แอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด
APPLICATION DETECT JAUNDICE IN NEWBORN

ชุติมา ฟองอ่อน
CHUTIMA FONG-ON
ภคพล สุวิทย์เรืองฤทธิ์
PHAKAPON SUVITRUEANGRIT

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

แอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด
APPLICATION DETECT JAUNDICE IN NEWBORN

ชุติมา ฟองอ่อน
CHUTIMA FONG-ON
ภคพล สุวิทย์เรืองฤทธิ์
PHAKAPON SUVITRUEANGRIT

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

แอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด
APPLICATION DETECT JAUNDICE IN NEWBORN

นางสาวชุติมา ฟองอ่อน รหัสนักศึกษา 57010328
นายภคพล สุวิทย์เรืองฤทธิ์ รหัสนักศึกษา 57010945

อาจารย์ที่ปรึกษา
รศ. ดร. ชูชาติ ปิณฑวิรุจน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมชีวการแพทย์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชา วิศวกรรมชีวการแพทย์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด

Application Detect Jaundice in Newborn

ผู้จัดทำ

นางสาวชุติมา ฟองอ่อน รหัสนักศึกษา 57010328

นายภคพล สุวิทย์เรืองฤทธิ์ รหัสนักศึกษา 57010945

ปริญญานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

(รศ. ดร. ชูชาติ ปิณฑวิรุจน์)

อาจารย์ที่ปรึกษา

หัวข้อปริญญานิพนธ์	แอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด
นักศึกษา	นางสาวชุติมา ฟองอ่อน รหัสนักศึกษา 57010328
	นายภคพล สุวิทย์เรืองฤทธิ์ รหัสนักศึกษา 57010945
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมชีวการแพทย์
ปีการศึกษา	2560
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ. ดร. ชูชาติ ปิณฑวิรุจน์

บทคัดย่อ

ภาวะตัวเหลืองเป็นภาวะที่พบได้มากในเด็กทารกแรกเกิด ในการวินิจฉัยภาวะตัวเหลืองมีความจำเป็นที่จะต้องทำการเจาะเลือดหรือใช้อุปกรณ์เพื่อทำการวัดทางการแพทย์ ซึ่งก่อให้เกิดความเจ็บปวดและเกิดความเสี่ยงที่จะทำให้เกิดการติดเชื้อ โดยปริญญานิพนธ์นี้มีวัตถุประสงค์เพื่อพัฒนาวิธีการใหม่ในการวัดค่าความเหลืองของผิวหนังของเด็กทารกแรกเกิด โดยอาศัยความรู้ทางด้านการประมวลผลภาพดิจิตอล ทั้งยังนำไปพัฒนาร่วมกับสมาร์ทโฟนบนระบบปฏิบัติการแอนดรอยด์ในรูปแบบของแอพพลิเคชั่น โดยมีฟังก์ชันในการตรวจสอบสภาวะตัวเหลืองในเด็กแรกเกิดโดยถ่ายภาพบริเวณอกของเด็กทารกแล้วทำการประมวลผลเพื่อแสดงผลผ่านทางแอปพลิเคชัน ทั้งยังสามารถส่งข้อมูลเพื่อเก็บภายในฐานข้อมูลออนไลน์ เพื่อช่วยให้บุคลากรทางการแพทย์วินิจฉัยและติดตามตรวจสอบประวัติย้อนหลังได้ จากการทดลองกับเด็กทารกแรกเกิดจำนวน 30 ราย พบว่าค่าที่ได้จากแอปพลิเคชันและค่าที่ได้จากผลเลือดมีค่าสัมประสิทธิ์สหสัมพันธ์ที่ 0.825 และค่าเฉลี่ยบิลิรูบินที่ได้จากการตรวจเลือดคือ 9.113 mg/dL ค่าเฉลี่ยบิลิรูบินจากแอปพลิเคชันคือ 9.213 mg/dL ความคลาดเคลื่อนของการทดลองคือ 1.169% ความคลาดเคลื่อนมาตรฐานของการพยากรณ์คือ 1.35 พบว่ามีนัยสำคัญทางสถิติระหว่างค่าบิลิรูบินจากแอปพลิเคชันและค่าบิลิรูบินจากการตรวจเลือด ($P<0.05$)

Thesis Title	Application Detect Jaundice in Newborn
Student	Miss Chutima Tong-on ID 57010328
	Mr.Phakapon Suvitrueangrit ID 57010945
Degree	Bachelor of Engineering
Major	Biomedical Engineering
Year	2017
Thesis Advisor	Assoc. Prof. Dr. Chuchart Pintavirooj

ABSTRACT

Jaundice is highly common condition in newborn. In the diagnosis of jaundice, it is necessary to perform a blood test or to use a device for medical measurements which is causing the pain, and the risk that it will cause the infection. This research aimed to develop a new method for measuring neonatal jaundice of neonates based on the knowledge of Digital Image Processing and also developed with the android operating system on smartphone. This application has a function of detecting neonatal conditions by simply taking a picture of the newborn chest and then display result of estimate total transcutaneous digital bilirubin. Data collected of newborn can also be send to database to assist physicians for diagnosis and monitoring history of application used. We evaluated application on 30 newborns. The mean Microbilirubin = 9.113 mg/dL and mean total transcutaneous digital bilirubin (TcdB) = 9.213 mg/dL, coefficient of correlation = 0.825, error = 1.169%, standard error of estimate = 1.35, significant statistic between value of MB and TcdB was found ($P < 0.05$).

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ.ดร.ชูชาติ ปิณฑวิรุจน์ ที่คอยให้ความช่วยเหลือ ให้คำชี้แนะในการแก้ไขปัญหาต่างๆ รวมทั้งให้คำปรึกษาและแนะนำแนวทางการทำปริญญานิพนธ์นี้

ขอขอบพระคุณ คณะกรรมการการสอบหัวข้อและโครงร่างปริญญานิพนธ์ทุกท่าน ที่ได้กรุณาแนะนำตลอดจนชี้แนะแนวทาง จนทำให้ปริญญานิพนธ์สำเร็จลงได้

ขอขอบพระคุณ พยาบาลดูแลห้องพักหลังคลอดและหน่วยงานวิจัย โรงพยาบาลสิรินธร ที่ให้การสนับสนุนในการเก็บข้อมูลผลตรวจเลือดที่นำมาใช้ในการวิจัยในครั้งนี้ ทั้งขอขอบพระคุณผู้ปกครองของเด็กทารกแรกเกิดที่ยินยอมให้ทำการเก็บผลวิจัยในครั้งนั้

ขอขอบคุณ พี่และเพื่อน ๆ ที่คอยสนับสนุนให้ความช่วยเหลือทำให้ปริญญานิพนธ์สำเร็จลงได้

ชุติมา ฟองอ่อน
ภคพล สุวิทย์เรืองฤทธิ์

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VI

สารบัญภาพ.....VII

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 ข้อสมมุติฐาน.....1

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย..... 1

1.5 ขอบเขตของการวิจัย.....2

1.6 แผนการดำเนินงาน.....2

1.7 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 งานวิจัยที่เกี่ยวข้อง

2.1 ภาวะตัวเหลืองในเด็กแรกเกิด.....3

2.2 แนวทางปฏิบัติเมื่อพบว่าทารกมีภาวะตัวเหลือง.....6

2.3 การตรวจวัดค่าบิลิรูบิน.....9

2.4 แนวทางการรักษา.....11

2.5 การประมวลผลภาพทางดิจิทัลเบื้องต้น..... 17

2.6 ระบบสี (Color Model) และ ฮิสโตแกรม (Histogram).....21

2.7 การแปลงแอฟไฟน์ (Affine Transform)..... 25

2.8 ระบบพิกัดโฮโมจีเนียส (Homogeneous Coordinate).....25

2.9 ภาษา Java..... 27

2.10 การทำ Authentication และ Registration.....39

2.11 ระบบปฏิบัติการแอนดรอยด์ (Android).....40

2.12 การวิเคราะห์สหสัมพันธ์และการวิเคราะห์การถดถอย.....46

2.13 ประเภทของการวิเคราะห์สหสัมพันธ์และการถดถอย.....46
2.14 สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน 49
(Pearson product-moment correlation coefficient)

บทที่ 3 การออกแบบระบบ
3.1 การหาสมการ Total Transcutaneous Digital Bilirubin..... 51
3.2 Android Application.....52
3.3 Firebase..... 61
บทที่ 4 ผลการทดลองและอภิปรายผล
4.1 Preliminary Method.....63
4.2 ผลการวิเคราะห์สหสัมพันธ์และการถดถอย.....68
4.3 หน้าต่างแสดงผลการทำงานของแอปพลิเคชัน.....71
4.4 ผลการทดสอบและการวิเคราะห์ผลจากแอปพลิเคชัน.....73
4.5 ผลการวิเคราะห์ข้อมูล..... 74
บทที่ 5 บทสรุป
5.1 สรุป.....76
5.2 ข้อเสนอแนะ..... 76
บรรณานุกรม
ภาคผนวก

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดง cephalocaudal distribution of bilirubin..... 8
ตารางที่ 2.2 แสดงค่าที่ได้จากการตรวจ LFT.....10
ตารางที่ 2.3 ระดับ Serum bilirubin ที่ควรเริ่มให้การรักษาในทารก.....13
คลอดก่อนกำหนดหรือน้ำหนักตัวน้อย
ตารางที่ 2.4 แสดง Version และ API Level ของระบบปฏิบัติการต่างๆ.....42
ตารางที่ 3.1 ตัวอย่างตารางบันทึกผลค่าบิลริรูบินจากการตรวจ.....51
ทางห้องปฏิบัติการของเด็กแรกเกิด
ตารางที่ 4.1 แสดงข้อมูลที่ได้จากการเก็บผลและภาพถ่ายส่วนหน้าอกของเด็กทารก.....63
ตารางที่ 4.2 ผลวิเคราะห์สหสัมพันธ์.....68
ตารางที่ 4.3 ค่าสถิติที่ใช้พิจารณาความเหมาะสมของสมการถดถอยของตัวแปร..... 69
ตารางที่ 4.4 การตรวจสอบตัวแปรอิสระ ที่สามารถทำนายค่าของตัวแปร MB.....69
ตารางที่ 4.5 ค่าสัมประสิทธิ์การถดถอยของตัวแปรทั้งสามตัวแปรกับตัวแปร MB..... 70
ตารางที่ 4.6 แสดงข้อมูลเด็กทารกแรกเกิดที่ได้รับการทดสอบ..... 73
ตารางที่ 4.7 แสดงค่าความถี่ (Frequency) และค่าร้อยละ (Percentage) ข้อมูลทั่วไป.....74
ของผู้ได้รับการทดลอง จำแนกตามน้ำหนัก

สารบัญภาพ

หน้า

ภาพที่ 2.1 Bilirubin Metabolism..... .3
ภาพที่ 2.2 Dermal Zone..... 8
ภาพที่ 2.3 แผนผังการวินิจฉัยแยกโรค และสาเหตุของ Neonatal Jaundice.....9
ภาพที่ 2.4 Bilirubinometer ผลิตโดยบริษัท Reichert Technologies.....11
ภาพที่ 2.5 หลักการการเกิด Isomerization ของการรักษาวิธี Phototherapy..... 12
ภาพที่ 2.6 การใช้ Phototherapy ในทารกที่อายุครรภ์.....12
เท่ากับหรือมากกว่า 35 สัปดาห์
ภาพที่ 2.7 ทารกที่ทำการรักษาโดยวิธี Phototherapy.....14
ภาพที่ 2.8 การทำ Exchange transfusion ในทารกที่อายุครรภ์..... 15
เท่ากับหรือมากกว่า 35 สัปดาห์
ภาพที่ 2.9 การแปลงภาพอนาล็อคให้เป็นภาพดิจิทัล.....17
ภาพที่ 2.10 (a) ตัวอย่างภาพ Bitmap , (b) ตัวอย่างภาพแบบเวกเตอร์..... 18
ภาพที่ 2.11 ค่าในแต่ละพิกเซลของภาพชนิด Binary..... 19
ภาพที่ 2.12 ตัวอย่างค่าในแต่ละพิกเซลของภาพ grayscale..... 19
ภาพที่ 2.13 ภาพชนิด RGB.....20
ภาพที่ 2.14 ตัวอย่างการเก็บค่าสีแบบ Indexed Image.....20
ภาพที่ 2.15 RGB color cube..... 22
ภาพที่ 2.16 CMYK color cube.....23
ภาพที่ 2.17 การบรรยายสีในระบบ CIE Lab ในระนาบ 2 มิติ.....24
ภาพที่ 2.18 การบรรยายสีในระบบ CIE lab ในระนาบ 3 มิติ.....24
ภาพที่ 2.19 การแปลงเชิงเรขาคณิตชนิดต่างๆ..... 25
ภาพที่ 2.20 Java Logo.....27
ภาพที่ 2.21 ลักษณะของ Shared Prefernece..... 38
ภาพที่ 2.22 การใช้งาน Authentication ด้วยวิธีต่างๆ.....39
ภาพที่ 2.23 Firebase app Development.....39
ภาพที่ 2.24 Android Logo..... 41
ภาพที่ 2.25 โปรแกรม Android Studio.....42

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 2.26 บริการต่างๆของ Firebase Platform.....45
ภาพที่ 3.1 แผนผังแสดงการทำงานของแอปพลิเคชัน.....52
ภาพที่ 3.2 Layout Activity Login.....53
ภาพที่ 3.3 แผนผังขั้นตอนการทำงานใน Activity Login.....53
ภาพที่ 3.4 Layout Activity Sign up.....54
ภาพที่ 3.5 แผนผังขั้นตอนการทำงานใน Activity Sign up..... 54
ภาพที่ 3.6 Layout Activity 5.....55
ภาพที่ 3.7 แผนผังแสดงขั้นตอนการทำงานใน Activity 5.....55
ภาพที่ 3.8 Layout Activity 2.....56
ภาพที่ 3.9 แผนผังแสดงขั้นตอนการทำงานใน Activity 2.....56
ภาพที่ 3.10 Layout Activity 3.....57
ภาพที่ 3.11 แผนผังแสดงขั้นตอนการทำงานของ Activity 3..... 57
ภาพที่ 3.12 Layout Activity 4.....58
ภาพที่ 3.13 แผนผังแสดงขั้นตอนการทำงานใน Activity 4.....58
ภาพที่ 3.14 Layout Activity 6.....59
ภาพที่ 3.15 แผนผังแสดงขั้นตอนการทำงานใน Activity 6.....59
ภาพที่ 3.16 Project setting ของ Firebase..... 61
ภาพที่ 3.17 Authentication ระหว่าง Firebase และ Application.....61
ภาพที่ 4.1 ภาพแสดงความสัมพันธ์ระหว่างตัวแปร YM และ MB..... 68
ภาพที่ 4.2 หน้าต่าง Login และ Sign up..... 71
ภาพที่ 4.3 หน้าต่างสำหรับกรอกข้อมูลของเด็กแรกเกิด.....71
ภาพที่ 4.4 หน้าต่างการทดสอบและแสดงผลระดับบิลิรูบิน.....72
ภาพที่ 4.5 หน้าต่างแสดงข้อมูลสำหรับการอัปโหลด.....72
ภาพที่ 4.6 Scatter plot ระหว่างค่าบิลิรูบินที่ได้จากวิธีทางห้องปฏิบัติการกับแอปพลิเคชัน.....75

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ภาวะตัวเหลืองในเด็กแรกเกิด (Newborn Jaundice) เป็นปัญหาสำคัญและพบได้มาก ต้องได้รับการวินิจฉัยและรักษาในเวลาที่เหมาะสม โดยในทางการแพทย์ อาการตัวเหลืองเกิดจากการมีสารสีเหลืองที่เรียกว่า บิลิรูบิน (Bilirubin) จำนวนมากกว่าปกติคั่งอยู่ในอวัยวะต่าง ๆ ทำให้เห็นว่ามีผิวหนังทั่วตัวและตาขาวเป็นสีเหลือง ปัจจุบันค่าบิลิรูบินที่ได้จากการตรวจเลือดถือเป็นมาตรฐานในการวินิจฉัยและการรักษาซึ่งมีความจำเป็นที่จะต้องมีการตรวจเลือดทารกหรือใช้อุปกรณ์วัดทางการแพทย์ซึ่งไม่สามารถใช้งานได้นอกเหนือการตั้งค่าการใช้งานทางการแพทย์ อีกทั้งการเจาะเลือดทำให้เกิดความเจ็บปวดและอาจทำให้เกิดการติดเชื้อ ซึ่งส่งผลเสียต่อทารกได้ ผู้จัดทำจึงหาวิธีการวัดค่าความเหลืองทางผิวหนังในเด็กแรกเกิด โดยวิธีการทางการประมวลผลภาพและจัดทำแอปพลิเคชันสำหรับถ่ายภาพและประมวลผลภาพทางดิจิตอลเป็นระดับบิลิรูบินที่วัดได้ทางผิวหนัง

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 เพื่อพัฒนาวิธีการใหม่ในการวัดค่าความเหลืองของผิวหนังเด็กแรกเกิด

1.2.2 เพื่อที่จะนำความรู้ทางด้านการประมวลผลภาพมาใช้เพื่อเกิดประโยชน์แก่ส่วนรวม

1.2.3 เพื่อเพิ่มความสะดวกสบายและประหยัดเวลาในการวินิจฉัยทางการแพทย์

1.2.4 เพื่อออกแบบแอปพลิเคชัน สำหรับการภาพถ่ายและประมวลผลระดับบิลิรูบิน

1.2.5 เพื่อเพิ่มความสะดวกแก่ผู้ปกครองในกรณีที่ไม่ได้เข้ารับการตรวจทางการแพทย์

1.3 ข้อสมมุติฐาน

1.3.1 ค่าสี Y-M ในระบบ CMYK มีความสัมพันธ์กันกับระดับบิลิรูบินที่ได้จากผลทางห้องปฏิบัติการ

1.3.2 ระดับบิลิรูบินที่ได้จากการทดสอบแอปพลิเคชันมีความสัมพันธ์กับระดับบิลิรูบินที่ได้จากผลทางห้องปฏิบัติการ

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

การศึกษาค้นคว้าเพื่อการทำแอปพลิเคชันสำหรับตรวจวัดระดับบิลิรูบินผ่านทางผิวหนังนี้ มีส่วนสำคัญมาจากหลักการการวิเคราะห์และประมวลผลภาพดิจิตอล (Digital Image Processing)

1.5 ขอบเขตของการวิจัย

1.5.1 ภาวะตัวเหลืองในเด็กแรกเกิด (Newborn Jaundice)

1.5.1.1 สารบิลิรูบิน (Bilirubin)

1.5.1.2 การตรวจเลือดเพื่อหาสารบิลิรูบิน (Bilirubin Test)

1.5.1.3 ค่า Micro bilirubin ในเด็กที่มีภาวะตัวเหลือง

1.5.2 แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ (Android)

1.5.2.1 กระบวนการทาง Image Processing

1.5.3.1 Open CV

1.5.3 ความสัมพันธ์ทางสถิติของระดับบิลิรูบินโดยวิธีทางการแพทย์และการใช้ภาพดิจิตอล

1.5.4 การประมวลภาพทางดิจิตอล (Digital Image Processing)

1.5.4.1 ระบบสี (Color Model)

1.5.4.2 การหาความแปรปรวนของสีในภาพ

1.6 แผนการดำเนินงาน

แผนการดำเนินงาน	ภาคการศึกษาที่ 1/2560					ภาคการศึกษาที่ 2/2560		
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ค้นคว้าและรวบรวมข้อมูลที่เกี่ยวข้องกับภาวะตัวเหลืองในทารกแรกเกิดและการประมวลผลภาพ								
2. ออกแบบและสร้างแอปพลิเคชันสำหรับการประมวลผลภาพดิจิตอล								
3. เก็บข้อมูลภาพถ่ายทารกแรกเกิดและนำมาคำนวณหาสมการเพื่อพยากรณ์ระดับบิลิรูบิน								
4. ทดสอบและคำนวณค่าความผิดพลาด								
5. ปรับปรุงแอปพลิเคชันและทำการทดสอบซ้ำ								
6. ทดลองและวิเคราะห์ผลการทดลอง								
7. จัดทำรูปเล่มรายงาน								

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1.7.1 เกิดประโยชน์สำหรับบุคลากรทางการแพทย์และผู้ที่ต้องการวินิจฉัยการเกิดภาวะตัวเหลืองในเด็กแรกเกิด

1.7.2 พัฒนาแอปพลิเคชันสำหรับตรวจวัดค่าบิลิรูบินเพื่อระบุภาวะตัวเหลืองในเด็กแรกเกิด

1.7.3 สามารถใช้เป็นแนวทางในการพัฒนาแอปพลิเคชันต่อไปในอนาคตได้

1.7.4 เกิดประโยชน์สำหรับผู้ปกครองที่ต้องการตรวจวัดระดับบิลิรูบินเบื้องต้น

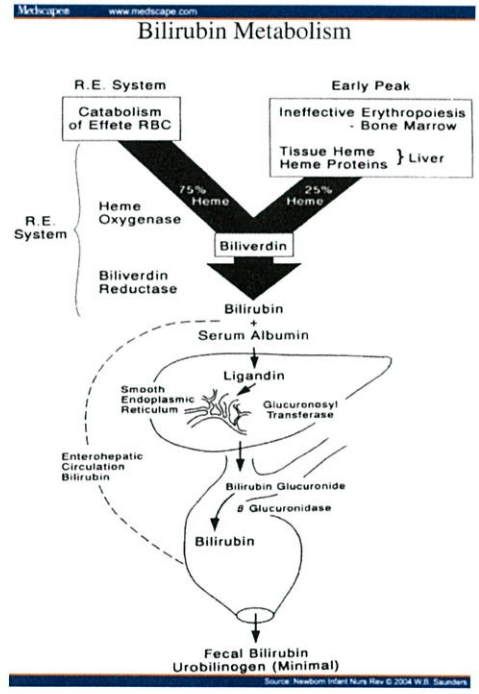
บทที่ 2
งานวิจัยที่เกี่ยวข้อง

2.1 ภาวะตัวเหลืองในเด็กแรกเกิด (Neonatal Jaundice)

ภาวะตัวเหลืองเป็นปัญหาที่พบบ่อยที่สุดในทารกแรกเกิด ในสัปดาห์แรกหลังเกิดทารกเกือบทุกคนจะมีระดับ Bilirubin สูงกว่าค่าปกติในผู้ใหญ่ และมากกว่า 2 ใน 3 ของทารกแรกเกิดจะมีอาการตัวเหลืองที่เห็นได้ด้วยตาเปล่า ส่วนใหญ่แล้วเป็นภาวะที่พบได้ในทารกปกติโดยไม่ก่อให้เกิดอันตราย (physiologic jaundice) แต่มีทารกบางรายที่มีภาวะตัวเหลืองมากกว่าปกติ (pathologic jaundice) จากสาเหตุต่างๆ ได้หลายประการและระดับ Unconjugated Bilirubin ที่สูงมากสามารถทำให้เกิดความผิดปกติของสมองอย่างถาวรได้ การวินิจฉัยแยกภาวะทั้งสองออกจากกันเพื่อเลือกให้การรักษาที่เหมาะสม จึงเป็นปัญหาที่สำคัญ ในการดูแลทารกแรกเกิด

2.1.1 Bilirubin metabolism

Bilirubin สร้างมาจากการแตกทำลายของเม็ดเลือดแดงใน Reticuloendothelial system (RE system) ได้ เป็น Unconjugated หรือ Indirect Bilirubin และจะต้องถูกกำจัดออกจากร่างกาย Indirect Bilirubin ถูกเปลี่ยนแปลงต่อโดยกระบวนการดังนี้



ภาพที่ 2.1 Bilirubin Metabolism

จาก ภาพที่ 2.1 จะเห็นได้ว่า Indirect Bilirubin จับกับ Albumin ในเลือดเพื่อส่งไปที่ตับ และตับจะทำหน้าที่ Conjugate Bilirubin โดยอาศัยเอนไซม์หลักคือ Uridine diphosphate glucuronosyl transferase (UDPG-T) ให้เป็น Conjugated หรือ Direct Bilirubin ซึ่งสามารถละลายน้ำได้และจะถูกขับออกสู่ลำไส้ทางน้ำดี Direct Bilirubin จะถูกเปลี่ยนแปลงต่อโดยแบคทีเรียในลำไส้ให้เป็นสารที่ถูกขับออกจากร่างกายได้ทางอุจจาระ และถ้าถูกกำจัดออกช้าจะถูกเอนไซม์ Beta Glucuronidase ซึ่งยังมีอยู่มากในลำไส้ของทารกที่อยู่ในครรภ์และทารกแรกเกิด และอาจจะเกิด Deconjugate ทำให้ Direct Bilirubin กลับไปเป็น Indirect Bilirubin และจะถูกดูดซึมกลับเข้าไปในกระแสเลือดได้อีก เรียกเหตุการณ์นี้ว่า Enterohepatic Circulation

2.1.2 ประเภทของการเกิดภาวะตัวเหลืองในเด็กแรกเกิด

Neonatal Jaundice เป็นภาวะตัวเหลืองที่พบได้ในทารกแรกเกิดถึงร้อยละ 25-50 แบ่งได้เป็น 2 ประเภทคือ

2.1.2.1 Physiologic jaundice ในทารกปกติระดับของ Indirect Bilirubin ใน Cord Blood จะมีค่าประมาณ 1.4-1.9 mg/dL หลังเกิดระดับจะเพิ่มขึ้น ช้าๆ ไม่เกินวันละ 5 mg/dL โดยจะเริ่มมองเห็นว่าเหลืองหลัง 24 ชั่วโมง และเหลืองสูงสุดประมาณวันที่ 4-5 หลังเกิด ในทารกครบกำหนดระดับสูงสุดจะอยู่ประมาณ 12-15 mg/dL หรือมากกว่าในทารกชาวเอเชีย หลังจากนั้นจะค่อยๆ ลดลงและหายไปในเวลา ประมาณ 10-14 วันหลังเกิด ส่วนในทารกเกิดก่อนกำหนด Bilirubin จะคงอยู่ในระดับสูงสุดไปถึงประมาณวันที่ 7-10 หลังเกิด และระดับสูงสุดอาจมากกว่าที่พบในทารกครบกำหนด ด้วย

สาเหตุของการเกิด Physiologic jaundice

1. ทารกมีการสร้าง Bilirubin มากกว่าผู้ใหญ่ เนื่องจากว่า
   1.1 มี Hematocrit (HCT) สูงกว่าผู้ใหญ่ ปริมาณเม็ดเลือดแดงต่อน้ำหนักตัวจึงมากกว่า
   1.2 เม็ดเลือดแดงทารกมีอายุสั้นกว่าคือประมาณ 90 วัน ซึ่งของผู้ใหญ่คือ 120 วัน
   1.3 มีเม็ดเลือดแดงที่สร้างไม่สมบูรณ์ (Ineffective erythropoiesis) มากกว่า
   1.4 มีการแตกทำลาย (tum over) ของ heme protein เร็วกว่า
2. ตับทารกมีการ uptake Bilirubin ได้น้อยกว่า conjugate และขับ Bilirubin ออกจากตับได้ช้ากว่าผู้ใหญ่
3. มี enterohepatic circulation มากกว่า เนื่องจากยังมี beta glucuronidase activity ที่สูง และมี bacteria ในทางเดินอาหารน้อยกว่าผู้ใหญ่

2.1.2.2 Pathologic jaundice เป็นภาวะตัวเหลืองที่มากกว่าปกติ ซึ่งในบางครั้งอาจจะแยกออกจาก physiologic jaundice ได้ยาก เนื่องจากเกิดอยู่ในช่วงอายุเดียวกัน แต่จำเป็นต้องแยกจากกันเพื่อจะได้ตรวจหาสาเหตุและให้การรักษาต่อไป สามารถแบ่งได้เป็น 2 กลุ่มใหญ่ คือ

