อภินันทนาการ

สำนักหอสมุด

การศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ต ร่วมกับการเข้ารหัสเลขคณิต

A STUDY OF IMAGE COMPRESSION TECHNIQUE USING DISCETE WAVELET TRANSFORM WITH ARITHMETIC CODING

นายภูษิต เมฆนันทไพศิฐ รหัส 53362990

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 12 มิ.ย. 2558
เลขทะเบียน 1719 6994
เลขเรียกหนังสือ

ปร
ก 678ก
2557

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2557

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ต ร่วมกับการเข้ารหัสเลขคณิต

ผู้ดำเนินโครงงาน นายภูษิต เมฆนันทไพศิฐ รหัส 53362990

ที่ปรึกษาโครงงาน ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2557

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล)

.....กรรมการ

(อาจารย์รัฐภูมิ วรานุสาสน์)

ชื่อหัวข้อโครงงาน	การศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ต ร่วมกับการเข้ารหัสเลขคณิต
ผู้ดำเนินโครงงาน	นายภูษิต เมฆนันทไพศิฐ รหัส 53362990
ที่ปรึกษาโครงงาน	ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2557

.....

บทคัดย่อ

โดยทั่วไปการบีบอัดภาพเป็นวิธีที่ทำให้การติดต่อสื่อสารได้เร็วขึ้น และประหยัดพื้นที่ในการจัดเก็บข้อมูล ปริญญานิพนธ์ฉบับนี้ นำเสนอการศึกษาเทคนิคการบีบอัดภาพโดยใช้การแปลงเวฟเล็ตเต็มหน่วย ร่วมกับการเข้ารหัสเลขคณิต โดยใช้ตัวกรองฮาร์ และตัวกรองเดบุชชี4 เพื่อประเมินประสิทธิภาพของวิธีการที่นำเสนอนั้นได้ใช้ภาพทดสอบจำนวน 5 ภาพ คือ ภาพมนุษย์ถ่ายรูป ภาพบาร์บารา ภาพลีน่า ภาพพริกไทย และภาพแมนดริล จากผลการศึกษาพบว่าวิธีการแปลงเวฟเล็ตเต็มหน่วย ชนิดตัวกรองฮาร์มีประสิทธิภาพดีกว่าการแปลงเวฟเล็ตเต็มหน่วย ชนิดตัวกรองเดบุชชี4 ในแง่ของอัตราส่วนการบีบอัด

Project title	A Study Image Compression Technique using Discrete Wavelet Transform with Arithmetic Coding
Name	Mr. Pusit Meknuntapaisit ID. 53362990
Project advisor	Assistant Professor Suchart Yammen , Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2014

.....

Abstract

In general, image compression is a way to make communication faster and space save in storage This project proposes a study of image compression technique using the discrete wavelet transform (DWT) with arithmetic coding. Haar and Daubechies4 filters are used in the DWT method. To evaluate the performance of the proposed method, the experiments are performed with five images: Cameraman, Barbara, Lena, Pepper and Mandril. The results reveal that the Haar DWT method performs much better than the Daubechies4 DWT method in terms of compression ratio.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีจาก ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น ซึ่งเป็นที่ปรึกษาโครงงานและให้ความกรุณาในการตรวจทานปริญญานิพนธ์ผู้ดำเนินโครงงานขอกราบขอบ พระคุณเป็นอย่างสูงและขอระลึกถึงความกรุณาของท่านตลอดไป

ขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล และอาจารย์รัฐภูมิ วรานุสาสน์ ซึ่งเป็นคณะกรรมการในการสอบโครงงานที่ให้คำแนะนำ ชี้แนะแนวทาง และข้อคิดเห็นต่างๆที่เป็นประโยชน์ในโครงงานนี้ ทำให้โครงงานนี้ออกมาสมบูรณ์แบบยิ่งขึ้น

ขอบพระคุณคณาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆตลอดระยะเวลาของการศึกษาเล่าเรียน ซึ่งเป็นความรู้ที่สามารถนำไปใช้ในการทำโครงงานนี้และยังสามารถนำไปใช้ในการประกอบอาชีพในอนาคต

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณของบิดามารดา ผู้มอบความรักความเมตตากรุณา และเป็นกำลังใจให้เสมอมา รวมทั้งเป็นผู้ให้ทุกสิ่งทุกอย่างตั้งแต่วัยเยาว์จวบจนปัจจุบัน คอยเป็นกำลังใจให้ได้รับความสำเร็จอย่างทุกวันนี้ และขอขอบคุณทุกๆคนในครอบครัวของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวไว้ ณ ที่นี้ด้วย

นายภูษิต เมฆนันทไพศิฐ

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก

บทคัดย่อภาษาไทย ..... ข

บทคัดย่อภาษาอังกฤษ ..... ค

กิตติกรรมประกาศ ..... ง

สารบัญ ..... จ

สารบัญตาราง ..... ฉ

สารบัญรูปภาพ ..... ญ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของงาน 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ผลที่คาดว่าจะได้รับ 3

1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎี 4

2.1 ภาพเชิงดิจิตอล 4

2.2 ประเภทการบีบอัดภาพ 5

2.2.1 การบีบอัดข้อมูลภาพแบบไม่สูญเสีย 5

2.2.2 การบีบอัดข้อมูลภาพแบบสูญเสีย 5

2.3 การบีบอัดข้อมูลภาพ 6

2.4 การวัดประสิทธิภาพการบีบอัดสัญญาณภาพดิจิตอล 6

สารบัญ(ต่อ)

หน้า

2.5 การบันทึกไฟล์ข้อมูลฐานสอง 9
2.5.1 นิยามและความหมายของข้อมูลฐานสอง 9
2.5.2 การบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็บ 9
2.6 การแปลงเวฟเล็ต 13
2.6.1 การแปลงเวฟเล็ตเต็มหน่วยแบบ 1 มิติ 13
2.6.2 การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 17
2.6.3 การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ระดับ 20
2.7 การเลือกใช้ตระกูลเวฟเล็ตในการศึกษาการบีบอัดภาพ 22
2.7.1 ตัวกรองเวฟเล็ตฮาร์ 22
2.7.2 ตัวกรองเวฟเล็ตเดบุชชี4 23
2.8 การเข้ารหัสเลขคณิต 24
2.8.1 การเข้ารหัสอิมพลีเมนต์เตชั่น 25
2.8.2 การถอดรหัสอิมพลีเมนต์เตชั่น 26

บทที่ 3 วิธีดำเนินการ 27
3.1 ศึกษาและค้นคว้าทฤษฎีพื้นฐานที่เกี่ยวข้อง 27
3.2 ศึกษาการแปลงเวฟเล็ต 27
3.3 ศึกษาการเข้ารหัสเลขคณิต 28
3.4 ออกแบบโปรแกรมการบีบอัดภาพ 29
3.4.1 การบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์รวมทั้งการเข้ารหัสเลขคณิต 29
3.4.2 การคายการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์รวมทั้งการเข้ารหัสเลขคณิต 31

สารบัญ(ต่อ)

หน้า

3.4.3 การบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองเดบุชชี4 รวมทั้งการเข้ารหัสเลขคณิต..... 32

3.4.4 การคายการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองเดบุชชี4 รวมทั้งการเข้ารหัสเลขคณิต..... 33

3.5 ทดสอบอัลกอริทึม..... 33

3.6 วิเคราะห์ผลการทดลอง สรุปผลการทดลอง ปัญหา และข้อเสนอแนะ..... 33

3.7 จัดทำรูปเล่มปริญญานิพนธ์..... 33

บทที่ 4 ผลการทดลองและการวิเคราะห์ผลการทดลอง..... 34

4.1 ทดสอบอัลกอริทึมการเข้ารหัสเลขคณิต..... 34

4.2 ทดสอบอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต..... 51

4.3 ภาพดิจิตอลก่อนการทำการบีบอัดภาพ และหลังการบีบอัดภาพ..... 58

4.4 ผลการทดลองจากการบีบอัดภาพดิจิตอล..... 59

4.4.1 การทดลองการบีบอัดภาพ แบบไม่บันทึกตาราง..... 59

4.4.2 การทดลองการบีบอัดภาพ แบบบันทึกตาราง..... 61

4.5 สรุปผลการทดลอง..... 62

4.5.1 การเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบไม่บันทึกตาราง..... 63

4.5.2 การเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบบันทึกตาราง..... 63

สารบัญ(ต่อ)

หน้า

บทที่ 5 สรุปผลและข้อเสนอแนะ..... 64

5.1 สรุปผลการดำเนินงาน..... 64

5.2 ปัญหาและแนวทางแก้ไข..... 65

5.3 ข้อเสนอแนะและแนวทางการพัฒนา..... 66

เอกสารอ้างอิง..... 67

ภาคผนวก ก รายละเอียดฟังก์ชัน new_arithme..... 68

ภาคผนวก ข โปรแกรมการบีบอัดภาพดิจิตอล

Wavelet Transform (Haar)+Arithmetic Coding..... 71

ภาคผนวก ค โปรแกรมการบีบอัดภาพดิจิตอล

Wavelet Transform (Db4)+Arithmetic Coding..... 109

ภาคผนวก ง รายละเอียดโปรแกรม Arithmetic Coding..... 147

ประวัติผู้ดำเนินโครงงาน..... 164

สารบัญตาราง

ตารางที่ หน้า

2.1 ตารางความแม่นยำบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็บ..... 11

2.2 ตารางความแม่นยำในการเขียนแมชชีนฟอร์แมท..... 13

4.1 ตารางการแมปข้อมูลจากเวิ์กเตอร์แนวนอนไปเป็นลำดับในการเข้ารหัสเลขคณิต..... 34

4.2 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิด ฮาร์ ร่วมกับการเข้ารหัสเลขคณิต แบบไม่บันทึกตาราง..... 60

4.3 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิด เดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต แบบไม่บันทึกตาราง..... 60

4.4 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิด ฮาร์ ร่วมกับการเข้ารหัสเลขคณิต แบบบันทึกตาราง..... 61

4.5 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิด เดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต แบบบันทึกตาราง..... 62

4.6 ตารางการเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบไม่บันทึกตาราง..... 63

4.7 ตารางการเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบบันทึกตาราง..... 63

สารบัญรูป

รูปที่		หน้า
2.1	ภาพเชิงดิจิตอล	4
2.2	แบบการบีบอัดภาพโดยทั่วไป	6
2.3	แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 1 มิติ	14
2.4	แผนภาพการลดข้อมูลลงครึ่งหนึ่ง	14
2.5	แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 1 มิติ	15
2.6	แผนภาพการเพิ่มข้อมูลขึ้นสองเท่า	15
2.7	แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 3 แบนด์ย่อย	16
2.8	แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 3 แบนด์ย่อย	17
2.9	แผนภาพการแยกองค์ประกอบด้วยเวฟเล็ต	18
2.10	แผนภาพการสร้างกลับองค์ประกอบด้วยเวฟเล็ต	18
2.11	แบนด์ย่อยที่ผ่านการแปลงเวฟเล็ตหลายครั้ง	19
2.12	แผนภาพการแยกองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง	20
2.13	แผนภาพการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง	21
2.14	เวฟเล็ตฮาร์ และพัลส์ฟังก์ชั่น	23
3.1	แผนภาพการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต	29
3.2	แผนภาพการคายการบีบอัดภาพดิจิตอลด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการถอดรหัสเลขคณิต	31
4.1	มนุษย์ถ่ายรูป	58
4.2	บาร์บารา	58
4.3	ลีน่า	58
4.4	พริกไทย	59
4.5	แมนดริล	59
4.6	กู้คืนมนุษย์ถ่ายรูป	58
4.7	กู้คืนบาร์บารา	58
4.8	กู้คืนลีน่า	58
4.9	กู้คืนพริกไทย	59
4.10	กู้คืนแมนดริล	59

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากในปัจจุบันข้อมูลภาพดิจิตอลได้เข้ามามีบทบาทสำคัญในชีวิตประจำวันของทุกคน ทั้งในด้านการถ่ายภาพ การสื่อสาร การศึกษา การแพทย์ การทหาร เป็นต้น ซึ่งรูปภาพดิจิตอลส่วนใหญ่จะมีขนาดใหญ่ ทำให้ต้องใช้เนื้อที่จำนวนมากในการจัดเก็บข้อมูล การบีบอัดรูปภาพจึงเป็นวิธีการหนึ่งที่จะช่วยให้การติดต่อสื่อสารให้เร็วขึ้น เนื่องจากการบีบอัดข้อมูลทำให้ไฟล์มีขนาดเล็กลงและคุณภาพของข้อมูลไม่สูญเสียดังนั้นโครงงานนี้ให้ความสนใจการศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิต

1.2 วัตถุประสงค์ของโครงงาน

เพื่อศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิต

1.3 ขอบข่ายของงาน

1) โครงงานนี้เป็นวิธีการบีบอัดแบบไม่สูญเสีย
2) ภาพที่นำมาทดสอบการบีบอัดภาพทั้ง 2 วิธีนี้ใช้ภาพทดสอบจำนวน 5 ภาพ คือ ภาพระดับเทามนุษย์ถ่ายรูป (Cameraman) ขนาด 256X256 พิกเซล ภาพระดับเทาบาร์บารา (Barbara) ขนาด 512X512 พิกเซล ภาพสีลีน่า (Lena) ขนาด 512X512 พิกเซล ภาพสีพริกไทย (Peppers) ขนาด 512x512 พิกเซล และภาพสีแมนดริล (Mandril) ขนาด 512X512 พิกเซล ทำการทดสอบเปรียบเทียบระหว่างการใช้การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรอง haarและ db4 รวมทั้งการเข้ารหัสเลขคณิต มาทำการเปรียบเทียบอัตราส่วนการบีบอัด ค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE)และอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด(Peak Signal to Noise Ratio: PSNR)กับอัตราส่วนสัญญาณต่อสัญญาณรบกวน(Signal to Noise Ratio: SNR)
3) ใช้โปรแกรม Matlab2014a

1.4 ขั้นตอนการดำเนินงาน

รายละเอียด	พ.ศ.2557								
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.
1. ศึกษาและค้นคว้าทฤษฎีพื้นฐาน									
2.ศึกษาเทคนิคการแปลงเวฟเล็ตทรานฟอร์ม									
3.ศึกษาเทคนิคการเข้ารหัสเลขคณิต									
4.ศึกษาการเลือกใช้ตระกูลเวฟเล็ต									
5.ออกแบบโปรแกรม การบีบอัดภาพโดยใช้ MATLAB									
6.ทดสอบอัลกอริทึม									
7.วิเคราะห์ผลการทดลอง สรุปผลการทดลองปัญหาและข้อเสนอแนะ									
8.จัดทำรูปเล่มปริญญานิพนธ์									

1.5 ผลที่คาดว่าจะได้รับ

1) นิสิตได้พัฒนาทักษะการใช้งานโปรแกรม Matlab
2) ทำให้เข้าใจถึงวิธีการประยุกต์ใช้การเข้ารหัสเลขคณิตเพื่อนำไปใช้กับข้อมูลรูปภาพ
3) ทำให้เข้าใจถึงวิธีการประยุกต์ใช้การแปลงเวฟเล็ตเต็มหน่วยร่วมกับการเข้ารหัสเลขคณิต เพื่อนำไปใช้กับข้อมูลรูปภาพ
4) ช่วยประหยัดพื้นที่ในการจัดเก็บข้อมูลภาพ

1.6 งบประมาณ

1) ค่าถ่ายเอกสาร	500	บาท
2) ค่าจัดทำรูปเล่ม	500	บาท
รวมเป็นเงิน(หนึ่งพันบาทถ้วน)	1,000	บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในบทนี้ผู้วิจัยได้ศึกษาหลักการและทฤษฎีที่เกี่ยวข้องซึ่งเริ่มต้นศึกษาจากภาพเชิงดิจิตอล ประเภทการบีบอัดข้อมูลภาพ, การบีบอัดข้อมูลภาพ, การวัดประสิทธิภาพการบีบอัดสัญญาณภาพดิจิตอลการบันทึกไฟล์ข้อมูลฐานสอง, การแปลงเวฟเล็ต, การแปลงเวฟเล็ตเต็มหน่วยแบบ 1 มิติ, การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ, การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ครั้ง, การเลือกใช้ตระกูลเวฟเล็ตที่ใช้ในการศึกษาการบีบอัดภาพ และการเข้ารหัสเลขคณิตกับการถอดรหัสเลขคณิต โดยมีรายละเอียดดังนี้

2.1 ภาพเชิงดิจิตอล (Digital Image)

ภาพดิจิตอล[1] เป็นการแสดงผลภาพลักษณะสองมิติในหน่วยที่เรียกว่า"พิกเซล"ภาพดิจิตอลสามารถนิยามเป็นฟังก์ชันสองมิติ f(x, y) โดยที่ค่าของ x และ y เป็นพิกัดในระนาบ(spatial /plane coordinate) และค่าของฟังก์ชัน f พิกัด (x, y) ใดๆ ภายในภาพคือ ค่าความเข้มแสงหรือค่าระดับเทาของภาพ (Intensity /gray level) ที่ตำแหน่งนั้นๆ เมื่อ x y และ f มีค่าแน่นอนเป็นจำนวนเต็ม จึงเรียกรูปภาพนี้ว่าเป็นภาพดิจิตอล

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

รูปที่ 2.1 ภาพเชิงดิจิตอล[2]

ในปัจจุบันมีมาตรฐานการจัดเก็บข้อมูลภาพที่แตกต่างกันอย่างไรก็ตามเมื่อใช้คำสั่ง imread ในการอ่านไฟล์ภาพแล้วจะบรรจุข้อมูลภาพที่ได้ลงในโครงสร้างการเก็บข้อมูลที่เหมาะสมซึ่งมีให้เลือก 4 แบบเนื่องจากโครงงานนี้ผู้วิจัยได้ใช้ภาพเชิงดิจิตอลที่ประกอบด้วยภาพระดับเทา และภาพสีในการทดลองการบีบอัดข้อมูลภาพ ซึ่งมีโครงสร้างการเก็บข้อมูลภาพ 2 แบบคือ ภาพความเข้มและภาพอาร์จีบี (IMAGE RGB) โดยมีรายละเอียด ดังนี้

ภาพความเข้มเป็นรูปแบบการเก็บข้อมูลภาพที่ออกแบบมาเพื่อใช้กับภาพระดับเทาโดยตรง โครงสร้างที่ใช้เป็นเมตริกซ์ขนาด mxn โดยสมาชิกแต่ละค่าในเมตริกซ์ภาพบอกถึงระดับความสว่าง

หรือความมืดของพิกเซล ทั้งนี้ในการทดลองการบีบอัดข้อมูลภาพด้วย MATLAB ได้ใช้ตัวแปร uint8 ซึ่งเก็บค่าตัวเลขจำนวนเต็มที่มีค่าตั้งแต่ 0 ถึง 255 โดยหากมีค่าเป็น 0 คือสีดำสนิท และใช้ตัวเลข 255 แทนสีขาวภาพอาร์จีบี เป็นโครงสร้างการเก็บภาพที่ได้รับการออกแบบมาเพื่อใช้กับภาพสี โครงสร้างที่ใช้เป็นเมตริกซ์ขนาด mxnx3 คือการนำเอาเมตริกซ์ขนาด mxn จำนวนสามชุดมาต่อกันเป็นเมตริกซ์ที่มีมิติเพิ่มเป็นสามมิติ เมตริกซ์ mxn แต่ละชุดบรรจุค่าความเข้มของสีแดง (R) สีเขียว (G) และสีน้ำเงิน (B) ตามลำดับ ฉะนั้นสีของพิกเซลแต่ละจุดจะถูกกำหนดจากปริมาณสีของแต่ละสีที่นำมาผสมกัน ทั้งนี้ในการทดลองการบีบอัดข้อมูลภาพด้วยแม็ทแล็ป (MATLAB) ได้ใช้ตัวแปร uint8 นี้พิกเซลแต่ละจุดจะมีการใช้จำนวนบิตรวมต่อจุดเท่ากับ 24 บิต ซึ่งเก็บค่าตัวเลขจำนวนเต็มที่มีค่าตั้งแต่ 0 ถึง 16,777,216 โดยหากมีค่าเป็น 0 คือสีดำสนิท ส่วนค่าตรงกลางก็เป็นสีต่างๆกันไป และใช้ตัวเลข 16,777,216 แทนสีขาว

2.2 ประเภทการบีบอัดข้อมูลภาพ (Type Image Compression)

การบีบอัดข้อมูลภาพเป็นกระบวนการทำให้ขนาดข้อมูลภาพมีขนาดที่ลดลง ซึ่งไฟล์ภาพที่เราได้มาโดยทั่วไปนั้นมักจะมีขนาดใหญ่ และจำเป็นต้องใช้พื้นที่หน่วยความจำเป็นจำนวนมากเพื่อที่จะจัดเก็บข้อมูลเหล่านี้ ดังนั้นกระบวนการในการบีบอัดข้อมูลภาพจะช่วยลดพื้นที่หน่วยความจำในการจัดเก็บข้อมูลภาพ จำเป็นต้องใช้เทคนิคของการเข้ารหัสและถอดรหัสมาช่วยโดยที่การบีบอัดข้อมูลภาพสามารถแบ่งออกเป็น 2 ประเภท ดังนี้

2.2.1 การบีบอัดข้อมูลภาพแบบไม่สูญเสีย (Lossless Compression)

ข้อมูลภาพที่ถูกแปลงกลับจะเหมือนกับภาพต้นฉบับทุกประการ โดยจะถูกนำมาใช้ในกรณีที่ข้อมูลภาพที่มีอยู่นั้นยากต่อการได้มาและเป็นข้อมูลที่สำคัญ เช่น ข้อมูลภาพถ่ายทางการแพทย์[2]

2.2.2 การบีบอัดข้อมูลภาพแบบสูญเสีย (Lossy Compression)

เป็นการบีบอัดข้อมูลที่ยอมให้ข้อมูลบางส่วนสูญเสียไปได้ โดยคงไว้ซึ่งข้อมูลหลักที่จำเป็น เหมาะสำหรับข้อมูลภาพที่ไม่ได้มีความสำคัญมากนักและผู้ใช้สามารถสร้างกลับข้อมูลภาพได้อย่างรวดเร็วแต่ข้อมูลที่สร้างขึ้นมาใหม่จะมีการสูญเสียข้อมูลไปบางส่วนและอัตราการบีบอัดที่ได้จะสูงกว่าการบีบอัดแบบไม่สูญเสียซึ่งในปัจจุบันเป็นที่นิยมมาก เช่น การบีบอัดแบบเจเพ็กซ์ (JPEG)[2]

2.3 การบีบอัดข้อมูลภาพ (Image Compression)

ในการบีบอัดข้อมูลภาพ สามารถเขียนเป็นแผนภาพบล็อกทั่วไปได้ ดังแสดงในรูป 2.2 ประกอบด้วยการแปลงข้อมูล, การเข้ารหัส, ไฟล์ข้อมูลฐานสอง, การถอดรหัส และการแปลงผกผันกลับ ในการบีบอัดข้อมูลภาพนั้นจะประกอบด้วย 2 ส่วนหลัก คือ การบีบอัดข้อมูลภาพ และการคายการบีบอัดข้อมูลภาพ โดยมีรายละเอียดดังนี้

กระบวนการบีบอัดภาพนั้นเริ่มแรกจะนำภาพต้นฉบับมาแปลงข้อมูลมาทำการเข้ารหัสข้อมูลโดยผลลัพธ์ที่ได้จะเป็นข้อมูลฐานสอง ซึ่งจะถูกนำมาเก็บเป็นไฟล์ภาพที่ถูกบีบอัดที่ถูกลดขนาดลงแล้ว ต่อมาจะมีการคำนวณหาอัตราส่วนการบีบอัดข้อมูลระหว่างขนาดไฟล์ของภาพต้นฉบับกับขนาดไฟล์ของภาพที่ถูกบีบอัด

กระบวนการคายการบีบอัดข้อมูลภาพจะนำเอาไฟล์ข้อมูลฐานสองที่ถูกบันทึกมาทำการถอดรหัสแล้วจึงนำมาแปลงผกผันกลับก็จะได้ภาพที่ผ่านการลดขนาดและแปลงกลับมาได้ภาพกู้คืนที่มีลักษณะเหมือนภาพต้นฉบับทุกประการ นอกจากนี้แล้วยังมีการทดสอบประสิทธิภาพการบีบอัดสัญญาณภาพดิจิตอล ด้วยการคำนวณหาค่าความผิดพลาดกำลังสองเฉลี่ย (MSE) และการหาค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) กับการคำนวณหาค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) โดยจะทำการคำนวณจากภาพต้นฉบับกับภาพกู้คืน[2]

รูปที่ 2.2 รูปแบบการบีบอัดภาพโดยทั่วไป[5]

2.4 การวัดประสิทธิภาพการบีบอัดสัญญาณภาพดิจิตอล

การบีบอัดและการเข้ารหัสข้อมูลภาพนั้นเป็นกระบวนการลดขนาดของเนื้อที่ในการจัดเก็บข้อมูลภาพ ซึ่งมีการแสดงค่าเป็นอัตราส่วนขนาดของข้อมูลต้นฉบับและขนาดของข้อมูลที่ถูกบีบอัด เช่น 2:1 และ4:1 เป็นต้น โดยจะมีการวัดประสิทธิภาพการบีบอัดสัญญาณภาพดิจิตอลด้วยการหาค่าอัตราส่วนการบีบอัดข้อมูล

การวัดค่าอัตราส่วนของการบีบอัดข้อมูล (Compression Ratio: CR) คือ อัตราส่วนของขนาดของข้อมูลต้นฉบับ (n_1) กับของขนาดข้อมูลที่ถูกบีบอัดแล้ว (n_2) ดังสมการที่ 2.1[7]

$$CR = \frac{n_1}{n_2} \tag{2.1}$$

สำหรับการหาประสิทธิภาพการคายการบีบอัดและการถอดรหัสข้อมูลภาพด้วยการทำภาพกู้คืนให้เหมือนกับภาพต้นฉบับมากที่สุด จะใช้ดัชนีชี้วัดประสิทธิภาพ 3 ตัว คือ ดัชนีวัดค่าผิดพลาดเฉลี่ยกำลังสอง (MSE) และดัชนีค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (PSNR) กับดัชนีวัดค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNR)

ดัชนีวัดค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) คือผลต่างกำลังสองของภาพกู้คืนกับภาพต้นฉบับ ดังสมการที่ 2.2[8]

$$MSE = \frac{1}{mn}\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}[f(x,y) - g(x,y)]^2 \tag{2.2}$$

โดยที่ m คือ จำนวนพิกเซลตามความกว้างของภาพ n คือ จำนวนพิกเซลตามความสูงของภาพ $f(x,y)$ คือ ค่าความเข้มที่ตำแหน่ง (x,y) ของภาพต้นฉบับ $g(x,y)$ คือ ค่าความเข้มที่ตำแหน่ง (x,y) ของภาพกู้คืน

ดัชนีวัดค่าความผิดพลาดกำลังสองเฉลี่ย ถ้ามีค่าน้อยแสดงว่าภาพกู้คืนจากการบีบอัดสัญญาณมีความผิดเพี้ยนไปจากภาพต้นฉบับน้อย หรือถ้าดัชนีวัดค่าความผิดพลาดกำลังสองเฉลี่ยมีค่ามากแสดงว่าภาพกู้คืนที่ได้จากการบีบอัดสัญญาณมีความผิดเพี้ยนไปจากภาพต้นฉบับมาก และถ้าดัชนีวัดค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าเป็นศูนย์แสดงว่าภาพกู้คืนเหมือนกับภาพต้นฉบับทุกประการ

ดัชนีวัดค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) คืออัตราส่วนของกำลังสัญญาณภาพต้นฉบับสูงสุดต่อกำลังสัญญาณรบกวนของภาพกู้คืน โดยที่ดัชนีวัดค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุดเป็นตัววัดคุณภาพในการคายการบีบอัดข้อมูลแล้วคุณภาพของภาพที่กู้คืนมีการสูญเสียมากน้อยเพียงใด ดังสมการที่ 2.3[9]

$$PSNR = 10.log_{10}\left(\frac{MAX_I{}^2}{MSE}\right) \qquad [decibel] \tag{2.3}$$

โดยที่ MAX_I คือ ค่าความเข้มของพิกเซลที่มากที่สุดที่เป็นไปได้ของภาพต้นฉบับ และดัชนีวัดค่าผิดพลาดเฉลี่ยกำลังสอง (MSE) คือผลต่างกำลังสองของภาพกู้คืนกับภาพต้นฉบับ

ค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดถ้ามีค่าน้อยแสดงว่าภาพกู้คืนมีความผิดเพี้ยนไปจากภาพต้นฉบับมาก และถ้าอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดมีค่ามาก แสดงว่าภาพกู้คืนมีความผิดเพี้ยนไปจากภาพต้นฉบับน้อย นอกจากนี้ถ้าค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดถ้ามีค่าเป็นอนันต์แสดงว่าภาพกู้คืนเหมือนกับภาพต้นฉบับทุกประการ

ดัชนีวัดค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน(Signal to Noise Ratio: SNR) คือ อัตราส่วนของกำลังสัญญาณภาพต้นฉบับต่อกำลังสัญญาณรบกวนของภาพกู้คืน โดยดัชนีวัดค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนจะเป็นตัววัดคุณภาพในการคายการบีบอัดข้อมูลแล้วคุณภาพของภาพที่กู้คืนมีการสูญเสียมากน้อยเพียงใด ดังสมการที่ 2.4[10]

$$SNR = 10\,log\left\{\frac{\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}f(x,y)^2}{\sum_{x=0}^{m-1}\sum_{y=0}^{n-1}[f(x,y)-g(x,y)]}\right\}\ [decibel] \tag{2.4}$$

โดยที่ $\mathbf{m}$ คือ จำนวนพิกเซลตามความกว้างของภาพ, $\mathbf{n}$ คือ จำนวนพิกเซลตามความสูงของภาพ $\mathbf{f(x,y)}$ คือ ค่าความเข้มที่ตำแหน่ง $\mathbf{(x,y)}$ ของภาพต้นฉบับ และ $\mathbf{g(x,y)}$คือ ค่าความเข้มที่ตำแหน่ง $\mathbf{(x,y)}$ ของภาพกู้คืน

ดัชนีวัดค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวน ถ้ามีค่าน้อยแสดงว่าภาพกู้คืนมีความผิดเพี้ยนไปจากภาพต้นฉบับมาก และถ้าดัชนีวัดอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดมีค่ามาก แสดงว่าภาพกู้คืนมีความผิดเพี้ยนไปจากภาพต้นฉบับน้อย และถ้าดัชนีวัดอัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงสุดมีค่าเป็นอนันต์ แสดงว่าภาพกู้คืนเหมือนกับภาพต้นฉบับทุกประการ

2.5 การบันทึกไฟล์ข้อมูลฐานสอง

ในการเขียนโปรแกรมนั้นไฟล์ข้อมูลจะถูกแบ่งออกตามลักษณะการจัดเก็บได้ 2 ประเภท คือ เท็กซ์ไฟล์ (Text file) และไฟล์ข้อมูลฐานสอง (Binary File) เนื่องจากในโครงงานนี้ผู้วิจัยจะต้องมีการบันทึกไฟล์ข้อมูลภาพที่ถูกบีบอัดจึงทำให้ต้องเลือกบันทึกแบบไฟล์ข้อมูลฐานสอง เนื่องจากเมื่อบันทึกไฟล์ชนิดนี้แล้วจะไม่มีการเปลี่ยนรหัสคำสั่ง ซึ่งในโครงงานนี้ผู้วิจัยได้ศึกษานิยามและความหมายของข้อมูลฐานสอง กับการบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็ป (MATLAB) โดยมีรายละเอียดต่างๆดังนี้

2.5.1 นิยามและความหมายของข้อมูลฐานสอง (Binary)

ไฟล์ (File) คือ ที่เก็บข้อมูลต่างๆของคอมพิวเตอร์ และในภาษาซี (Language C) นั้นไฟล์หมายถึงอุปกรณ์ต่างที่ต่อกับเครื่องคอมพิวเตอร์ด้วย เช่น คีย์บอร์ด จอภาพ การติดต่อกับไฟล์จะต้องผ่านลอจิกคอลอินเตอร์เฟส (Logical Interfaces) ที่เรียกว่าสตรีม (Stream) สตรีมจะช่วยให้ผู้ใช้เขียนโปรแกรมติดต่อกับอุปกรณ์ต่างๆได้ ซึ่งสตรีมที่ใช้ติดต่อกับไฟล์ ไฟล์จะมีอยู่ 2 ประเภทคือ เท็กซ์ไฟล์(Text file) เป็นไฟล์ของตัวอักษร เพราะมีโครงสร้างในการเก็บข้อมูลจะเป็นตัวอักษรไฟล์นั้นจึงไม่สามารถที่จะเก็บข้อมูลที่ค่าตัวเลขจำนวนเต็ม จุดทศนิยม หรือในลักษณะที่เป็นโครงสร้างซึ่งการเก็บข้อมูลจะถูกแปลงเป็นเลขฐานสองตามรหัสแอสกี (ASCII) ไฟล์ข้อมูลฐานสอง (Binary File) เป็นไฟล์ที่เก็บข้อมูลที่อยู่ในรูปแบบของค่าตรงๆซึ่งข้อมูลที่ไฟล์ประเภทนี้จัดเก็บ จะสามารถเป็นได้ทั้งตัวเลขจำนวนเต็ม ตัวเลขทศนิยมตัวอักษร อาร์เรย์ และข้อมูลแบบโครงสร้าง โดยการจัดเก็บนั้นจะเก็บลงไปตรงๆเลยทำให้รหัสคำสั่งไม่มีการเปลี่ยนแปลง

2.5.2 การบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็ป

ในการบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็ปนั้นเริ่มโดยคำสั่ง fwrite และการเขียนข้อมูลฐานสองมีหน้าที่ในการบันทึกไฟล์เป็นลักษณะข้อมูลฐานสอง

รูปแบบคำสั่ง

fwrite (fileID, A)

fwrite (fileID, A, precision)

fwrite (fileID, A, precision, skip)

fwrite (fileID, A, precision, skip, machine format)

count = fwrite (...)

