย่อมเป็นไปได้ที่จะสร้างต้นไม้แผ่ทั่วที่น้อยที่สุด ดังนั้นจึงไม่สามารถคำนวณค่า R ได้ การศึกษาฟังก์ชันการแจกแจงการเรียงสับเปลี่ยนของทดสอบรันเป็นการศึกษาค่าความน่าจะเป็นที่ค่า $R=k$ หรือ $\Pr[R=k]$ การทดลองเพื่อหาค่า $\Pr[R=k]$ ประกอบด้วย 3 ขั้นตอนดังนี้

1. กำหนดตำแหน่งเริ่มต้นทั้งหมดจำนวน $m+n$ ตำแหน่งแบบสุ่ม
2. จากตำแหน่งที่กำหนดในข้อที่ 1. ทำการเรียงสับเปลี่ยนทั้งหมด m ตำแหน่งเพื่อเป็นจุดในเซต X และเรียงสับเปลี่ยน n ตำแหน่งเพื่อเป็นจุดในเซต Y
3. คำนวณค่า R ของสำหรับการเรียงสับเปลี่ยนแต่ละวิธี
4. คำนวณ $\Pr[R=k]$

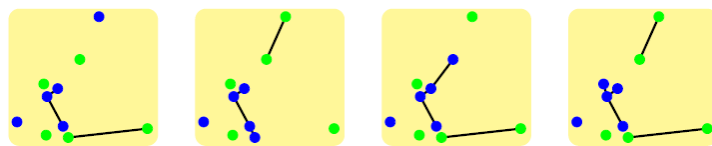
ตัวอย่างของการทดลองหาค่า $\Pr[R=k]$ ถูกแสดงไว้ในภาพที่ 3.7 กำหนดให้ $m, n=5$ ภาพที่ 3.7 ก. แสดงการสุ่มตำแหน่งทั้งหมด 10 ตำแหน่ง ภาพที่ 3.7 ข. แสดงการเรียงสับเปลี่ยนตำแหน่งของจุดในเซต X และ Y เนื่องจากกำหนดให้ $m, n=5$ ดังนั้นจึงมีวิธีในการเรียงสับเปลี่ยนทั้งหมด $\frac{10!}{5!5!} = 252$ วิธี ภาพที่ 3.7 ค. แสดงวิธีการคำนวณค่า R ของการเรียงสับเปลี่ยนแต่ละวิธี และภาพที่ 3.7 ง. แสดงกราฟ $\Pr[R=k]$ ที่ได้จากการทดลอง



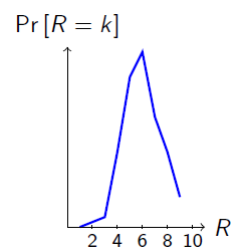
ก. การสุ่มเลือกตำแหน่งทั้งหมด 10 ตำแหน่ง



ข. การเรียงสับเปลี่ยนตำแหน่งของจุดในเซต X และ Y



ค. คำนวณค่า R ของการเรียงสับเปลี่ยนแต่ละวิธี



ง. กราฟ $\Pr[R = k]$

ภาพที่ 3.7. การทดลองหาค่า $\Pr[R = k]$

จากกราฟ $\Pr[R = k]$ ที่แสดงไว้ในภาพที่ 3.7.ง. เราสามารถคำนวณค่าเฉลี่ยได้เท่ากับ 6 ในทางปฏิบัติเราไม่สามารถทดลองหา $\Pr[R = k]$ เมื่อ m และ n มีค่ามากๆ ได้ Friedman และ Rafsky ได้พิสูจน์ทางคณิตศาสตร์จนได้ข้อสรุปว่าฟังก์ชันการแจกแจงการเรียงสับเปลี่ยนของคั่นไม้แผ่นที่น้อยที่สุดของค่า

$$W = \frac{R - E[R]}{\sqrt{\text{Var}[R|C]}} \quad (31)$$

คู่เข้าสู่ฟังก์ชันการแจกแจงแบบปกติ [66] โดยที่ค่าเฉลี่ยและค่าความแปรปรวนของ R คำนวณจาก

$$E[R] = \frac{2mn}{N} + 1 \quad (32)$$

$$\begin{aligned} \text{Var}[R|C] = & \frac{2mn}{N(N-1)} \\ & \left\{ \frac{2mn-N}{N} + \frac{C-N+2}{(N-2)(N-3)} [N(N-1)-4mn+2] \right\} \end{aligned} \quad (33)$$

โดยที่ $N = m + n$ และ C เท่ากับจำนวนคู่ของ Edge ทั้งหมดในต้นไม้แต่ตัวที่น้อยที่สุด กำหนดให้ D_i เป็น Degree ของ Vertex ที่ i จำนวนคู่ของ Edge ทั้งหมดของ Vertex ที่ i เท่ากับ $C_i = \frac{1}{2} D_i (D_i - 1)$ ดังนั้นจำนวนคู่ของ Edge ทั้งหมดสามารถคำนวณได้จาก

$$C = \sum_{i=1}^N C_i = \frac{1}{2} \sum_{i=1}^N D_i (D_i - 1) \quad (34)$$

W สามารถใช้ในการวัดความเหมือนในลักษณะที่ยังค่า W เท่าไหร่ ความเหมือนยิ่งมากขึ้นเท่านั้น

บทที่ 4.

การวัดความเหมือนโดยใช้ทดสอบแบบหลายมิติของ Wald และ Wolfowitz

ทดสอบแบบหลายมิติของ Wald และ Wolfowitz เป็นวิธีที่มีประสิทธิภาพในการวัดความเหมือนของฟังก์ชันแจกแจงหลายมิติ ในงานวิจัยนี้ได้นำทดสอบมาใช้ในการวัดความเหมือนของฟังก์ชันการกระจายของพื้นผิวที่ถูกใช้ในภาพ X-Ray ทางกรแพทย์ 2 ภาพ จากทฤษฎีที่ได้อธิบายในหัวข้อที่ 3.3 เราสามารถวัดความเหมือนของฟังก์ชันแจกแจงของพื้นผิวโดยการสร้างต้นไม้แฟร็กทัลที่น้อยที่สุดของพื้นผิวที่อยู่ในภาพทั้ง 2 ในปริภูมิ Texture และนับจำนวน Edge ที่เชื่อมต่อระหว่างพื้นผิวที่ได้มาจากภาพทั้ง 2 บวกด้วย 1 จากวิธีของ Prim จะได้ว่าค่า R ที่คำนวณได้จะเท่ากับจำนวนคู่ของพื้นผิวทั้งหมดที่อยู่ในภาพ 2 ภาพที่เหมือนกัน ถ้า R มีค่าสูงหมายความว่าภาพทั้ง 2 มีพื้นผิวที่คล้ายกันมาก ดังนั้นภาพทั้ง 2 น่าจะเหมือนกัน ถ้า R มีค่าน้อยหมายความว่าภาพทั้ง 2 ไม่ค่อยมีพื้นผิวที่คล้ายกัน ดังนั้นภาพทั้ง 2 ไม่น่าจะเหมือนกัน แต่อย่างไรก็ตามในการนำทดสอบแบบหลายมิติของ Wald และ Wolfowitz มาใช้วัดความเหมือนของภาพมี ปัญหาที่ต้องได้รับการแก้ไขทั้งหมด 3 ปัญหาดังต่อไปนี้

1. ปัญหาความสม่ำเสมอในการรับรู้ทางพื้นผิว

เพื่อสร้างระบบค้นหารูปภาพโดยรายละเอียดของภาพให้มีประสิทธิภาพ ค่า R ที่คำนวณได้จะต้องเท่ากับจำนวนคู่พื้นผิวจากภาพทั้ง 2 ที่เหมือนกับที่สังเกตโดยมนุษย์ ดังนั้นการออกแบบคุณลักษณะทางพื้นผิวจึงเป็นเรื่องสำคัญมากในการคำนวณค่าทดสอบ คุณลักษณะทางพื้นผิวจะต้องมีคุณสมบัติสม่ำเสมอในการรับรู้ในลักษณะที่ค่าความแตกต่างของพื้นผิวที่คำนวณได้ จะต้องเท่ากับค่าความแตกต่างของพื้นผิวที่มนุษย์สังเกตได้ งานวิจัยได้ใช้วิธีการคำนวณคุณลักษณะทางพื้นผิวโดยใช้การแปลง Wavelet ซึ่งได้นำเสนอไว้ในบทความที่ [68] ซึ่งให้ผลลัพธ์ที่มีประสิทธิภาพมาก

2. ปัญหาขนาดข้อมูล

การสร้างต้นไม้แฟร็กทัลที่น้อยที่สุดสำหรับสีทั้งหมดที่ได้จากภาพทั้ง 2 ใช้เวลานานมาก ซึ่งไม่เหมาะสม ในทางปฏิบัติ ตัวอย่างเช่นความซับซ้อนของการสร้างต้นไม้แฟร็กทัลจากภาพ X-Ray ขนาด 192×128 จุดเท่ากับ $O((2 \times 192 \times 128)^2)$ ดังนั้นจึงจำเป็นต้องใช้เทคนิคในการลดขนาดข้อมูลเพื่อลดจำนวนพื้นผิวที่อยู่ในภาพทั้ง 2 ให้มีขนาดเล็กลง นอกจากนี้เนื่องจากเราต้องการวัดความเหมือนของฟังก์ชันแจกแจงของพื้นผิวของภาพทั้ง 2 ดังนั้น ฟังก์ชันการแจกแจงพื้นผิวที่ได้จากการลดข้อมูลจะต้องคล้ายกับฟังก์ชันการแจกแจงพื้นผิวของเดิมให้มากที่สุดเท่าที่จะเป็นไปได้ ในงานวิจัยนี้ได้ใช้เทคนิคการแบ่งคลัสเตอร์แบบ k-means ในการลดข้อมูล

3. ปัญหาการรักษาความสมดุลระหว่างความเร็วในการคำนวณและประสิทธิภาพของทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz

การรักษาความสมดุลระหว่างความเร็วในการคำนวณ กับประสิทธิภาพของการวัดความเหมือนเป็นเรื่องที่สำคัญ ในระบบการค้นหารูปภาพโดยรายละเอียดของภาพ ระบบที่ดีจะต้องให้ผลลัพธ์ของการค้นหาที่มีประสิทธิภาพสูงในขณะที่ใช้เวลาในการคำนวณน้อย แต่เนื่องจากจำเป็นต้องลดขนาดข้อมูลให้เล็กลงก่อนที่จะคำนวณทดสอบรัน ดังนั้นประสิทธิภาพของการวัดความเหมือนจึงต่ำลง ซึ่งทำให้ไม่เหมาะสมสำหรับระบบการค้นหารูปภาพ จึงจำเป็นต้องมีวิธีในการเพิ่มประสิทธิภาพของทดสอบรันให้เพิ่มขึ้นโดยไม่เพิ่มเวลาในการคำนวณ

ในบทความที่ [57] ผู้วิจัยได้เสนอวิธีการวัดความเหมือนที่ดัดแปลงมาจากการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่เรียกว่าวิธีทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนัก เพื่อแก้ปัญหารักษาความสมดุลระหว่างความเร็วและประสิทธิภาพ ในงานวิจัยนี้จึงเลือก ใช้วิธีการทดสอบรันแบบใหม่นี้ในการวัดความเหมือนของคุณลักษณะทางพื้นผิวของภาพ X-Ray

4.1 การคำนวณคุณลักษณะทางพื้นผิวโดยใช้การแปลง Wavelet

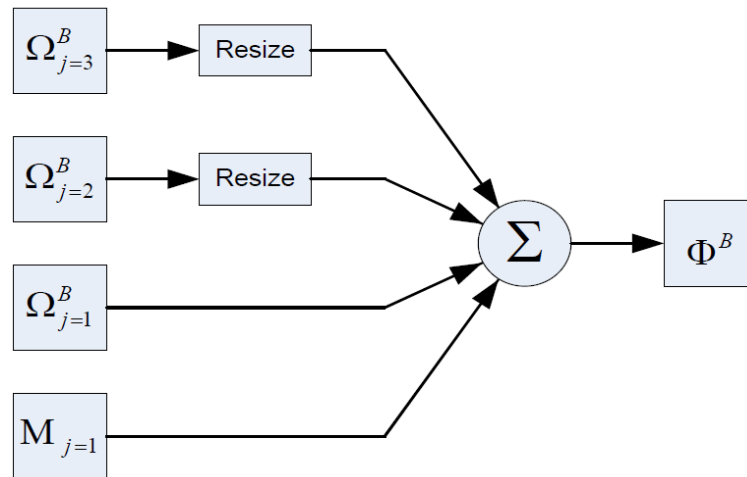
การแปลง Wavelet เป็นเครื่องมือที่สำคัญในการวิเคราะห์คุณลักษณะทางพื้นผิว และการแปลง Wavelet แบบเร็วที่ได้อธิบายในหัวข้อที่ 3.1.4 ทำให้สามารถคำนวณสัมประสิทธิ์ Scaling และ Wavelet ได้อย่างรวดเร็ว แต่อย่างไรก็ตามปัญหาที่สำคัญของการแปลง Wavelet แบบเร็วคือการเกิด Shift Variance ซึ่งเกิดจากกระบวนการ Down Sampling หรือการตัดสัมประสิทธิ์ Scaling หรือ Wavelet ที่อยู่ ณ ตำแหน่งที่เป็นเลขคู่ออกไปที่ทุกระดับความละเอียด การทำ Down Sampling ทำให้ลดจำนวนข้อมูลที่จะต้องเก็บให้น้อยลง แต่ทำให้เกิดปัญหาความไม่แน่นอน (Uncertainty) ในการวิเคราะห์เวลาและความถี่ (ถ้าต้องการเพิ่ม ความแน่นอนในแกนความถี่ จะทำให้แกนเวลามีความแน่นอนลดลง) เป็นผลให้ถ้าภาพที่ต้องการแปลง Wavelet มีการเลื่อนเพียงเล็กน้อย จะทำให้ค่าสัมประสิทธิ์มีการเปลี่ยนแปลงค่อนข้างมาก และทำให้ลักษณะการกระจายของพลังงานในแต่ละระดับความละเอียดมีการเปลี่ยนแปลงมาก และเป็นไปได้ที่สัมประสิทธิ์การแปลง Wavelet ของภาพต้นฉบับและภาพที่มีการเลื่อนเพียงเล็กน้อยแตกต่างกันมาก ปัญหาดังกล่าวจะทำให้การวิเคราะห์คุณลักษณะทางพื้นผิวของภาพต้นฉบับและภาพที่มีการเลื่อนเพียงเล็กน้อย มีความแตกต่างกันอย่างสิ้นเชิง ซึ่งทำให้ประสิทธิภาพในการเปรียบเทียบคุณลักษณะทางพื้นผิวมีค่าลดลง

เราสามารถแก้ปัญหา Shift Variance ได้โดยการเลือก Mother Wavelet ที่มีรูปร่างสมมาตร เป็นที่รู้กันดีว่า Symlet Wavelet เป็น Mother Wavelet ที่มีรูปร่างสมมาตรน้อยที่สุดและมีจำนวน Vanish Moment มากที่สุด ดังนั้นงานวิจัยนี้จึงเลือกใช้ Symlet ในการแปลง Wavelet เพื่อวิเคราะห์คุณลักษณะทางพื้นผิว

กำหนดให้ $w_{j,k,i}^{HL}$, $w_{j,k,i}^{LH}$ และ $w_{j,k,i}^{HH}$ เป็นค่าสัมประสิทธิ์ในย่านความถี่ HL, LH และ HH ที่ระดับความละเอียด j ตามลำดับของการแปลง Symlet Wavelet กำหนดให้ $u_{j,k,i}$ เป็นสัมประสิทธิ์ในย่านความถี่ LL ของระดับความละเอียด j บทความที่ [68] ได้เสนอวิธีการนำเอา $w_{j,k,i}^{HL}$, $w_{j,k,i}^{LH}$, $w_{j,k,i}^{HH}$ และ $u_{j,k,i}$ มาใช้สร้างคุณลักษณะทางพื้นผิวดังต่อไปนี้

กำหนดให้ B หมายถึงย่านความถี่ HL, LH หรือ HH และ Ω_j^B เป็นภาพที่ได้จากการแปลง Wavelet โดยที่ $\Omega_j^B(k,i) = w_{j,k,i}^B$ กำหนดให้ M_j ภาพของสัมประสิทธิ์ Scaling โดยที่ $M_j(k,i) = u_{j,k,i}$ กำหนดให้ Φ^B เป็นภาพของข้อมูลพื้นผิวในย่านความถี่ B เราสามารถสร้าง Φ^B ได้ดังนี้

1. คำนวณหา $\Omega_{j=1}^B$, $\Omega_{j=2}^B$, $\Omega_{j=3}^B$ และ $M_{j=1}$ โดยใช้ Symlet Wavelet
2. ขยายขนาดของ $\Omega_{j=2}^B$ และ $\Omega_{j=3}^B$ ให้เท่ากับขนาดของ $\Omega_{j=1}^B$
3. $\Phi^B = \Omega_{j=1}^B + \Omega_{j=2}^B + \Omega_{j=3}^B + M_{j=1}$



ภาพที่ 4.1 ฟังก์ชันการคำนวณ Φ^B

คุณลักษณะทางพื้นผิวที่ใช้ในงานวิจัยนี้เป็นเซตของ

$$\Phi = \left\{ \left(\Phi^{HL}(x,y), \Phi^{LH}(x,y), \Phi^{HH}(x,y) \right) \middle| x = \left[0, \dots, \frac{w}{2} - 1 \right], y = \left[0, \dots, \frac{h}{2} - 1 \right] \right\}$$

โดยที่ w และ h เป็นความกว้างและสูงของภาพ

4.2 Vector Quantization และการแบ่งคลัสเตอร์แบบ k-Means

กำหนดให้ Φ เป็นเซตของคุณลักษณะทางพื้นผิวของภาพ X-Ray ตามที่นิยามในหัวข้อที่ 3.1 และกำหนดให้ $p(x)$ เป็นฟังก์ชันการแจกแจงของคุณลักษณะทางพื้นผิวของภาพ เราสามารถพิจารณา $p(x)$ เป็นฟังก์ชันความหนาแน่นของจุด โดยที่ $p(x)\Delta$ เท่ากับจำนวนจุดของคุณลักษณะ

ทางพื้นที่ที่ถูกใช้ในภาพที่อยู่ในปริมาตร Δ รอบๆตำแหน่ง x เพื่อลดเวลาในการคำนวณทดสอบ
 รันแบบหลายมิติของ Wald และ Wolfowitz จำเป็นต้องลดขนาดของเซต Φ ที่มีจำนวนสมาชิกเท่ากับ
 $\frac{w}{2} \times \frac{h}{2}$ ให้เป็นเซต $\bar{\Phi} = \{\bar{\Phi}_i\}_{i=1}^N$ โดยที่ $N \ll \frac{w}{2} \times \frac{h}{2}$ กำหนดให้ $\lambda(x)$ เป็นฟังก์ชันความหนาแน่นของ
 จุดของเซต $\bar{\Phi}$ เพื่อให้ค่าทดสอบรันของเซต $\bar{\Phi}$ มีค่าใกล้เคียงกับทดสอบรันของเซต Φ ฟังก์ชัน
 $\lambda(x)$ จะต้องใกล้เคียงกับฟังก์ชัน $p(x)$ ให้มากที่สุดเท่าที่จะเป็นไปได้

ในงานวิจัยนี้ได้ใช้เทคนิคการแบ่งคลัสเตอร์แบบ k-Means เนื่องจากมันได้รับการ พิสูจน์
 มาแล้วว่าเป็นวิธีที่มีประสิทธิภาพในการลดขนาดข้อมูล โดยที่ยังคงรักษาฟังก์ชันความหนาแน่นของ
 จุดให้ใกล้เคียงกับฟังก์ชันเดิมไว้ได้มากที่สุด

4.2.1 Vector Quantization, Integral ของ Bennett และฟังก์ชันความหนาแน่นของจุด

กำหนดให้ $\Omega \subseteq \mathbf{R}^d$, โดยที่ $d \geq 2$ และเซตของจุด $C = \{c_i | c_i \in \Omega\}_{i=1}^N$ เรียกว่า
 Codebook และเรียกจุด c_i ว่า Codeword นอกจากนี้บริเวณที่อยู่รอบๆ Codeword แต่ละจุดที่เรียกว่า
 Voronoi Cell S_i หมายถึงจุดทั้งหมดที่อยู่ใกล้กับ c_i มากกว่า Codeword อื่นๆ หรือสามารถเขียนเป็น
 สมการดังนี้

$$S_i = \{z \in \Omega | \|z - c_i\| \leq \|z - c_j\|\} \quad (35)$$

สำหรับทุกค่า $j \in [1, \dots, N]$ โดยที่ $\|\bullet\|$ กำหนดให้เป็นการวัดระยะทางแบบยูคลิด จากสมการที่ (35)
 เราสามารถเรียก c_i ว่าเป็น Centroid ของ S_i และน้ำหนักของ c_i หมายถึงจำนวนสมาชิกทั้งหมดของ
 S_i

การทำ Vector Quantization N ระดับ ด้วย Codebook $C : Q_{NC}(z)$ หมายถึงฟังก์ชันส่ง
 $\Omega \rightarrow \Omega$ โดยที่ทุก ๆ $z \in S_i$ ถูกแทนที่ด้วยค่า c_i ดังนั้นถ้า z มีฟังก์ชันการความน่าจะเป็นของความ
 หนาแน่นเป็น $p(z) : \Omega \rightarrow \mathbf{R}$ เราสามารถคำนวณหาค่าความบิดเบี้ยวยกกำลัง r ได้จาก

$$\begin{aligned} D(Q_{N,C,p}) &= \frac{1}{d} \int \|z - Q_{N,C}(z)\|^r p(z) dz \\ &= \frac{1}{d} \sum_{i=1}^N \int_{S_i} \|z - c_i\|^r p(z) dz \end{aligned} \quad (36)$$