1. Conjugated hyperbilirubin สาเหตุการเกิดอาจเป็นจากนอกตับ (extrahepatic) หรือจากในตับ (intrahepatic)

2. Unconjugated hyperbilirubin สาเหตุที่ทำให้เกิด ได้แก่

2.1 มีการสร้าง Bilirubin เพิ่มขึ้น

2.1.1 ภาวะ hemolysis เช่น ABO, Rh, minor blood group incompatibility, G6PD deficiency

2.1.2 การมีเลือดออกในร่างกาย เช่น cephalhematoma หรือ hematoma ในบริเวณอื่น หรือเลือดออกในลำไส้

2.1.3 มีภาวะเลือดข้น (polycythemia)

2.2 มี enterohepatic circulation เพิ่มขึ้น ยกตัวอย่างเช่น

2.2.1 การงดอาหารหรือได้รับอาหารทางลำไส้มีปริมาณน้อย

2.2.2 ลำไส้เคลื่อนไหวน้อยหรือมีการอุดตัน (bowel ileus / obstruction)

2.3 ตับกำจัด Bilirubin ได้น้อยกว่าปกติ ยกตัวอย่างเช่น

2.3.1 ทารกคลอดก่อนกำหนด

2.3.2 G6PD deficiency

2.3.3 Inborn error of metabolism เช่น Crigler-Najiar, Galactosemia

ลักษณะที่บ่งชี้ว่าทารกมีภาวะตัวเหลือง ชนิด pathologic jaundice ได้แก่

1. เริ่มตัวเหลืองเมื่ออายุน้อยกว่า 24 ชั่วโมง
2. อัตราการเพิ่มของ Bilirubin มากกว่า 5 mg/dL ต่อวัน หรือมากกว่า 0.2 mg/dL ต่อชั่วโมง
3. ระดับ serum Bilirubin มากกว่า 12-15 mg/dL
4. ตัวเหลืองนานเกินกว่า 14 วัน โดยเฉพาะในทารกที่ไม่ได้กินนมแม่
5. ระดับ conjugated Bilirubin มากกว่า 2mg/dL

2.1.3 สาเหตุของ Pathological jaundice ที่พบบ่อย

สาเหตุของ Pathological jaundice ที่พบบ่อย ได้แก่ hemolysis, G6PD และภาวะตัวเหลืองที่สัมพันธ์กับการกินนมแม่

2.1.3.1 Hemolytic disease of the newborn (HDN) เป็นภาวะที่มีการแตกทำลายของเม็ดเลือดแดง (hemolysis) มากผิดปกติ เป็นผลให้ทารกมีอาการซีดและตัวเหลืองมากกว่าปกติ สาเหตุเกิดจากมารดาและทารกมีหมู่เลือดที่ไม่ตรงกันและมี IgG antibody ที่ผ่านจากมารดาไปสู่ทารกทำให้เกิด hemolysis ในทารก อาจเป็น ABO, Rh หรือ minor blood group incompatibility

2.1.3.2 G6PD deficiency โรคนี้มีอุบัติการณ์สูงในประเทศไทย เป็นสาเหตุที่พบบ่อยอย่างหนึ่งของภาวะตัวเหลือง ผิดปกติในทารกเอเชีย ทารกอาจ เริ่มมีอาการเหลืองกว่าปกติได้ตั้งแต่วันแรกๆหลังเกิดหรือค่อยๆเหลืองมากขึ้นในวันหลังๆก็ได้ สาเหตุเป็นได้ทั้งจาก hemolysis

หรือไม่มี hemolysis ก็ได้ ส่วนใหญ่ในทารกแรกเกิดมักไม่พบสาเหตุที่ทำให้hemolysisชัดเจนอย่างไรก็ตามควรซักประวัติการ สัมผัสสิ่ง กระตุ้นเช่น การใช้ยาหรือลูกเหม็นไว้ด้วย

การตรวจทางห้องปฏิบัติการ จะพบว่ามี G6PD deficiency ในรายที่มี hemolysis จะมี Hct ต่ำ, reticulocyte count สูง และperipheral blood smear พบ basket cell , blister cell หรือ bite cell แต่ถ้าไม่มี hemolysis จะพบว่า Hct, reticulocyte count และ blood smear ปกติ

2.1.3.3 ภาวะตัวเหลืองที่สัมพันธ์กับการกินนมแม่ แบ่งได้เป็น 2 ชนิด คือ

1. Breast feeding jaundice ทารกที่กินนมแม่มักมีระดับ Bilirubin สูงกว่าทารกที่กินนมผสมในช่วงอายุ 3-4 วันแรก สาเหตุเกิดจากการที่นมแม่ยังมีปริมาณไม่มากทารกได้รับนมน้อยจึงทำให้มี enterohepatic circulation มาก ภาวะนี้จะหายไป เมื่อทารกได้รับนมในปริมาณที

2. Breast milk jaundice เป็นภาวะที่พบได้ประมาณร้อยละ 10 ของทารกที่กินนมแม่ มักเกิดหลังอายุ 5 วันไปแล้ว ทารก จะมีอาการปกติ ดูดนมแม่ได้ดี น้ำหนักเพิ่มดี ถ่ายอุจจาระสีเหลืองปกติ แต่ตัวเหลืองไม่หายหลังจากเลยช่วง physiologic jaundice ไปแล้ว บางรายอาจเหลืองเพิ่มขึ้นและเหลืองอยู่ได้นาน 3-12 สัปดาห์ สาเหตุยังไม่มีข้อสรุปที่แน่ชัด เชื่อว่าอาจเป็นจาก สารบางตัวในนมแม่ การจะวินิจฉัยว่าทารกที่ตัวเหลืองนานกว่า 3 สัปดาห์ว่าเกิดจาก breast milk jaundice ต้องตรวจร่างกายหรือตรวจทาง ห้องปฏิบัติการเพื่อแยกสาเหตุอื่นที่ทำให้ตัวเหลืองมากและนานออกไป โดยเฉพาะ hypothyroidism, biliary atresia และ G6PD deficiency เป็นต้น

2.2 แนวทางปฏิบัติเมื่อพบว่าทารกมีภาวะตัวเหลือง

แนวทางการปฏิบัติเมื่อพบว่าทารกมีภาวะตัวเหลือง ควรทำตามขั้นตอนดังต่อไปนี้ ได้แก่

2.2.1 ซักประวัติที่เกี่ยวข้องกับอาการตัวเหลือง ได้แก่

1. ประวัติทารก

1.1 อายุครรภ์ของทารก และน้ำหนักแรกเกิด ทารกที่อายุครรภ์น้อยกว่า 38 สัปดาห์ จะเสี่ยงต่อการเกิดตัวเหลืองที่รุนแรงกว่าทารกที่มี Gestational age (GA) มากกว่า 38 สัปดาห์

1.2 อายุของทารกเป็นชั่วโมงขณะตรวจ มีความสำคัญอย่างยิ่ง ที่จะบอกถึงความรุนแรงและเป็นข้อมูลสำคัญใน การเลือกให้การรักษาแก่ทารกเนื่องจากระดับ บิลิรูบินจะเพิ่มขึ้นตามอายุเป็นชั่วโมงที่เพิ่ม การประเมินโดยนับอายุเป็นวันจะทำให้เกิดความผิดพลาดมาก

1.3 วิธีการคลอด การบาดเจ็บจากการคลอด ถ้ามีเลือดออกฟกช้ำใต้ผิวหนังเป็นสาเหตุให้เหลืองมากขึ้น

1.4 อาการทั่วไปของทารกหลังเกิด อาการผิดปกติที่บ่งชี้ว่าทารกอาจมีภาวะ Bilirubin Encephalopathy หรือ sepsis เช่น อาการซึมเศร้า ดูดนมไม่ดี ร้องเสียงแหลม เป็นต้น

1.5 ประวัติการกินนม ทารกที่กินนมแม่จะมีปัญหาตัวเหลืองได้บ่อยกว่าทารกที่กินนมผสม โดยเฉพาะอย่างยิ่งถ้านมแม่ยังไหลน้อย มีปัญหาหัวนมแตกหรือเจ็บ ได้นมแม่น้อยกว่า 8 มื้อต่อวัน ใช้เวลาดูดนมเกิน 40 นาทีต่อมื้อ ทำให้ทารกได้รับนมไม่เพียงพอจนเกิดมีปัญหา

breastfeeding jaundice ส่วนทารกที่กินนมแม่ได้ดีแต่เหลืองนานอาจเป็นจาก breast milk jaundice

1.6 การขับถ่ายอุจจาระ ปัสสาวะ ทารกที่ได้รับนมเพียงพอควรจะถ่ายปัสสาวะได้อย่างน้อย 4-6 ครั้งต่อวัน ถ่ายอุจจาระอย่างน้อย 3-4 ครั้งต่อวัน และลักษณอุจจาระเปลี่ยนจากสีเทาไปเป็นสีเหลืองภายหลังจากวันเกิด 3-4 วัน ถ้าถ่าน้อยกว่าปกติจะเสี่ยงต่อตัวเหลืองมากขึ้น ส่วนทารกที่มีอุจจาระสีซีดควรคำนึกถึงภาวะ obstructive jaundice

2. ประวัติมารดา

2.1 หมู่เลือดและ Rh ของมารดาอาจจะบ่งชี้ถึงภาวะ HDN

2.2 โรคที่เกิดในมารดาระหว่างตั้งครรภ์ที่สามารถส่งผลให้ตัวเหลือง เช่น เบาหวาน

2.3 ยาที่มารดาได้รับระหว่างตั้งครรภ์และการคลอด เช่น ยากลุ่ม sulfa, oxytocin

3. ประวัติครอบครัว

3.1 การมีบุตรคนก่อนหน้าตัวเหลืองหลังเกิด

3.2 ประวัติคนในครอบครัวที่เป็นโรคตับ โรคเกี่ยวกับความผิดปกติทางเลือด

3.3 เชื้อชาติ ชาวเอเชียจะมีผิวเหลืองมากกว่า

2.2.2 การตรวจร่างกาย

1. การประเมินระดับความเหลืองของทารกโดยทั่วไปจะสังเกตเห็นว่าทารกเริ่มเหลืองที่บริเวณหน้าก่อนแล้วจึงเห็นที่ลำตัวและแขนขาเมื่อระดับ Bilirubin สูงขึ้น (cephalocaudal progression)โดยใช้นิ้วกดผิวหนังแล้วดูสีผิวบริเวณที่ถูกกดจะเห็นสีเหลืองชัดขึ้น ในทารกครบกำหนดอาจใช้ดูด้วยสายตาเพื่อประมาณระดับ Bilirubin จาก (ภาพที่ 2.2, ตารางที่ 2.1) แต่การประมาณด้วยวิธีนี้มีความคลาดเคลื่อนได้มาก ขึ้นกับประสบการณ์ของผู้ตรวจและสีผิวของทารก จึงควรใช้เป็นเพียงแนวทางเพื่อบอกว่าทารกรายใดควรได้รับการตรวจหาระดับ Bilirubinในเลือดต่อไปปัจจุบันมีอุปกรณ์ที่ใช้วัดระดับตัวเหลืองจากผิวหนัง (transcutaneous Bilirubinometer) ใช้ได้ดีสำหรับการคัดกรองหาทารกที่เหลืองมาก แต่ค่าวัดที่ได้อาจแตกต่างจากการ ตรวจเลือดอยู่บ้างและอุปกรณ์นี้ยังมีราคาค่อนข้างแพง

2. การตรวจร่างกายทารก เพื่อหาความผิดปกติในระบบอวัยวะอื่น ซึ่งอาจจะสัมพันธ์กับภาวะตัวเหลือง

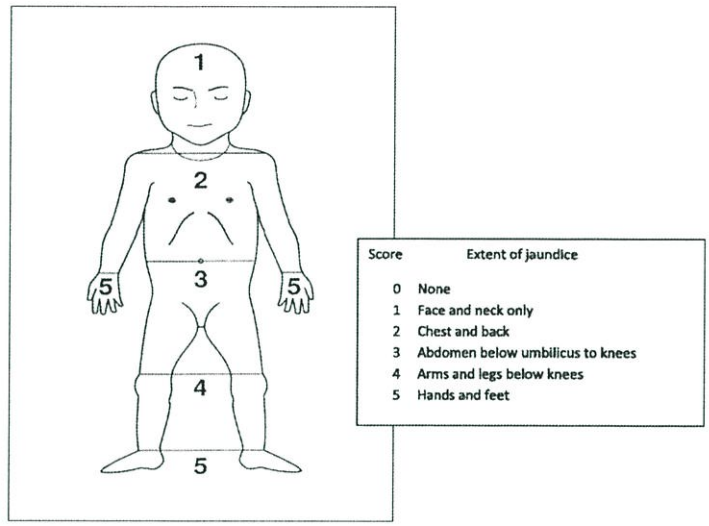
2.1 อาการทั่วไป เช่น ซึมเศร้า, อุณหภูมิสูงหรือต่ำกว่าปกติ, ตัวอ่อน (hypotonia) ดูดนมไม่ดี (poor sucking), Moro reflex ลดลง อาจพบในภาวะ sepsis, metabolic disturbance หรือเป็นอาการระยะแรกของ Bilirubin encephalopathy

2.2 น้ำหนักตัวเทียบกับแรกคลอด โดยทั่วไปในสัปดาห์แรก ทารกครบกำหนดจะมีน้ำหนักลดลงประมาณ 6 – 8 % ถ้าลดลงมากกว่าที่ควรจะเป็น แสดงว่าทารกอาจจะได้รับนมไม่เพียงพอ

2.3 ผิวหนังตรวจหาภาวะซีด (anemia) ซึ่งจะพบในรายที่มีการแตกทำลายของเม็ดเลือดแดงมากๆ (severse HDN) ภาวะตัวแดง (polycythemia) และการมีเลือดออกใต้ผิวหนังมากๆ (ecchymosis, hematoma) ซึ่งเป็นสาเหตุทำให้ทารกตัวเหลืองได้ การพบจ้ำเลือดตามตัว (petechiae, purpura) อาจเกิดจากการติดเชื้อในครรภ์ (TORCH infection)

2.4 ศีรษะ ตรวจหาภาวะเลือดออกที่หนังศีรษะ (cephalhematoma, subgaleal hematoma) วัดขนาดเส้นรอบศีรษะ (OFC) เพื่อบอกภาวะ microcephaly หรือ hydrocephalus จากการติดเชื้อในครรภ์ คลำความตึงและขนาดของ fontanelle ถ้า anterior fontanelle โป่งตึงนึกถึงภาวะ intracranial hemorrhage หรือ CNS infection

2.5 ท้อง ตรวจหาภาวะท้องอืด สะดืออักเสบ



ภาพที่ 2.2 Dermal Zone

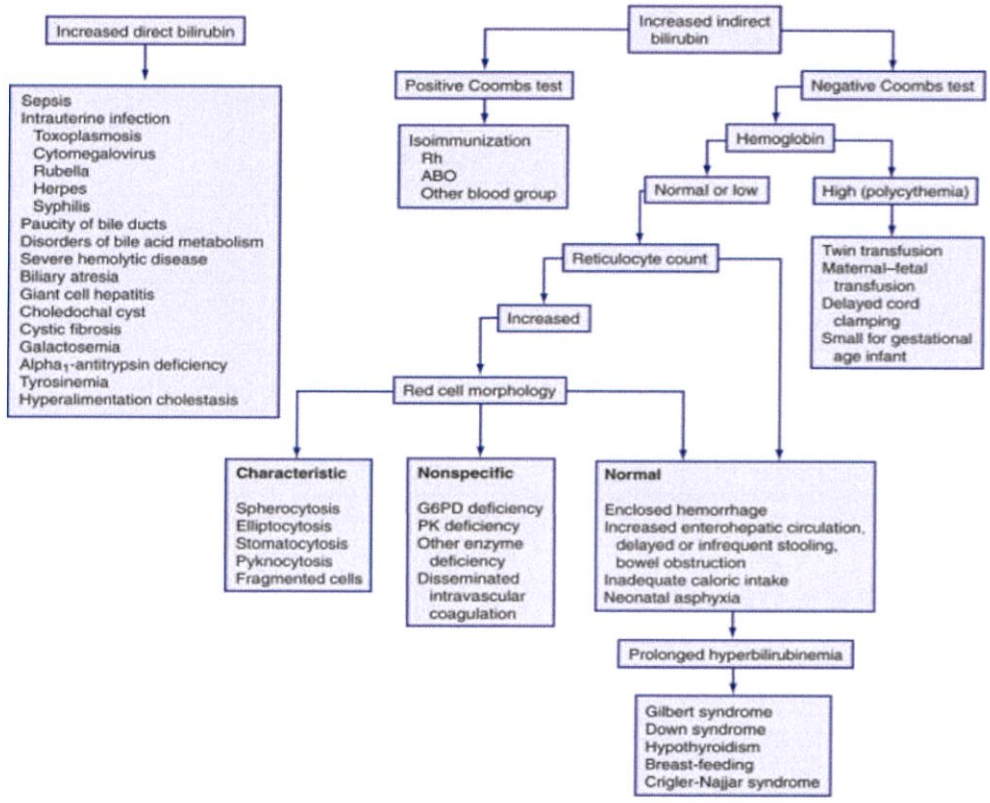
ตารางที่ 2.1 แสดง cephalocaudal distribution of Bilirubin

Zone	Jaundice	Serum indirect Bilirubin (mg/dL)	
		Average	Maximum
1	Face and Neck only	6	8
2	Chest and back	9	12
3	Abdomen below umbilicus to knees	12	16
4	Arms and legs below knees	15	18
5	Hands and feet	>15	

2.2.3 การตรวจทางห้องปฏิบัติการ

1. การตรวจเลือดทารก ควรส่งตรวจ HCT, Total Bilirubin, direct Bilirubin, blood group, Rh, Coombs' test, G6PD, peripheral blood smear

2. การตรวจเลือดมารดา ควรส่งตรวจ blood group, Rh, และ Coombs'test



ภาพที่ 2.3 แผนผังการวินิจฉัยแยกโรค และสาเหตุของ Neonatal Jaundice

2.3 การตรวจวัดค่าบิลิรูบิน

หากแพทย์วินิจฉัยเบื้องต้นว่าทารกมีภาวะตัวเหลือง จะทำการตรวจหาระดับบิลิรูบินในเลือด โดยสามารถใช้วิธีการดังต่อไปนี้

2.3.1 Bilirubin Blood Test

ในวิธีการทางห้องปฏิบัติการ จะมีการทำการตรวจหาระดับบิลิรูบินจากการตรวจ Liver function test (LFT) ซึ่งเป็นตรวจเพื่อหาความผิดปกติของการทำงานในตับ เนื่องจากตับมีหน้าที่เกี่ยวข้องกับการทำงานของระบบต่างๆในร่างกาย การตรวจดู LFT สามารถแสดงค่าต่างๆดังนี้

ตารางที่ 2.2 แสดงค่าที่ได้จากการตรวจ LFT

	ค่าปกติของผลเลือด
Total Protein	6.0 – 8.0 g/dL
Albumin	3.5 – 5.0 g/dL
Globulin	1.5 – 3.2 g/dL
Total Bilirubin	0.1 – 1.0 mg%
Direct Bilirubin	0.1 – 0.3 mg%
AST (SGOT)	8 – 50 U/L
ALT (SGPT)	8 – 50 U/L
ALP (Alk. Phosphatase)	35 – 110 U/L

ค่าต่างๆเหล่านี้ สามารถแบ่งเป็นกลุ่มๆได้ดังนี้

1. กลุ่มผลเลือดที่บ่งชี้ถึงสภาวะเซลล์ตับ หรือเอนไซม์ (Enzyme) ตับ ค่าที่ใช้วินิจฉัย ได้แก่ AST (SGOT) , ALT (SGPT) , Alkaline Phosphatase (ALP) การที่มีค่าเอนไซม์เหล่านี้ โดยเฉพาะค่า SGOT และ SGPT ที่สูงนั้น บ่งชี้ได้ว่า เซลล์ตับกำลังมีการอักเสบหรือแตกสลายอยู่ จึงทำให้เอนไซม์ตับซึ่งปกติจะอยู่แต่ภายในเซลล์ตับเท่านั้น หลุดออกมาในกระแสเลือดมากขึ้น อีกทั้งค่าเอนไซม์ ALP ที่สูงขึ้นจนผิดปกติก็สามารถพบได้ในกรณีที่มีก้อนเนื้อแอบแฝงเบียดในตับได้ จำเป็นที่จะต้องตรวจเพิ่มเติม เช่น อัลตราซาวน์ (Ultrasound) เป็นต้น

2. กลุ่มผลเลือดที่บ่งชี้เกี่ยวกับการสังเคราะห์และขับน้ำดี ค่าที่ใช้วินิจฉัย ได้แก่ Total Bilirubin (T. Bilirubin) , Direct Bilirubin (D. Bilirubin) , Indirect Bilirubin (ปกติแล้วจะไม่มีการเขียนแสดงค่าในใบรายงานผล แต่สามารถคำนวณจากสูตร Indirect Bilirubin = T. Bilirubin – D. Blirubin)

การแปลผลจะเน้นที่ค่า Direct (Conjugated) Bilirubin และ Indirect (Unconjugated) Bilirubin โดยทั่วไปแล้วเวลาที่เม็ดเลือดแดงถูกทำลาย Hemoglobin จะเกิดการสลายตัว ได้เป็น Unconjugated Bilirubin (Indirect Bilirubin) ดังนั้น ค่าของ Indirect Bilirubin ที่มากกว่าปกติอาจบ่งชี้ถึงสภาวะเม็ดเลือดแดงแตกมากผิดปกติ จนเกิดภาวะตัวเหลือง ตาเหลือง มากกว่าที่จะเป็นปัญหาโดยตรงของตับ หลังจากการสลายตัว Unconjugated Bilirubin จะถูกส่งไปที่ตับ ผ่านกระบวนการกำจัดทิ้งโดยการ Conjugate ทำให้กลายสภาพเป็น Conjugated Bilirubin ซึ่งละลายน้ำได้ดีขึ้นเพื่อส่งต่อไปขจัดทิ้งทางท่อน้ำดีออกไปรวมกับอุจจาระ ทำให้อุจจาระมีสีเหลือง การที่เซลล์ตับมีการอักเสบส่งผลให้ไม่สามารถเอา Unconjugated Bilirubin เข้าไปในเซลล์เพื่อทำการ Conjugate ได้ ซึ่งก็อาจจะเป็นสาเหตุที่ทำให้ค่า Indirect Bilirubin ในเลือดสูงผิดปกติได้เช่นกัน

3. กลุ่มผลเลือดที่บ่งชี้เกี่ยวกับความสามารถของตับในการสร้างโปรตีน ค่าที่ใช้วินิจฉัย ได้แก่ Albumin (Alb), Globulin (Glob), Protein (Prot) เนื่องจากการทำงานของตับคือการสร้างโปรตีนหลายๆชนิด ถ้าตับทำงานแย่ลงหรือเสื่อมสภาพ เช่น ตับอักเสบ, ตับแข็ง ก็จะทำให้มีความสามารถในการสร้างโปรตีนลดลง ถ้าตรวจเลือดก็จะพบว่าค่าของ Albumin, Globulin, Protein ต่ำลงเช่นกัน

4 กลุ่มผลเลือดที่บ่งชี้เกี่ยวกับความสามารถของตับในการสร้างสารที่ทำให้เลือดแข็งตัว สารที่ช่วยให้เลือดแข็งตัวก็คือโปรตีนชนิดหนึ่ง ถ้าความสามารถในการสร้างสารต่างๆ ของตับลดลงมากจนไม่สามารถสร้างโปรตีนในกลุ่มนี้ได้มากพอ ก็จะมีปัญหาเลือดออกง่ายในผู้ป่วยกลุ่มนี้ตามมา

2.3.2 Transcutaneous Bilirubin Test

การตรวจวัดค่า Bilirubin ในปัจจุบันมีเครื่องมือที่ได้วัดค่าชนิด Non-invasive คือ Bilirubinometer โดยไม่ต้องใช้ตัวอย่างเลือดเพื่อทำการวินิจฉัย แต่จะใช้หลักการของ Light-emitting sensor ส่องไปที่ผิวของเด็กแรกเกิดบริเวณ Sternum หรือ Forehead ลำแสงที่สะท้อนกลับมาจะแยกออกเป็นสองลำโดย dichroic mirror และความยาวคลื่น 455 nm และ 575 nm จะถูกวัดจาก optical detectors แต่เครื่องมือวัดค่านี้ยังมีราคาที่สูงมาก จึงใช้ไม่แพร่หลายในการวินิจฉัยภาวะตัวเหลืองในเด็กแรกเกิด อ้างอิงข้อมูลจาก World Health Organization เครื่อง Bilirubinometer มีราคาอยู่ที่ 3,100 – 7,000 USD และระยะเวลาการใช้งาน (product life time) 6 – 8 ปีเท่านั้น

ภาพที่ 2.4 Bilirubinometer ผลิตโดยบริษัท Reichert Technologies

2.4 แนวทางการรักษา

การรักษา neonatal jaundice มีวัตถุประสงค์เพื่อ ลดหรือป้องกันระดับ unconjugated Bilirubin ไม่ให้สูงถึงระดับที่ทำให้เกิด Bilirubin encephalopathy และลดการเกิดภาวะแทรกซ้อนจากการรักษา หลีกเลี่ยงการรักษาที่มีภาวะแทรกซ้อนมากถ้าทำได้ เช่น การเปลี่ยนถ่ายเลือด วิธีการรักษาภาวะตัวเหลืองมีทั้งหมด 3 วิธี ได้แก่

2.4.1 รักษาโดยการส่องไฟ (Phototherapy)

หลักการของ Phototherapy คือการรักษาด้วยลำแสงที่มีความยาวคลื่น (wavelength) อยู่ในช่วง 420 – 475 nm. แสงความยาวคลื่นนี้จะเปลี่ยน Indirect Bilirubin ที่ผิวหนังให้เป็น isomer อื่นๆ เรียกกระบวนการนี้ว่า Isomerization หรืออาจเป็นสารอื่นซึ่งละลายน้ำได้และไม่เป็นอันตรายต่อสมอง สามารถขับออกทางร่างกาย ทางอุจจาระและปัสสาวะได้

1. ข้อบ่งชี้ในการรักษาด้วย Phototherapy

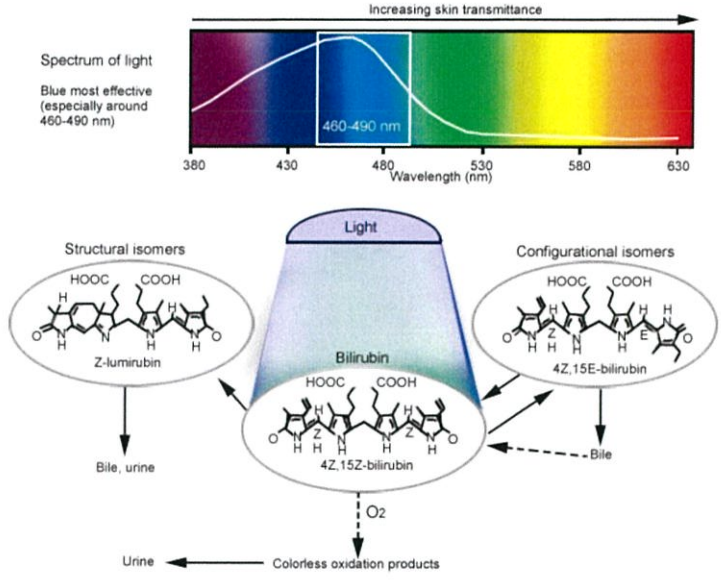
1.1 ในทารกที่อายุครรภ์ 35 สัปดาห์ขึ้นไป พิจารณาตามอายุครรภ์ทารก อายุหลังเกิดเป็นชั่วโมง และปัจจัยเสี่ยงของทารก