รายละเอียด

fwrite (fileID, A) ทำหน้าที่เขียนองค์ประกอบอาร์เรย์ (array) A เป็นบิตแบบไม่มีเครื่องหมาย (uint8) จำนวน 8 บิต บันทึกไปยังในไฟล์ข้อมูลฐานสองตามลำดับคอลัมน์

fwrite (fileID, A, precision) การเขียนค่าอาร์เรย์ A ไปตามรูปแบบ และขนาดที่ได้กำหนดไว้จากความแม่นยำ

fwrite (fileID, A, precision, skip) ข้าม (skip) จะสั่งให้บันทึกข้อมูลข้ามไปตามจำนวนไบต์ (Bytes) หรือจำนวนบิต ที่กำหนดก่อนที่จะเขียนแต่ละค่าข้ามไป ถ้าค่าความแม่นยำนั้นเป็นบิตแบบมีเครื่องหมาย (bit n) จำนวน n บิต หรือ บิตแบบไม่มีเครื่องหมาย (ubit) จำนวน n บิต เพื่อกำหนดให้กระโดดข้ามไปเท่ากับจำนวนบิตเหล่านั้น

fwrite (fileID, A, precision, skip, machine format) เป็นการเขียนข้อมูลจำเพาะตามรูปแบบแมชชีนฟอร์แมท (machine format) โดยที่สามารถเลือกข้ามพารามิเตอร์ (parameter) ได้

count = fwrite(...) เป็นการสั่งนับองค์ประกอบของอาร์เรย์ A ที่ได้ทำตามคำสั่ง fwrite ได้เสร็จสมบูรณ์ในการเขียนไฟล์

ตัวแปรนำเข้า

fileID คือตัวกำหนดไฟล์ มีหน้าที่จำเพาะให้เป็นอย่างใดอย่างหนึ่งใน
- A เป็นตัวกำหนดไฟล์ที่ได้มาจากคำสั่ง fopen
- 1 เป็นเอาท์พุตมาตรฐาน (ที่แสดงออกทางหน้าจอ)
- 2 เป็นตัวรายงานมาตรฐานความผิดพลาด (Error)

A คือ ตัวเลขหรืออาร์เรย์ของตัวอักษร

ความแม่นยำ (precision) เป็นตัวแปรสตริงที่อยู่เครื่องหมาย"ข้อความสตริง" ที่จะทำการควบคุมรูปแบบ และขนาดของเอาท์พุต ซึ่งแสดงในตารางที่ 2.1 ดังต่อไปนี้คือค่าความเป็นไปได้ของค่าความแม่นยำ

ตารางที่ 2.1 ตารางความแม่นยำบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็บ

ชนิดของค่า	ความแม่นยำ	บิต (ไบต์)
integers,unsigned	Uint	32 (4)
	uint8	8 (1)
	uint16	16 (2)
	uint32	32 (4)
	uint64	64 (8)
	Uchar	8 (1)
	unsigned char	8 (1)
	Ushot	16 (2)
	Ulong	32 (4)
	ubit *n*	1 n 64
integers, signed	Int	32 (4)
	int8	8 (1)
integers, signed	int16	16 (2)
	Int32	32 (4)
	Int64	64 (8)
	integer*1	8 (1)
	integer*2	16 (2)
	integer*4	32 (4)
	integer*8	64 (8)

ตารางที่ 2.1 ตารางความแม่นยำบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็บ (ต่อ)

ชนิดของค่า	ความแม่นยำ	บิต (ไบต์)
integers, signed	Schar	8 (1)
	signed char	8 (1)
	Short	16 (2)
	Long	32 (4)
	bits n	1 n 64
floating-point numbers	Single	32 (4)
	Double	64 (8)
	Float	32 (4)
	float32	32 (4)
	float64	64 (8)
	real*4	32 (4)
	real*8	64 (8)
Characters	char*1	8 (1)
	Char	ขึ้นอยู่กับรูปแบบการเข้ารหัสที่เกี่ยวข้องกับแฟ้ม ตั้งค่าการเข้ารหัสด้วย fopen

ถ้าเราทำการเจาะจงค่าความแม่นยำหนึ่งๆของบิตแบบมีเครื่องหมาย (bit n) และค่าหนึ่งๆ มันเกินกำหนด โดยคำสั่ง fwrite จะเป็นการตั้งให้บิตที่ค่าเกินกำหนดเหล่านั้นทั้งหมดนั้นเป็นค่าดีฟอล์:uint8

ข้าม (skip) จะสั่งให้บันทึกข้อมูลข้ามไปตามจำนวนไบต์ (Bytes) หรือจำนวนบิต ที่กำหนดก่อนที่จะเขียนแต่ละค่าข้ามไป ถ้าค่าความแม่นยำนั้นเป็นบิตแบบมีเครื่องหมาย (bit n) จำนวน n บิต หรือ บิตแบบไม่มีเครื่องหมาย (ubit) จำนวน n บิต เพื่อกำหนดให้กระโดดข้ามไปเท่ากับจำนวนบิตเหล่านั้น ใช้พารามิเตอร์นี้ในการแทรกข้อมูลไปยังฟิลด์ของข้อมูลที่ไม่เชื่อมต่อกันในระยะที่ข้อมูลคงที่ค่าดีฟอล์: 0

แมชชีนฟอร์แมท (Machine format) เป็นตัวแปรสตริงที่ประกอบไปด้วยลำดับจำเพาะในการเขียนไบต์ไปยังไฟล์ สำหรับบิตแบบมีเครื่องหมาย (n bit) และบิตแบบไม่มีเครื่องหมาย (ubit)

ความแม่นยำในการกำหนดค่าจำเพาะของลำดับในการเขียนบิตภายใน 1 ไบต์ ซึ่งแสดงในตารางที่ 2.2 ดังต่อไปนี้คือค่า แมชชีนฟอร์แมท

ตารางที่ 2.2 ตารางความแม่นยำในการเขียนแมชชีนฟอร์แมท

'n' หรือ 'native'	ระบบของลำดับการเขียนไบต์ของคุณ (ค่าดีฟอล์)
'b' หรือ 'ieee-be'	ลำดับบิ๊กเอนเดี้ยนออเดอร์ริ่ง (Big-endian ordering)
'l' หรือ 'ieee-le'	ลำดับลิตเติ้ลเอนเดี้ยนออเดอร์ริ่ง (Little-endian ordering)
's' หรือ 'ieee-be.l64'	ลำดับบิ๊กเอนเดี้ยนออเดอร์ริ่ง (Big-endian ordering), 64-บิต สำหรับข้อมูลที่ยาว
'a' หรือ 'ieee-le.l64'	ลำดับลิตเติ้ลเอนเดี้ยนออเดอร์ริ่ง (Little-endian ordering), 64-บิต สำหรับข้อมูลที่ยาว

โดยค่าค่าดีฟอล์ ได้กำหนดให้สนับสนุนทั้งหมดเป็นรูปแบบลิตเติ้ลเอนเดี้ยนออเดอร์ริ่ง (little-endian ordering) สำหรับการสร้างไฟล์ใหม่ สำหรับไฟล์ฐานสองต่างๆที่สร้างไว้แล้วสามารถใช้บิ๊กเอนเดี้ยน (big-endian) หรือลิตเติ้ลเอนเดี้ยนออเดอร์ริ่งอื่นๆได้

2.6 การแปลงเวฟเล็ต

ในหัวข้อนี้ผู้วิจัยจะศึกษาถึงทฤษฎีพื้นฐานของการแปลงเวฟเล็ตเต็มหน่วย (Discrete Wavelet Transform: DWT) แบบ 1 มิติ ด้วยหลักการฟิลเตอร์แบงค์ (Filter Bank) การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ด้วยหลักการแยกองค์ประกอบเป็นแบนด์ย่อย (Subband Decomposition) และการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ครั้ง ด้วยหลักการแยกองค์ประกอบเป็นหลายแบนด์ย่อย โดยมีรายละเอียดดังนี้

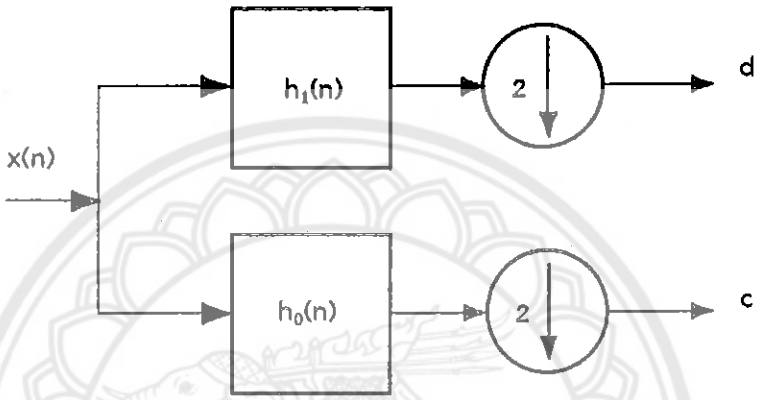
2.6.1 การแปลงเวฟเล็ตเต็มหน่วยแบบ 1 มิติ

การแปลงเวฟเล็ตเต็มหน่วยแบบ 1 มิติ โดยใช้หลักการฟิลเตอร์แบงค์ (filter bank) มีหลักการวิเคราะห์ (analysis) สัญญาณด้วยฟิลเตอร์แบงค์ เป็นดังภาพที่ 2.3 กำหนดให้ x(n) คือ ข้อมูลอินพุต$h_0(n)$ คือ ตัวกรองความถี่ต่ำผ่าน $h_1(n)$ คือ ตัวตัวกรองความถี่สูงผ่าน และมีแผนภาพการลดข้อมูลครึ่งหนึ่ง (down sampling) เป็นดังภาพที่ 2.4 สามารถเขียนเป็นสมการวิเคราะห์ได้ดังสมการที่ 2.5 และ2.6[3]

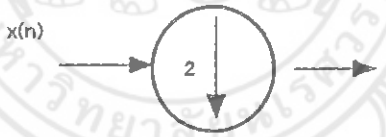
$$c(k) = \sum_m h_0\,(2k - m)x(m) \tag{2.5}$$

$$d(k) = \sum_m h_1\,(2k - m)x(m) \tag{2.6}$$

เมื่อ c คือ สัมประสิทธิ์เวฟเล็ตขององค์ประกอบความถี่ต่ำ และ d คือ สัมประสิทธิ์เวฟเล็ตขององค์ประกอบความถี่สูง

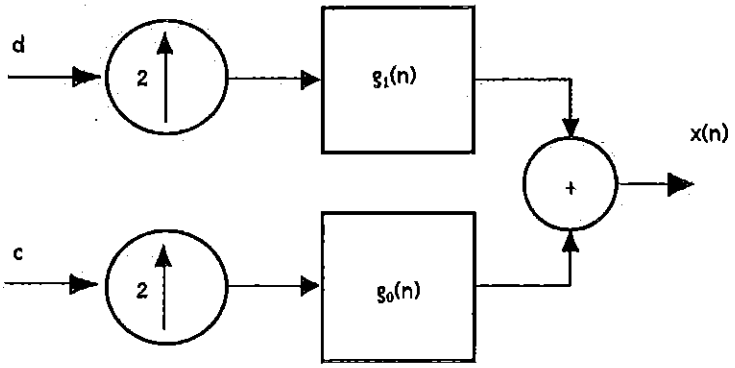


รูปที่ 2.3 แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ 1 มิติ[3]

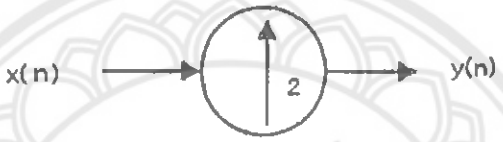


รูปที่ 2.4 แผนภาพการลดข้อมูลลงครึ่งหนึ่ง[3]

และมีหลักการสังเคราะห์ (Synthesis) สัญญาณด้วยฟิลเตอร์แบงค์เป็นดังภาพที่ 2.5 กำหนดให้ $g_0(n)$ คือ ตัวกรองความถี่ต่ำผ่าน และ $g_1(n)$ คือ ตัวกรองความถี่สูงผ่าน และมีแผนภาพการเพิ่มข้อมูลขึ้นเป็นสองเท่า (Up sampling) เป็นดังภาพที่ 2.6 สามารถเขียนเป็นสมการสังเคราะห์ได้ดังสมการที่ 2.5[3]



รูปที่ 2.5 แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 1 มิติ[3]



รูปที่ 2.6 แผนภาพการเพิ่มข้อมูลขึ้นเป็นสองเท่า[3]

การวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบหลายแบนด์ย่อย สามารถทำได้โดยการนำเอาการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบ 1 มิติแบบ 1 แบนด์ย่อยมาประยุกต์ใช้ได้ดังสมการที่ 2.7และ2.8 และมีแผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบหลายแบนด์ย่อยดังภาพที่ 2.7 ซึ่งภาพที่ 2.7 เป็นการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบ 3 แบนด์ย่อย[3]

$$c_j(k) = \sum_m h_0\,(2k-m)c_{j+1}(m) \tag{2.7}$$

$$d_j(k) = \sum_m h_1\,(2k-m)c_{j+1}(m) \tag{2.8}$$

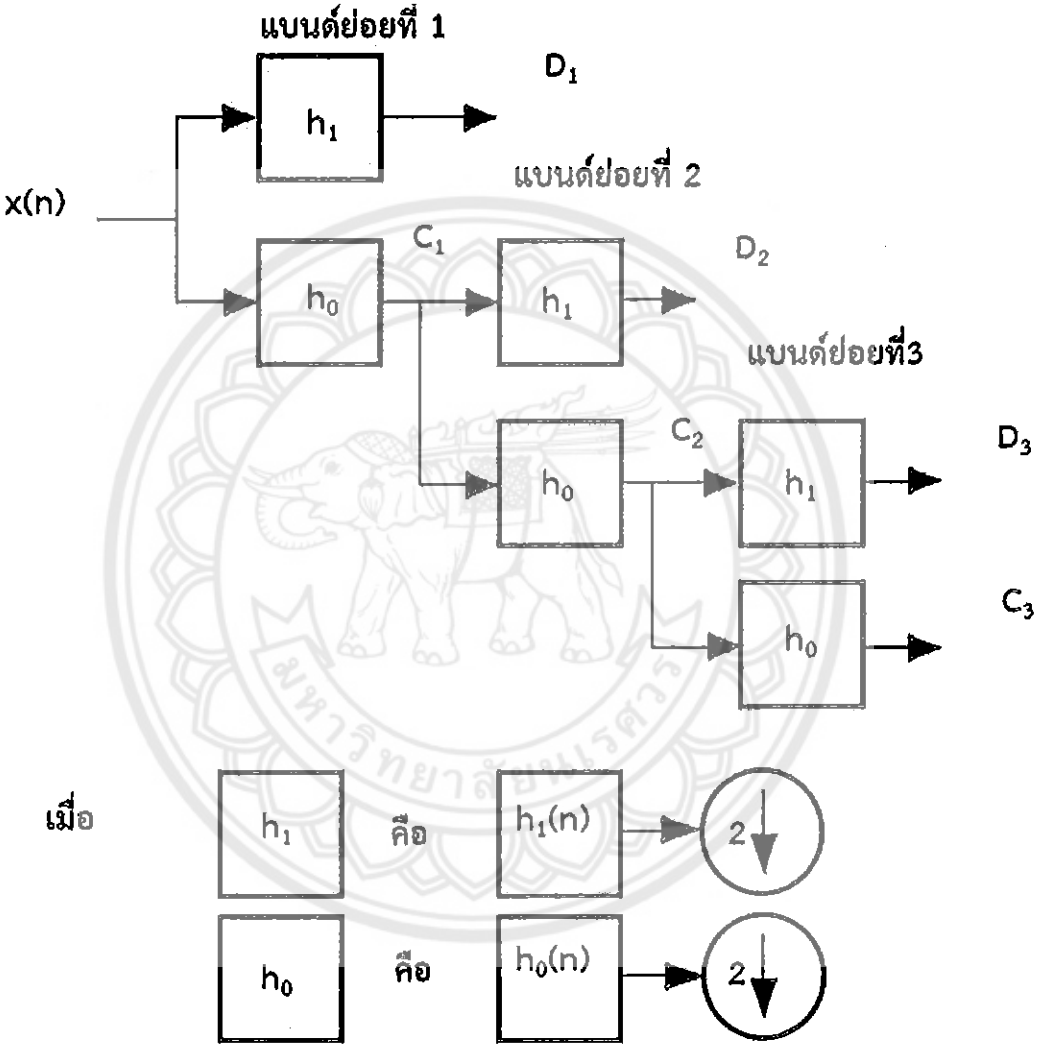
เมื่อ c_j คือ สัมประสิทธิ์เวฟเล็ตขององค์ประกอบความถี่ต่ำ ที่แบนด์ย่อยที่ j

d_j คือ สัมประสิทธิ์เวฟเล็ตขององค์ประกอบความถี่สูง ที่แบนด์ย่อยที่ j

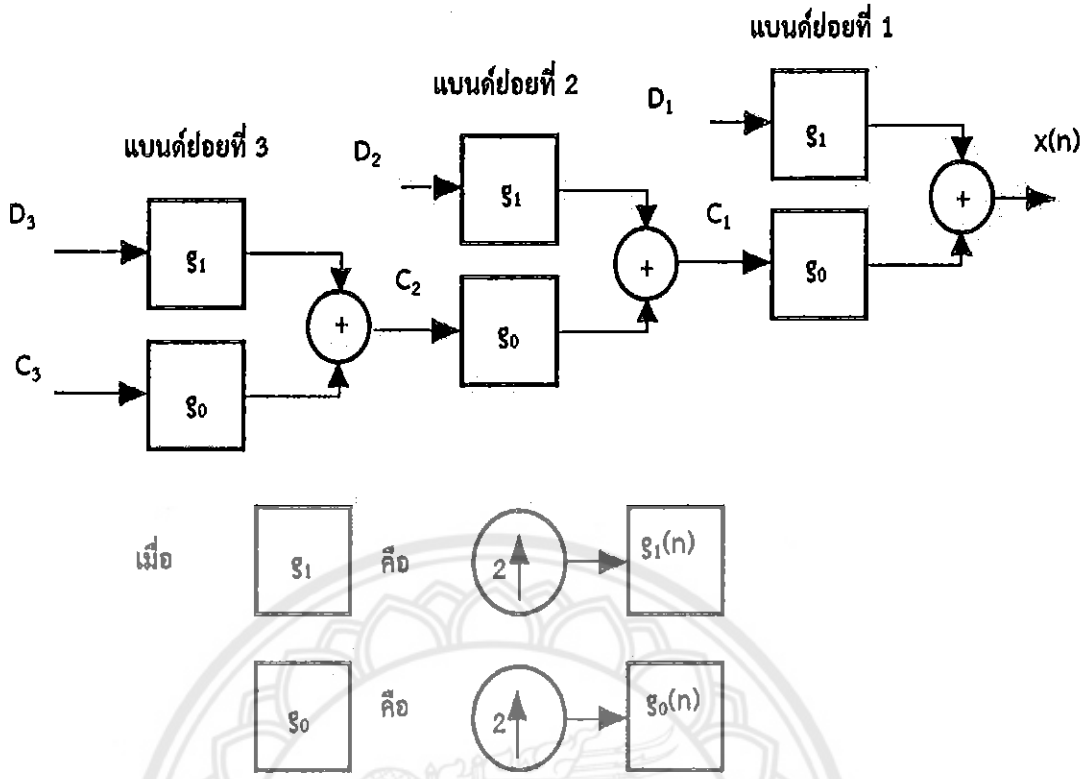
การสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบ 1 มิติ แบบ 1 แบนด์ย่อย มาประยุกต์ใช้ได้ดังสมการที่ 2.9 และ 2.10 และมีแผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบหลายแบนด์ย่อยดังภาพที่ 2.8 ซึ่งภาพที่ 2.8 เป็นการสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์แบบ 3 แบนด์ย่อย[3]

$$x(n) = \sum_m c_1(m)\, g_0(n-2m) + \sum_m d_1(m)\, g_1(n-2m) \tag{2.9}$$

$$c_{j+1}(k) = \sum_p c_j(p)\, g_0(k-2p) + \sum_p d_j(p)\, g_1(k-2p) \tag{2.10}$$



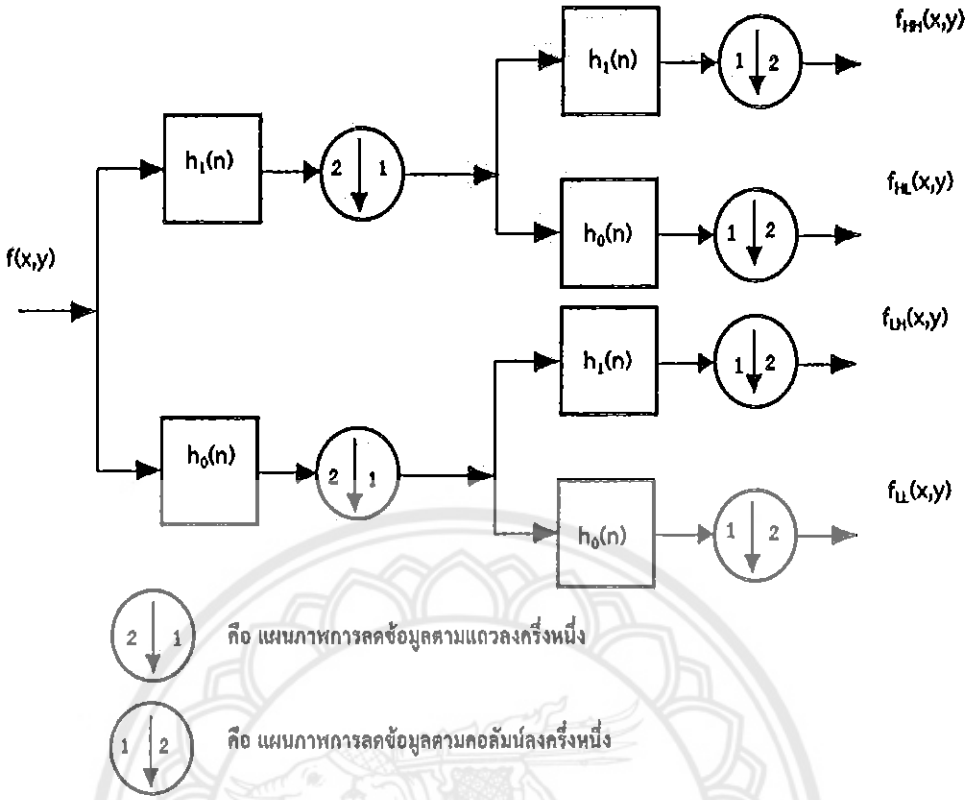
รูปที่ 2.7 แผนภาพการวิเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 3 แบนด์ย่อย [3]



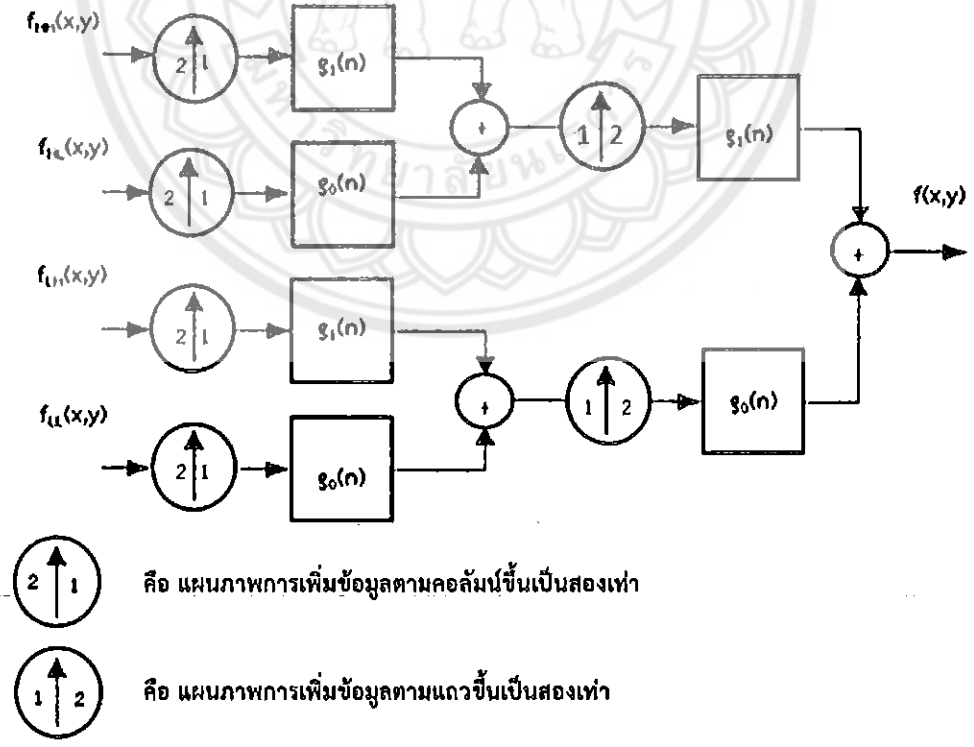
รูปที่ 2.8 แผนภาพการสังเคราะห์สัญญาณด้วยฟิลเตอร์แบงค์ แบบ 3 แบนด์ย่อย[3]

2.6.2 การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ

การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ใช้หลักการแยกองค์ประกอบเป็นแบนด์ย่อย (Subband decomposition) (Stephane, 1989; Marc, Michel, Pierre, and Ingrid, 1992) มีหลักการประมวลผลภาพ โดยใช้การแปลงเวฟเล็ตออกเป็นสองขั้นตอนคือ การแยกองค์ประกอบด้วยเวฟเล็ต มีวิธีการแปลงเป็นดังภาพที่ 2.9 กำหนดให้ $f(x,y)$ คือภาพต้นฉบับ $f_{LL}(x,y)$ $f_{LH}(x,y)$ $f_{HL}(x,y)$ และ$f_{HH}(x,y)$ คือสัมประสิทธิ์เวฟเล็ต และการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต เป็นดังภาพที่ 2.10 จากภาพที่ 2.9 ภาพต้นฉบับจะถูกกรองไปตามแนวแถวด้วยฟิลเตอร์ $h_0(n)$ และ $h_1(n)$ และลดจำนวนข้อมูลลงครึ่งหนึ่ง โดยการเก็บข้อมูล 1 คอลัมน์จากข้อมูล 2 คอลัมน์ จะได้เอาต์พุทความถี่สูง และความถี่ต่ำ จากนั้นเอาต์พุททั้งสองจะถูกกรองตามแนวคอลัมน์ด้วยฟิลเตอร์ $h_0(n)$ และ $h_1(n)$ และลดจำนวนข้อมูลลงครึ่งหนึ่ง โดยการเก็บข้อมูลมา 1 แถวจากข้อมูล 2 แถวจะได้ภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ $f_{LL}(x,y)$ จำนวนหนึ่งภาพ และภาพของสัมประสิทธิ์แบนด์ย่อยความถี่สูงจำนวน 3 ภาพคือ $f_{HL}(x,y)$ $f_{LH}(x,y)$ และ$f_{HH}(x,y)$ โดยแต่ละภาพจะมีขนาดลดลงครึ่งหนึ่งในสี่ของภาพต้นฉบับ[3]

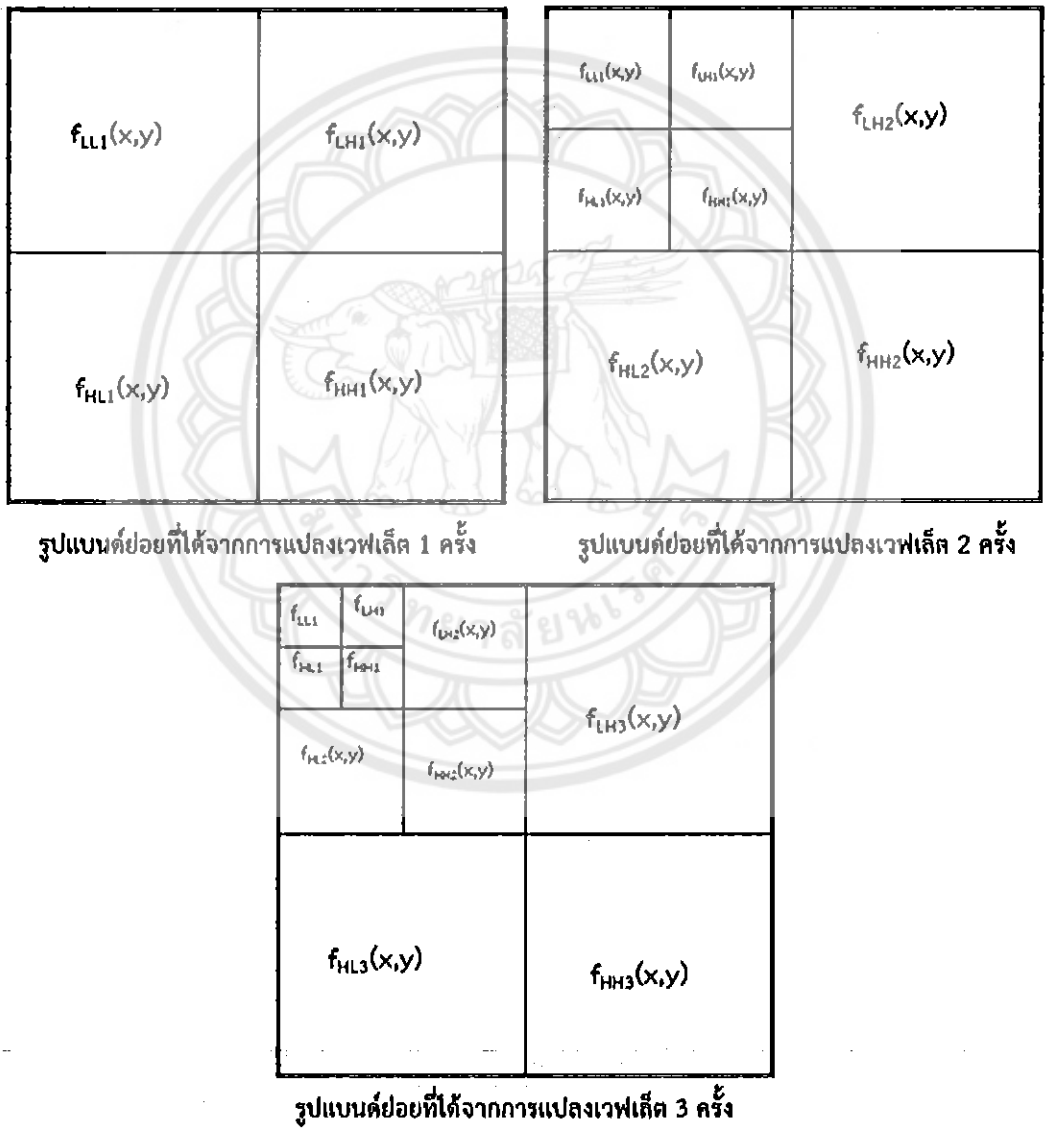


รูปที่ 2.9 แผนภาพการแยกองค์ประกอบด้วยเวฟเล็ต[3]



รูปที่ 2.10 แผนภาพการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต[3]

จากรูปที่ 2.10 การสร้างภาพขึ้นมาใหม่จากองค์ประกอบแบนด์ย่อย เริ่มจากภาพของสัมประสิทธิ์แบนด์ย่อยทั้ง 4 ภาพจะถูกเพิ่มจำนวนข้อมูลขึ้นเป็น 2 เท่า โดยการเติมศูนย์เข้าไปในระหว่างแต่ละคู่ของแถว หลังจากเพิ่มจำนวนข้อมูลแล้วผลลัพธ์ที่ได้จะถูกกรองไปตามคอลัมน์ด้วยตัวกรอง $g_0(n)$ และ $g_1(n)$ และบวกเข้าด้วยกันตามรูปที่ 2.10 จากนั้นนำผลลัพธ์ที่ได้มาเติมศูนย์เข้าไปในระหว่างแต่ละคู่ของคอลัมน์ หลังจากเพิ่มจำนวนข้อมูลแล้วผลลัพธ์ที่ได้จะถูกกรองไปตามแถวด้วยตัวกรอง $g_0(n)$ และ $g_1(n)$ และบวกเข้าด้วยกันตามรูปที่ 2.10 ผลลัพธ์ที่ได้คือภาพต้นฉบับที่กู้คืนกลับขึ้นมาใหม่



รูปแบนด์ย่อยที่ได้จากการแปลงเวฟเล็ต 1 ครั้ง

รูปแบนด์ย่อยที่ได้จากการแปลงเวฟเล็ต 2 ครั้ง

รูปแบนด์ย่อยที่ได้จากการแปลงเวฟเล็ต 3 ครั้ง

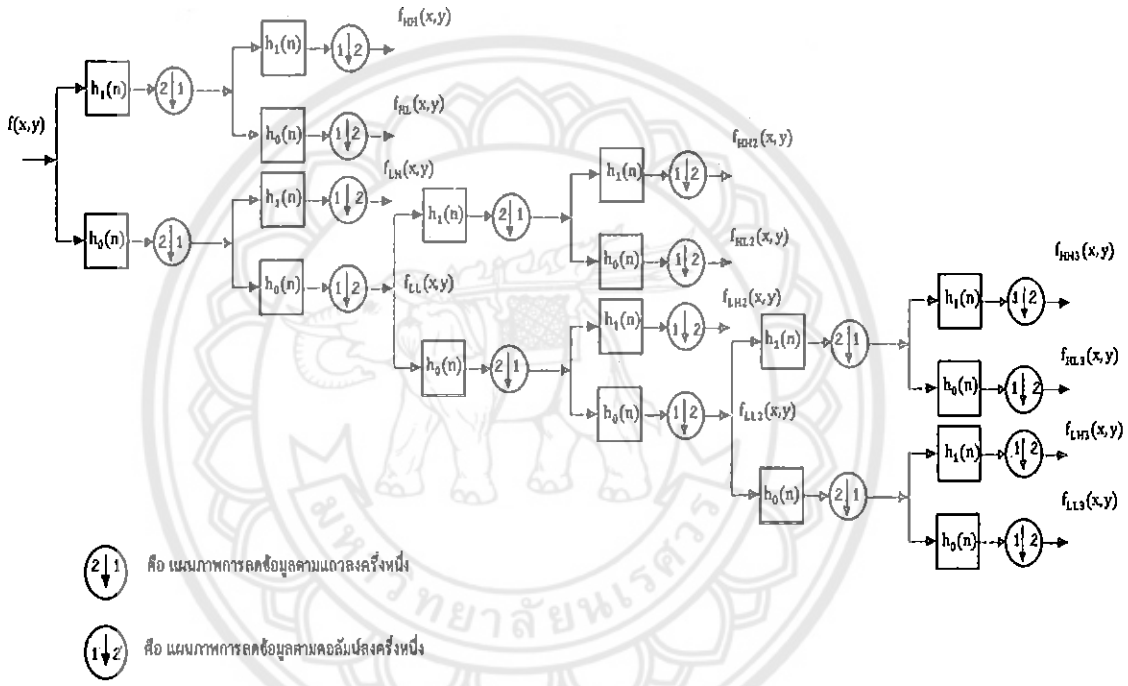
รูปที่ 2.11 แสดงรูปแบนด์ย่อยที่ผ่านการแปลงเวฟเล็ตหลายครั้ง[3]

การแปลงเวฟเล็ตของข้อมูลภาพแบบหลายครั้งเป็นดังรูปที่ 2.11 ซึ่งรูปที่ 2.11 จะแสดงรูปแบนด์ย่อยที่ผ่านการแปลงเวฟเล็ตหลายครั้ง

2.6.3 การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ครั้ง

การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ครั้ง ใช้หลักการแยกองค์ประกอบเป็นแบนด์ย่อย (Subband decomposition) มีหลักการประมวลผลภาพ โดยใช้การแปลงเวฟเล็ตออกเป็นสองขั้นตอนคือ การแยกองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง มีวิธีการแปลงเป็นดังภาพที่ 2.12 และการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง มีวิธีการแปลงเป็นดังภาพที่ 2.13

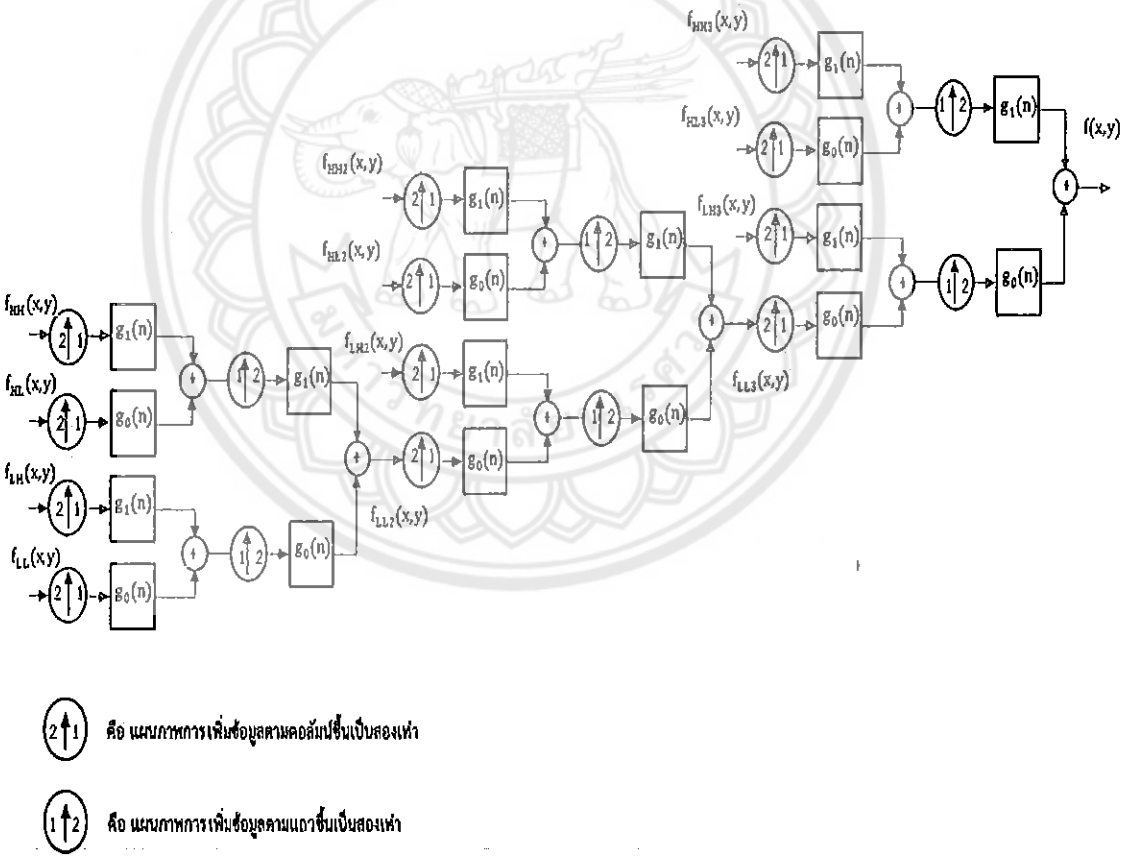
การแปลงเวฟเล็ตแบบ 2 มิติ 3 ครั้ง เป็นดังรูปที่ 2.12 ซึ่งรูปที่ 2.12 จะแสดงแผนภาพการแยกองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง



รูปที่ 2.12 แผนภาพการแยกองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง[3]