จากงานบุกเบิกของ Bennett [69] แสดงให้เห็นว่าเราสามารถประมาณค่าความผิดพลาด
 เฉลี่ยกำลังสอง ($r=2$) ของการควอนไทซ์แบบสเกลาร์ ($d=1$) โดยที่ Voronoi Cell มีขนาดเล็ก (N มีค่าสูง)
 และมี c_i อยู่กึ่งกลาง ได้จาก

$$D(Q_{N,C,p}) \cong \frac{1}{12N^2} \int \frac{1}{\lambda(z)^2} p(z) dz \quad (37)$$

โดยที่ $\lambda(z)$ เป็นฟังก์ชันความหนาแน่นจุดของ Centroid ด้านขวามือของสมการถูกเรียกว่า Integral ของ Bennett นอกจากนี้เขายังได้แสดงให้เห็นว่าในกรณีที่เป็นการทำ Vector Quantization ที่ดีที่สุด เราจะได้ $\lambda(z) \cong p(z)^{1/3} / \int p(z)^{1/3} dz$

จากนั้น Na และ Neuhoff ได้ศึกษา Integral ของ Bennett แบบหลายมิติ [70] และได้ผลลัพธ์ดังนี้สมมติให้ Vector Quantization $Q_{N,C}$ เป็นแบบ d มิติ ที่มี Centroid จำนวนมาก (N มีค่าสูง) สมมติให้ขนาดของ Voronoi Cell มีขนาดเล็กและสมมติให้ฟังก์ชัน $p(z)$ เรียบ ดังนั้นเราสามารถประมาณสมการที่ (37) ดังนี้

$$\begin{aligned} D(Q_{N,C,p}) &\cong \frac{1}{d} \sum_{i=1}^N p(z) \int_{S_i} \|z - c_i\|^r dz \\ &= \sum_{i=1}^N p(c_i) M(S_i) v(S_i)^{1+r/d} \end{aligned} \quad (38)$$

โดยที่ $v(S_i)$ เป็นปริมาตรของ S_i และ $M(S_i)$ เป็นโมเมนต์ความเฉื่อยมาตรฐาน (Normalized Moment of Inertia) ของ S_i ที่จุด c_i สำหรับความบิดเบือนยกกำลัง r ซึ่งสามารถคำนวณจาก

$$M(S_i) = \frac{\int_{S_i} \|z - c_i\|^r dz}{d \times v(S_i)^{1+r/d}} \quad (39)$$

การทำให้เป็นมาตรฐานในสมการที่ (39) ทำให้ $M(S_i)$ ไม่ขึ้นกับมาตราส่วนของ S_i และสำหรับ $r=2$ เราจะได้ค่า $M(\text{cube}) = 1/12$ สำหรับทุกๆมิติ d

สมมติว่ามีฟังก์ชัน $\lambda(z)$ (โดยปกติต้องเรียบ) ที่ทำให้ค่าทุกค่า z น้ำหนักของ Centroid ที่อยู่ใน บริเวณเล็กๆ Δ ที่บรรจุ z อยู่สามารถประมาณได้เป็น $\lambda(z)v(z)$ เราเรียกฟังก์ชันนี้ว่า ความหนาแน่นของจุดของ $Q_{N,C}$ ซึ่งสามารถคำนวณได้ดังนี้

$$\lambda(z) \cong \frac{1}{Nv(S_i)} \text{ สำหรับ } z \in S_i \quad (40)$$

การมีอยู่ของฟังก์ชันความหนาแน่นของจุดทำให้คาดการณ์ได้ว่า เซลล์ที่อยู่ติดกันโดยส่วนใหญ่จะมีปริมาตรที่เท่ากัน

นอกจากนี้สมมติว่ามีฟังก์ชัน $m(z)$ (โดยปกติจะเรียบ) ที่ทำให้สำหรับทุกค่า z เซลล์ที่อยู่ใกล้กับ z มีโมเมนต์ความเฉื่อยแบบมาตรฐานใกล้เคียงกับ $m(z)$ ฟังก์ชันนี้จะเรียกว่ารูปโครงร่างของความเฉื่อย (Inertial Profile) ของ $Q_{N,C}$ การมีอยู่ของฟังก์ชันนี้ทำให้คาดการณ์ว่าเซลล์ที่อยู่ใกล้เคียงมีโมเมนต์ความเฉื่อยแบบมาตรฐานที่ใกล้เคียงกัน

เมื่อแทนฟังก์ชันความหนาแน่นของจุดและฟังก์ชันรูปโครงร่างความเฉื่อยลงในสมการที่ (38) เราจะได้

$$D(Q_{N,C,p}) \cong \frac{1}{N^{r/d}} \sum_{i=1}^N p(c_i) \frac{m(c_i)}{\lambda(c_i)^{r/d}} v(S_i) \quad (41)$$

ซึ่งสมการดังกล่าวเป็นการประมาณค่าของการอินทิเกรตแบบ Riemann ดังนั้นเราสามารถ เขียนสมการใหม่ได้ดังนี้

$$D(Q_{N,C,p}) \cong \frac{1}{N^{r/d}} \int p(z) \frac{m(z)}{\lambda(z)^{r/d}} dz \quad (42)$$

ซึ่งเป็นการ Integrall ของ Bennett แบบใหม่

สมการที่ (42) แสดงให้เห็นว่าการบิดเบี้ยวจะลดลงถ้า $\frac{1}{N^{r/d}}$ มีค่ามากขึ้นและแปรผันตรงกับค่าคงที่ที่ขึ้นกับค่าฟังก์ชันความหนาแน่น ความหนาแน่นของจุด และรูปโครงร่างความเฉื่อย สมการที่ (42) ทำให้ทราบว่าประสิทธิภาพของ Vector Quantization ขึ้นกับฟังก์ชันความหนาแน่นจุด และรูปโครงร่างความเฉื่อย

จากการคาดเดาที่มีชื่อเสียงของ Gersho [71] ถ้า N มีค่าสูงขึ้น จะทำให้เซลล์เกือบทั้งหมดของการควอนไทซ์ที่ดีที่สุดใน d มิติ มีรูปร่างคล้ายกับ Tessellating Polytope $H_{r,d}^*$ ที่มีค่าโมเมนต์ความเฉื่อยแบบมาตรฐานต่ำที่สุด สมมติว่ารูปโครงร่างความเฉื่อยคงที่ตามสมการ

$$m_{r,d}^*(z) \triangleq M(H_{r,d}^*) \triangleq M_{r,d}^* \quad (43)$$

และจากทฤษฎีของ Holder เราสามารถพิสูจน์ได้ว่าสมการที่ (42) จะมีค่าต่ำสุดเมื่อ [71]

$$\lambda_{r,d}^*(z) \triangleq \frac{p(z)^{d/(d+r)}}{\int p(z')^{d/(d+r)} dz'} \quad (44)$$

เราสามารถพิจารณาสมการที่ (44) ว่าเป็นฟังก์ชันการแจกแจงของข้อมูลที่ถูกลดขนาดโดยใช้วิธี Vector Quantization

4.2.2 วิธีการแบ่งคลัสเตอร์แบบ k-Means

ในงานวิจัยนี้ใช้วิธีการแบ่งคลัสเตอร์แบบ k-Means ซึ่งวิธีที่รู้จักกันดีสำหรับการทำ Vector Quantization การแบ่งคลัสเตอร์แบบ k-Means จะให้ผลลัพธ์ออกมาเป็นเซต $\{(c_i, w_i)\}_{i=1}^N$ โดยที่ c_i เป็น Centroid ของเซลล์ที่ i และ w_i เป็นจำนวนคุณลักษณะทางพื้นผิวทั้งหมดที่ถูกใช้ในภาพที่เป็นสมาชิกของเซลล์ที่ i เพื่อแก้ปัญหาเรื่องการกำหนดตำแหน่งเริ่มต้นของ Centroid ในงานวิจัยนี้ได้ใช้วิธี Binary Splitting ที่ถูกพัฒนาโดย Linde และคณะ [72]

ขั้นตอนการทำงานของ k-Means ที่ใช้ในงานวิจัยนี้สามารถเขียนเป็นรหัสจำลองได้ดังนี้

1. กำหนดค่าเริ่มต้น

- กำหนดให้เซตของคุณลักษณะทางพื้นผิวเป็น Φ และกำหนดค่าความผิดพลาด $\varepsilon > 0$ ที่มีขนาดเล็กมากๆ

- กำหนดให้ $N^* = 1$ และ

$$c_1^* = \frac{1}{w \times h} \sum_{i=1}^{w \times h} x_i$$

คำนวณ

$$D_{ave}^* = \frac{1}{w \times h \times d} \sum_{i=1}^{w \times h} \|x_i - c_1^*\|^2$$

2. การแยก Centroid

- สำหรับทุกค่า $i = 1, 2, \dots, N^*$

ตั้งค่า

$$c_i^{(0)} = (1 + \varepsilon) c_i^*$$

$$c_{N^*+i}^{(0)} = (1 - \varepsilon) c_i^*$$

- ถ้า $2N^* \leq N$ ดังนั้น

ตั้งค่า

$$N^* = 2N^*$$

ถ้าไม่ใช่

ตั้งค่า

$$N^* = N$$

3. การวนรอบ

- กำหนดให้ $D_{ave}^{(0)} = D_{ave}^*$ และตั้งค่าดัชนีการวนรอบ $j = 0$

สำหรับ $i = 1, \dots, w \times h$ ให้หาค่าต่ำสุดของ

$$\|x_i - c_n^{(j)}\|^2$$

ก) สำหรับทุกค่า $n = 1, \dots, N^*$ กำหนดให้ n^* เป็นค่าที่ทำให้ $\|x_i - c_n^{(j)}\|^2$ ต่ำที่สุด
กำหนดให้

$$Q(x_i) = c_{n^*}^{(j)}$$

ข) สำหรับ $n = 1, \dots, N^*$ ปรับค่าของ Centroid ให้เป็นค่าปัจจุบันดังนี้

$$c_n^{j+1} = \frac{\sum_{Q(x_i)=c_n} x_i}{\sum_{Q(x_i)=c_n} 1}$$

ค) ตั้งค่า $j = j + 1$

ง) คำนวณ

$$D_{ave}^{(j)} = \frac{1}{w \times h \times d} \sum_{i=1}^{w \times h} \|x_i - Q(x_i)\|^2$$

จ) ถ้า $\frac{D_{ave}^{(j-1)} - D_{ave}^{(j)}}{D_{ave}^{(j-1)}} > \varepsilon$ ดังนั้นกลับไปทำงานในขั้นตอน (ก)

ฉ) ตั้งค่า $D_{ave}^* = D_{ave}^{(j)}$

สำหรับ $n = 1, \dots, N^*$

ตั้งค่า

$$c_n^* = c_n^{(i)}$$

4. ย้อนกลับไปทำในขั้นตอนที่ 2 และ 3 จนกระทั่ง $N^* = N$

4.3 นิยามของวิธีทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนัก

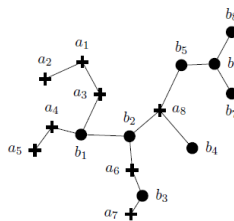
กำหนดให้ Φ_1 และ Φ_2 เป็นเซตของคุณลักษณะทางพื้นผิวของภาพ I_1 และ I_2 ตามลำดับ และกำหนดให้ f และ g เป็นฟังก์ชันการแจกแจงของคุณลักษณะทางพื้นผิวของภาพ I_1 และ I_2 ตามลำดับ จากการหาค่าของ k -Means ที่ได้อธิบายใน หัวข้อที่ 3.2.2 เราจะได้ $r = 2$ และมีมิติของปริภูมิคุณลักษณะทางพื้นผิว $d = 3$ เมื่อแทนค่าลงในสมการที่ (44) เราจะได้ฟังก์ชันความหนาแน่นของจุดของ $\{a_i\}_{i=1}^N$ และ $\{b_i\}_{i=1}^N$ ดังนี้

$$\lambda_A^*(z) = \frac{f(z)^{0.6}}{\int f(z')^{0.6} dz'}$$

และ

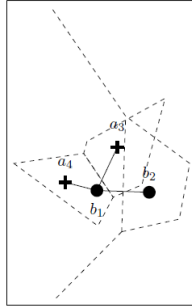
$$\lambda_B^*(z) = \frac{g(z)^{0.6}}{\int g(z')^{0.6} dz'}$$

ในทางทฤษฎีการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ได้จากต้นไม้แผ่ทั่วที่น้อยที่สุดของ $\Phi_1 \cup \Phi_2$ เป็นวิธีการวัดความเหมือนที่มีประสิทธิภาพมาก แต่ในทางปฏิบัติการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดของ $\Phi_1 \cup \Phi_2$ ใช้เวลานานมาก ดังนั้นงานวิจัยนี้จึงได้เสนอวิธีทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนัก ซึ่งเป็นวิธีที่ใช้ในการประมาณจำนวน Inter-Set Edge ดังต่อไปนี้

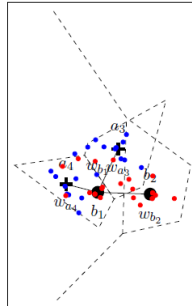


ภาพที่ 4.2 ตัวอย่างของต้นไม้แผ่ทั่วที่น้อยที่สุดของ $\{a_1, \dots, a_8, b_1, \dots, b_8\}$

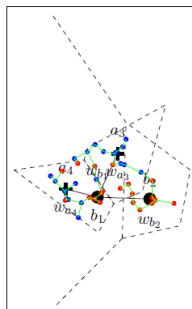
กำหนดให้ $A = \{(a_i, w_{a_i})\}_{i=1}^N$ และ $B = \{(b_i, w_{b_i})\}_{i=1}^N$ เวกเตอร์ของลักษณะเฉพาะที่ได้จาก Φ_1 และ Φ_2 ตามลำดับ กำหนดให้ S_{a_i} และ S_{b_i} เป็น Voronoi Cell ตามนิยามในสมการที่ (35) ของ a_i และ b_i ตามลำดับ



ก. ตัวอย่างของ b_1 , Neighbor และ Voronoi cell



ข. สีของจุดภาพที่เป็นสมาชิกของบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$



ค. ตัวอย่างของทดสอบรันในบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$

ภาพที่ 4.3 ตัวอย่างทดสอบรันในบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$

เมื่อพิจารณาต้นไม้แผ่ทั่วที่น้อยที่สุดของ $A \cup B$ ที่แสดงไว้ในภาพที่ 4.2 จะเห็นว่า Neighbor ของ Centroid b_1 คือ Centroid a_3, a_4 และ b_2 ดังแสดงไว้ในภาพที่ 4.3 ก จากทฤษฎีของ Vector Quantization ทำให้ทราบว่า มีสีของจุดภาพจากภาพ I_1 จำนวนเท่ากับ w_{a_3} และ w_{a_4} อยู่รอบๆ Centroid a_3, a_4 และมีสีของจุดภาพจากภาพ I_2 จำนวนเท่ากับ w_{b_1} และ w_{b_2} อยู่รอบๆ Centroid

b_1, b_2 ดังแสดงในภาพที่ 4.3 ข เราสามารถคำนวณทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ในบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$ ได้ดังแสดงในภาพที่ 4.3 ก

ในทางปฏิบัติการสร้างต้นไม้แฟร็กต์ที่น้อยที่สุดของเซตที่มีจุดจำนวนมาก จำเป็นต้องใช้เวลานานมาก แต่เราสามารถนำทฤษฎีของการแจกแจงการเรียงสับเปลี่ยนที่ได้อธิบายไว้ในหัวข้อที่ 3.3.2 มาใช้ประมาณค่าของทดสอบรันในบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$ ค่าทดสอบรันที่มีความเป็นไปได้มากที่สุดคือ ค่าเฉลี่ยของการแจกแจงการเรียงสับเปลี่ยนซึ่งสามารถคำนวณได้จาก สมการที่ (32) ค่าโดยประมาณของจำนวน Inter-Set Edge ในบริเวณ $S_{a_3} \cup S_{a_4} \cup S_{b_1} \cup S_{b_2}$ คำนวณได้ดังนี้

$$R(b_1) = \frac{2(w_{b_1} + w_{b_2})(w_{a_3} + w_{a_5})}{(w_{b_1} + w_{b_2})(w_{a_3} + w_{a_5})}$$

เราสามารถใช้วิธีการประมาณค่าดังกล่าวในการประมาณจำนวน Inter-Set Edge รอบๆ Centroid ที่เหลือทั้งหมด

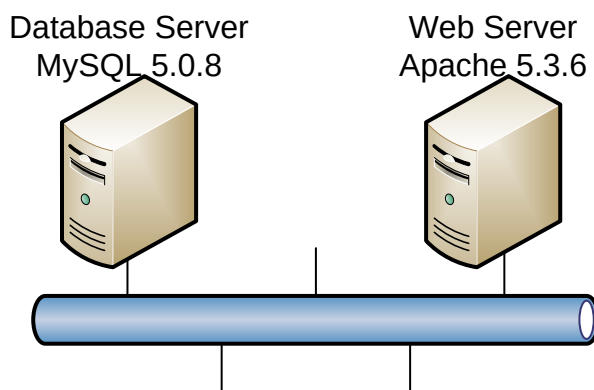
เราเรียกผลรวมของการประมาณจำนวน Inter-Set Edge ของ Centroid ทั้งหมดว่าการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนัก ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$R(I_1, I_2) = \sum_{i=1}^M R(a_i) + \sum_{i=1}^M R(b_i) \quad (45)$$

บทที่ 5.

ระบบค้นหารูปภาพโดยใช้รายละเอียดของภาพ

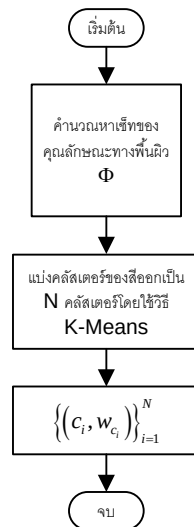
ระบบค้นหารูปภาพโดยใช้รายละเอียดของภาพในงานวิจัยนี้ประกอบด้วย



ภาพที่ 5.1 ระบบค้นหารูปภาพโดยรายละเอียดของภาพ

1. คอมพิวเตอร์ CPU Intel Core i7 ความเร็ว 3.2 GHz หน่วยความจำ 16 GB ระบบปฏิบัติการ Windows 7 64 bit จำนวน 2 เครื่องเพื่อใช้เป็น Web Server และ Database Server
2. เครื่องที่ใช้เป็น Database Server ได้ลงโปรแกรม MySQL 5.0.8 เพื่อใช้ในการจัดการฐานข้อมูลภาพที่ใช้ในงานวิจัย
3. เครื่องที่ใช้เป็น Web Server ได้ลงโปรแกรม PHP 5.3.6
4. ในเครื่อง Web Server ได้ติดตั้ง Web Service ที่พัฒนาโดยใช้ภาษา Visual C++ 2010 ร่วมกับ OpenCV 2.4.3 จำนวน 2 โปรแกรมดังนี้
 - VqKmeanCalculate ทำหน้าที่ในการคำนวณเวกเตอร์ของลักษณะเฉพาะ $\{(c_i, w_{c_i})\}_{i=1}^N$ จากภาพที่ผู้ใช้งานต้องการค้นหา โดยใช้ภาษา Visual C++ 2010 ร่วมกับ OpenCV 2.4.3
 - VqRetrieval ทำหน้าที่คำนวณทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ระหว่างเวกเตอร์ของลักษณะเฉพาะของภาพที่ต้องการค้นหากับภาพที่อยู่ในฐานข้อมูล
5. ในเครื่อง Web Server ได้ติดตั้ง Web Page ที่พัฒนาโดยภาษา PHP และ JQuery เพื่อใช้เก็บภาพที่ต้องการลงในฐานข้อมูลภาพ
6. ในเครื่อง Web Server ได้ติดตั้ง Web Page ที่พัฒนาโดยภาษา PHP และ JQuery เพื่อให้ผู้ใช้บริการค้นหาภาพ ป้อนภาพที่ต้องการค้นหา และแสดงภาพที่ได้จากการค้นหาภาพ

5.1 การทำงานของโปรแกรม VqKmeanCalculate



ภาพที่ 5.2 Flowchart การทำงานของโปรแกรม VqKmeanCalculate

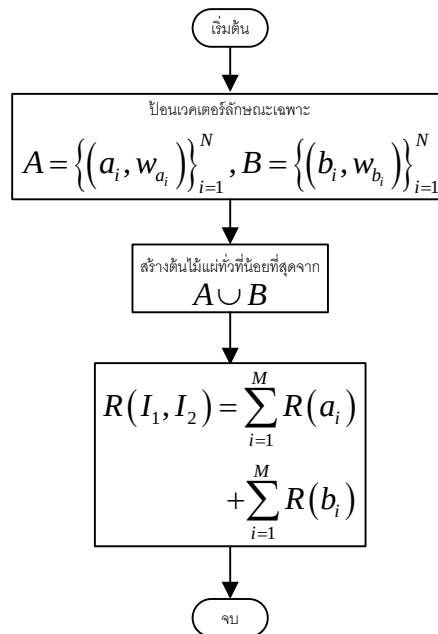
ขั้นตอนการทำงานของโปรแกรม VqKmeanCalculate สามารถเขียนเป็น Flowchart ดังในภาพที่ 5.2 ซึ่งสามารถอธิบายได้ดังนี้

1. คำนวณหาเซตของคุณลักษณะทางพื้นผิว Φ ดังที่ได้อธิบายในหัวข้อที่ 4.1
2. ทำการแบ่งคลัสเตอร์ออกเป็น N คลัสเตอร์โดยใช้วิธี k-Means โดยกำหนดให้การวัดระยะทางใน k-Means $\|\bullet\|$ เป็นการวัดระยะทางแบบยูคลิด ตามที่อธิบายในหัวข้อที่ 4.2.2
3. นำผลลัพธ์ของวิธี k-Means มาใช้สร้างเวกเตอร์ลักษณะเฉพาะ $\{(c_i, w_{c_i})\}_{i=1}^N$ และเก็บเวกเตอร์ลักษณะเฉพาะในไฟล์แบบ YAML