1.2 ในทารกอายุครรภ์น้อยกว่า 35 สัปดาห์หรือน้ำหนักตัวน้อย ให้พิจารณา ตารางที่ 2.3

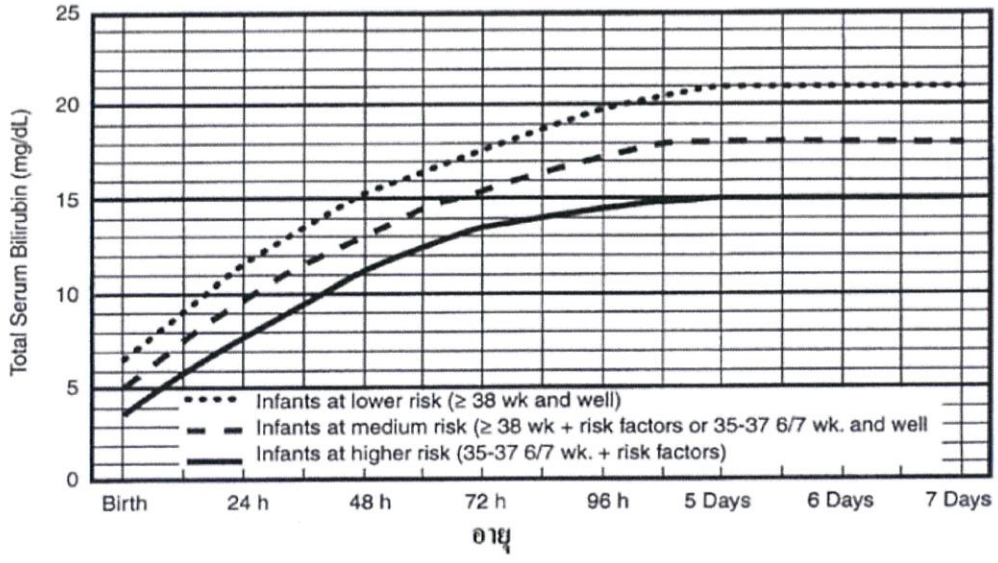
2. ข้อห้ามในการทำ Phototherapy

2.1 ทารกที่เป็น Direct hyperBilirubinemia

2.2 ทารกที่มีประวัติครอบครัวเป็น Light sensitive porphyria



ภาพที่ 2.5 หลักการการเกิด Isomerization ของการรักษาวิธี Phototherapy



ภาพที่ 2.6 การใช้ Phototherapy ในทารกที่อายุครรภ์เท่ากับหรือมากกว่า 35 สัปดาห์

Risk factors ได้แก่ isoimmune hemolytic disease (ABO, Rh, minor blood group incompatibility), G6PD deficiency, asphyxia (Apgar score ที่ 5 นาที < 7 หรือมีอาการเช่น hypoxic ischemic encephalopathy, renal impairment), significant lethargy. temperature instability, sepsis, acidosis, albumin < 3.0 g/dL

ตารางที่ 2.3 ระดับ Serum Bilirubin ที่ควรเริ่มให้การรักษาในทารกคลอดก่อนกำหนดหรือน้ำหนักตัวน้อย

น้ำหนักตัว (กรัม)	Serum Bilirubin level (mg/dL)	
	Uncomplicated	Complicated*
<1000	12-13	10-12
1001-1250	12-14	10-12
1251-1499	14-16	12-14
1500-1999	16-20	15-17
2000-2500	20-22	18-20

* Complicated คือทารกที่มีภาวะ Perinatal asphyxia, acidosis, hypoxia, hypothermia, hypoalbuminemia, meningitis, intraventricular hemorrhage, intraventricular hemorrhage, hemolysis, hypoglycemia, Bilirubin encephalopathy

3. วิธีปฏิบัติในการทำ Phototherapy

3.1 ใช้ผ้าทึบแสงปิดตาให้มิดชิด เพื่อป้องกันอันตรายต่อจอประสาทตา (Retina)

3.2 ให้ทารกสวมเสื้อผ้าน้อยชิ้น เพื่อให้ผิวหนังได้สัมผัสกับแสงมากที่สุด

3.3 วัดอุณหภูมิร่างกายทุก 4 ชั่วโมง เฝ้าระวังและแก้ไขภาวะอุณหภูมิร่างกายสูงหรือต่ำกว่าปกติ

3.4 ทารกควรได้รับนมหรือสารน้ำเพิ่มขึ้นกว่าปริมาณปกติ ประมาณร้อยละ 20 -30 เพื่อชดเชยการเสียน้ำจากการระเหยทางผิวหนังและอุจจาระที่มากขึ้น ควรพิจารณาให้สารน้ำทางหลอดเลือดดำ ในทารกที่มีอาการขาดน้ำ (dehydrate) และกินอาหารได้น้อย

3.5 ชั่งน้ำหนักทารกทุกวัน ติดตามจำนวนครั้งที่ถ่ายอุจจาระเพื่อเฝ้าระวังภาวะขาดน้ำ

3.6 ตรวจเลือดเพื่อหาระดับ TSB ทุก 12-24 ชั่วโมงหรือบ่อยกว่านี้เป็นทุก 4-6 ชั่วโมงถ้าทารกมีระดับ TSB สูงใกล้ระดับที่จะต้องทำ exchange transfusion เมื่อทารกได้รับ phototherapy แล้ว จะไม่สามารถใช้การประเมิน ระดับ TSB ด้วยสายตาหรือใช้ transcutaneous Bilirubinometer ได้ เพราะระดับBilirubinที่ผิวจะลดลงทำให้ดูทารกเหลืองน้อยลงกว่าระดับที่แท้จริง

3.7 หลอดไฟที่ใช้ในเครื่องส่องไฟมีหลายชนิด ที่ใช้ทั่วไปและหาง่ายราคาถูกคือ daylight fluorescent ใช้ 4-8 หลอด วางทารก ห่างแผงหลอดไฟประมาณ30ซม.แต่ประสิทธิภาพใน การลดระดับ Bilirubin น้อยกว่าหลอด special blue (F20T12BB ของ General Electrics,TL52/20W ของ Phillips,FL18W/T8/DB หรือ deep blue lamp ของ Toshiba) ข้อเสียของแสงสีฟ้าคืออาจ

ทำให้ ผู้ปฏิบัติงานรู้สึกมึนงง ไม่สบายตา คลื่นไส้และทำให้ดูไม่ออกว่าทารกเขียวหรือไม่ จึงอาจเลือกใช้หลอด special blue light ตรงกลาง และหลอด day light อยู่ด้านริมเพื่อช่วยลดผลข้างเคียงต่อผู้ปฏิบัติงานลง ส่วนหลอดไฟชนิด halogen มีประสิทธิภาพใกล้เคียงกับ daylight แต่ส่องได้เป็นพื้นที่แคบและมีความร้อนสูงเป็นอันตรายต่อทารกได้ง่ายและหลอดขาดบ่อยจึงไม่นิยมใช้ ปัจจุบันหลอดที่มี ประสิทธิภาพดีที่สุดและร้อนน้อยคือ light-emitting diodes (LED) แต่ยังมีราคาแพง ส่วนfiber optic light ซึ่งส่องแสงผ่านแผ่น พลาสติกที่ใช้ห่มหรือวางใต้ตัวทารกได้โดยไม่ร้อน ราคาแพงและใช้ได้ดีกับทารกน้ำหนักน้อยๆ ส่วนทารกครบกำหนดมักใช้ในกรณีที่ เหลืองไม่มากนัก

3.8 Intensive phototherapy หมายถึงการส่องไฟทารกด้วยหลอดไฟที่ให้ความยาวคลื่น 430-490 nm. ด้วยระดับความเข้มแสง 30 -35 $\mu N/cm^2/nm$ ขึ้นไป และเมื่อทำแล้วควร ช่วยลดระดับ total serum Bilirubin ให้ได้อย่างน้อย 0.5 –1 mg/dL/hr ใน 4-8 ชั่วโมงแรก

3.9 ตรวจความเข้มของแสงจากเครื่องส่องไฟ ให้อยู่ในระดับที่มากกว่า standard phototherapy หรือ intensive phototherapy อยู่เสมอ เนื่องจากหลอดไฟที่ใช้งานไปแล้วจะให้แสงที่มีความเข้มลดลงเรื่อยๆ

4. ภาวะแทรกซ้อนจาก Phototherapy

4.1 จอประสาทตาถูกทำลาย (retina damage)

4.2 ผ้าปิดตาเลื่อนไปปิดจมูกทารก ส่งผลให้หายใจไม่ได้

4.3 ถ่ายเหลว แต่จะเกิดขึ้นเพียงชั่วคราวเท่านั้น และจะหายเมื่อหยุดส่องไฟ

4.4 การขาดน้ำจากการสูญเสียน้ำทางผิวหนังและถ่ายเหลวมากขึ้น

4.5 ผื่นแดงตามผิวหนัง (nonspecific erythematous rash)

4.6 อุณหภูมิร่างกายสูงหรือต่ำเกินไป

4.7 ผิวหนังไหม้จากความร้อนของหลอดไฟ (burn)

4.8 Bronze baby จากการส่องไฟในทารกที่เป็น direct hyperbilirubinemia

5. ข้อบ่งชี้ในการหยุดทำ Phototherapy

ทารกสามารถหยุดส่องไฟได้เมื่อระดับ Bilirubin ต่ำกว่า 13-15 mg/dL

ภาพที่ 2.7 ทารกที่ทำการรักษาโดยวิธี Phototherapy

2.4.2 การรักษาโดยการเปลี่ยนถ่ายเลือด (Exchange transfusion)

หลักการของการรักษาโดยเปลี่ยนถ่ายเลือด คือ การเอาเลือดของทารกที่มีระดับ Bilirubin สูง และ antibody ที่ทำให้เกิด hemolysis ออกจากร่างกายแล้วใส่เลือดใหม่ทดแทน เมื่อใช้เลือด 2 เท่าของปริมาณเลือดที่ทารกมีอยู่ในร่างกายจะเปลี่ยนเลือดทารกออกไปได้ประมาณร้อยละ 75-85 หลังทำ ระดับของ Bilirubin จะลดลงเหลือประมาณร้อยละ 50 ของระดับก่อนทำ หลังจากนั้น 30-45 นาที Bilirubin ในเนื้อเยื่อจะออกมาสู่เลือดทำให้เพิ่มเป็นร้อยละ 60-75 ของระดับก่อนทำ

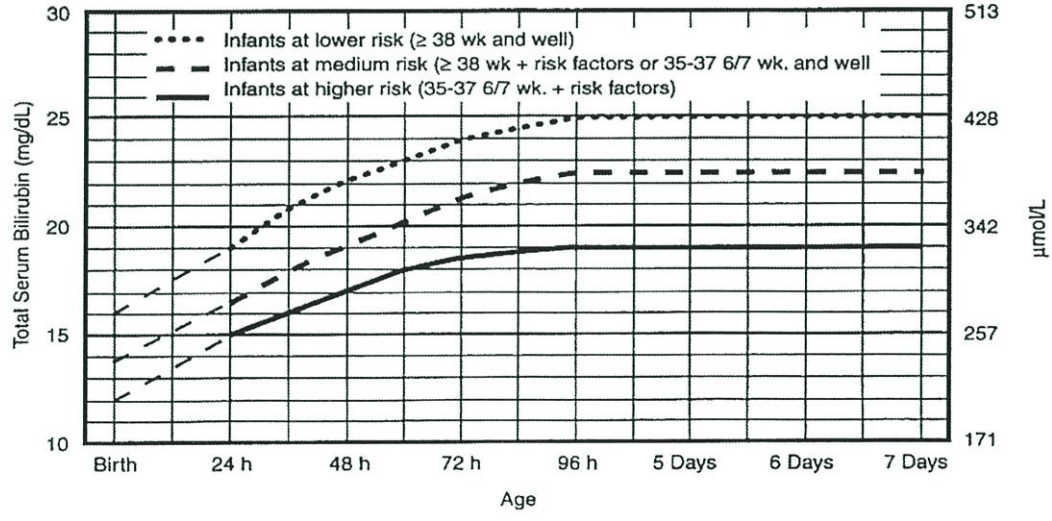
1. ข้อบ่งชี้ของการรักษาด้วย Exchange transfusion

1.1 ในทารกอายุครรภ์น้อยกว่า 35 สัปดาห์หรือน้ำหนักตัวน้อย ให้พิจารณาตารางที่ 2.4

1.2 ในทารกที่อายุครรภ์ 35 สัปดาห์ขึ้นไป พิจารณาตามอายุครรภ์ทารก อายุหลังเกิดเป็นชั่วโมง และปัจจัยเสี่ยงของทารก ในภาพที่ 2.8

1.2.1 ทารกที่มีอาการของ acute Bilirubin Encephalopathy หรือมีระดับ Serum Total Bilirubin สูงกว่าระดับในภาพที่ 2.8 มากกว่าหรือเท่ากับ 5 mg/dL ควรได้รับการทำ exchange transfusion ทันทีโดยไม่ต้องรอดูผลของการทำ intensive phototherapy

1.2.1 ทารกที่ไม่มีอาการของ Bilirubin encephalopathy และ serum total Bilirubin สูงกว่าระดับในภาพน้อยกว่า 5 mg/dL ให้เริ่ม intensive phototherapy ทันทีและตรวจวัดระดับ Bilirubin ทุก 2-3 ชั่วโมง ร่วมกับเฝ้าระวังอาการของ acute Bilirubin encephalopathy ด้วย ถ้าระดับ serum total Bilirubin ยังสูงกว่าระดับ ในภาพ หลังได้รับintensive phototherapy ไปแล้วหรือทารกเริ่มมีอาการของ Bilirubin encephalopathy ก็ควรทำ exchange transfusion [5]



ภาพที่ 2.8 การทำ Exchange transfusion ในทารกที่อายุครรภ์เท่ากับหรือมากกว่า 35 สัปดาห์

2. วิธีปฏิบัติ

2.1 เลือดที่จะใช้ทำ Exchange transfusion ควรเป็นเลือดใหม่สุดที่จะหาได้ โดยไม่ควรเก็บมาไว้นานเกิน 5 วัน

2.2 ปริมาณเลือดที่ใช้เป็นซีซี คือ 2 x 80 x น้ำหนักตัวทารก (กิโลกรัม)

2.3 ชนิดของเลือดที่ใช้

2.3.1 ถ้ามารดา group O ทารก group A หรือ B เลือกใช้ packed red cell (PRC) group O ผสมกับ fresh frozen plasma (FFP) group AB หรือ PRC group O ผสมกับ FFP group เดียวกับทารก

2.3.2 ถ้ามารดา Rh negative ทารก Rh positive ต้องใช้เลือด Rh negative

2.3.3 ถ้าเป็น HDN จากสาเหตุอื่นๆ ต้องใช้เลือดที่ cross match แล้วเข้าได้กับทั้งเลือดแม่และเลือดลูก

2.4 ดูดเลือดออกในปริมาณไม่เกิน 10 ซีซี ต่อน้ำหนักตัวทารกเป็นกิโลกรัมแล้วใส่เลือดเข้าไปทดแทนในปริมาณเท่ากัน ระยะเวลาที่ใช้ทำทั้งหมดควรจะประมาณ 60 – 90 นาที

2.5 ตรวจหาระดับ Hct, TB ก่อนทำจากเลือดที่ดูดจากตัวทารกครั้งแรก หลังทำประมาณครึ่งชั่วโมงและ 4 – 6 ชั่วโมงต่อมา ส่วนการส่งเลือดเพื่อทำ hemoculture, liver function test, electrolyte ก่อนและหลังทำนั้นควรพิจารณาเป็นรายๆ ตามความเหมาะสม

2.6 หลังทำเสร็จแล้ว ทารกควรได้รับ Phototherapy ต่อไปจนกว่าจะตรวจซ้ำแล้วแน่ใจว่าระดับ Bilirubin ลดลงอยู่ในเกณฑ์ที่ปลอดภัยแล้ว

3. ภาวะแทรกซ้อนจาก Exchange transfusion

3.1 ระบบหัวใจและระบบหายใจ คือ อาจเกิดการหยุดหายใจ, hypovolvemia, volume overload, หัวใจเต้นผิดจังหวะหรือ หัวใจหยุดเต้น

3.2 ระบบหลอดเลือดอาจเกิดลิ่มเลือดอุดตัน (thrombosis embolism) หรือเส้นเลือดตีบเกร็ง (vasospasm)

3.3 ระบบโลหิต อาจเกิดเกร็ดเลือดต่ำ เลือดแข็งตัวผิดปกติ (coagulopathy), ซีด, เม็ดเลือดแดงแตก (hemolysis) และภาวะ graft versus host reaction

3.4 ระบบทางเดินอาหาร อาจเกิดเส้นเลือดในท้องทะลุ (hemoperitomuen), portalhypertension, ลำไส้ทะลุ และลำไส้อักเสบ (NEC)

3.5 การติดเชื้ออาจเกิด sepsis, ติดเชื้อ CMV, HIV, hepatitis

3.6 ภาวะอุณหภูมิกายต่ำ

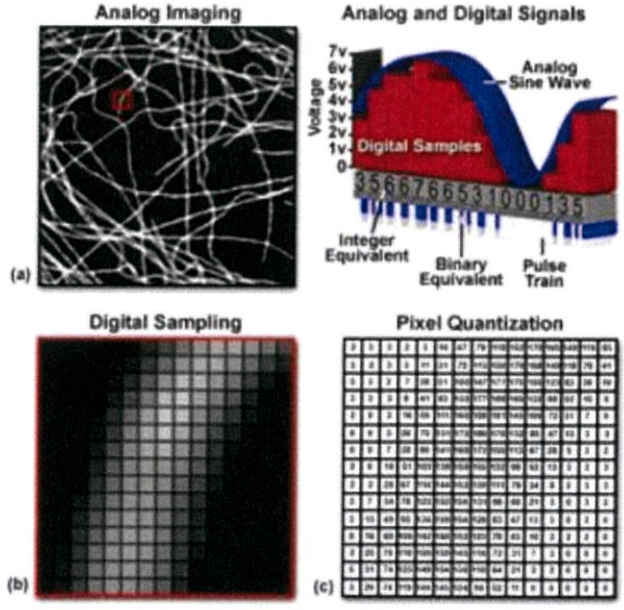
2.4.3 การรักษาด้วยยา (Pharmacological agents)

2.5 การประมวลผลภาพทางดิจิทัลเบื้องต้น

การประมวลผลภาพทางดิจิตอล (Digital image processing) หรือที่เรียกว่าการถ่ายภาพด้วยระบบคอมพิวเตอร์ (Computer imaging) หมายถึงข้อมูลที่ได้มาจากกระบวนการ Acquisition และการประมวลผลข้อมูลภาพด้วยคอมพิวเตอร์ ความสำคัญของการประมวลผลภาพดิจิตอลมาจากข้อเท็จจริงที่ว่า “Our primary sense is our visual sense” ระบบการมองเห็นของเราช่วยให้เราสามารถรวบรวมข้อมูลได้โดยไม่ต้องมีปฏิสัมพันธ์ทางกาย ช่วยให้เราสามารถวิเคราะห์ข้อมูลทุกประเภทได้โดยตรงจากรูปภาพหรือวิดีโอ ช่วยให้เราสามารถนำทางสภาพแวดล้อมของเราและระบบภาพของมนุษย์เป็นระบบประสาทที่ทันสมัยที่สุดในร่างกายมนุษย์ การค้นพบและความก้าวหน้าทางวิทยาศาสตร์ส่วนใหญ่ได้อาศัยระบบภาพสำหรับการพัฒนาเทคโนโลยีต่างๆ

กระบวนการนี้เกี่ยวข้องกับการแปลงข้อมูลรูปที่เป็นสัญญาณอนาล็อค (Analog signal) ให้อยู่ในรูปของสัญญาณดิจิทัล (Digital signal) เพื่อใช้ในการประมลผลผ่านทางคอมพิวเตอร์ได้ และยังสามารถนำมาใช้ในการลดปัญหาของภาพได้ เช่น การลดสัญญาณรบกวนในภาพ การเพิ่มความคมชัดของภาพ (Image sharpening) เป็นต้น

ในการแปลงภาพให้เป็นสัญญาณดิจิทัลนั้น ระบบจะนำรูปที่รับเข้ามาไปคำนวณ โดยผ่านกระบวนการ Sampling และ Quantization และส่งข้อมูลออกมาในรูปแบบดิจิทัล คอมพิวเตอร์จะเก็บข้อมูลภาพลงหน่วยความจำ โดยการจองหน่วยความจำภายในเครื่องในรูปแบบของอาร์เรย์ (Array) โดยค่าในแต่ละช่องอาร์เรย์แสดงถึงคุณสมบัติต่างๆของรูปที่จุดพิกเซล (Pixel) นั้นๆ และตำแหน่งของช่องอาร์เรย์ก็เป็นตัวกำหนดตำแหน่งของจุดพิกเซลภายในภาพด้วยเช่นกัน



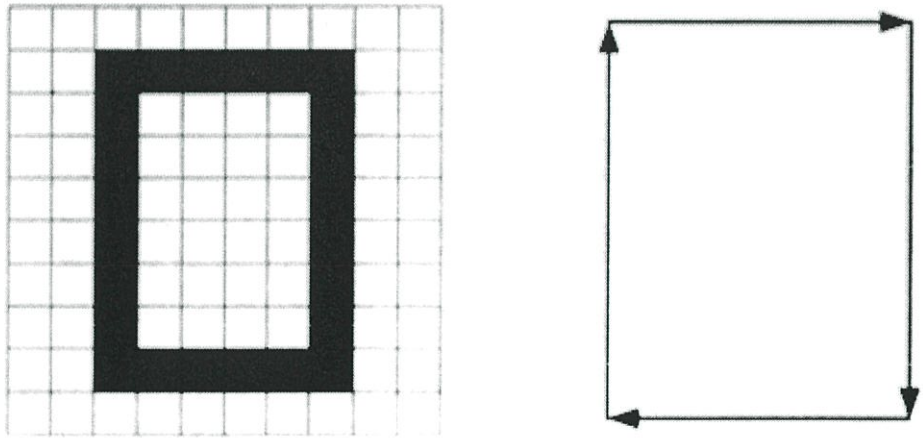
ภาพที่ 2.9 การแปลงภาพอนาล็อคให้เป็นภาพดิจิทัล

2.5.1 ชนิดของภาพที่ใช้งานบนเครื่องคอมพิวเตอร์

โดยทั่วไปแล้วสามารถแบ่งรูปภาพที่จะกฎและใช้งานบนเครื่องคอมพิวเตอร์ออกเป็น 2 ประ - เภท คือ บิตแมป (Bitmap Image) และ เวกเตอร์ (Vector Image) โดยรูปภาพแบบบิตแมปและพิจารณาตัวรูปภาพซึ่งถูกแบ่งออกเป็นส่วนย่อยเล็กๆ หลายๆ ส่วน หรือที่เรียกว่า พิกเซล (Pixel) ที่ถูกนำมารวมกันและใช้แสดงผลภาพ ส่วนรูปภาพแบบเวกเตอร์จะประกอบด้วยเส้นลายต่างๆ ที่สร้างขึ้นจากการคำนวณทางคณิตศาสตร์ของลักษณะทางเรขาคณิตเพื่อสร้างรูปทรงต่างๆ ซึ่งเรียกว่า เวกเตอร์ (Vector)

1. ภาพชนิดบิตแมป (Bitmap Image)

ภาพบิตแมปหรือเรียกอีกอย่างหนึ่งว่าภาพแบบราสเตอร์ (Raster Image) เป็นภาพที่ประกอบขึ้นจากจุดที่มีขนาดเล็กๆ หรือที่เรียกว่าพิกเซลจำนวนมากที่เรียงต่อกันจนเป็นภาพภาพหนึ่ง ภาพบิตแมปนี้จะมีจำนวนจุดขนาดเล็กๆ จำนวนมาก ดังนั้นดวงตาของมนุษย์ไม่สามารถที่จะมองเห็นและแยกแยะรายละเอียดส่วนย่อยเล็กๆ นั้นได้ แต่เมื่อลองขยายภาพดูจะเห็นเป็นรูปตาราง ยิ่งขยายใหญ่เท่าไร ตารางสี่เหลี่ยมก็ยิ่งมีขนาดใหญ่ขึ้นจนทำให้มองเห็นจุดของภาพหรือพิกเซล จึงมีผลทำให้รายละเอียดของภาพมีความไม่ชัดเจนมากขึ้น โดยทั่วไปแล้วภาพบิตแมปเป็นภาพประเภทที่นิยมใช้กันมากในภาพถ่าย



(a) (b)

ภาพที่ 2.10 (a) ตัวอย่างภาพ Bitmap , (b) ตัวอย่างภาพแบบเวกเตอร์

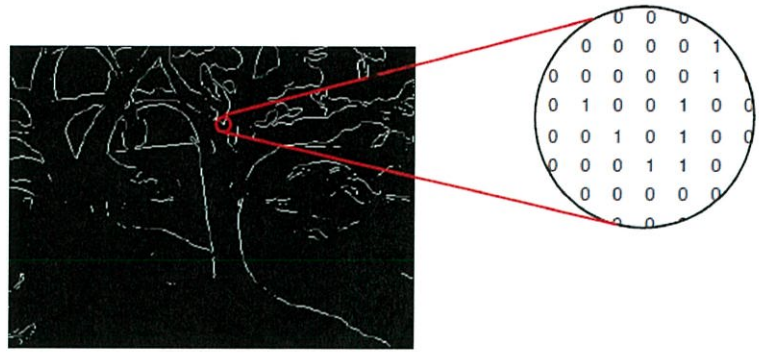
2. ภาพแบบเวกเตอร์ (Vector Image)

ภาพแบบเวกเตอร์จะมีคุณสมบัติที่แตกต่างกับแบบบิตแมป คือ ภาพประเภทนี้ไม่ว่าจะขยายใหญ่แค่ไหน ก็จะยังคงรายละเอียดละความคมชัดไว้ได้เช่นเดิม โดยไม่ผิดเพี้ยน เนื่องจากภาพแบบเวกเตอร์นั้นประกอบไปด้วยเส้นตรง เส้นโค้ง และรูปทรงต่างๆ ภาพที่ได้จะสร้างขึ้นจากคำสั่งที่บอกถึงลักษณะของภาพ ในรูปแบบทางเรขาคณิตด้วยสมการทางคณิตศาสตร์

2.5.2 ประเภทของภาพ (Image Types)

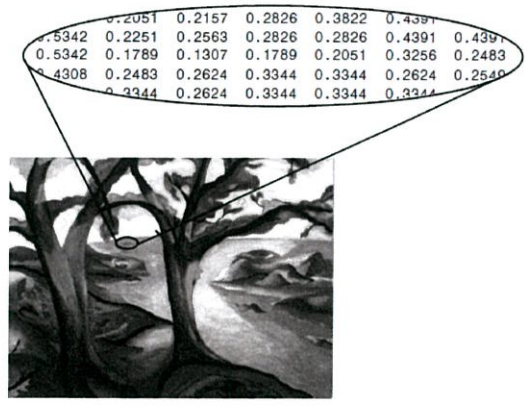
โดยทั่วไปเราสามารถแบ่งประเภทของภาพบิตแมป (bitmap) ตามคุณสมบัติของสีออกได้เป็น 4 ประเภท ดังนี้

1. Binary Image หรือภาพขาวดำ เป็นภาพที่ใช้เนื้อที่เพียง 1 บิต ต่อ พิกเซล โดยค่าสีจะมีแค่สองค่าคือ 0 และ 1 ดังแสดงในภาพที่ 2.11



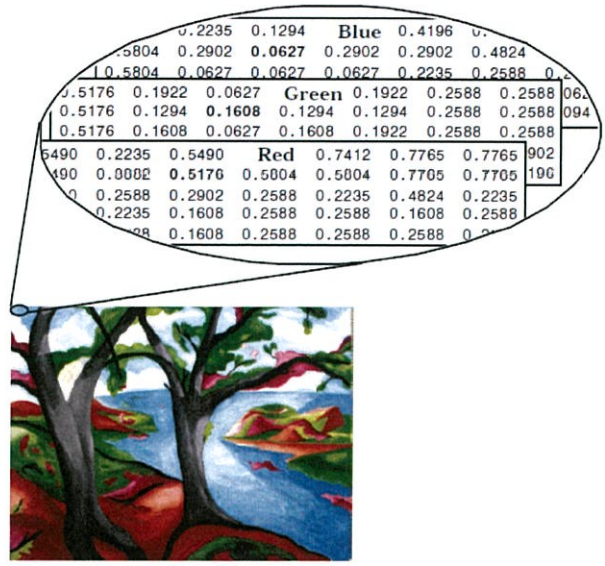
ภาพที่ 2.11 ค่าในแต่ละพิกเซลของภาพชนิด Binary