จากรูปที่ 2.12 เริ่มต้นจากการนำภาพต้นฉบับมาผ่านตัวกรองความถี่ต่ำ และตัวกรองความถี่สูงและลดข้อมูลลงครึ่งหนึ่ง จะได้ผลลัพธ์ความถี่สูงและความถี่ต่ำ จากนั้นผลลัพธ์ทั้งสองจะถูกกรองตามแนวคอลัมน์ด้วยตัวกรองความถี่ต่ำและตัวกรองความถี่สูง และลดจำนวนข้อมูลลงครึ่งหนึ่ง จะได้ภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ $f_{LL1}(x,y)$ จำนวนหนึ่งภาพ และภาพของสัมประสิทธิ์แบนด์ย่อยความถี่สูงจำนวน 3 ภาพคือ $f_{HL1}(x,y)$ $f_{LH1}(x,y)$ และ$f_{HH1}(x,y)$ โดยแต่ละภาพจะมีขนาดลดลงครึ่งหนึ่งในสี่ของภาพเมตริกซ์ต้นฉบับ และนับว่าเป็นการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติครั้งที่ 1 ต่อมาจะนำค่าสัมประสิทธ์ $f_{LL1}(x,y)$ มาผ่านตัวกรองความถี่ต่ำและตัวกรองความถี่สูงและลดข้อมูลลงครึ่งหนึ่ง จะได้ผลลัพธ์ความถี่สูง และความถี่ต่ำ จากนั้นผลลัพธ์ทั้งสองจะถูกกรองตามแนวคอลัมน์ด้วยตัวกรองความถี่ต่ำและตัวกรองความถี่สูง และลด

จำนวนข้อมูลลงครึ่งหนึ่งจะได้ภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ $f_{LL2}(x,y)$ จำนวนหนึ่งภาพ และภาพของสัมประสิทธิ์แบนด์ย่อยความถี่สูงจำนวน 3 ภาพคือ $f_{HL2}(x,y)$ $f_{LH2}(x,y)$ และ $f_{HH2}(x,y)$ โดยแต่ละภาพจะมีขนาดลดลงครึ่งหนึ่งในสี่ของภาพเมตริกซ์ต้นฉบับ และนับว่าเป็นการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ครั้งที่ 2ต่อไปจะนำค่าสัมประสิทธิ์ $f_{LL2}(x,y)$ มาผ่านตัวกรองความถี่ต่ำ และตัวกรองความถี่สูงและลดข้อมูลลงครึ่งหนึ่ง จะได้ผลลัพธ์ความถี่สูง และความถี่ต่ำ จากนั้นผลลัพธ์ทั้งสองจะถูกกรองตามแนวคอลัมน์ด้วยตัวกรองความถี่ต่ำและตัวกรองความถี่สูง และลดจำนวนข้อมูลลงครึ่งหนึ่ง จะได้ภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ $f_{LL3}(x,y)$ จำนวนหนึ่งภาพ และภาพของสัมประสิทธิ์แบนด์ย่อยความถี่สูงจำนวน 3 ภาพคือ $f_{HL3}(x,y)$ $f_{LH3}(x,y)$ และ$f_{HH3}(x,y)$ โดยแต่ละภาพจะมีขนาดลดลงครึ่งหนึ่งในสี่ของภาพเมตริกซ์ต้นฉบับ และนี่นับว่าเป็นการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ครั้งที่ 3สุดท้ายจะได้ผลลัพธ์เมตริกซ์สัมประสิทธิ์เวฟเล็ตทั้ง 3 ครั้ง



รูปที่ 2.13 แผนภาพการสร้างกลับจากองค์ประกอบด้วยเวฟเล็ต 3 ครั้ง[3]

ส่วนการแปลงเวฟเล็ตเต็มหน่วยผกผันแบบ 2 มิติ เริ่มจากภาพของสัมประสิทธิ์กู้คืน ซึ่งแบนด์ย่อยทั้ง 4 ภาพจะถูกเพิ่มจำนวนข้อมูลขึ้นเป็น 2 เท่าแล้วผลลัพธ์ที่ได้จะถูกกรองไปตามคอลัมน์ด้วยตัวกรองความถี่ต่ำและตัวกรองความถี่สูง และบวกเข้าด้วยกัน จากนั้นนำผลลัพธ์ที่ได้เพิ่มจำนวนข้อมูลแล้ว ผลลัพธ์ที่ได้จะถูกกรองไปตามแถวด้วยวกรองความถี่ต่ำ และตัวกรองความถี่สูง และบวกเข้าด้วยกัน ผลลัพธ์ที่ได้คือภาพต้นฉบับของครั้งที่ 3 นับว่าเป็นการแปลงกลับ 1 ครั้ง ทำซ้ำแบบนี้อีกสองรอบ สุดท้ายจะได้ผลลัพธ์ภาพต้นฉบับของครั้งที่ 1

2.7 การเลือกใช้ตัวกรองเวฟเล็ตในการศึกษาการบีบอัดภาพ

ในการศึกษาการบีบอัดข้อมูลภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยประสิทธิภาพในการบีบอัดนั้นจะขึ้นอยู่กับการเลือกใช้เวฟเล็ตเพื่อทำการบีบอัดข้อมูลภาพ เนื่องจากมีตัวกรองเวฟเล็ตให้เลือกใช้จำนวนมาก และมีคุณสมบัติในการประยุกต์ใช้ที่ต่างกัน

เวฟเล็ตตัวกรองฮาร์ (Haar) มีความเหมาะสมในการนำมาใช้บีบอัดข้อมูลที่ซ้ำกันหรือข้อมูลที่ไม่มีการเปลี่ยนแปลงทันทีทันใด และเวฟเล็ตตัวกรองเดบุชชี (Daubechies) มีความเหมาะสมในการนำมาใช้บีบอัดข้อมูลที่ไม่ค่อยซ้ำกันหรือข้อมูลที่มีการเปลี่ยนแปลงทันทีทันใด

ในโครงงานนี้ผู้วิจัยได้ทำการศึกษาการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยภาพที่นำมาทดสอบนั้นมีทั้งข้อมูลที่มีการเปลี่ยนแปลงทันทีทันใดและไม่เปลี่ยนแปลงทันทีทันใด จึงได้นำเวฟเล็ตฮาร์และเดบุชชี4 มาใช้ในการศึกษาการบีบอัดข้อมูลภาพ โดยตัวกรองที่ใช้มีรายละเอียดดังต่อไปนี้

2.7.1 ตัวกรองเวฟเล็ตฮาร์ (Haar Wavelet Filter)

เวฟเล็ตฮาร์เป็นเวฟเล็ตชนิดแรกที่เริ่มใช้กันและเมื่อเร็วๆนี้การแปลงฮาร์ถูกมองว่าเป็นการแปลงเวฟเล็ตที่ง่าย เวฟเล็ตชนิดนี้เป็นชนิดไม่ต่อเนื่องกันและมีลักษณะคล้ายกับฟังก์ชั่นขั้นบันได[4] ฮาร์มีสเกลลิ่งฟังก์ชั่น (Scaling Function) $\emptyset(x)$ และเวฟเล็ตฟังก์ชั่น (Wavelet Function) $\varphi(x)$ ดังสมการต่อนี้

$$\emptyset(x) = \emptyset(2x) + \emptyset(2x-1) = \frac{1}{\sqrt{2}}\sqrt{2}\ \emptyset(2x) + \frac{1}{\sqrt{2}}\sqrt{2}\ \emptyset(2x-1) \qquad (2.11)$$

ดังนั้นตัวกรองของสเกลลิ่งฟังก์ชั่นนี้ คือ

$$h_0 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$$

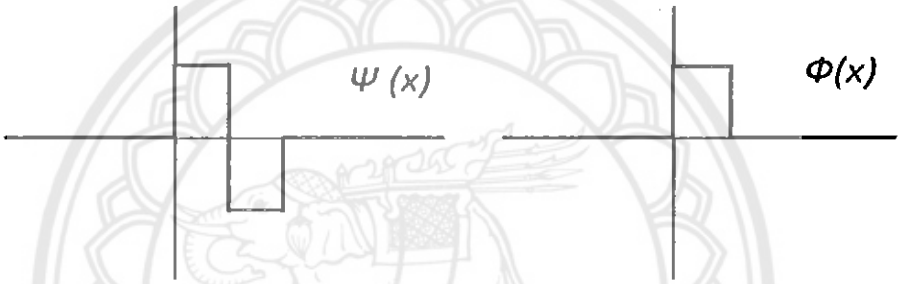
เช่นเดียวกับฮาร์เวฟเล็ตฟังก์ชั่นจะได้เป็น

$$\varphi(x) = \emptyset(2x) - \emptyset(2x-1) = \frac{1}{\sqrt{2}}\sqrt{2}\ \emptyset(2x) + \frac{1}{\sqrt{2}}\sqrt{2}\ \emptyset(2x-1) \quad (2.12)$$

ตัวกรองของฮาร์เวฟเล็ตฟังก์ชั่นนี้ คือ

$$h_0 = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$$

และตามรูปที่ 2.14 เช่นเดียวกับฟังก์ชั่นเวฟเล็ตของเรา ω(x) เราสามารถบีบอัดและขยายเวฟเล็ตนี้ในแนวนอนหรือแนวตั้งและเลื่อนได้



รูปที่ 2.14 เวฟเล็ตฮาร์ และพัลส์ฟังก์ชั่น[4]

2.7.2 ตัวกรองเวฟเล็ตเดบุชชี4 (Daubechies 4 Wavelet Filter)

เดบุชชี4 มีสเกลลิ่งฟังก์ชั่น (Scaling Function) $\emptyset(x)$ และเวฟเล็ตฟังก์ชั่น(Wavelet Function) $\varphi(x)$ ดังสมการต่อนี้

$$\emptyset(x) = h_0\emptyset(2x) + h_1\emptyset(2x-1) + h_2\emptyset(2x-2) + h_1\emptyset(2x-2) \quad (2.13)$$

$$\varphi(x) = h_0\emptyset(2x-1) - h_1\emptyset(2x) + h_2\emptyset(2x+1) + h_3\emptyset(2x+2) \quad (2.14)$$

ขณะที่ค่าของสัมประสิทธิ์ตัวกรองคือ

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \approx 0.48296$$
$$h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \approx 0.83652$$
$$h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \approx 0.22414$$
$$h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \approx -0.12941$$

จากสี่เวกเตอร์ประกอบด้วยค่าที่เหมือนกันแต่แตกต่างตามลำดับและเครื่องหมาย เราสามารถประยุกต์ใช้เวฟเล็ตเดบุชชี4 โดยการคูณเมตริกซ์ ขณะที่เมตริกซ์สำหรับการแปลงเวฟเล็ตเต็มหน่วยแบบ 1 มิติ เขียนอยู่ในรูปเมตริกซ์ขนาด 8*8[4]

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_3 & -h_2 & h_1 & -h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_3 & -h_2 & h_1 & -h_0 \\ h_1 & -h_0 & 0 & 0 & 0 & 0 & h_3 & -h_2 \end{bmatrix}$$

พบว่าค่าสัมประสิทธิ์ตัวกรองที่ทับซ้อนกันระหว่างแถว ซึ่งไม่ได้เป็นเหมือนกรณีเมตริกซ์ฮาร์ ถ้าใช้เวฟเล็ตเดบุชชี่4 ค่าเฉลี่ยของสัมประสิทธิ์จะเรียบเนียนกว่าใช้เวฟเล็ตฮาร์ รูปแบบของเมตริกซ์ ดังกล่าวข้างต้นจะคล้ายกับหมุนวนเป็นวงกลมกับตัวกรองหนึ่งมิติ

$[\, h_0\ h_1\ h_2\ h_3 \,]$ และ $[\, h_3\ -h_2\ h_1\ -h_0 \,]$ การแปลงเวฟเล็ตเต็มหน่วย(Discrete Wavelet Transform: DWT) สามารถมองมันในรูปแบบตัวกรองนั่นเองซึ่งก็คือ 2 ตัวกรองที่เป็นที่รู้จักคือตัวกรองต่ำผ่าน (Low-Pass) และตัวกรองสูงผ่าน (High-Pass)

2.8 การเข้ารหัสเลขคณิต (Arithmetic Coding)

การเข้ารหัสเลขคณิตเป็นอัลกอริทึมที่ใช้ในการบีบอัดข้อมูลแบบไม่มีการสูญเสีย ใช้ในกรณีที่ผู้บีบอัดไม่ต้องการให้เกิดการสูญเสียของข้อมูลเกิดขึ้นเนื่องจากข้อมูลเหล่านั้นยากต่อการได้มา โดยเฉพาะอย่างยิ่งในทางการแพทย์ และทหาร ปกติสายอักขระ เช่น คำว่า "hello there" จะใช้แทนค่าคงที่ของจำนวนบิตต่อตัวอักษรเช่นเดียวกับในรหัสแอสกี (ASCII) เมื่อสตริงถูกแปลงเข้าเป็นรหัสเลขคณิตแล้วตัวอักษรที่เกิดขึ้นบ่อยจะถูกเก็บไว้กับจำนวนบิตที่น้อยลง และส่วนตัวอักษรที่เกิดขึ้นไม่บ่อยครั้งจะถูกเก็บไว้กับจำนวนบิตที่มากขึ้น ส่งผลให้ใช้จำนวนบิตลดลง

i 719 694 ปร ภาษา 2557
12 ต.ค. 2560
สำนักหอสมุด

2.8.1 การเข้ารหัสอิมพลีเมนต์เตชั่น (Encoder Implementation)

หลักการทำงานทั่วไปเหมือนกันกับการเข้ารหัสเลขคณิต แต่วิธีนี้สามารถแก้ไขปัญหาหน่วยความจำที่ถูกใช้ไปเป็นอย่างมากในการเก็บข้อมูลจุดทศนิยม โดยการเปลี่ยนขอบบน (upper bound) และขอบล่าง (lower bound) จากจุดทศนิยมไปเป็นจำนวนเต็มในรูปแบบเลขฐานสอง เริ่มต้นจากการตัดสินใจหาความยาวเวิร์ด (word length) ซึ่งถูกแทนด้วยตัว m เราจะทำการแมปค่าที่สำคัญในช่วง [0,1) จนถึงช่วงของเวิร์ดฐานสอง 2^m

ถ้าค่าเป็น 0 จะแมปค่าได้ 00…0 แล้วแต่ความยาวของเวิร์ดนั้นๆ
ถ้าค่าเป็น 1 จะแมปค่าได้ 11…1 แล้วแต่ความยาวของเวิร์ดนั้นๆ
ถ้าค่าเป็น 0.5 จะแมปค่าได้ 100…0 แล้วแต่ความยาวของเวิร์ดนั้นๆ
เราจะนิยามให้ n_i คือจำนวนครั้งของสัญลักษณ์ (symbol) i ที่เกิดขึ้นในลำดับของความยาวจำนวนนับทั้งหมด (Total count) ดังนั้น $F_x(k)$ สามารถเขียนได้ดังนี้

$$F_x(k) = \frac{\sum_{i=1}^{k} n_i}{Total\ count} \tag{2.15}$$

เราจะให้นิยามเพิ่ม

$$\text{Cum_}Count = \sum_{i=1}^{k} n_i \tag{2.16}$$

กำหนดให้จำนวนนับของแต่ละสัญลักษณ์ (Cum_Count) n_i

เราสามารถหาค่าขอบบน และขอบล่างได้จากสมการข้างล่างนี้ คือ

$$l^n = l^{n-1} + \left[\frac{(u^{n-1} - l^{n-1} + 1) * Cum_{Count(x_n - 1)}}{Total\ Count}\right] \tag{2.17}$$

$$u^n = l^{n-1} + \left[\frac{(u^{n-1} - l^{n-1} + 1) * Cum_{Count(x_n)}}{Total\ Count}\right] - 1 \tag{2.18}$$

โดยที่ x_n คือ ลำดับที่ n ของสัญลักษณ์ที่ทำการเข้ารหัส

เราสามารถทำการหาความยาวเวิร์ดได้จากสมการข้างล่างนี้ คือ

$$m = [log_2\ Total\ Count] + 2 \tag{2.19}$$

ข้อมูลที่ได้จากการส่งเกิดขึ้นจาก 3 เงื่อนไข ดังนี้

1) เงื่อนไขที่ 1 (E1 condition) คือ เมื่อบิตที่มีนัยสำคัญมากที่สุด (Most significant bit : MSB) ของขอบล่าง และขอบบนเป็น "0" เหมือนกัน จะทำการส่งบิต"0"แล้วทำการเลื่อนบิตซ้ายมือสุดของทั้งขอบล่างและขอบบนไป 1 บิต และทำการเติมบิต "0" ไปที่บิตที่มีนัยสำคัญต่ำที่สุด (Least significant bit : LSB)ของขอบล่าง และเติมบิต "1" ไปที่บิตที่มีนัยสำคัญต่ำที่สุดของขอบบน

2) เงื่อนไขที่ 2 (E2 condition) คือ เมื่อบิตที่มีนัยสำคัญมากที่สุดของขอบล่าง และขอบบนเป็น "1" เหมือนกันจะทำการส่งบิต "1" แล้วทำการเลื่อนบิตซ้ายมือสุดของทั้งขอบล่าง และขอบบนไป 1 บิต และทำการเติมบิต "0" ไปที่บิตที่มีนัยสำคัญต่ำที่สุด ของขอบล่าง และเติมบิต "1" ไปที่บิตที่มีนัยสำคัญต่ำที่สุดของขอบบน

3) เงื่อนไขที่ 3 (E3 condition) คือ เมื่อบิตที่มีนัยสำคัญมากที่สุดที่ 2 ของขอบล่างเป็น"0" และบิตที่มีนัยสำคัญมากที่สุดที่ 2 ของขอบบนเป็น"1"แล้วทำการเลื่อนบิตซ้ายมือสุดแล้วให้ทำการกลับค่าบิตของบิตที่มีนัยสำคัญมากที่สุดที่ 2 ของทั้งขอบล่าง และขอบบนพร้อมทั้งทำการเติมบิต"0"ไปที่บิตที่มีนัยสำคัญต่ำที่สุดของขอบล่าง และเติมบิต"1"ไปที่บิตที่มีนัยสำคัญต่ำที่สุดของขอบบน เมื่อทราบเงื่อนไขทั้ง 3 เงื่อนไขแล้วให้ทำการตรวจสอบทั้ง 3 เงื่อนไขนี้ต่อไปเรื่อยๆ จนไม่ตรวจพบทั้ง 3 เงื่อนไขจึงทำการเข้ารหัสตัวต่อไปได้

2.8.2 การถอดรหัสอิมพลีเมนต์เตชั่น(Decoder Implementation)

ขั้นตอนการถอดรหัสแท็ก มีดังนี้

1) กำหนดค่าเริ่มต้นของขอบล่างและขอบบน $l^0 = 0, u^0 = 2^m - 1$

2) อ่านค่าแรก m บิต ของของกระแสบิตที่ได้รับไปเป็นค่าแท็ก t

3) ทำการคำนวณหาค่าเริ่มต้นที่ใช้ในการถอดรหัส โดยใช้สมการที่ 2.19

$$\left\lfloor \frac{(t - l + 1) * Total\ Count - 1}{u - l + 1} \right\rfloor \qquad (2.20)$$

4) ทำการเปรียบเทียบค่าที่คำนวณได้จากขั้นตอนที่ 3 กับค่าจำนวนนับของแต่ละสัญลักษณ์ (Cum Count) ว่าค่าเริ่มต้นที่ใช้ในการถอดรหัสอยู่ในช่วงใด

5) ทำการอัพเดทค่าขอบบน (l) กับค่าขอบล่าง (u)

6) ถ้ามีมากกว่า 1 สัญลักษณ์ให้ย้อนกลับไปทำขั้นตอนที่ 2 ใหม่ ทำจนครบทุกสัญลักษณ์

บทที่ 3
วิธีดำเนินโครงงาน

เพื่อให้การศึกษาเทคนิคการบีบอัดภาพด้วยการใช้การแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิตได้อย่างมีประสิทธิภาพ ผู้วิจัยมีขั้นตอนในการดำเนินการดังนี้

3.1 ศึกษาและค้นคว้าทฤษฎีพื้นฐานที่เกี่ยวข้อง

เริ่มต้นศึกษาจากภาพเชิงดิจิตอล ประเภทของการบีบอัดข้อมูลภาพ การบีบอัดข้อมูลภาพ การวัดประสิทธิภาพการบีบอัดสัญญาณภาพดิจิตอล การบันทึกไฟล์ข้อมูลฐานสองและการเลือกใช้ตัวกรองเวฟเล็ตที่ใช้ในการศึกษาการบีบอัดภาพ โดยมีการเรียงตามลำดับการศึกษาในหัวข้อที่ 2.1,2.2 , 2.3 , 2.4 และ 2.5 ของบทที่ 2

3.2 ศึกษาการแปลงเวฟเล็ต

ศึกษาการแปลงเวฟเล็ต โดยที่การแปลงเวฟเล็ตในโครงงานนี้เป็นการแปลงเวฟเล็ตแบบเต็มหน่วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 1 มิติ ด้วยหลักการฟิลเตอร์แบงค์ เริ่มต้นจากการนำสัญญาณมาผ่านตัวกรองความถี่ต่ำและตัวกรองความถี่สูงก่อน ซึ่งจะได้สัมประสิทธิ์ค่าประมาณและสัมประสิทธิ์รายละเอียดออกมาตามลำดับ นำสัมประสิทธิ์ค่าประมาณ และสัมประสิทธิ์รายละเอียดไปผ่านการลดข้อมูลครึ่งหนึ่ง นำส่วนที่เป็นสัมประสิทธิ์รายละเอียดไปทำซ้ำจนได้ข้อมูลตามที่ต้องการ

ส่วนการแปลงเวฟเล็ตเต็มหน่วยผกผันแบบ 1 มิติ นำสัมประสิทธิ์ค่าประมาณและสัมประสิทธิ์รายละเอียดที่ผ่านการลดข้อมูลครึ่งหนึ่งไปเพิ่มข้อมูลขึ้นเป็นสองเท่าต่อมานำสัมประสิทธิ์ค่าประมาณและสัมประสิทธิ์รายละเอียดที่เพิ่มข้อมูลไปผ่านตัวกรองความถี่ต่ำ และตัวกรองความถี่สูงและนำส่วนประกอบย่อยๆมาเรียงต่อกันเพื่อประกอบเป็นสัญญาณเดิม โดยมีรายละเอียดอยู่ในหัวข้อ 2.6.1 ของบทที่ 2

การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ใช้หลักการแยกองค์ประกอบเป็นแบนด์ย่อยเริ่มต้นจากการนำภาพต้นแบบมาผ่านตัวกรองความถี่ต่ำ และตัวกรองความถี่สูงและลดข้อมูลลงครึ่งหนึ่งจะได้เอาต์พุทความถี่สูง และความถี่ต่ำ จากนั้นเอาต์พุททั้งสองจะถูกกรองตามแนวคอลัมน์ด้วยตัวกรองความถี่ต่ำและตัวกรองความถี่สูง และลดจำนวนข้อมูลลงครึ่งหนึ่ง จะได้ภาพของสัมประสิทธิ์แบนด์ย่อยความถี่ต่ำคือ $f_{LL}(x,y)$ จำนวนหนึ่งภาพ และภาพของสัมประสิทธิ์แบนด์ย่อยความถี่สูงจำนวน 3 ภาพคือ $f_{HL}(x,y)$ $f_{LH}(x,y)$ และ$f_{HH}(x,y)$ โดยแต่ละภาพจะมีขนาดลดลงครึ่งหนึ่งในสี่ของภาพต้นฉบับ

ส่วนการแปลงเวฟเล็ตเต็มหน่วยผกผันแบบ 2 มิติ เริ่มจากภาพของสัมประสิทธิ์แบนด์ย่อยทั้ง 4 ภาพจะถูกเพิ่มจำนวนข้อมูลขึ้นเป็น 2 เท่าแล้วผลลัพธ์ที่ได้จะถูกกรองไปตามคอลัมน์ด้วยตัวกรองความถี่ต่ำและตัวกรองความถี่สูง และบวกเข้าด้วยกัน จากนั้นนำผลลัพธ์ที่ได้เพิ่มจำนวนข้อมูลแล้ว ผลลัพธ์ที่ได้จะถูกกรองไปตามแถวด้วยตัวกรองความถี่ต่ำ และตัวกรองความถี่สูง และบวกเข้าด้วยกัน ผลลัพธ์ที่ได้คือภาพต้นฉบับที่กู้คืนกลับขึ้นมาใหม่ โดยมีรายละเอียดอยู่ในหัวข้อ 2.6.2 ของบทที่ 2

3.3 ศึกษาการเข้ารหัสเลขคณิต

ศึกษาการเข้ารหัสเลขคณิตแบบอิมพลีเมนต์เตชั่น เริ่มต้นจากการนำแปลงข้อมูลเมตริกซ์เป็นเวิ์กเตอร์แนวนอน ต่อมาจะทำการแมปข้อมูลจากเวิ์กเตอร์แนวนอนไปเป็นลำดับในการเข้ารหัสเลขคณิตหลังจากที่แปลงเมตริกซ์ข้อมูลเป็นเวิ์กเตอร์แนวนอนเสร็จแล้วจะนำข้อมูลของสมาชิกที่ไม่ซ้ำกันเก็บอยู่ในรูปเมตริกซ์ตาราง หลังจากแมปข้อมูลเสร็จแล้วทำการหาค่าความยาวเวิร์ด(m) ต่อมากำหนดค่าเริ่มต้นในการเข้ารหัสเลขคณิตซึ่งกำหนดได้จากความยาวเวิร์ด จะทำการคำนวณในการเข้ารหัสเลขคณิตตามลำดับจนครบ ส่วนข้อมูลที่ได้จากการส่งเกิดขึ้นจาก 3 เงื่อนไข คือ เงื่อนไขที่ 1 ส่งบิต “0” เงื่อนไขที่ 2 ส่งบิต “1” และเงื่อนไขที่ 3 ไม่มีการส่งบิต เมื่อทราบเงื่อนไขทั้ง 3 เงื่อนไขแล้ว ให้ทำการตรวจสอบทั้ง 3 เงื่อนไขไปเรื่อยๆ จนไม่ตรวจพบ ทั้ง 3 เงื่อนไข จึงทำการเข้ารหัสตัวต่อไปได้ ทำแบบนี้จนครบทุกลำดับการเข้ารหัสเลขคณิต โดยมีรายละเอียดอยู่ในหัวข้อ 2.7.1 ของบทที่ 2

การถอดรหัสเลขคณิตแบบอิมพลีเมนต์เตชั่น เริ่มจากกำหนดค่าเริ่มต้นในการถอดรหัสเลขคณิต อ่านค่าแรก m บิต ของกระแสบิตที่ได้รับไปเป็นค่าแท็ก t แล้วคำนวณหาค่าเริ่มต้นที่ใช้เปรียบเทียบสำหรับการถอดรหัส ต่อมาทำการเปรียบเทียบค่าที่คำนวณได้กับค่าจำนวนนับสะสม ว่าค่าเริ่มต้นที่ใช้ในการถอดรหัสว่าอยู่แล้วทำการอัพเดทค่าในช่วงใด แล้วอัพเดทค่า สัญลักษณ์ต่อไปทำการอ่านค่าถัดไป m บิต ทำจนครบทุกสัญลักษณ์ โดยมีรายละเอียดอยู่ในหัวข้อ 2.7.2 ของบทที่ 2

3.4 ออกแบบโปรแกรมการบีบอัดภาพ

การออกแบบโปรแกรมการบีบอัดภาพด้วยการใช้การแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ และตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต เพื่อเปรียบเทียบประสิทธิภาพของทั้ง 2 ชนิดตัวกรอง โดยใช้ CR , MSE และ PSNR กับ SNR ซึ่งมีขั้นตอนในการดำเนินการดังนี้

3.4.1 การบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต

รูปที่ 3.1 แผนภาพการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต

ขั้นตอนในการดำเนินการบีบอัดภาพมี 6 ขั้นตอน ดังนี้

ขั้นตอนที่หนึ่ง: เริ่มต้นรับภาพระดับเทาหรือภาพสีเข้ามาจัดเก็บในพื้นที่ทำงาน(workspace) ของโปรแกรม Matlab ภาพระดับเทาที่เข้ามาจะอยู่ในรูปเมตริกซ์ระดับเทา สำหรับภาพสีจะต้องแยกออกเป็นเมตริกซ์สามสีหลักก่อน คือ เมตริกซ์สีแดง เมตริกซ์สีเขียว และเมตริกซ์สีน้ำเงิน

ขั้นตอนที่สอง: รับภาพจากพื้นที่ทำงานที่ได้จากขั้นตอนที่หนึ่ง ต่อมาแล้วนำภาพระดับเทาที่เข้ามาในรูปเมตริกซ์ระดับเทา หรือภาพสีที่แยกออกเป็นเมตริกซ์สามสีหลักก่อน คือ เมตริกซ์สีแดง เมตริกซ์สีเขียว และเมตริกซ์สีน้ำเงิน มาแปลงด้วยเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ระดับ (Discrete Wavelet Transform: DWT) โดยใช้ตัวกรองฮาร์ (haar) ซึ่งจะได้ผลลัพธ์เมตริกซ์สัมประสิทธิ์เวฟเล็ตทั้ง 3 ระดับ โดยมีรายละเอียดอยู่ในหัวข้อ 2.6.1 ของบทที่ 2

ขั้นตอนที่สาม: จากนั้นนำเมตริกซ์สัมประสิทธิ์เวฟเล็ตทั้ง 3 ระดับ ของทั้งเมตริกซ์สัมประสิทธิ์เวฟเล็ตระดับเทา ในส่วนเมตริกซ์สัมประสิทธิ์เวฟเล็ตสามสีหลักทั้ง 3 ระดับ แล้วนำเมตริกซ์สัมประสิทธิ์เวฟเล็ตทั้งของภาพระดับเทา และภาพสีมาแปลงเป็นเวกเตอร์แนวนอน

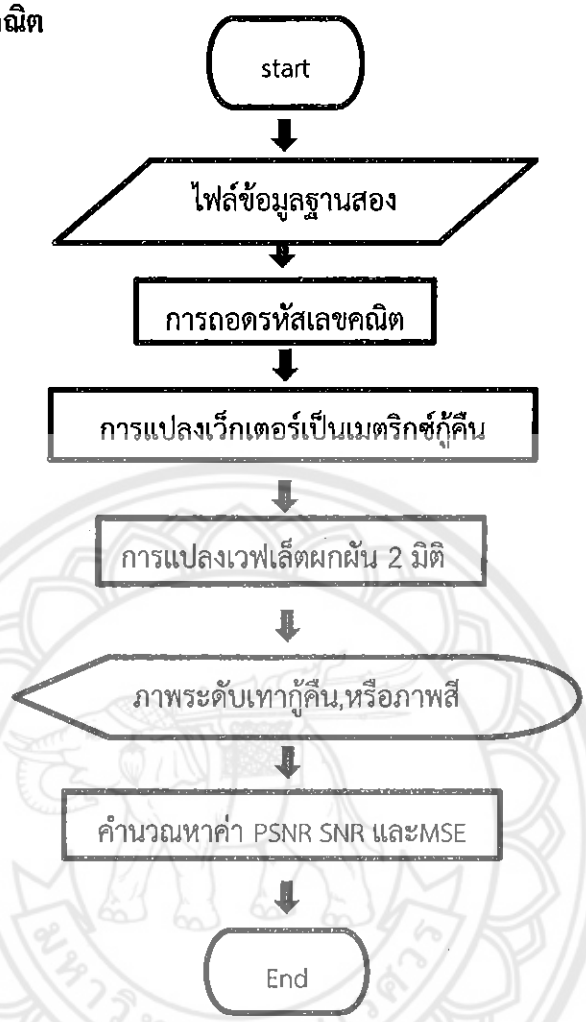
ขั้นตอนที่สี่: การเข้ารหัสเลขคณิต เริ่มต้นจากการแมปข้อมูลจากเวกเตอร์แนวนอนของสัมประสิทธิ์เวฟเล็ตระดับเทา หรือการแมปข้อมูลจากเวกเตอร์แนวนอนของสัมประสิทธิ์เวฟเล็ตสามสีหลักไปเป็นลำดับในการเข้ารหัสเลขคณิต หลังจากนั้นจะนำข้อมูลของสมาชิกที่ไม่ซ้ำกันไปเก็บอยู่ในรูปเมตริกซ์ตาราง แล้วทำการเข้ารหัสจนครบทุกลำดับการเข้ารหัสเลขคณิต ผลลัพธ์ที่ได้คือกระแสข้อมูลบิต โดยมีรายละเอียดอยู่ในหัวข้อ 2.7.1 ของบทที่ 2

ขั้นตอนที่ห้า: นำผลลัพธ์ที่ได้จากขั้นตอนที่สี่มาทำการบันทึกตารางกระแสข้อมูลบิตเป็นไฟล์ข้อมูลฐานสอง โดยมีรายละเอียดอยู่ในหัวข้อ 2.5 ของบทที่ 2

ขั้นตอนที่หก: ทำการคำนวณหาค่าอัตราส่วนการบีบอัดข้อมูล โดยมีรายละเอียดอยู่ในหัวข้อ 2.4.1 ของบทที่ 2

ซึ่งขั้นตอนในการดำเนินการบีบอัดภาพดังแสดงในรูปที่ 3.1

3.4.2 การคายการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต



รูปที่ 3.2 แผนภาพการคายการบีบอัดภาพดิจิตอลด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติโดยใช้ตัวกรองฮาร์ รวมทั้งการถอดรหัสเลขคณิต

ขั้นตอนในการดำเนินการคายการบีบอัดภาพมี 6 ขั้นตอน ดังนี้

ขั้นตอนที่หนึ่ง: นำไฟล์กระแสข้อมูลบิต และตารางกระแสข้อมูลบิตที่ได้จากการดำเนินการบีบอัดภาพทั้งภาพระดับเทา และภาพสี

ขั้นตอนที่สอง: จากนั้นนำไฟล์กระแสข้อมูลบิต และตารางกระแสข้อมูลบิตที่ได้จากการดำเนินการบีบอัดภาพทั้งภาพระดับเทา และภาพสี แล้วนำมาถอดรหัสเลขคณิต ทำให้ได้รับข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืนระดับเทา หรือข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืนสามสีหลัก คือ ข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืนสีแดง ข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืนสีเขียว และข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืนสีน้ำเงิน ซึ่งข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืนจะอยู่ในรูปเวกเตอร์ โดยมีรายละเอียดอยู่ในหัวข้อ 2.7.2 ของบทที่ 2

ขั้นตอนที่สาม: ต่อมานำผลลัพธ์ที่ได้จากขั้นตอนที่สอง โดยการแปลงเวกเตอร์ที่ได้รับเป็นเมตริกซ์สัมประสิทธิ์กู้คืนระดับเทา หรือเมตริกซ์สัมประสิทธิ์กู้คืนสามสีหลัก คือ เมตริกซ์สัมประสิทธิ์กู้คืนสีแดง เมตริกซ์สัมประสิทธิ์กู้คืนสีเขียว และเมตริกซ์สัมประสิทธิ์กู้คืนสีน้ำเงิน

ขั้นตอนที่สี่: นำเมตริกซ์สัมประสิทธิ์กู้คืนที่ได้จากขั้นตอนที่สาม กรณีภาพระดับเทาซึ่งอยู่ในรูปของเมตริกซ์สัมประสิทธิ์กู้คืนระดับเทา ในส่วนของภาพสีผลลัพธ์จะเป็นเมตริกซ์สัมประสิทธิ์กู้คืนสามสีหลักคือ เมตริกซ์สัมประสิทธิ์กู้คืนสีแดง เมตริกซ์สัมประสิทธิ์กู้คืนสีเขียว และเมตริกซ์สัมประสิทธิ์กู้คืนสีน้ำเงิน เพื่อทำการแปลงเวฟเล็ตเต็มหน่วยผกผันแบบ 2 มิติ 3 ระดับ โดยใช้ตัวกรองฮาร์ สุดท้ายจะได้ผลลัพธ์ภาพต้นฉบับระดับเทา รวมทั้งภาพต้นแบบสามสีหลัก คือ ภาพต้นฉบับสีแดง ภาพต้นฉบับสีเขียว และภาพต้นฉบับสีน้ำเงินที่กู้คืน โดยมีรายละเอียดอยู่ในหัวข้อ 2.6.2 ของบทที่ 2

ขั้นตอนที่ห้า: นำผลลัพธ์ที่ได้จากขั้นตอนที่สี่มาประกอบคืน เมื่อได้ภาพกู้คืนทั้งสามสีหลัก สุดท้ายจะประกอบภาพ 3 สีหลักเป็นภาพสีกู้คืน ส่วนกรณีภาพระดับเทาจะได้ภาพกู้คืนออกมาเลยไม่มีการประกอบภาพ