5.2 การทำงานของโปรแกรม VqRetrieval

ขั้นตอนการทำงานของโปรแกรม VqRetrieval สามารถเขียนเป็น Flowchart ดังในภาพที่ 5.3 ซึ่งสามารถอธิบายได้ดังนี้

1. อ่านไฟล์ YAML ที่เก็บเวกเตอร์ลักษณะเฉพาะ $A = \{(a_i, w_{a_i})\}_{i=1}^N$ ของภาพ X-Ray ในฐานข้อมูล และสร้างเวกเตอร์ลักษณะเฉพาะ $B = \{(b_i, w_{b_i})\}_{i=1}^N$ ของภาพ X-Ray ของผู้ป่วยที่ต้องการค้นหา
2. สร้างต้นไม้แผ่ทั่วที่น้อยที่สุดจากยูเนียนเซต $A \cup B$
3. ทำการคำนวณทดสอบรันแบบคิตน้ำหนักตามสมการที่ (45)



ภาพที่ 5.3 Flowchart การทำงานของโปรแกรม VqRetrieval

5.3 การทำงานของ Web Page ที่ใช้ในการเก็บภาพลงในฐานข้อมูล

ระบบค้นหารูปภาพโดยรายละเอียดของภาพจำเป็นต้องมี Web page สำหรับให้ผู้บริหารระบบใช้ในการเก็บภาพที่ต้องการลงในฐานข้อมูลของระบบ Web Page ถูกพัฒนา โดยภาษา PHP ทำงานร่วมกับ JQuery ภาพที่ 5.4 แสดง Web Page ที่ใช้ในการเก็บภาพลงในฐานข้อมูล ซึ่งสามารถแบ่งออกเป็น 2 ส่วนคือ

1. ส่วนป้อนภาพลงในฐานข้อมูล ผู้บริหารสามารถเลือกฐานข้อมูลที่ต้องการนำภาพไปเก็บ และกดปุ่ม Upload a File เพื่อเลือกภาพที่ต้องการนำไปเก็บไว้ในฐานข้อมูล จากนั้นโปรแกรม VqKmeanCalculate จะถูกเรียกเพื่อสร้างไฟล์ YAML ของลักษณะเฉพาะของภาพ จากนั้นไฟล์ภาพและไฟล์ YAML จะถูกนำไปเก็บไว้ในฐานข้อมูล ดังแสดงในภาพที่ 5.4

2. ส่วนสร้างฐานข้อมูลใหม่ เพื่อให้ผู้บริหารสามารถสร้างฐานข้อมูลภาพสำหรับระบบค้นหารูปภาพสำหรับการทำงานที่แตกต่างกันได้ การใช้งานเริ่มจากกรอกชื่อของฐานข้อมูลภาพลงในช่อง “สร้างฐานข้อมูล” และกรอกคำอธิบายของฐานข้อมูลที่จะสร้างลงในช่อง “ข้อมูลของฐานข้อมูล” จากนั้นจึงกดปุ่ม “ยืนยันการสร้าง” เพื่อสร้างฐานข้อมูลภาพใหม่ ดังแสดงในภาพที่ 5.5



ภาพที่ 5.4 Web page สำหรับป้อนภาพลงในฐานข้อมูล

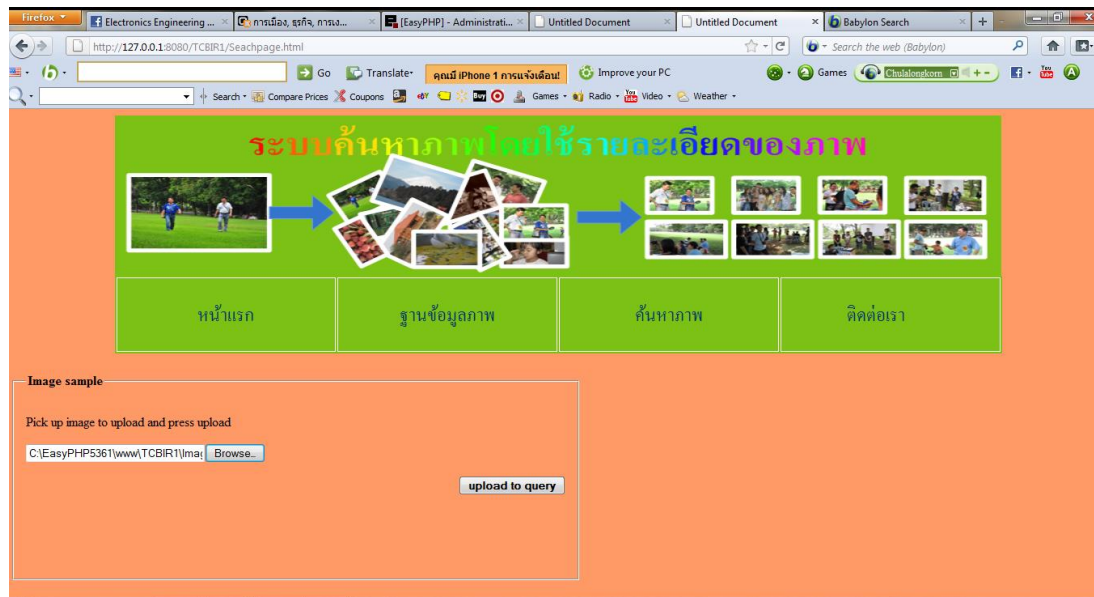


ภาพที่ 5.5 Web page สำหรับสร้างฐานข้อมูลใหม่

5.4 การทำงานของ Web Page ที่ใช้ในการค้นหาภาพในฐานข้อมูล

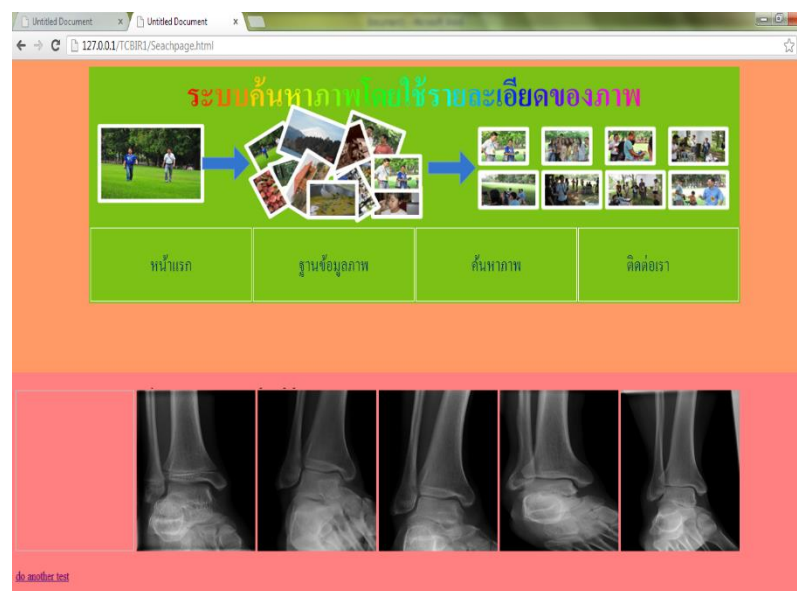
Web Page ที่ใช้ในการค้นหาภาพแบ่งออกเป็น 2 Web Page คือ

1. Web Page ที่ใช้ป้อนภาพที่ต้องการค้นหา ถูกพัฒนาโดยโปรแกรมภาษา PHP ร่วมกับ JQuery เพื่อใช้ป้อนภาพที่ต้องการค้นหาให้กับระบบ ดังแสดงในภาพที่ 5.4 การใช้งานเริ่มจากผู้ใช้กดปุ่ม “Browse” เพื่อเลือกไฟล์ภาพที่ต้องการค้นหา จากนั้นกดปุ่ม “upload to query” เพื่อส่งไฟล์ภาพที่ต้องการค้นหาให้กับระบบค้นหาภาพ ดังแสดงในภาพที่ 5.6



ภาพที่ 5.6 Web page สำหรับป้อนภาพที่ต้องการค้นหา

2. Web Page สำหรับแสดงผลลัพธ์ของการค้นหา ดังแสดงในภาพที่ 5.7 หลังจากระบบทำการค้นหาภาพในฐานข้อมูลแล้ว ก็จะนำเอาภาพที่เหมือนกับภาพที่ต้องการค้นหามากที่สุดจำนวน 10 ภาพแรกมาแสดงผล การแสดงผลเริ่มจากภาพที่ต้องการค้นหาอยู่บนสุด จากนั้นเป็นภาพผลลัพธ์ของการค้นหา โดยเริ่มจากภาพที่เหมือนที่สุด จากนั้นจึงเป็นภาพที่เหมือนอันดับถัดไปจนครบ 10 รูป



ภาพที่ 5.7 Web Page สำหรับแสดงผลลัพธ์ของการค้นหา

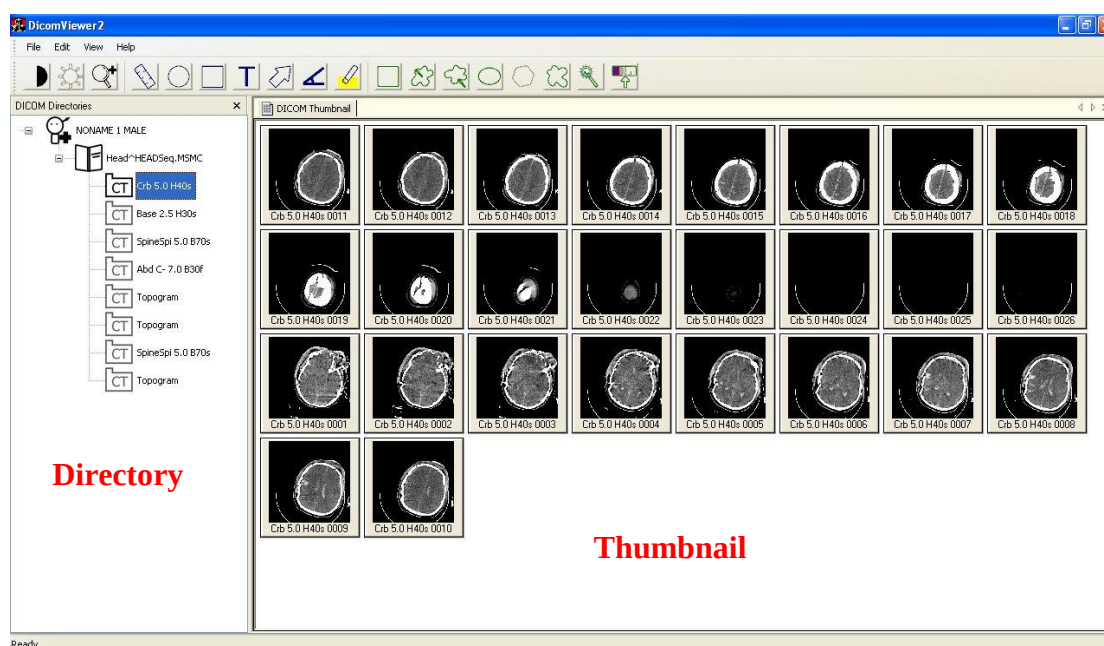
บทที่ 6.

ผลการทดลอง

6.1 การทดลองระบบจัดเก็บและสื่อสารภาพทางการแพทย์

6.1.1 การทำงานของโปรแกรม DICOM Viewer

โปรแกรม DICOM Viewer ได้ถูกจัดทำในรูปแบบของ Win32 Application ซึ่งมี user interface เบื้องต้น ดังแสดงในรูปข้างล่าง



ภาพที่ 6.1 DICOM Viewer

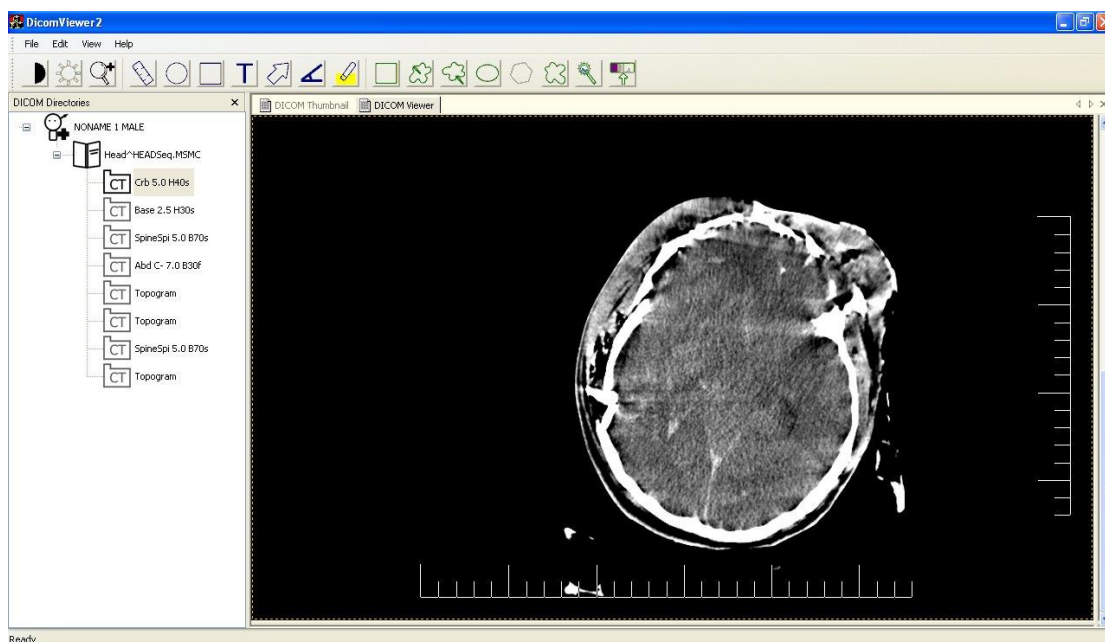
ภายใน DICOM Viewer ประกอบด้วยหน้าต่างหลัก 3 ชนิดคือ

1. DICOM Directory

กินพื้นที่ซ้ายมือของโปรแกรม ทำหน้าที่แสดงรายละเอียดของ DICOM Directory (ประกอบด้วยชื่อคนไข้ และข้อมูลการสแกนทุกครั้งที่ผ่านมา) แพทย์สามารถที่จะตรวจสอบข้อมูลการตรวจรักษาต่างๆของผู้ป่วยได้เช่น วิธีการศึกษา (ตัวอย่างจาก ภาพที่ 6.1 คือ Head^HEADSeq, MSMC) และชนิดของอนุกรม (จากรูป ตัวอย่างจะเป็น Crb 5.0 H40s, Base 2.5 H30s, SpineSpi 5.0 B70s, Abd C-7.0 B30f, Topogram, Topogram, SpineSpi 5.0 B70s, Topogram)

2. Thumbnail

กินพื้นที่ขวามือของโปรแกรม ทำหน้าที่แสดงไฟล์ภาพทั้งหมดจากอนุกรมการตรวจรักษาที่แพทย์คลิกเลือก เช่นจากภาพที่ 6.1 ถ้าแพทย์เลือกอนุกรม Crb 5.0 H40s ภาพทั้งหมดของอนุกรม Crb 5.0 H40s จะถูกแสดงไว้ใน Thumbnail โดยภาพใน Thumbnail จะถูกเรียงลำดับแบบเดียวกับที่แสดงไว้ใน Directory เช่น Crb 5.0 H40s 0001, Crb 5.0 H40s 0002 และ Crb 5.0 H40s 0003, ... เป็นต้น



ภาพที่ 6.2 การแสดงภาพของ DICOM Viewer

3. DICOM Viewer

เมื่อแพทย์คลิกที่ภาพใน Thumbnail ภาพดังกล่าวจะถูกแสดงไว้ใน DICOM Viewer ดังแสดงในภาพที่ 6.2 โดยแพทย์สามารถที่จะแสดงภาพได้พร้อมกันหลายภาพ (จำนวนภาพทั้งหมดที่สามารถเปิดได้ ขึ้นกับขนาดหน่วยความจำของเครื่องคอมพิวเตอร์ที่ใช้เปิด DICOM Viewer) ในส่วนนี้ จะมีเครื่องมือพื้นฐานต่างๆเพื่ออำนวยความสะดวกให้กับแพทย์เพื่อใช้วิเคราะห์คุณสมบัติของภาพ โดยสามารถเรียกใช้เครื่องมือต่างๆจาก Toolbar ดังในภาพที่ 6.3



ภาพที่ 6.3 Toolbar ของ DICOM Viewer

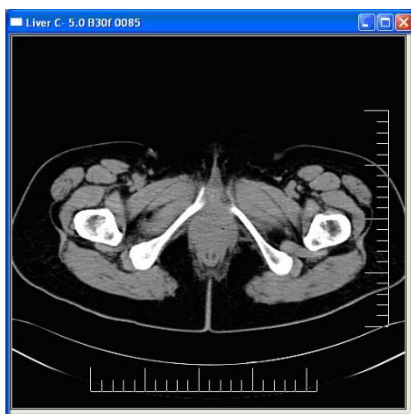
6.1.2 การทำงานของเครื่องมือของโปรแกรม DICOM Viewer

6.1.2.1 การปรับ Window/Level



ผู้ใช้งานปุ่ม

บน Toolbar เพื่อปรับ Window/Level หรือปรับรับของภาพ



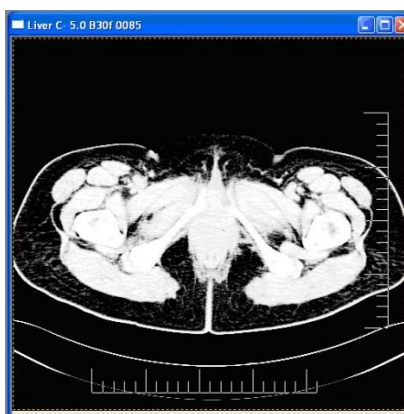
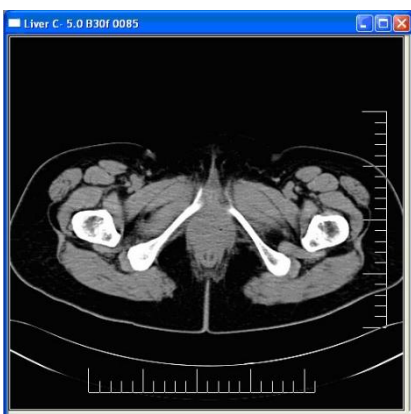
ภาพที่ 6.4 การทดลองปรับ Window/Level

6.1.2.2 การปรับ Alpha Channel



ผู้ใช้งานปุ่ม

บน Toolbar เพื่อปรับ Alpha Channel ของภาพ



ภาพที่ 6.5 การทดลองปรับ Alpha Channel

6.1.2.3 การใช้แว่นขยาย

ผู้ใช้กดปุ่ม  บน Toolbar เพื่อปรับ Alpha Channel ของภาพ



ภาพที่ 6.6 การทดลองใช้แว่นขยายเพื่อส่องดูเฉพาะบริเวณที่ต้องการ

6.1.1.4 การย่อ/ขยายภาพ

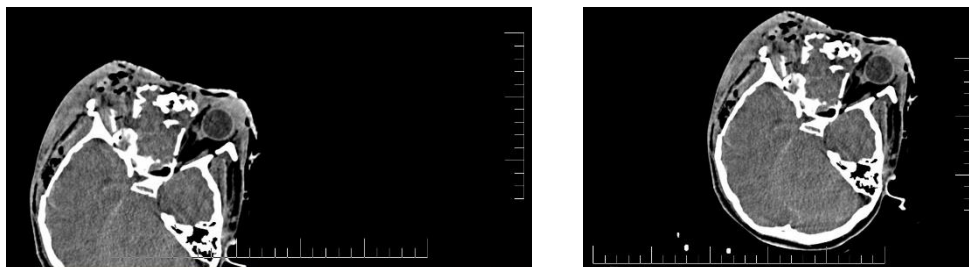
ผู้ใช้สามารถลดหรือขยายภาพโดยการกดปุ่มกลางของเมาส์แล้วเลื่อนขึ้นหรือลง



ภาพที่ 6.7 การทดลองย่อ/ขยายภาพ

6.1.1.5 การเลื่อนภาพ

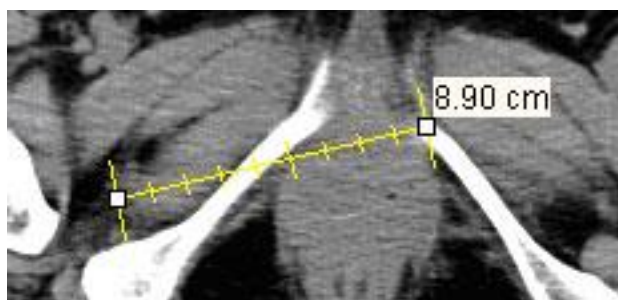
ผู้ใช้สามารถเลื่อนภาพไปยังตำแหน่งต่างๆ โดยการกดปุ่มขวาของเมาส์แล้วเลื่อนภาพไปยังตำแหน่งที่ต้องการ



ภาพที่ 6.8 การทดลองเลื่อนภาพ

6.1.1.6 การวัดขนาด

ผู้ใช้กดปุ่ม  บน Toolbar เพื่อวัดขนาดของวัตถุที่สนใจในภาพ



ภาพที่ 6.9 การทดลองวัดขนาดของวัตถุที่สนใจในภาพ

6.1.1.7 การสร้างวงรี

ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างวงรีเพื่อล้อมกรอบวัตถุที่สนใจใน

ภาพ

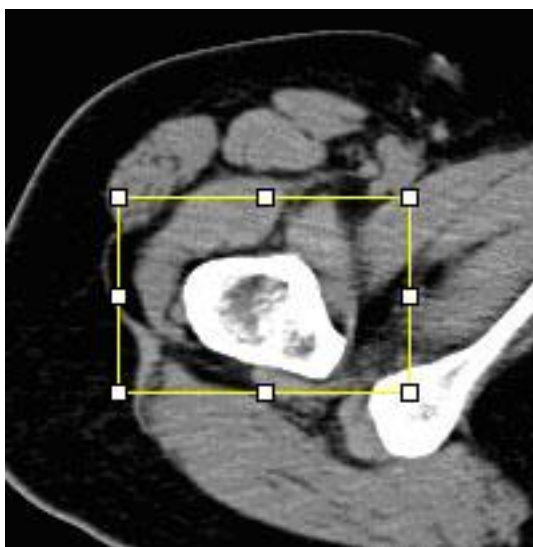


ภาพที่ 6.10 การทดลองสร้างวงรีเพื่อล้อมกรอบวัตถุที่สนใจในภาพ

6.1.1.8 การสร้างกรอบสี่เหลี่ยม

ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างกรอบสี่เหลี่ยมรอบวัตถุที่สนใจใน