2. Grayscale Image เป็นภาพที่เก็บโดยใช้รูปแบบของอาร์เรย์ 2 มิติ โดยค่าในแต่ละจุดภาพคือค่าความเข้มของสีแต่ละตำแหน่งของพิกเซลนั้นๆ ซึ่งค่าที่เป็นไปได้ของภาพระดับเทาทั้งหมดขึ้นอยู่กับจำนวนบิตที่ใช้



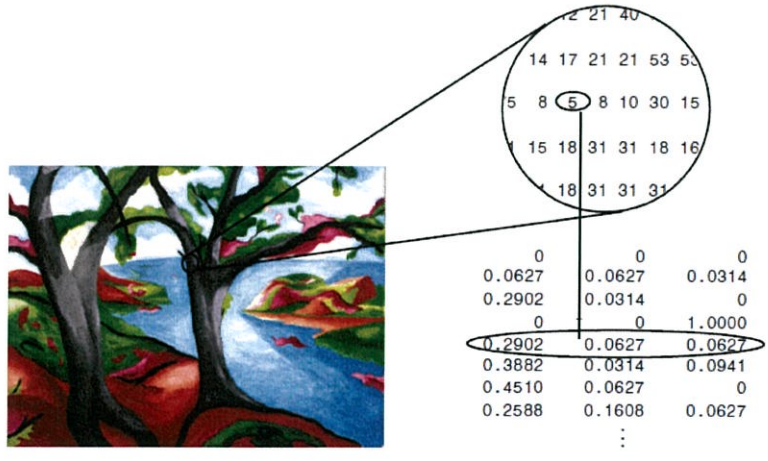
ภาพที่ 2.12 ตัวอย่างค่าในแต่ละพิกเซลของภาพ grayscale

3. RGB Image หรือ Truecolor Image เป็นรูปที่เก็บโดยใช้อาร์เรย์ 3 มิติ ขนาด m x n x 3 โดยที่ m คือความยาว และ n คือความกว้างของภาพในหน่วยพิกเซล ส่วนมิติสุดท้ายนั้น ในแต่ละมิติจะเก็บค่าสีแยกกัน คือ สีแดง (Red) , สีเขียว (Green), สีน้ำเงิน (Blue)



ภาพที่ 2.13 ภาพชนิด RGB

4. Indexed Image เป็นรูปที่มีรูปแบบการเก็บแบบ indexed คือ ภาพประเภทนี้จะเก็บค่าสีเป็น indexed และในแต่ละช่องอาร์เรย์ จะเก็บตำแหน่งของสีใน index นั้นๆไว้



ภาพที่ 2.14 ตัวอย่างการเก็บค่าสีแบบ Indexed Image

2.6 ระบบสี (Color Model) และ ฮิสโตแกรม (Histogram)

แบบจำลองสี เป็นวิธีที่กำหนดสีต่างๆ ให้เป็นแบบมาตรฐาน ซึ่งแบบจำลองสีในแต่ละชนิดก็จะมีคุณสมบัติแตกต่างกันออกไป จึงเหมาะสมสำหรับการใช้งานที่แตกต่างกัน ในแบบจำลองของสีนั้นจะใช้สีหลักๆที่เรียกว่า แม่สี มาผสมผสานกันเพื่อใช้ในการแสดงค่าสีอื่นๆ โดยแม่สีหลักนั้นจะแตกต่างกันออกไปตามแบบจำลองแต่ละชนิด โดยในงานวิจัยฉบับนี้ จะกล่าวถึง แบบจำลองสี RGB , CMYK, L*a*b

2.6.1 ปัจจัยที่ทำให้เกิดสี

สี เป็นคุณสมบัติเชิงแสงที่สามารถใช้บรรยายคุณลักษณะของวัสดุได้ง่ายที่สุดวิธีหนึ่ง ในการอธิบายสีของวัตถุด้วยคำพูด มาตรฐานของการบรรยายลักษณะสีอาจจะแตกต่างกัน ขึ้นอยู่กับประสบการณ์ ลักษณะทางกายภาพของตาของผู้บรรยาย ลักษณะแสงที่ตกกระทบ ฯลฯ ดังนั้นการวัด และบรรยายสีในเชิงวิชาการจึงต้องมีการจัด มาตรฐานเพื่อ เป็นการลดความไม่เป็นกลาง (bias) ของ ผู้บรรยายสีของวัสดุนั้น ๆ ปัจจัยที่ทำให้เกิดสีมีอยู่ 3 ประเภทคือ

1. แหล่งกำเนิดแสง (light source) แหล่งกำเนิดแสงมีผลอย่างมากในการบรรยายสีของวัตถุ แหล่งกำเนิดแสงถ้ามีแสงแตกต่างจากแสงขาว เมื่อตกกระทบกับวัตถุจะทำให้แสงที่สะท้อนกลับมาเกิดสีที่แตกต่างกันออกไป

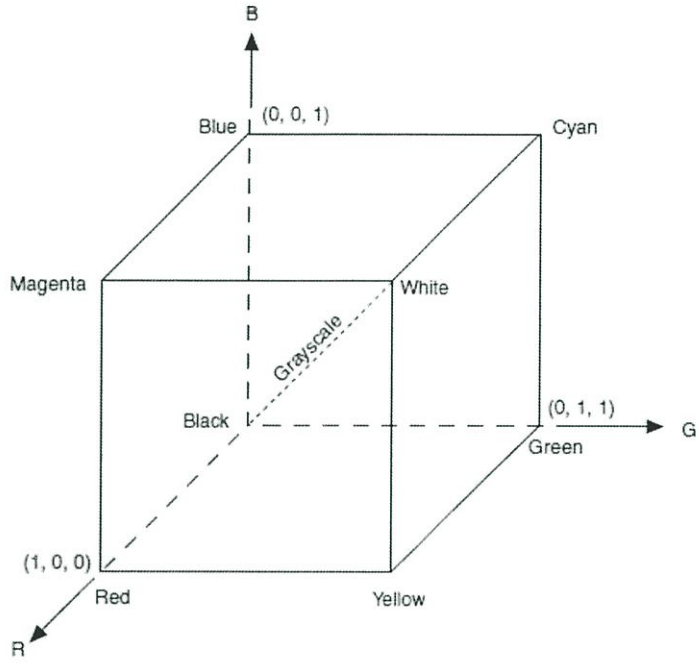
2. วัตถุที่มอง (specimen) วัตถุที่ทึบแสง (opaque) จะให้การสะท้อนของแสงเพื่อเกิดสีแตกต่างจากวัตถุที่โปร่งแสง (translucent) และโปร่งใส (transparent)

3. ผู้สังเกตการณ์ (observer) ผู้สังเกตการณ์นั้นมีผลอย่างยิ่งต่อการบรรยายสีที่มองเห็น ผู้สังเกตการณ์ต่างคนจะบรรยายลักษณะสีที่ต่างกันขึ้นอยู่กับสรีระทางกายภาพของตาแต่ละคน ในร่างกายคนจะมีเซลล์อยู่ 2 ชนิดที่เกี่ยวข้องกับการรับสี คือ เซลล์รูปแท่งและเซลล์รูปโคน เซลล์รูปแท่งจะตอบสนองได้ดีกับการมองเห็นในที่ที่เกี่ยวกับความมืด ความสว่าง ส่วนเซลล์รูปโคนจะตอบสนองต่อสีที่มองเห็นโดยตรง

จากหลักการพื้นฐานเรื่องสีข้างต้น จึงได้มีการพัฒนาอุปกรณ์เพื่อใช้วัดสีที่มีมาตรฐาน และความไม่เป็นกลาง เนื่องจากปัจจัยของแหล่งกำเนิดแสง และ สังเกตการณ์ องค์กรที่มีบทบาทสำคัญในการกำหนดมาตรฐานด้านสี คือ Commission International de l'Eclairage (CIE) ซึ่งองค์กรนี้ได้กำหนดมาตรฐานการวัดสีซึ่งเป็นที่ยอมรับอย่างกว้างในวงการวิชาการและการวิจัย คือระบบ CIE L*a*b

2.6.2 แบบจำลองสี RGB

แบบจำลองสีนี้เป็นแบบจำลองของแม่สีหลัก 3 สี (Primary Color) ซึ่งเป็นสิ่งที่เกิดจากการรวมกันของแสง (Additive Color) ประกอบด้วยสีที่สำคัญ 3 สีด้วยกัน ได้แก่ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) ถูกนำมาใช้งานในด้านการแสดงผลข้อมูลบนจอภาพ รวมไปถึงการเก็บข้อมูลภาพในระบบคอมพิวเตอร์



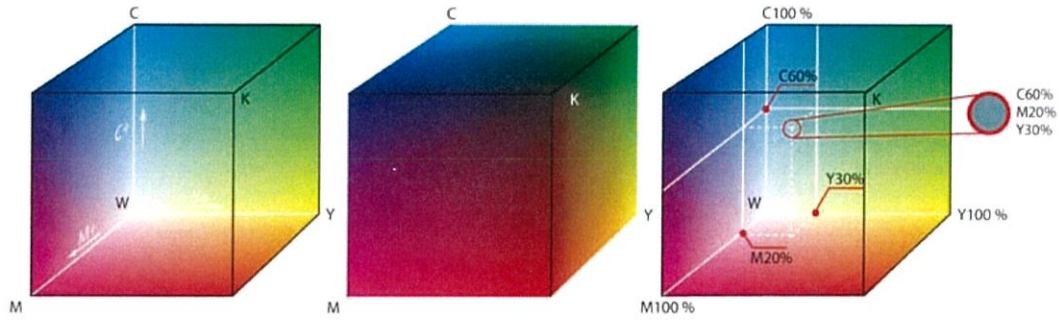
ภาพที่ 2.15 RGB color cube

2.6.3 แบบจำลองสี CMYK

ระบบสี CMYK เป็นระบบสีที่ใช้กับเครื่องพิมพ์ CMYK ย่อมาจาก cyan, magenta, yellow, key ซึ่งเป็นชื่อสีที่นำมาใช้ การผสมสีทั้งสี่นี้ จะทำให้เกิดสีได้อีกหลายร้อยสี นำมาใช้ในการพิมพ์สีต่าง ๆ ซึ่งปกติการเลือกใช้สีนั้น จะมีด้วยกันอยู่ 2 แบบคือ CMYK และ RGB สามารถแบ่งแยกประเภทการใช้งานได้ง่ายๆ นั้นก็คือ ถ้าเป็นสีที่ต้องพิมพ์ออกมา ไม่ว่าจะพิมพ์ในรูปแบบใดก็ตาม จะต้องใช้ค่าสีของ CMYK แต่ถ้าต้องการสีที่แสดงผลออกทางหน้าจอ ก็จะเลือกใช้ RGB เท่านั้น

การกำหนดค่าสี RGB ในส่วนของ RGB การเลือกสีต่างๆ นั้น สามารถเลือกได้ถึง 2 สี - 16 ล้านสี (ข้อมูลอ้างอิงจากกระทรวงศึกษาธิการ หลักสูตรการศึกษาด้านเทคโนโลยีสารสนเทศ ฉบับปี 2545) โดยการเข้ารหัสสีของแต่ละแม่สี ทั้งสามสี (Red, Green และ Blue ตามลำดับ) โดยค่ารหัสแม่สีแต่ละแม่สี **ต้อง**เป็นตัวเลขฐาน 16 (ซึ่งมีตัวเลข 0-9 และ A-F) *2 ตัวต่อหนึ่งแม่สี* รวม 3 สี = 6 ตัวอักษร อย่างเช่นสี A2589D คือ เป็นสีที่มีค่าสีแดง (red) = A2, สีเขียว (green) = 58 และ น้ำเงิน (blue) = 9D

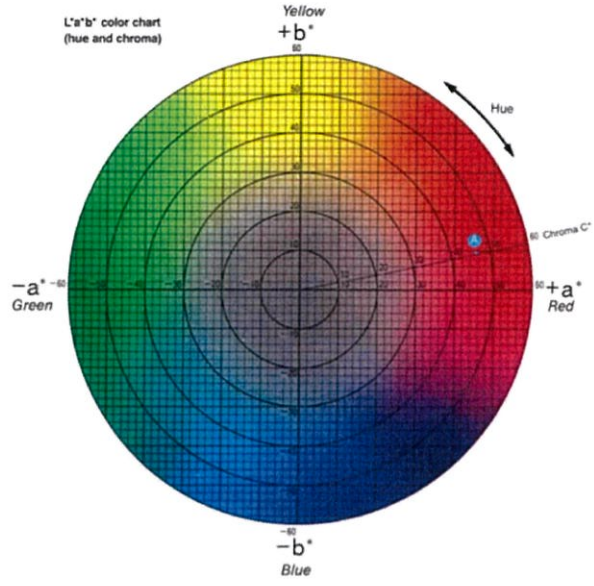
ในส่วนของ CMYK จะใช้หลักการเลือกสีรูปแบบเดียวกับ RGB แต่ว่าค่าสีจะถูกตัดออกไปเป็นจำนวนเยอะมาก ทำให้มีค่าสีอยู่แค่หลักร้อย หรือ พันกว่าสีเท่านั้น โดยที่ค่าสีของ CMYK จะตัดค่าสีที่ตาเรามองไม่เห็น หรือไม่สามารถแยกแยะออกถึงความใกล้เคียงกันมาเกินไปได้ อย่างในกรณีของใบไม้ที่เราอาจจะมองว่าเป็นสีเขียว แต่ถ้าจะให้ระบุสีในใบไม้เดียวกันให้ใกล้เคียงที่สุด อาจจะได้คำตอบที่เป็น สีเขียว, เขียวแก่, เขียวแก่กว่า, เขียวอ่อน, เขียวอมเหลือง, เขียวแกรมฟ้า ฯลฯ อย่างหาข้อสรุปไม่ได้ เพราะในสายตาคนเรามักจะมองรูปแบบสีหลักๆ เท่านั้น ดังนั้นปัญหาเรื่องของสีโหมด CMYK จึงจะตัดค่าสีที่ห่างกันไม่มากออกไปเพียงเท่านั้น



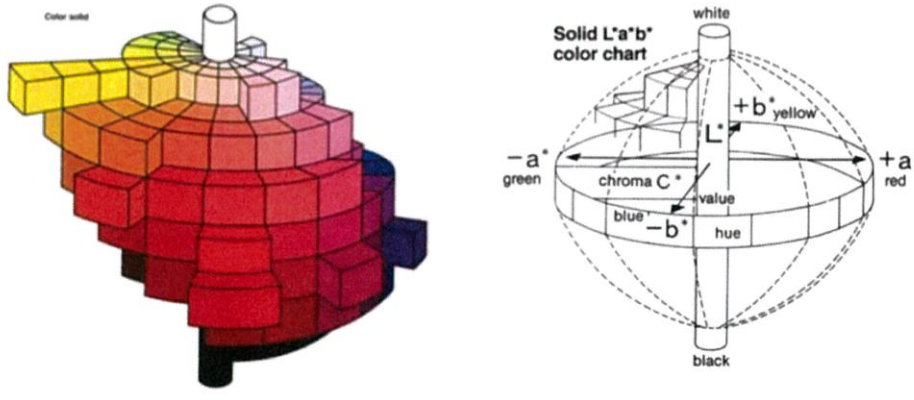
ภาพที่ 2.16 CMYK color cube

2.6.4 แบบจำลองสี L*a*b

ระบบ L*- a*-b* เป็นระบบการบรรยายสีแบบ 3 มิติ โดยที่แกน L* จะบรรยายถึงความสว่าง (lightness) จากค่า +L* แสดงถึงสีขาว จนไปถึง –L* แสดงถึงสีดำ แกน a* จะบรรยายถึง แกนสีจากเขียว (-a*) ไปจนถึง แดง (+a*) ส่วนแกน b* จะบรรยายถึงแกนสีจากน้ำเงิน (-b*) ไปเหลือง (+b*) ลักษณะการบรรยายสี ของ CIE แสดงได้ดังภาพที่ 2.17 และ 2.18



ภาพที่ 2.17 การบรรยายสีในระบบ CIE Lab ในระนาบ 2 มิติ



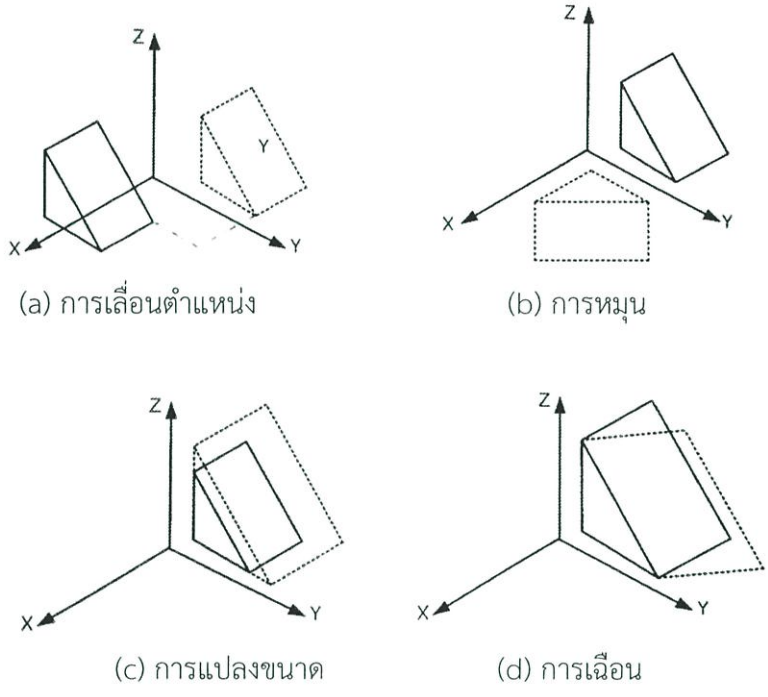
ภาพที่ 2.18 การบรรยายสีในระบบ CIE lab ในระนาบ 3 มิติ

2.6.5 ฮิสโตแกรม (Histogram)

ฮิสโตแกรมหมายถึงการกระจายของค่าพิกเซลในภาพ ในกรณีของภาพระดับเทา (Gray Level Image) ฮิสโตแกรมคือการกระจายของค่าความเข้มในภาพ ในกรณีของภาพสี ฮิสโตแกรมเป็นการดูการกระจายของค่าสีในแต่ละองค์ประกอบ

2.7 การแปลงแอฟไฟน์ (Affine Transform)

เป็นการแปลงเชิงเรขาคณิตที่ประกอบไปด้วย การหมุน การเลื่อนตำแหน่ง การเปลี่ยนแปลงขนาดและการเฉือน ภาพที่ 2.19 ตัวอย่างการแปลงรูปแบบต่างๆ ประกอบไปด้วย (a). การเลื่อนตำแหน่ง (b). การหมุน (c). การแปลงขนาด (d). การเฉือน



(a) การเลื่อนตำแหน่ง (b) การหมุน

(c) การแปลงขนาด (d) การเฉือน

ภาพที่ 2.19 การแปลงเชิงเรขาคณิตชนิดต่างๆ

2.8 ระบบพิกัดโฮโมจีเนียส (Homogeneous Coordinate)

การใช้เวคเตอร์ขนาด 3 มิติ $(x_k, y_j, 1)$ แทนเวคเตอร์ตำแหน่งขนาด 2 มิติ จะช่วยให้สมการการแปลงรวมอยู่ในรูปที่กระทัดรัดสะดวกต่อการนำไปใช้มากขึ้น เราเรียกว่าเป็นการเขียนแทนในระบบพิกัดโฮโมจีเนียส (Homogeneous Coordinate) ตัวอย่างเช่น การแปลงเรขาคณิตของการเคลื่อนที่ในระบบพิกัดโฮโมจีเนียสคือ [1]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad (2.1)$$

การแปลงเรขาคณิตของการสเกลในระบบพิกัดโฮโมจีเนียสคือ

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.2}$$

การเขียนในระบบพิกัดโฮโมจีเนียสสามารถทำให้หาเมทริกซ์การแปลงรวมที่ง่ายขึ้น เช่นการกาการแปลงเรขาคณิตรวมของการหมุนวัตถุรอบจุดหมุนใดๆ (x_c,y_c) ซึ่งประกอบด้วย *3* ขบวนการคือ (1) เลื่อนจุดหมุนไปที่จุดกำเนิด (2) ทำการหมุนภาพและ (3) เลื่อนจุดหมุนกลับไปที่เดิม ดังนั้นการแปลงเรขาคณิตรวมคือ

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.3}$$

เมื่อรวมการแปลงทั้งหมดเราได้การแปลงเรขาคณิตรวมดังนี้

$$T = S \cdot Sh \cdot R \cdot Tr \tag{2.4}$$

โดยที่

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; Sh = \begin{bmatrix} 1 & shx & 0 \\ shy & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Tr = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

และโดยที่เมทริกซ์ R เป็นการหมุนรอบจุดหมุน (x_c,y_c) นั่นคือ

$$R = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix}$$

ซึ่งการแปลง T สามารถลดรูปอยู่ในรูปเมทริกซ์เดี่ยวขนาด 3 x 3 ดังนั้นจะได้

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.5}$$

การแปลงเรขาคณิตย้อนกลับในระบบโฮโมจีเนียสสามารถทำได้โดยการทำ Inverse Matrix ของเมทริกซ์การแปลงเรขาคณิตขนาด 3 x 3 ในระบบพิกัดโฮโมจีเนียส เช่นการแปลงเรขาคณิตย้อนกลับของการเคลื่อนที่ในระบบพิกัดโฮโมจีเนียสคือ

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.6)$$

การแปลงเรขาคณิตย้อนกลับของการสเกลในระบบพิกัดโฮโมจีเนียสคือ

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.7)$$

ในกรณีทั่วไป การแปลงเรขาคณิตย้อนกลับของสมการ (2.5) อยู่ในรูป

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} c_0 & c_1 & c_2 \\ d_0 & d_1 & d_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (2.8)$$

2.9 ภาษา Java

Java เป็นภาษาคอมพิวเตอร์ที่ถูกนำไปใช้พัฒนาซอฟแวร์กันอย่างแพร่หลาย ไม่ว่าจะเป็นทางด้านธุรกิจ อุตสาหกรรม หรือเทคโนโนโลยีสารสนเทศ และโดยเฉพาะอย่างยิ่งการพัฒนาแอปพลิเคชั่นบนระบบ Android ก็ใช้ภาษา Java เป็นเครื่องมือในการพัฒนเช่นกัน ดังนั้นก่อนที่เราจะก้าวไปสู่การพัฒนาแอปพลิเคชั่นบน Android ก็จำเป็นต้องเขียนโปรแกรมภาษา Java ให้ได้

Java เป็นภาษาที่สร้างขึ้นโดย James Gosling ให้กับบริษัท Sun Microsystems และได้เริ่มใช้งานมาตั้งแต่ปี ค.ศ. 1995 ซึ่งชื่อของภาษานี้ได้มาจากการใช้โปรแกรมสุ่มเลือกคำ เมื่อคำว่า “Java” ถูกเลือก รูปถ้วยกาแฟก็ถูกใช้เป็นโลโก้ด้วยเช่นกัน เนื่องจากว่าคำว่า Java ในภาษาอังกฤษหมายถึง“กาแฟ”นั่นเอง [9]

ภาพที่ 2.20 Java Logo

2.9.1 จุดเด่นของภาษาจาวา

ปัจจุบันจาวากลายเป็นภาษาคอมพิวเตอร์ที่มีผู้ใช้งานมากที่สุดอีกภาษาหนึ่งของโลกซึ่งจุดเด่นบางประการที่ทำให้จาวาได้รับความนิยม มีดังนี้คือ

1. จาวามีลักษณะโครงสร้างคล้ายคลึงกับ C / C++ แต่มีรูปแบบที่สละสลวยและเรียนรู้ได้ง่ายกว่า

2. จาวาจะทำงานอยู่บน Java Virtual Machine (JVM) ซึ่งสามารถนำไปติดตั้งในระบบหรืออุปกรณ์ที่หลากหลายดังนั้นจึงสามารถทำงานข้ามแพลตฟอร์มได้ (Platform Independent) ถ้าหากว่าอุปกรณ์เหล่านั้นมี JVM ติดตั้งอยู่

3. จาวาเป็นภาษาที่ยึดหลักการเขียนโปรแกรมแบบ OOP อย่างเข้มงวด ดังนั้นเราจึงสามารถใช้ประโยชน์จากข้อกำหนดต่างๆของ OOP เพื่อพัฒนาแอปพลิเคชั่นได้อย่างเต็มประสิทธิภาพ

4. จาวาสามารถแก้ไขข้อบกพร่องหลายอย่างในภาษา C++ เช่น Pointer, Garbage Collection, การจัดการหน่วยความจำรวมถึงหลักการด้าน OOP ของ C++ ที่อาจก่อให้เกิดปัญหา

5. จาวามีความคงทน (Robust) ในการทำงานมากกว่า เนื่องจากว่าจาวามีกล-ไกลการป้องกันข้อผิดพลาดต่างๆที่อาจเกิดขึ้นจนเป็นสาเหตุให้แอปพลิเคชั่นต้องหยุดการทำงานลง

6. จาวานั้นถูกใช้ในธุรกิจและอุตสาหกรรมต่างๆ มากมาย เช่น การเงินธนาคาร, สื่อสารโทรคมนาคม, ไอที, อิเล็กทรอนิกส์, มัลติมิเดีย (Multimedia), เกม ฯลฯ

7. จาวานั้นถูกเลือกให้เป็นภาษาในการพัฒนาแอปพลิเคชั่นบน Android ดังนั้นผู้ที่ศึกษาการเขียนโปรแกรมสำหรับระบบแอนดรอยด์จึงจำเป็นที่จะต้องมีพื้นฐานการเขียนจาวา

2.9.2 หลักการประมวลผลของจาวา

ตามปกติแล้ว โปรแกรมที่พัฒนาขึ้นมาจากแพลตฟอร์ม (Platform) หนึ่งก็จำกัดการใช้งานอยู่แต่บนแพลตฟอร์มนั้นเท่านั้น ไม่สามารถนำไปใช้ข้ามแพลตฟอร์มได้ ภาษาจาวาจึงถูกสร้างขึ้นเพื่อแก้ปัญหานี้โดยโปรแกรมที่เขียนด้วยจาวาไม่ว่าบนแพลตฟอร์มใดก็ตาม จะสามารถนำไปใช้งานบนแพลตฟอร์มอื่นๆ ได้ทันที แต่การจะทำงานแบบข้ามแพลตฟอร์มได้ต้องอาศัยการประมวลผลคำสั่งใน 2 รูปแบบร่วมกันคือ

1. การคอมไพล์ (Compile) โดยโค้ดเขียนเป็นภาษาจาวา (มีส่วนขยายเป็น.java) จะถูกคอมไฟล์ออกมาเป็น Java byte code (มีส่วนขยายเป็น .class) ซึ่งเป็นโค้ดมาตรฐานในการทำงานข้ามแพลตฟอร์มของจาวา

2. การแปลคำสั่ง (Interpret) เมื่อนำ Java byte code นั้นไปรันบนแพลตฟอร์มใดก็ได้ที่มี Java Virtual Machine สำหรับแพลตฟอร์มนั้นอยู่ Java byte code นั้นจะถูกแปลคำสั่ง (Interpret) ให้ทำงานตามที่กำหนด

2.9.3 การเขียนโปรแกรมด้วยจาวา

ในการเขียนโปรแกรมไม่ว่าจะเป็นระดับพื้นฐานหรือระดับสูงก็จำเป็นต้องใช้องค์ประกอบพื้นฐานบางอย่างเหมือนกันสำหรับภาษาในจาวาก็มีสิ่งที่เราควรรู้จักในเบื้องต้นมีต่อไปนี้