ขั้นตอนที่หก: ทำการคำนวณหาค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด(Peak Signal to Noise Ratio: PSNR) และอัตราส่วนสัญญาณต่อสัญญาณรบกวน(Signal to Noise Ratio: SNR) กับการหาค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) โดยมีรายละเอียดอยู่ในหัวข้อ 2.4.2, 2.4.3 และ 2.4.4 ของบทที่ 2

ซึ่งขั้นตอนในการดำเนินการคายบีบอัดภาพดังแสดงในรูปที่ 3.2 แผนภาพการคายบีบอัดภาพดิจิตอลด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต

3.4.3 การบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองเดบุชชี4 รวมทั้งการเข้ารหัสเลขคณิต

มีขั้นตอนในการดำเนินการคล้ายกับที่แสดงไว้ในหัวข้อที่ 3.4.1 การบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต แต่จะมีการเปลี่ยนตัวกรองที่ใช้เป็นเดบุชชี4 แทน ซึ่งจะมีขั้นตอนการดำเนินการที่เหลือทั้งหมดเหมือนดังแสดงไว้ในรูปที่ 3.1

3.4.4 การคายการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองเดบุชชี4 รวมทั้งการถอดรหัสเลขคณิต

มีขั้นตอนในการดำเนินการคล้ายกับหัวข้อที่ 3.4.2 การคายการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการถอดรหัสเลขคณิต แต่จะมีการเปลี่ยนชนิดตัวกรองเป็นเดบุชชี4 แทน ขั้นตอนการดำเนินการที่เหลือทั้งหมดเหมือนดังแสดงในรูปที่ 3.2

3.5 ทดสอบอัลกอริทึม

มีการทดสอบอัลกอริทึมการเข้ารหัสเลขคณิต และทดสอบอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต ผลการทดลองเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิต ภาพดิจิตอลก่อนการทำการบีบอัดภาพและหลังการบีบอัดภาพ ผลการทดลองจากการบีบอัดภาพดิจิตอล และสรุปผลการทดลอง ซึ่งแสดงตามลำดับในการทดสอบตามหัวข้อที่ 4.1 และ 4.2 ของบทที่ 4

3.6 วิเคราะห์ผลการทดลอง สรุปผลการทดลอง ปัญหา และข้อเสนอแนะ

จากการทดลอง และสรุปผลการทดลองของทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิต ดังแสดงในบทที่ 4 และแสดงในหัวข้อ 5.1, 5.2 และ 5.3 ของบทที่ 5

3.7 จัดทำรูปเล่มปริญญานิพนธ์

มีการแก้ไขปรับปรุงรูปเล่ม เรียบเรียงข้อมูลพร้อมทั้งการจัดทำรูปเล่มปริญญานิพนธ์

บทที่ 4

ผลการทดลองและการวิเคราะห์ผลการทดลอง

ในการทดสอบประสิทธิภาพการบีบอัดภาพดิจิตอล ได้ออกแบบการทดลองการบีบอัดภาพด้วยการแปลงเวฟเล็ตชนิดตัวกรองฮาร์ และชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิตนั้น จะแบ่งการทดลองเป็นการบีบอัดแบบบันทึกตารางการเข้ารหัสเลขคณิต กับการบีบอัดแบบไม่บันทึกตารางการเข้ารหัสเลขคณิต ซึ่งภาพที่นำมาทดสอบการบีบอัดภาพนั้นใช้ภาพทดสอบจำนวน 5 ภาพ คือ ภาพระดับเทามนุษย์ถ่ายรูป (Cameraman) ขนาด 256x256 พิกเซล ภาพระดับเทาบาร์บารา (Barbara) ขนาด 512x512 พิกเซล ภาพสีลีน่า (Lena) ขนาด 512x512 พิกเซล ภาพสีพริกไทย (Peppers) ขนาด 512x512 พิกเซล และภาพสีแมนดริล (Mandril) ขนาด 512x512 พิกเซล โดยมีการทดสอบอัลกอริทึม และผลการทดลองดังนี้

4.1 ทดสอบอัลกอริทึมการเข้ารหัสเลขคณิต

สมมุติว่าข้อมูล X เป็นเมตริกซ์ภาพดิจิตอลขนาด 4x4 ซึ่งมีค่าดังนี้

$$X = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

การเข้ารหัสเลขคณิต

เริ่มต้นจากการนำแปลงเมตริกซ์เป็นเวิกเตอร์แนวนอน ดังนี้ 1 1 1 1 3 2 1 1 1 2 1 1 1 1 2 1 ต่อมาจะทำการแมปข้อมูลจากเวิกเตอร์แนวนอนไปเป็นลำดับในการเข้ารหัสเลขคณิต

ตารางที่ 4.1 ตารางการแมปข้อมูลจากเวิกเตอร์แนวนอนไปเป็นลำดับในการเข้ารหัสเลขคณิต

symbol	sequence	count
1	1	12
3	2	1
2	3	3
Total count		16

โดยมีลำดับการเข้ารหัสดังนี้ 1 1 1 1 2 3 1 1 1 3 1 1 1 1 3 1

หลังจากที่แปลงเมตริกซ์ข้อมูล X เป็นเวิกเตอร์แนวนอนเสร็จแล้วจะนำข้อมูลของสมาชิก X ที่ไม่ซ้ำกันเก็บอยู่ในรูปเมตริกซ์ตารางดังนี้ $table = [1 \quad 3 \quad 2]$

หลังจากแมปข้อมูลเสร็จแล้วทำการหาค่าความยาวเวิร์ด(m)ได้จากสมการที่ 2.18

$$m = [\log_2 Total\ Count] + 2 = [log_2\ 16] + 2 = 4 + 2 = 6$$

กำหนดค่าเริ่มต้นในการเข้ารหัสเลขคณิต ซึ่งกำหนดได้จากความยาวเวิร์ด

$$l^{(0)} = 0 = (000000)_2\ , u^{(0)} = 63 = (111111)_2$$

เข้ารหัสตัวที่ 1 คือ การเข้ารหัสลำดับที่ 1

$$l^{(n)} = l^{(n-1)} + [\frac{(u^{n-1} - l^{n-1} + 1) * \text{Cum_Count}(n-1)}{\text{Total_Count}}]$$

$$u^{(n)} = l^{(n-1)} + [\frac{(u^{n-1} - l^{n-1} + 1) * \text{Cum_Count}(n)}{\text{Total_Count}}] - 1$$

$$l^{(1)} = 0 + \left[\frac{(63 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(1)} = 0 + \left[\frac{(63 - 0 + 1) * 12}{16}\right] - 1 = 47 = (101111)_2$$

เข้ารหัสตัวที่ 2 คือ การเข้ารหัสลำดับที่ 1

$$l^{(2)} = 0 + \left[\frac{(47 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(2)} = 0 + \left[\frac{(47 - 0 + 1) * 12}{16}\right] - 1 = 35 = (100011)_2$$

เข้ารหัสตัวที่ 3 คือ การเข้ารหัสลำดับที่ 1

$$l^{(3)} = 0 + \left[\frac{(35 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(3)} = 0 + \left[\frac{(35 - 0 + 1) * 12}{16}\right] - 1 = 26 = (011010)_2$$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(3)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(3)}$

$$l^{(3)} = (000000)_2 = 0$$

$$u^{(3)} = (110101)_2 = 53$$

เข้ารหัสตัวที่ 4 คือ การเข้ารหัสลำดับที่ 1

$$l^{(4)} = 0 + \left[\frac{(53 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(4)} = 0 + \left[\frac{(53 - 0 + 1) * 12}{16}\right] - 1 = 39 = (100111)_2$$

เข้ารหัสตัวที่ 5 คือ การเข้ารหัสลำดับที่ 2

$$l^{(5)} = 0 + \left[\frac{(39 - 0 + 1) * 12}{16}\right] = 30 = (011110)_2$$

$$u^{(5)} = 0 + \left[\frac{(39 - 0 + 1) * 13}{16}\right] - 1 = 39 = (011111)_2$$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ **LSB ของ** $l^{(5)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$

$$l^{(5)} = (111100)_2 = 60$$

$$u^{(5)} = (111111)_2 = 63$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ **LSB ของ** $l^{(5)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$

$$l^{(5)} = (111000)_2 = 56$$

$$u^{(5)} = (111111)_2 = 63$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ **LSB ของ** $l^{(5)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$

$$l^{(5)} = (100000)_2 = 32$$

$$u^{(5)} = (111111)_2 = 63$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(5)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$

$$l^{(5)} = (000000)_2 = 0$$

$$u^{(5)} = (111111)_2 = 63$$

เข้ารหัสตัวที่ 6 คือ การเข้ารหัสลำดับที่ 3

$$l^{(6)} = 0 + \left[\frac{(63 - 0 + 1) * 13}{16}\right] = 52 = (110100)_2$$

$$u^{(6)} = 0 + \left[\frac{(63 - 0 + 1) * 16}{16}\right] - 1 = 63 = (111111)_2$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(6)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(6)}$

$$l^{(6)} = (101000)_2 = 40$$

$$u^{(6)} = (111111)_2 = 63$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(6)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(6)}$

$$l^{(6)} = (010000)_2 = 16$$

$$u^{(6)} = (111111)_2 = 63$$

เข้ารหัสตัวที่ 7 คือ การเข้ารหัสลำดับที่ 1

$$l^{(7)} = 16 + \left[\frac{(63 - 16 + 1) * 0}{16}\right] = 16 = (010000)_2$$

$$u^{(7)} = 16 + \left[\frac{(63 - 16 + 1) * 12}{16}\right] - 1 = 51 = (110011)_2$$

เข้ารหัสตัวที่ 8 คือ การเข้ารหัสลำดับที่ 1

$$l^{(8)} = 16 + \left[\frac{(51 - 16 + 1) * 0}{16}\right] = 16 = (010000)_2$$

$$u^{(8)} = 16 + \left[\frac{(51 - 16 + 1) * 12}{16}\right] - 1 = 42 = (101010)_2$$

อยู่เงื่อนไข E3 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว นอกจากนั้นทำการกลับค่าบิตตัวที่สองของทั้ง $l^{(8)}, u^{(8)}$ แล้วเติมบิต 0 ที่ LSB ของ $l^{(8)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(8)}$

$$l^{(8)} = (000000)_2 = 0$$

$$u^{(8)} = (110101)_2 = 53$$

เข้ารหัสตัวที่ 9 คือ การเข้ารหัสลำดับที่ 1

$$l^{(9)} = 0 + \left[\frac{(53 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(9)} = 0 + \left[\frac{(53 - 0 + 1) * 12}{16}\right] - 1 = 39 = (100111)_2$$

เข้ารหัสตัวที่ 10 คือ การเข้ารหัสลำดับที่ 3

$$l^{(10)} = 0 + \left[\frac{(39 - 0 + 1) * 13}{16}\right] = 32 = (100000)_2$$

$$u^{(10)} = 0 + \left[\frac{(39 - 0 + 1) * 16}{16}\right] - 1 = 39 = (100111)_2$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ **LSB ของ** $l^{(10)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(10)}$

$$l^{(10)} = (000000)_2 = 0$$

$$u^{(10)} = (001111)_2 = 15$$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ **LSB ของ** $l^{(10)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(10)}$

$$l^{(10)} = (000000)_2 = 0$$

$$u^{(10)} = (011111)_2 = 31$$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ **LSB ของ** $l^{(10)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(10)}$

$$l^{(10)} = (000000)_2 = 0$$

$$u^{(10)} = (111111)_2 = 63$$

เข้ารหัสตัวที่ 11 คือ การเข้ารหัสลำดับที่ 1

$$l^{(11)} = 0 + \left[\frac{(63 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(11)} = 0 + \left[\frac{(63 - 0 + 1) * 12}{16}\right] - 1 = 47 = (101111)_2$$

เข้ารหัสตัวที่ 12 คือ การเข้ารหัสลำดับที่ 1

$$l^{(12)} = 0 + \left\lfloor \frac{(47 - 0 + 1) * 0}{16} \right\rfloor = 0 = (000000)_2$$

$$u^{(12)} = 0 + \left\lfloor \frac{(47 - 0 + 1) * 12}{16} \right\rfloor - 1 = 35 = (100111)_2$$

เข้ารหัสตัวที่ 13 คือ การเข้ารหัสลำดับที่ 1

$$l^{(13)} = 0 + \left\lfloor \frac{(35 - 0 + 1) * 0}{16} \right\rfloor = 0 = (000000)_2$$

$$u^{(13)} = 0 + \left\lfloor \frac{(35 - 0 + 1) * 12}{16} \right\rfloor - 1 = 26 = (011010)_2$$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(13)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(13)}$

$$l^{(13)} = (000000)_2 = 0$$

$$u^{(13)} = (110101)_2 = 53$$

เข้ารหัสตัวที่ 14 คือ การเข้ารหัสลำดับที่ 1

$$l^{(14)} = 0 + \left\lfloor \frac{(53 - 0 + 1) * 0}{16} \right\rfloor = 0 = (000000)_2$$

$$u^{(14)} = 0 + \left\lfloor \frac{(53 - 0 + 1) * 12}{16} \right\rfloor - 1 = 39 = (100111)_2$$

เข้ารหัสตัวที่ 15 คือ การเข้ารหัสลำดับที่ 3

$$l^{(15)} = 0 + \left\lfloor \frac{(39 - 0 + 1) * 13}{16} \right\rfloor = 32 = (100000)_2$$

$$u^{(15)} = 0 + \left\lfloor \frac{(39 - 0 + 1) * 16}{16} \right\rfloor - 1 = 39 = (100111)_2$$

อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(15)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(15)}$

$$l^{(15)} = (000000)_2 = 0$$

$$u^{(15)} = (001111)_2 = 15$$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(15)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(15)}$

$l^{(15)} = (000000)_2 = 0$

$u^{(15)} = (011111)_2 = 31$

อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(15)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(15)}$

$l^{(15)} = (000000)_2 = 0$

$u^{(15)} = (111111)_2 = 63$

เข้ารหัสตัวที่ 16 คือ การเข้ารหัสลำดับที่ 1

$$l^{(16)} = 0 + \left\lfloor \frac{(63 - 0 + 1) * 0}{16} \right\rfloor = 0 = (000000)_2$$

$$u^{(16)} = 0 + \left\lfloor \frac{(63 - 0 + 1) * 12}{16} \right\rfloor - 1 = 47 = (101111)_2$$

ลำดับส่งสุดท้ายคือ 00111111100001000000000.

<u>การถอดรหัสเลขคณิต</u>

ถอดรหัสลำดับ: 00111111100001000000000

ขั้นแรกในการถอดรหัสเลขคณิตเราจะทำการหาค่า tag โดยเลือกจากค่า m บิตแรก ดังนั้น

ค่า tag คือ 001111

$tag = (001111)_2 = 15$

หลังจากได้ค่า tag เราจะทำการกำหนดค่าเริ่มต้นในการถอดรหัสเลขคณิต คือ

$l = (00000000)_2 = 0$

$u = (111111)_2 = 63$

ขั้นตอนต่อมาเราจะคำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิต โดยใช้สมการที่ 2.19

$$\left\lfloor \frac{(t - l + 1) * Total_{Count} - 1}{u - l + 1} \right\rfloor = \left\lfloor \frac{(15 - 0 + 1) * 16 - 1}{63 - 0 + 1} \right\rfloor = 3$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count $= \begin{bmatrix} 0 \\ 12 \\ 13 \\ 16 \end{bmatrix}$

ดังนั้น $0 < 3 < 12$,

หลังจากกทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 1 คือ 1 และมีการอัพเดทค่า $l^{(1)}$ และ $u^{(1)}$

$$l^{(n)} = l^{(n-1)} + \left[\frac{(u^{n-1} - l^{n-1} + 1) * \text{Cum_Count}(n-1)}{\text{Total_Count}}\right]$$

$$u^{(n)} = l^{(n-1)} + \left[\frac{(u^{n-1} - l^{n-1} + 1) * \text{Cum_Count}(n)}{\text{Total_Count}}\right] - 1$$

$$l^{(1)} = 0 + \left[\frac{(63 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(1)} = 0 + \left[\frac{(63 - 0 + 1) * 12}{16}\right] - 1 = 47 = (101111)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 2

$$\left[\frac{(t - l + 1) * \text{Total}_{\text{Count}} - 1}{u - l + 1}\right] = \left[\frac{(15 - 0 + 1) * 16 - 1}{47 - 0 + 1}\right] = 5$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0 < 5 < 12$,

หลังจากกทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 2 คือ 1 และมีการอัพเดทค่า $l^{(2)}$ และ $u^{(2)}$

$$l^{(2)} = 0 + \left[\frac{(47 - 0 + 1) * 0}{16}\right] = 0 = (000000)_2$$

$$u^{(2)} = 0 + \left[\frac{(47 - 0 + 1) * 12}{16}\right] - 1 = 35 = (100011)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 3

$$\left[\frac{(t - l + 1) * \text{Total}_{\text{Count}} - 1}{u - l + 1}\right] = \left[\frac{(15 - 0 + 1) * 16 - 1}{35 - 0 + 1}\right] = 7$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0 < 7 < 12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 3 คือ 1 และมีการอัพเดทค่า $l^{(3)}$ และ $u^{(3)}$

$$l^{(3)} = 0 + \left\lfloor \frac{(35 - 0 + 1) * 0}{16} \right\rfloor = 0 = (000000)_2$$

$$u^{(3)} = 0 + \left\lfloor \frac{(35 - 0 + 1) * 12}{16} \right\rfloor - 1 = 26 = (011010)_2$$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(3)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(3)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$$l^{(3)} = (000000)_2 = 0$$

$$u^{(3)} = (110101)_2 = 53$$

$$tag = (011111)_2 = 31$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 4

$$\left\lfloor \frac{(t - l + 1) * Total_{Count} - 1}{u - l + 1} \right\rfloor = \left\lfloor \frac{(31 - 0 + 1) * 16 - 1}{53 - 0 + 1} \right\rfloor = 9$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count $= \begin{bmatrix} 0 \\ 12 \\ 13 \\ 16 \end{bmatrix}$

ดังนั้น $0 < 9 < 12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 4 คือ 1 และมีการอัพเดทค่า $l^{(4)}$ และ $u^{(4)}$

$$l^{(4)} = 0 + \left\lfloor \frac{(53 - 0 + 1) * 0}{16} \right\rfloor = 0 = (000000)_2$$

$$u^{(4)} = 0 + \left\lfloor \frac{(53 - 0 + 1) * 12}{16} \right\rfloor - 1 = 39 = (100111)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 5

$$\left\lfloor \frac{(t - l + 1) * Total_{Count} - 1}{u - l + 1} \right\rfloor = \left\lfloor \frac{(31 - 0 + 1) * 16 - 1}{39 - 0 + 1} \right\rfloor = 12$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count $= \begin{bmatrix} 0 \\ 12 \\ 13 \\ 16 \end{bmatrix}$

ดังนั้น 12 <= 12 < 13,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 5 คือ 2 และมีการอัพเดทค่า $l^{(5)}$ และ $u^{(5)}$

$$l^{(5)} = 0 + \left\lfloor \frac{(39 - 0 + 1) * 12}{16} \right\rfloor = 30 = (011110)_2$$

$$u^{(5)} = 0 + \left\lfloor \frac{(39 - 0 + 1) * 13}{16} \right\rfloor - 1 = 31 = (011111)_2$$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(5)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(5)} = (111100)_2 = 60$

$u^{(5)} = (111111)_2 = 63$

$tag = (111111)_2 = 63$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(5)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(5)} = (111000)_2 = 56$

$u^{(5)} = (111111)_2 = 63$

$tag = (111111)_2 = 63$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(5)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(5)} = (110000)_2 = 48$

$u^{(5)} = (111111)_2 = 63$

$tag = (111110)_2 = 62$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(5)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(5)} = (100000)_2 = 32$

$u^{(5)} = (111111)_2 = 63$

$tag = (111100)_2 = 60$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(5)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(5)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$$l^{(5)} = (000000)_2 = 0$$

$$u^{(5)} = (111111)_2 = 63$$

$$\text{tag} = (111000)_2 = 56$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 6

$$\left[\frac{(t-l+1)*\text{Total}_{\text{Count}}-1}{u-l+1}\right] = \left[\frac{(56-0+1)*16-1}{63-0+1}\right] = 14$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count $= \begin{bmatrix} 0 \\ 12 \\ 13 \\ 16 \end{bmatrix}$

ดังนั้น $13 < 14 < 16$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 6 คือ 3 และมีการอัพเดทค่า $l^{(6)}$ และ $u^{(6)}$

$$l^{(6)} = 0 + \left[\frac{(63-0+1)*13}{16}\right] = 52 = (110100)_2$$

$$u^{(6)} = 0 + \left[\frac{(63-0+1)*16}{16}\right] - 1 = 63 = (111111)_2$$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(6)}$ แล้วเติมบิต 1 ที่ LSB ของ $u^{(6)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$$l^{(6)} = (101000)_2 = 40$$

$$u^{(6)} = (111111)_2 = 63$$

$$\text{tag} = (110000)_2 = 48$$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(6)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(6)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$$l^{(6)} = (010000)_2 = 16$$

$$u^{(6)} = (111111)_2 = 63$$

$$\text{tag} = (100001)_2 = 33$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 7

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right] = \left[\frac{(33-16+1)*16-1}{63-16+1}\right] = 5$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0 < 5 < 12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 7 คือ 1 และมีการอัพเดทค่า $l^{(7)}$ และ $u^{(7)}$

$$l^{(7)} = 16 + \left[\frac{(63-16+1)*0}{16}\right] = 16 = (010000)_2$$

$$u^{(7)} = 16 + \left[\frac{(63-16+1)*12}{16}\right] - 1 = 51 = (110011)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 8

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right] = \left[\frac{(33-16+1)*16-1}{51-16+1}\right] = 7$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0 < 7 < 12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 8 คือ 1 และมีการอัพเดทค่า $l^{(8)}$ และ $u^{(8)}$

$$l^{(8)} = 16 + \left[\frac{(51-16+1)*0}{16}\right] = 16 = (010000)_2$$

$$u^{(8)} = 16 + \left[\frac{(51-16+1)*12}{16}\right] - 1 = 42 = (101010)_2$$

ทั้งคู่อยู่เงื่อนไข E3 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว นอกจากนั้นทำการกลับค่าบิตตัวที่สองของทั้ง $l^{(8)}$, $u^{(8)}$ แล้วเติมบิต 0 ที่ LSB ของ $l^{(8)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(8)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(8)} = (000000)_2 = 0$

$u^{(8)} = (110101)_2 = 53$

$tag = (100010)_2 = 34$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 9

$$\left\lceil \frac{(t - l + 1) * Total_{Count} - 1}{u - l + 1} \right\rceil = \left\lceil \frac{(34 - 0 + 1) * 16 - 1}{53 - 0 + 1} \right\rceil = 10$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix} 0 \\ 12 \\ 13 \\ 16 \end{bmatrix}$

ดังนั้น 0 < 10 < 12,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 9 คือ 1 และมีการอัพเดทค่า $l^{(9)}$ และ $u^{(9)}$

$$l^{(9)} = 0 + \left\lceil \frac{(53 - 0 + 1) * 0}{16} \right\rceil = 0 = (000000)_2$$

$$u^{(9)} = 0 + \left\lceil \frac{(53 - 0 + 1) * 12}{16} \right\rceil - 1 = 39 = (100111)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 10

$$\left\lceil \frac{(t - l + 1) * Total_{Count} - 1}{u - l + 1} \right\rceil = \left\lceil \frac{(34 - 0 + 1) * 16 - 1}{39 - 0 + 1} \right\rceil = 13$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix} 0 \\ 12 \\ 13 \\ 16 \end{bmatrix}$

ดังนั้น 13 <= 13 < 16,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 10 คือ 3 และมีการอัพเดทค่า $l^{(10)}$ และ $u^{(10)}$

$$l^{(10)} = 0 + \left\lceil \frac{(39 - 0 + 1) * 13}{16} \right\rceil = 32 = (100000)_2$$

$$u^{(10)} = 0 + \left\lceil \frac{(39 - 0 + 1) * 16}{16} \right\rceil - 1 = 39 = (100111)_2$$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(10)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(10)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(10)} = (000000)_2 = 0$

$u^{(10)} = (001111)_2 = 15$

$tag = (000100)_2 = 4$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(10)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(10)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(10)} = (000000)_2 = 0$

$u^{(10)} = (011111)_2 = 31$

$tag = (001000)_2 = 8$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(10)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(10)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(10)} = (000000)_2 = 0$

$u^{(10)} = (111111)_2 = 63$

$tag = (010000)_2 = 16$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 11

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right] = \left[\frac{(16-0+1)*16-1}{63-0+1}\right] = 4$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0 < 4 < 12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 11 คือ 1 และมีการอัพเดทค่า $l^{(11)}$ และ $u^{(11)}$

$$l^{(11)} = 0 + \left[\frac{(63-0+1)*0}{16}\right] = 0 = (000000)_2$$

$$u^{(11)} = 0 + \left[\frac{(63-0+1)*12}{16}\right] - 1 = 47 = (101111)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 12

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right]=\left[\frac{(16-0+1)*16-1}{47-0+1}\right]=5$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0<5<12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 12 คือ 1 และมีการอัพเดทค่า $l^{(12)}$ และ $u^{(12)}$

$$l^{(12)}=0+\left[\frac{(47-0+1)*0}{16}\right]=0=(000000)_2$$

$$u^{(12)}=0+\left[\frac{(47-0+1)*12}{16}\right]-1=35=(100011)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 13

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right]=\left[\frac{(16-0+1)*16-1}{35-0+1}\right]=7$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0<7<12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 13 คือ 1 และมีการอัพเดทค่า $l^{(13)}$ และ $u^{(13)}$

$$l^{(13)}=0+\left[\frac{(35-0+1)*0}{16}\right]=0=(000000)_2$$

$$u^{(13)}=0+\left[\frac{(35-0+1)*12}{16}\right]-1=26=(011010)_2$$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(13)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(13)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(13)} = (000000)_2 = 0$

$u^{(13)} = (110101)_2 = 53$

$tag = (100000)_2 = 32$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 14

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right] = \left[\frac{(32-0+1)*16-1}{53-0+1}\right] = 9$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น 0 < 9 < 12,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 14 คือ 1 และ**มีการอัพเดทค่า** $l^{(14)}$ และ $u^{(14)}$

$$l^{(14)} = 0 + \left[\frac{(53-0+1)*0}{16}\right] = 0 = (000000)_2$$

$$u^{(14)} = 0 + \left[\frac{(53-0+1)*12}{16}\right] - 1 = 39 = (100111)_2$$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 15

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right] = \left[\frac{(32-0+1)*16-1}{39-0+1}\right] = 13$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น 13 <= 13 < 16,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 15 คือ 3 และมีการอัพเดทค่า $l^{(15)}$ และ $u^{(15)}$

$$l^{(15)} = 0 + \left[\frac{(39-0+1)*13}{16}\right] = 32 = (100000)_2$$

$$u^{(15)} = 0 + \left[\frac{(39-0+1)*16}{16}\right] - 1 = 39 = (100111)_2$$

ทั้งคู่อยู่เงื่อนไข E2 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(15)}$**แล้วเติมบิต 1 ที่ LSB ของ** $u^{(15)}$ **และทำการอ่านค่าในบิตถัดไปจากกระแสบิต**

$l^{(15)} = (000000)_2 = 0$

$u^{(15)} = (001111)_2 = 15$

$tag = (000000)_2 = 0$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(15)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(15)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(15)} = (000000)_2 = 0$

$u^{(15)} = (011111)_2 = 31$

$tag = (000000)_2 = 0$

ทั้งคู่อยู่เงื่อนไข E1 อัพเดทค่า โดยทำการเลื่อนบิตทางซ้ายมือไป 1 ตัว แล้วเติมบิต 0 ที่ LSB ของ $l^{(15)}$แล้วเติมบิต 1 ที่ LSB ของ $u^{(15)}$ และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$l^{(15)} = (000000)_2 = 0$

$u^{(15)} = (111111)_2 = 63$

$tag = (000000)_2 = 0$

ทำการถอดรหัสสัญลักษณ์ตัวต่อไป

คำนวณหาค่าเริ่มต้นในการถอดรหัสเลขคณิตสัญลักษณ์ตัวที่ 16

$$\left[\frac{(t-l+1)*Total_{Count}-1}{u-l+1}\right] = \left[\frac{(0-0+1)*16-1}{63-0+1}\right] = 0$$

จากนั้นจะค่าเริ่มต้นในการถอดรหัสเลขคณิตมาทำการเปรียบเทียบค่า cum_count = $\begin{bmatrix}0\\12\\13\\16\end{bmatrix}$

ดังนั้น $0 <= 0 < 12$,

หลังจากทำการเปรียบเทียบเสร็จแล้วเราจะทำถอดรหัสสัญลักษณ์ตัวที่ 16 คือ 1 และมีการอัพเดทค่า $l^{(16)}$ และ $u^{(16)}$

$$l^{(16)} = 0 + \left[\frac{(63-0+1)*0}{16}\right] = 0 = (000000)_2$$

$$u^{(16)} = 0 + \left[\frac{(63-0+1)*12}{16}\right] - 1 = 47 = (101111)_2$$

และทำการอ่านค่าในบิตถัดไปจากกระแสบิต

$tag = (000000)_2 = 0$

4.2 ทดสอบอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ รวมทั้งการเข้ารหัสเลขคณิต

สมมุติว่าข้อมูล X เป็นเมตริกซ์ขนาด 4x4 ซึ่งมีค่าดังนี้

$$X = \begin{bmatrix} 1 & 3 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

% การแปลงเวฟเล็ตเต็มหน่วยระดับที่ 1 ชนิดตัวกรองฮาร์

เริ่มต้นด้วยการนำข้อมูล X มาทำการแปลงเวฟเล็ตเต็มหน่วยระดับที่ 1 ชนิดตัวกรองฮาร์ ผลลัพธ์ที่ได้คือค่าสัมประสิทธิ์เวฟเล็ตทั้งหมด 4 ค่า แสดงอยู่ในรูปเมตริกซ์ ดังนี้

[cA1,cH1,cV1,cD1] = dwt2(X,'haar')

$$cA1 = \begin{bmatrix} 10.50 & 13.00 & 15.50 & 19.00 \\ 11.50 & 10.00 & 15.00 & 17.00 \\ 20.00 & 22.00 & 25.00 & 30.00 \\ 20.00 & 22.00 & 25.00 & 30.00 \end{bmatrix}$$

$$cH1 = \begin{bmatrix} 9.50 & 9.00 & 9.50 & 11.00 \\ 8.50 & 12.00 & 9.00 & 9.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$cV1 = \begin{bmatrix} -1.50 & -1.00 & -0.50 & -1.00 \\ -0.50 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$cD1 = \begin{bmatrix} 1.50 & 1.00 & -0.50 & 1.00 \\ 0.50 & -3.00 & 0 & 3.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

นำค่าสัมประสิทธิ์เวฟเล็ตทั้งหมดที่ได้จากการแปลงเวฟเล็ตเต็มหน่วยชนิดตัวกรองฮาร์ ทั้ง 2 ระดับการแปลงมาเรียงเป็นเวกเตอร์ในแนวดิ่ง

```
% lv1
cA1i=cA1(:);
cH1i=cH1(:);
cV1i=cV1(:);
cD1i=cD1(:);
```

ผลลัพธ์ที่ได้จากการแปลงสัมประสิทธิ์เวฟเล็ตเป็นเวกเตอร์แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$cA1i = \begin{bmatrix} 10.50 \\ 11.50 \\ 20.00 \\ 20.00 \\ 13.00 \\ 10.00 \\ 22.00 \\ 22.00 \\ 15.50 \\ 15.00 \\ 25.00 \\ 25.00 \\ 19.00 \\ 17.00 \\ 30.00 \\ 30.00 \end{bmatrix} \quad cH1i = \begin{bmatrix} 9.50 \\ 8.50 \\ 0 \\ 0 \\ 9.00 \\ 12.00 \\ 0 \\ 0 \\ 9.50 \\ 9.00 \\ 0 \\ 0 \\ 11.00 \\ 9.00 \\ 0 \\ 0 \end{bmatrix} \quad cV1i = \begin{bmatrix} -1.50 \\ -0.50 \\ 0 \\ 0 \\ -1.00 \\ 1.00 \\ 0 \\ 0 \\ -0.50 \\ 0 \\ -1.00 \\ -1.00 \\ -1.00 \\ 1.00 \\ 0 \\ 0 \end{bmatrix} \quad cD1i = \begin{bmatrix} 1.50 \\ 0.50 \\ 0 \\ 0 \\ 1.00 \\ -3.00 \\ 0 \\ 0 \\ -0.50 \\ 0 \\ 0 \\ 0 \\ 1.00 \\ 3.00 \\ 0 \\ 0 \end{bmatrix}$$

เนื่องจากเราไม่สามารถนำเอาค่าเวกเตอร์สัมประสิทธิ์เวฟเล็ตไปทำการเข้ารหัสเลขคณิตกับฟังก์ชั่น arithenco ได้โดยตรง เลยทำให้ต้องมีการสร้างฟังก์ชั่นขึ้นมาเพื่อแปลงข้อมูลเหล่านั้นให้เป็นเวกเตอร์ลำดับการเข้ารหัสเลขคณิต(Xn) และcount จึงจะสามารถทำการเข้ารหัสเลขคณิตได้ ซึ่งฟังก์ชั่นที่ใช้ในวิทยานิพนธ์นี้ชื่อ new_arithme ช่วยในการแปลงข้อมูลก่อนที่จะนำไปเข้าฟังก์ชั่น arithenco จึงสามารถทำการเข้ารหัสเลขคณิตได้

```
cA1o1=new_arithme(cA1i,4,4,Na{1})
```

หลังจากที่แปลงเมตริกซ์สัมประสิทธิ์เวฟเล็ต cA1 เป็นเวกเตอร์สัมประสิทธิ์เวฟเล็ตแนวนอน cA1i เสร็จแล้วจะนำข้อมูลของสมาชิก cA1 ที่ไม่ซ้ำกันเก็บอยู่ในรูปเมตริกซ์ตารางดังนี้

$tablecA1 = [10.50 \quad 11.50 \quad 20.00 \quad 13.00 \quad 10.00 \quad 22.00 \quad 15.50 \quad 15.00 \quad 25.00 \quad 19.0 \quad 17.00 \quad 30.00]$

ต่อไปจะทำการจะแทนค่าเมตริกซ์ของ XncA1 เป็นลำดับในการเข้ารหัสเลขคณิต ซึ่งแสดงในรูป

เมตริกซ์ได้ดังนี้ $XncA1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \\ 3 & 6 & 9 & 12 \end{bmatrix}$

นอกจากนี้ยังมีการนับจำนวนของสมาชิก XncA1 ซึ่งเก็บอยู่ในตัวแปร countcA1 สามารถแสดงในรูปเมตริกซ์ดังนี้

$countcA1 = \begin{bmatrix} 1 & 1 & 2 & 1 & 1 & 2 & 1 & 1 & 2 & 1 & 1 & 2 \end{bmatrix}$

ขั้นตอนต่อมาเมื่อได้ตัวแปร XncA1 และ countcA1 แล้วนำไปทำการเข้ารหัสเลขคณิต ซึ่งมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการเข้ารหัสเลขคณิตในหัวข้อที่ 4.1 โดยผลลัพธ์ที่ได้จากการเข้ารหัสเลขคณิตคือ codecA1 แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$codecA1 =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

การถอดรหัสเลขคณิตมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการถอดรหัสเลขคณิตในหัวข้อที่ 4.1 โดย XnocA1 คือผลลัพธ์ที่ได้จากการถอดรหัสเลขคณิต ซึ่งเป็นลำดับจากการถอดรหัสเลขคณิตมีผลลัพธ์แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$XnocA1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \\ 3 & 6 & 9 & 12 \end{bmatrix}$

ต่อไปเป็นการแปลงกลับจะได้เมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืน cA1o1 ดังนี้

$cA1o1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \\ 3 & 6 & 9 & 12 \end{bmatrix}$

ต่อไปเมตริกซ์สัมประสิทธิ์เวฟเล็ต cH1o1 ดังนี้

cH1o1=new_arithme(cH1i,4,4,Na{2})

หลังจากที่แปลงเมตริกซ์สัมประสิทธิ์เวฟเล็ต cH1 เป็นเวิ็กเตอร์สัมประสิทธิ์เวฟเล็ตแนวนอน cH1i เสร็จแล้วจะนำข้อมูลของสมาชิก cH1 ที่ไม่ซ้ำกันเก็บอยู่ในรูปเมตริกซ์ตารางดังนี้

$tablecH1 = [9.50 \quad 8.50 \quad 0 \quad 9.00 \quad 12.00 \quad 11.00 \quad 9.00]$

ต่อไปจะทำการจะแทนค่าเมตริกซ์ของ XncH1 เป็นลำดับในการเข้ารหัสเลขคณิต ซึ่งแสดงในรูป

เมตริกซ์ได้ดังนี้ $XncH1 = \begin{bmatrix} 1 & 4 & 1 & 6 \\ 2 & 5 & 4 & 7 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$