ภาพ



ภาพที่ 6.11 การทดลองสร้างกรอบสี่เหลี่ยมรอบวัตถุที่สนใจในภาพ

6.1.1.9 การเขียนคำอธิบาย

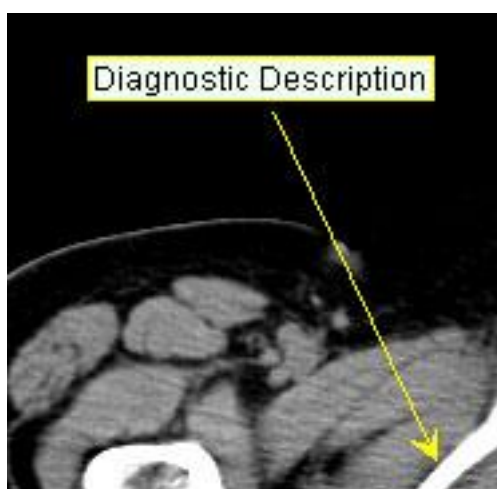
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อคำอธิบายในภาพ



ภาพที่ 6.12 การทดลองเขียนคำอธิบายในภาพ

6.1.1.10 การสร้างลูกศร

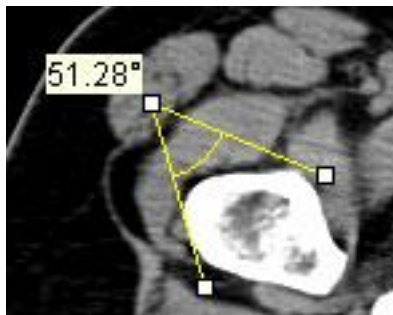
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างลูกศรชี้ไปยังวัตถุที่สนใจในภาพ



ภาพที่ 6.13 การทดลองเพื่อสร้างลูกศรชี้ไปยังวัตถุที่สนใจในภาพ

6.1.1.11 การวัดขนาดมุม

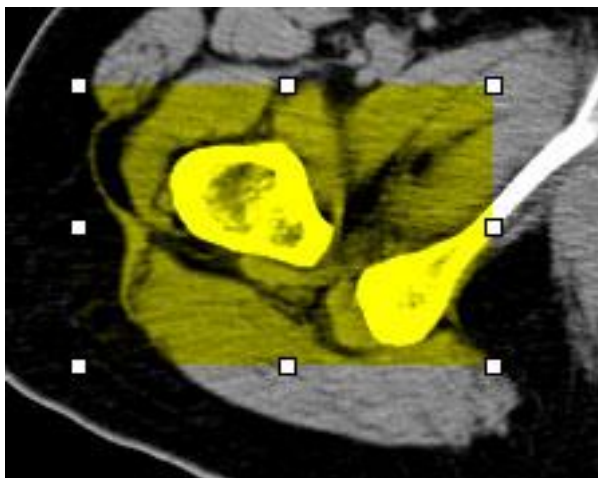
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อขนาดมุมของวัตถุที่สนใจในภาพ



ภาพที่ 6.14 การทดลองวัดขนาดมุมของวัตถุที่สนใจในภาพ

6.1.1.12 การทำ Highlight

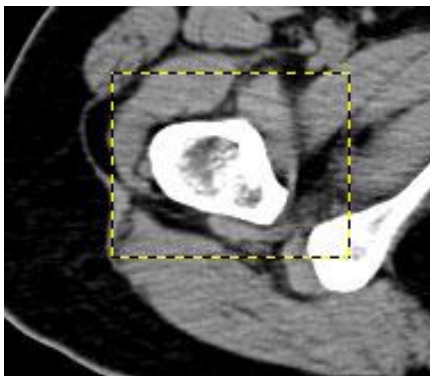
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อ Highlight วัตถุที่สนใจในภาพ



ภาพที่ 6.15 การทดลอง Highlight วัตถุที่สนใจในภาพ

6.1.1.13 การสร้างกรอบเลือกแบบสี่เหลี่ยม

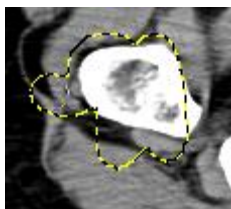
ผู้ใช้งานปุ่ม  บน Toolbar เพื่อสร้างกรอบเลือกวัตถุแบบสี่เหลี่ยม



ภาพที่ 6.16 การทดลองสร้างกรอบเลือกแบบสี่เหลี่ยม

6.1.1.14 การสร้างกรอบเลือกแบบอิสระ

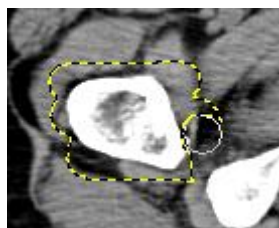
ผู้ใช้งานปุ่ม  บน Toolbar เพื่อสร้างกรอบเลือกวัตถุแบบอิสระ



ภาพที่ 6.17 การทดลองสร้างกรอบเลือกแบบอิสระ

6.1.1.15 การลดขนาดกรอบเลือกแบบอิสระ

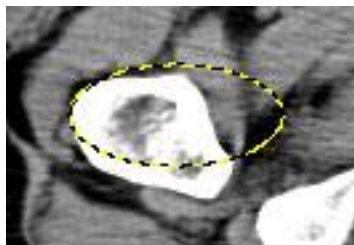
ผู้ใช้งานปุ่ม  บน Toolbar เพื่อลดขนาดกรอบเลือกวัตถุแบบอิสระ



ภาพที่ 6.18 การทดลองลดขนาดกรอบเลือกแบบอิสระ

6.1.1.16 การสร้างกรอบเลือกแบบวงรี

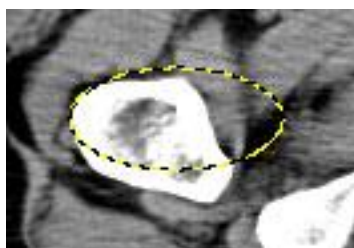
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างกรอบเลือกวัตถุแบบวงรี



ภาพที่ 6.19 การทดลองสร้างกรอบเลือกแบบวงรี

6.1.1.17 การสร้างกรอบเลือกแบบหลายเหลี่ยม

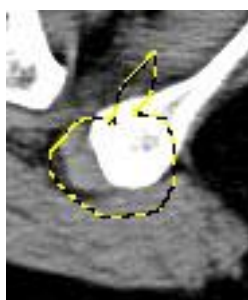
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างกรอบเลือกวัตถุแบบหลายเหลี่ยม



ภาพที่ 6.20 การทดลองสร้างกรอบเลือกแบบหลายเหลี่ยม

6.1.1.18 การสร้างกรอบเลือกแบบ Freehand

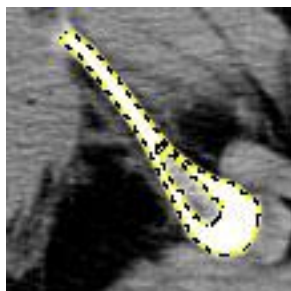
ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างกรอบเลือกวัตถุแบบ Freehand



ภาพที่ 6.21 การทดลองสร้างกรอบเลือกแบบ Freehand

6.1.1.19 การสร้างกรอบเลือกที่มีค่าความสว่างตามที่กำหนด

ผู้ใช้กดปุ่ม  บน Toolbar เพื่อสร้างกรอบเลือกวัตถุที่มีค่าความสว่างตามที่กำหนด



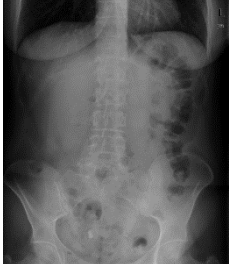
ภาพที่ 6.22 การทดลองสร้างกรอบเลือกที่มีค่าความสว่างตามที่กำหนด

6.2 การทดลองระบบค้นหาภาพทางการแพทย์ด้วยรายละเอียดของภาพ

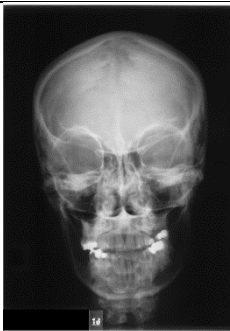
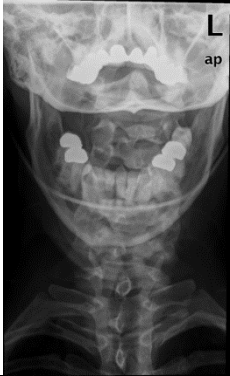
6.2.1 ฐานข้อมูลภาพ

ฐานข้อมูลภาพที่ใช้ในการทดลองประกอบด้วยภาพ X-Ray ทั้งหมด 20 กลุ่ม แต่ละกลุ่มประกอบด้วยภาพทั้งหมด 50 ภาพ รวมเป็น 1000 ภาพ เพื่อพิสูจน์ว่าระบบการค้นหารูปภาพที่ใช้สามารถนำไปใช้กับการค้นหาภาพทางการแพทย์ได้จริง ผู้วิจัยได้พิจารณากลุ่มของภาพจากบริเวณต่าง ๆ ของร่างกาย และทิศทางการถ่ายภาพแบบต่างๆ ดังสรุปไว้ในตารางที่ 6.1

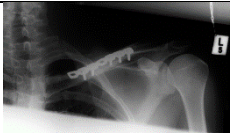
ตารางที่ 6.1 กลุ่มของภาพที่ใช้ในการทดลอง

บริเวณที่ถ่าย	ระบบการทำงาน	ทิศทางการถ่าย	ตัวอย่างภาพ
มือซ้าย	กล้ามเนื้อและกระดูก	ไม่กำหนด	
เท้าซ้าย	กล้ามเนื้อและกระดูก	ไม่กำหนด	
ช่องท้อง	ทางเดินอาหาร	ตามแกน Coronal จากด้านหลังไป ด้านหน้า	

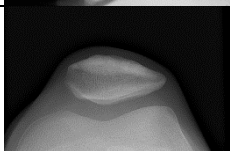
ตารางที่ 6.1 (ต่อ)

บริเวณที่ถ่าย	ระบบการทำงาน	ทิศทางการถ่าย	ตัวอย่างภาพ
กะโหลก	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไปด้านหลัง	
คอ	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไปด้านหลัง	
กระดูกสันหลัง	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไปด้านหลัง	
กระดูกข้อมือซ้าย	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไปด้านหลัง	

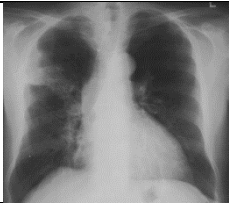
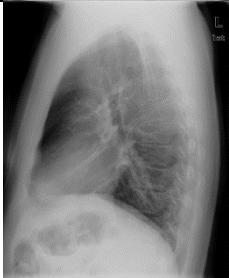
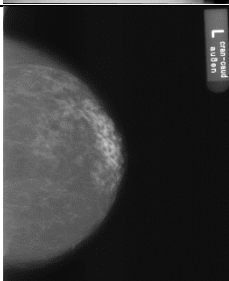
ตารางที่ 6.1 (ต่อ)

บริเวณที่ถ่าย	ระบบการทำงาน	ทิศทางการถ่าย	ตัวอย่างภาพ
กระดูกไหปลาร้า	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไป ด้านหลัง	
กระดูกเชิงกราน	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไป ด้านหลัง	
ข้อเท้าด้านขวา	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไป ด้านหลัง	
หัวเข่าขวา	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไป ด้านหลัง	
สะโพกซ้าย	กล้ามเนื้อและกระดูก	ตามแกน Coronal จากด้านหน้าไป ด้านหลัง	

ตารางที่ 6.1 (ต่อ)

บริเวณที่ถ่าย	ระบบการทำงาน	ทิศทางการถ่าย	ตัวอย่างภาพ
กระดูกสันหลัง บริเวณเอว	กล้ามเนื้อและ กระดูก	ตามแกน Sagittal จากขวาไปซ้าย	
สมองและ ประสาท	กล้ามเนื้อและ กระดูก	ตามแกน Sagittal จากซ้ายไปขวา	
กระดูกสันหลัง ส่วนคอ	กล้ามเนื้อและ กระดูก	ตามแกน Sagittal จากซ้ายไปขวา	
กระดูกสะบ้า	กล้ามเนื้อและ กระดูก	ตามแกน Axial Caudocranial Transversal	
กระโหลกส่วน หน้า	กล้ามเนื้อและ กระดูก	ตามแนว Occipitomental	

ตารางที่ 6.1 (ต่อ)

บริเวณที่ถ่าย	ระบบการทำงาน	ทิศทางการถ่าย	ตัวอย่างภาพ
หน้าอก	ไม่กำหนด	ตามแกน Coronal จากด้านหลังไป ด้านหน้า	
หน้าอก	ไม่กำหนด	ตามแกน Sagittal Lateral จากขวาไป ซ้าย	
หน้าอกผู้หญิง	ระบบสืบพันธุ์ ผู้หญิง	ตามแกน Axial Craniocaudal	

6.2.2 ขั้นตอนการทดลอง

การทดลองสำหรับวัดประสิทธิภาพของระบบการค้นหภาพทางการแพทย์ในงานวิจัยนี้ เป็นดังนี้คือ

1. เลือกภาพจากฐานข้อมูลจำนวน 1 ภาพเพื่อใช้เป็นภาพที่ต้องการค้นหา
2. นำภาพที่ถูกเลือกไปค้นหาภาพในฐานข้อมูลภาพ
3. แสดงเฉพาะภาพที่เหมือนที่สุดจำนวน 10 ภาพแรก

ภาพที่ 6.23 แสดงผลลัพธ์ของการค้นหาภาพช่องท้อง ซึ่งจะเห็นว่าจากภาพที่เหมือนที่สุด 10 ภาพแรก เป็นภาพช่องท้อง 9 ภาพ และเป็นภาพหัวเข่าขวา 1 ภาพ ภาพที่ 6.24 แสดงผลลัพธ์ของการค้นหาภาพกระดูกสันหลัง ซึ่งจะเห็นว่าจากภาพที่เหมือนที่สุด 10 ภาพแรก เป็นภาพกระดูกสันหลังทั้งหมด 10 ภาพ

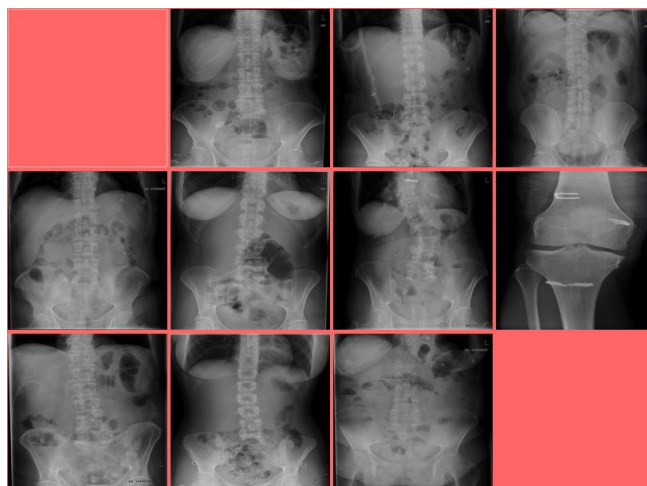
จากตัวอย่างในภาพที่ 6.23 และ 6.24 จะเห็นได้ว่าภาพผลลัพธ์ที่ถูกต้องคือภาพที่อยู่ในกลุ่มเดียวกับภาพที่ต้องการค้นหา เราจะเรียกภาพที่อยู่ในกลุ่มเดียวกับภาพที่ต้องการค้นหาว่าภาพที่เกี่ยวข้อง (Relevant Image) และภาพที่ไม่ได้อยู่ในกลุ่มเดียวกับภาพที่ต้องการค้นหาว่า ภาพที่ไม่เกี่ยวข้อง (Irrelevant Image) ระบบการค้นหภาพที่ดีจะต้องให้ผลลัพธ์เป็นภาพที่เกี่ยวข้องจำนวนมาก ดังนั้นค่าประสิทธิภาพของผลลัพธ์ของการค้นหาภาพที่ใช้ในงานวิจัยนี้เราจะใช้ค่า Precision ซึ่งสามารถคำนวณได้จาก

$$\text{Pr} = \frac{N_R}{10} \quad (46)$$

โดย N_R เท่ากับจำนวนรูปภาพผลลัพธ์ที่เกี่ยวข้อง ตัวอย่างเช่นค่า Pr ของตัวอย่างในภาพที่ 6.23 และ 6.24 คือ 0.9 และ 1.0 ตามลำดับ



ก. ภาพช่องท้องที่ต้องการค้นหา

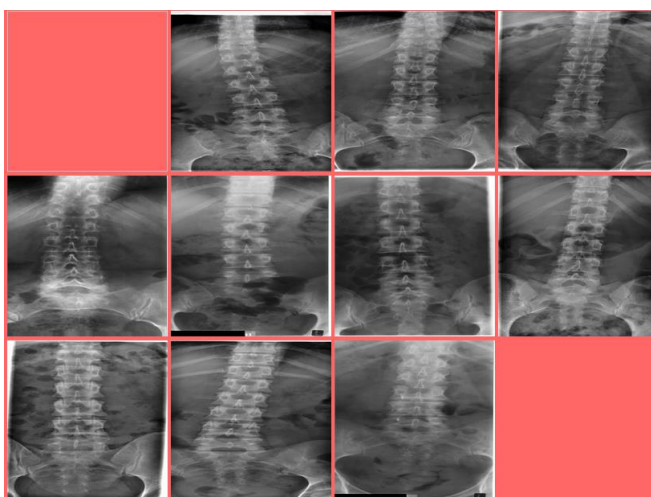


ข. ภาพผลลัพธ์ของการค้นหาภาพช่องท้อง

ภาพที่ 6.23 ตัวอย่างการค้นหาภาพช่องท้อง



ก. ภาพกระดูกสันหลังที่ต้องการค้นหา



ข. ภาพผลลัพธ์ของการค้นหาภาพกระดูกสันหลัง

ภาพที่ 6.24 ตัวอย่างการค้นหาภาพกระดูกสันหลัง

ในการทดลองนี้ใช้ค่าเฉลี่ยของ Pr (Average Precision : AP) เพื่อใช้วัดประสิทธิภาพของระบบการค้นหาภาพ ซึ่งขั้นตอนการวัดประสิทธิภาพของระบบการค้นหาภาพที่ใช้ในการทดลองนี้เป็นดังนี้

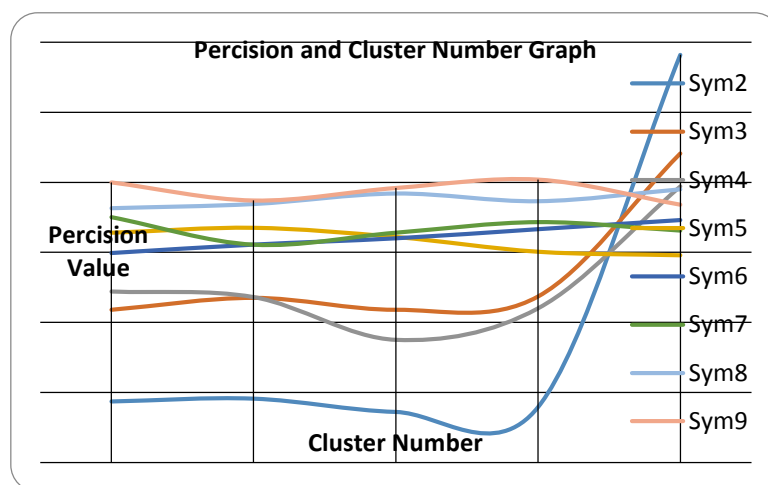
1. สุ่มเลือกภาพ 3 ภาพจากแต่ละกลุ่ม รวมเป็นทั้งหมด 60 ภาพ เพื่อใช้เป็นภาพที่ต้องการค้นหา
2. ป้อนภาพที่ถูกเลือกจากข้อที่ 1 เข้าไปให้ระบบทำการค้นหารูปภาพ
3. ทำการคำนวณค่า Pr สำหรับรูปภาพที่ต้องการค้นหาแต่ละภาพ
4. คำนวณค่าเฉลี่ยของ Pr ทั้งหมดที่ได้จากข้อที่ 3

ค่าเฉลี่ยของ Pr ที่ได้จากการทดลองดังกล่าวหมายถึงค่าความน่าจะเป็นภาพที่เกี่ยวข้องในภาพผลลัพธ์ของการค้นหา

6.2.3 ผลการทดลอง

ตารางที่ 6.2 ผลการทดลองการวัดค่า Precision เฉลี่ย

จำนวนคลัสเตอร์	Sym2	Sym3	Sym4	Sym5	Sym6	Sym7	Sym8	Sym9
20	4.435	5.09	5.22	5.64	5.495	5.75	5.815	6
40	4.455	5.175	5.18	5.675	5.555	5.555	5.845	5.87
60	4.36	5.09	4.875	5.61	5.6	5.64	5.92	5.96
80	4.395	5.185	5.1	5.505	5.665	5.715	5.865	6.02
100	6.91	6.205	5.97	5.48	5.73	5.655	5.95	5.84



ภาพที่ 6.25 กราฟ Precision เฉลี่ย

งานวิจัยนี้ได้ทำการทดลองเพื่อคำนวณหาค่าเฉลี่ยของ Pr สำหรับทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนัก สำหรับคุณลักษณะทางพื้นผิวของภาพ X-Ray ทางกรแพทย์ชนิดต่างๆ เพื่อค้นหาชนิดของ Symlet Wavelet และจำนวนคลัสเตอร์ที่ดีที่สุดสำหรับการวัดความเหมือนของภาพ X-Ray ทางกรแพทย์