1. คลาส

เนื่องจากจาวานั้นเป็นภาษาแบบ Object Oriented Programming (OOP) ซึ่งทุกอย่างจะถูกกำหนดใว้ภายใน คลาส (Class) โดยรูปแบบอย่างง่ายของคลาสมีดังนี้

```
Class ชื่อคลาส {
}
```

1.1 Class เป็นคีย์เวิร์ดที่บ่งบอกว่าเป็นคลาส

1.2 ชื่อคลาสควรกำหนดให้สื่อความหมายว่าเป็นคลาสเกี่ยวกับอะไร

2. บล็อก{}

ในจาวาจะใช้วงเล็บ {...} ในการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของบล็อกกการทำงานในแต่ละส่วน เช่น คลาส, เมธอด หรือการทำงานตามเงื่อนไขและลูปต่างๆ นอกจากนี้ยังนำ {...} มาใช้ในอีกหลายกรณี นอกจากนี้ภายในบล็อก {...} อาจมีบล็อกย่อยๆ ซ้อนลงไปอีกตามความซับซ้อนของโปรแกรม เช่น

```
Class Example {
        public static void main(String[] args ) {
                for (...) {
                    if (...) {
                      ...
                    } else {
                      ...
                }
            }
        }
    }
```

2.9.4 การแทรกคำอธิบาย

การแทรกคำอธิบาย (Comment) ไว้ในโค้ด จะช่วยให้อ่านโค้ด ได้เข้าใจง่ายยิ่งขึ้น รวมถึงกรณีที่เรากลับมาแก้ไขโค้ดในภายหลัง ก็จะช่วยให้เราทราบรายละเอียดของโค้ดตรงนั้นได้รวดเร็วกว่า ทั้งนี้การแทรกคำอธิบายในจาวาทำได้ดังนี้

1. การใช้สัญลักษณ์ //

สำหรับคำอธิบายเพียงบรรทัดเดียวโดยโปรแกรมจะถือได้ว่าตั้งแต่เครื่องหมาย // ไปจนสิ้นสุดบรรทัดเป็นคำอธิบายทั้งหมด จะไม่นำมาพิจารณาในโปรแกรม อาจเขียนไว้ในคลาส หรือนอกคลาสก็ได้เช่น

```
// ตัวอย่าง
Class Employee {
        private String name :  // ชื่อ
        private int age ;          // อายุ
}
```

2. การใช้สัญลักษณ์ /* */

กรณีที่คำอธิบายของเรานั้นยาวหลายบรรทัด การใช้ // หลายๆ ครั้งอาจไม่สะดวก เราอาจเปลี่ยนมาใช้ /*...*/ แทนได้ โดยโปรแกรมจะถือว่าตั้งแต่สัญลักษณ์ /* เป็นต้นไป จะเป็นคำอธิบายทั้งหมด จนกว่าจะเจอสัญลักษณ์ */ จึงจะถือว่าเป็นการสิ้นสุดคำอธิบาย เช่น

```
/* ตัวอย่าง
   การใช้ */
Class Employee {
        private String name :  /* ชื่อ*/
        private int age ;          /*อายุ*/
}
```

2.9.5 เครื่องหมายสิ้นสุดคำสั่ง

ในจาวาเราจะใช้เครื่องหมาย semicolon(;) เป็นตัวแสดงจุดสิ้นสุดในแต่ละคำสั่ง หากเราไม่ใส่โปรแกรมจะถือว่าเป็นคำสั่งเดียวกันไปตลอดแม้ อยู่คนละบรรทัดก็ตามทำให้เกิดข้อผิดพลาดได้ เช่น

```
int a = 10 ;
int b = a + 10 ;
String str = "java";
```

2.9.6 Main Class

หากคลาสนั้นเป็นคลาสหลักของแอปพลิเคชั่นที่ต้องเรียกขึ้นมาทำงานเป็นลำดับแรก คลาสนั้นจะต้องประกอบไปด้วยเมธอด main() เสมอ หรือเรียกว่า Main Class ซึ่งลักษณะของเมธอด main() เป็นดังนี้

```
class Sample {
        public static void main ( String[] args ) {
        //คำสั่งต่างๆ
)
}
```

2.9.7 Operator & Control Statement

การที่จะเขียนโปรแกรมให้ทำงานได้ตามที่เราต้องการนั้น จำเป็นต้องใช้คำสั่งเพื่อควบคุมให้เกิดการประมวลผลไปตามลำดับ เช่น การกำหนดเงื่อนไข และ การใช้ลูปต่างๆ เป็นต้น

1. คำสั่ง if ... else

คำสั่ง if ... else ใช้ในกรณีที่เราต้องกำหนดทางเลือกอื่นในการทำงาน หากเงื่อนไขไม่ตรงกับที่ระบุใน if โดยมีรูปแบบดังนี้

```
If (เงื่อนไข) {
        คำสั่งต่างๆ ถ้าเงื่อนไขตรงกับที่กำหนด
} else {
        คำสั่งต่างๆ ถ้าเงื่อนไขไม่ตรงกับที่กำหนด
}
```

2. คำสั่ง else if

ในกรณีที่มีการตรวจสอบหลยๆ เงื่อนไข ทำให้ไม่สะดวกต่อการใช้ if หลายๆ ครั้ง เราสามารถที่จะใช้ else if เข้ามาช่วยตรวจสอบในแต่ละเงื่อนไข โดยใช้รูปแบบดังต่อไปนี้

```
if (เงื่อนไข) {
    คำสั่งต่างๆถ้าตรงกับเงื่อนไขที่ 1
} else if ( เงื่อนไขที่ 2 ) {
 คำสั่งต่างๆถ้าตรงกับเงื่อนไขที่ 2
} else if ( เงื่อนไขที่ 3 ) {
 คำสั่งต่างๆถ้าตรงกับเงื่อนไขที่ 3
...
} else  { //กรณีอื่นๆ (ถ้ามี)
 คำสั่งต่างๆถ้าไม่ตรงกับเงื่อนไขใดๆเลย
}
```

ทั้งนี้จะมีการตรวจสอบเงื่อนไขด้วย else if กี่ครั้งก็ได้ และเราสามารถนำการตรวจสอบเงื่อนไขทั้ง if, else, else if มาใช้ร่วมกันได้ แต่ต้องวาง else ไว้ในลำดับท้ายสุด

2.9.8 Class & Object

จาวาเป็นภาษาที่ยึดหลักการเขียนโปรแกรม แบบ OOP อย่างเข้มงวด ซึ่งสิ่งที่เราควรรู้จักในเบื้องต้นก็คือ คลาสและองค์ประกอบต่างๆ เช่น เมธอด, ฟิลด์, พารามิสเตอร์ และโมดิฟายเออร์ที่ต้องใช้ในเบื้องต้น ดังต่อไปนี้

1. การสร้างเมธอด

เมธอด (Method) ใช้สำหรับกำหนดการกระทำอย่างใดอย่างหนึ่งของคลาส เช่น การคำนวณหาผลลัพธ์ การแสดงผล หรือ อ่านค่าบางอย่าง เป็นต้น ดังนั้นเมธอดจึงถือเป็นหัวใจหลักของคลาสที่เราต้องศึกษาให้เข้าใจก่อนเป็นอันดับแรกโดยมีรายละเอียดดังต่อไปนี้

1.1 เมธอดที่ไม่มีการส่งค่ากลับ (Void Method) จะกำหนดเพียงการกระทำเท่านั้น แต่ไม่ มีการส่งค่าใดกลับออกไปจากเมธอดมีรูปแบบโดยทั่วไปดังนี้

```
โมดิฟายเออร์ void ชื่อเมธอด ( พารามิเตอร์ ) {
   // คำสั่งต่างๆ
}
```

1.1.1 โมดิฟายเออร์ เป็นการกำหนดระดับการเข้าถึงเมธอด ในเบื้องต้นนี้ให้กำหนดเป็น public static

1.1.2 void หากเมธอดนั้นไม่ต้องส่งผลลัพธ์กลับออกไป ให้ระบุคีย์เวิร์ด void ไว้หน้าชื่อเมธอดเสมอ

1.1.3 ชื่อเมธอด มีหลักเกณฑ์คล้ายกับการตั้งชื่อตัวแปร ในจาวาเรานิยมตั้งชื่อเมธอดโดยให้ตัวอักษรของคำแรกขึ้นต้นด้วยพิมพ์เล็กแต่ตัวอักษรของคำถัดไปให้ขึ้นต้นด้วยตัวพิมพ์ใหญ่ เช่น deleteData, isValidInput เป็นต้นแต่นั้นไม่ใช่ข้อบังคับ

1.1.4 พารามิเตอร์ เป็นข้อมูลที่เราต้องการนำมาใช้ในเมธอด แต่เมธอดอาจมีพารามิเตอร์จำนวนกี่ตัวหรือไม่มีเลยก็ได้ในการระบุพารามิเตอร์ต้องระบุทั้งชนิดและตัวแปรที่จะรับข้อมูลเข้ามาหากมีพารามิเตอร์มากกว่า 1ตัว ให้คั่นด้วยเครื่องหมายคอมม่า(,) ลักษณะการกำหนดเมธอดแบบไม่ส่งค่ากลับ เช่น

```
Pubic static void SayHello () {
   System.out.print("Hello");
}

public static void ShowMessage ( String message ) {
  System.out.print (message);
}
// ความเป็นจริงต้องเขียนเมธอดไว้ในคลาสแต่เพื่อไม่ให้โค้ดยาวจนเกินไป จึงขอละไว้ ณ ที่นี้
```

1.2 เมธอดที่มีการส่งค่ากลับ เนื่องจากเมธอดนั้น ใช้กำหนดการกระทำอย่างใดอย่างหนึ่ง เมื่อได้ผลลัพธ์มา เราอาจจะส่งค่าที่ได้กลับมาไปให้ส่วนที่เรียกใช้เมธอดนั้น ซึ่งรูปแบบโดยทั่วไปของเมธอดที่ส่งค่ากลับมีดังนี้

```
โมดิฟายเออร์ <ชนิดข้อมูลที่จะส่งกลับ> ชื่อเมธอด ( พารามิเตอร์ ) {
   คำสั่งต่างๆ ภายในเมธอด
   return ข้อมูลที่จะส่งกลับ
}
```

1.2.1 ชนิดของข้อมูลที่จะส่งกลับ เป็นการบอกให้ทราบว่า เมธอดนี้จะส่งข้อมูลชนิดใดกลับคืนไป

1.2.2 return เป็นคำสั่งจะใช้คำสั่งข้อมูลจากเมธอดออกไป ซึ่งข้อมูลที่จะ-ส่งกลับออกไปนี้ เป็นชนิดเดียวกับที่เรากำหนดไว้ที่หน้าชื่อเมธอด มินั้นเกิดข้อผิดพลาดขึ้นได้ ลักษณะเมธอดแบบไม่มีพารามิเตอร์ แต่มีการส่งผลลัพธ์ชนิด byte กลับไป

```
Public static int getRandomNum ( ) {
   int r = ( int ) ( Math.random ( ) * 100 ) ;
   return r ;
   System.out.print ( "เลขสุ่มที่ได้คือ " + r ) ; // คำสั่งนี้ไม่มีผล ( Unreachable statement )
}
```

เราต้องวาง return ไว้เป็นคำสั่งสุดท้ายของเมธอด เพราะหลังจากคำสั่ง return ไปแล้ว คำสั่งที่อยู่ต่อจากนั้นจะไม่มีผลใดๆ เลย เช่น

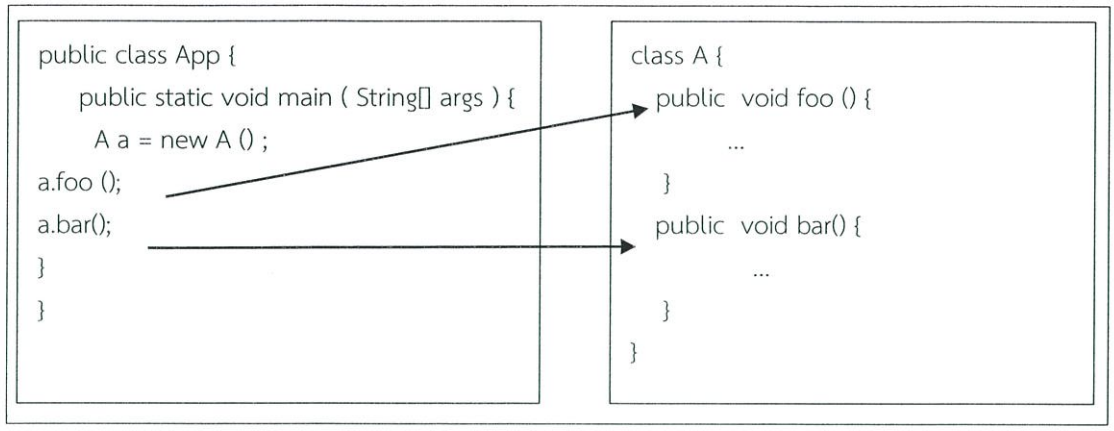
```
Public static byte selectMenu ( ) {
    System.out.println ( " กรุณาเลือกเมนู " ) ;
    System.out.println ( " 1 = ถอน , 2 = ถามยอด , 3 = โอน " ) ;
     Scanner scan = new Scanner ( System.in);
    byte menu = scan.nextByte ( ) ;
return menu;
}
```

2.9.9 โมดิฟายเออร์แบบ public และprivate

โมดิฟายเออร์ (Modifier) เป็นกลุ่มคีย์เวิร์ดที่ใช้ในการควบคุม หรือกำหนดระดับการเข้าถึงองค์ประกอบต่างๆของคลาส ซึ่งในจาวานั้นมีโมดิฟายเออร์ที่สำคัญ หลายอัน เช่น public, private, protected, static, final เป็นต้น

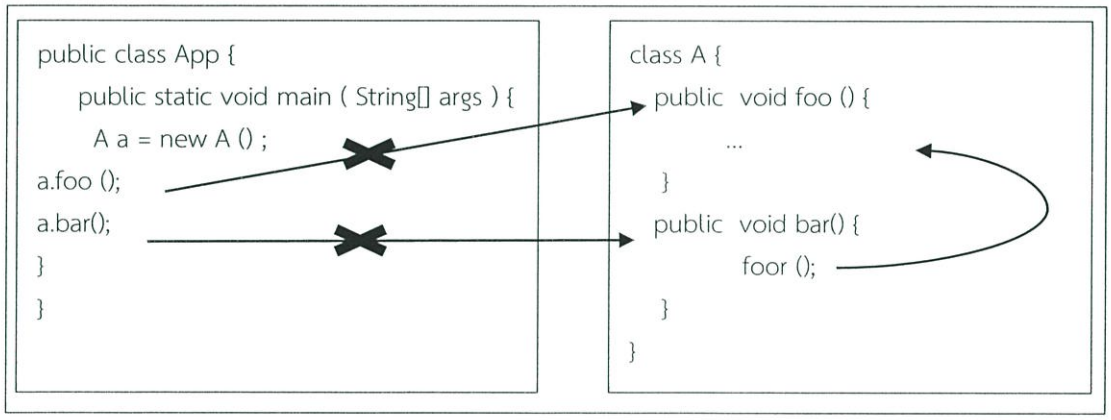
1. โมดิฟายเออร์แบบ public

โมดิฟายเออร์แบบ public จะใช้เมื่อเราไม่ต้องการปกปิดการเข้าถึงองค์ประกอบส่วนนั้นๆ หรือกล่าวได้ว่าเป็นการเปิดให้เข้าใช้งานได้อย่างอิสระนั่นเอง ซึ่งจากที่ผ่านมานั้น จะพบว่าเราได้ใช้คีย์เวิร์ด public นำหน้าทั้งคลาสและเมธอด และจากในภาพจะพบว่าใน คลาส A นั้นเราได้ใช้คีย์เวิร์ดpublic กำกับและสามารถเรียกใช้เมธอดที่อยู่ในคลาสเหล่านี้จากคลาสหลักได้



2. โมดิฟายเออร์แบบ private

โมดิฟายเออร์แบบ private จะใช้เมื่อเราต้องจำกัดสิทธิในการใช้งานเฉพาะภายในเมธอดนั้นโดยไม่ต้องการให้เข้าถึงจากภายนอกคลาสได้ สำหรับโมดิฟายเออร์นี้ สามารถใช้ได้เฉพาะกับสมาชิกภายในคลาส เช่น เมธอด หรือ ฟิลด์แต่ไม่สามารถ ใช้กับคลาสได้ (ยกเว้นกรณีที่เป็น Inner Class) เพราะตามปกติคลาสต้องถูก เรียกใช้งานโดยโปรแกรมหรือจากคลาสอื่นๆ อยู่แล้ว เช่นหากกำหนดโค้ดแบบนี้จะเกิดข้อผิดพลาด



จากในภาพจะพบว่าเมธอดในคลาส A เรากำหนดโมดิฟายเออร์เป็น private ดังนั้นจึงไม่สามารถเข้าถึงจากคลาสอื่นๆ ได้ ยกเว้นการเรียกใช้ภายในคลาสเดียวกันเท่านั้น

3. การไม่ระบุโมดิฟายเออร์

ภาษาจาวาไม่บังคับว่าเราต้องกำหนดโมดิฟายเออร์ทั้งนี้หากเราไม่กำหนดโมดิฟาย-เออร์ จะเรียกว่าเป็นการใช้ Default Modifier โดยมีผลคล้ายกับ public คือเป็นการเปิดให้เข้าถึงสมาชิกนั้นจากภายนอกคลาสได้แต่คลาสที่จะสามารถเรียกใช้ได้ต้องอยู่ในแพคเก็จ

(Package) เดียวกันเท่านั้น ซึ่งต่างจากกรณีที่เราระบุโมดิฟายเออร์เป็น public โดยตรง ที่จะสามารถเข้าถึงได้อย่างอิสระแม้จะอยู่คนละแพคเก็จก็ตามลักษณะสมาชิกของคลาสที่ไม่ระบุโมดิฟายเออร์หรือ Default Modifier เช่น

```
Class GameDemo {
        void startGame ( ) { ... }
            void startGame ( ) { ... }
            boolean  isGameOver ( ) { ... }
}
```

2.9.10 การใช้ this สำหรับการ Callback

การใช้คีย์เวิร์ด this ที่เราควรรู้จักเพิ่มเติมในอีกลักษณะหนึ่งก็คือ การใช้ในกรณีเราต้องการส่งคลาสที่เป็นตัวมันเอง เพื่อเป็นอาร์กิวเมนต์ให้กับเมธอดของคลาสอื่นๆ เนื่องจากคลาสปลายทางนั้นต้องเรียกย้อนกลับมาใช้งานในบางเมธอดของคลาสต้นทางหรือการเรียกย้อนกลับ (Callback) นั่นเอง ให้ผู้อ่านลองดู 2 คลาสในโค้ดถัดไปจะเข้าใจได้ง่ายยิ่งขึ้น

```
Public class Num {
     public static void main (String[] args ) {
         Square sq = new Square ( ) ;
         int  sum = sq.getSquare(this);
    }
    public int getNum ( ) {
         return ( int ) Math.random( ) *10;
    }
}
public class Square {
     public int getSquare ( Num n ) {
         return n.getNum ( ) * n.getNum ( ) ;     // callback
     }
}
```

คลาส Num จะเป็นผู้เรียกใช้เมธอด getSquared() ของคลาส Square แต่เมธอดนี้มีพารามิเตอร์เป็นตัวอ้างอิงของคลาส Num ดังนั้นจึงส่งคีย์เวิร์ด this ซึ่งใช้อ้างถึงตัวมันเอง[1]

2.9.11 Shared Preferences

เป็นคลาสที่ใช้สำหรับเก็บข้อมูลถาวรที่เป็นค่าของตัวแปร โดย Shared Preferences จะมีอยู่ด้วยกันสองชนิด คือ Shared Preferences กับ Preferences โดย Shared Preference จะเป็นการเก็บข้อมูลตัวแปรที่สามารถดึงไปใช้งานที่ไหนก็ได้ภายในแอปพลิเคชันนั้นๆดังนั้นถ้าจะส่งข้อมูระหว่าง Activity ก็ใช้ Shared Preferences ได้ เช่น Activity A เก็บข้อมูลบางอย่าง แล้วให้ Activity B ดึงขึ้นมาใช้งาน เป็นต้น Preferences จะเป็นการเก็บข้อมูลตัวแปรที่ใช้งานได้เฉพาะที่ที่เรียกใช้เท่านั้น เช่น Activity A สร้าง Preferences เก็บค่าบางอย่างไว้ เมื่อ Activity B จะดึงค่าที่ Activity A เก็บไว้ก็จะไม่สามารถทำได้[16]

ข้อมูลตัวแปรที่สามารถเก็บด้วย Shared Preferences จะมีทั้งหมดดังนี้เท่านั้น

1. Boolean
2. Float
3. Integer
4. Long
5. String
6. String Set (Array)

การเก็บข้อมูลลงใน Shared Preferences

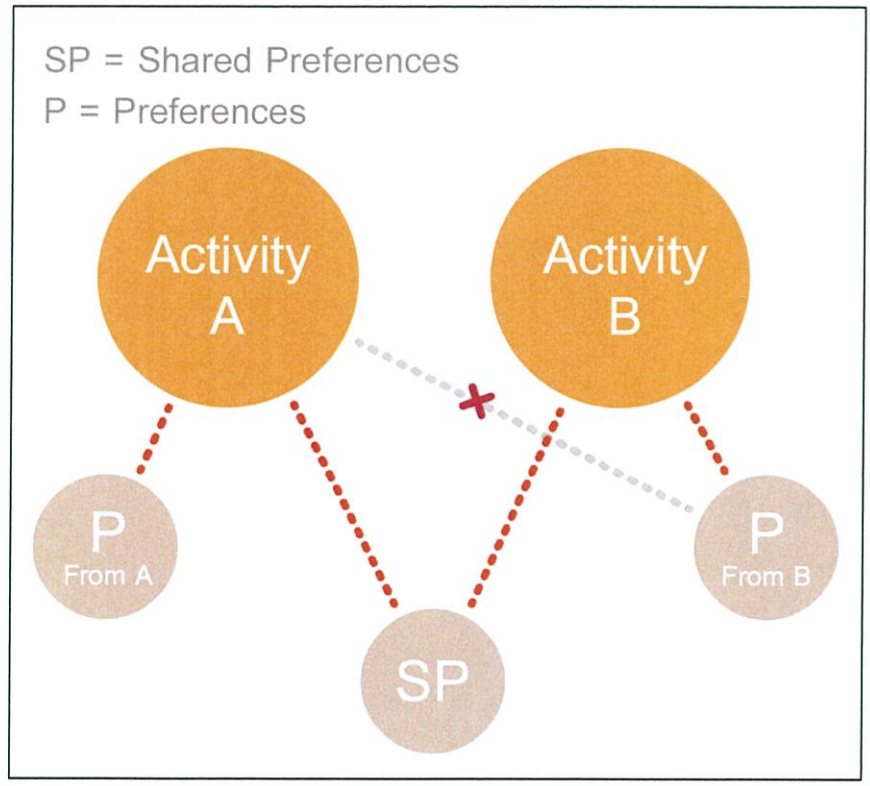
```
SharedPreferences sp = getSharedPreferences("PREF_NAME",
Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sp.edit();
editor.putInt("My_Value", x);
editor.commit();
```

เมื่อกำหนดข้อมูลที่จะเก็บเรียบร้อย ให้ใช้คำสั่ง Commit เพื่อทำการบันทึกข้อมูลลงใน Shared Preferences ถ้าใช้คำสั่ง Put จำนวนหลายข้อมูล แล้วไม่ได้ใช้คำสั่ง Commit ต่อท้าย ข้อมูลก็จะไม่ได้เก็บลงใน Shared Preferences

การอ่านข้อมูลจาก Shared Preferences

```
boolean getBoolean(String key, boolean defValue);
int getInt(String key, int defValue);
float getFloat(String key, float defValue);
long getLong(String key, long defValue);
String getString(String key, String defValue);
Set<String> getStringSet(String key, Set<String> defValue);
```

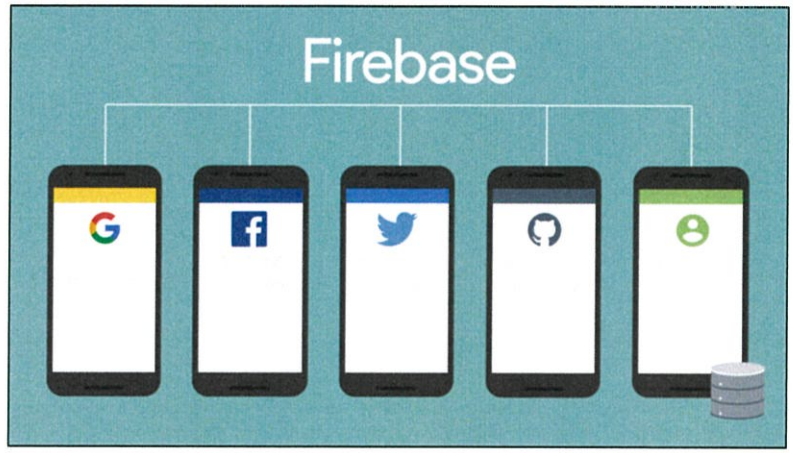
สำหรับการอ่านข้อมูลจาก Shared Preferences ก็จะคล้ายๆกับการบันทึกข้อมูล คือ สามารถที่จะเรียกใช้เมื่อไรก็ได้ ไม่ได้จำกัดตายตัว แต่ต่างกันตรงที่ไม่ต้องมี Commit เพราะเป็นการดึงข้อมูลออกมาใช้งาน และที่สำคัญ กำหนดชื่อ Shared Preferences ที่ต้องการดึงข้อมูลให้ถูกต้องแล้ว Keyword ก็ต้องถูกต้อง และที่สำคัญคือคลาส Shared Preferences ที่ประกาศจะสามารถใช้ร่วมกันได้เลยไม่ว่าจะบันทึกข้อมูลหรืออ่านข้อมูลก็ตาม[12]



ภาพที่ 2.21 ลักษณะของ Shared Preference

2.10 การทำ Authentication และ Registration

Firebase Authentication เป็นบริการที่เข้ามาจัดการ backend ให้ทั้งหมด ทั้งการ Register, การ Sign-in, การ Reset password โดยที่จะมี SDK ในระบบ Android, IOS และ Web ที่สามารถนำไปติดตั้งและใช้งานได้ ซึ่งรองรับการ Sign-in หลากหลายรูปแบบเช่น Email, การไม่ระบุตัวตน (Anonymous) หรือ Application Social Network



ภาพที่ 2.22 การใช้งาน Authentication ด้วยวิธีต่างๆ

โดยที่การเพิ่ม Firebase เข้าสู่โปรเจค Android นั้นจะมีข้อกำหนดเบื้องต้นในการใช้งาน Firebase ดังนี้

1.รองรับ Android 4.0 (Ice Cream Sandwich) เป็นต้นไป

2.รองรับ Dependency ของ Google Play Service 11.8.0 เป็นต้นไป

3.พัฒนาด้วย Android Studio ตั้งแต่เวอร์ชัน 3.0.1 เป็นต้นไป [8]

ภาพที่ 2.23 Firebase app Development

2.11 ระบบปฏิบัติการแอนดรอยด์ (Android)

แอนดรอยด์ (อังกฤษ: Android) เป็นระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ต คอมพิวเตอร์ เริ่มพัฒนาโดยบริษัท แอนดรอยด์ จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดย Google และนำแอนดรอยด์ไปพัฒนาต่อในภายหลัง ถูกพัฒนาในนามของ Open Handset Alliance ทาง Googleได้เปิด ให้ นักพัฒนาสามารถแก้ไขโค้ด ต่างๆ ด้วยภาษาจาวา และควบคุมอุปกรณ์ผ่านทางชุด Java libraries ที่ Google พัฒนาขึ้น