นอกจากนี้ยังมีการนับจำนวนของสมาชิก XncH1 ซึ่งเก็บอยู่ในตัวแปร countcH1 สามารถแสดงในรูปเมตริกซ์ดังนี้

$countcH1 = [2 \quad 1 \quad 8 \quad 2 \quad 1 \quad 1 \quad 1]$

ขั้นตอนต่อมาเมื่อได้ตัวแปร XncH1 และ countcH1 แล้วนำไปทำการเข้ารหัสเลขคณิตซึ่งมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการเข้ารหัสเลขคณิตในหัวข้อที่ 4.1 โดยผลลัพธ์ที่ได้จากการเข้ารหัสเลขคณิตคือ codecH1 แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$codecH1 =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

การถอดรหัสเลขคณิตมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการถอดรหัสเลขคณิตในหัวข้อที่ 4.1 โดย XnocA1 คือผลลัพธ์ที่ได้จากการถอดรหัสเลขคณิต ซึ่งเป็นลำดับจากการถอดรหัสเลขคณิตมีผลลัพธ์แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$XnocH1 = \begin{bmatrix} 1 & 4 & 1 & 6 \\ 2 & 5 & 4 & 7 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

ต่อไปเป็นการแปลงกลับจะได้เมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืน cH1o1 ดังนี้

$$cH1o1 = \begin{bmatrix} 9.50 & 9.00 & 9.50 & 11.00 \\ 8.50 & 12.00 & 9.00 & 9.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

ต่อไปเมตริกซ์สัมประสิทธิ์เวฟเล็ต cV1o1 ดังนี้

cV1o1=new_arithme(cV1i,4,4,Na{3})

หลังจากที่แปลงเมตริกซ์สัมประสิทธิ์เวฟเล็ต cV1 เป็นเวกเตอร์สัมประสิทธิ์เวฟเล็ตแนวนอน cV1i เสร็จแล้วจะนำข้อมูลของสมาชิก cV1 ที่ไม่ซ้ำกันเก็บอยู่ในรูปเมตริกซ์ตาราง ดังนี้

$tablecV1 = [-1.50 \quad -0.50 \quad 0 \quad -1.00 \quad 1.00 \quad -0.50 \quad -1.00 \quad -1.00 \quad 1.00]$

ต่อไปจะทำการจะแทนค่าเมตริกซ์ของ XncV1 เป็นลำดับในการเข้ารหัสเลขคณิต ซึ่งแสดงในรูปเมตริกซ์ได้ดังนี้

$$XncV1 = \begin{bmatrix} 1 & 4 & 6 & 8 \\ 2 & 5 & 3 & 9 \\ 3 & 3 & 7 & 3 \\ 3 & 3 & 7 & 3 \end{bmatrix}$$

นอกจากนี้ยังมีการนับจำนวนของสมาชิก XncV1 ซึ่งเก็บอยู่ในตัวแปร countcV1 สามารถแสดงในรูปเมตริกซ์ดังนี้

$$countcV1 = \begin{bmatrix} 1 & 1 & 7 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

ขั้นตอนต่อมาเมื่อได้ตัวแปร XncV1 และ countcV1 แล้วนำไปทำการเข้ารหัสเลขคณิต ซึ่งมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการเข้ารหัสเลขคณิตในหัวข้อที่ 4.1 โดยผลลัพธ์ที่ได้จากการเข้ารหัสเลขคณิตคือ codecV1 แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$codecV1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

การถอดรหัสเลขคณิตมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการถอดรหัสเลขคณิตในหัวข้อที่ 4.1 โดย XnocV1 คือผลลัพธ์ที่ได้จากการถอดรหัสเลขคณิต ซึ่งเป็นลำดับจากการถอดรหัสเลขคณิตมีผลลัพธ์แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$XnocV1 = \begin{bmatrix} 1 & 4 & 6 & 8 \\ 2 & 5 & 3 & 9 \\ 3 & 3 & 7 & 3 \\ 3 & 3 & 7 & 3 \end{bmatrix}$$

ต่อไปเป็นการแปลงกลับจะได้เมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืน cH1o1 ดังนี้

$$cV1o1 = \begin{bmatrix} -1.50 & -1.00 & -0.50 & -1.00 \\ -0.50 & 1.00 & 0 & 1.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

ต่อไปเมตริกซ์สัมประสิทธิ์เวฟเล็ต cD1o1 ดังนี้

```
cD1o1=new_arithme(cD1i,4,4,Na{4})
```

หลังจากที่แปลงเมตริกซ์สัมประสิทธิ์เวฟเล็ต cD1 เป็นเว็กเตอร์สัมประสิทธิ์เวฟเล็ตแนวนอน cD1i เสร็จแล้วจะนำข้อมูลของสมาชิก cD1 ที่ไม่ซ้ำกันเก็บอยู่ในรูปเมตริกซ์ตาราง ดังนี้

$$tablecD1 = [1.50 \quad 0.50 \quad 0 \quad 1.00 \quad -3.00 \quad -0.50 \quad 1.00 \quad 3.00]$$

ต่อไปจะทำการจะแทนค่าเมตริกซ์ของ XncD1 เป็นลำดับในการเข้ารหัสเลขคณิต ซึ่งแสดงในรูปเมตริกซ์ได้ดังนี้

$$XncD1 = \begin{bmatrix} 1 & 4 & 6 & 7 \\ 2 & 5 & 3 & 8 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

นอกจากนี้ยังมีการนับจำนวนของสมาชิก XncD1 ซึ่งเก็บอยู่ในตัวแปร countcD1 สามารถแสดงในรูปเมตริกซ์ดังนี้

$$countcD1 = [1 \quad 1 \quad 9 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$$

การถอดรหัสเลขคณิตมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการถอดรหัสเลขคณิตในหัวข้อที่ 4.1 โดย XnocV1 คือผลลัพธ์ที่ได้จากการถอดรหัสเลขคณิต ซึ่งเป็นลำดับจากการถอดรหัสเลขคณิต มีผลลัพธ์แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$XncD1 = \begin{bmatrix} 1 & 4 & 6 & 7 \\ 2 & 5 & 3 & 8 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

ขั้นตอนต่อมาเมื่อได้ตัวแปร XncD1 และ countcD1 แล้วนำไปทำการเข้ารหัสเขคณิตซึ่งมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการเข้ารหัสเลขคณิตในหัวข้อที่ 4.1 โดยผลลัพธ์ที่ได้จากการเข้ารหัสเลขคณิตคือ codeD1 แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$codecD1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

การถอดรหัสเลขคณิตมีวิธีคำนวณเหมือนกับตัวอย่างการคำนวณการถอดรหัสเลขคณิตในหัวข้อที่ 4.1 โดย XnocV1 คือผลลัพธ์ที่ได้จากการถอดรหัสเลขคณิต ซึ่งเป็นลำดับจากการถอดรหัสเลขคณิต มีผลลัพธ์แสดงอยู่ในรูปเมตริกซ์ ดังนี้

$$XnocD1 = \begin{bmatrix} 1 & 4 & 6 & 7 \\ 2 & 5 & 3 & 8 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{bmatrix}$$

ต่อไปเป็นการแปลงกลับจะได้เมตริกซ์สัมประสิทธิ์เวฟเล็ตกู้คืน cD1o1 ดังนี้

$$cD1o1 = \begin{bmatrix} 1.50 & 1.00 & -0.50 & 1.00 \\ 0.50 & -3.00 & 0 & 3.00 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

% การแปลงเวฟเล็ตเต็มหน่วยผกผันครั้งที่ 1

นำค่าสัมประสิทธิ์เวฟเล็ตกู้คืนจากการแปลงเวฟเล็ตเต็มหน่วยชนิดตัวกรองฮาร์ ครั้งที่ 1 มาแปลงกลับ โดยแสดงผลลัพธ์ในรูปเมตริกซ์

A1 = idwt2(cA1o1,cH1o1,cV1o1,cD1o1,'haar')

$$A1 = \begin{bmatrix} 10 & 10 & 11 & 11 & 12 & 13 & 15 & 15 \\ -1 & 2 & 1 & 3 & 3 & 3 & 3 & 5 \\ 10 & 10 & 10 & 12 & 12 & 12 & 15 & 11 \\ 1 & 2 & 1 & -3 & 3 & 3 & 3 & 5 \\ 10 & 10 & 11 & 11 & 12 & 13 & 15 & 15 \\ 10 & 10 & 11 & 11 & 12 & 13 & 15 & 15 \\ 10 & 10 & 11 & 11 & 12 & 13 & 15 & 15 \\ 10 & 10 & 11 & 11 & 12 & 13 & 15 & 15 \end{bmatrix}$$

ทำการวัดประสิทธิภาพการบีบอัดภาพดิจิตอล โดยที่ A1 เมตริกซ์กู้คืน และX เป็นเมตริกซ์ต้นฉบับ มีผลลัพธ์ดังนี้

[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(A1),uint8(X))

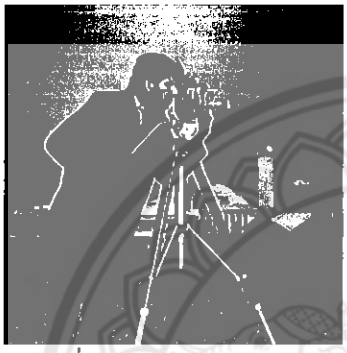
MSE= 0

SNR= Inf

P_SNR= Inf

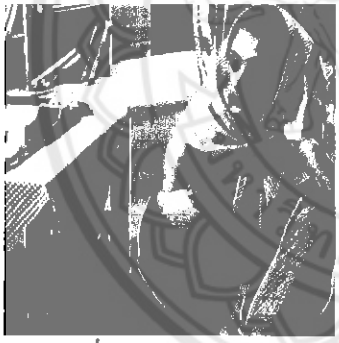
4.3 ภาพดิจิตอลก่อนการทำการบีบอัดภาพ และหลังการบีบอัดภาพ

ในการเตรียมภาพดิจิตอลก่อนทำการบีบอัดภาพที่นำมาแสดงเป็นตัวอย่าง นำไฟล์รูปภาพมาจากอินเตอร์เน็ต ดังแสดงในรูปที่ 4.1-4.5 เราจะทำการบันทึกภาพดิจิตอลหลังทำการบีบอัดภาพจากโปรแกรม Matlab2014a ซึ่งผลการทดลองการบีบอัดภาพในตารางที่ 4.1, 4.2, 4.3 และ4.4 เป็นการบีบอัดแบบไม่สูญเสีย ซึ่งจากสี่ตารางจะแตกต่างกันตรงค่า อัตราส่วนการบีบอัด จึงทำให้ภาพดิจิตอลหลังทำการบีบอัดภาพนั้นมีคุณสมบัติเหมือนกันทุกประการ เลยขอแสดงภาพดิจิตอลหลังทำการบีบอัดภาพเพียงชุดเดียว ดังแสดงในรูปที่ 4.6-4.10



รูปที่ 4.1 มนุษย์ถ่ายรูป[11]

รูปที่ 4.6 ภาพดิจิตอลกู้คืนมนุษย์ถ่ายรูป



รูปที่ 4.2 บาร์บารา[11]

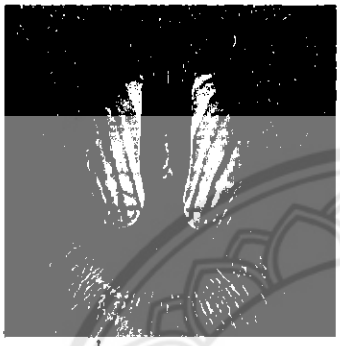
รูปที่ 4.7 ภาพดิจิตอลกู้คืนบาร์บารา

รูปที่ 4.3 ลีน่า[11]

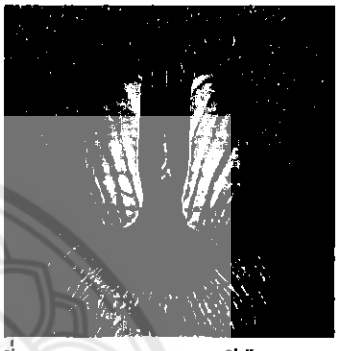
รูปที่ 4.8 ภาพดิจิตอลกู้คืนลีน่า

รูปที่ 4.4 พริกไทย[11]

รูปที่ 4.9 ภาพดิจิตอลกู้คืนพริกไทย



รูปที่ 4.5 แมนดริล[11]



รูปที่ 4.10 ภาพดิจิตอลกู้คืนแมนดริล

4.4 ผลการทดลองจากการบีบอัดภาพดิจิตอล

การทดลองการบีบอัดภาพด้วยอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ และตัวกรอง เดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต (DWT2(Haar,Db4)+Arithmetic Coding) มีการทดสอบวัดค่าอัตราส่วนการบีบอัด (Compression Ratio: CR) ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน(Signal to Noise Ratio: SNR) กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) นอกจากนั้นยังแบ่งผลการทดลองเป็น 2 แบบ คือ แบบไม่บันทึกตารางและแบบบันทึกตาราง

4.4.1 การทดลองการบีบอัดภาพ แบบไม่บันทึกตาราง

การทดลองการบีบอัดภาพด้วยอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ร่วมกับการเข้ารหัสเลขคณิต (DWT2(Haar)+Arithmetic Coding) มีการทดสอบวัดค่าอัตราส่วนการบีบอัด (Compression Ratio: CR) ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) ดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ร่วมกับการเข้ารหัสเลขคณิต แบบไม่บันทึกตาราง

Image Name	Total size(bytes)	Compression Raito	PSNR(db)	SNR(db)	MSE	Size comp(bytes)
cameraman	65536	2.7025:1	infinity	infinity	0	24250
barbara	262144	2.3237:1	infinity	infinity	0	112814
lena	786432	2.0185:1	infinity	infinity	0	389609
peppers	786432	1.9296:1	infinity	infinity	0	407554
mandril	786432	1.7111:1	infinity	infinity	0	459608

จากตาราง 4.2 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ร่วมกับการเข้ารหัสเลขคณิต พบว่ามีอัตราส่วนการบีบอัด (Compression Ratio: CR) อยู่ประมาณช่วง 1.7111-2.7025 ส่วนค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio:PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) มีค่าเป็นอนันต์ กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) เป็นศูนย์ เพราะว่าอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิด ฮาร์ ร่วมกับการเข้ารหัสเลขคณิตนั้นเป็นการบีบอัดข้อมูลแบบไม่สูญเสีย

การทดลองการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4ร่วมกับการเข้ารหัสเลขคณิต (DWT2(Db4)+Arithmetic Coding) มีการทดสอบวัดค่าอัตราส่วนการบีบอัด(Compression Ratio: CR) ค่าพีเอสเอ็นอาร์ (Peak Signal to Noise Ratio: PSNR) และค่าเอสเอ็นอาร์ (Signal to Noise Ratio: SNR) กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) ดังตารางที่ 4.3

ตารางที่ 4.3 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต แบบไม่บันทึกตาราง

Image Name	Total size(bytes)	Compression Raito	PSNR(db)	SNR(db)	MSE	Size comp(bytes)
cameraman	65536	1.5846:1	infinity	infinity	0	41357
Barbara	262144	1.4482:1	infinity	infinity	0	181015
Lena	786432	1.1334:1	infinity	infinity	0	693889
Peppers	786432	1.1344:1	infinity	infinity	0	693245
Mandril	786432	1.1332:1	infinity	infinity	0	693968

จากตาราง 4.3 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต พบว่ามีอัตราส่วนการบีบอัด (Compression Ratio: CR) อยู่ประมาณช่วง1.1332-1.5846 ส่วนค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) มีค่าเป็นอนันต์กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) เป็นศูนย์ เพราะว่าอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิตนั้นเป็นการบีบอัดข้อมูลแบบไม่สูญเสีย

4.4.2 การทดลองการบีบอัดภาพ แบบบันทึกตาราง

การทดลองการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ ร่วมกับการเข้ารหัสเลขคณิต (DWT2(Haar)+Arithmetic Coding) มีการทดสอบวัดค่าอัตราส่วนการบีบอัด(Compression Ratio: CR) มีการทดสอบวัดค่าอัตราส่วนการบีบอัด (Compression Ratio: CR) ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) ดังตารางที่ 4.4

ตารางที่ 4.4 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ ร่วมกับการเข้ารหัสเลขคณิต แบบบันทึกตาราง

Image Name	Total size(bytes)	Compression Raito	PSNR(db)	SNR(db)	MSE	Size comp(bytes)
cameraman	65536	2.6303:1	infinity	infinity	0	24916
Barbara	262144	2.2888:1	infinity	infinity	0	114533
Lena	786432	1.9986:1	infinity	infinity	0	393495
Peppers	786432	1.9053:1	infinity	infinity	0	412753
Mandril	786432	1.6841:1	infinity	infinity	0	466964

จากตาราง 4.4 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ ร่วมกับการเข้ารหัสเลขคณิต พบว่ามีอัตราส่วนการบีบอัด (Compression Ratio: CR) อยู่ประมาณช่วง 1.6841-2.6303 ส่วนค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) มีค่าเป็นอนันต์ กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) เป็นศูนย์ เพราะว่าอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ ร่วมกับการเข้ารหัสเลขคณิตนั้นเป็นการบีบอัดข้อมูลแบบไม่สูญเสีย

การทดลองการบีบอัดภาพด้วยการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4ร่วมกับการเข้ารหัสเลขคณิต (DWT2(Db4)+Arithmetic Coding) มีการทดสอบวัดค่าอัตราส่วนการบีบอัด (Compression Ratio: CR) ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) ดังตารางที่ 4.5

ตารางที่ 4.5 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต แบบบันทึกตาราง

Image Name	Total size(bytes)	Compression Raito	PSNR(db)	SNR(db)	MSE	Size comp(bytes)
cameraman	65536	1.4740:1	Infinity	infinity	0	44461
Barbara	262144	1.3594:1	Infinity	infinity	0	192839
Lena	786432	1.0646:1	Infinity	infinity	0	738701
Peppers	786432	1.0656:1	Infinity	infinity	0	737985
Mandril	786432	1.0703:1	Infinity	infinity	0	738812

จากตาราง 4.5 ผลการทดลองการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต พบว่ามีอัตราส่วนการบีบอัด(Compression Ratio: CR) อยู่ประมาณช่วง 1.0646-1.4740 ส่วนค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (Peak Signal to Noise Ratio: PSNR) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) มีค่าเป็นอนันต์ กับค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE) เป็นศูนย์ เพราะว่าอัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิด เดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิตนั้นเป็นการบีบอัดข้อมูลแบบไม่สูญเสีย

4.5 สรุปผลการทดลอง

ภาพที่นำมาทดสอบการบีบอัดภาพทั้ง 2 วิธีนี้ ใช้ภาพทดสอบจำนวน 5 ภาพ คือ ภาพระดับเทามนุษย์ถ่ายรูป (cameraman) ภาพระดับเทาบาร์บารา (Barbara) ภาพสีลีน่า (lena) ภาพสีพริกไทย(peppers) และภาพสีแมนดริล (mandril) ซึ่งสามารถแสดงเป็นตารางเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอลได้ดังนี้

4.5.1 การเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบไม่บันทึกตาราง

ตารางที่ 4.6 ตารางการเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบไม่บันทึกตาราง

Algorithm	**Compression Ratio**				
	cameraman	barbara	lena	peppers	mandril
DWT(haar)+Arithmetic	2.7025	2.3237	2.0185	1.9296	1.7111
DWT(db4)+Arithmetic	1.5846	1.4482	1.1334	1.1344	1.1332

จากตารางที่ 4.6 ตารางการเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบไม่บันทึกตารางพบว่าการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ ร่วมกับการเข้ารหัสเลขคณิตแบบไม่บันทึกตาราง มีอัตราส่วนการบีบอัด(Compression Ratio: CR) สูงกว่าการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต แบบไม่บันทึกตาราง

4.5.2 การเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบบันทึกตาราง

ตารางที่ 4.7 ตารางการเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบบันทึกตาราง

Algorithm	**Compression Ratio**				
	cameraman	barbara	Lena	peppers	mandril
DWT(haar)+Arithmetic	2.6303	2.2888	1.9986	1.9053	1.6841
DWT(db4)+Arithmetic	1.4740	1.3594	1.0646	1.0656	1.0703

เนื่องจากในโครงงานเล่มนี้เป็นการศึกษาการบีบอัดภาพแบบไม่สูญเสีย เลยทำให้ค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด (psnr) และค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (snr) กับค่าผิดพลาดเฉลี่ยกำลังสอง (mse) มีค่าเท่ากันทั้งสองวิธี จึงต้องวัดประสิทธิภาพด้วยการใช้ค่าอัตราส่วนการบีบอัด

จากตารางที่ 4.7 ตารางการเปรียบเทียบประสิทธิภาพการบีบอัดภาพดิจิตอล แบบบันทึกตารางพบว่าการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองฮาร์ ร่วมกับการเข้ารหัสเลขคณิตแบบบันทึกตาราง มีอัตราส่วนการบีบอัด(Compression Ratio: CR) สูงกว่าการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต แบบบันทึกตาราง

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

ในโครงงานนี้ผู้ทำวิจัยได้ทำการศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิตซึ่งเป็นการบีบอัดภาพแบบไม่มีการสูญเสียข้อมูลโดยใช้อัลกอริทึมการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ระดับ ซึ่งมีการเลือกใช้ตัวกรองเวฟเล็ตที่ใช้ในการศึกษาการบีบอัดภาพ 2 ตัวกรอง คือ ตัวกรองฮาร์กับ เดบุชชี4 เนื่องจากในโครงงานผู้วิจัยได้ทำการศึกษาการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยภาพที่นำมาทดสอบนั้นมีทั้งข้อมูลที่มีการเปลี่ยนแปลงทันทีทันใดและไม่เปลี่ยนแปลงทันทีทันใด จึงได้นำเวฟเล็ตตัวกรองฮาร์และเวฟเล็ตตัวกรองเดบุชชี4 มาใช้ในการศึกษาการบีบอัดข้อมูลภาพและนอกจากนี้ยังมีการเลือกใช้อัลกอริทึมในการเข้ารหัสเป็นการเข้ารหัสเลขคณิตกับการถอดรหัสเลขคณิต

เพื่อให้การศึกษาเทคนิคการบีบอัดภาพด้วยการใช้การแปลงเวฟเล็ตร่วมกับการเข้ารหัสเลขคณิตได้อย่างมีประสิทธิภาพ ผู้วิจัยมีขั้นตอนในการดำเนินการแบ่งเป็น 2 ขั้นตอน คือ ขั้นตอนในการดำเนินการการบีบอัดภาพ เริ่มต้นจากการรับภาพเข้ามาในพื้นที่ทำงานของโปรแกรม Matlab เข้ามาจะอยู่ในรูปเมตริกซ์ ต่อมารับภาพจากพื้นที่ทำงานที่เข้ามาในรูปเมตริกซ์มาแปลงด้วยเวฟเล็ตเต็มหน่วยแบบ 2 มิติ 3 ระดับ (Discrete Wavelet Transform: DWT) โดยใช้ตัวกรองฮาร์หรือตัวกรองเดบุชชี4 ซึ่งจะได้ผลลัพธ์เมตริกซ์สัมประสิทธิ์เวฟเล็ตทั้ง 3 ระดับ จากนั้นนำเมตริกซ์สัมประสิทธิ์เวฟเล็ตทั้ง 3 ระดับ มาแปลงเป็นเวกเตอร์แนวนอน ส่วนการเข้ารหัสเลขคณิต เริ่มต้นจากการแมปข้อมูลจากเวกเตอร์แนวนอนของสัมประสิทธิ์เวฟเล็ตไปเป็นลำดับในการเข้ารหัสเลขคณิต หลังจากนั้นจะนำข้อมูลของสมาชิกที่ไม่ซ้ำกันไปเก็บอยู่ในรูปเมตริกซ์ตารางแล้วนำข้อมูลเมตริกซ์ตารางที่ได้มาทำการบันทึกตารางกระแสข้อมูลบิตเป็นไฟล์ข้อมูลฐานสอง เข้ารหัสจนครบทุกลำดับการเข้ารหัสเลขคณิต ผลลัพธ์ที่ได้คือกระแสข้อมูลบิต แล้วทำการคำนวณหาค่าอัตราส่วนการบีบอัดข้อมูล และขั้นตอนในการดำเนินการคายการบีบอัดภาพนำไฟล์กระแสข้อมูลบิต และตารางกระแสข้อมูลบิตที่ได้จากการดำเนินการบีบอัดภาพ แล้วนำมาถอดรหัสเลขคณิต ทำให้ได้รับข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ต ซึ่งข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตจะอยู่ในรูปเวกเตอร์ ต่อมานำข้อมูลเมตริกซ์สัมประสิทธิ์เวฟเล็ตอยู่ในรูปเวกเตอร์ โดยการแปลงเวกเตอร์ที่ได้รับเป็นเมตริกซ์สัมประสิทธิ์กู้คืน นำเมตริกซ์สัมประสิทธิ์กู้คืนที่ได้เพื่อทำการแปลงเวฟเล็ตเต็มหน่วยผกผันแบบ 2 มิติ 3 ระดับ โดยใช้ตัวกรองฮาร์ หรือตัวกรองเดบุชชี4 สุดท้ายจะได้ผลลัพธ์ภาพต้นฉบับและทำการคำนวณหาค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด(Peak Signal to Noise Ratio:PSNR)

และอัตราส่วนสัญญาณต่อสัญญาณรบกวน(Signal to Noise Ratio:SNR) กับการหาค่าผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error: MSE)

ในการทดสอบประสิทธิภาพการบีบอัดภาพดิจิตอล ได้ออกแบบการทดลองการบีบอัดภาพด้วยการแปลงเวฟเล็ตชนิดตัวกรองฮาร์ และชนิดตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต นั้นจะแบ่งการทดลองเป็นการบีบอัดแบบบันทึกตารางการเข้ารหัสเลขคณิต กับการบีบอัดแบบไม่บันทึกตารางการเข้ารหัสเลขคณิต จากผลการทดลองพบว่าการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ โดยใช้ตัวกรองฮาร์ร่วมกับการเข้ารหัสเลขคณิต มีประสิทธิภาพในการบีบอัดภาพดีกว่าการแปลงเวฟเล็ตเต็มหน่วยแบบ 2มิติ โดยใช้ตัวกรองเดบุชชี4 ร่วมกับการเข้ารหัสเลขคณิต เพราะว่าข้อมูลของภาพที่ใช้ในโครงงานนี้มีการเปลี่ยนแปลงของสีไม่มากทำให้เมื่อเราใช้ตัวกรองฮาร์ ในการแปลงจึงทำได้ดีกว่าตัวกรองเดบุชชี4นอกจากนี้ผลการทดลองแบบไม่บันทึกตารางการเข้ารหัสเลขคณิตจะมีอัตราส่วนการบีบอัดที่สูงกว่าแบบบันทึกตารางการเข้ารหัสเลขคณิตในหลักทฤษฎี โดยที่คุณภาพของภาพกู้คืนนั้นเท่ากันซึ่งวัดได้จากดัชนีชี้วัดประสิทธิภาพ 3 ตัว ซึ่งมีค่าเท่ากัน คือ ดัชนีวัดค่าผิดพลาดเฉลี่ยกำลังสอง (MSE) เป็นศูนย์และดัชนีค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวนสูงสุด(PSNR) กับดัชนีวัดค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน(SNR) มีค่าเป็นอนันต์

5.2 ปัญหาและแนวทางแก้ไข

จากปัญหาที่พบในการศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ต ร่วมกับการเข้ารหัสเลขคณิต จำเป็นต้องศึกษา แนวทาง แก้ไขปัญหาที่เกิดขึ้น ดังต่อไปนี้

1) ปัญหาที่พบในการทำโครงงานนี้ คือ ขนาดไฟล์ภาพต้นฉบับที่ใช้ในการทดสอบการบีบอัดภาพมีขนาดไม่เท่ากับขนาดไฟล์ภาพตามทฤษฎีเนื่องจากภาพที่ได้ไม่ใช่ภาพต้นฉบับจริงๆ จึงเลือกใช้ภาพทดสอบที่นิยมในด้านการประมวลผลภาพ (Image Processing) เพราะสามารถหาภาพต้นฉบับได้ง่าย
2) ปัญหาที่พบในการทำโครงงานนี้ คือ ภาพดิจิตอลสีไม่สามารถนำไปแปลงเวฟเล็ตเต็มหน่วยได้โดยตรง จึงต้องแยกเมตริกซ์ภาพสีเป็นเมตริกซ์สามสีหลักก่อนแล้วนำเมตริกซ์แต่ละสีไปแปลงเวฟเล็ตเต็มหน่วยทีละอัน
3) ปัญหาที่พบในการทำโครงงานนี้ คือ เนื่องจากเราไม่สามารถนำข้อมูลที่เราได้มาเข้ารหัสเลขคณิตได้โดยตรง จึงต้องมีการสร้างฟังก์ชั่นใหม่ขึ้นมาเพื่อใช้ในการแปลงข้อมูลเมตริกซ์เป็นข้อมูลเว็กเตอร์ แล้วทำการแปลงเว็กเตอร์ข้อมูลให้เป็นเว็กเตอร์ลำดับในการเข้ารหัสเลขคณิต และทำการนับข้อมูลของเว็กเตอร์ลำดับอยู่ในรูปตัวแปร count จึงสามารถทำการเข้ารหัสเลขคณิต

5.3 ข้อเสนอแนะและแนวทางการพัฒนา

จากการศึกษาเทคนิคการบีบอัดภาพด้วยการแปลงเวฟเล็ต ร่วมกับการเข้ารหัสเลขคณิตได้มีแนวทางในการนำเสนอเพื่อการพัฒนาเทคนิคการบีบอัดภาพดังต่อไปนี้

1) ควรปรับปรุงโปรแกรมให้มีการทำงานที่เร็วขึ้นเพื่อลดเวลาในการประมวลผล
2) ถ้ามีการเปลี่ยนจากการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ จากระดับ 3 เป็นการแปลงการแปลงเวฟเล็ตเต็มหน่วยแบบ 2 มิติ ระดับที่ 4 นั้นจะทำให้ผลของอัตราส่วนการบีบอัดภาพดีขึ้นไปอีก
3) นอกจากนี้ถ้ามีการใช้ ตัวกรองเดบุชชี ที่สูงกว่าตัวกรองเดบุชชี4 ผลของอัตราส่วนการบีบอัดภาพดีขึ้น

เอกสารอ้างอิง

[1] Rafael C, Gonzalez and Richard E. Wood, (2010). Digital Image Processing (3rd Edition): Prentice Hall.

[2] พนมขวัญ ริยะมงคล. เอกสารประกอบการเรียนการสอนวิชา Digital image Processing (305434).

[3] กิตติ อัตถกิจมงคล. (2546). การบีบอัดสัญญาณภาพถ่ายจากดาวเทียมโดยการแปลงเวฟเล็ต และการศึกษาเทคนิคการบีบอัดสัญญาณภาพของ JPEG 2000. คณะวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีสุรนารี

[4] Alasdair McAndrew, Jung-Hua wang and Chun-Shun Tseng, (2010). Introduction to Digital Image Processing with MATLAB Asia Edition. Taiwan: Cengage Learning Asia Pte Ltd

[5] Khalid Sayood, Introduction to data compression (3rd Edition): Morgan Kaufmanm Publication.

[6] การบันทึกไฟล์ข้อมูลฐานสองในแม็ทแล็บ. สืบค้นวันที่ 10 ธันวาคม 2557, จาก http://www.mathworks.com/help/matlab/ref/fwrite.html.

[7] การคำนวณอัตราส่วนการบีบอัดข้อมูล. สืบค้นวันที่ 10 ธันวาคม 2557, จาก http://en.wikipedia.org/wiki/Data_compression_ratio.

[8] การคำนวณหาค่าผิดพลาดเฉลี่ยกำลังสอง. สืบค้นวันที่ 10 ธันวาคม 2557, จาก http://en.wikipedia.org/wiki/Mean_squared_error.

[9] การคำนวณค่าพีเอสเอ็นอาร์. สืบค้นวันที่ 10 ธันวาคม 2557, จาก http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

[10] การคำนวณค่าเอสเอ็นอาร์. สืบค้นวันที่ 10 ธันวาคม 2557, จาก http://en.wikipedia.org/ wiki/Signal-to-noise_ratio.

[11] Morgan McGuire and N.D. McGuire, Graphics Data. สืบค้นวันที่ 10 ธันวาคม 2557, จาก http://graphics.cs.williams.edu/data/images.xml.