ตารางที่ 6.2 แสดงค่าเฉลี่ยของ Pr สำหรับวิธีทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนักสำหรับ Symlet Wavelet ตั้งแต่ Order ที่ 2 ถึง Order ที่ 9 (Symlet Wavelet ที่ Order ที่ N จะมีจำนวน Vanish Moment เท่ากับ N และมีความกว้างเท่ากับ $2N-1$) และจำนวนคลัสเตอร์ตั้งแต่ 10 จนถึง 100 และภาพที่ 6.25 แสดงกราฟของค่า Precision เฉลี่ย ที่ได้จากการทดลอง

ในทางปฏิบัติแพทย์หรือรังสีแพทย์มักต้องการระบบการค้นหภาพที่มีความแม่นยำสูง และใช้เวลาในการค้นหาที่ไม่นาน เนื่องจากระบบที่ขาดความแม่นยำจะให้ผลลัพธ์ที่ไม่เป็นประโยชน์ต่อ

การรักษาผู้ป่วย หรือระบบที่มีความแม่นยำแต่ใช้เวลาในการคำนวณนานเกินไป จะทำให้แพทย์ไม่สามารถรักษาผู้ป่วยได้อย่างทัน ท่วงที ดังนั้นระบบการค้นหภาพที่ดีจะต้องมีค่า Precision เฉลี่ยที่สูง หรือมีโอกาสดูภาพที่ต้องการสูง และมีระยะเวลาในการค้นหภาพน้อย

จากการทดลองพบว่าระยะเวลาส่วนใหญ่ในการค้นหภาพ ขึ้นกับระยะเวลาในการคำนวณการแปลง Wavelet และระยะเวลาในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุด ระยะเวลาในการแปลง Wavelet จะขึ้นกับความยาวของ Wavelet เนื่องจากความยาวของ Symlet Wavelet เท่ากับ $2N-1$ โดยที่ N เท่ากับ Order ของ Symlet Wavelet ในทางปฏิบัติเราสามารถลดระยะเวลาในการแปลง Wavelet ได้ โดยการแบ่งออกเป็น การแปลง Wavelet ของภาพที่อยู่ในฐานข้อมูล ซึ่งเราสามารถที่จะแปลงได้ล่วงหน้าก่อนที่จะมีการค้นหภาพ และการแปลง Wavelet ของภาพที่ต้องการค้นหา ซึ่งเราจำเป็นต้องแปลงทุกครั้งที่ใช้ภาพที่ต้องการค้นหา ดังนั้นในการค้นหภาพแต่ละครั้ง ระยะเวลาในการแปลง Wavelet จะเท่ากับระยะเวลาในการแปลง Wavelet ของภาพที่ต้องการค้นหาเท่านั้น

ระยะเวลาในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดขึ้นกับจำนวนคลัสเตอร์ ในการค้นหภาพแต่ละครั้งเราจะต้องสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดระหว่างภาพที่ต้องการค้นหากับภาพที่อยู่ในฐานข้อมูล ดังนั้นถ้ามีภาพที่อยู่ในฐานข้อมูลทั้งหมด 1000 ภาพ ในการค้นหภาพแต่ละครั้งจะต้องใช้เวลาในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุด 1000 ครั้ง

ดังนั้นระยะเวลาส่วนใหญ่ในการค้นหภาพจะเท่ากับระยะเวลาในการแปลง Wavelet ของภาพที่ต้องการ ค้นหา และระยะเวลาในการสร้างต้นไม้แผ่ทั่วที่น้อยที่สุดจำนวนเท่ากับภาพทั้งหมดที่อยู่ในฐานข้อมูล เราสามารถสรุปได้ว่าจำนวนคลัสเตอร์เป็นตัวแปรที่สำคัญที่สุดในการกำหนดระยะเวลาในการค้นหภาพแต่ละครั้ง

จากตารางที่ 6.2 จะเห็นว่า Symlet Order 2 และจำนวนคลัสเตอร์เท่ากับ 100 จะให้ค่า Precision เฉลี่ยสูงสุด แต่เนื่องจากจำนวนคลัสเตอร์เท่ากับ 100 ทำให้มีระยะเวลาในการค้นหภาพที่นานที่สุดสำหรับในการทดลองนี้ จึงไม่เหมาะสมที่จะนำมาใช้ออกแบบระบบการค้นหภาพ

จากการพิจารณาตารางที่ 6.2 ระบบการค้นหภาพที่มีความแม่นยำสูงและใช้เวลาในการค้นหภาพน้อยคือ ระบบที่ใช้ Symlet Wavelet Order 9 และมีจำนวนคลัสเตอร์เท่ากับ 20 ซึ่งมีค่า Precision เฉลี่ยเท่ากับ 6.0

บทที่ 7. สรุปผลและวิจารณ์

ผู้วิจัยได้ทดลองระบบการค้นหารูปภาพโดยรายละเอียดของภาพ กับภาพ X-Ray ทางการแพทย์ ที่ถ่ายจากบริเวณต่าง ๆ ของร่างกาย ทิศทางการถ่ายภาพต่าง ๆ และระบบการทำงานของร่างกายที่แตกต่างกัน เป็นจำนวน 20 กลุ่ม 1000 ภาพ เพื่อให้แน่ใจว่าระบบที่พัฒนาขึ้นสามารถนำไปใช้งานได้จริง และจากการทดลองสามารถสรุปได้ว่าระบบการค้นหภาพที่ใช้วิธีทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนัก โดย Symlet Wavelet Order 9 และจำนวนคลัสเตอร์เท่ากับ 20 เป็นระบบการค้นหภาพที่เหมาะสมในการใช้งานมากที่สุด

แนวทางในการพัฒนาในอนาคตคือ จะนำเอาวิธีการทดสอบรันแบบหลายมิติของ Wald และ Wolfowitz ที่ขึ้นกับ Centroid แบบมีน้ำหนักไปประยุกต์ใช้กับระบบค้นหภาพทางการแพทย์ชนิดอื่นๆ เช่นภาพ Ultrasound, MRI, CT และอื่นๆ และพัฒนาวิธีการคำนวณคุณลักษณะของภาพที่เหมาะสมกับภาพชนิดนั้นๆ เพื่อให้ได้การระบบการค้นหภาพที่ดียิ่งขึ้นไป

บทที่ 8. โปรแกรม

8.1 โปรแกรม PACS

```
static LILSELOPTION Opt;
static LILITEMOPTION MOpt;

L_BOOL CALLBACK BrowseDlgProc (HWND hDlg, UINT message, WPARAM
wParam, LPARAM lParam)
{
    RECT rcWin;
    L_INT nRet;
    /* selection options structure */
    LILITEMSEL * pSel=NULL;
    BITMAPHANDLE Bitmap;
    THUMBOPTIONS ThumbOptions;
    L_TCHAR szFileName[_MAX_PATH];
    L_INT nID;
    L_TCHAR szCaption[_MAX_PATH+40];
    L_INT iCount;
    L_INT iSelCount;
    L_TCHAR szText1[L_MAXPATH];
    L_TCHAR szText2[L_MAXPATH];
    int iIdx = 0;
    HBITMAPLIST hList;

    ZeroMemory(&ThumbOptions, sizeof(THUMBOPTIONS));

    switch (message)
    {
    case WM_PALETTECHANGED:
        /* forward the message to the ImageList Control */
        if (hCtl)
            SendMessage(hCtl, WM_PALETTECHANGED, wParam, lParam);
        break;
    case WM_INITDIALOG:
        /* create the LEAD ImageList Control */
        L_UseImageListControl();
```

/*Dynamically loads the appropriate DLL to let your program use the
LTIMGLISTCLASS registered class.

Comments

Because the LTIMGLISTCLASS registered class uses messages rather than function calls, the required dynamic link library (LTLST15u.dll) is not loaded automatically. Therefore, the toolkit provides this dummy function in the DLL, which you can call before using the registered class messages.*/

```
GetClientRect(hDlg, &rcWin);
```



```

        hCtl =
L_CreateImageListControl(L_ILS_ACCEPTDROPPFILES|WS_CHILD|WS_VISIBLE|WS
_BORDER,
                                0,                                /* x
location                                */
                                0,                                /* y
location                                */
                                RECTWIDTH(&rcWin), /* width
*/
                                RECTHEIGHT(&rcWin), /* height
*/
                                hDlg,                                /* parent
window                                */
                                IDC_LEADIMAGELIST, /* control
ID                                */
                                RGB(128,128,128)); /*
background color */

```

*/*L_LTLST_API HWND **L_CreateImageListControl**(dwStyle, x, y, nWidth, nHeight, hWndParent, nID, crBack)*

```

DWORD dwStyle;    /* window style */
L_INT x;          /* horizontal position of window */
L_INT y;          /* vertical position of window */
L_INT nWidth;     /* window width */
L_INT nHeight;    /* window height */
HWND hWndParent;  /* handle to parent window */
L_INT nID;        /* control ID */
COLORREF crBack;  /* background color of control */

```

Creates the image list control as a child of the specified parent window./*

```

if(!IsWindow(hCtl))
{
    MessageBox(hDlg, TEXT("Error creating control"),
TEXT("Error"), MB_OK);
    DestroyWindow(hDlg);
}

// Set LEADTOOLS icon to the dialog
SendMessage(hDlg, WM_SETICON, (WPARAM)ICON_SMALL,
(LPARAM)LoadIcon(hInst, MAKEINTRESOURCE(IDI_MAIN)));

// Enable / Disable selection style & method according to the
flag at preferences
Opt.uStructSize = sizeof(LILSELOPTION);
L_ImgListGetSelOptions(hCtl, &Opt);
Opt.uAllowSelection = uAllowSelection;
Opt.uSelectionStyle = uSelectionStyle;
L_ImgListSetSelOptions(hCtl, &Opt);
/*
L_ImgListGetSelOptions(hCtl,&Opt); =
SendMessage(
    hCtl, L_ILM_GETSELOPTIONS,0,(LPARAM)(pLILSELOPTION)(&Opt))
L_ImgListSetSelOptions(hCtl,&Opt); =

```

```

SendMessage(
    hCtl, L_ILM_SETSELOPTIONS, 0, (LPARAM) (pLILSELOPTION) (&Opt))
*/
    // Enable / Disable display text according to the flag at
preferences
    memset(&MOpt, 0, sizeof(MOpt));
    MOpt.uStructSize = sizeof(LILITEMOPTION);
    L_ImgListGetItemOptions(hCtl, &MOpt);
    MOpt.bDisplayItemText = bDisplayItemText;
    L_ImgListSetItemOptions(hCtl, &MOpt);
/*
SendMessage(hCtl, L_ILM_GETITEMOPTIONS, 0,
    (LPARAM) (pLILITEMOPTION)pItemOptions)

SendMessage(hCtl, L_ILM_SETITEMOPTIONS, 0,
    (LPARAM) (pLILITEMOPTION)pItemOptions)
*/

    // Enable / Disable keyboard according to the flag at
preferences
    L_ImgListEnableKeyboard(hCtl, bKeyboard);

    // Set scroll style Vert. / Horz according to the flags at
preferences
    L_ImgListSetScrollStyle(hCtl, uScrollStyle);

    SetFocus(hCtl);
    return(FALSE);

case WM_DESTROY:
    if(IsWindow(hCtl))
    {
        L_ImgListClear(hCtl);
        DestroyWindow(hCtl);

        hCtl = NULL;
        iSortMethod = IMAGELIST_NOTSORTED;

        CheckMenuItem(GetMenu(hWndFrame), IDM_SORTASND, MF_BYCOMMAND
| MF_UNCHECKED);
        CheckMenuItem(GetMenu(hWndFrame), IDM_SORTDSND, MF_BYCOMMAND
| MF_UNCHECKED);
    }
    break;

case WM_SIZE:
    GetClientRect(hDlg, &rcWin);
    MoveWindow(hCtl, 0, 0, RECTWIDTH(&rcWin), RECTHEIGHT(&rcWin),
TRUE);
    break;

case WM_COMMAND:
    nID = CTLID(wParam, lParam);
    switch (CTLID(wParam, lParam))
    {
        case IDOK:
        case IDCANCEL:
            DestroyWindow(hDlg);
            hBrowseDlg = NULL;
            return(TRUE);
        case IDC_LEADIMAGELIST:

```

```

if (HIWORD(wParam) == L_ILN_DROPFILES)
{
    if (lParam)
    {
        BITMAPHANDLE Bitmap;
        LILITEM Item;
        L_INT nResult;
        THUMBOPTIONS to;
        L_TCHAR * pTemp;

        memset (&Bitmap, 0, sizeof (BITMAPHANDLE));
        Bitmap.uStructSize = sizeof (BITMAPHANDLE);

        pTemp = _tcsrchr((L_TCHAR *)lParam, '\\');
        pTemp++;

        memset (&to, 0, sizeof (THUMBOPTIONS));
        to.uStructSize = sizeof (THUMBOPTIONS);
        to.nWidth = 80;
        to.nHeight = 80;
        to.nBits = 24;
        to.uCRFlags = 0;
        to.bMaintainAspect = TRUE;
        to.bForceSize = FALSE;
        to.crBackColor = 0;
        to.bLoadStamp = FALSE;
        to.bResample = TRUE;

        nResult = L_CreateThumbnailFromFile((L_TCHAR *)lParam,
                                            &Bitmap,

sizeof (BITMAPHANDLE),

                                            &to,
                                            NULL, NULL, NULL,

NULL);

        if (nResult == SUCCESS)
        {
            memset(&Item, 0, sizeof (LILITEM));
            Item.uStructSize = sizeof (LILITEM);
            Item.pText = pTemp;
            Item.pTextExt = (L_TCHAR *)lParam;
            Item.pBitmap = &Bitmap;
            Item.uBitmapStructSize = sizeof (BITMAPHANDLE);
            Item.bSelected = FALSE;
            Item.lData = 1;
            Item.uMask = LILITEM_ALL;
            ListBox_AddString(GetDlgItem(hBrowseDlg,
IDC_HIDDENLIST), (L_TCHAR *)lParam);
            L_ImgListInsert(hCtl, &Item);
            L_ImgListEnsureVisible(hCtl,
L_ImgListGetItemCount(hCtl)-1);
        }
    }
}

if (HIWORD(wParam) == L_ILN_LOADDBSTATUS)
{
    LILITEM Item;

    L_INT nIndex;

```

```

        nIndex = (L_INT)lParam;
        memset(&Item, 0, sizeof(LILITEM));
        Item.uStructSize = sizeof(LILITEM);
        Item.uMask = LILITEM_TEXTTEXT;
        L_ImgListGetItem(hCtl, nIndex, &Item);
        if(Item.pTextExt && lstrlen(Item.pTextExt))
            ListBox_AddString(GetDlgItem(hBrowseDlg,
IDC_HIDDENLIST), Item.pTextExt);
        else
        {
            //file probably was not really a ImgList DB
            ListBox_AddString(GetDlgItem(hBrowseDlg,
IDC_HIDDENLIST), szDBFile);
        }
        L_ImgListEnsureVisible(hCtl, nIndex);
    }

    if(HIWORD(wParam) == L_ILN_ITEMSEL)
    {
        if(!fInProcess)
        {
            iSelCount = (L_INT)L_ImgListGetSelCount(hCtl);
            iCount = (L_INT)L_ImgListGetItemCount(hCtl);

            if(iSelCount > 0)
            {
                pLILITEM pItems = NULL;
                L_INT i;
                pItems = (pLILITEM) GlobalAllocPtr(GHND,
sizeof(LILITEM) * iSelCount);
                for(i=0;i<iSelCount;i++)
                {
                    pItems[i].uStructSize = sizeof(LILITEM);
                    pItems[i].uBitmapStructSize =
sizeof(BITMAPHANDLE);
                }

                L_ImgListGetSelItems(hCtl, pItems);
                for(iIdx = 0; iIdx < iCount; iIdx++)
                {
                    L_TCHAR strTemp[L_MAXPATH];
                    errno_t strError = 0 ;

                    ZeroMemory (szText1, sizeof(szText1));
                    ZeroMemory (szText2, sizeof(szText2));

                    if(pItems[iSelCount - 1].pText)
                    {
                        _tcscpy_s(strTemp, L_MAXPATH, TEXT("\\"));
                        _tcscat_s(strTemp, L_MAXPATH,
pItems[iSelCount - 1].pText);
                        _tcscpy_s(szText1, L_MAXPATH, strTemp);
                        strError = _tcsupr_s(szText1, L_MAXPATH);
                    }

                    pSel = (LILITEMSEL *)lParam;
                    ListBox_GetText(GetDlgItem(hDlg,
IDC_HIDDENLIST), iIdx, szFileName);

                    _tcscpy_s(szText2, L_MAXPATH, szFileName);
                    _tcsupr_s(szText2, MAX_PATH);
                }
            }
        }
    }

```

```

        if(strError == 0)
        {
            if(_tcsstr(szText2, szText1) != NULL)
            {
                L_GetDefaultLoadFileOption(&LoadFileOption, sizeof(LOADFILEOPTION));
                LoadFileOption.PageNumber =
                pItem[iSelCount - 1].lData;
                break;
            }
        }
    }

    // Get item that was just selected
    pLILITEMSEL p = (pLILITEMSEL)lParam;
    if (p)
    {
        ListBox_GetText(GetDlgItem(hDlg, IDC_HIDDENLIST),
        p->lIndex, szFileName);
    }

    L_FileInfo(szFileName, &foFileInfo, sizeof(FILEINFO),
    FILEINFO_TOTALPAGES, &LoadFileOption);

    if (foFileInfo.Format == FILE_GIF)
    {
        nRet = L_LoadBitmapList (szFileName, &hList, 0,
        fIsBGRDevice ? ORDER_BGRORGRAY : ORDER_RGBORGRAY, NULL ,&foFileInfo);

        if(nRet == SUCCESS)
        {
            L_GetBitmapListItem(hList, 0,
            &Bitmap, sizeof(BITMAPHANDLE));
            L_GetBitmapColors(&Bitmap, 0, Bitmap.nColors,
            foFileInfo.GlobalPalette);
            nRet = L_CreateBitmap( &Bitmap,
                                sizeof(BITMAPHANDLE),
                                TYPE_CONV,
                                foFileInfo.GlobalWidth,
                                foFileInfo.GlobalHeight,
                                foFileInfo.BitsPerPixel,
                                fIsBGRDevice ? ORDER_BGR
                                : ORDER_RGB,
                                foFileInfo.GlobalPalette,
                                foFileInfo.ViewPerspective,
                                NULL, 0);

            if(nRet == SUCCESS)
            {
                if(foFileInfo.Flags &
                FILEINFO_HAS_GLOBALBACKGROUND)
                    Bitmap.Background =
                    foFileInfo.GlobalBackground;
                CreateChildWindow (szFileName, &Bitmap,
                hList, (foFileInfo.Flags & FILEINFO_HAS_GLOBALLOOP) != 0, 1, TRUE);
            }
            else
            {
                L_DestroyBitmapList(hList);
            }
        }
    }
}

```

```

    }
    }
    else
    {
        if( fLoadCompressed )
            nRet = L_LoadFile (szFileName, &Bitmap,
sizeof(BITMAPHANDLE), 0, fIsBGRDevice ? ORDER_BGRORGRAY :
ORDER_RGBORGRAY,
                                LOADFILE_ALLOCATE |
LOADFILE_STORE | LOADFILE_COMPRESSED, NULL, NULL, &LoadFileOption,
&foFileInfo);
        else
            nRet = L_LoadFile (szFileName, &Bitmap,
sizeof(BITMAPHANDLE), 0, fIsBGRDevice ? ORDER_BGRORGRAY :
ORDER_RGBORGRAY,
                                LOADFILE_ALLOCATE |
LOADFILE_STORE, NULL, NULL, &LoadFileOption, &foFileInfo);
        if(nRet != SUCCESS)
        {
            L_FileError (hDlg, nRet, TEXT("Open File"),
szFileName);
            break;
        }
        DisplayDICOMLoadMsg(hWndFrame, NULL, szFileName);
        CreateChildWindow (szFileName, &Bitmap, NULL,
1,1,FALSE);
    }

    SetActiveWindow(hDlg);
}
else
    fKillProgress = TRUE;
}
else if( (HIWORD(wParam) == L_ILN_KEYDOWN) ||
        (HIWORD(wParam) == L_ILN_CLICKED) )
{
    fKillProgress = TRUE;
}
}
break;

case WM_BROWSE:
    /* start the browse */
    L_ImgListClear(hCtl);
    ListBox_ResetContent(GetDlgItem(hDlg, IDC_HIDDENLIST));
    fKillProgress = FALSE;
    fInProcess = TRUE;

    ThumbOptions.uStructSize      = sizeof(THUMBOPTIONS);
    ThumbOptions.bForceSize      = FALSE;
    ThumbOptions.bLoadStamp      = FALSE;
    ThumbOptions.bMaintainAspect = TRUE;
    ThumbOptions.bResample       = FALSE;
    ThumbOptions.crBackColor     = RGB(128,128,128);
    ThumbOptions.nBits           = 24;
    ThumbOptions.nHeight         = 80;
    ThumbOptions.nWidth          = 80;
    ThumbOptions.uCRFlags        = CRF_BYTEORDERBGR;

    _tcscpy_s(szCaption, (_MAX_PATH+40), TEXT("Browsing: "));

```

```

_tcscat_s(szCaption, 300, szImageDir);
SetWindowText(hDlg, szCaption);

nRet = L_BrowseDir(szImageDir,
                  TEXT("*. *"),
                  &ThumbOptions,
                  bStopOnError,
                  bIncludeSubD,
                  bExpandMulti,
                  0,
                  0, //4000*1024, /* set limit on image size to
4MB*/

                  BrowseDirCB,
                  (L_VOID *)hCtl);

fInProcess = FALSE;
if(nRet != SUCCESS)
{
    if(nRet != ERROR_USER_ABORT)
    {
        L_FileError(hDlg, nRet, TEXT("Browse"), szImageDir);
        DestroyWindow(hDlg);
    }
    else
        MessageBox(hDlg, TEXT("Browse Canceled!"),
TEXT("Browse"), MB_OK);
    break;
}
return(FALSE);
}