2.11.1 เกี่ยวกับระบบ Android

Android เป็นระบบปิบัติการแบบเปิดเผยซอร์สคด้ด (Open Source) ภายใต้ลิขสิทธิ์ของ Google ซึ่งบริษัทต่างๆ สามารถที่จะนำไปพัมนาเพิ่มเติม หรือปรับแต่งใช้เองให้ใช้ร่วมกับผลิตภัณฑ์ของตนได้ จึงทำให้แอนดรอยด์ไม่ยึดติดกับอุปกรณ์ หรือผู้ผลิตรายใดรายหนึ่ง ดังนั้นในปัจจุบันเราจึงสามารถพบเห็นระบบแอนดรอยด์ได้อย่างหลากหลายทั้งในอุปกรณ์เคลื่อนที่ (Mobile Device) ชนิดต่างๆ เช่น โทรศัพท์มือถือ, แท็ลเล็ต, นาฬิกา หรือกล้องดิจิตอล รวมถึง เครื่องใช้ไฟฟ้าบางชนิด เช่น โทรทัศน์ นอกจากนี้รถยนต์บางรุ่นก็ได้เริ่มนำระบบแอนดรอยด์มาใช้กันแล้ว

2.11.2 ประวัติความเป็นมาของ Android

Android เป็นระบบปฏิบัติการที่ริเริ่มสร้างขึ้นมาโดย Andy Rubin โดยในตอนแรกนั้น เขาตั้งใจที่จะนำไปใช้กับกล้องถ่ายรูป แต่กลับไม่ได้รับสนใจจากผู้ผลิต ทำให้เขาเปลี่ยนเป้าหมายไปสู่การพัฒนาเป็นระบบปฏิบัติการสำหรับโทรศัพท์มือถือในรูปแบบของ Open Source กล่าวได้คือ แจกจ่ายให้แก่ผู้ผลิตโทรศัพท์มือถือนำโค้ดไปใช้ได้ฟรี และสามารถปรับแต่งหรือพัฒนาต่อเติมได้ตามความเหมาะสม โดยผู้สร้าง Android อย่างเขาจะได้รับผลตอบแทนในรูปแบบของการสร้างบริการเสริมต่างๆ ต่อมา Rubin ได้ชักชวนอดีตเพื่อนร่วมงานเก่าของเขาอันประกอบไปด้วย Chris White, Nick Sears และ Rich Miner มาร่วมกันก่อตั้งเป็นบริษัท Android Inc. แต่อย่างไรก็ตามในช่วงแรกๆ นั้น ระบบ Android ยังไม่ประสบความสำเร็จเท่าที่ควร เนื่องจากผู้ผลิตโทรศัพท์ในขณะนั้นให้ความเห็นว่าผู้ใช้งานไม่ได้สนใจว่าโทรศัพท์เครื่องนั้นจะใช้ระบบปฏิบัตการใด แต่สนใจที่ตัวเครื่องในแต่ละรุ่นมากกว่า รวมถึงไม่ต้องการที่จะเปลี่ยนแปลงวิธีการผลิตไปจากเดิมอีกด้วยจึงปฏิเสธที่จะนำระบบแอนดรอยด์มาใช้งาน ทำให้ Rubin ต้องดิ้นรนอย่างหนักเพื่อผลักดันให้ระบบAndroidของเขาเป็นที่ยอมรับ แต่โชคก็ยังเข้าข้างพวกเขา โดยในเวลานั้นทาง Google เองก็กำลังมีแนวคิดที่จะพัฒนาระบบปฏิบัติการสำหรับโทรศัพท์มือถือในรูปแบบของ Open Source อยู่เช่นกัน เมื่อทราบว่าบริษัท Android Inc. กำลังประสบปัญหาจึงได้ให้ทุนช่วยเหลือ และเสนอตัวเข้าร่วมพัฒนาด้วย แล้วในเวลาต่อมานั้น เมื่อ Google มองเห็นช่องทางและโอกาสที่จะประสบความสำเร็จได้ จึงซื้อกิจการของ Android Inc. ไปทั้งหมด ทำให้ระบบ Android ตกเป็นของ Google ไปด้วย แล้วก็ได้รับการพัฒนามาอย่างต่อเนื่อง จนในปัจจุบัน Android ได้กลายเป็นระบบปฏิบัติการสำหรับพกพาที่มีผู้ใช้งานมากที่สุด

2.11.3 ที่มาของชื่อ Android และ โลโก้หุ่นยนต์

ก่อนที่ Andy Rubin จะก่อตั้งบริษัท Android Inc. เขามีบริษัทที่ตั้งขึ้นมาเองคือ Danger Inc. ซึ่งเคยมีผลงานการพัมนาระบบโทรศัพท์ให้กับ T-Mobile ที่รู้จักกันคือรุ่น Danger Hiptop หรือ T-Mobile Sidekick และก่อนนั้นก็ได้เคยผ่านการทำงานกับบริษัทยักษ์ใหญ่มาแล้วหลายแห่ง โดยเฉพาะอย่างยิ่งในช่วงหนึ่งเขาเคยทำงานเป็น Software Engineer ให้กับบริษัท Apple ซึ่งสิ่งที่ Andy Rubin คลั่งไคล้มากเป็นพิเศษก้คือ หุ่นยนต์ (Robot) โดยเขาทุ่มเทเวลาส่วนใหญ่ไปกับการเขียนโค้ดเพื่อควบคุมชิ้นส่วนต่างๆ ของหุ่นยนต์ จนเพื่อนร่วมงานที่บริษัท Apple ในขณะนั้นเรียกชื่อล้อเลียนเขาว่า Android ซึ่งเกิดจากการรวมคำระหว่าง Andy + Droid (คำว่า Andy คือชื่อของเขส ส่วนคำว่า Droid เป็หุ่นยนต์ในหนังStar Wars)จนกระทั่งเมื่อสร้างระบบปฏิบัติการขึ้นมา จึงได้นำสมญานาม **Android** มาตั้งเป็นชื่อระบบปฏิบัติการของเขาด้วย และเลือกใช้หุ่นยนต์ที่ตัวเองชอบมาเป็นโลโก้อีกด้วย

ภาพที่ 2.24 Android Logo

2.11.4 Android Application

Android Application คือ Application ต่างๆ ที่สามารถติดตั้งใช้งานได้กับสมาร์ทโฟนที่ใช้ระบบปฏิบัติการ Android ซึ่งในปัจจุบันได้มีการพัฒนา Android Application กันมาก เนื่องจาก Android เป็น OS ที่เป็น Open Source สามารถใช้งานได้ฟรี และติดตั้งได้กับสมาร์ทโฟนหรืออุปกรณ์ที่หลากหลาย และ นักพัฒนาก็สามารถพัฒนา Android Application ได้ด้วย Notebook หรือ PC Desktop ธรรมดาได้ เรียกได้ว่า ความสามารถของAndroidที่สามารถทำงานร่วมกับ Hardware อย่างเป็นอิสระได้เกือบทุกอย่างจึงทำให้ได้รับความนิยมจากองค์กรธุรกิจจำนวนมากมาย และมีการนำ Android Application มาใช้งานร่วมกับธุรกิจหลายๆ ประเภท ทั้ง Application ที่สามารถโหลดมาใช้งานได้เลย หรือ Application ที่ต้องซื้อหรือต้องเสียค่าบริการก็ตาม ถ้า Application นั้นมีประโยชน์ต่อผู้ใช้ การพัฒนา Android Applicationในปัจจุบันได้มีผู้พัฒนาและบริษัทที่ให้บริการรับพัฒนา อย่างแพร่หลาย เนื่องจาก Android เป็นระบบปฏิบัติการบนสมาร์ทโฟนแบบเปิดอิสระทำให้ มีผู้ให้ความสนใจสร้าง Android Application กันมากในปัจจุบัน

2.11.5 เวอร์ชั่นและ API Level ของ Android

ระบบแอนดรอยด์นั้นมีการปรับปรุงเปลี่ยนแปลงอยู่ตลอด และมีการพัฒนามาแล้วหลากหลายเวอร์ชั่น โดยแต่ละเวอร์ชั่นจะสัมพันธ์กับระดับ API ซึ่งเราจะต้องนำไปพิจารณาในการสร้างแอปพลิเคชั่นด้วย เช่น ถ้าเราต้องการให้แอปพลิเคชั่นนั้นรองรับตั้งแต่เวอร์ชั่น 4.4 (KitKat) ขึ้นไป ก็จะต้องเลือก Minimum API เป็น Level19 เป็นต้น

ตารางที่ 2.4 แสดง Version และ API Level ของระบบปฏิบัติการต่างๆ

Code Name	Version	API Level
Nougat	7.0	24
Marshmallow	6.0	23
Lollipop	5.1	22
	5.0	21
KitKat (Watch)	4.4w	20
KitKat	4.4 – 4.4.4	19
Jelly Bean	4.3.x	19
	4.2.x	17
	4.1.x	16
Ice Cream Sandwich	4.0.3 – 4.0.4	15
	4.0.1 – 4.0.2	14

2.11.6 วัตถุประสงค์ของ Android Studio

Android Studio นั้นเป็น Official IDE Tool จาก Google ไว้พัฒนา Android โดยเฉพาะจากแนวคิดพื้นฐาน InteliJ IDEA คล้ายๆ กับการทำงานของ Eclipse หรือ Netbean และ Android ADT Plugin วัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android โดยเฉพาะให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน Smart Phone แต่ล่ะรุ่น [2]

ภาพที่ 2.25 โปรแกรม Android Studio

2.11.7 OpenCV in Android Studio

OpenCV หรือ Open Source Computer Vision เป็นกลุ่มฟังก์ชันที่เรียกว่า ไลบรารี่ (Library) ที่ผู้ใช้สามารถเรียกใช้งานขณะเขียนโปรแกรมผ่านภาษาต่างๆ วัตถุประสงค์เพื่อใช้ในงานประมวลผลภาพอิงเวลาจริงบนคอมพิวเตอร์ (Real-time Computer Vision) และการความสามารถทางด้านการมองเห็นของคอมพิวเตอร์ (Computer Vision) ความสามารถต่างๆของ OpenCV คือ ชุดคำสั่งหรือฟังก์ชั่นที่สามารถถูกเรียกใช้งานได้อย่างง่ายดาย ละดวก และรวดเร็ว จึงทำให้ OpenCV ถูกนำมาใช้เป็นเครื่องมือในกลุ่มของนักวิจัยและนักพัฒนาโปรแกรมคอมพิวเตอร์ด้าน Image - Processing และ Computer Vision อย่างกว้างขวาง โดยเป็นโครงการของบริษัทอินเทล (Intel's Research) ภายใต้การกับกับดูแลของ Nizhny Novgorod ซึ่งเป็นชาวรัสเซีย ต่อมาถูกพัฒนาต่อโดย Willow Garage ขณะนี้อยู่ภายใต้การดูแลของ Itseez ความสามารถที่โดดเด่นคือสามารถทำงานข้ามแพลตฟอร์ม (Cross-platform) และถูกแจกจ่ายเพื่อใช้แบบฟรีภายใต้เงื่อนไขแบบ BSD license ใช้งานได้อย่างอิสระทั้งด้านการค้าและไม่ใช่การค้า (Bradski and Kaehler, 2008) OpenCV เป็นไลบรารีที่รวบรวมฟังก์ชันการใช้งานเกี่ยวกับการประมวลผลภาพ ไว้มากมาย เช่น การกรองรูป (Filters), การหาขอบภาพ (Edges), การเปลี่ยนแปลงเกี่ยวกับรูป (Transformation), การสกัดวัตถุออกจากภาพ (Segmentation), การค้นหาลักษณะเด่น (Detection), การประมวลผลแบบติดตามวัตถุ (Tracking) และการเปรียบเทียบความเหมือนของภาพ (Matching) นอกจากนี้ยังรวบรวมอัลกอริทึมด้านปัญญาประดิษฐ์ต่างๆไว้ด้วย เช่น SVM (Support Vector Machine), ANN (Artificial Neural Network) และ KNN (K-nearest Neighbor) และมีเว็บไซต์ www.opencv.org เป็นแหล่งข้อมูลเกี่ยวกับการพัฒนาการประมวลผลภาพด้วยไลบรารี OpenCV ซึ่งมีทั้งเอกสาร โปรแกรม และคู่มือการติดตั้งไว้ให้ศึกษา

เนื่องจากกระบวนการทาง Image Processing บน Android ซึ่งต้องการการประมวลผลที่ซับซ้อนขึ้น การจะใช้เพียงแค่ Bitmap pixel processing คงจะทำให้ไม่สามารถทำงานได้อย่างเต็มที่ จึงทำให้ต้องมีการนำ OpenCV มาใช้ทดแทนในส่วนนี้ ซึ่งแม้ว่า OpenCV นั้นจะมีความสามารที่หลากหลายมากขึ้น แต่ยังมีข้อกำจัดเมื่อทำงานร่วมกับระบบ Android ดังนี้

1. ความแตกต่างระหว่างภาษา คือ เนื่องจาก OpenCV ถูกพัฒนาหลักด้วย C++ เวลามาใช้งานบนระบบ Android ซึ่งภาษาเขียนเป็น Java จึงมีทางออกหลักๆ 2 ทางคือ เขียนโค้ดบน Android ด้วย Java (ใช้ได้เลย โดยการใช้ Instant run ได้) หรือ เขียนโค้ดบน Android ด้วย C++ (แต่การทำงานในลักษณะนี้ ต้องลง LLDC, CMake หรือ NDK เพิ่ม และใช้ Instant run ไม่ได้)

2. กระบวนการติดตั้งและการใช้งานค่อนข้างไม่เสถียรและยังมีการเปลี่ยนตามเวอร์ชันของ OpenCV และ Android Studio อยู่ตลอดทำให้เวลาหาข้อมูลเพื่อนำมาอ้างอิงหรือลองทดสอบอาจจะใช้ไม่ได้กับเครื่องบางรุ่น

3. ไฟล์แอปพลิเคชั่นค่อนข้างใหญ่ (ประมาณ 20 MB หรือมากกว่า) คือต้องมีการผนวก library ของ OpenCV เข้าไปด้วย แต่มีทางเลือกนึงคือให้ผู้ใช้ไปติดตั้ง OpenCV library (OpenCV Manager) ก่อนบน Play store แล้วแอปพลิเคชั่นจะทำให้ขนาดเล็กลงได้ แต่นั่นจะไม่สะดวกต่อผู้ใช้ [20]

2.11.8 Firebase (ไฟร์เบส)

ในปัจจุบันธุรกิจได้นำระบบต่างๆ เข้ามาช่วยในการทำงาน เมื่อก่อนจะใช้แรงงานมนุษย์ในการควบคุมและปฏิบัติเกือบทั้งหมด ดังนั้นการนำระบบเทคโนโลยีที่ทันสมัยเข้ามาช่วยดำเนินธุรกิจจะสามารถช่วยลดความเสี่ยงในการทำงาน การคิดวิเคราะห์ที่แม่นยำ และความรวดเร็ว หรือแม้แต่การขายของ การติดต่อสื่อสาร การเก็บข้อมูลจำนวนมากๆ และการให้บริการต่างๆบนหน้าเว็บไซต์ ทั้งหมดนี้เป็นการใช้เทคโนโลยีที่ทันสมัยเข้ามาเพิ่มความสะดวก โดยที่ Google ได้ซื้อโปรเจคชื่อว่า Firebase (ไฟร์เบส) บริการ backend และ แพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอพพลิเคชั่น

Firebase (ไฟร์เบส) คือ บริการ backend และ แพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอพพลิเคชั่น และโปรแกรมประยุกต์บนเว็บแพลตฟอร์มที่มีเครื่องมือและโครงสร้างพื้นฐานที่ได้รับการออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอพพลิเคชั่นพลิเคที่มีคุณภาพสูง Firebase (ไฟร์เบส) ถูกสร้างขึ้นจากคุณสมบัติเสริมว่านักพัฒนาสามารถผสมและจับคู่เพื่อให้พอดีกับความต้องการของตน บริษัท ก่อตั้งขึ้นในปี 2011 โดยแอนดรูลีและเจมส์ เทมปลิน สินค้าเริ่มต้น Firebase (ไฟร์เบส) เป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่จะช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล **Google Firebase- 2.0** กูลเกิลได้ซื้อกิจการ Firebase (ไฟร์เบส) และมีการพัฒนาให้สามารถ จากบริการ backend เก็บข้อมูลอย่างเดียว มาเป็น แพลตฟอร์ม ครบวงจรสำหรับนักพัฒนาแอพพลิเคชั่น (รองรับ iOS, Android, Web) รองรับบริการเกือยทุกอย่างที่นักพัฒนาแอพพลิเคชั่นต้องใช้งาน โดยสิ่งที่ Firebase ให้บริการมีดังนี้

Firebase Analytics บริการวิเคราะห์ข้อมูล ดึงเทคโนโลยีมาจาก Google Analytics และยังเปิดให้ใช้แบบไม่คิดค่าบริการแบบไม่จำกัดปริมาณข้อมูลใดๆ

ระบบส่งข้อความแจ้งเตือน **Google Cloud Messaging (GCM)** ภายหลังเปลี่ยนชื่อมาเป็น Firebase Cloud Messaging (FCM) ใช้งานฟรีไม่จำกัดปริมาณข้อความ โดยกูเกิลกล่าวว่าตอนนี้ FCM ให้บริการข้อความแจ้งเตือนถึง 1.7 แสนล้านข้อความต่อวัน

Firebase Storage บริการพื้นที่เก็บข้อมูล นำไว้เก็บภาพ วิดีโอ หรือไฟล์ขนาดใหญ่จากแอพพลิเคชั่นพลิเคชั่นของผู้ใช้ โดยสร้างอยู่บน Google Cloud Storage

Firebase Remote Config ตัวช่วยอัพเดตคอนฟิกของแอพพลิเคชั่น สำหรับการปรับแต่งค่าต่างๆ ในแอพพลิเคชั่นพลิเคชั่นจากระยะไกล (ตัวอย่างเช่น เกมที่ต้องปรับสมดุลของเกมตลอดเวลา) สามารถใช้ร่วมกับ Firebase **Analytics** เพื่อกำหนดผู้ใช้งานแยกเป็นกลุ่มๆ ได้

Firebase Crash Reporting ตัวรายงานการแครชของแอพพลิเคชั่นพลิเคชั่น รองรับทั้ง iOS และ Android

บริการทดสอบแอพพลิเคชั่นบนฮาร์ดแวร์จริง Cloud Test Lab ที่เปิดตัวเมื่อปี 2016 แล้วเปลี่ยนชื่อมาเป็น Firebase Test Lab for Android

Firebase Notifications เป็นคอนโซลสำหรับนักพัฒนา เพื่อยิงข้อความผ่าน FCM ไปยังผู้ใช้งาน สำหรับโปรโมทหรือกระตุ้นให้ผู้ใช้กลับมาเปิดแอพพลิเคชั่น

Firebase Dynamic Links บริการ URL กลาง ที่สามารถชี้ทางไปยังเพจต่างๆ แปรผันตามอุปกรณ์หรือคุณสมบัติของผู้ใช้ ตัวอย่างเช่น แต่ละประเทศกดลิงก์เดียวกัน เข้าคนละเพจกัน

Firebase Invites ระบบเชิญเพื่อนมาใช้แอพพลิเคชั่น มีฟีเจอร์ referral คนชวนจะได้รับสิทธิประโยชน์

Firebase App Indexing มีการเปลี่ยนชื่อมาจาก Google App Indexing ที่ช่วยให้ Google Search ค้นเจอเนื้อหาภายในแอพพลิเคชั่น[10]

ภาพที่ 2.26 บริการต่างๆของ Firebase Platform

2.12 การวิเคราะห์สหสัมพันธ์และการวิเคราะห์การถดถอย

การวิเคราะห์สหสัมพันธ์ เป็นสถิติวิเคราะห์ที่ใช้เพื่อศึกษาความเกี่ยวข้องเชื่อมโยงระหว่างตัวแปรสองตัวแปร ที่เป็นตัวแปรเมทริกซ์ โดยที่ความสัมพันธ์ระหว่างตัวแปรในความเป็นจริงอาจจะเป็นความสัมพันธ์ทางเดียวหรือความสัมพันธ์เชิงสาเหตุ หรือความสัมพันธ์สองทาง หรือไม่มีความสัมพันธ์กันก็ได้ ผลการวิเคราะห์สหสัมพันธ์จะทำให้ทราบข้อเท็จจริงเกี่ยวกับลักษณะความสัมพันธ์ระหว่างตัวแปรทั้งสองตัวแปรรวม 4 ประการคือ

1. มีความสัมพันธ์กันจริงหรือไม่
2. ขนาดความสัมพันธ์มากน้อยเท่าไร
3. ทิศทางความสัมพันธ์เป็นแบบสอดคล้องหรือแบบผกผัน
4. ความแปรปรวนร่วมระหว่างตัวแปรทั้งสองมีมากน้อยเท่าไร

การวิเคราะห์ถดถอย เป็นสถิติวิเคราะห์ที่ใช้เพื่อศึกษาความสัมพันธ์เชิงสาเหตุ ตามกรอบแนวคิดเชิงทฤษฎี ในกรณีที่มีตัวแปรต้น หรือตัวทำนาย ตั้งแต่หนึ่งตัวขึ้นไป และมีตัวแปรตาม หรือตัวแปรผล หนึ่งตัวแปร โดยที่ตัวแปรทั้งตัวแปรต้นและตัวแปรตามต้องเป็นตัวแปรเมทริกซ์ ผลการวิเคราะห์ถดถอย ทำให้ได้ผลการวิเคราะห์ข้อมูลที่ใช้ประโยชน์ได้ 5 ประการ คือ

1. บอกปริมาณความแปรปรวนในตัวแปรตามที่อธิบายได้ด้วยตัวแปรต้น
2. เปรียบเทียบขนาดและทิศทางอิทธิพลของตัวแปรต้นแต่ละตัวที่มีต่อตัวแปรตาม
3. สร้างสมการพยากรณ์ตัวแปรตามได้เมื่อรู้ค่าตัวแปรต้น
4. ศึกษาอิทธิพลหลัก อิทธิพลปฏิสัมพันธ์ระหว่างตัวแปรต้นที่มีต่อตัวแปรตาม
5. ศึกษาเปรียบเทียบค่าเฉลี่ยกรณีประชากรหลายกลุ่มได้ แบบเดียวกับผลการวิเคราะห์จากการวิเคราะห์ความแปรปรวน (ANOVA)

2.13 ประเภทของการวิเคราะห์สหสัมพันธ์และการถดถอย

การวิเคราะห์สหสัมพันธ์ และการวิเคราะห์การถดถอย ที่ใช้ในการวิเคราะห์ข้อมูล แบ่งออกได้เป็นหลายประเภทแตกต่างกัน ตามเกณฑ์ที่ใช้ในการจัดแบ่งประเภท ดังต่อไปนี้

2.13.1 ประเภทของการวิเคราะห์สหสัมพันธ์

การวิเคราะห์สหสัมพันธ์ แบ่งตามลักษณะข้อมูลได้เป็น 3 ประเภท ดังนี้

1. การวิเคราะห์สหสัมพันธ์แบบง่าย (Simple correlation analysis) หมายถึง สถิติวิเคราะห์ที่ใช้ในการศึกษาความสัมพันธ์ระหว่างตัวแปรต้น 1 ตัวแปร และตัวแปรตาม 1 ตัวแปร การวิเคราะห์สหสัมพันธ์แบบง่าย ยังแบ่งตามระดับการวัดของตัวแปรได้อีกหลายแบบ ได้แก่ สหสัมพันธ์เพียร์สัน ซึ่งเป็นสถิติพาราเมทริกซ์ สหสัมพันธ์แต่ละแบบเหมาะสมกับข้อมูลที่มีระดับการวัดต่างกัน ดังตารางต่อไปนี้

ตารางที่ 2.5 สหสัมพันธ์ประเภทต่างๆ จำแนกตามระดับการวัดของตัวแปรสองตัวแปร

ระดับการวัด	นามบัญญัติ	นันเมทริกซ์ 2 ค่า	นันเมทริกซ์หลายค่า	เมทริกซ์
นามบัญญัติ	สปส.ฟาย			
นันเมทริกซ์ 2 ค่า	สปส.ฟาย	สหสัมพันธ์เตตราคลอริก		
นันเมทริกซ์หลายค่า			สหสัมพันธ์โพลีคลอริก สหสัมพันธ์สเปียร์แมน	
เมทริกซ์		สหสัมพันธ์ไบซีเรียล	สหสัมพันธ์โพลีซีเรียล สหสัมพันธ์สเปียร์แมน	สหสัมพันธ์เพียร์สัน

2. การวิเคราะห์สหสัมพันธ์พาร์เชียล (Partial correlation analysis) หมายถึง สถิติวิเคราะห์ที่ใช้ในการศึกษาความสัมพันธ์ที่แท้จริงระหว่างตัวแปรสองตัว แบบมีการควบคุมโดยการขจัด (partial out) อิทธิพลจากตัวแปรแทรกซ้อน แบ่งได้เป็น 3 ประเภท คือ สหสัมพันธ์แบบง่ายหรือสหสัมพันธ์ที่ไม่มีการควบคุม (simple correlation or zero-order correlation) สหสัมพันธ์พาร์เชียลที่มีการควบคุมตัวแปรแทรกซ้อน k ตัวแปร (k^{th}- order partial correlation) โดยการขจัดความแปรปรวนของตัวแปรแทรกซ้อนออกจากตัวแปรทั้งสองตัวแปรที่นำมาศึกษาความสัมพันธ์ และสหสัมพันธ์กึ่งพาร์เชียล ที่มีการควบคุมตัวแปรแทรกซ้อน k ตัวแปร (k^{th}- order semi-partial correlation) โดยการขจัดความแปรปรวนของตัวแปรแทรกซ้อนออกจากตัวแปรที่นำมาหาความสัมพันธ์เพียงตัวเดียว

3. การวิเคราะห์สหสัมพันธ์พหุคูณ (Multiple correlation analysis) หมายถึง สถิติวิเคราะห์ที่ใช้ในการศึกษาความสัมพันธ์ระหว่างตัวแปรต้นตั้งแต่สองตัวแปรขึ้นไป กับตัวแปรตามตั้งแต่สองตัวแปรขึ้นไป เป็นสถิติวิเคราะห์ที่เกี่ยวข้องกับการวิเคราะห์จำแนก และการวิเคราะห์ความแปรปรวนตัวแปรพหุนาม ซึ่งเป็นสถิติวิเคราะห์ขั้นสูง

2.13.2 ประเภทของการวิเคราะห์ถดถอย

การวิเคราะห์การถดถอยแบ่งตามลักษณะของข้อมูล และสมการถดถอย หรือโมเดลวิเคราะห์ถดถอย (regression analysis model) ที่แสดงความสัมพันธ์ระหว่างตัวแปรต้นและตัวแปรตาม ได้เป็น 4 ประเภท ดังนี้