ภาคผนวก ก

รายละเอียดฟังก์ชั่น new_arithme

```
function [Y]=new_arithme(X,M,N,namesave,namesavetable,namsesavecounts)
Xn=X(:)'; %แปลงเมตริกซ์ให้เป็นเวกเตอร์
% ' Compute Table of Symbols ...'
% Table เอาไว้เก็บค่าจริง(ซ้ำไม่เอา)
% Table(1)=Data(1);
Table(1)=Xn(1);
S_AC=size(Xn);
  for jAC=1:S_AC(2)
    S_2AC=size(Table);Flag=0;
    for kAC=1:S_2AC(2)
      if (Table(kAC)==Xn(jAC))
        Flag=1;
      end;
    end;
    if (Flag==0) Table(S_2AC(2)+1)=Xn(jAC); end;
  end;
fid = fopen(namesavetable, 'w');
fwrite(fid, Table, 'ubit1');
fclose(fid);
%  ' Compute the probability of the symbols...'
%-----------------------------------------------------------------------------------------
 S_2AC=size(Table);
 Counts(1:S_2AC(2))=0;
  for kAC=1:S_AC(2)
    iAC=1;
    while (Table(iAC)~=Xn(kAC))
      iAC=iAC+1;
    end;
     Xn(kAC)=iAC;
     Counts(iAC)=Counts(iAC)+1;
  end;
```

```
% Counts
fid = fopen(namesavecounts, 'w'); % บันทึกจำนวนข้อมูลที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
fwrite(fid, Counts, 'ubit1');
fclose(fid);
code=0;
%    ' Apply Arithmetic coding .... Now'
code = arithenco(Xn,Counts) %เข้ารหัสเลขคณิต
fid = fopen(namesave, 'w');
fwrite(fid, code, 'ubit1'); %บันทึกภาพที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
fclose(fid);
Counts=double(Counts);
%      ' Compute data size'
    length_Xn=0;
    Size_Counts=size(Counts);
    for iAD=1:Size_Counts(2)
    length_Xn=length_Xn+Counts(iAD);
    end;
%      'Apply Arithmetic Decode..'
    Xno = arithdeco(code,Counts,length_Xn) %ถอดรหัสเลขคณิต
    Xnew=0;
%      'Return original data'
    for iAD=1:length_Xn
     Xnew(iAD)=Table(Xno(iAD));
end;
Y=reshape(Xnew,[M,N]); %รีคอนสตรัค
```

ภาคผนวก ข

โปรแกรมการบีบอัดภาพดิจิตอล

Wavelet Transform(Haar)+Arithmetic Coding

```
close all; clear all; clc;
tic % ฟังก์ชั่นจับเวลา
X=imread('cameraman.tif'); %X คือข้อมูลภาพดิจิตอลระดับเทามนุษย์ถ่ายรูป
figure(1),imshow(X); % แสดงภาพต้นแบบระดับเทามนุษย์ถ่ายรูป
[M,N]=size(X); %ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
% แปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
[cA1,cH1,cV1,cD1] = dwt2(X,'haar');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
[cA2,cH2,cV2,cD2] = dwt2(cA1,'haar');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
[cA3,cH3,cV3,cD3] = dwt2(cA2,'haar');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1i=cA1(:);
cH1i=cH1(:);
cV1i=cV1(:);
cD1i=cD1(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2i=cA2(:);
cH2i=cH2(:);
cV2i=cV2(:);
cD2i=cD2(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3i=cA3(:);
cH3i=cH3(:);
cV3i=cV3(:);
cD3i=cD3(:);
```

```
%บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_haar_cameraman_cA1o.PM','bit_haar_cameraman_cH1o.PM',
'bit_haar_cameraman_cV1o.PM','bit_haar_cameraman_cD1o.PM'
'bit_haar_cameraman_cA2o.PM','bit_haar_cameraman_cH2o.PM','bit_haar_cameraman_cV2o.P
M',
'bit_haar_cameraman_cD2o.PM'
'bit_haar_cameraman_cA3o.PM','bit_haar_cameraman_cH3o.PM','bit_haar_cameraman_cV3o.P
M',
'bit_haar_cameraman_cD3o.PM'};
%-------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_haar_cameraman_cA1o.PM','bit_table_haar_cameraman_cH1o.PM',
'bit_table_haar_cameraman_cV1o.PM','bit_table_haar_cameraman_cD1o.PM'
'bit_table_haar_table_cameraman_cA2o.PM','bit_table_haar_cameraman_cH2o.PM',
'bit_table_haar_cameraman_cV2o.PM','bit_table_haar_cameraman_cD2o.PM'
'bit_table_haar_cameraman_cA3o.PM','bit_table_haar_cameraman_cH3o.PM',
'bit_table_haar_cameraman_cV3o.PM','bit_table_haar_cameraman_cD3o.PM'};
%-------------------------------------------------------------------------------------------
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_haar_cameraman_cA1o.PM','bit_count_haar_cameraman_cH1o.PM',
'bit_count_haar_cameraman_cV1o.PM','bit_count_haar_cameraman_cD1o.PM'
'bit_count_haar_table_cameraman_cA2o.PM','bit_count_haar_cameraman_cH2o.PM',
'bit_count_haar_cameraman_cV2o.PM','bit_count_haar_cameraman_cD2o.PM'
'bit_count_haar__cameraman_cA3o.PM','bit_count_haar_cameraman_cH3o.PM',
'bit_count_haar_cameraman_cV3o.PM','bit_count_haar_cameraman_cD3o.PM'};
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
cA1o1=new_arithme(cA1i,128,128,Na{1},Ta{1},Ca{1});
cH1o1=new_arithme(cH1i,128,128,Na{2},Ta{2},Ca{2});
cV1o1=new_arithme(cV1i,128,128,Na{3},Ta{3},Ca{3});
cD1o1=new_arithme(cD1i,128,128,Na{4},Ta{4},Ca{4});
```

```
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
cA2o1=new_arithme(cA2i,64,64,Na{5},Ta{5},Ca{5});
cH2o1=new_arithme(cH2i,64,64,Na{6},Ta{6},Ca{6});
cV2o1=new_arithme(cV2i,64,64,Na{7},Ta{7},Ca{7});
cD2o1=new_arithme(cD2i,64,64,Na{8},Ta{8},Ca{8});
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
cA3o1=new_arithme(cA3i,32,32,Na{9},Ta{9},Ca{9});
cH3o1=new_arithme(cH3i,32,32,Na{10},Ta{10},Ca{10});
cV3o1=new_arithme(cV3i,32,32,Na{11},Ta{11},Ca{11});
cD3o1=new_arithme(cD3i,32,32,Na{12},Ta{12},Ca{12});
%-----------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
A3 = idwt2(cA3o1,cH3o1,cV3o1,cD3o1,'haar');
%-----------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
A2 = idwt2(cA2o1,cH2o1,cV2o1,cD2o1,'haar');
%-----------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
A1 = idwt2(cA1o1,cH1o1,cV1o1,cD1o1,'haar');
%-----------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
figure(), imshow(uint8(A1));title('reconstruct DWT  Lv1 ');
%-----------------------------------------------------------------------------------------------
```

```
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
figure(), imshow(uint8(A2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
figure(),imshow(uint8(A3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(A1),uint8(X))
toc
```

```
close all; clear all; clc;
tic %ฟังก์ชั่นจับเวลา
X=imread('barbara.png'); %X คือข้อมูลภาพดิจิตอลระดับเทาบาร์บารา
figure(1),imshow(X); % แสดงภาพต้นแบบระดับเทาบาร์บารา
[M,N]=size(X); %ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
% แปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
[cA1,cH1,cV1,cD1] = dwt2(X,'haar');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
[cA2,cH2,cV2,cD2] = dwt2(cA1,'haar');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
[cA3,cH3,cV3,cD3] = dwt2(cA2,'haar');
%-----------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1i=cA1(:);
cH1i=cH1(:);
cV1i=cV1(:);
cD1i=cD1(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2i=cA2(:);
cH2i=cH2(:);
cV2i=cV2(:);
cD2i=cD2(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3i=cA3(:);
cH3i=cH3(:);
cV3i=cV3(:);
cD3i=cD3(:);
```

```
%-----------------------------------------------------------------------------------------------
%บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_haar_barbara_cA1o.PM','bit_haar_barbara_cH1o.PM','bit_haar_barbara_cV1o.PM',
'bit_haar_barbara_cD1o.PM'
'bit_haar_barbara_cA2o.PM','bit_haar_barbara_cH2o.PM','bit_haar_barbara_cV2o.PM',
'bit_haar_barbaracD2o.PM'
'bit_haar_barbara_cA3o.PM','bit_haar_barbara_cH3o.PM','bit_haar_barbara_cV3o.PM',
'bit_haar_barbara_cD3o.PM'};
%-----------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_haar_barbara_cA1o.PM','bit_table_haar_barbara_cH1o.PM',
'bit_table_haar_barbara_cV1o.PM','bit_table_haar_barbara_cD1o.PM'
'bit_table_haar_barbara_cA2o.PM','bit_table_haar_barbara_cH2o.PM',
'bit_table_haar_barbara_cV2o.PM','bit_table_haar_barbaracD2o.PM'
'bit_table_haar_barbara_cA3o.PM','bit_table_haar_barbara_cH3o.PM',
'bit_table_haar_barbara_cV3o.PM','bit_table_haar_barbaracD3o.PM'};
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_haar_barbara_cA1o.PM','bit_count_haar_barbara_cH1o.PM',
'bit_count_haar_barbara_cV1o.PM','bit_count_haar_barbara_cD1o.PM'
'bit_count_haar_barbara_cA2o.PM','bit_count_haar_barbara_cH2o.PM',
'bit_count_haar_barbara_cV2o.PM','bit_count_haar_barbaracD2o.PM'
'bit_count_haar_barbara_cA3o.PM','bit_count_haar_barbara_cH3o.PM',
'bit_count_haar_barbara_cV3o.PM','bit_count_haar_barbaracD3o.PM'};
```

```
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
cA1o1=new_arithme(cA1i,256,256,Na{1},Ta{1},Ca{1});
cH1o1=new_arithme(cH1i,256,256,Na{2},Ta{2},Ca{2});
cV1o1=new_arithme(cV1i,256,256,Na{3},Ta{3},Ca{3});
cD1o1=new_arithme(cD1i,256,256,Na{4},Ta{4},Ca{4});
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
cA2o1=new_arithme(cA2i,128,128,Na{5},Ta{5},Ca{5});
cH2o1=new_arithme(cH2i,128,128,Na{6},Ta{6},Ca{6});
cV2o1=new_arithme(cV2i,128,128,Na{7},Ta{7},Ca{7});
cD2o1=new_arithme(cD2i,128,128,Na{8},Ta{8},Ca{8});
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
cA3o1=new_arithme(cA3i,64,64,Na{9},Ta{9},Ca{9});
cH3o1=new_arithme(cH3i,64,64,Na{10},Ta{10},Ca{10});
cV3o1=new_arithme(cV3i,64,64,Na{11},Ta{11},Ca{11});
cD3o1=new_arithme(cD3i,64,64,Na{12},Ta{12},Ca{12});

%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
A3 = idwt2(cA3o1,cH3o1,cV3o1,cD3o1,'haar');
```

```
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
A2 = idwt2(cA2o1,cH2o1,cV2o1,cD2o1,'haar');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
A1 = idwt2(cA1o1,cH1o1,cV1o1,cD1o1,'haar');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
figure(), imshow(uint8(A1));title('reconstruct DWT  Lv1 ');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
figure(), imshow(uint8(A2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
figure(). imshow(uint8(A3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(A1),uint8(X))
toc
```

```
close all; clear all; clc;
tic %ฟังก์ชั่นจับเวลา
X= imread('lena512color.tiff'); %X คือข้อมูลภาพดิจิตอลสีลีน่า
figure(1);
imshow(X); title('original image'); % แสดงภาพต้นแบบสีลีน่า
[sr,sc,pp]=size(X); %ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
%--------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 1 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA1r,cH1r,cV1r,cD1r] = dwt2(double(X(:,:,1)),'haar');
% Do the 2D DWT on the green component
[cA1g,cH1g,cV1g,cD1g] = dwt2(double(X(:,:,2)),'haar');
% Do the 2D DWT on the blue component
[cA1b,cH1b,cV1b,cD1b] = dwt2(double(X(:,:,3)), 'haar');
%--------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 2 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA2r,cH2r,cV2r,cD2r] = dwt2(double(cA1r(:,:,1)),'haar');
% Do the 2D DWT on the green component
[cA2g,cH2g,cV2g,cD2g] = dwt2(double(cA1g(:,:,1)),'haar');
% Do the 2D DWT on the blue component
[cA2b,cH2b,cV2b,cD2b] = dwt2(double(cA1b(:,:,1)),'haar');
%--------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 3 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA3r,cH3r,cV3r,cD3r] = dwt2(double(cA2r(:,:,1)),'haar');
```

```
% Do the 2D DWT on the green component
[cA3g,cH3g,cV3g,cD3g] = dwt2(double(cA2g(:,:,1)),'haar');
% Do the 2D DWT on the blue component
[cA3b,cH3b,cV3b,cD3b] = dwt2(double(cA2b(:,:,1)),'haar');
%-------------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1ri=cA1r(:);
cA1gi=cA1g(:);
cA1bi=cA1b(:);

cH1ri=cH1r(:);
cH1gi=cH1g(:);
cH1bi=cH1b(:);
cV1ri=cV1r(:);
cV1gi=cV1g(:);
cV1bi=cV1b(:);

cD1ri=cD1r(:);
cD1gi=cD1g(:);
cD1bi=cD1b(:);
%-------------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2ri=cA2r(:);
cA2gi=cA2g(:);
cA2bi=cA2b(:);
```

```
cH2ri=cH2r(:);
cH2gi=cH2g(:);
cH2bi=cH2b(:);
cV2ri=cV2r(:);
cV2gi=cV2g(:);
cV2bi=cV2b(:);

cD2ri=cD2r(:);
cD2gi=cD2g(:);
cD2bi=cD2b(:);
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3ri=cA3r(:);
cA3gi=cA3g(:);
cA3bi=cA3b(:);

cH3ri=cH3r(:);
cH3gi=cH3g(:);
cH3bi=cH3b(:);

cV3ri=cV3r(:);
cV3gi=cV3g(:);
cV3bi=cV3b(:);

cD3ri=cD3r(:);
cD3gi=cD3g(:);
cD3bi=cD3b(:);
```

% บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง

Na={'bit_haar_lena_cA1ro.PM','bit_haar_lena_cA1go.PM','bit_haar_lena_cA1bo.PM'
'bit_haar_lena_cH1ro.PM','bit_haar_lena_cH1go.PM','bit_haar_lena_cH1bo.PM'
'bit_haar_lena_cV1ro.PM','bit_haar_lena_cV1go.PM','bit_haar_lena_cV1bo.PM'
'bit_haar_lena_cD1ro.PM','bit_haar_lena_cD1go.PM','bit_haar_lena_cD1bo.PM'
'bit_haar_lena_cA2ro.PM','bit_haar_lena_cA2go.PM','bit_haar_lena_cA2bo.PM'
'bit_haar_lena_cH2ro.PM','bit_haar_lena_cH2go.PM','bit_haar_lena_cH2bo.PM'
'bit_haar_lena_cV2ro.PM','bit_haar_lena_cV2go.PM','bit_haar_lena_cV2bo.PM'
'bit_haar_lena_cD2ro.PM','bit_haar_lena_cD2go.PM','bit_haar_lena_cD2bo.PM'
'bit_haar_lena_cA3ro.PM','bit_haar_lena_cA3go.PM','bit_haar_lena_cA3bo.PM'
'bit_haar_lena_cH3ro.PM','bit_haar_lena_cH3go.PM','bit_haar_lena_cH3bo.PM'
'bit_haar_lena_cV3ro.PM','bit_haar_lena_cV3go.PM','bit_haar_lena_cV3bo.PM'
'bit_haar_lena_cD3ro.PM','bit_haar_lena_cD3go.PM','bit_haar_lena_cD3bo.PM'};

%----------

%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง

Ta={'bit_table_haar_lena_cA1ro.PM','bit_table_haar_lena_cA1go.PM',
'bit_table_haar_lena_cA1bo.PM'
'bit_table_haar_lena_cH1ro.PM','bit_table_haar_lena_cH1go.PM','bit_table_haar__ena_cH1bo.P
M'
'bit_table_haar_lena_cV1ro.PM','bit_table_haar_lena_cV1go.PM','bit_table_haar_lena_cV1bo.P
M'
'bit_table_haar_lena_cD1ro.PM','bit_table_haar_lena_cD1go.PM','bit_table_haar_lena_cD1bo.P
M'
'bit_table_haar_lena_cA2ro.PM','bit_table_haar_lena_cA2go.PM','bit_table_haar_lena_cA2bo.P
M'
'bit_table_haar_lena_cH2ro.PM','bit_table_haar_lena_cH2go.PM','bit_table_haar_lena_cH2bo.P
M'
'bit_table_haar_lena_cV2ro.PM','bit_table_haar_lena_cV2go.PM','bit_table_haar_lena_cV2bo.P
M'
'bit_table_haar_lena_cD2ro.PM','bit_table_haar_lena_cD2go.PM','bit_table_haar_lena_cD2bo.P
M'

'bit_table_haar__lena_cA3ro.PM','bit_table_haar_lena_cA3go.PM','bit_table_haar_lena_cA3bo.P
M'
'bit_table_haar_lena_cH3ro.PM','bit_table_haar_lena_cH3go.PM','bit_table_haar_lena_cH3bo.P
M'
'bit_table_haar_lena_cV3ro.PM','bit_table_haar_lena_cV3go.PM','bit_table_haar_lena_cV3bo.P
M'
'bit_table_haar_lena_cD3ro.PM','bit_table_haar_lena_cD3go.PM','bit_table_haar_lena_cD3bo.P
M'};

```
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_haar_lena_cA1ro.PM','bit_count_haar_lena_cA1go.PM',
'bit_count_haar_lena_cA1bo.PM'
'bit_count_haar_lena_cH1ro.PM','bit_count_haar_lena_cH1go.PM','bit_count_haar_lena_cH1bo.
PM'
'bit_count_haar_lena_cV1ro.PM','bit_count_haar_lena_cV1go.PM','bit_count_haar_lena_cV1bo.
PM'
'bit_count_haar_lena_cD1ro.PM','bit_count_haar_lena_cD1go.PM','bit_count_haar_lena_cD1bo.
PM'
'bit_count_haar_lena_cA2ro.PM','bit_count_haar_lena_cA2go.PM','bit_count_haar_lena_cA2bo.
PM'
'bit_count_haar_lena_cH2ro.PM','bit_count_haar_lena_cH2go.PM','bit_count_haar_lena_cH2bo.
PM'
'bit_count_haar_lena_cV2ro.PM','bit_count_haar_lena_cV2go.PM','bit_count_haar_lena_cV2bo.
PM'
'bit_count_haar_lena_cD2ro.PM','bit_count_haar_lena_cD2go.PM','bit_count_haar_lena_cD2bo.
PM'
'bit_count_haar_lena_cA3ro.PM','bit_count_haar_lena_cA3go.PM','bit_count_haar_lena_cA3bo.
PM'
'bit_count_haar_lena_cH3ro.PM','bit_count_haar_lena_cH3go.PM','bit_count_haar_lena_cH3bo.
PM'
'bit_count_haar_lena_cV3ro.PM','bit_count_haar_lena_cV3go.PM','bit_count_haar_lena_cV3bo.
PM'
'bit_count_haar_lena_cD3ro.PM','bit_count_haar_lena_cD3go.PM','bit_count_haar_lena_cD3bo.
PM'};
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
%cA1
cA1ro1=new_arithme(cA1ri,256,256,Na{1},Ta{1},Ca{1});
cA1go1=new_arithme(cA1gi,256,256,Na{2},Ta{2},Ca{2});
cA1bo1=new_arithme(cA1bi,256,256,Na{3},Ta{3},Ca{3});
%cH1
```

```
cH1ro1=new_arithme(cH1ri,256,256,Na{4},Ta{4},Ca{4});
cH1go1=new_arithme(cH1gi,256,256,Na{5},Ta{5},Ca{5});
cH1bo1=new_arithme(cH1bi,256,256,Na{6},Ta{6},Ca{6});
%cV1
cV1ro1=new_arithme(cV1ri,256,256,Na{7},Ta{7},Ca{7});
cV1go1=new_arithme(cV1gi,256,256,Na{8},Ta{8},Ca{8});
cV1bo1=new_arithme(cV1bi,256,256,Na{9},Ta{9},Ca{9});
```

```
%cD1
cD1ro1=new_arithme(cD1ri,256,256,Na{10},Ta{10},Ca{10});
cD1go1=new_arithme(cD1gi,256,256,Na{11},Ta{11},Ca{11});
cD1bo1=new_arithme(cD1bi,256,256,Na{12},Ta{12},Ca{12});
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
%cA2
cA2ro1=new_arithme(cA2ri,128,128,Na{13},Ta{13},Ca{13});
cA2go1=new_arithme(cA2gi,128,128,Na{14},Ta{14},Ca{14});
cA2bo1=new_arithme(cA2bi,128,128,Na{15},Ta{15},Ca{15});
%cH2
cH2ro1=new_arithme(cH2ri,128,128,Na{16},Ta{16},Ca{16});
cH2go1=new_arithme(cH2gi,128,128,Na{17},Ta{17},Ca{17});
cH2bo1=new_arithme(cH2bi,128,128,Na{18},Ta{18},Ca{18});
%cV2
cV2ro1=new_arithme(cV2ri,128,128,Na{19},Ta{19},Ca{19});
cV2go1=new_arithme(cV2gi,128,128,Na{20},Ta{20},Ca{20});
cV2bo1=new_arithme(cV2bi,128,128,Na{21},Ta{21},Ca{21});
%cD2
cD2ro1=new_arithme(cD2ri,128,128,Na{22},Ta{22},Ca{22});
cD2go1=new_arithme(cD2gi,128,128,Na{23},Ta{23},Ca{23});
cD2bo1=new_arithme(cD2bi,128,128,Na{24},Ta{24},Ca{24});
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
%cA3
cA3ro1=new_arithme(cA3ri,64,64,Na{25},Ta{25},Ca{25});
cA3go1=new_arithme(cA3gi,64,64,Na{26},Ta{26},Ca{26});
cA3bo1=new_arithme(cA3bi,64,64,Na{27},Ta{27},Ca{27});
```

```
%cH3
cH3ro1=new_arithme(cH3ri,64,64,Na{28},Ta{28},Ca{28});
cH3go1=new_arithme(cH3gi,64,64,Na{29},Ta{29},Ca{29});
cH3bo1=new_arithme(cH3bi,64,64,Na{30},Ta{30},Ca{30});
%cV3
cV3ro1=new_arithme(cV3ri,64,64,Na{31},Ta{31}),Ca{31});
cV3go1=new_arithme(cV3gi,64,64,Na{32},Ta{32},Ca{32});
cV3bo1=new_arithme(cV3bi,64,64,Na{33},Ta{33},Ca{33});
%cD3
cD3ro1=new_arithme(cD3ri,64,64,Na{34},Ta{34},Ca{34});
cD3go1=new_arithme(cD3gi,64,64,Na{35},Ta{35},Ca{35});
cD3bo1=new_arithme(cD3bi,64,64,Na{36},Ta{36},Ca{36});
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
Ar3 = idwt2(cA3ro1,cH3ro1,cV3ro1,cD3ro1,'haar');
% Do the 2D DWT on the green component
Ag3 = idwt2(cA3go1,cH3go1,cV3go1,cD3go1,'haar');
% Do the 2D DWT on the blue component
Ab3 = idwt2(cA3bo1,cH3bo1,cV3bo1,cD3bo1,'haar');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
Ar2 = idwt2(cA2ro1,cH2ro1,cV2ro1,cD2ro1,'haar');
% Do the 2D DWT on the green component
Ag2 = idwt2(cA2go1,cH2go1,cV2go1,cD2go1,'haar');
% Do the 2D DWT on the blue component
Ab2 = idwt2(cA2bo1,cH2bo1,cV2bo1,cD2bo1,'haar');
```

```
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
Ar1 = idwt2(cA1ro1,cH1ro1,cV1ro1,cD1ro1,'haar');
% Do the 2D DWT on the green component
Ag1 = idwt2(cA1go1,cH1go1,cV1go1,cD1go1,'haar');
% Do the 2D DWT on the blue component
Ab1 = idwt2(cA1bo1,cH1bo1,cV1bo1,cD1bo1,'haar');
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
DeIm14(:,:,1)=Ar1;
DeIm14(:,:,2)=Ag1;
DeIm14(:,:,3)=Ab1;
figure(), imshow(uint8(DeIm14));title('reconstruct DWT  Lv1 ');
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
DeIm2(:,:,1)=Ar2;
DeIm2(:,:,2)=Ag2;
DeIm2(:,:,3)=Ab2;
figure(), imshow(uint8(DeIm2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
DeIm3(:,:,1)=Ar3;
DeIm3(:,:,2)=Ag3;
DeIm3(:,:,3)=Ab3;
figure(), imshow(uint8(DeIm3));title('reconstruct DWT  Lv3 ');
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(DeIm14),uint8(X))
toc
```

```
close all; clear all; clc;
tic
X = imread('peppers.png'); %X คือข้อมูลภาพดิจิตอลสีพริกไทย
figure(1);
imshow(X); title('original image'); % แสดงภาพต้นแบบสีพริกไทย
[sr,sc,pp]=size(X); %ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
%--------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 1 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA1r,cH1r,cV1r,cD1r] = dwt2(double(X(:,:,1)),'haar');
% Do the 2D DWT on the green component
[cA1g,cH1g,cV1g,cD1g] = dwt2(double(X(:,:,2)),'haar');
% Do the 2D DWT on the blue component
[cA1b,cH1b,cV1b,cD1b] = dwt2(double(X(:,:,3)), 'haar');
%--------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 2 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA2r,cH2r,cV2r,cD2r] = dwt2(double(cA1r(:,:,1)),'haar');
% Do the 2D DWT on the green component
[cA2g,cH2g,cV2g,cD2g] = dwt2(double(cA1g(:,:,1)),'haar');
% Do the 2D DWT on the blue component
[cA2b,cH2b,cV2b,cD2b] = dwt2(double(cA1b(:,:,1)),'haar');
%--------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 3 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA3r,cH3r,cV3r,cD3r] = dwt2(double(cA2r(:,:,1)),'haar');
```

```
% Do the 2D DWT on the green component
[cA3g,cH3g,cV3g,cD3g] = dwt2(double(cA2g(:,:,1)),'haar');
% Do the 2D DWT on the blue component
[cA3b,cH3b,cV3b,cD3b] = dwt2(double(cA2b(:,:,1)),'haar');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1ri=cA1r(:);
cA1gi=cA1g(:);
cA1bi=cA1b(:);

cH1ri=cH1r(:);
cH1gi=cH1g(:);
cH1bi=cH1b(:);

cV1ri=cV1r(:);
cV1gi=cV1g(:);
cV1bi=cV1b(:);

cD1ri=cD1r(:);
cD1gi=cD1g(:);
cD1bi=cD1b(:);
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2ri=cA2r(:);
cA2gi=cA2g(:);
cA2bi=cA2b(:);
cH2ri=cH2r(:);
cH2gi=cH2g(:);
cH2bi=cH2b(:);
```

```
cV2ri=cV2r(:);
cV2gi=cV2g(:);
cV2bi=cV2b(:);

cD2ri=cD2r(:);
cD2gi=cD2g(:);
cD2bi=cD2b(:);
%-----------------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3ri=cA3r(:);
cA3gi=cA3g(:);
cA3bi=cA3b(:);

cH3ri=cH3r(:);
cH3gi=cH3g(:);
cH3bi=cH3b(:);

cV3ri=cV3r(:);
cV3gi=cV3g(:);
cV3bi=cV3b(:);

cD3ri=cD3r(:);
cD3gi=cD3g(:);
cD3bi=cD3b(:);
```

```
%บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_haar_peppers_cA1ro.PM','bit_haar_peppers_cA1go.PM','bit_haar_peppers_cA1bo.PM'
'bit_haar_peppers_cH1ro.PM','bit_haar_peppers_cH1go.PM','bit_haar_peppers_cH1bo.PM'
'bit_haar_peppers_cV1ro.PM','bit_haar_peppers_cV1go.PM','bit_haar_peppers_cV1bo.PM'
'bit_haar_peppers_cD1ro.PM','bit_haar_peppers_cD1go.PM','bit_haar_peppers_cD1bo.PM'
'bit_haar_peppers_cA2ro.PM','bit_haar_peppers_cA2go.PM','bit_haar_peppers_cA2bo.PM'
'bit_haar_peppers_cH2ro.PM','bit_haar_peppers_cH2go.PM','bit_haar_peppers_cH2bo.PM'
'bit_haar_peppers_cV2ro.PM','bit_haar_peppers_cV2go.PM','bit_haar_peppers_cV2bo.PM'
'bit_haar_peppers_cD2ro.PM','bit_haar_peppers_cD2go.PM','bit_haar_peppers_cD2bo.PM'
'bit_haar_peppers_cA3ro.PM','bit_haar_peppers_cA3go.PM','bit_haar_peppers_cA3bo.PM'
'bit_haar_peppers_cH3ro.PM','bit_haar_peppers_cH3go.PM','bit_haar_peppers_cH3bo.PM'
'bit_haar_peppers_cV3ro.PM','bit_haar_peppers_cV3go.PM','bit_haar_peppers_cV3bo.PM'
'bit_haar_peppers_cD3ro.PM','bit_haar_peppers_cD3go.PM','bit_haar_peppers_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_haar_peppers_cA1ro.PM','bit_table_haar_peppers_cA1go.PM',
'bit_table_haar_peppers_cA1bo.PM'
'bit_table_haar_peppers_cH1ro.PM','bit_table_haar_peppers_cH1go.PM',
'bit_table_haar_peppers_cH1bo.PM'
'bit_table_haar_peppers_cV1ro.PM','bit_table_haar_peppers_cV1go.PM',
'bit_table_haar_peppers_cV1bo.PM'
'bit_table_haar_peppers_cD1ro.PM','bit_table_haar_peppers_cD1go.PM',
'bit_table_haar_peppers_cD1bo.PM'
'bit_table_haar_peppers_cA2ro.PM','bit_table_haar_peppers_cA2go.PM',
'bit_table_haar_peppers_cA2bo.PM'
'bit_table_haar_peppers_cH2ro.PM','bit_table_haar_peppers_cH2go.PM',
'bit_table_haar_peppers_cH2bo.PM'
'bit_table_haar_peppers_cV2ro.PM','bit_table_haar_peppers_cV2go.PM',
'bit_table_haar_peppers_cV2bo.PM'
```

'bit_table_haar_peppers_cD2ro.PM','bit_table_haar_peppers_cD2go.PM',

'bit_table_haar_peppers_cD2bo.PM'

'bit_table_haar_peppers_cA3ro.PM','bit_table_haar_peppers_cA3go.PM',

'bit_table_haar_peppers_cA3bo.PM'

'bit_table_haar_peppers_cH3ro.PM','bit_table_haar_peppers_cH3go.PM',

'bit_table_haar_peppers_cH3bo.PM'

'bit_table_haar_peppers_cV3ro.PM','bit_table_haar_peppers_cV3go.PM',

'bit_table_haar_peppers_cV3bo.PM'

'bit_table_haar_peppers_cD3ro.PM','bit_table_haar_peppers_cD3go.PM','bit_table_haar_peppers_cD3bo.PM'};

%----------

% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง

Ca={'bit_count_haar_peppers_cA1ro.PM','bit_count_haar_peppers_cA1go.PM',

'bit_count_haar_peppers_cA1bo.PM'

'bit_count_haar_peppers_cH1ro.PM','bit_count_haar_peppers_cH1go.PM',

'bit_count_haar_peppers_cH1bo.PM'

'bit_count_haar_peppers_cV1ro.PM','bit_count_haar_peppers_cV1go.PM','bit_count_haar_peppers_cV1bo.PM'

'bit_count_haar_peppers_cD1ro.PM','bit_count_haar_peppers_cD1go.PM',

'bit_count_haar_peppers_cD1bo.PM'

'bit_count_haar_peppers_cA2ro.PM','bit_count_haar_peppers_cA2go.PM',

'bit_count_haar_peppers_cA2bo.PM'

'bit_count_haar_peppers_cH2ro.PM','bit_count_haar_peppers_cH2go.PM',

'bit_count_haar_peppers_cH2bo.PM'

'bit_count_haar_peppers_cV2ro.PM','bit_count_haar_peppers_cV2go.PM',

'bit_count_haar_peppers_cV2bo.PM'