```

8.2 โปรแกรม VqKmeanCalculate

8.2.1 VQ_KmeanKDTree.h

```

#include "opencv2/opencv.hpp"
#include <vector>
#include <algorithm>

using namespace cv;

typedef struct{
    double B,G,R;
} RgbPixelDouble;

typedef struct{
    float B,G,R;
} RgbPixelFloat;

typedef struct{
    double C0 ,C1 ,C2;
} centerDouble;

typedef struct{
    float C0 ,C1 ,C2;
} centerFloat;

#pragma once
class VQ_KmeanKDTree

```

```

{
public:
    VQ_KmeanKDTree(void);
    ~VQ_KmeanKDTree(void);
    void setDataImageBGR(Mat &srcImg);
    void setDataImageBGROpenCV(Mat &srcImg, bool isUnique);
    void kmeansExecute(int cluster, int loopNumber);
    void openCVKmeansExecute(int cluster, int loopNumber);
    bool kmeansExecuteOpenCV(int cluster_count, int loopNumber);
    bool writeDataYAML(char *filename);
    vector<int> eachClusterNumber;
    vector<centerDouble> eachDoubleClusterCenter;
    vector<centerFloat> eachFloatClusterCenter;
private:
    vector<int> vectorPixelColorAll;
    vector<RgbPixelDouble> vectorDoubleUniquePixelRGB;
    vector<float> vectorFloatUniquePixelRGB;
    Mat pointSamples;
    Mat centerCluster;
    Mat eachClusterCount;
    void uniqueData(void);
    void prepare2Double(void);
    void prepare2FloatForOpenCV(void);
    void prepareDataWithUnique(Mat &srcImg, Mat &pointSamples);
    void prepareDataWithOutUnique(Mat &srcImg, Mat &pointSamples);
};

```

8.2.2 VQ_KmeanKDTree.cpp

```

#include "VQ_KmeanKDTree.h"

VQ_KmeanKDTree::VQ_KmeanKDTree(void)
{
}

VQ_KmeanKDTree::~~VQ_KmeanKDTree(void)
{
    if(this->pointSamples.refcount!=NULL)
        this->pointSamples.release();
}

void VQ_KmeanKDTree::setDataImageBGROpenCV(Mat &srcImg, bool isUnique)
{
    if (isUnique)
    {
        this->prepareDataWithUnique(srcImg, this->pointSamples);
    }else
    {
        this->prepareDataWithOutUnique(srcImg, this->pointSamples);
    }
}

void VQ_KmeanKDTree::prepareDataWithUnique(Mat &srcImg, Mat &pointSamples)
{
    int lenghtPixel = srcImg.rows * srcImg.cols;

    vector<int> vectorIntTemp(lenghtPixel);

```



```

        MatIterator_<Vec3b> it = srcImg.begin<Vec3b>(), it_end =
srcImg.end<Vec3b>();

        int tempRGBInt;
        for(int i=0; it != it_end; ++it, ++i)
        {
            tempRGBInt = (int) (*it) [0];
            tempRGBInt += ((int) (*it) [1]) << 8;
            tempRGBInt += ((int) (*it) [2]) << 16;
            vectorIntTemp[i] = tempRGBInt;
        }

        // unique process using STL
        sort( vectorIntTemp.begin(), vectorIntTemp.end() );
        vectorIntTemp.erase( unique( vectorIntTemp.begin(),
vectorIntTemp.end() ), vectorIntTemp.end() );

        Mat processMat(vectorIntTemp.size(), 1, CV_MAKETYPE(CV_8U, 3));

        it = processMat.begin<Vec3b>();
        it_end = processMat.end<Vec3b>();

        int intBGR;
        int mask = 255;
        for(int i=0; it != it_end; ++it, ++i)
        {
            intBGR = vectorIntTemp[i];
            (*it) [0] = saturate_cast<uchar>(intBGR & mask);
            (*it) [1] = saturate_cast<uchar>((intBGR >> 8) & mask);
            (*it) [2] = saturate_cast<uchar>((intBGR >> 16) & mask);
        }
        //Mat pointSamples;
        processMat.convertTo(pointSamples, CV_32FC3, 1.0/255.0, 0.0);
        processMat.release();

        //return pointSamples;
    }

void VQ_KmeanKDTree::prepareDataWithOutUnique(Mat &srcImg, Mat
&pointSamples)
{
    //Mat pointSamples;
    srcImg.convertTo(pointSamples, CV_32FC3, 1.0/255.0, 0.0);
    pointSamples = pointSamples.reshape(3,
srcImg.rows*srcImg.cols);
    //return pointSamples;
}

void VQ_KmeanKDTree::setDataImageBGR(Mat &srcImg)
{
    int lenghtPixel = srcImg.rows * srcImg.cols;
    this->vectorPixelColorAll.resize(lenghtPixel);
    uchar* data = (uchar *) srcImg.data;
    int intBRG;
    for (int i = 0; i < lenghtPixel; i++)
    {
        intBRG = (int) (*(data++));
        intBRG += (int) (*(data++)) << 8;
        intBRG += (int) (*(data++)) << 16;
        this->vectorPixelColorAll[i] = intBRG;
    }
}

```

```

        this->uniqueData();
    }

void VQ_KmeanKDTree::uniqueData(void)
{
    sort( vectorPixelColorAll.begin(), vectorPixelColorAll.end() );
    vectorPixelColorAll.erase( unique( vectorPixelColorAll.begin(),
vectorPixelColorAll.end() ), vectorPixelColorAll.end() );
}
void VQ_KmeanKDTree::prepare2Double(void)
{
    int nPts = vectorPixelColorAll.size();
    this->vectorDoubleUniquePixelRGB.resize(nPts);
    int intBGR;
    int mask = 255;
    for (int j = 0; j < nPts; j++) {
        intBGR = vectorPixelColorAll[j];
        this->vectorDoubleUniquePixelRGB[j].B = ((double) (intBGR
& mask))/255.0;
        this->vectorDoubleUniquePixelRGB[j].G =
((double) ((intBGR>>8) & mask))/255.0;
        this->vectorDoubleUniquePixelRGB[j].R =
((double) ((intBGR>>16) & mask))/255.0;
    }
}

void VQ_KmeanKDTree::prepare2FloatForOpenCV(void)
{
    int nPts = vectorPixelColorAll.size();
    this->vectorFloatUniquePixelRGB.resize((nPts*3));

    int intBGR;
    int mask = 255;
    for (int j = 0; j < nPts; j++) {
        intBGR = vectorPixelColorAll[j];
        this->vectorFloatUniquePixelRGB[(j*3)] = ((float) (intBGR
& mask))/255.0f;
        this->vectorFloatUniquePixelRGB[(j*3+1)] =
((float) ((intBGR>>8) & mask))/255.0f;
        this->vectorFloatUniquePixelRGB[(j*3+2)] =
((float) ((intBGR>>16) & mask))/255.0f;
    }
}

void VQ_KmeanKDTree::openCVKmeansExecute(int cluster,int loopNumber)
{
    int cluster_count = cluster; /* number of cluster */
    this->prepare2FloatForOpenCV();

    Mat src_img = imread("c:""baboon.jpg");

    int lenghtPixel = src_img.rows * src_img.cols;

    vector<int> vectorIntTemp(lenghtPixel);

    MatIterator_<Vec3b> it = src_img.begin<Vec3b>(),
        it_end = src_img.end<Vec3b>();
    int tempRGBint;

    for(int i=0; it != it_end; ++it,++i)

```

```

{
    tempRGBint = (int)(*it)[0];
    tempRGBint += ((int)(*it)[1])<<8;
    tempRGBint += ((int)(*it)[2])<<16;
    vectorIntTemp[i] = tempRGBint;
}
cout << "non unique number :" << vectorIntTemp.size() << endl;
sort( vectorIntTemp.begin(), vectorIntTemp.end() );
vectorIntTemp.erase( unique( vectorIntTemp.begin(),
vectorIntTemp.end() ), vectorIntTemp.end() );
cout << "unique number :" << vectorIntTemp.size() << endl;

Mat processMat(vectorIntTemp.size(),1, CV_MAKETYPE(CV_8U,3));

it = processMat.begin<Vec3b>();
it_end = processMat.end<Vec3b>();

int intBGR;
int mask = 255;
for(int i=0; it != it_end; ++it,++i)
{
    intBGR = vectorIntTemp[i];
    (*it)[0] = saturate_cast<uchar>(intBGR & mask);
    (*it)[1] = saturate_cast<uchar>((intBGR>>8) & mask);
    (*it)[2] = saturate_cast<uchar>((intBGR>>16) & mask);
}

Mat pointSamples;
processMat.convertTo(pointSamples, CV_32FC3,1.0/255.0,0.0);
Mat<int> clusters(pointSamples.size(), CV_32SC1);
// (3)run k-means clustering algorithm to segment pixels in RGB
color space
//Mat<int> clusters(points.size(), CV_32SC1);
//cv::Mat centers(cluster_count,1,CV_32FC3);
cv::Mat centers;
//kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
1.0), 1, KMEANS_PP_CENTERS, &centers);
kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
1.0), 1, KMEANS_PP_CENTERS, &centers);
centers = centers.reshape(3, cluster_count);

MatIterator<Vec3f> itf = centers.begin<Vec3f>();
MatIterator<Vec3f> itf_end = centers.end<Vec3f>();

for(int i=0; itf != itf_end; ++itf,++i) {
    cout << (*itf)[0] << " ";
    cout << (*itf)[1] << " ";
    cout << (*itf)[2]<< endl;
}
}

bool VQ_KmeanKDTree::kmeansExecuteOpenCV(int cluster_count,int
loopNumber)
{
    if ((this->pointSamples.refcount==NULL))
        return false;

    Mat clusters = cv::Mat<int>(this-
>pointSamples.size(),CV_32SC1);

```

```

        // (3)run k-means clustering algorithm to segment pixels in RGB
        color space
        //kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
1.0), 1, KMEANS_PP_CENTERS, &centers);
        //kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
1.0), 1, KMEANS_USE_INITIAL_LABELS, &centers);
        cv::kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
0.5), 1, KMEANS_PP_CENTERS, &this->centerCluster);

        this->centerCluster = this->centerCluster.reshape(3,
cluster_count);
        MatIterator_<Vec3f> itf      = this-
>centerCluster.begin<Vec3f>();
        MatIterator_<Vec3f> itf_end = this->centerCluster.end<Vec3f>();

        this->eachClusterCount =
cv::Mat_<float>(cluster_count,1,CV_32FC1);
        this->eachClusterCount.setTo(cv::Scalar(0.0));
        MatIterator_<Vec<float, 1>> itff      = this-
>eachClusterCount.begin<Vec<float, 1>>();

        Mat tempCount = cv::Mat_<int>(cluster_count,1,CV_32SC1);
        tempCount.setTo(cv::Scalar(0.0));

        MatIterator_<Vec<int, 1>> itI      =
tempCount.begin<Vec<int, 1>>();
        MatIterator_<Vec<int, 1>> itI_end  = tempCount.end
<Vec<int, 1>>();
        MatIterator_<Vec<int, 1>> itItmp;

        MatIterator_<Vec<int, 1>> itIC      = clusters.begin<Vec<int,
1>>();
        MatIterator_<Vec<int, 1>> itIC_end = clusters.end<Vec<int,
1>>();

        for(int i = 0; itIC != itIC_end; ++itIC) {
            int temp = (*itIC)[0];
            itItmp = itI+temp;
            (*itItmp)[0]++;
        }

        float totalPixel = (float)this->pointSamples.rows;
        for(int i = 0; itI != itI_end; ++itI,++itff,++i) {
            (*itff)[0] = ((float)((*itI)[0]))/totalPixel;
            cout <<i<<" :: "<< (*itff)[0] << " ";
            cout << endl;
        }
        return true;
    }

bool VQ_KmeanKDTree::writeDataYAML(char *filename)
{
    if (this->centerCluster.refcount==NULL)
        return false;
    cv::FileStorage fs(filename, cv::FileStorage::WRITE);
    fs << "ClusterCenter" << this->centerCluster;
    fs << "ClusterWeight" << this->eachClusterCount;
    fs.release();
}

```

```

        return true;
    }

void VQ_KmeanKDTree::kmeansExecute(int clusterNumber, int loopNumber)
{
    KMterm term(loopNumber, 0, 0, 0,          // run for 100
stages      0.10, 0.10, 3,                  // other typical parameter
values      0.50, 10, 0.95);

    this->prepare2Double();
    int dim = 3;
    int nPts = this->vectorDoubleUniquePixelRGB.size();
    KMpointArray source = kmAllocPts(nPts, dim);
    for (int j = 0; j < nPts; j++)
    {
        source[j][0] = this->vectorDoubleUniquePixelRGB[j].B;
        source[j][1] = this->vectorDoubleUniquePixelRGB[j].G;
        source[j][2] = this->vectorDoubleUniquePixelRGB[j].R;
    }
    KMdata *dataPts = new KMdata(dim, nPts);
    kmCopyPts(nPts, dim, source, dataPts->getPts());

    kmDeallocPts(source);
    dataPts->buildKcTree();          // build filtering
structure

    KMfilterCenters *ctrs = new KMfilterCenters(clusterNumber,
*dataPts);          // allocate centers
    KMlocalEZ_Hybrid kmAlg(*ctrs, term);
    *ctrs = kmAlg.execute();          // execute

    double* sqDist = new double[nPts];
    KMctrIdxArray newCands = new KMctrIdx[nPts];
    ctrs->getAssignments(newCands, sqDist);
    eachClusterNumber.resize(clusterNumber);

    for (int j = 0; j < nPts; j++)
    {
        eachClusterNumber[newCands[j]]++;
    }
    delete []sqDist;
    delete []newCands;

    eachDoubleClusterCenter.resize(clusterNumber);
    KMcenterArray centerPoint = ctrs->getCtrPts();
    for (int j = 0; j < clusterNumber; j++)
    {
        eachDoubleClusterCenter[j].C0 = centerPoint[j][0];
        eachDoubleClusterCenter[j].C1 = centerPoint[j][1];
        eachDoubleClusterCenter[j].C2 = centerPoint[j][2];
    }
    delete ctrs;
    delete dataPts;

    for (int j = 0; j < clusterNumber; j++)
    {
        cout << eachDoubleClusterCenter[j].C0 << " , ";
        cout << eachDoubleClusterCenter[j].C1 << " , ";
        cout << eachDoubleClusterCenter[j].C2 << endl;
    }
}

```

```

    }
    return;
}

```

8.2.3 kmeanCluster.h

```

#include "opencv2/opencv.hpp"
#include <vector>
#include <algorithm>

using namespace cv;

#pragma once
class kmeanCluster
{
public:
    kmeanCluster(void);
    ~kmeanCluster(void);
    void setDataImageBGROpenCV(Mat &srcImg, bool isUnique);
    bool kmeansExecuteOpenCV(int cluster_count, int loopNumber);
    bool writeDataYAML(const char *filename);

private:
    Mat pointSamples;
    Mat centerCluster;
    Mat eachClusterCount;
    void prepareDataWithUnique(Mat &srcImg, Mat &pointSamples);
    void prepareDataWithOutUnique(Mat &srcImg, Mat &pointSamples);
};

```

8.2.4 kmeanCluster.cpp

```

#include "kmeanCluster.h"

kmeanCluster::kmeanCluster(void)
{
}

kmeanCluster::~~kmeanCluster(void)
{
}

void kmeanCluster::setDataImageBGROpenCV(Mat &srcImg, bool isUnique)
{
    if (isUnique)
    {
        this->prepareDataWithUnique(srcImg, this->pointSamples);
    } else
    {
        this->prepareDataWithOutUnique(srcImg, this->pointSamples);
    }
}

```

```

void kmeanCluster::prepareDataWithUnique(Mat &srcImg,Mat
&pointSamples)
{
    int lenghtPixel = srcImg.rows * srcImg.cols;
    vector<int> vectorIntTemp(lenghtPixel);

    MatIterator_<Vec3b> it = srcImg.begin<Vec3b>(),it_end =
srcImg.end<Vec3b>();

    int tempRGBInt;
    for(int i=0; it != it_end; ++it,++i)
    {
        tempRGBInt = (int)(*it)[0];
        tempRGBInt += ((int)(*it)[1])<<8;
        tempRGBInt += ((int)(*it)[2])<<16;
        vectorIntTemp[i] = tempRGBInt;
    }

    // unique process using STL
    sort( vectorIntTemp.begin(), vectorIntTemp.end() );
    vectorIntTemp.erase( unique( vectorIntTemp.begin(),
vectorIntTemp.end() ), vectorIntTemp.end() );

    Mat processMat(vectorIntTemp.size(),1, CV_MAKETYPE(CV_8U,3));

    it = processMat.begin<Vec3b>();
    it_end = processMat.end<Vec3b>();

    int intBGR;
    int mask = 255;
    for(int i=0; it != it_end; ++it,++i)
    {
        intBGR = vectorIntTemp[i];
        (*it)[0] = saturate_cast<uchar>(intBGR & mask);
        (*it)[1] = saturate_cast<uchar>((intBGR>>8) & mask);
        (*it)[2] = saturate_cast<uchar>((intBGR>>16) & mask);
    }
    //Mat pointSamples;
    processMat.convertTo(pointSamples, CV_32FC3,1.0/255.0,0.0);
    processMat.release();
    //return pointSamples;
}

void kmeanCluster::prepareDataWithOutUnique(Mat &srcImg,Mat
&pointSamples)
{
    srcImg.convertTo(pointSamples, CV_32FC3,1.0/255.0,0.0);
    pointSamples = pointSamples.reshape(3,
srcImg.rows*srcImg.cols);
}

bool kmeanCluster::kmeansExecuteOpenCV(int cluster_count,int
loopNumber)
{
    if ((this->pointSamples.refcount==NULL))

        return false;

    Mat clusters = cv::Mat_<int>(this-
>pointSamples.size(),CV_32SC1);

```

```

        // (3)run k-means clustering algorithm to segment pixels in RGB
        color space
        //kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
1.0), 1, KMEANS_PP_CENTERS, &centers);
        //kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
1.0), 1, KMEANS_USE_INITIAL_LABELS, &centers);
        cv::kmeans(pointSamples, cluster_count,
clusters,cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, loopNumber,
0.5), 1, KMEANS_PP_CENTERS, this->centerCluster);
        this->centerCluster = this->centerCluster.reshape(3,
cluster_count);
        MatIterator_<Vec3f> itf      = this-
>centerCluster.begin<Vec3f>();
        MatIterator_<Vec3f> itf_end = this->centerCluster.end<Vec3f>();

        this->eachClusterCount =
cv::Mat_<float>(cluster_count,1,CV_32FC1);
        this->eachClusterCount.setTo(cv::Scalar(0.0));
        MatIterator_<Vec<float, 1>> itff      = this-
>eachClusterCount.begin<Vec<float, 1>>();

        Mat tempCount = cv::Mat_<int>(cluster_count,1,CV_32SC1);
        tempCount.setTo(cv::Scalar(0.0));

        MatIterator_<Vec<int, 1>> itI      =
tempCount.begin<Vec<int, 1>>();
        MatIterator_<Vec<int, 1>> itI_end  = tempCount.end
<Vec<int, 1>>();
        MatIterator_<Vec<int, 1>> itItmp;

        MatIterator_<Vec<int, 1>> itIC      = clusters.begin<Vec<int,
1>>();
        MatIterator_<Vec<int, 1>> itIC_end = clusters.end<Vec<int,
1>>();

        for(int i = 0; itIC != itIC_end; ++itIC) {
            int temp = (*itIC)[0];
            itItmp = itI+temp;
            (*itItmp)[0]++;
        }

        float totalPixel = (float)this->pointSamples.rows;
        for(int i = 0; itI != itI_end; ++itI,++itff,++i) {
            (*itff)[0] = ((float)((*itI)[0]))/totalPixel;
        }
        return true;
    }

bool kmeanCluster::writeDataYAML(const char *filename)
{
    if (this->centerCluster.refcount==NULL)
        return false;
    cv::FileStorage fs(filename, cv::FileStorage::WRITE);
    fs << "ClusterCenter" << this->centerCluster;

    fs << "ClusterWeight" << this->eachClusterCount;
    fs.release();
}

```



```

        return true;
    }

```

8.3 โปรแกรม VqRetrieval

8.3.1 yamlWriteRead.h

```

#include "opencv2/opencv.hpp"
#include <vector>
#include <algorithm>
using namespace cv;

#pragma once
class yamlWriteRead
{
public:
    bool readDataVq(char* filename, Mat &centerCluster, Mat
&eachClusterCount);
    bool readDataVqConst(const char* filename, Mat &centerCluster,
Mat &eachClusterCount);
    bool writeDataYAML(char* filename, Mat &centerCluster, Mat
&eachClusterCount);
    yamlWriteRead(void);
    ~yamlWriteRead(void);
};

```

8.3.2 yamlWriteRead.cpp

```

#include "yamlWriteRead.h"

yamlWriteRead::yamlWriteRead(void)
{
}

yamlWriteRead::~yamlWriteRead(void)
{
}

bool yamlWriteRead::readDataVq(char* filename, Mat &centerCluster,
Mat &eachClusterCount)
{
    FileStorage fsRead(filename, FileStorage::READ);

    fsRead["ClusterCenter"] >> centerCluster;
    fsRead["ClusterWeight"] >> eachClusterCount;
    fsRead.release();
    return true;
}

bool yamlWriteRead::readDataVqConst(const char* filename, Mat
&centerCluster, Mat &eachClusterCount)
{
    FileStorage fsRead(filename, FileStorage::READ);

    fsRead["ClusterCenter"] >> centerCluster;
    fsRead["ClusterWeight"] >> eachClusterCount;
    fsRead.release();
    return true;
}

```

```

bool yamlWriteRead::writeDataYAML(char* filename, Mat &centerCluster,
Mat &eachClusterCount)
{
    cv::FileStorage fs(filename, cv::FileStorage::WRITE);
    fs << "ClusterCenter" << centerCluster;
    fs << "ClusterWeight" << eachClusterCount;
    fs.release();
    return true;
}