1. โมเดลการวิเคราะห์ถดถอยพหุคูณ (multiple regression analysis model) เป็นโมเดลการวิเคราะห์ถดถอยใช้ศึกษาความสัมพันธ์ระหว่างตัวแปร Y กับตัวแปร X_1, X_2, ... เมื่อตัวแปรมีความสัมพันธ์แบบเส้นตรง ดังสมการ

$$Y = b_0 + b_1X_1 + b_2X_2 + \cdots + e \qquad (2.9)$$

2. โมเดลการวิเคราะห์ถดถอยโพลีโนเมียล (polynomial regression analysis model) เป็นโมเดลการวิเคราะห์ถดถอยใช้ศึกษาความสัมพันธ์ระหว่างตัวแปรต้นตัวเดียว และตัวแปรตามตัวเดียว เมื่อมีความสัมพันธ์แบบเส้นโค้ง ดังสมการ

$$Y = b_0 + b_1X + b_2X^2 + \cdots + e \qquad (2.10)$$

3. โมเดลการวิเคราะห์ถดถอยพหุคูณมีตัวแปรดัมมี่ (multiple regression analysis model with dummy variables) เป็นโมเดลการวิเคราะห์ถดถอย ใช้ศึกษาความสัมพันธ์ระหว่างตัวแปรตามหนึ่งตัวแปร กับตัวแปรต้นที่เป็นตัวแปรนันเมทริกซ์ ที่ได้รับการให้รหัสใหม่ (recode) สามารถใช้ศึกษาได้ทั้งกรณีที่ตัวแปรมีความสัมพันธ์แบบเส้นตรงและเส้นโค้ง ได้ผลการวิเคราะห์เช่นเดียวกันกับโมเดลการวิเคราะห์ความแปรปรวน (ANOVA model) ดังสมการ เมื่อมี D_1, D_2, ... D_{m-1} เป็นตัวแปรดัมมี่ที่สร้างขึ้นแทนตัวแปรต้น X ซึ่งเป็นตัวแปรนันเมทริกซ์ที่มี m ค่า ดังนี้

$$Y = b_0 + b_1D_1 + b_2D_2 + \cdots + D_{m-1} + e \qquad (2.11)$$

4. โมเดลการวิเคราะห์ถดถอยพหุคูณมีเทอมปฏิสัมพันธ์ (multiple regression analysis model with interaction term) เป็นโมเดลการวิเคราะห์ถดถอยใช้ศึกษาความสัมพันธ์ระหว่างตัวแปรกรณีที่มีอิทธิพลทั้งอิทธิพลหลักจาก X, Z และอิทธิพลปฏิสัมพันธ์ (X*Z) ต่อตัวแปรตาม ดังสมการ

$$Y = b_0 + b_1X + b_2Z + b_2XZ \cdots + e \qquad (2.12)$$

นอกจากนี้ยังมีการพัฒนาโมเดลการวิเคราะห์ถดถอยเพื่อศึกษาความสัมพันธ์ระหว่างตัวแปรในรูปเส้นโค้งตัว ‘S’ และโมเดลแบบอื่นๆ เช่น การวิเคราะห์ถดถอยโลจิสติก (logistic regression analysis) การวิเคราะห์ล็อก-ลิเนียร์ (log-linear analysis) การวิเคราะห์โลจิท (logit analysis) เป็นต้น

2.14 สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน (Pearson product-moment correlation coefficient)

สัมประสิทธิ์สหสัมพันธ์แบบเพียร์สัน เป็นวิธีที่ใช้วัดความสัมพันธ์ระหว่างตัวแปร หรือ ข้อมูล 2 ชุด โดยที่ตัวแปร หรือข้อมูล 2 ชุดนั้นจะต้องอยู่ในรูปของข้อมูลในมาตราอันตรภาคหรืออัตราส่วน (Interval or Ratio scale) เรียกว่า "สัมประสิทธิ์สหสัมพันธ์" เช่น การหาความสัมพันธ์ระหว่างภาวะสุขภาพกับการดูแลตนเอง การหาความสัมพันธ์ระหว่างน้ำหนักแรกเกิดของทารกกับอายุของมารดา เป็นต้น โดยมีเงื่อนไขบื้องต้น ดังนี้ [4]

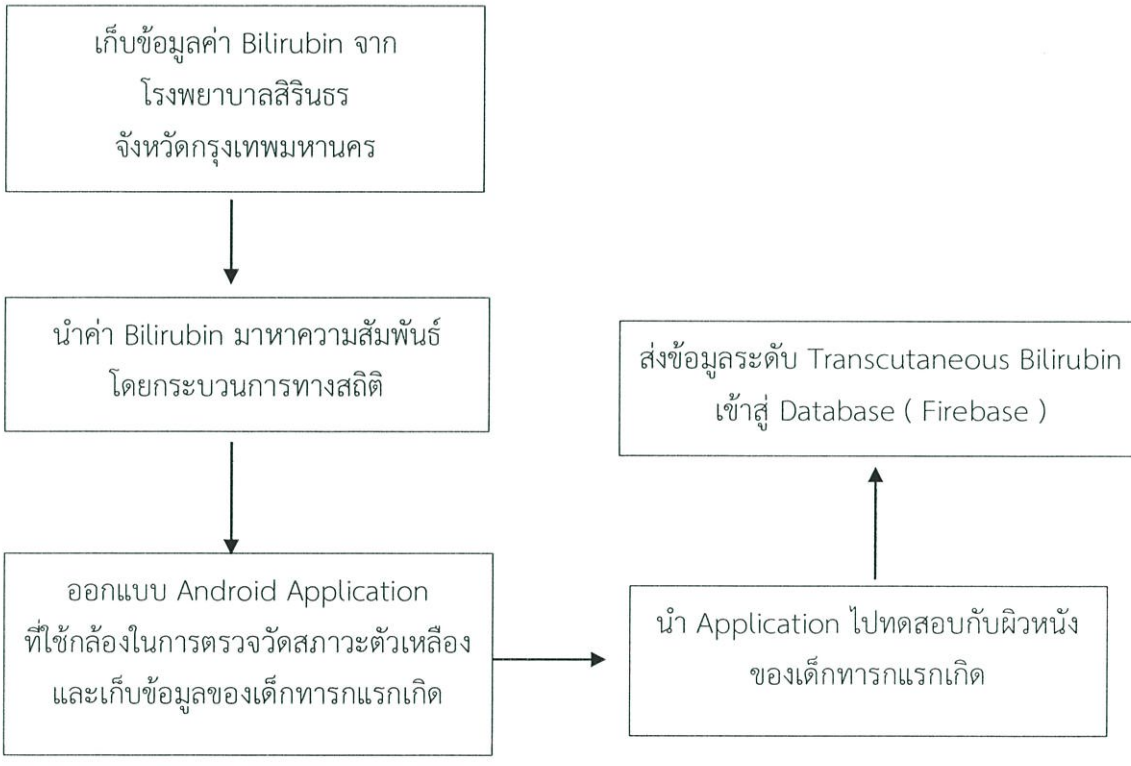
1. ตัวแปรหรือข้อมูลทั้ง2 ชุดอยู่ในมาตราอันตรภาค หรือมาตราอัตราส่วน
2. ข้อมูลทั้ง2 ชุด มีการแจกแจงแบบปกติ และมีความสัมพันธ์เชิงเส้นตรง
3. ข้อมูลในแต่ละชุดจะต้องมีความเป็นอิสระต่อกันโดยค่าสัมประสิทธิ์สหสัมพันธ์โดยปกตินั้นจะมีค่าอยู่ระหว่าง -1.00 ถึง 1.00 ซึ่งแปลผลได้ว่า
   - 3.1 ถ้ามีค่าติดลบหมายความว่า ตัวแปร 2 ตัว มีความสัมพันธ์ในทิศทางตรงกันข้าม
   - 3.2 ถ้ามีค่าเป็นบวกหมายความว่า ตัวแปร 2 ตัวมีความสัมพันธ์ในทิศทางเดียวกัน
   - 3.3 ถ้ามีค่าเป็น 0 หมายความว่าตัวแปร 2 ตัวไม่มีความสัมพันธ์กัน

บทที่ 3
การออกแบบระบบ

การทำโครงงานวิจัยในส่วนนี้จะแสดงถึงการออกแบบแอนดรอยด์แอพพลิเคชั่น (Android Application) ที่ใช้ในการประเมินค่า Transcutaneous digital Bilirubin จากการถ่ายภาพด้วยกล้องจากโทรศัพท์ที่ใช้ระบบปฏิบัติการ Android ในบริเวณผิวหนัง รวมถึงขั้นตอนการหาความสัมพันธ์ระหว่างโมเดลสีและระดับบิลิรูบินจากการตรวจเลือด โดยในบทนี้จะเป็นการอธิบายถึงการหาสมการ, หลักการทำงานของแอพพลิเคชั่น, การส่งผ่านข้อมูลผ่านเซิร์ฟเวอร์ ซึ่งมีรายละเอียดดังต่อไปนี้

งานวิจัยชิ้นนี้ประกอบไปด้วย 3 ส่วน คือ

1. การหาสมการ Total Transcutaneous digital bilirubin
2. การออกแบบ Android Application
3. การส่งข้อมูลผ่านเซิร์ฟเวอร์



3.1 การหาสมการ Total Transcutaneous Digital Bilirubin

ความสัมพันธ์ทางสถิติระหว่างค่าบิลิรูบินที่หาได้จากวิธีการทางห้องปฏิบัติการกับค่าที่ได้จากวิธีการทางการประมลผลภาพนั้น มีการวิเคราะห์โดยเริ่มจากการเก็บข้อมูลดังตารางต่อไปนี้

ตารางที่ 3.1 ตัวอย่างตารางบันทึกผลค่าบิริรูบินจากการตรวจทางห้องปฏิบัติการของเด็กแรกเกิด

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	Total serum bilirubin (mg/dL)	ภาพถ่าย
1				
2				
3				

ข้อมูลภาพที่นำมาใช้ในการวิเคราะห์สมการมีดังนี้

1. กล้องถ่ายภาพดิจิตัล Nikon D5300 + Lens 35mm f1.4
2. ISO 400
3. speed shutter 1/60
4. White Balance AUTO

เมื่อได้ข้อมูลมาแล้ว จะทำการประมาณค่า Illuminant ของภาพ จากกระบวนการ Simple color balancing ผ่านโปรแกรม Matlab โดยในขั้นตอนนี้จะทำการกำจัดสีที่ทำให้ภาพเพี้ยนออกไปจากภาพโดยการสเกลฮิสโตแกรมของโมเดล RGB จากนั้นจะทำการนำข้อมูลภาพที่ได้มาตรวจสอบหาค่าสี RGB และ CMYK โดยกำหนด L* = 70% จาก L* a* b* color system เพื่อลดความแปรปรวนของสีในภาพ จากนั้นจะนำค่า Y และ M ไปใช้เพื่อหาความสัมพันธ์ทางสถิติระหว่างค่า Y และ M และ Total serum bilirubin ที่ได้จากวิธีการทางห้องปฏิบัติการ

3.1.1 Illumination Estimation

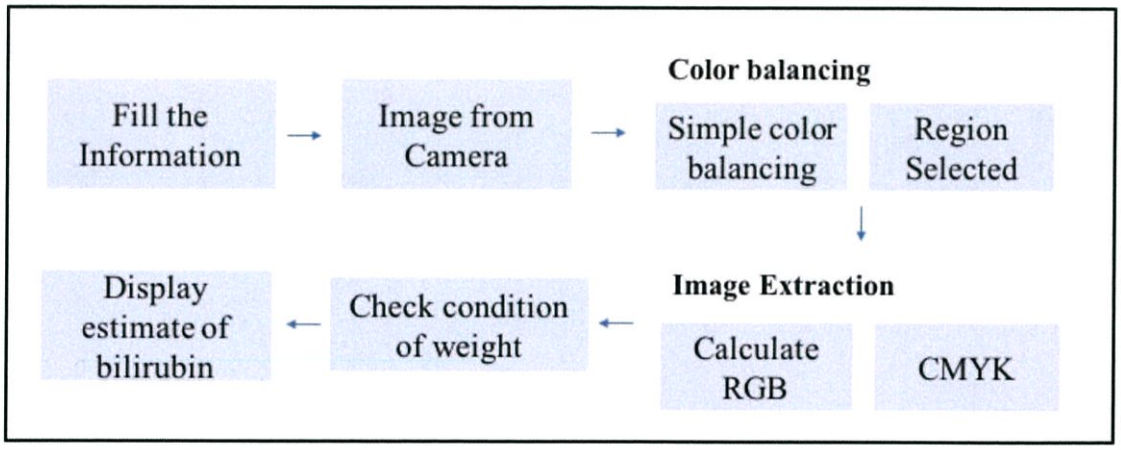
ในโครงงานวิจัยฉบับนี้ใช้กระบวนการ Simplest color balance algorithm โดยกระบวนการนี้เป็นการกำจัดสีที่ทำให้ภาพเพี้ยน (Color cast) ออกไปจากภาพ จากสมมุติฐานที่ว่าค่าที่สูงที่สุดของ RGB components จะสัมพันธ์กับสีขาว และค่าต่ำสุดจะสัมพันธ์กับสีดำ กระบวนการนี้จะทำการยืดการสเกลฮิสโตแกรม ของ RGB components ครอบคลุมช่วงสูงที่สุดที่เป็นไปได้ [0,255] โดยการทำ affine transform ในแต่ละช่องของ RGB

3.2 Android Application

การทำงานบน Android Application จะมีขั้นตอนการทำงาน โดยเริ่มจากการถ่ายภาพบริเวณผิวหนังที่ต้องการตรวจวัด จากนั้นจะทำการ Crop ภาพเพื่อจำกัดส่วนที่ไม่ต้องการออก จากนั้นจะนำไปหาค่า RGB จากภาพ เพื่อนำค่า RGB เหล่านี้ไปคำนวณหาค่าจากสมการ (3.1)

$$K = 1 - \max(R', G', B'), C = \frac{1-R'-K}{1-K}, M = \frac{1-G'-K}{1-K}, Y = \frac{1-B'-K}{1-K} \qquad (3.1)$$

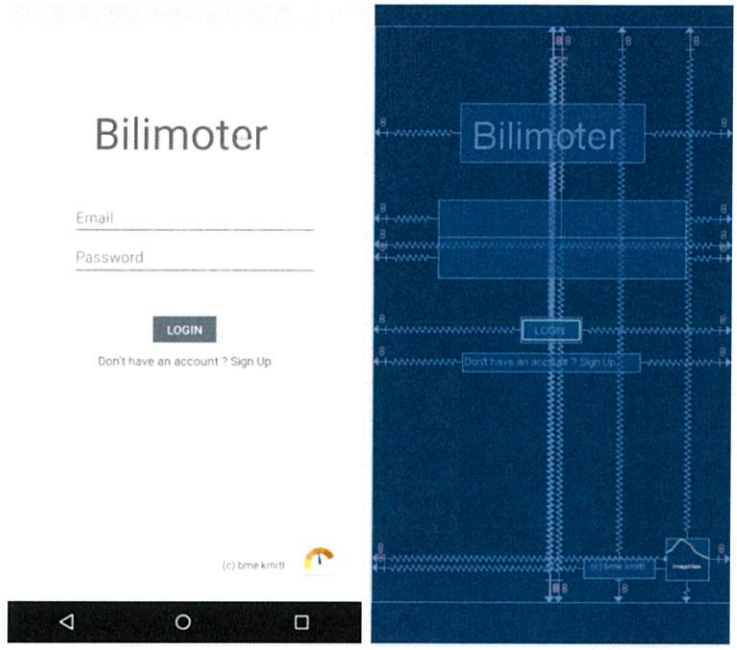
โดยที่ R' = R/255, G' = G/255, B'=B/255 เพื่อนำค่า CMYK ที่ได้มาหาค่า Transcutaneous digital Bilirubin และนำมาตรวจสอบกับเงื่อนไขที่กำหนดไว้ถ้าหากว่าค่า Transcutaneous digital Bilirubin นั้นอยู่ในช่วงระดับปกติที่กำหนดจะแสดงผลผ่านทางจอว่า Jaundice แต่ถ้าหากว่าค่า Transcutaneous Bilirubin นั้นไม่ได้อยู่ในเงื่อนไขจะแสดงผลผ่านทางจอว่า Normal แล้วจากนั้นจะเข้าสู่หน้าต่างการเก็บข้อมูล โดยจะเก็บข้อมูลของเด็กทารก ซึ่งประกอบด้วย ชื่อ - นามสกุล ,วันเกิด , น้ำหนัก จากนั้นจะส่งข้อมูลและค่า Transcutaneous digital Bilirubin ที่ได้นั้นเข้าสู่ Database โดยในส่วนของการออกแบบแอปพลิเคชันจะมี Activity ที่ใช้ในการเขียนโค้ดคำสั่งและออกแบบหน้าต่างของแอพพลิเคชั่น โดยการทำงานของแอปพลิเคชันจะอธิบายในแต่ละส่วนของ Activity ดังนี้



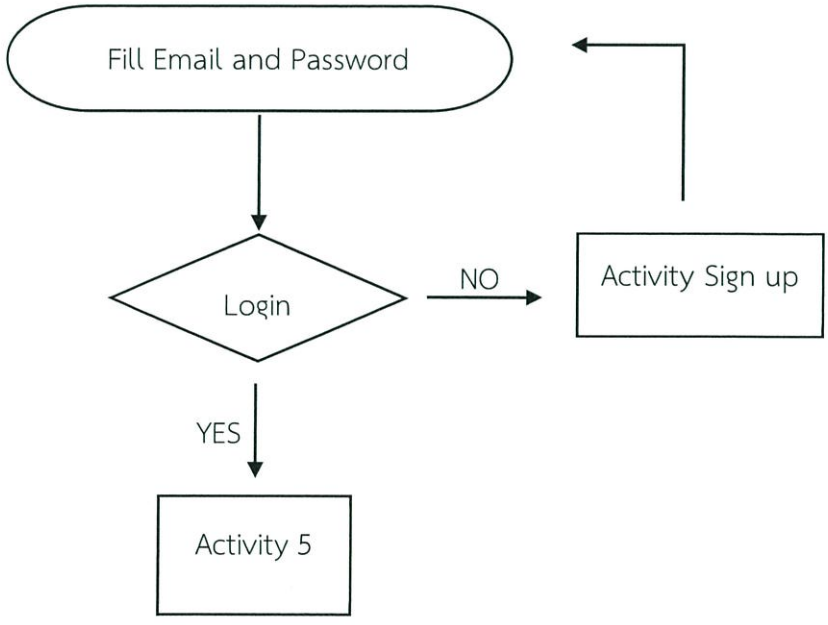
ภาพที่ 3.1 แผนผังแสดงการทำงานของแอปพลิเคชัน

3.2.1 Activity Login

ในส่วนของ Activity Login จะเป็นส่วนที่ใช้ในการ Login โดยใช้ Email และ Password เพื่อเข้าใช้งาน Application และมีส่วนที่ใช้เชื่อมโยงไปยัง Activity Signup ซึ่งใช้ในการ Sign up ถ้าหากผู้ใช้งานไม่ยังไม่มีข้อมูลในระบบ ดังภาพที่ 3.2



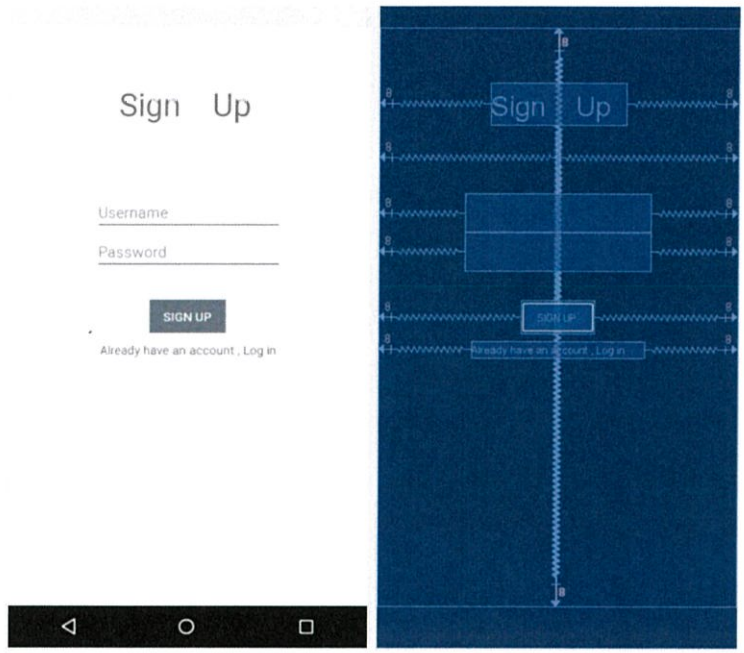
ภาพที่ 3.2 Layout Activity Login



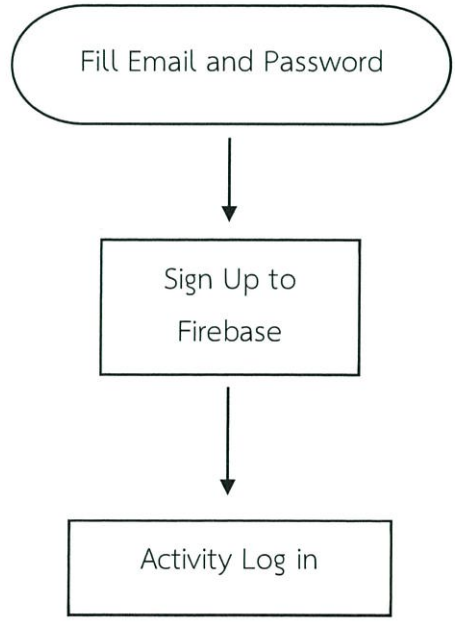
ภาพที่ 3.3 แผนผังขั้นตอนการทำงานใน Activity Login

3.2.2 Activity Sign Up

ในส่วนของ Activity Sign up จะเป็นส่วนที่ใช้ในการ Sign up โดยต้องทำการกรอก Email โดยใช้เป็นฟอร์มของ Gmail และ Password จากนั้นเมื่อทำการกด Sign up แอปพลิเคชันจะทำการลงทะเบียน Email และ Password เข้าไปสู่ Database เพื่อใช้ในการยืนยันตัวตนในการเข้าใช้งานApplication และเก็บข้อมูลของแต่ละ User ดังภาพที่ 3.4



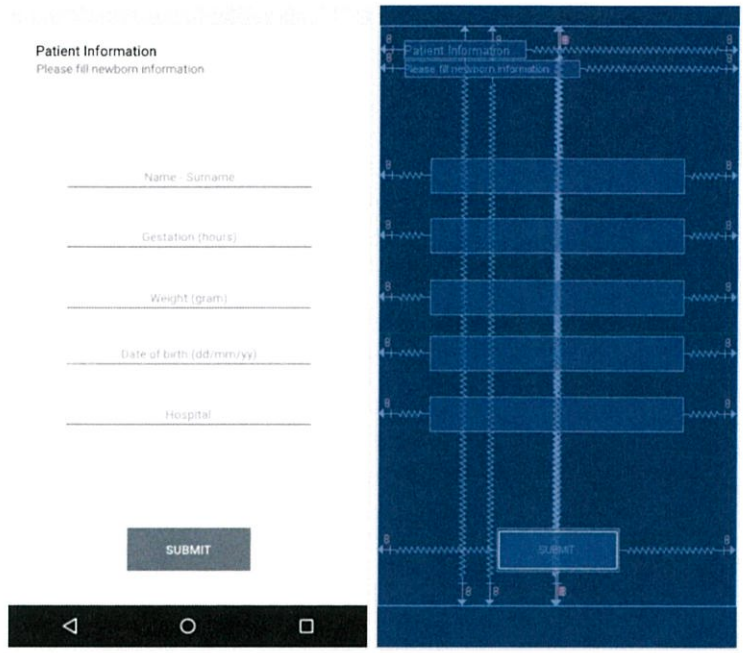
ภาพที่ 3.4 Layout Activity Sign up



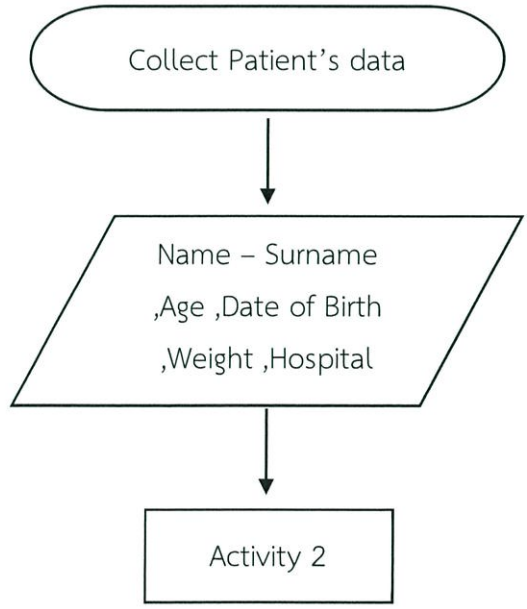
ภาพที่ 3.5 แผนผังขั้นตอนการทำงานใน Activity Sign up

3.2.3 Activity 5

ในส่วนของ Activity 5 จะเป็นส่วนในการเก็บข้อมูลของเด็กทารก ซึ่งประกอบไป ด้วย ชื่อ – นามสกุล ,อายุ ,น้ำหนัก ,วันเดือนปีเกิด และโรงพยาบาล จากนั้นส่งข้อมูลไปยังส่วนถัดไป หรือส่งเข้าผ่าน database (Firebase) โดยมีการจัดวาง Layout ดังภาพที่ 3.6



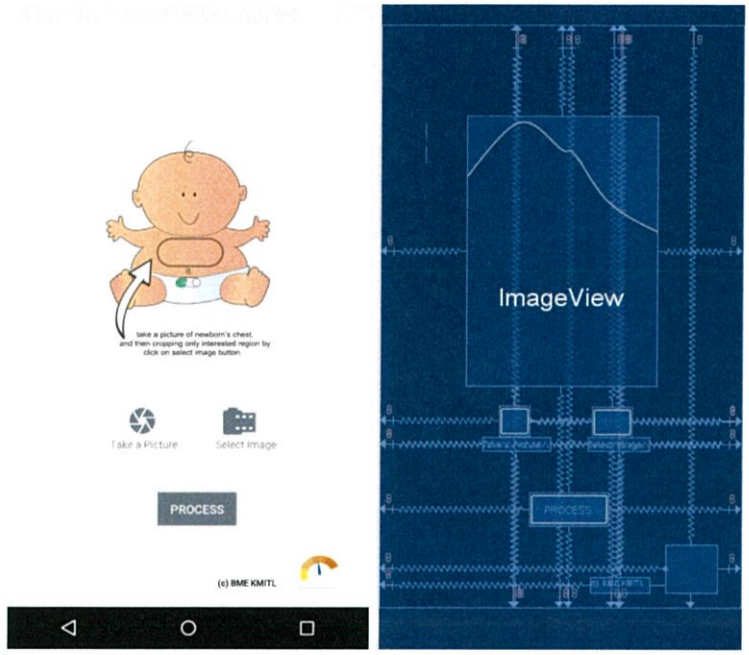
ภาพที่ 3.6 Layout Activity 5



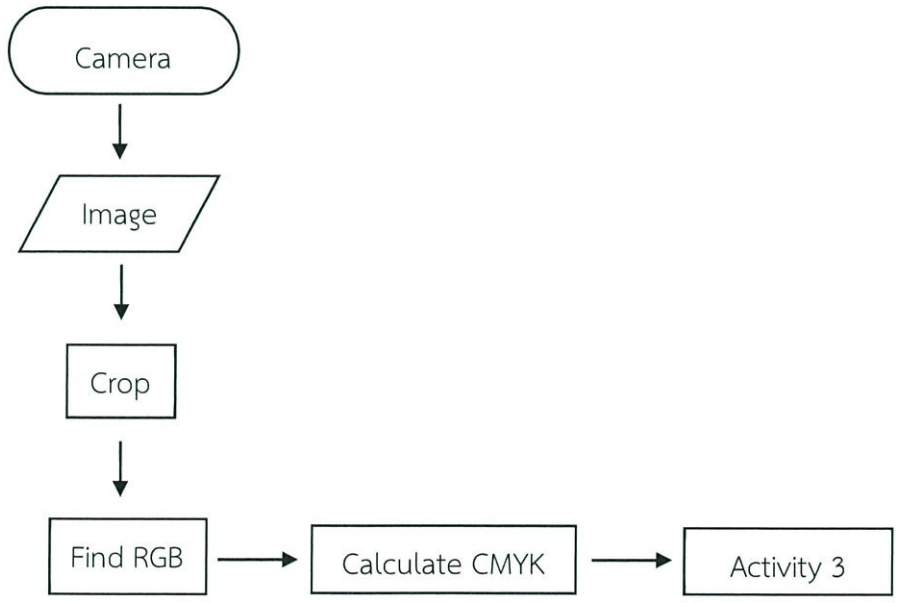
ภาพที่ 3.7 แผนผังแสดงขั้นตอนการทำงานใน Activity 5

3.2.4 Activity 2

ในส่วนของ Activity 2 นี้ จะเป็นส่วนของการทำงานในการเรียกใช้ฟังก์ชันการถ่ายรูปจากการกดปุ่ม Button เพื่อทำการถ่ายภาพ และมีการใช้งานของฟังก์ชัน Crop เพื่อ กำจัดส่วนที่ไม่ได้ต้องการออกจากภาพ แล้วนำมาหาค่า RGB จากภาพ แล้วนำมาคำนวณสู่ ค่า CMYK จากสมการ และนำค่าเหล่านี้ส่งไปแสดงผลยัง Activity 3 ต่อไป โดยมีการจัดวาง Layout ดังภาพที่ 3.8