'bit_count_haar_peppers_cD2ro.PM','bit_count_haar_peppers_cD2go.PM',

'bit_count_haar_peppers_cD2bo.PM'

```
'bit_count_haar_peppers_cA3ro.PM','bit_count_haar_peppers_cA3go.PM',
'bit_count_haar_peppers_cA3bo.PM'
'bit_count_haar_peppers_cH3ro.PM','bit_count_haar_peppers_cH3go.PM',
'bit_count_haar_peppers_cH3bo.PM'
'bit_count_haar_peppers_cV3ro.PM','bit_count_haar_peppers_cV3go.PM',
'bit_count_haar_peppers_cV3bo.PM'
'bit_count_haar_peppers_cD3ro.PM','bit_count_haar_peppers_cD3go.PM',
'bit_count_haar_peppers_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
%cA1
cA1ro1=new_arithme(cA1ri,192,256,Na{1},Ta{1});
cA1go1=new_arithme(cA1gi,192,256,Na{2},Ta{2});
cA1bo1=new_arithme(cA1bi,192,256,Na{3},Ta{3});
%cH1
cH1ro1=new_arithme(cH1ri,192,256,Na{4},Ta{4});
cH1go1=new_arithme(cH1gi,192,256,Na{5},Ta{5});
cH1bo1=new_arithme(cH1bi,192,256,Na{6},Ta{6});
%cV1
cV1ro1=new_arithme(cV1ri,192,256,Na{7},Ta{7});
cV1go1=new_arithme(cV1gi,192,256,Na{8},Ta{8});
cV1bo1=new_arithme(cV1bi,192,256,Na{9},Ta{9});
%cD1
cD1ro1=new_arithme(cD1ri,192,256,Na{10},Ta{10});
cD1go1=new_arithme(cD1gi,192,256,Na{11},Ta{11});
cD1bo1=new_arithme(cD1bi,192,256,Na{12},Ta{12});
```

```
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
%cA2
cA2ro1=new_arithme(cA2ri,96,128,Na{13},Ta{13});
cA2go1=new_arithme(cA2gi,96,128,Na{14},Ta{14});
cA2bo1=new_arithme(cA2bi,96,128,Na{15},Ta{15});
%cH2
cH2ro1=new_arithme(cH2ri,96,128,Na{16},Ta{16});
cH2go1=new_arithme(cH2gi,96,128,Na{17},Ta{17});
cH2bo1=new_arithme(cH2bi,96,128,Na{18},Ta{18});
%cV2
cV2ro1=new_arithme(cV2ri,96,128,Na{19},Ta{19});
cV2go1=new_arithme(cV2gi,96,128,Na{20},Ta{20});
cV2bo1=new_arithme(cV2bi,96,128,Na{21},Ta{21});
%cD2
cD2ro1=new_arithme(cD2ri,96,128,Na{22},Ta{22});
cD2go1=new_arithme(cD2gi,96,128,Na{23},Ta{23});
cD2bo1=new_arithme(cD2bi,96,128,Na{24},Ta{24});
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
%cA3
cA3ro1=new_arithme(cA3ri,48,64,Na{25},Ta{25});
cA3go1=new_arithme(cA3gi,48,64,Na{26},Ta{26});
cA3bo1=new_arithme(cA3bi,48,64,Na{27},Ta{27});
%cH3
cH3ro1=new_arithme(cH3ri,48,64,Na{28},Ta{28});
cH3go1=new_arithme(cH3gi,48,64,Na{29},Ta{29});
cH3bo1=new_arithme(cH3bi,48,64,Na{30},Ta{30});
```

```
%cV3
cV3ro1=new_arithme(cV3ri,48,64,Na{31},Ta{31});
cV3go1=new_arithme(cV3gi,48,64,Na{32},Ta{32});
cV3bo1=new_arithme(cV3bi,48,64,Na{33},Ta{33});
%cD3
cD3ro1=new_arithme(cD3ri,48,64,Na{34},Ta{34});
cD3go1=new_arithme(cD3gi,48,64,Na{35},Ta{35});
cD3bo1=new_arithme(cD3bi,48,64,Na{36},Ta{36});
%-------------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มระดับ 3 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
Ar3 = idwt2(cA3ro1,cH3ro1,cV3ro1,cD3ro1,'haar');
% Do the 2D DWT on the green component
Ag3 = idwt2(cA3go1,cH3go1,cV3go1,cD3go1,'haar');
% Do the 2D DWT on the blue component
Ab3 = idwt2(cA3bo1,cH3bo1,cV3bo1,cD3bo1,'haar');
%-------------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มระดับ 2 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
Ar2 = idwt2(cA2ro1,cH2ro1,cV2ro1,cD2ro1,'haar');
% Do the 2D DWT on the green component
Ag2 = idwt2(cA2go1,cH2go1,cV2go1,cD2go1,'haar');
% Do the 2D DWT on the blue component
Ab2 = idwt2(cA2bo1,cH2bo1,cV2bo1,cD2bo1,'haar');
%-------------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มระดับ 1 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
Ar1 = idwt2(cA1ro1,cH1ro1,cV1ro1,cD1ro1,'haar');
```

```
% Do the 2D DWT on the green component
Ag1 = idwt2(cA1go1,cH1go1,cV1go1,cD1go1,'haar');
% Do the 2D DWT on the blue component
Ab1 = idwt2(cA1bo1,cH1bo1,cV1bo1,cD1bo1,'haar');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
DeIm14(:,:,1)=Ar1;
DeIm14(:,:,2)=Ag1;
DeIm14(:,:,3)=Ab1;
figure()
imshow(uint8(DeIm14));title('reconstruct DWT  Lv1 ');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
DeIm2(:,:,1)=Ar2;
DeIm2(:,:,2)=Ag2;
DeIm2(:,:,3)=Ab2;
figure()
imshow(uint8(DeIm2));title('reconstruct DWT  Lv2 ');
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
DeIm3(:,:,1)=Ar3;
DeIm3(:,:,2)=Ag3;
DeIm3(:,:,3)=Ab3;
figure()
imshow(uint8(DeIm3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(DeIm14),uint8(X))
toc
```

```
close all; clear all; clc;
tic %ฟังก์ชั่นจับเวลา
X= imread('mandril.tif'); %X คือข้อมูลภาพดิจิตอลสีแมนดริล
figure(1);
imshow(X); title('original image'); % แสดงภาพต้นแบบสีแมนดริล
[sr,sc,pp]=size(X); %ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
%--------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 1 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA1r,cH1r,cV1r,cD1r] = dwt2(double(X(:,:,1)),'haar');
% Do the 2D DWT on the green component
[cA1g,cH1g,cV1g,cD1g] = dwt2(double(X(:,:,2)),'haar');
% Do the 2D DWT on the blue component
[cA1b,cH1b,cV1b,cD1b] = dwt2(double(X(:,:,3)), 'haar');
%--------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 2 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA2r,cH2r,cV2r,cD2r] = dwt2(double(cA1r(:,:,1)),'haar');
% Do the 2D DWT on the green component
[cA2g,cH2g,cV2g,cD2g] = dwt2(double(cA1g(:,:,1)),'haar');
% Do the 2D DWT on the blue component
[cA2b,cH2b,cV2b,cD2b] = dwt2(double(cA1b(:,:,1)),'haar');
%--------------------------------------------------------------------------------------------
```

```
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์มระดับ 3 ชนิดตัวกรอง haar
% Do the 2D DWT on the red component
[cA3r,cH3r,cV3r,cD3r] = dwt2(double(cA2r(:,:,1)),'haar');

% Do the 2D DWT on the green component
[cA3g,cH3g,cV3g,cD3g] = dwt2(double(cA2g(:,:,1)),'haar');

% Do the 2D DWT on the blue component
[cA3b,cH3b,cV3b,cD3b] = dwt2(double(cA2b(:,:,1)),'haar');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1ri=cA1r(:);
cA1gi=cA1g(:);
cA1bi=cA1b(:);

cH1ri=cH1r(:);
cH1gi=cH1g(:);
cH1bi=cH1b(:);

cV1ri=cV1r(:);
cV1gi=cV1g(:);
cV1bi=cV1b(:);

cD1ri=cD1r(:);
cD1gi=cD1g(:);
cD1bi=cD1b(:);
%-------------------------------------------------------------------------------------------
```

```
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2ri=cA2r(:);
cA2gi=cA2g(:);
cA2bi=cA2b(:);

cH2ri=cH2r(:);
cH2gi=cH2g(:);
cH2bi=cH2b(:);

cV2ri=cV2r(:);
cV2gi=cV2g(:);
cV2bi=cV2b(:);

cD2ri=cD2r(:);
cD2gi=cD2g(:);
cD2bi=cD2b(:);
%-------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3ri=cA3r(:);
cA3gi=cA3g(:);
cA3bi=cA3b(:);

cH3ri=cH3r(:);
cH3gi=cH3g(:);
cH3bi=cH3b(:);
```

```
cV3ri=cV3r(:);
cV3gi=cV3g(:);
cV3bi=cV3b(:);

cD3ri=cD3r(:);
cD3gi=cD3g(:);
cD3bi=cD3b(:);
%-------------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
Na={'bit_haar_mandril_cA1ro.PM','bit_haar_mandril_cA1go.PM','bit_haar_mandril_cA1bo.PM'
'bit_haar_mandril_cH1ro.PM','bit_haar_mandril_cH1go.PM','bit_haar_mandril_cH1bo.PM'
'bit_haar_mandril_cV1ro.PM','bit_haar_mandril_cV1go.PM','bit_haar_mandril_cV1bo.PM'
'bit_haar_mandril_cD1ro.PM','bit_haar_mandril_cD1go.PM','bit_haar_mandril_cD1bo.PM'
'bit_haar_mandril_cA2ro.PM','bit_haar_mandril_cA2go.PM','bit_haar_mandril_cA2bo.PM'
'bit_haar_mandril_cH2ro.PM','bit_haar_mandril_cH2go.PM','bit_haar_mandril_cH2bo.PM'
'bit_haar_mandril_cV2ro.PM','bit_haar_mandril_cV2go.PM','bit_haar_mandril_cV2bo.PM'
'bit_haar_mandril_cD2ro.PM','bit_haar_mandril_cD2go.PM','bit_haar_mandril_cD2bo.PM'
'bit_haar_mandril_cA3ro.PM','bit_haar_mandril_cA3go.PM','bit_haar_mandril_cA3bo.PM'
'bit_haar_mandril_cH3ro.PM','bit_haar_mandril_cH3go.PM','bit_haar_mandril_cH3bo.PM'
'bit_haar_mandril_cV3ro.PM','bit_haar_mandril_cV3go.PM','bit_haar_mandril_cV3bo.PM'
'bit_haar_mandril_cD3ro.PM','bit_haar_mandril_cD3go.PM','bit_haar_mandril_cD3bo.PM'};
%-------------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_haar_mandril_cA1ro.PM','bit_table_haar_mandril_cA1go.PM',
'bit_table_haar_mandril_cA1bo.PM'
'bit_table_haar_mandril_cH1ro.PM','bit_table_haar_mandril_cH1go.PM',
'bit_table_haar_mandril_cH1bo.PM'
'bit_table_haar_mandril_cV1ro.PM','bit_table_haar_mandril_cV1go.PM',
'bit_table_haar_mandril_cV1bo.PM'
```

```
'bit_table_haar_mandril_cD1ro.PM','bit_table_haar_mandril_cD1go.PM',
'bit_table_haar_mandril_cD1bo.PM'
'bit_table_haar_mandril_cA2ro.PM','bit_table_haar_mandril_cA2go.PM',
'bit_table_haar_mandril_cA2bo.PM'
'bit_table_haar_mandril_cH2ro.PM','bit_table_haar_mandril_cH2go.PM',
'bit_table_haar_mandril_cH2bo.PM'
'bit_table_haar_mandril_cV2ro.PM','bit_table_haar_mandril_cV2go.PM',
'bit_table_haar_mandril_cV2bo.PM'
'bit_table_haar_mandril_cD2ro.PM','bit_table_haar_mandril_cD2go.PM',
'bit_table_haar_mandril_cD2bo.PM'
'bit_table_haar_mandril_cA3ro.PM','bit_table_haar_mandril_cA3go.PM',
'bit_table_haar_mandril_cA3bo.PM'
'bit_table_haar_mandril_cH3ro.PM','bit_table_haar_mandril_cH3go.PM',
'bit_table_haar_mandril_cH3bo.PM'
'bit_table_haar_mandril_cV3ro.PM','bit_table_haar_mandril_cV3go.PM',
'bit_table_haar_mandril_cV3bo.PM'
'bit_table_haar_mandril_cD3ro.PM','bit_table_haar_mandril_cD3go.PM',
'bit_table_haar_mandril_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_haar_mandril_cA1ro.PM','bit_count_haar_mandril_cA1go.PM',
'bit_count_haar_mandril_cA1bo.PM'
'bit_count_haar_mandril_cH1ro.PM','bit_count_haar_mandril_cH1go.PM',
'bit_count_haar_mandril_cH1bo.PM'
'bit_count_haar_mandril_cV1ro.PM','bit_count_haar_mandril_cV1go.PM',
'bit_count_haar_mandril_cV1bo.PM'
```

```
'bit_count_haar_mandril_cD1ro.PM','bit_count_haar_mandril_cD1go.PM',
'bit_count_haar_mandril_cD1bo.PM'
'bit_count_haar_mandril_cA2ro.PM','bit_count_haar_mandril_cA2go.PM',
'bit_count_haar_mandril_cA2bo.PM'
'bit_count_haar_mandril_cH2ro.PM','bit_count_haar_mandril_cH2go.PM',
'bit_count_haar_mandril_cH2bo.PM'
'bit_count_haar_mandril_cV2ro.PM','bit_count_haar_mandril_cV2go.PM',
'bit_count_haar_mandril_cV2bo.PM'
'bit_count_haar_mandril_cD2ro.PM','bit_count_haar_mandril_cD2go.PM',
'bit_count_haar_mandril_cD2bo.PM'
'bit_count_haar_mandril_cA3ro.PM','bit_count_haar_mandril_cA3go.PM',
'bit_count_haar_mandril_cA3bo.PM'
'bit_count_haar_mandril_cH3ro.PM','bit_count_haar_mandril_cH3go.PM',
'bit_count_haar_mandril_cH3bo.PM'
'bit_count_haar_mandril_cV3ro.PM','bit_count_haar_mandril_cV3go.PM',
'bit_count_haar_mandril_cV3bo.PM'
'bit_count_haar_mandril_cD3ro.PM','bit_count_haar_mandril_cD3go.PM',
'bit_count_haar_mandril_cD3bo.PM'};
%-------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง haar
%cA1
cA1ro1=new_arithme(cA1ri,256,256,Na{1},Ta{1},Ca{1});
cA1go1=new_arithme(cA1gi,256,256,Na{2},Ta{2},Ca{2});
cA1bo1=new_arithme(cA1bi,256,256,Na{3},Ta{3},Ca{3});
%cH1
cH1ro1=new_arithme(cH1ri,256,256,Na{4},Ta{4},Ca{4});
cH1go1=new_arithme(cH1gi,256,256,Na{5},Ta{5},Ca{5});
cH1bo1=new_arithme(cH1bi,256,256,Na{6},Ta{6},Ca{6});
```

```
%cV1
cV1ro1=new_arithme(cV1ri,256,256,Na{7},Ta{7},Ca{7});
cV1go1=new_arithme(cV1gi,256,256,Na{8},Ta{8},Ca{8});
cV1bo1=new_arithme(cV1bi,256,256,Na{9},Ta{9},Ca{9});
%cD1
cD1ro1=new_arithme(cD1ri,256,256,Na{10},Ta{10},Ca{10});
cD1go1=new_arithme(cD1gi,256,256,Na{11},Ta{11},Ca{11});
cD1bo1=new_arithme(cD1bi,256,256,Na{12},Ta{12},Ca{12});
%-------------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง haar
%cA2
cA2ro1=new_arithme(cA2ri,128,128,Na{13},Ta{13},Ca{13});
cA2go1=new_arithme(cA2gi,128,128,Na{14},Ta{14},Ca{14});
cA2bo1=new_arithme(cA2bi,128,128,Na{15},Ta{15},Ca{15});
%cH2
cH2ro1=new_arithme(cH2ri,128,128,Na{16},Ta{16},Ca{16});
cH2go1=new_arithme(cH2gi,128,128,Na{17},Ta{17},Ca{17});
cH2bo1=new_arithme(cH2bi,128,128,Na{18},Ta{18},Ca{18});
%cV2
cV2ro1=new_arithme(cV2ri,128,128,Na{19},Ta{19},Ca{19});
cV2go1=new_arithme(cV2gi,128,128,Na{20},Ta{20},Ca{20});
cV2bo1=new_arithme(cV2bi,128,128,Na{21},Ta{21},Ca{21});
%cD2
cD2ro1=new_arithme(cD2ri,128,128,Na{22},Ta{22},Ca{22});
cD2go1=new_arithme(cD2gi,128,128,Na{23},Ta{23},Ca{23});
cD2bo1=new_arithme(cD2bi,128,128,Na{24},Ta{24},Ca{24});
%-------------------------------------------------------------------------------------------------
```

```
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง haar
%cA3
cA3ro1=new_arithme(cA3ri,64,64 ,Na{25},Ta{25},Ca{25});
cA3go1=new_arithme(cA3gi,64,64 ,Na{26},Ta{26},Ca{26});
cA3bo1=new_arithme(cA3bi,64,64 ,Na{27},Ta{27},Ca{27});
%cH3
cH3ro1=new_arithme(cH3ri,64,64 ,Na{28},Ta{28},Ca{28});
cH3go1=new_arithme(cH3gi,64,64 ,Na{29},Ta{29},Ca{29});
cH3bo1=new_arithme(cH3bi,64,64 ,Na{30},Ta{30},Ca{30});
%cV3
cV3ro1=new_arithme(cV3ri,64,64 ,Na{31},Ta{31},Ca{31});
cV3go1=new_arithme(cV3gi,64,64 ,Na{32},Ta{32},Ca{32});
cV3bo1=new_arithme(cV3bi,64,64 ,Na{33},Ta{33},Ca{33});
%cD3
cD3ro1=new_arithme(cD3ri,64,64 ,Na{34},Ta{34},Ca{34});
cD3go1=new_arithme(cD3gi,64,64 ,Na{35},Ta{35},Ca{35});
cD3bo1=new_arithme(cD3bi,64,64 ,Na{36},Ta{36},Ca{36});
%-----------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
% Do the 2D DWT on the red component
Ar3 = idwt2(cA3ro1,cH3ro1,cV3ro1,cD3ro1,'haar');
% Do the 2D DWT on the green component
Ag3 = idwt2(cA3go1,cH3go1,cV3go1,cD3go1,'haar');
% Do the 2D DWT on the blue component
Ab3 = idwt2(cA3bo1,cH3bo1,cV3bo1,cD3bo1,'haar');
%-----------------------------------------------------------------------------------------
```

```
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
% Do the 2D DWT on the red component
Ar2 = idwt2(cA2ro1,cH2ro1,cV2ro1,cD2ro1,'haar');
% Do the 2D DWT on the green component
Ag2 = idwt2(cA2go1,cH2go1,cV2go1,cD2go1,'haar');
% Do the 2D DWT on the blue component
Ab2 = idwt2(cA2bo1,cH2bo1,cV2bo1,cD2bo1,'haar');
%-------------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
% Do the 2D DWT on the red component
Ar1 = idwt2(cA1ro1,cH1ro1,cV1ro1,cD1ro1,'haar');
% Do the 2D DWT on the green component
Ag1 = idwt2(cA1go1,cH1go1,cV1go1,cD1go1,'haar');
% Do the 2D DWT on the blue component
Ab1 = idwt2(cA1bo1,cH1bo1,cV1bo1,cD1bo1,'haar');
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 1
DeIm14(:,:,1)=Ar1;
DeIm14(:,:,2)=Ag1;
DeIm14(:,:,3)=Ab1;
figure()
imshow(uint8(DeIm14));title('reconstruct DWT  Lv1 ');
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 2
DeIm2(:,:,1)=Ar2;
DeIm2(:,:,2)=Ag2;
DeIm2(:,:,3)=Ab2;
figure()
imshow(uint8(DeIm2));title('reconstruct DWT  Lv2 ');
```

```
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง haar ระดับที่ 3
DeIm3(:,:,1)=Ar3;
DeIm3(:,:,2)=Ag3;
DeIm3(:,:,3)=Ab3;
figure()
imshow(uint8(DeIm3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(DeIm14),uint8(X))
toc
```

ภาคผนวก ค

โปรแกรมการบีบอัดภาพดิจิตอล

Wavelet Transform(Db 4)+Arithmetic Coding

```
close all, clear all clc
tic % ฟังก์ชั่นจับเวลา
X=imread('cameraman.tif');%X คือข้อมูลภาพดิจิตอลระดับเทามนุษย์ถ่ายรูป
figure(1),imshow(X);% แสดงภาพต้นแบบระดับเทามนุษย์ถ่ายรูป
[M,N]=size(X);;%ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
% แปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4
[cA1,cH1,cV1,cD1] = dwt2(X,'db4');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4
[cA2,cH2,cV2,cD2] = dwt2(cA1,'db4');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4
[cA3,cH3,cV3,cD3] = dwt2(cA2,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1i=cA1(:);
cH1i=cH1(:);
cV1i=cV1(:);
cD1i=cD1(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2i=cA2(:);
cH2i=cH2(:);
cV2i=cV2(:);
cD2i=cD2(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3i=cA3(:);
cH3i=cH3(:);
cV3i=cV3(:);
cD3i=cD3(:);
```

```
% บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_db4_cameraman_cA1o.PM','bit_db4_cameraman_cH1o.PM',
'bit_db4_cameraman_cV1o.PM','bit_db4_cameraman_cD1o.PM'
'bit_db4_cameraman_cA2o.PM','bit_db4_cameraman_cH2o.PM','bit_db4_cameraman_cV2o.PM'
,
'bit_db4_cameraman_cD2o.PM'
'bit_db4_cameraman_cA3o.PM','bit_db4_cameraman_cH3o.PM','bit_db4_cameraman_cV3o.PM'
,
'bit_db4_cameraman_cD3o.PM'};
%-----------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_db4_cameraman_cA1o.PM','bit_table_db4_cameraman_cH1o.PM','bit_table_db4
_cameraman_cV1o.PM','bit_table_db4_cameraman_cD1o.PM'
'bit_table_db4_cameraman_cA2o.PM','bit_table_db4_cameraman_cH2o.PM',
'bit_table_db4_cameraman_cV2o.PM','bit_table_db4_cameraman_cD2o.PM'
'bit_table_db4_cameraman_cA3o.PM','bit_table_db4_cameraman_cH3o.PM',
'bit_table_db4_cameraman_cV3o.PM','bit_table_db4_cameraman_cD3o.PM'};

% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_db4_cameraman_cA1o.PM','bit_count_db4_cameraman_cH1o.PM',
'bit_count_db4_cameraman_cV1o.PM','bit_count_db4_cameraman_cD1o.PM'
'bit_count_db4_cameraman_cA2o.PM','bit_count_db4_cameraman_cH2o.PM',
'bit_count_db4_cameraman_cV2o.PM','bit_count_db4_cameraman_cD2o.PM'
'bit_count_db4_cameraman_cA3o.PM','bit_count_db4_cameraman_cH3o.PM',
'bit_count_db4_cameraman_cV3o.PM','bit_count_db4_cameraman_cD3o.PM'};
%-----------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง db4
cA1o1=new_arithme(cA1i,131,131,Na{1},Ta{1},Ca{1});
cH1o1=new_arithme(cH1i,131,131,Na{2},Ta{2},Ca{2});
cV1o1=new_arithme(cV1i,131,131,Na{3},Ta{3},Ca{3});
cD1o1=new_arithme(cD1i,131,131,Na{4},Ta{4},Ca{4});
```

```
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง db4
cA2o1=new_arithme(cA2i,69,69,Na{5},Ta{5},Ca{5});
cH2o1=new_arithme(cH2i,69,69,Na{6},Ta{6},Ca{6});
cV2o1=new_arithme(cV2i,69,69,Na{7},Ta{7},Ca{7});
cD2o1=new_arithme(cD2i,69,69,Na{8},Ta{8},Ca{8});
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง db4
cA3o1=new_arithme(cA3i,38,38,Na{9},Ta{9},Ca{9});
cH3o1=new_arithme(cH3i,38,38,Na{10},Ta{10},Ca{10});
cV3o1=new_arithme(cV3i,38,38,Na{11},Ta{11},Ca{11});
cD3o1=new_arithme(cD3i,38,38,Na{12},Ta{12},Ca{12});
%-------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
A3 = idwt2(cA3o1,cH3o1,cV3o1,cD3o1,'db4');
%-------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 2
A2 = idwt2(cA2o1,cH2o1,cV2o1,cD2o1,'db4');
%-------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
A1 = idwt2(cA1o1,cH1o1,cV1o1,cD1o1,'db4');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
figure()
imshow(uint8(A1));title('reconstruct DWT  Lv1 ');
```

```
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
figure()
imshow(uint8(A2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
figure()
imshow(uint8(A3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(A1),uint8(X))
toc
```

```
close all, clear all, clc
tic % ฟังก์ชั่นจับเวลา
X=imread('barbara.png');%X คือข้อมูลภาพดิจิตอลระดับเทาบาร์บารา
figure(1),imshow(X);% แสดงภาพต้นแบบระดับเทาบาร์บารา
[M,N]=size(X);%ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
% แปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4
[cA1,cH1,cV1,cD1] = dwt2(X,'db4');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่2 ชนิดตัวกรอง db4
[cA2,cH2,cV2,cD2] = dwt2(cA1,'db4');
% แปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4
[cA3,cH3,cV3,cD3] = dwt2(cA2,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1i=cA1(:);
cH1i=cH1(:);
cV1i=cV1(:);
cD1i=cD1(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2i=cA2(:);
cH2i=cH2(:);
cV2i=cV2(:);
cD2i=cD2(:);
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3i=cA3(:);
cH3i=cH3(:);
cV3i=cV3(:);
cD3i=cD3(:);
```

```
% บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_db4_barbara_cA1o.PM','bit_db4_barbara_cH1o.PM','bit_db4_barbara_cV1o.PM',
'bit_db4_barbara_cD1o.PM'
'bit_db4_barbara_cA2o.PM','bit_db4_barbara_cH2o.PM','bit_db4_barbara_cV2o.PM',
'bit_db4_barbara_cD2o.PM'
'bit_db4_barbara_cA3o.PM','bit_db4_barbara_cH3o.PM','bit_db4_barbara_cV3o.PM',
'bit_db4_barbara_cD3o.PM'};
%-------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_db4_barbara_cA1o.PM','bit_table_db4_barbara_cH1o.PM',
'bit_table_db4_barbara_cV1o.PM','bit_table_db4_barbara_cD1o.PM'
'bit_table_db4_barbara_cA2o.PM','bit_table_db4_barbara_cH2o.PM',
'bit_table_db4_barbara_cV2o.PM','bit_table_db4_barbaracD2o.PM'
'bit_table_db4_barbara_cA3o.PM','bit_table_db4_barbara_cH3o.PM',
'bit_table_db4_barbara_cV3o.PM','bit_table_db4_barbara_cD3o.PM'};
%-------------------------------------------------------------------------------------------
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_db4_barbara_cA1o.PM','bit_count_db4_barbara_cH1o.PM',
'bit_count_db4_barbara_cV1o.PM','bit_count_db4_barbara_cD1o.PM'
'bit_count_db4_barbara_cA2o.PM','bit_count_db4_barbara_cH2o.PM',
'bit_count_db4_barbara_cV2o.PM','bit_count_db4_barbaracD2o.PM'
'bit_count_db4_barbara_cA3o.PM','bit_count_db4_barbara_cH3o.PM','bit_count_db4_barbara_c
V3o.PM','bit_count_db4_barbara_cD3o.PM'};
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 1 ชนิดตัวกรอง db4
cA1o1=new_arithme(cA1i,259,259,Na{1},Ta{1},Ca{1});
cH1o1=new_arithme(cH1i,259,259,Na{2},Ta{2},Ca{2});
cV1o1=new_arithme(cV1i,259,259,Na{3},Ta{3},Ca{3});
cD1o1=new_arithme(cD1i,259,259,Na{4},Ta{4},Ca{4});
```

```
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 2 ชนิดตัวกรอง db4
cA2o1=new_arithme(cA2i,133,133,Na{5},Ta{5},Ca{5});
cH2o1=new_arithme(cH2i,133,133,Na{6},Ta{6},Ca{6});
cV2o1=new_arithme(cV2i,133,133,Na{7},Ta{7},Ca{7});
cD2o1=new_arithme(cD2i,133,133,Na{8},Ta{8},Ca{8});
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของ ดิสครีทเวฟเล็ตทรานฟอร์มระดับที่ 3 ชนิดตัวกรอง db4
cA3o1=new_arithme(cA3i,70,70,Na{9},Ta{9},Ca{9});
cH3o1=new_arithme(cH3i,70,70,Na{10},Ta{10},Ca{10});
cV3o1=new_arithme(cV3i,70,70,Na{11},Ta{11},Ca{11});
cD3o1=new_arithme(cD3i,70,70,Na{12},Ta{12},Ca{12});
%-------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
A3 = idwt2(cA3o1,cH3o1,cV3o1,cD3o1,'db4');
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 2
A2 = idwt2(cA2o1,cH2o1,cV2o1,cD2o1,'db4');
%-------------------------------------------------------------------------------------------
%การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
A1 = idwt2(cA1o1,cH1o1,cV1o1,cD1o1,'db4');
%-------------------------------------------------------------------------------------------
```

```
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
figure()
imshow(uint8(A1));title('reconstruct DWT  Lv1 ');
%-------------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 2
figure()
imshow(uint8(A2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
figure()
imshow(uint8(A3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(A1),uint8(X))
toc
```

```
close all, clear all, clc
tic % ฟังก์ชั่นจับเวลา
X = imread('lena512color.tiff');  %X คือข้อมูลภาพดิจิตอลสีลีน่า
[sr,sc,pp]=size(X);%ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
figure(1);
imshow(X); title('original image');% แสดงภาพต้นแบบสีลีน่า
%-------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
%-------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA1r,cH1r,cV1r,cD1r] = dwt2(double(X(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA1g,cH1g,cV1g,cD1g] = dwt2(double(X(:,:,2)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA1b,cH1b,cV1b,cD1b] = dwt2(double(X(:,:,3)), 'db4');
%-------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA2r,cH2r,cV2r,cD2r] = dwt2(double(cA1r(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA2g,cH2g,cV2g,cD2g] = dwt2(double(cA1g(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA2b,cH2b,cV2b,cD2b] = dwt2(double(cA1b(:,:,1)),'db4');
%-------------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA3r,cH3r,cV3r,cD3r] = dwt2(double(cA2r(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA3g,cH3g,cV3g,cD3g] = dwt2(double(cA2g(:,:,1)),'db4');
```

```
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA3b,cH3b,cV3b,cD3b] = dwt2(double(cA2b(:,:,1)),'db4');
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1ri=cA1r(:);
cA1gi=cA1g(:);
cA1bi=cA1b(:);

cH1ri=cH1r(:);
cH1gi=cH1g(:);
cH1bi=cH1b(:);

cV1ri=cV1r(:);
cV1gi=cV1g(:);
cV1bi=cV1b(:);

cD1ri=cD1r(:);
cD1gi=cD1g(:);
cD1bi=cD1b(:);
%-------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2ri=cA2r(:);
cA2gi=cA2g(:);
cA2bi=cA2b(:);

cH2ri=cH2r(:);
cH2gi=cH2g(:);
cH2bi=cH2b(:);
```

```
cV2ri=cV2r(:);
cV2gi=cV2g(:);
cV2bi=cV2b(:);

cD2ri=cD2r(:);
cD2gi=cD2g(:);
cD2bi=cD2b(:);
%-----------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3ri=cA3r(:);
cA3gi=cA3g(:);
cA3bi=cA3b(:);

cH3ri=cH3r(:);
cH3gi=cH3g(:);
cH3bi=cH3b(:);

cV3ri=cV3r(:);
cV3gi=cV3g(:);
cV3bi=cV3b(:);

cD3ri=cD3r(:);
cD3gi=cD3g(:);
cD3bi=cD3b(:);
%-----------------------------------------------------------------------------------------------
```

```
% บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_db4__lena_cA1ro.PM','bit_db4__lena_cA1go.PM','bit_db4__lena_cA1bo.PM'
   'bit_db4__lena_cH1ro.PM','bit_db4__lena_cH1go.PM','bit_db4__lena_cH1bo.PM'
   'bit_db4_lena_cV1ro.PM','bit_db4_lena_cV1go.PM','bit_db4_lena_cV1bo.PM'
   'bit_db4_lena_cD1ro.PM','bit_db4_lena_cD1go.PM','bit_db4_lena_cD1bo.PM'
   'bit_db4_lena_cA2ro.PM','bit_db4_lena_cA2go.PM','bit_db4_lena_cA2bo.PM'
   'bit_db4_lena_cH2ro.PM','bit_db4_lena_cH2go.PM','bit_db4_lena_cH2bo.PM'
   'bit_db4_lena_cV2ro.PM','bit_db4_lena_cV2go.PM','bit_db4_lena_cV2bo.PM'
   'bit_db4_lena_cD2ro.PM','bit_db4_lena_cD2go.PM','bit_db4_lena_cD2bo.PM'
   'bit_db4_lena_cA3ro.PM','bit_db4_lena_cA3go.PM','bit_db4_lena_cA3bo.PM'
   'bit_db4_lena_cH3ro.PM','bit_db4_lena_cH3go.PM','bit_db4_lena_cH3bo.PM'
   'bit_db4_lena_cV3ro.PM','bit_db4_lena_cV3go.PM','bit_db4_lena_cV3bo.PM'
   'bit_db4_lena_cD3ro.PM','bit_db4_lena_cD3go.PM','bit_db4_lena_cD3bo.PM'};
%----------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_db4_lena_cA1ro.PM','bit_table_db4_lena_cA1go.PM',
'bit_table_db4_lena_cA1bo.PM'
'bit_table_db4_lena_cH1ro.PM','bit_table_db4_lena_cH1go.PM','bit_table_db4_lena_cH1bo.PM'
'bit_table_db4_lena_cV1ro.PM','bit_table_db4_lena_cV1go.PM','bit_table_db4_lena_cV1bo.PM'
'bit_table_db4_lena_cD1ro.PM','bit_table_db4_lena_cD1go.PM','bit_table_db4_lena_cD1bo.PM'
'bit_table_db4_lena_cA2ro.PM','bit_table_db4_lena_cA2go.PM','bit_table_db4_lena_cA2bo.PM'
'bit_table_db4_lena_cH2ro.PM','bit_table_db4_lena_cH2go.PM','bit_table_db4_lena_cH2bo.PM'
'bit_table_db4_lena_cV2ro.PM','bit_table_db4_lena_cV2go.PM','bit_table_db4_lena_cV2bo.PM'
'bit_table_db4_lena_cD2ro.PM','bit_table_db4_lena_cD2go.PM','bit_table_db4_lena_cD2bo.PM'
'bit_table_db4_lena_cA3ro.PM','bit_table_db4_lena_cA3go.PM','bit_table_db4_lena_cA3bo.PM'
'bit_table_db4_lena_cH3ro.PM','bit_table_db4_lena_cH3go.PM','bit_table_db4_lena_cH3bo.PM'
'bit_table_db4_lena_cV3ro.PM','bit_table_db4_lena_cV3go.PM','bit_table_db4_lena_cV3bo.PM'
'bit_table_db4_lena_cD3ro.PM','bit_table_db4_lena_cD3go.PM','bit_table_db4_lena_cD3bo.PM'
};
```

```
%-------------------------------------------------------------------------------
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_db4_lena_cA1ro.PM','bit_count_db4_lena_cA1go.PM',
'bit_count_db4_lena_cA1bo.PM'
'bit_count_db4_lena_cH1ro.PM','bit_count_db4_lena_cH1go.PM','bit_count_db4_lena_cH1bo.P
M'
'bit_count_db4_lena_cV1ro.PM','bit_count_db4_lena_cV1go.PM','bit_count_db4_lena_cV1bo.P
M'
'bit_count_db4_lena_cD1ro.PM','bit_count_db4_lena_cD1go.PM','bit_count_db4_lena_cD1bo.P
M'
'bit_count_db4_lena_cA2ro.PM','bit_count_db4_lena_cA2go.PM','bit_count_db4_lena_cA2bo.P
M'
'bit_count_db4_lena_cH2ro.PM','bit_count_db4_lena_cH2go.PM','bit_count_db4_lena_cH2bo.P
M'
'bit_count_db4_lena_cV2ro.PM','bit_count_db4_lena_cV2go.PM','bit_count_db4_lena_cV2bo.P
M'
'bit_count_db4_lena_cD2ro.PM','bit_count_db4_lena_cD2go.PM','bit_count_db4_lena_cD2bo.P
M'
'bit_count_db4_lena_cA3ro.PM','bit_count_db4_lena_cA3go.PM','bit_count_db4_lena_cA3bo.P
M'
'bit_count_db4_lena_cH3ro.PM','bit_count_db4_lena_cH3go.PM','bit_count_db4_lena_cH3bo.P
M'
'bit_count_db4_lena_cV3ro.PM','bit_count_db4_lena_cV3go.PM','bit_count_db4_lena_cV3bo.P
M'
'bit_count_db4_lena_cD3ro.PM','bit_count_db4_lena_cD3go.PM','bit_count_db4_lena_cD3bo.P
M'};
%-------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิด
ตัวกรอง db4
%cA1
cA1ro1=new_arithme(cA1ri,259,259,Na{1},Ta{1},Ca{1});
cA1go1=new_arithme(cA1gi,259,259,Na{2},Ta{2},Ca{2});
```

```
cA1bo1=new_arithme(cA1bi,259,259,Na{3},Ta{3},Ca{3});
%cH1
cH1ro1=new_arithme(cH1ri,259,259,Na{4},Ta{4},Ca{4});
cH1go1=new_arithme(cH1gi,259,259,Na{5},Ta{5},Ca{5});
cH1bo1=new_arithme(cH1bi,259,259,Na{6},Ta{6},Ca{6});
```

```
%cV1
cV1ro1=new_arithme(cV1ri,259,259,Na{7},Ta{7},Ca{7});
cV1go1=new_arithme(cV1gi,259,259,Na{8},Ta{8},Ca{8});
cV1bo1=new_arithme(cV1bi,259,259,Na{9},Ta{9},Ca{9});
%cD1
cD1ro1=new_arithme(cD1ri,259,259,Na{10},Ta{10},Ca{10});
cD1go1=new_arithme(cD1gi,259,259,Na{11},Ta{11},Ca{11});
cD1bo1=new_arithme(cD1bi,259,259,Na{12},Ta{12},Ca{12});
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิด
ตัวกรอง db4
%cA2
cA2ro1=new_arithme(cA2ri,133,133,Na{13},Ta{13},Ca{13});
cA2go1=new_arithme(cA2gi,133,133,Na{14},Ta{14},Ca{14});
cA2bo1=new_arithme(cA2bi,133,133,Na{15},Ta{15},Ca{15});
%cH2
cH2ro1=new_arithme(cH2ri,133,133,Na{16},Ta{16},Ca{16});
cH2go1=new_arithme(cH2gi,133,133,Na{17},Ta{17},Ca{17});
cH2bo1=new_arithme(cH2bi,133,133,Na{18},Ta{18},Ca{18});
%cV2
cV2ro1=new_arithme(cV2ri,133,133,Na{19},Ta{19},Ca{19});
cV2go1=new_arithme(cV2gi,133,133,Na{20},Ta{20},Ca{20});
cV2bo1=new_arithme(cV2bi,133,133,Na{21},Ta{21},Ca{21});
%cD2
cD2ro1=new_arithme(cD2ri,133,133,Na{22},Ta{22},Ca{22});
cD2go1=new_arithme(cD2gi,133,133,Na{23},Ta{23},Ca{23});
cD2bo1=new_arithme(cD2bi,133,133,Na{24},Ta{24},Ca{24});
%----------------------------------------
```

```
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4
%cA3
cA3ro1=new_arithme(cA3ri,70,70,Na{25},Ta{25},Ca{25});
cA3go1=new_arithme(cA3gi,70,70,Na{26},Ta{26},Ca{26});
cA3bo1=new_arithme(cA3bi,70,70,Na{27},Ta{27},Ca{27});
%cH3
cH3ro1=new_arithme(cH3ri,70,70,Na{28},Ta{28},Ca{28});
cH3go1=new_arithme(cH3gi,70,70,Na{29},Ta{29},Ca{29});
cH3bo1=new_arithme(cH3bi,70,70,Na{30},Ta{30},Ca{30});
%cV3
cV3ro1=new_arithme(cV3ri,70,70,Na{31},Ta{31},Ca{31});
cV3go1=new_arithme(cV3gi,70,70,Na{32},Ta{32},Ca{32});
cV3bo1=new_arithme(cV3bi,70,70,Na{33},Ta{33},Ca{33});
%cD3
cD3ro1=new_arithme(cD3ri,70,70,Na{34},Ta{34},Ca{34});
cD3go1=new_arithme(cD3gi,70,70,Na{35},Ta{35},Ca{35});
cD3bo1=new_arithme(cD3bi,70,70,Na{36},Ta{36},Ca{36});
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar3 = idwt2(cA3ro1,cH3ro1,cV3ro1,cD3ro1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag3 = idwt2(cA3go1,cH3go1,cV3go1,cD3go1,'db4');
%-------------------------------------------------------------------------------------------
```

```
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab3 = idwt2(cA3bo1,cH3bo1,cV3bo1,cD3bo1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar2 = idwt2(cA2ro1,cH2ro1,cV2ro1,cD2ro1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag2 = idwt2(cA2go1,cH2go1,cV2go1,cD2go1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab2 = idwt2(cA2bo1,cH2bo1,cV2bo1,cD2bo1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar1 = idwt2(cA1ro1,cH1ro1,cV1ro1,cD1ro1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag1 = idwt2(cA1go1,cH1go1,cV1go1,cD1go1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab1 = idwt2(cA1bo1,cH1bo1,cV1bo1,cD1bo1,'db4');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
DeIm14(:,:,1)=Ar1;
DeIm14(:,:,2)=Ag1;
DeIm14(:,:,3)=Ab1;
figure()
imshow(uint8(DeIm14));title('reconstruct DWT  Lv1 ');
```

```
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 2
DeIm2(:,:,1)=Ar2;
DeIm2(:,:,2)=Ag2;
DeIm2(:,:,3)=Ab2;
figure()
imshow(uint8(DeIm2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
DeIm3(:,:,1)=Ar3;
DeIm3(:,:,2)=Ag3;
DeIm3(:,:,3)=Ab3;
figure()
imshow(uint8(DeIm3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(DeIm14),uint8(X))
toc
```

```
close all, clear all, clc
tic % ฟังก์ชั่นจับเวลา
X = imread('peppers1.png'); %X คือข้อมูลภาพดิจิตอลสีพริกไทย
[sr,sc,pp]=size(X);%ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
figure(1);
imshow(X); title('original image');% แสดงภาพต้นแบบสีพริกไทย
%-------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
%-------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA1r,cH1r,cV1r,cD1r] = dwt2(double(X(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA1g,cH1g,cV1g,cD1g] = dwt2(double(X(:,:,2)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA1b,cH1b,cV1b,cD1b] = dwt2(double(X(:,:,3)), 'db4');
%-------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA2r,cH2r,cV2r,cD2r] = dwt2(double(cA1r(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA2g,cH2g,cV2g,cD2g] = dwt2(double(cA1g(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA2b,cH2b,cV2b,cD2b] = dwt2(double(cA1b(:,:,1)),'db4');
%-------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA3r,cH3r,cV3r,cD3r] = dwt2(double(cA2r(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA3g,cH3g,cV3g,cD3g] = dwt2(double(cA2g(:,:,1)),'db4');
```

```
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA3b,cH3b,cV3b,cD3b] = dwt2(double(cA2b(:,:,1)),'db4');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1ri=cA1r(:);
cA1gi=cA1g(:);
cA1bi=cA1b(:);

cH1ri=cH1r(:);
cH1gi=cH1g(:);
cH1bi=cH1b(:);

cV1ri=cV1r(:);
cV1gi=cV1g(:);
cV1bi=cV1b(:);

cD1ri=cD1r(:);
cD1gi=cD1g(:);
cD1bi=cD1b(:);
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2ri=cA2r(:);
cA2gi=cA2g(:);
cA2bi=cA2b(:);

cH2ri=cH2r(:);
cH2gi=cH2g(:);
cH2bi=cH2b(:);
```

```
cV2ri=cV2r(:);
cV2gi=cV2g(:);
cV2bi=cV2b(:);

cD2ri=cD2r(:);
cD2gi=cD2g(:);
cD2bi=cD2b(:);
%-------------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3ri=cA3r(:);
cA3gi=cA3g(:);
cA3bi=cA3b(:);

cH3ri=cH3r(:);
cH3gi=cH3g(:);
cH3bi=cH3b(:);

cV3ri=cV3r(:);
cV3gi=cV3g(:);
cV3bi=cV3b(:);

cD3ri=cD3r(:);
cD3gi=cD3g(:);
cD3bi=cD3b(:);
%-------------------------------------------------------------------------------------------------
```

```
% บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_db4_peppers_cA1ro.PM','bit_db4_peppers_cA1go.PM','bit_db4_peppers_cA1bo.PM'
'bit_db4_peppers_cH1ro.PM','bit_db4_peppers_cH1go.PM','bit_db4_peppers_cH1bo.PM'
'bit_db4_peppers_cV1ro.PM','bit_db4_peppers_cV1go.PM','bit_db4_peppers_cV1bo.PM'
'bit_db4_peppers_cD1ro.PM','bit_db4_peppers_cD1go.PM','bit_db4_peppers_cD1bo.PM'
'bit_db4_peppers_cA2ro.PM','bit_db4_peppers_cA2go.PM','bit_db4_peppers_cA2bo.PM'
'bit_db4_peppers_cH2ro.PM','bit_db4_peppers_cH2go.PM','bit_db4_peppers_cH2bo.PM'
'bit_db4_peppers_cV2ro.PM','bit_db4_peppers_cV2go.PM','bit_db4_peppers_cV2bo.PM'
'bit_db4_peppers_cD2ro.PM','bit_db4_peppers_cD2go.PM','bit_db4_peppers_cD2bo.PM'
'bit_db4_peppers_cA3ro.PM','bit_db4_peppers_cA3go.PM','bit_db4_peppers_cA3bo.PM'
'bit_db4_peppers_cH3ro.PM','bit_db4_peppers_cH3go.PM','bit_db4_peppers_cH3bo.PM'
'bit_db4_peppers_cV3ro.PM','bit_db4_peppers_cV3go.PM','bit_db4_peppers_cV3bo.PM'
'bit_db4_peppers_cD3ro.PM','bit_db4_peppers_cD3go.PM','bit_db4_peppers_cD3bo.PM'};
%-------------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_db4_peppers_cA1ro.PM','bit_table_db4_peppers_cA1go.PM',
'bit_table_db4_peppers_cA1bo.PM'
'bit_table_db4_peppers_cH1ro.PM','bit_table_db4_peppers_cH1go.PM',
'bit_table_db4_peppers_cH1bo.PM'
'bit_table_db4_peppers_cV1ro.PM','bit_table_db4_peppers_cV1go.PM',
'bit_table_db4_peppers_cV1bo.PM'
'bit_table_db4_peppers_cD1ro.PM','bit_table_db4_peppers_cD1go.PM',
'bit_table_db4_peppers_cD1bo.PM'
'bit_table_db4_peppers_cA2ro.PM','bit_table_db4_peppers_cA2go.PM',
'bit_table_db4_peppers_cA2bo.PM'
'bit_table_db4_peppers_cH2ro.PM','bit_table_db4_peppers_cH2go.PM',
'bit_table_db4_peppers_cH2bo.PM'
'bit_table_db4_peppers_cV2ro.PM','bit_table_db4_peppers_cV2go.PM',
'bit_table_db4_peppers_cV2bo.PM'
```

```
'bit_table_db4_peppers_cD2ro.PM','bit_table_db4_peppers_cD2go.PM',
'bit_table_db4_peppers_cD2bo.PM'
'bit_table_db4_peppers_cA3ro.PM','bit_table_db4_peppers_cA3go.PM',
'bit_table_db4_peppers_cA3bo.PM'
'bit_table_db4_peppers_cH3ro.PM','bit_table_db4_peppers_cH3go.PM',
'bit_table_db4_peppers_cH3bo.PM'
'bit_table_db4_peppers_cV3ro.PM','bit_table_db4_peppers_cV3go.PM',
'bit_table_db4_peppers_cV3bo.PM'
'bit_table_db4_peppers_cD3ro.PM','bit_table_db4_peppers_cD3go.PM',
'bit_table_db4_peppers_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_db4_peppers_cA1ro.PM','bit_count_db4_peppers_cA1go.PM',
'bit_count_db4_peppers_cA1bo.PM'
'bit_count_db4_peppers_cH1ro.PM','bit_count_db4_count_cH1go.PM',
'bit_count_db4_peppers_cH1bo.PM'
'bit_count_db4_peppers_cV1ro.PM','bit_count_db4_peppers_cV1go.PM',
'bit_count_db4_peppers_cV1bo.PM'
'bit_count_db4_peppers_cD1ro.PM','bit_count_db4_peppers_cD1go.PM',
'bit_count_db4_peppers_cD1bo.PM'
'bit_count_db4_peppers_cA2ro.PM','bit_count_db4_peppers_cA2go.PM',
'bit_count_db4_peppers_cA2bo.PM'
'bit_count_db4_peppers_cH2ro.PM','bit_count_db4_peppers_cH2go.PM',
'bit_count_db4_peppers_cH2bo.PM'
'bit_count_db4_peppers_cV2ro.PM','bit_count_db4_peppers_cV2go.PM',
'bit_count_db4_peppers_cV2bo.PM'
'bit_count_db4_peppers_cD2ro.PM','bit_count_db4_peppers_cD2go.PM',
'bit_count_db4_peppers_cD2bo.PM'
```

```
'bit_count_db4__peppers_cA3ro.PM','bit_count_db4_peppers_cA3go.PM',
'bit_count_db4_peppers_cA3bo.PM'
'bit_count_db4_peppers_cH3ro.PM','bit_table_db4_peppers_cH3go.PM',
'bit_count_db4_peppers_cH3bo.PM'
'bit_count_db4_peppers_cV3ro.PM','bit_count_db4_peppers_cV3go.PM',
'bit_count_db4_peppers_cV3bo.PM'
'bit_count_db4_peppers_cD3ro.PM','bit_count_db4_peppers_cD3go.PM',
'bit_count_db4_peppers_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4
%cA1
cA1ro1=new_arithme(cA1ri,259,259,Na{1},Ta{1},Ca{1});
cA1go1=new_arithme(cA1gi,259,259,Na{2},Ta{2},Ca{2});
cA1bo1=new_arithme(cA1bi,259,259,Na{3},Ta{3},Ca{3});
%cH1
cH1ro1=new_arithme(cH1ri,259,259,Na{4},Ta{4},Ca{4});
cH1go1=new_arithme(cH1gi,259,259,Na{5},Ta{5},Ca{5});
cH1bo1=new_arithme(cH1bi,259,259,Na{6},Ta{6},Ca{6});
%cV1
cV1ro1=new_arithme(cV1ri,259,259,Na{7},Ta{7},Ca{7});
cV1go1=new_arithme(cV1gi,259,259,Na{8},Ta{8},Ca{8});
cV1bo1=new_arithme(cV1bi,259,259,Na{9},Ta{9},Ca{9});
%cD1
cD1ro1=new_arithme(cD1ri,259,259,Na{10},Ta{10},Ca{10});
cD1go1=new_arithme(cD1gi,259,259,Na{11},Ta{11},Ca{11});
cD1bo1=new_arithme(cD1bi,259,259,Na{12},Ta{12},Ca{12});
```

```
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4
%cA2
cA2ro1=new_arithme(cA2ri,133,133,Na{13},Ta{13},Ca{13});
cA2go1=new_arithme(cA2gi,133,133,Na{14},Ta{14},Ca{14});
cA2bo1=new_arithme(cA2bi,133,133,Na{15},Ta{15},Ca{15});
%cH2
cH2ro1=new_arithme(cH2ri,133,133,Na{16},Ta{16},Ca{16});
cH2go1=new_arithme(cH2gi,133,133,Na{17},Ta{17},Ca{17});
cH2bo1=new_arithme(cH2bi,133,133,Na{18},Ta{18},Ca{18});
%cV2
cV2ro1=new_arithme(cV2ri,133,133,Na{19},Ta{19},Ca{19});
cV2go1=new_arithme(cV2gi,133,133,Na{20},Ta{20},Ca{20});
cV2bo1=new_arithme(cV2bi,133,133,Na{21},Ta{21},Ca{21});
%cD2
cD2ro1=new_arithme(cD2ri,133,133,Na{22},Ta{22},Ca{22});
cD2go1=new_arithme(cD2gi,133,133,Na{23},Ta{23},Ca{23});
cD2bo1=new_arithme(cD2bi,133,133,Na{24},Ta{24},Ca{24});
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4
%cA3
cA3ro1=new_arithme(cA3ri,70,70,Na{25},Ta{25},Ca{25});
cA3go1=new_arithme(cA3gi,70,70,Na{26},Ta{26},Ca{26});
cA3bo1=new_arithme(cA3bi,70,70,Na{27},Ta{27},Ca{27});
```

```
%cH3
cH3ro1=new_arithme(cH3ri,70,70,Na{28},Ta{28},Ca{28});
cH3go1=new_arithme(cH3gi,70,70,Na{29},Ta{29},Ca{29});
cH3bo1=new_arithme(cH3bi,70,70,Na{30},Ta{30},Ca{30});
%cV3
cV3ro1=new_arithme(cV3ri,70,70,Na{31},Ta{31},Ca{31});
cV3go1=new_arithme(cV3gi,70,70,Na{32},Ta{32},Ca{32});
cV3bo1=new_arithme(cV3bi,70,70,Na{33},Ta{33},Ca{33});
%cD3
cD3ro1=new_arithme(cD3ri,70,70,Na{34},Ta{34},Ca{34});
cD3go1=new_arithme(cD3gi,70,70,Na{35},Ta{35},Ca{35});
cD3bo1=new_arithme(cD3bi,70,70,Na{36},Ta{36},Ca{36});
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar3 = idwt2(cA3ro1,cH3ro1,cV3ro1,cD3ro1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag3 = idwt2(cA3go1,cH3go1,cV3go1,cD3go1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab3 = idwt2(cA3bo1,cH3bo1,cV3bo1,cD3bo1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar2 = idwt2(cA2ro1,cH2ro1,cV2ro1,cD2ro1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag2 = idwt2(cA2go1,cH2go1,cV2go1,cD2go1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab2 = idwt2(cA2bo1,cH2bo1,cV2bo1,cD2bo1,'db4');
```

```
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar1 = idwt2(cA1ro1,cH1ro1,cV1ro1,cD1ro1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag1 = idwt2(cA1go1,cH1go1,cV1go1,cD1go1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab1 = idwt2(cA1bo1,cH1bo1,cV1bo1,cD1bo1,'db4');
%-----------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
DeIm14(:,:,1)=Ar1;
DeIm14(:,:,2)=Ag1;
DeIm14(:,:,3)=Ab1;
figure(), imshow(uint8(DeIm14));title('reconstruct DWT  Lv1 ');
%-----------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 2
DeIm2(:,:,1)=Ar2;
DeIm2(:,:,2)=Ag2;
DeIm2(:,:,3)=Ab2;
figure(), imshow(uint8(DeIm2));title('reconstruct DWT  Lv2 ');
%-----------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
DeIm3(:,:,1)=Ar3;
DeIm3(:,:,2)=Ag3;
DeIm3(:,:,3)=Ab3;
figure()' imshow(uint8(DeIm3));title('reconstruct DWT  Lv3 ');
%-----------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(DeIm14),uint8(X))
toc
```

```
close all, clear all, clc
tic % ฟังก์ชั่นจับเวลา
X = imread('mandril.tif');%X คือข้อมูลภาพดิจิตอลสีแมนดริล
[sr,sc,pp]=size(X);%ฟังก์ชั่นบอกขนาดของภาพดิจิตอล
figure(1);
imshow(X); title('original image');% แสดงภาพต้นแบบสีแมนดริล
%-----------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
%-----------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA1r,cH1r,cV1r,cD1r] = dwt2(double(X(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA1g,cH1g,cV1g,cD1g] = dwt2(double(X(:,:,2)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA1b,cH1b,cV1b,cD1b] = dwt2(double(X(:,:,3)), 'db4');
%-----------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA2r,cH2r,cV2r,cD2r] = dwt2(double(cA1r(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA2g,cH2g,cV2g,cD2g] = dwt2(double(cA1g(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA2b,cH2b,cV2b,cD2b] = dwt2(double(cA1b(:,:,1)),'db4');
%-----------------------------------------------------------------------------------------------
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีแดง
[cA3r,cH3r,cV3r,cD3r] = dwt2(double(cA2r(:,:,1)),'db4');
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีเขียว
[cA3g,cH3g,cV3g,cD3g] = dwt2(double(cA2g(:,:,1)),'db4');
```

```
% การแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
[cA3b,cH3b,cV3b,cD3b] = dwt2(double(cA2b(:,:,1)),'db4');
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 1 เป็นเวกเตอร์
cA1ri=cA1r(:);
cA1gi=cA1g(:);
cA1bi=cA1b(:);

cH1ri=cH1r(:);
cH1gi=cH1g(:);
cH1bi=cH1b(:);

cV1ri=cV1r(:);
cV1gi=cV1g(:);
cV1bi=cV1b(:);

cD1ri=cD1r(:);
cD1gi=cD1g(:);
cD1bi=cD1b(:);
%-------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 2 เป็นเวกเตอร์
cA2ri=cA2r(:);
cA2gi=cA2g(:);
cA2bi=cA2b(:);

cH2ri=cH2r(:);
cH2gi=cH2g(:);
cH2bi=cH2b(:);
```

```
cV2ri=cV2r(:);
cV2gi=cV2g(:);
cV2bi=cV2b(:);

cD2ri=cD2r(:);
cD2gi=cD2g(:);
cD2bi=cD2b(:);
%-------------------------------------------------------------------------------------------------
% การแปลงสัมประสิทธิ์เวฟเล็ตระดับที่ 3 เป็นเวกเตอร์
cA3ri=cA3r(:);
cA3gi=cA3g(:);
cA3bi=cA3b(:);

cH3ri=cH3r(:);
cH3gi=cH3g(:);
cH3bi=cH3b(:);

cV3ri=cV3r(:);
cV3gi=cV3g(:);
cV3bi=cV3b(:);

cD3ri=cD3r(:);
cD3gi=cD3g(:);
cD3bi=cD3b(:);
%-------------------------------------------------------------------------------------------------
```

```
% บันทึกสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Na={'bit_db4_mandril_cA1ro.PM','bit_db4_mandril_cA1go.PM','bit_db4_mandril_cA1bo.PM'
'bit_db4_mandril_cH1ro.PM','bit_db4_mandril_cH1go.PM','bit_db4_mandril_cH1bo.PM'
'bit_db4_mandril_cV1ro.PM','bit_db4_mandril_cV1go.PM','bit_db4_mandril_cV1bo.PM'
'bit_db4_mandril_cD1ro.PM','bit_db4_mandril_cD1go.PM','bit_db4_mandril_cD1bo.PM'
'bit_db4_mandril_cA2ro.PM','bit_db4_mandril_cA2go.PM','bit_db4_mandril_cA2bo.PM'
'bit_db4_mandril_cH2ro.PM','bit_db4_mandril_cH2go.PM','bit_db4_mandril_cH2bo.PM'
'bit_db4_mandril_cV2ro.PM','bit_db4_mandril_cV2go.PM','bit_db4_mandril_cV2bo.PM'
'bit_db4_mandril_cD2ro.PM','bit_db4_mandril_cD2go.PM','bit_db4_mandril_cD2bo.PM'
'bit_db4_mandril_cA3ro.PM','bit_db4_mandril_cA3go.PM','bit_db4_mandril_cA3bo.PM'
'bit_db4_mandril_cH3ro.PM','bit_db4_mandril_cH3go.PM','bit_db4_mandril_cH3bo.PM'
'bit_db4_mandril_cV3ro.PM','bit_db4_mandril_cV3go.PM','bit_db4_mandril_cV3bo.PM'
'bit_db4_mandril_cD3ro.PM','bit_db4_mandril_cD3go.PM','bit_db4_mandril_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
%บันทึกตารางสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ta={'bit_table_db4_mandril_cA1ro.PM','bit_table_db4_mandril_cA1go.PM',
'bit_table_db4_mandril_cA1bo.PM'
'bit_table_db4_mandril_cH1ro.PM','bit_table_db4_mandril_cH1go.PM',
'bit_table_db4_mandril_cH1bo.PM'
'bit_table_db4_mandril_cV1ro.PM','bit_table_db4_mandril_cV1go.PM',
'bit_table_db4_mandril_cV1bo.PM'
'bit_table_db4_mandril_cD1ro.PM','bit_table_db4_mandril_cD1go.PM',
'bit_table_db4_mandril_cD1bo.PM'
'bit_table_db4_mandril_cA2ro.PM','bit_table_db4_mandril_cA2go.PM','bit_table_db4_mandril_c
A2bo.PM'
'bit_table_db4_mandril_cH2ro.PM','bit_table_db4_mandril_cH2go.PM','bit_table_db4_mandril_c
H2bo.PM'
```