```

8.3.3 VqRetrieval.cpp

```

#include "yamlWriteRead.h"
#include <opencv2/opencv.hpp>
#include <emmintrin.h>
#include <vector>
#include <string>
#include <libmysqlwrapped.h>
#include <mysql.h>

#include <iostream>                // for std::cout
#include <utility>                  // for std::pair
#include <algorithm>                 // for std::for_each
#include <boost/graph/graph_traits.hpp>
#include <boost/graph/adjacency_list.hpp>
#include <boost/graph/dijkstra_shortest_paths.hpp>

#include <boost/config.hpp>
#include <iostream>
#include <boost/graph/prim_minimum_spanning_tree.hpp>

#include <vector>
#include <math.h>

#include "kmeanCluster.h"
using namespace boost;

using namespace cv;
using namespace std;

struct Centet3Dfloat {
    float c0;
    float c1;
    float c2;
};

struct dbInfo{
    long Id;
    string imagePath;
    string yamlPath;
};

struct dbConnect{
    string dbHost;
    string dbUser;
    string dbPass;
    string dbName;
};

vector<dbInfo> readDataDbMysql(dbConnect dbConn,string DBID);

```

```

void MWWTestCompute(Mat & sampleMat, Mat & compareMat, Mat
&samplePopulaWeight, Mat & comparePopulaWeight, int
&MWWTestVaule, double & MWWTestWeightVaule);
void MWWTestWeightCompute(void);

int main(int argc, char **argv)
{
    // 0 name program
    // 1 filepath image input
    // 2 cluster number
    // 3 loop number
    // 4 db file for sqlite
    // 5 flag for 0 for MWWTest other for MWWTestWeight

    if (argc != 10)
        return -1;

    const char *imagename = argv[1];
    const int clusterNum = atoi(argv[2]);
    const int loopNum = atoi(argv[3]);
    const int flagChooseMWW = atoi(argv[4]);

    dbConnect dbConn;
    string DBID;
    dbConn.dbHost = argv[5];
    dbConn.dbUser = argv[6];
    dbConn.dbPass = argv[7];

    if (dbConn.dbPass.compare("null")==0 ||
dbConn.dbPass.compare("NULL")==0)
    {
        dbConn.dbPass = "";
    }

    dbConn.dbDBName = argv[8];
    DBID = argv[9];

    if (DBID.compare("null")==0 || DBID.compare("NULL")==0)
    {
        DBID = "";
    }

    if (!(clusterNum < 100 && clusterNum > 1 && loopNum < 200 &&
loopNum > 10))
        return -1;

    Mat src_img = imread(imagename, 1);
    if(!src_img.data || src_img.channels()!=3)
        return -1;

    kmeanCluster kCluster;
    kCluster.setDataImageBGROpenCV(src_img, false);

    if (!(kCluster.kmeansExecuteOpenCV(clusterNum, loopNum)))
        return -1;

    Mat sampleCenterCluster;
    Mat sampleEachClusterCount;

```

```

t);
kCluster.getKmeanData(sampleCenterCluster,sampleEachClusterCoun
t);
yamlWriteRead yamlWR;

vector<dbInfo> vecListFile = readDataDbMysql(dbConn,DBID);

if (vecListFile.size()<1)
{
    return -1;
}

vector<std::pair<Mat,Mat>> vqDataAllImage(vecListFile.size());

vector<dbInfo>::iterator it;
vector<std::pair<Mat,Mat>>::iterator itYaml;

map<int,string> MWWTestMapList;
map<double,string> MWWTestWeightMapList;

for ( it=vecListFile.begin(),itYaml = vqDataAllImage.begin() ;
it != vecListFile.end(); it++, itYaml++ )
{
    string finalPath = it->yamlPath;

    Mat centerCluster;
    Mat eachClusterCount;
    yamlWR.readDataVqConst( finalPath.c_str()
,centerCluster,eachClusterCount);

    int MWWTestVaule = 0;
    double MWWTestWeightVaule = 0.0f;
    MWWTestCompute(sampleCenterCluster, centerCluster,
sampleEachClusterCount, eachClusterCount, MWWTestVaule,
MWWTestWeightVaule);
    MWWTestMapList[MWWTestVaule] = it->imagePath;
    MWWTestWeightMapList[MWWTestWeightVaule] = it->imagePath;
}
int i =0;
if (flagChooseMWW)
{
    map<double,string>::reverse_iterator ritDouble;
    // show content:
    for ( ritDouble=MWWTestWeightMapList.rbegin() ; ritDouble
!= MWWTestWeightMapList.rend() && i<10 ; ritDouble++ ,i++)
        cout << ritDouble->first << ";" << ritDouble->second << endl;
    return 1;
}
map<int,string>::reverse_iterator ritInt;
// show content:
for ( ritInt=MWWTestMapList.rbegin() ; ritInt !=
MWWTestMapList.rend() && i<10; ritInt++ ,i++)
    cout << ritInt->first << ";" << ritInt->second << endl;
return 1;
}

vector<dbInfo> readDataDbMysql(dbConnect dbConn,string DBID)
{
    vector<dbInfo> vecResult;
    Database
db(dbConn.dbHost,dbConn.dbUser,dbConn.dbPass,dbConn.dbDBName);

```

```

Query q(db);

if (!db.Connected() || !q.Connected())
    return vecResult;
// retrieve data
string sql = "SELECT IDImage,imagePath,yamlPath from filedata";
if (DBID.compare("")!=0)
{
    sql = "SELECT IDImage,imagePath,yamlPath from filedata
WHERE DBID = "+ DBID;
}
q.get_result(sql);

while (q.fetch_row())
{
    dbInfo tmpAdd;
    tmpAdd.Id          = q.getval();
    tmpAdd.imagePath   = q.getstr();
    tmpAdd.yamlPath    = q.getstr();
    vecResult.push_back(tmpAdd);
}
q.free_result();
return vecResult;
}

void MWWTestCompute(Mat & sampleMat,Mat & compareMat,Mat
&samplePopulaWeight,Mat & comparePopulaWeight,int
&MWWTestVaule,double & MWWTestWeightVaule)
{
    if (sampleMat.rows != compareMat.rows)
        return;

    int num_nodesMat          = sampleMat.rows + compareMat.rows;
    int num_EdgeCompleteMat   = (num_nodesMat*(num_nodesMat-1))/2;

    //first index of vertex
    //first index of vertex too
    typedef std::pair < int, int > EMat;

    //first index of all vertex
    //second type of each vertex
    //typedef std::pair < int, int > VMat;
    vector<EMat> allEdgesMat(num_EdgeCompleteMat);

    vector<int>    vertexInput(num_nodesMat);
    vector<Vec3f>  vertexInputVec3f(num_nodesMat);
    vector<float>  vertexInputWeightf(num_nodesMat);

    MatIterator_<Vec3f> sampleItf          = sampleMat.begin<Vec3f>();
    MatIterator_<Vec3f> sampleItf_end      = sampleMat.end<Vec3f>();
    MatIterator_<Vec3f> compareItf         = compareMat.begin<Vec3f>();
    MatIterator_<Vec3f> compareItf_end     = compareMat.end<Vec3f>();

    MatIterator_<Vec<float, 1>> sampleWeightItf      =
samplePopulaWeight.begin<Vec<float, 1>>();
    MatIterator_<Vec<float, 1>> compareWeightItf      =
comparePopulaWeight.begin<Vec<float, 1>>();

    int countVertex = 0;

```

```

        for(; sampleItf != sampleItf_end;
++sampleItf,++sampleWeightItf) {
            vertexInput[countVertex]                = 0;
            vertexInputVec3f[countVertex]            = *sampleItf;
            vertexInputWeightf[countVertex]          =
(*sampleWeightItf)[0];
            countVertex++;
        }
        for (; compareItf != compareItf_end;
++compareItf,++compareWeightItf){
            vertexInput[countVertex]                = 1;
            vertexInputVec3f[countVertex]            = *compareItf;
            vertexInputWeightf[countVertex]          =
(*compareWeightItf)[0];
            countVertex++;
        }

        int countEdge    = 0;
        vector<double> weightsMat(num_EdgeCompleteMat);

        for(int i=0;i< num_nodesMat ;i++)
        {
            for(int j=i+1;j<num_nodesMat;j++)
            {
                allEdgesMat[countEdge].first  = i;
                allEdgesMat[countEdge].second = j;
                Vec3f tmpSrc  = vertexInputVec3f[i];
                Vec3f tmpDst  = vertexInputVec3f[j];

                double dim00 = (tmpSrc[0] - tmpDst[0]);
                double dim01 = (tmpSrc[1] - tmpDst[1]);
                double dim02 = (tmpSrc[2] - tmpDst[2]);

                weightsMat[countEdge] =
sqrt((dim00*dim00)+(dim01*dim01)+(dim02*dim02));
                countEdge++;
            }
        }

        typedef adjacency_list < vecS, vecS,
undirectedS,property<vertex_distance_t, double>, property <
edge_weight_t, double > > Graph;

        Graph g(num_nodesMat);
        property_map<Graph, edge_weight_t>::type weightmap =
get(edge_weight, g);

        int sizeofEdge = num_EdgeCompleteMat;

        for (std::size_t j = 0; j < sizeofEdge ; ++j) {
            graph_traits<Graph>::edge_descriptor e;
            bool inserted;
            boost::tie(e, inserted) = add_edge(allEdgesMat[j].first,
allEdgesMat[j].second, g);
            weightmap[e] = weightsMat[j];
        }

        std::vector < graph_traits < Graph >::vertex_descriptor >
p(num_vertices(g));
        property_map<Graph, vertex_distance_t>::type distance =
get(vertex_distance, g);

```

```

        property_map<Graph, vertex_index_t>::type indexmap      =
        get(vertex_index, g);
        prim_minimum_spanning_tree(g, *vertices(g).first, &p[0],
        distance, weightmap, indexmap, default_dijkstra_visitor());

        int MWWValueInner = 0;
        for (std::size_t i = 0; i != p.size(); ++i)
            (p[i] != i) && (vertexInput[i] != vertexInput[p[i]]) ?
MWWValueInner++ : 0 ;

        vector<vector<int>> parentNode(num_nodesMat);
        for (std::size_t i = 0; i != p.size(); ++i)
            if (p[i] != i ){
                parentNode[p[i]].push_back(i);
            }
        float MWWTestWeightInner = 0.0f;
        for (int i=0 ; i<num_nodesMat ;++i){
            // (term00 + term01) / term02
            // term00 = a0 + a1
            // term01 = b0 + b1
            // term02 = a0+a1+b0+b1
            int tmpComp = vertexInput[i];
            if (tmpComp == 0){
                float term00 = vertexInputWeightf[i];
                float term01 = 0.0f;
                float term02 = vertexInputWeightf[i];
                vector<int> tempInner = parentNode[i];
                if (tempInner.size() > 0)
                {
                    for (int j =0; j < tempInner.size();j++){
                        float tmpValue =
vertexInputWeightf[tempInner[j]];
                        term02 += tmpValue;
                        if (vertexInput[tempInner[j]] ==
tmpComp){
                            term00 += tmpValue;
                        }else{
                            term01 += tmpValue;
                        }
                    }
                    if (term02>0.0f)
                        MWWTestWeightInner +=
(term00*term01)/term02;
                }
            }
        }
        MWWTestVaule      = MWWValueInner ;
        MWWTestWeightVaule = MWWTestWeightInner;
        return;
    }

```

8.4 โปรแกรม Web Page สำหรับเก็บภาพลงในฐานข้อมูลภาพ

8.4.1 DBCommand.php

```
<?php
session_start();

$info = pathinfo($_SERVER['REQUEST_URI']);
$path="http://".$_SERVER['SERVER_NAME'].$info['dirname'];

if (empty($info['extension']))

    $path.="/".$info['basename'];

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="fileuploader.css" rel="stylesheet" type="text/css"
media="all" />

<title>Untitled Document</title>

<style type="text/css">

    #DBSelect { font-family: AngsanaUPC;

                font-size: 30px;

                font-weight: bolder;

                color: #00F;

                background-color: #CCC;

                text-align: center;

                margin: 0px;

                padding: 0px;

                }

    #CreateDatabase {

                font-family: AngsanaUPC;
```



```

        font-size: 30px;

        font-weight: bolder;

        color: #00F;

        background-color: #CCC;

        text-align: center;

        margin: 0px;
            81

        padding: 0px;

        border-top-style: solid;

        border-right-style: solid;

        border-bottom-style: solid;

        border-left-style: solid;
    }

    #SelectDB {

        text-align: right;

        font-family: AngsanaUPC;

        font-size: 30px;

        font-weight: bolder;

    }

    #Image #InputImage {

        font-family: AngsanaUPC;

        font-size: 30px;

        font-weight: bolder;

        color: #00F;

        background-color: #CCC;

        text-align: center;

        margin: 0px;

        padding: 0px;

    }

```

```

#apDiv1 {

    position:absolute;

    left:290px;

    top:820px;
        82

    width:494px;

    height:155px;

    z-index:1;

}

#JqPostForm fieldset legend {

    text-align: left;

}

</style>

</head>

<body>

<label for="jumpmenu"></label>

<div>

<div id="DBSelect">

<form id="JqPostForm">

<fieldset>

<legend>เลือกฐานข้อมูล</legend>

<form id="form2" name="form2" method="post" action="" >

<select name="datafolder" id="datafolder" style="width:150px"

onchange="location.href='DBCommand.php?select='+this.value;">

<?php

@session_start();

session_unset();

```

```

mysql_connect("localhost","root","") or die(mysql_error());
mysql_select_db("TCBIR");
$query="SELECT DBID,Name FROM groupdatabase";
$result = mysql_query($query);
$checkValue = $_GET['select'];
while($nt=mysql_fetch_array($result))
{
    if ($checkValue == $nt[Name] || $checkValue =="" )
    {
        echo "<option selected value =
$nt[Name]>$nt[Name]</option>";

        $checkValue = $nt[Name];
        $checkValueID = $nt[DBID];
    }else
    {
        echo "<option value = $nt[Name]>$nt[Name]</option>";
    }
}
?>

</select>

</form>

<fieldset>

<p>ป้อนภาพลงในฐานข้อมูล </p>

<div id="file-uploader">

<script src="fileuploader.js" language="javascript1.1"></script>

<script>

function createUploader()

{

```

```

var uploader = new qq.FileUploader({
    // pass the dom node (ex. $(selector)[0] for jQuery users)
    element: document.getElementById('file-uploader'),
    // path to server-side upload script. In our case server/php.php
    action: 'php.php',
    // additional data to send, name-value pairs
    params: {
        dataBaseSelect: <?php echo "'".$checkValue."' ?>,
        dataBaseSelectID: <?php echo "'".$checkValueID."' ?>
    }
    //debug: true
});
}

window.onload = createUploader;

</script>

<script type="text/javascript">

var img_id=0

var image = new Array()

document.getElementById('send').onclick=function()

{

    img_id++

    var id="imgid"+img_id

    image = document.getElementById('file-uploader').value;

    document.getElementById('div').innerHTML="<img id='"+id+"'
src='"+image+"' width=500px height=200px>"

}

</script>

</div>

</fieldset>

```

```

</form>

<fieldset>

<div id="CreateDatabase">

<div>

<p>สร้างฐานข้อมูลใหม่</p>

<form id="form1" name="form1" method="post" action="Newtable.php">

<p>

<label for="Namedatabase">สร้างฐานข้อมูล: </label>

<input type="text" name="Namedatabase" id="Namedatabase" />

</p>

<p>

<label for="Detailofdatabase">ข้อมูลของฐานข้อมูล:</label>

<textarea name="Detailofdatabase" id="Detailofdatabase" cols="45"
rows="5"></textarea>

</p>

<p>

<input type="submit" name="submit" id="submit" value="ยืนยันการสร้าง"/>

</p>

</form>

<p>&nbsp;</p>

</form>

</fieldset>

</body>
</html>

```

8.4.2 Php.php

```

<?php
require_once('Connections/conn.php');
session_start();

/**

```

```

* Handle file uploads via XMLHttpRequest
*/

class qqUploadedFileXhr {

/**

* Save the file to the specified path
* @return boolean TRUE on success
*/

function save($path) {

$input = fopen("php://input", "r");

$temp = tmpfile();

$realSize = stream_copy_to_stream($input, $temp);

fclose($input);

if ($realSize != $this->getSize()){

    return false;

}

$target = fopen($path, "w");

fseek($temp, 0, SEEK_SET);

stream_copy_to_stream($temp, $target);

fclose($target);

return true;

}

function getName() {

return $_GET['qqfile'];

}

function getSize() {

if (isset($_SERVER["CONTENT_LENGTH"])){

    return (int)$_SERVER["CONTENT_LENGTH"];

} else {

```

```

        throw new Exception('Getting content length is not supported.');
```

```

    }

}

}

/**
 * Handle file uploads via regular form post (uses the $_FILES array)
 */

class qqUploadedFileForm {

    /**
     * Save the file to the specified path
     * @return boolean TRUE on success
     */

    function save($path) {

        if(!move_uploaded_file($_FILES['qqfile']['tmp_name'], $path)){

            return false;

        }

        return true;

    }

    function getName() {

        return $_FILES['qqfile']['name'];

    }

    function getSize() {

        return $_FILES['qqfile']['size'];

    }

}

```

```

class qqFileUploader {

private $allowedExtensions = array();

private $sizeLimit = 10485760;

private $file;

function __construct(array $allowedExtensions = array(), $sizeLimit =
10485760){

$allowedExtensions = array_map("strtolower", $allowedExtensions);

$this->allowedExtensions = $allowedExtensions;

$this->sizeLimit = $sizeLimit;


$this->checkServerSettings();

if (isset($_GET['qqfile'])) {

//do this part

        $this->file = new qqUploadedFileXhr();

    } elseif (isset($_FILES['qqfile'])) {

        $this->file = new qqUploadedFileForm();

    } else {

        $this->file = false;

    }

}

private function checkServerSettings(){

$postSize = $this->toBytes(ini_get('post_max_size'));

$uploadSize = $this->toBytes(ini_get('upload_max_filesize'));

if ($postSize < $this->sizeLimit || $uploadSize < $this->sizeLimit){

        $size = max(1, $this->sizeLimit / 1024 / 1024) . 'M';

```



```

        die("{\"error':'increase post_max_size and upload_max_filesize to
$size'}");
    }
}

private function toBytes($str){
    $val = trim($str);
    $last = strtolower($str[strlen($str)-1]);
    switch($last) {
        case 'g': $val *= 1024;
        case 'm': $val *= 1024;
        case 'k': $val *= 1024;
    }
    return $val;
}

/**
 * Returns array('success'=>true) or array('error'=>'error message')
 */

function handleUpload($uploadDirectory, $replaceOldFile = FALSE){
    if (!is_writable($uploadDirectory)){
        return array('error' => "Server error. Upload directory isn't
writable.");
    }

    if (!$this->file){
        return array('error' => 'No files were uploaded.');
```

```

if ($size == 0) {
    return array('error' => 'File is empty');
}

if ($size > $this->sizeLimit) {
    return array('error' => 'File is too large');
}

$pathinfo = pathinfo($this->file->getName());

//filename
$filename = $pathinfo['filename'];
//echo $filename ;
//$filename = $pathinfo['filename'];
//$filename = md5(uniqid());
$ext = $pathinfo['extension'];

if($this->allowedExtensions && !in_array(strtolower($ext), $this->allowedExtensions)){
    $these = implode(', ', $this->allowedExtensions);
    return array('error' => 'File has an invalid extension, it should be one of '. $these . '.');
}

if(!$replaceOldFile){
    /// don't overwrite previous files that were uploaded
    while (file_exists($uploadDirectory . $filename . '.' . $ext)) {
        $filename .= rand(10, 99);
    }
}

```

```

}

try {

    /*** connect to SQLite database ***/
    92

    $hostname_conn = "localhost";

    $database_conn = "ae";

    $username_conn = "root";

    $password_conn = "";

    $dbh = new
PDO("mysql:host=$hostname_conn;dbname=$database_conn", $username_conn,
$password_conn);

    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $dbh->beginTransaction();

    $DBIDName = $_GET['dataBaseSelect'];

    $DBIDvalueID = $_GET['dataBaseSelectID'];

    $fullFileNameWithPath = $uploadDirectory . $filename . '.' . $ext;

    $yamlFilePath = "yaml/". $filename . '.yaml';

    $count = $dbh->exec("INSERT INTO
filedata(DBID,imagePath,yamlPath)
VALUES('$DBIDvalueID','$fullFileNameWithPath','$yamlFilePath')");

    if ($this->file->save($uploadDirectory . $filename . '.' . $ext)){

        $return_var = 0;

        $out = array();

        $exec = exec("VqKmeanCalculate.exe ".$fullFileNameWithPath."
".$yamlFilePath . " 60 30",$out,$return_var);

```

```

        if($return_var < 0)
        {
            unlink($fullFileNameWithPath);

            $dbh->rollback();

            $dbh = null;

            return array('error'=> 'Could not produce yaml file.');
```

}

```

        $resultBool = $dbh->commit();

        $dbh = null;

        return array('success'=>true);
    } else {
        $dbh->rollback();

        $dbh = null;

        return array('error'=> 'Could not save uploaded file.' .

            'The upload was cancelled, or server error encountered');
```

}

```

    }

    catch(PDOException $e)
    {
        $dbh->rollback();

        $dbh = null;
    }
}

$allowedExtensions = array();

// max file size in bytes

$sizeLimit = 10 * 1024 * 1024;
```

```

$uploader = new qqFileUploader($allowedExtensions, $sizeLimit);

$DBIDName = $_GET['dataBaseSelect'];

$Class2 = "Image/allMix/";

$result = $uploader->handleUpload($Class2);

echo htmlspecialchars(json_encode($result), ENT_NOQUOTES);

?>

```

8.5 โปรแกรม Web Page สำหรับค้นหาภาพในฐานข้อมูล

8.5.1 SearchInner.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>class.upload.php test forms</title>

<style>

fieldset {

    width: 50%;

    margin: 15px 0px 25px 0px;

    padding: 15px;

}

legend {

    font-weight: bold;

}

.button {

    text-align: right;

}

.button input {

    font-weight: bold;

}

```

```

body {

    background-color: #F96;

}

</style>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

</head>

<body>

<h1>Color Image Retrieval System</h1>

<fieldset>

<legend>Image sample</legend>

<p>Pick up image to upload and press upload </p>

<form    name="form2"    enctype="multipart/form-data"    method="post"
action="upload.php" />

<p><input type="file" size="32" name="my_field" value="" /></p>

<p class="button"><input type="hidden" name="action" value="image" />

<input type="submit" name="Submit" value="upload to query" /></p>

</form>

<p>&nbsp;</p>

<p>&nbsp;</p>

</fieldset>

</body>

</html>

```

8.5.2 Retrieval.php

```

<?php

session_start();

error_reporting(E_ALL);

```

```

// recieve namedata form select to query each folder

$DBIDValue = "null" ;

// we first include the upload class, as we will need it here to deal
with the uploaded file

include('class.upload.php');

// retrieve eventual CLI parameters

$cli = (isset($argc) && $argc > 1);

if ($cli) {

    if (isset($argv[1])) $_GET['file'] = $argv[1];

    if (isset($argv[2])) $_GET['dir'] = $argv[2];

    if (isset($argv[3])) $_GET['pics'] = $argv[3];

}

// set variables

$dir_dest = (isset($_GET['dir']) ? $_GET['dir'] : 'sampleImage');

$dir_pics = (isset($_GET['pics']) ? $_GET['pics'] : $dir_dest);

if (!$cli) {

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<meta http-equiv=Content-Type content="text/html; charset=utf-8">

<head>

<title>Color Image Retrieval System</title>

<style>

body {

}

fieldset {

```

```

        width: 50%;

        background: url(bg.gif);

        margin: 15px 0px 25px 0px;

        padding: 15px;
    }

    legend {

        font-weight: bold;

    }

    fieldset img {

        float: right;

    }

    fieldset p {

        font-size: 70%;

        font-style: italic;

    }

    .button {

        text-align: right;

    }

    .button input {

        font-weight: bold;

    }

</style>

</head>

<body bgcolor="#FF6666">

<h1>Color Image Retrieval System :: viewing images retrieve</h1>

<?php

}

```



```

if      ((isset($_POST['action'])      ?      $_POST['action']      :
(isset($_GET['action']) ? $_GET['action'] : '')) == 'image') {

// ----- IMAGE UPLOAD -----

// we create an instance of the class, giving as argument the PHP object
// corresponding to the file field from the form

// All the uploads are accessible from the PHP object $_FILES

$handle = new Upload($_FILES['my_field']);

// then we check if the file has been uploaded properly

// in its *temporary* location in the server (often, it is /tmp)

if ($handle->uploaded) {

// yes, the file is on the server

// below are some example settings which can be used if the uploaded
file is an image.