'bit_table_db4_mandril_cV2ro.PM','bit_table_db4_mandril_cV2go.PM','bit_table_db4_mandril_c
V2bo.PM'
'bit_table_db4_mandril_cD2ro.PM','bit_table_db4_mandril_cD2go.PM','bit_table_db4_mandril_c
D2bo.PM'
'bit_table_db4_mandril_cA3ro.PM','bit_table_db4_mandril_cA3go.PM',
'bit_table_db4_mandril_cA3bo.PM'
'bit_table_db4_mandril_cH3ro.PM','bit_table_db4_mandril_cH3go.PM',
'bit_table_db4_mandril_cH3bo.PM'
'bit_table_db4_mandril_cV3ro.PM','bit_table_db4_mandril_cV3go.PM',
'bit_table_db4_mandril_cV3bo.PM'
'bit_table_db4_mandril_cD3ro.PM','bit_table_db4_mandril_cD3go.PM',
'bit_table_db4_mandril_cD3bo.PM'};
%----------

% บันทึกจำนวนข้อมูลสัมประสิทธิ์เวฟเล็ตที่ถูกบีบอัดเป็นแฟ้มข้อมูลฐานสอง
Ca={'bit_count_db4_mandril_cA1ro.PM','bit_count_db4_mandril_cA1go.PM',
'bit_count_db4_mandril_cA1bo.PM'
'bit_count_db4_mandril_cH1ro.PM','bit_count_db4_mandril_cH1go.PM',
'bit_count_db4_mandril_cH1bo.PM'
'bit_count_db4_mandril_cV1ro.PM','bit_count_db4_mandril_cV1go.PM',
'bit_count_db4_mandril_cV1bo.PM'
'bit_count_db4_mandril_cD1ro.PM','bit_count_db4_mandril_cD1go.PM',
'bit_count_db4_mandril_cD1bo.PM'
'bit_count_db4_mandril_cA2ro.PM','bit_count_db4_mandril_cA2go.PM',
'bit_count_db4_mandril_cA2bo.PM'
'bit_count_db4_mandril_cH2ro.PM','bit_count_db4_mandril_cH2go.PM',
'bit_count_db4_mandril_cH2bo.PM'
'bit_count_db4_mandril_cV2ro.PM','bit_count_db4_mandril_cV2go.PM',
'bit_count_db4_mandril_cV2bo.PM'

```
'bit_count_db4_mandril_cD2ro.PM','bit_count_db4_mandril_cD2go.PM',
'bit_count_db4_mandril_cD2bo.PM'
'bit_count_db4_mandril_cA3ro.PM','bit_count_db4_mandril_cA3go.PM',
'bit_count_db4_mandril_cA3bo.PM'
'bit_count_db4_mandril_cH3ro.PM','bit_count_db4_mandril_cH3go.PM',
'bit_count_db4_mandril_cH3bo.PM'
'bit_count_db4_mandril_cV3ro.PM','bit_count_db4_mandril_cV3go.PM',
'bit_count_db4_mandril_cV3bo.PM'
'bit_count_db4_mandril_cD3ro.PM','bit_count_db4_mandril_cD3go.PM',
'bit_count_db4_mandril_cD3bo.PM'};
%-------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 1 ชนิด
ตัวกรอง db4
%cA1
cA1ro1=new_arithme(cA1ri,259,259,Na{1},Ta{1},Ca{1});
cA1go1=new_arithme(cA1gi,259,259,Na{2},Ta{2},Ca{2});
cA1bo1=new_arithme(cA1bi,259,259,Na{3},Ta{3},Ca{3});
%cH1
cH1ro1=new_arithme(cH1ri,259,259,Na{4},Ta{4},Ca{4});
cH1go1=new_arithme(cH1gi,259,259,Na{5},Ta{5},Ca{5});
cH1bo1=new_arithme(cH1bi,259,259,Na{6},Ta{6},Ca{6});
%cV1
cV1ro1=new_arithme(cV1ri,259,259,Na{7},Ta{7},Ca{7});
cV1go1=new_arithme(cV1gi,259,259,Na{8},Ta{8},Ca{8});
cV1bo1=new_arithme(cV1bi,259,259,Na{9},Ta{9},Ca{9});
```

```
%cD1
cD1ro1=new_arithme(cD1ri,259,259,Na{10},Ta{10},Ca{10});
cD1go1=new_arithme(cD1gi,259,259,Na{11},Ta{11},Ca{11});
cD1bo1=new_arithme(cD1bi,259,259,Na{12},Ta{12},Ca{12});
%-----------------------------------------------------------------------------------------------
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 2 ชนิด
ตัวกรอง db4
%cA2
cA2ro1=new_arithme(cA2ri,133,133,Na{13},Ta{13},Ca{13});
cA2go1=new_arithme(cA2gi,133,133,Na{14},Ta{14},Ca{14});
cA2bo1=new_arithme(cA2bi,133,133,Na{15},Ta{15},Ca{15});
%cH2
cH2ro1=new_arithme(cH2ri,133,133,Na{16},Ta{16},Ca{16});
cH2go1=new_arithme(cH2gi,133,133,Na{17},Ta{17},Ca{17});
cH2bo1=new_arithme(cH2bi,133,133,Na{18},Ta{18},Ca{18});
%cV2
cV2ro1=new_arithme(cV2ri,133,133,Na{19},Ta{19},Ca{19});
cV2go1=new_arithme(cV2gi,133,133,Na{20},Ta{20},Ca{20});
cV2bo1=new_arithme(cV2bi,133,133,Na{21},Ta{21},Ca{21});
%cD2
cD2ro1=new_arithme(cD2ri,133,133,Na{22},Ta{22},Ca{22});
cD2go1=new_arithme(cD2gi,133,133,Na{23},Ta{23},Ca{23});
cD2bo1=new_arithme(cD2bi,133,133,Na{24},Ta{24},Ca{24});
%-----------------------------------------------------------------------------------------------
```

```
%การเข้ารหัสและถอดรหัสเลขคณิตของการแปลงดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับ 3 ชนิดตัวกรอง db4
%cA3
cA3ro1=new_arithme(cA3ri,70,70 ,Na{25},Ta{25},Ca{25});
cA3go1=new_arithme(cA3gi,70,70 ,Na{26},Ta{26},Ca{26});
cA3bo1=new_arithme(cA3bi,70,70 ,Na{27},Ta{27},Ca{27});
%cH3
cH3ro1=new_arithme(cH3ri,70,70 ,Na{28},Ta{28},Ca{28});
cH3go1=new_arithme(cH3gi,70,70 ,Na{29},Ta{29},Ca{29});
cH3bo1=new_arithme(cH3bi,70,70 ,Na{30},Ta{30},Ca{30});
%cV3
cV3ro1=new_arithme(cV3ri,70,70 ,Na{31},Ta{31},Ca{31});
cV3go1=new_arithme(cV3gi,70,70 ,Na{32},Ta{32},Ca{32});
cV3bo1=new_arithme(cV3bi,70,70 ,Na{33},Ta{33},Ca{33});
%cD3
cD3ro1=new_arithme(cD3ri,70,70 ,Na{34},Ta{34},Ca{34});
cD3go1=new_arithme(cD3gi,70,70 ,Na{35},Ta{35},Ca{35});
cD3bo1=new_arithme(cD3bi,70,70 ,Na{36},Ta{36},Ca{36});
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar3 = idwt2(cA3ro1,cH3ro1,cV3ro1,cD3ro1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag3 = idwt2(cA3go1,cH3go1,cV3go1,cD3go1,'db4');
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 3 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab3 = idwt2(cA3bo1,cH3bo1,cV3bo1,cD3bo1,'db4');
```

```
%-------------------------------------------------------------------------------------------
% idwt lv 2
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar2 = idwt2(cA2ro1,cH2ro1,cV2ro1,cD2ro1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag2 = idwt2(cA2go1,cH2go1,cV2go1,cD2go1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 2 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab2 = idwt2(cA2bo1,cH2bo1,cV2bo1,cD2bo1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีแดง
Ar1 = idwt2(cA1ro1,cH1ro1,cV1ro1,cD1ro1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีเขียว
Ag1 = idwt2(cA1go1,cH1go1,cV1go1,cD1go1,'db4');
%-------------------------------------------------------------------------------------------
% การแปลงกลับดิสครีทเวฟเล็ตทรานฟอร์ม 2 มิติ ระดับที่ 1 ชนิดตัวกรอง db4 บนระนาบสีน้ำเงิน
Ab1 = idwt2(cA1bo1,cH1bo1,cV1bo1,cD1bo1,'db4');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 1
DeIm14(:,:,1)=Ar1;
DeIm14(:,:,2)=Ag1;
DeIm14(:,:,3)=Ab1;
figure()
imshow(uint8(DeIm14));title('reconstruct DWT  Lv1 ');
```

```
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 2
DeIm2(:,:,1)=Ar2;
DeIm2(:,:,2)=Ag2;
DeIm2(:,:,3)=Ab2;
figure()
imshow(uint8(DeIm2));title('reconstruct DWT  Lv2 ');
%-------------------------------------------------------------------------------------------
% รีคอนสตรักดิสครีทเวฟเล็ตทรานฟอร์มชนิดตัวกรอง db4 ระดับที่ 3
DeIm3(:,:,1)=Ar3;
DeIm3(:,:,2)=Ag3;
DeIm3(:,:,3)=Ab3;
figure()
imshow(uint8(DeIm3));title('reconstruct DWT  Lv3 ');
%-------------------------------------------------------------------------------------------
%ฟังก์ชั่นหาค่าความผิดพลาดเฉลี่ยกำลังสอง(MSE) ค่าพีเอสเอ็นอาร์ (PSNR) และค่าเอสเอ็นอาร์
(SNR)
[MSE, SNR,P_SNR]=PEAK_SNR_MSE(uint8(DeIm14),uint8(X))
toc
```

ภาคผนวก ง

รายละเอียดโปรแกรม Arithmetic Coding

```
function code = arithenco1(seq, counts)

%ARITHENCO Encode a sequence of symbols using arithmetic coding.

%  References:

%       [1] Sayood, K., Introduction to Data Compression,

%       Morgan Kaufmann, 2000, Chapter 4, Section 4.4.3.

% Check the incoming orientation and adjust if necessary

[row_s, col_s] = size(seq);

if (row_s > 1),

seq = seq.';

end

[row_c, col_c] = size(counts);

if (row_c > 1),

counts = counts.';

end

% Compute the cumulative counts vector from the counts

cum_counts = [0, cumsum(counts)];
```

```
% Compute the Word Length required.

% คำนวณหาความยาว บิต

total_count = cum_counts(end);

N = ceil(log2(total_count)) + 2;

% Initialize the lower and upper bounds.

dec_low = 0;

dec_up = 2^N-1;

E3_count = 0;

% Obtain an over estimate for the length of CODE and initialize CODE

code_len = length(seq) * ( ceil(log2(length(counts))) + 2 ) + N;

code = zeros(1, code_len);

code_index = 1;

th=ceil((2^N-1)/2);

    th1=ceil((2^N-1)/2)/2-1;

% Loop for each symbol in SEQ

for k = 1:length(seq)

symbol = seq(k);
```

```
    % Compute the new lower bound
    dec_low_new = dec_low + floor( (dec_up-dec_low+1)*cum_counts(symbol+1-1)/total_count
);
    % Compute the new upper bound
    dec_up = dec_low + floor( (dec_up-dec_low+1)*cum_counts(symbol+1)/total_count )-1;
    % Update the lower bound
    dec_low = dec_low_new;
    % Check for E1, E2 or E3 conditions and keep looping as long as they occur.
        while or(or(and(dec_low<th,dec_up<th), and(dec_low>=th,dec_up>=th)),...
and(dec_low>th1,dec_up<(th+th1)))
      % If it is an E1 or E2 condition,
if not(xor(dec_low>=th,dec_up>=th))
        % Get the MSB
        b = bitget(dec_low, N);
code(code_index) = b;
        code_index = code_index + 1;
        % Left shifts and add bit
        bi_low=dec2bin(dec_low,N);
        bi_low1=[bi_low '0'];
```

```
        bi_low=bi_low1(2:end);

        dec_low=bin2dec(bi_low);

        bi_up=dec2bin(dec_up,N);

        bi_up1=[bi_up '1'];

        bi_up=bi_up1(2:end);

dec_up=bin2dec(bi_up);

        % Check if E3_count is non-zero and transmit appropriate bits

if (E3_count > 0),

            % Have to transmit complement of b, E3_count times.

code(code_index:code_index+E3_count-1) = (~b).*ones(1, E3_count);

            code_index = code_index + E3_count;

            E3_count = 0;

end

      % Else if it is an E3 condition

elseif and(dec_low>th1,and(dec_up<(th+th1),dec_up>th)),

        % Left shifts

        bi_low=dec2bin(dec_low,N);

        bi_up=dec2bin(dec_up,N);
```

```
        bi_low1=[bi_low '0'];

        bi_low=bi_low1(2:end);

        bi_up1=[bi_up '1'];

        bi_up=bi_up1(2:end);

if bi_low(1)=='0'

          bi_low(1)='1' ;

else

          bi_low(1)='0';

end

if bi_up(1)=='0'

          bi_up(1)='1' ;

else

          bi_up(1)='0';

end

         dec_low=bin2dec(bi_low);

        dec_up=bin2dec(bi_up);
```

```
        % Increment E3_count to keep track of number of times E3 condition is hit.
        E3_count = E3_count+1;
end
end
end
% Terminate encoding
bin_low = de2bi(dec_low, N, 'left-msb');
if E3_count==0,
   % Just transmit the final value of the lower bound bin_low
code(code_index:code_index + N - 1) = bin_low;
   code_index = code_index + N;
else
  % Transmit the MSB of bin_low.
  b = bin_low(1);
code(code_index) = b;
  code_index = code_index + 1;
  % Then transmit complement of b (MSB of bin_low), E3_count times.
code(code_index:code_index+E3_count-1) = (~b).*ones(1, E3_count);
```

```
  code_index = code_index + E3_count;
  % Then transmit the remaining bits of bin_low
code(code_index:code_index+N-2) = bin_low(2:N);
  code_index = code_index + N - 1;
end
% Output only the filled values
code = code(1:code_index-1);
% Set the same output orientation as seq
if (row_s > 1)
code = code.';
end
function dseq = arithdeco1(code, counts, len)
%     seq = [2 3 1 3 3];
%     counts = [29 48 100];
%     code = arithenco(seq, counts)
%     % To decode this code (and recover the sequence of
%     % symbols it represents) use this command:
%     dseq = arithdeco(code, counts, 5)
```

```
% See also ARITHENCO.

% Copyright 1996-2011 The MathWorks, Inc.

% References:

%       [1] Sayood, K., Introduction to Data Compression,

%       Morgan Kaufmann, 2000, Chapter 4, Section 4.4.3.

% Check the incoming orientation and adjust if necessary

row_cd = size(code, 1);

if (row_cd > 1),

code = code.';

end

if (size(counts, 1) > 1),

counts = counts.';

end

% Compute the cumulative counts vector from the counts vector

cum_counts = [0, cumsum(counts)];

% Compute the Word Length (N) required.

total_count = cum_counts(end);

N = ceil(log2(total_count)) + 2;
```

```
% Initialize the lower and upper bounds.

dec_low = 0;

dec_up = 2^N-1;

% Read the first N number of bits into a temporary tag bin_tag

bin_tag = code(1:N);

dec_tag = bi2de(bin_tag, 'left-msb');

dseq = zeros(1,len); % bit ???????

dseq_index = 1;

k=N;

% This loop runs until all the symbols are decoded into DSEQ

while (dseq_index <= len)

   % Compute dec_tag_new

   dec_tag_new =floor( ((dec_tag-dec_low+1)*total_count-1)/(dec_up-dec_low+1) );

   % Decode a symbol based on dec_tag_new

ptr = pick(cum_counts, dec_tag_new);

   % Update DSEQ by adding the decoded symbol

dseq(dseq_index) = ptr;

   dseq_index = dseq_index + 1;
```

```
    % Compute the new lower bound

    dec_low_new = dec_low + floor( (dec_up-dec_low+1)*cum_counts(ptr-1+1)/total_count );

    % Compute the new upper bound

    dec_up = dec_low + floor( (dec_up-dec_low+1)*cum_counts(ptr+1)/total_count )-1;

    % Update the lower bound

    dec_low = dec_low_new;

th=ceil((2^N-1)/2);

    th1=ceil((2^N-1)/2)/2-1;

    % Check for E1, E2 or E3 conditions and keep looping as long as they occur.

     while or(or(and(dec_low<th,dec_up<th), and(dec_low>=th,dec_up>=th)),...

and(dec_low>th1,dec_up<(th+th1)))

      % Break out if we have finished working with all the bits in CODE

if ( k==length(code) ), break, end;

      k = k + 1;

      % If it is an E1 or E2 condition, do

if not(xor(dec_low>=th,dec_up>=th))

          % Left shifts and update

           bi_low=dec2bin(dec_low,N);
```

```
            bi_up=dec2bin(dec_up,N);

            bi_low1=[bi_low '0'];

            bi_low=bi_low1(2:end);

            bi_up1=[bi_up '1'];

            bi_up=bi_up1(2:end);

            dec_low=bin2dec(bi_low);

            dec_up=bin2dec(bi_up);

          % Left shift and read in code ********

          bi_tag=dec2bin(dec_tag);

if code(k)==1

              bi_tag=[bi_tag '1'];

else

              bi_tag=[bi_tag '0'];

end

          dec_tag=bin2dec(bi_tag);

          % Reduce to N for next loop *********

            dec_tag = bitset(dec_tag, N+1, 0);

            dec_low=bin2dec(bi_low);
```

```
        dec_up=bin2dec(bi_up);
    % Else if it is an E3 condition
elseif and(dec_low>th1,and(dec_up<(th+th1),dec_up>th)),
        % Left shifts and update
        bi_low=dec2bin(dec_low,N);
        bi_up=dec2bin(dec_up,N);
        bi_low1=[bi_low '0'];
        bi_low=bi_low1(2:end);
        bi_up1=[bi_up '1'];
        bi_up=bi_up1(2:end);
        dec_low=bin2dec(bi_low);
        dec_up=bin2dec(bi_up);
        % Left shift and read in code
        bi_tag=dec2bin(dec_tag);
if code(k)==1
          bi_tag=[bi_tag '1'];
else
          bi_tag=[bi_tag '0'];
end
        dec_tag=bin2dec(bi_tag);
```

```
        % Reduce to N for next loop

        dec_low = bitset(dec_low, N+1, 0);

        dec_up  = bitset(dec_up,  N+1, 0);

        dec_tag = bitset(dec_tag, N+1, 0);

        % Complement the new MSB of dec_low, dec_up and dec_tag

        dec_low = bitxor(dec_low, 2^(N-1) );

        dec_up  = bitxor(dec_up,  2^(N-1) );

        dec_tag = bitxor(dec_tag, 2^(N-1) );

end

end % end while

end % end while length(dseq)

% Set the same output orientation as code

if (row_cd > 1)

dseq = dseq.';

end
```

```
function [ptr] = pick(cum_counts, value)

% This internal function is used to find where value is positioned

% Check for this case and quickly exit

if value == cum_counts(end)

ptr = length(cum_counts)-1;

return

end

c = find(cum_counts <= value);

ptr = c(end);

function errorchk(code, counts, len)

% Function for validating the input parameters.

% Check to make sure a vector has been entered as input and not a matrix

if (length(find(size(code)==1)) ~= 1)

error(message('comm:arithdeco:InvalidCodeParameter'));

return;

end
```

```
if (length(find(size(counts)==1)) ~= 1)

error(message('comm:arithdeco:InvalidSymbolCountParameter'));

return;

end

% Check to make sure that CODE is binary

if any(code ~= 1 & code ~= 0)

error(message('comm:arithdeco:InvalidCodeParameter'));

return;

end

% Check to make sure that finite positive integer values (non-complex) are

% entered for COUNTS

if ~all(counts > 0) || ~all(isfinite(counts)) || ~isequal(counts, round(counts)) ||~isreal(counts)

error(message('comm:arithdeco:InvalidSymbolCountParameter'));

return;

end
```

```
% Check to make sure LEN is scalar

if ~isequal(size(len),[1 1])

error(message('comm:arithdeco:InvalidLength'));

return;

end

% Check to make sure that finite positive integer value (non-complex) is

% entered for LEN

if len < 1 || ~isfinite(len) || (len ~= round(len)) || ~isreal(len)

error(message('comm:arithdeco:InvalidLength'));

return;

end

% EOF
```

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายภูษิต เมฆนันทไพศิฐ

ภูมิลำเนา 1084/20 ถ.บรมไตรโลกนารถ 2 ต.ในเมือง อ.เมือง จ.พิษณุโลก

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: **new_heaven_hell@hotmail.com**