// now, we start the upload 'process'. That is, to copy the uploaded
file

// from its temporary location to the wanted location

//      It      could      be      something      like      $handle-
>Process('/home/www/my_uploads/');

$handle->Process($dir_dest);

// we check if everything went OK

if ($handle->processed) {

    // everything was fine !

    echo '<fieldset>';

    echo '  <legend>sample file uploaded with success</legend>';

    echo '  ';

    $info = getimagesize($handle->file_dst_pathname);

    echo '  link to the file : <a href="'. $dir_pics . '/' . $handle-
>file_dst_name . '"> . $handle->file_dst_name . '</a><br/>';

    echo '</fieldset>';

```

```

} else {

    // one error occurred

    echo '<fieldset>';

    echo '      <legend>file      not      uploaded      to      the      wanted
location</legend>';

    echo '  Error: ' . $handle->error . ' ';

    echo '</fieldset>';

}

// we delete the temporary files

$handle-> Clean();

$return_var = 0;

$out = array();

$exec = exec("VqRetrieval.exe ".$dir_pics.'/ ' . $handle->file_dst_name
. " 60 30 1 127.0.0.1 root null ae $DBIDValue",$out,$return_var);

if($return_var > 0)

    foreach ($out as $line) { // process array line by line

        $pieces = explode(";", $line);

        $fileSplitPath = explode("/", $pieces[1]);

        $countData = count($fileSplitPath);

        echo '<fieldset>';

        echo '      <legend>Retrieve      image      score      :=
'.$pieces[0].</legend>';

        echo '  ';

        $info = getimagesize($handle->file_dst_pathname);

        echo '  link to the file : <a href="'. $pieces[1] . '">' .
$fileSplitPath[$countData-1] . '</a><br/>';

        echo '</fieldset>';

    }

```

```
} else {  
    echo '<fieldset>';  
    echo '  <legend>file not uploaded on the server</legend>';  
    echo '  Error: ' . $handle->error . ' ';  
    echo '</fieldset>';  
}  
  
}  
  
if (!$cli) {  
    echo '<p><a href="Seachpage.html">do another test</a></p>';  
?  
  
</body>  
  
</html>  
  
<?php  
}  
?>
```

บรรณานุกรม/เอกสารอ้างอิง

- [1] R. Jain, "Nsf workshop on visual information management systems," SIGMOD Rec., vol. 22, no. 3, pp. 57-75, 1993.
- [2] A. E. Cawkill, "The british library's picture research projects: Image, word, and retrieval," Advanced Imaging, vol. 8, no. 10, pp. 38-40, Oct. 1993.
- [3] J. Dowe, "Content-based retrieval in multimedia imaging," Proc. SPIE Storage and Retrieval for Image and Video Database, vol. 1908, no. 164, 01 Feb. 1993.
- [4] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber, "Efficient and effective querying by image content," Journal of Intelligent Information Systems, vol. 3, pp. 231-262, 1994.
- [5] Y. Gong, H. J. Zhang, and T. C. Chua, "An image database system with content capturing and fast image indexing abilities," Proc. IEEE International Conference on Multimedia Computing and Systems, pp. 121-130, 14-19 May 1994.
- [6] H. J. Zhang and D. Zhong, "A scheme for visual feature-based image indexing," Proc. of SPIE conf. on Storage and Retrieval for Image and Video Databases III, pp. 36-46, Feb. 1995.
- [7] R. Picard and T. P. Minka, "Vision texture for annotation," Multimedia Systems: Special Issue on Contentbased Retrieval, vol. 3, no. 1, pp. 3-14, 1995.
- [8] T. P. Minka and R. W. Picard, "Interactive learning using a society of models," in IEEE Int. Conf. on Computer Vision and Pattern Recognition, 1996, pp. 447-452.
- [9] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Target testing and the pichunter bayesian multimedia retrieval system," in Advanced Digital Libraries Forum, May.
- [10] ___, "Pichunter: Bayesian relevance feedback for image retrieval," in Intl. Conf. on Pattern Recognition.
- [11] I. J. Cox, M. L. Miller, T. P. Minka, and P. N. Yianilos, "An optimized interaction strategy for bayesian relevance feedback," in IEEE Conf. CVPR, 1998.
- [12] T. V. Papathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos, "Psychophysical studies of the performance of an image database retrieval system," in IS&T/SPIE Conf. on Human Vision and Electronic Imaging III, 1998.

- [13] Y. Rui, T.S.Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in mars," vol. 2, 1997, pp. 815-818.
- [14] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," IEEE Trans. on Circuits and Systems for Video Technology, 1998.
- [15] Y. Rui, T. S. Huang, and S. F. Chang, "Image retrieval: current techniques, promising directions and open issues," Journal of Visual Communication and Image Representation, vol. 10, pp. 39-62, 1999.
- [16] C. Böhm, S. Berchtold, and D. A. Keim, "Searching in high-dimensional spaces : Index structures for improving the performance of multimedia databases," ACM Comput. Surv., vol. 33, no. 3, pp. 322-373, 2001.
- [17] J. R. Smith and S.-F. Chang, "Visualeek: A fully automated content-based image query system," in Proc. ACM Multimedia 96, 1996.
- [18] J. Huang, S. R. Kumar, M. Mitra, W. J. Zhu, and R. Zabih, "Image indexing using color correlogram," IEEE Int. Conf. on Computer Vision and Pattern Recognition, Puerto Rico, pp. 762-768, June 1997.
- [19] J. Huang, S. Kumar, M. Metra, W. J. Zhu, and R. Zabith, "Spatial color indexing and applications," Intl J. Computer Vision, vol. 35, no. 3, pp. 245-268, 1999.
- [20] M. Ioka, "A method of de_ning the similarity of images on the basis of color information," Technical Report RT-0030, IBM Tokyo Research Laboratory, Nov. 1989.
- [21] A. K. Jain, "Fundamental of Digital Image Processing," Prentice Hall, 1989.
- [22] E. Mathias, "Comparing the influence of color spaces and metrics in content-based image retrieval," in Proceedings of International Symposium on Computer Graphics, Image Processing, and Vision, 1998, pp. 371-378.
- [23] G. Pass and R. Zabith, "Comparing images using joint histograms," Multimedia Systems, vol. 7, pp. 234-240, 1999.
- [24] M. Stricker and M. Orengo, "Similarity of color images," SPIE Storage and Retrieval for Image and Video Databases III, vol. 2185, pp. 381-392, Feb.
- [25] M. Swain and D. Ballard, "Color indexing," Int'l J. Computer Vision, vol. 7, no. 1, pp. 11-32.
- [26] H. J. Zhang, Y. Gong, C. Y. Low, and S. W. Smoliar, "Image retrieval based on color features: An evaluation study," in SPIE Conf. on Digital Storage and Archival, Oct.

- [27] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The qbic system," *IEEE Computer*, vol. 28, no. 9, pp. 23-32, Sept.
- [28] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "Querying images by content, using color, texture, and shape," in *SPIE Conference on Storage and Retrieval for Image and Video Database*, vol. 1908, April 1993, pp. 173-187.
- [29] G. Pass and R. Zabith, "Histogram refinement for content-based image retrieval," *IEEE Workshop on Applications of Computer Vision*, pp. 96-102, 1996.
- [30] H. Tamura, S. Mori, and T. Yamawaki, "Texture features corresponding to visual perception," *IEEE Trans. On Systems, Man, and Cybernetics*, vol. Smc-8, no. 6, June 1978.
- [31] J. M. Francos, "Orthogonal decompositions of 2d random fields and their applications in 2d spectral estimation," *Signal Processing and its Application*, pp. 207-227, 1993.
- [32] F. Liu and R. W. Picard, "Periodicity, directionality, and randomness: Wold features for image modeling and retrieval," *IEEE Trans. on Pattern Analysis and Machine Learning*, vol. 18, no. 7, July 1996.
- [33] A. K. Jain and F. Farroknia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*.
- [34] J. G. Daugman, "Complete discrete 2d gabor transforms by neural networks for image analysis and compression," *IEEE Trans. ASSP*, vol. 36, pp. 1169-1179, July, 1998.
- [35] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Information Theory*, vol. 36, pp. 961-1005, Sept. 1990.
- [36] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674-693, July 1989.
- [37] T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree structured wavelet transform," *IEEE Trans. on Image Processing*, vol. 2, no. 4, pp. 429-441, October 1993.
- [38] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1186-1191, Nov. 1993.
- [39] J. E. Gary and R. Mehrotra, "Shape similarity-based retrieval in image database systems," in *Proc. of SPIE, Image Storage and Retrieval Systems*, vol. 1662, 1992, pp. 2-8.

- [40] W. I. Grosky and R. Mehrotra, "Index based object recognition in pictorial data management," *CVGIP*, vol. 52, no. 3, pp. 416-436, 1990.
- [41] H. V. Jagadish, "A retrieval technique for similar shapes," in *Proc. of Int. Conf. on Management of Data, SIGMOID91*, May 1991, pp. 208-217.
- [42] D. Tegolo, "Shape analysis for image retrieval," in *Proc. of SPIE, Storage and Retrieval for Image and Video Databases II*, no. 2185, Feb. 1994, pp. 59-69.
- [43] H. V. Jagadish, "A retrieval technique for similar shapes," in *Proc. of Int. Conf. on Management of Data, SIGMOID91*, May 1991, pp. 208-217.
- [44] E. M. Arkin, L. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209-226, 1991.
- [45] S. Sclaroff and A. Pentland, "Modal matching for correspondence and recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 545-561, June 1995.
- [46] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant fourier descriptors to recognition of 3d objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 640-647, 1990.
- [47] H. Kauppinen, T. Seppnen, and M. Pietikinen, "An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 17, no. 2, pp. 201-207, 1995.
- [48] E. Persoon and K. Fu, "Shape discrimination using fourier descriptors," *IEEE Trans. Syst., Man, and Cybern.*, vol. 7, pp. 170-179, 1977.
- [49] M. K. Hu, "Visual pattern recognition by moment invariants," *IEEE Trans. Information Theory*, vol. 8, no. 2, pp. 179-187, 1962.
- [50] L. Yang and F. Algreysen, "Fast computation of invariant geometric moments: A new method giving correct results," in *Proc. IEEE Int. Conf. on Image Processing*, 1994.
- [51] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," *Multimedia Systems*, vol. 7, no. 3, pp. 184-198, 1999.
- [52] ___, "Edge flow: a framework of boundary detection and image segmentation," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 1997, pp. 744-749.
- [53] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: a system for region-based image indexing and retrieval," in *In Third International Conference on Visual Information Systems*. Springer, 1999, pp. 509-516.

- [54] H. Voorhees and T. Poggio, "Computing texture boundaries from images," *Nature*, vol. 333, no. 6171, pp. 364-367, 1988.
- [55] C. Theoharatos, N. Laskaris, G. Economou, and S. Fotopoulos, "A generic scheme for color image retrieval based on the multivariate wald-wolfowitz test," *IEEE Trans. on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 808-819, June 2005.
- [56] J. Friedman and L. Rafsky, "Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests," *Annals of Statistics*, vol. 7, no. 4, pp. 697-717, July, 1979.
- [57] T. Leauhatong, K. Hamamoto, K. Atsuta, S. Kondo, "A New Content-based Image Retrieval Using the Multidimensional Generalization of Wald-Wolfowitz Runs Test," *IEEJ Trans. Electronics, Information and Systems*, vol. 129(2009), no. 1.
- [58] Y. Rubner, J. Puzicha, C. Tomasi, and J.M. Buhmann, "Empirical Evaluation of Dissimilarity Measures for Color and Texture," *Computer Vision and Image Understanding*, vol. 84, pp. 25-43, 2001.
- [59] Y. Rubner and C. Tomasi, "Perceptual Metrics for Image Database Navigation," Kluwer Academic Publishers, 2001.
- [60] P. Brodatz, "Textures: A Photographic Album for Artists and Designers," Dover Publication New York, 1966.
- [61] M. Tuceryan, A. K. Jain, "Texture Analysis," *Handbook of Pattern Recognition & Computer Vision*, Chapter 2.1, World Scientific, Singapore, 1993.
- [62] A. C. Bovik, M. Clark, W. S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE Trans. On Pattern Anal. & Machine Intell.*, vol. 12, pp. 55-73, 1990.
- [63] S. Mallat, "A Theory for Multiresolution Signal Decomposition : The Wavelet Representation," *IEEE Trans. Pattern Analysis & Machine Intell.*, vol. 11, no. 7, pp. 674-693, July 1989.
- [64] A. Kovac, "Wavelet Thresholding for Unequally Spaced Data," Ph. d. thesis, University of Bristol, March 1998.
- [65] I. Daubechies, "Ten Lectures on Wavelets," SIAM, Philadelphia, 1992.
- [66] J. R. Smith and S.-F. Chang, "Enhancing image search engines in visual information environments," in *IEEE 1st Multimedia Signal Processing Workshop*, June 1997.
- [67] A. Wald and J. Wolfowitz, "On a test whether two samples are from the same population," *Ann. Math. Statist.*, vol. 11, pp. 147-162, 1940.

- [68] P. Nakaram, T. Leauhatong, "A New Content-Based Medical Image Retrieval System Based on Wavelet Transform and Multidimensional Wald-Wolfowitz Runs Test," BMEiCON 2012, Dec 5-7, 2012.
- [69] S. Na and D. L. Neuho, "Bennett's Integral for Vector Quantizers," IEEE Trans. Information Theory, vol. 41, no. 4, July 1995.
- [70] S. Na and D. L. Neuhoff, "Bennett's integral for vector quantizers," IEEE Trans. Information Theory, vol. 41, no. 4, July 1995.
- [71] A. Gersho, "Asymptotically optimal block quantization," IEEE Trans. Inform. Theory, vol. IT-25, pp. 373-380, July 1979.
- [72] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., vol. COM-28, pp. 84-95, 1980.

สรุปค่าใช้จ่ายการดำเนินงานโครงการวิจัย

รายการ	จำนวนเงิน
1. งบประมาณ ค่าจ้างชั่วคราวผู้ช่วยนักวิจัย ปริญาตรี 2 คน x 11,680.00 บาท x (5 เดือน 23 วัน) ปริญาโท 1 คน x 15,300.00 บาท x (3 เดือน 29 วัน) รวม	 134,710 บาท 60,690 บาท 195,400 บาท
2. งบดำเนินงาน (ค่าตอบแทน ค่าใช้สอย ค่าวัสดุ ค่าสาธารณูปโภค)	ยอดรวมงบดำเนินงาน
- ค่าตอบแทน ค่าล่วงเวลา : รวม	- -
- ค่าใช้สอย ค่าจ้างเหมาเก็บตัวอย่างภาพที่ใช้ในฐานะข้อมูลภาพทางการแพทย์ รวม	200,000 บาท 200,000 บาท
- ค่าวัสดุ ค่าวัสดุคอมพิวเตอร์ประสิทธิภาพสูง เช่น เมนบอร์ด แผ่นวงจรอิเล็กทรอนิกส์ ค่าไลบรารีสำหรับระบบ PACS รวม	204,600 บาท 200,000 บาท 404,600 บาท
3. งบลงทุน ค่าครุภัณฑ์ รวม	
รวมงบประมาณที่เสนอขอ	800,000 บาท

ข้อมูลประวัติคณะผู้วิจัย

ประวัติส่วนตัว

ชื่อ-สกุล นาย เทอดศักดิ์ ลีวหาทอง

ตำแหน่งปัจจุบัน อาจารย์ระดับ 7

ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
Doctor of Engineering	Science and Technology	มหาวิทยาลัยโตไก	2552
วิศวกรรมศาสตร มหาบัณฑิต วศ.ม.	วิศวกรรมไฟฟ้า	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	2540
วิศวกรรมศาสตรบัณฑิต วศ.บ.	อิเล็กทรอนิกส์	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง	2535

สาขาวิจัยที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา)

- ระบบไมโครคอนโทรลเลอร์ และระบบสมองกลฝังตัว

- การออกแบบวงจรอิเล็กทรอนิกส์

- ระบบแจ้งเตือนการล้มของผู้สูงอายุ

- ระบบดูแลสุขภาพทางไกลของผู้สูงอายุ

- การประมวลผลภาพ (Image Processing) และการประมวลผลสัญญาณดิจิทัล (Digital Signal Processing)

- โปรแกรมจัดเก็บและบริหารภาพทางการแพทย์ (Picture Archives and Communication System : PACS)

ทุนการศึกษาและทุนวิจัยที่เคยได้รับ

ปี พ.ศ.	ทุนการศึกษาและทุนวิจัย	สถาบันที่ให้
2552	โครงการประยุกต์ข้อมูลทางการแพทย์ (Medical Information Application)	ศูนย์การแพทย์สมเด็จพระเทพ รัตนราชสุดาฯ สยามบรมราชกุมารี
2552	โครงการวิจัยเรื่องระบบการแยกส่วนภาพ เซลล์เม็ดเลือดขาว	งบประมาณเงินรายได้ของคณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2553	โครงการวิจัยเรื่องระบบดูแลผู้สูงอายุ ทางไกลผ่านโทรศัพท์เคลื่อนที่	งบประมาณเงินรายได้ของคณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2554	ระบบการค้นหารูปภาพด้วยรายละเอียดของ ภาพ	สำนักงานคณะกรรมการวิจัยแห่งชาติ
2555	ระบบแจ้งเตือนการล้มของผู้สูงอายุผ่าน โทรศัพท์เคลื่อนที่	งบประมาณเงินรายได้ของคณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยี พระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2556	ระบบแจ้งเตือนการล้มของผู้สูงอายุผ่านการ สื่อสารแบบ GSM ของสำนักงาน คณะกรรมการวิจัยแห่งชาติ	หัวหน้าโครงการวิจัย

ผลงานวิจัยที่ทำเสร็จแล้ว : ชื่อผลงานวิจัย ปีที่พิมพ์ การเผยแพร่ และแหล่งทุน (อาจมากกว่า 1 เรื่อง)

[1] T. Leauhatong, K. Hamamoto, K. Atsuta, S. Kondo, “A New Content-Based Image Retrieval Using the Multidimensional Generalization of Wald-Wolfowitz Runs Test,”

IEEJ Trans. Electronics, Information and Systems, vol. 129(2009), no. 1.

[2] S. Chinwaraphat, T. Leauhatong, M. Sangworasil, and C. Pintavirooj, “Breath Motion

Monitoring System,” The 2nd Biomedical Engineering International Conference, August

13-14, 2009.

[3] T. Phanprasit, T. Leauhatong, C. Pintavirooj, and M. Sangworasil, “Image Coding Using

Vector Quantization Based on Wavelet Transform Fuzzy C-Means and Principle Component Analysis,” Proceeding of the 9th International Symposium on Communication and Information Technology 2009, Sept. 28-30, 2009.

- [4] J. Cheewatanon, T. Leunatong, S. Airpaiboon, and M. Sangwarasil, "A New White Blood Cell Segmentation Using Mean Shift Filter and Region Growing Algorithm," *International Journal of Applied Biomedical Engineering*, vol. 4, no. 1, Jan.-June, 2011.
- [5] Sengto, A.; Leauhatong, T., "Human falling detection algorithm using back propagation neural network," *Biomedical Engineering International Conference (BMEiCON)*, 2012, vol., no., pp.1,5, 5-7 Dec. 2012

รองปลัด