ภาพที่ 3.8 Layout Activity 2



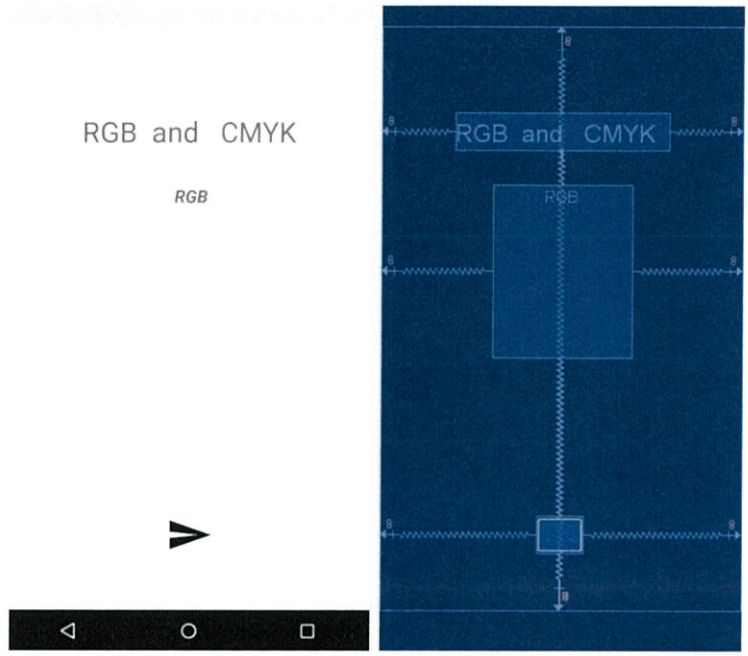
ภาพที่ 3.9 แผนผังแสดงขั้นตอนการทำงานใน Activity 2

3.2.5 Activity 3

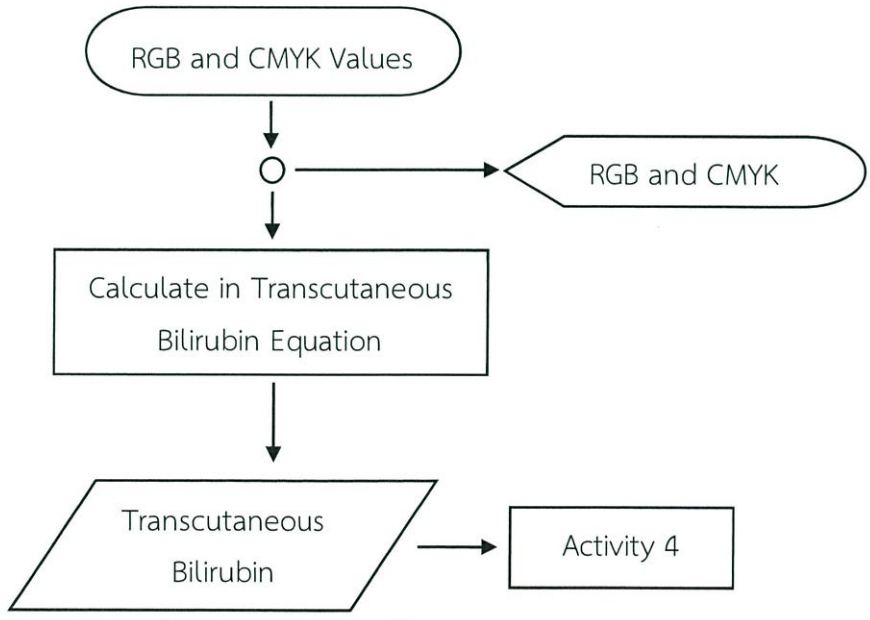
ในส่วนของ Activity 3 เป็นส่วนของการแสดงผลค่า RGB และ CMYK ที่ได้จากในส่วนก่อนหน้าเพื่อมาแสดงผล และนำมาคำนวณสมการในหาค่า Total Transcutaneous digital Bilirubin ดังสมการ (3.2)

$$TcdB = 2.996 + 58.548\,(Y - M) \qquad (3.2)$$

และนำค่าที่ได้จากสมการ ส่งไปยัง Activity 4 โดยมีการจัดวาง Layout ดังภาพที่ 3.10



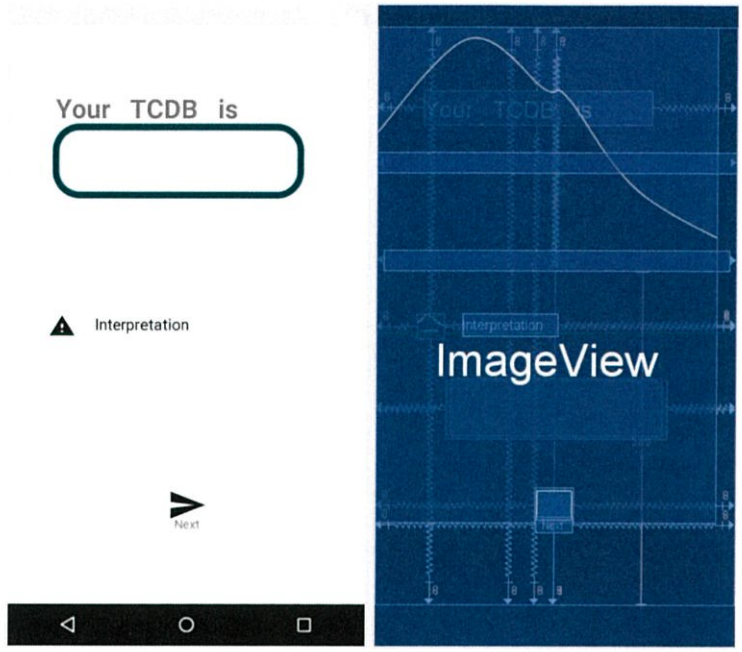
ภาพที่ 3.10 Layout Activity 3



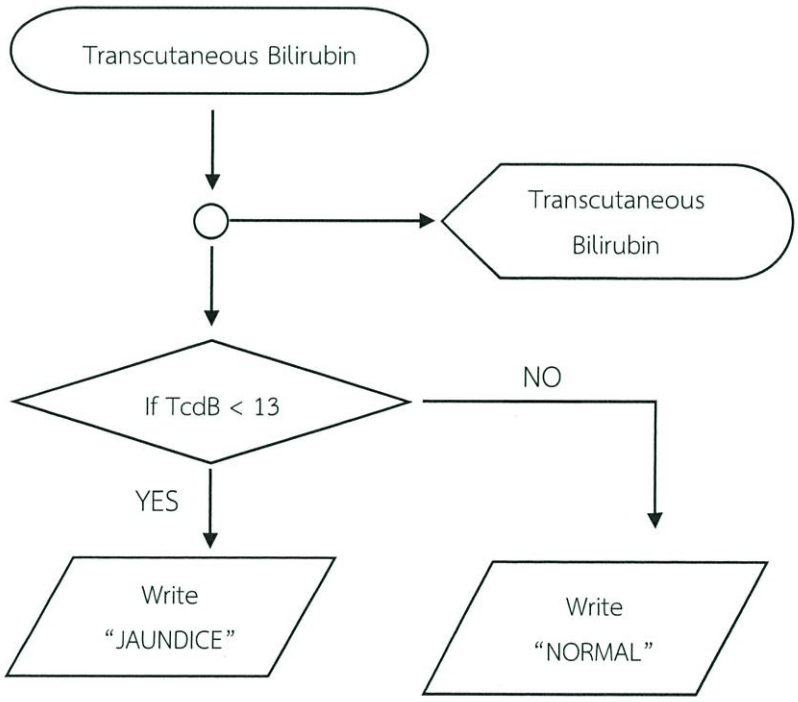
ภาพที่ 3.11 แผนผังแสดงขั้นตอนการทำงานของ Activity 3

3.2.6 Activity 4

ในส่วนของ Activity 4 จะเป็นส่วนของการแสดงผลค่า TcdB และประเมินระดับ ของค่า TcdB ว่าอยู่ในช่วงระดับปกติของ Transcutaneous digital Bilirubin หรือไม่ และ แสดงผลการประเมินผลผ่านทางหน้าจอ โดยมีการจัดเรียง Layout ดังภาพที่ 3.12



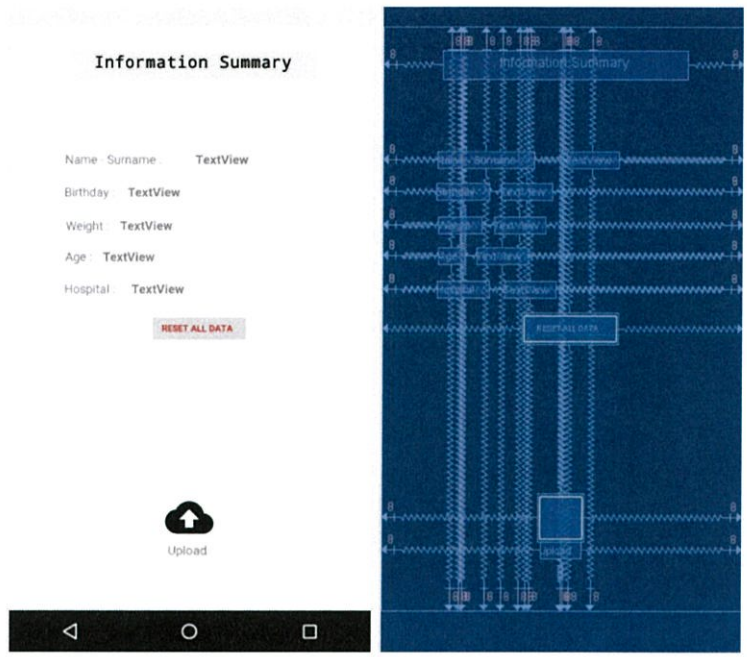
ภาพที่ 3.12 Layout Activity 4



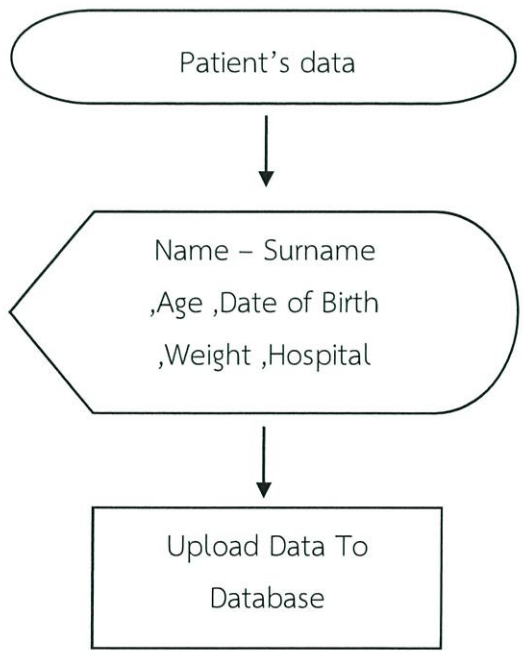
ภาพที่ 3.13 แผนผังแสดงขั้นตอนการทำงานใน Activity 4

3.2.7 Activity 6

ในส่วนของ Activity 6 จะเป็นส่วนที่แสดงข้อมูลของเด็กทารกที่ได้เก็บแล้วนำมาแสดง ประกอบด้วย ชื่อ - นามสกุล, วันเดือนปีเกิด และน้ำหนัก และมีปุ่มที่ใช้ในการ Upload ข้อมูลเหล่านี้ ไปยังDatabase โดยมีการจัดเรียง Layout ดังภาพที่ 3.14



ภาพที่ 3.14 Layout Activity 6



ภาพที่ 3.15 แผนผังแสดงขั้นตอนการทำงานใน Activity 6

3.2.8 AndroidManifest.xml

เป็นไฟล์จัดการการทำงานโดยรวมของแอปพลิเคชัน ซึ่งจะกำหนดคุณสมบัติต่างๆ ของแอพพลิเคชันไว้ในไฟล์นี้เป็นหลัก อย่างเช่น ชื่อแอปพลิเคชัน, เวอร์ชันของแอปพลิเคชัน, API ที่รองรับ Activity ที่จะให้ทำงาน, ขออนุญาติใช้งานบางอย่าง (Permission) เป็นต้น ซึ่งจะเป็นการทำงานโดยรวมของแอปพลิเคชันเท่านั้น โดยในแอพพลิเคชันที่ได้ออกแบบนั้นจะมีการใช้ Permission ในการเรียกใช้การทำงานของ Media Content ภายในโทรศัพท์ และการอ่านและเขียนที่ต้องเข้าถึง Storage ของโทรศัพท์

```
<uses-permission android:name="android.permission.CAMERA">
        //คำสั่งที่ใช้ในการขอเข้าถึงการใช้งานกล้องบนโทรศัพท์
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE">
        //คำสั่งที่ใช้ในการขอเข้าถึงในการอ่านไฟล์ ใน Storage
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE">
        //คำสั่งที่ใช้ในการขอรเข้าถึงในการเขียนไฟล์ ใน Storage
<uses-permission android:name="android.permission.MEDIA_CONTENT_CONTROL">
        //คำสั่งที่ใช้ในการขอการเข้าถึงในส่วนของ Media Content  ทั้งหลาย
<uses-permission android:name="android.permission.INTERNET" />
        //คำสั่งที่ใช้ในการขอการเข้าถึงการใช้งานแอพพลิเคชั่นผ่านระบบ Internet
```

3.2.9 Build Gradle

Gradle คือ build tools ตัวหนึ่งที่จะนำเอา code ที่ทำ ไปให้ compiler ทำการ compile ให้ และนำresources ต่างๆใน package ของมา compile จนกลายเป็น apk ไฟล์ในแอปพลิเคชันที่ได้ออกแบบนั้นจะมีกำหนด sdk ในเวอร์ชันที่เหมาะสมกับระบบปฏิบัติการ Android บนโทรศัพท์ ซึ่งเวอร์ชันที่ใช้คือ 16 ซึ่งเหมาะสมกับ Android 4.4.1 (kitkat) และมีการ Compile ด้วย openCVLibrary249 ซึ่งจะช่วยสนับสนุนกับการใช้งานระบบ Image processing ให้มีประสิทธิภาพมากยิ่งขึ้น

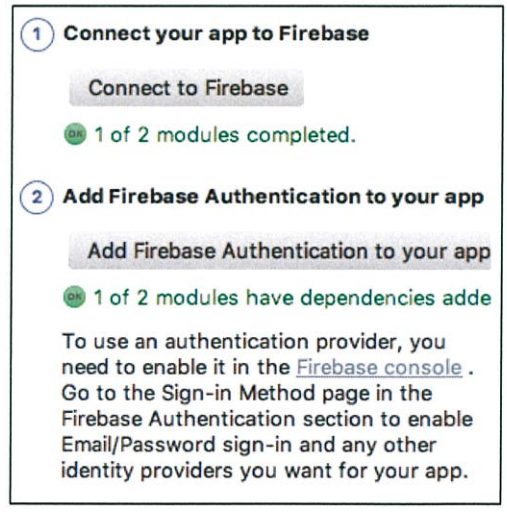
3.3 Firebase

หลังจากผ่านการประมวลผลภาพและได้ค่าบิลิรูบินจากแอปพลิเคชันแล้ว ผู้ใช้งานสามารถส่งข้อมูลผ่านเซิร์ฟเวอร์เพื่อให้ง่ายแต่การเข้าถึงของบุคลากรทางการแพทย์ที่ต้องการเก็บข้อมูลไว้ประกอบการวินิจฉัยในอนาคต ซึ่งทางผู้วิจัยได้ใช้ Firebase ซึ่งเป็น platform ที่ใช้ในการพัฒนาแอปพลิเคชันในสมาร์ทโฟนและเว็บไซต์ของ Google, Inc. โดยจะทำการเชื่อมต่อ database เข้ากับ Android studio

ภาพที่ 3.16 Project setting ของ Firebase

3.3.1 Authentication of Firebase

การทำ Authentication เป็นขั้นตอนที่สำคัญเนื่องจาก ข้อมูลจะต้องผ่านการอนุญาตจากระบบก่อน จึงจะสามารถส่งผ่านไปที่ฐานข้อมูลได้ โดยจะทำการ Authentication ภายใน Android Studio



ภาพที่ 3.17 Authentication ระหว่าง Firebase และ Application

บทที่ 4
ผลการทดลองและอภิปรายผล

การทดลองการใช้งานแอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด โดยศึกษาภาพถ่ายในเด็กทารกแรกเกิดอายุ 0-4 วัน จำนวน 30 ราย เปรียบเทียบกับระดับบิลิรูบินในเลือด และนำภาพนั้นมาประมวลผลทางสถิติสำหรับตอบวัตถุประสงค์ในการวิจัยให้ครบถ้วนตามที่ตั้งไว้ โดยมีลำดับขั้นตอนดังต่อไปนี้

1. ข้อมูลระดับบิลิรูบินในเลือดของเด็กทารกแรกเกิดและการหาความสัมพันธ์ของค่าสีจาก Color Model และระดับบิลิรูบินในเลือดเพื่อหาสมการ Total Transcutaneous digital bilirubin
2. ผลวิเคราะห์สหสัมพันธ์และการถดถอย
3. หน้าต่างแสดงผลการทำงานของแอปพลิเคชัน
4. ผลการทดสอบและการวิเคราะห์ผลจากแอปพลิเคชัน

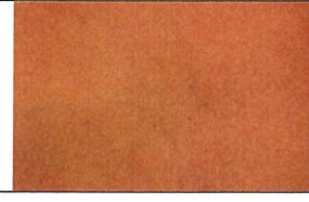
สัญลักษณ์ที่ใช้ในการเสนอผลการวิเคราะห์ข้อมูล

N	แทน	จำนวนคนในกลุ่มตัวอย่าง
Y	แทน	Dependent Variable
X	แทน	Independent Variable
$\bar{X}$	แทน	ค่าเฉลี่ย
S.D.	แทน	ค่าเบี่ยงเบียนมาตรฐาน
Sig	แทน	ระดับนัยสำคัญทางสถิติ (Significant)
*	แทน	นัยสำคัญทางสถิติที่ระดับ 0.05
**	แทน	นัยสำคัญทางสถิติที่ระดับ 0.01 (2-tailed)
α	แทน	ค่าความคลาดเคลื่อนในการทดสอบสมมติฐาน หรือ ระดับนัยสำคัญทางสถิติที่กำหนดไว้ (Level of significance) โดยกำหนดที่ระดับ 0.05
df	แทน	Degree of Freedom
SS	แทน	ผลบวกกำลังสองของคะแนน (Sum of squares)
t	แทน	ค่าสถิติที่ใช้พิจารณาความมีนัยสำคัญจากการแจกแจงแบบ t (t- distribution)
F	แทน	ค่าสถิติที่ใช้พิจารณาความมีนัยสำคัญจากการแจกแจงแบบ F (F- distribution)
H_0	แทน	Null hypothesis

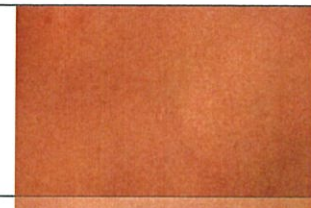
4.1 Preliminary Method

จากบทที่ 3 ซึ่งกล่าวถึงความสัมพันธ์ระหว่างค่าบิลิรูบินที่หาได้จากวิธีการทางห้องปฏิบัติการกับค่าที่ได้จากวิธีการทางการประมวลภาพนั้น มีการเก็บข้อมูลเด็กทารกจำนวน 30 ราย ซึ่งจะแสดงดังตารางต่อไปนี้

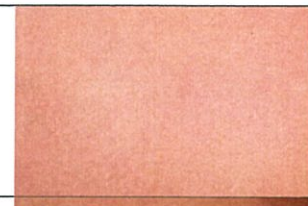
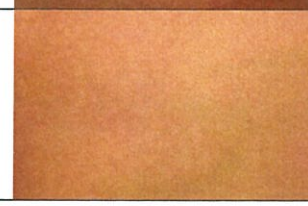
ตารางที่ 4.1 แสดงข้อมูลที่ได้จากการเก็บผลและภาพถ่ายส่วนหน้าอกของเด็กทารก

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	Total serum bilirubin (mg/dL)	ภาพถ่าย	Y-M
1	48	2745	8.9		0.008
2	48	2220	9.7		0.032
3	48	2815	10.1		0.101
4	48	2920	9.8		0.067
5	48	2860	9.5		0.167

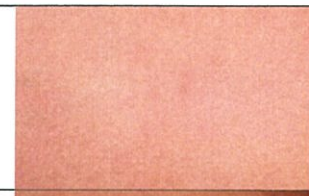
ตารางที่ 4.1 (ต่อ)

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	Total serum bilirubin (mg/dL)	ภาพถ่าย	Y-M
6	48	2530	11		0.151
7	48	3050	9.9		0.098
8	48	2985	10		0.046
9	48	2735	9.1		0.097
10	48	4355	9.7		0.088
11	48	2725	10.4		0.125
12	48	3145	9.4		0.095

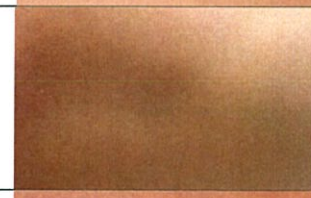
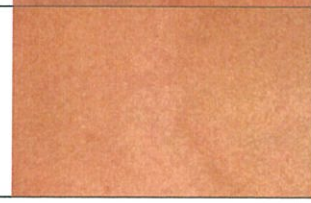
ตารางที่ 4.1 (ต่อ)

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	Total serum bilirubin (mg/dL)	ภาพถ่าย	Y-M
13	48	3550	6.9		0.031
14	48	2360	5.5		0.044
15	48	2790	5.4		0.043
16	48	3045	8.9		0.036
17	48	2990	8.3		0.096
18	48	3775	9		0.087
19	48	3845	8.5		0.037

ตารางที่ 4.1 (ต่อ)

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	Total serum bilirubin (mg/dL)	ภาพถ่าย	Y-M
20	39	2875	1.4		0.007
21	40	2890	10.7		0.098
22	48	2561	4.8		0.025
23	48	2960	9.5		0.072
24	48	3490	9.8		0.095
25	48	2830	7.3		0.019
26	48	3490	10		0.087

ตารางที่ 4.1 (ต่อ)

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	Total serum bilirubin (mg/dL)	ภาพถ่าย	Y-M
27	48	3310	8.3		0.07
28	48	3415	10.6		0.093
29	48	2475	12.4		0.102
30	48	3936	13.6		0.116

4.2 ผลวิเคราะห์สหสัมพันธ์และการถดถอย

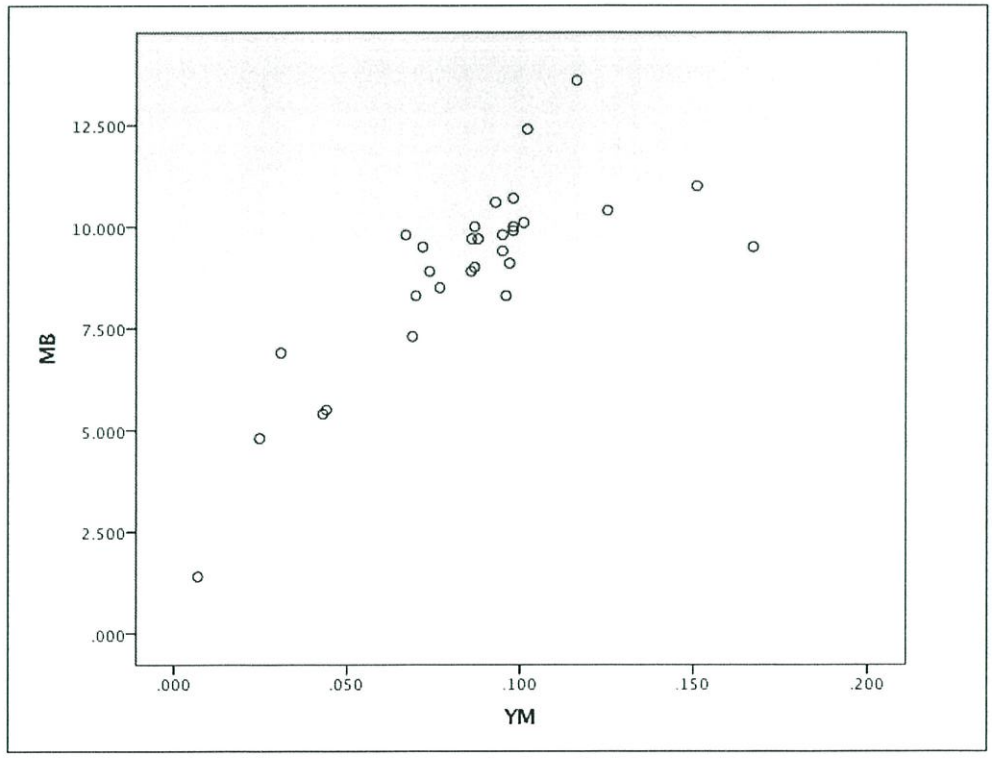
ในการวิเคราะห์ข้อมูลผ่านโปรแกรมสถิติสำเร็จรูป โดยการหาความสัมพันธ์ของตัวแปรสองตัวแปร คือ Total serum bilirubin แทนด้วยตัวแปร MB และ ค่าสีใน CMYK model โดยนำค่า M หักออกจากค่า Y แทนด้วยตัวแปร YM จะได้สหสัมพันธ์และการถดถอยดังนี้

สมมติฐาน H_0 = ตัวแปรอิสระ YM ไม่มีผลต่อ MB

ตารางที่ 4.2 ผลวิเคราะห์สหสัมพันธ์

		MB	YM
MB	Pearson Correlation	1	.786**
	Sig. (2-tailed)		.000
	N	30	30
YM	Pearson Correlation	.786**	1
	Sig. (2-tailed)	.000	
	N	30	30

** . Correlation is Significant at the 0.01 Level (2-tailed).



ภาพที่ 4.1 ภาพแสดงความสัมพันธ์ระหว่างตัวแปร YM และ MB

จากผลวิเคราะห์สหสัมพันธ์ดังตารางที่ 4.2 พบว่าตัวแปร YM และ MB มีค่าสัมประสิทธิ์ของสหสัมพันธ์ (r) ที่ 0.786 ซึ่งสรุปได้ว่า ตัวแปร YM มีความสัมพันธ์เชิงเส้นกับตัวแปร MB

ตารางที่ 4.3 ค่าสถิติที่ใช้พิจารณาความเหมาะสมของสมการถดถอยของตัวแปร

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.787[a]	.619	.575	1.531126

a. Predictors: (Constant), YM, Y, M

จากตารางที่ 4.3 พบว่า ตัวแปรอิสระ YM มีความสัมพันธ์กับตัวแปร MB ในระดับสูงคือ .786 ตัวแปรทั้งสามตัวสามารถอธิบายค่า MB ได้ 6.19% ด้วยความคลาดเคลื่อนของการกะประมาณ 1.531126 (ความผิดพลาดที่เกิดขึ้นจากการพยากรณ์โดยเฉลี่ย)

ตารางที่ 4.4 การตรวจสอบตัวแปรอิสระ ที่สามารถทำนายค่าของตัวแปร MB

Model		Sum of Squares	df	Mean Square	F	P
1	Regression	99.202	3	33.067	14.105	.000[b]
	Residual	60.953	26	2.344		
	Total	160.155	29			

b. Predictors: (Constant), YM, Y, M

จากตารางที่ 4.4 พบว่าตัวแปรอิสระอย่างน้อย 1 ตัวส่งผลต่อค่า MB อย่างมีนัยสำคัญทางสถิติที่ระดับ 0.05

ตารางที่ 4.5 ค่าสัมประสิทธิ์การถดถอยของตัวแปรทั้งสามตัวแปรกับตัวแปร MB

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	2.996	5.313		.564	.578
	YM	58.548	16.125	.833	3.631	.001
	M	6.354	18.326	.073	.347	.732
	Y	-3.789	13.365	-.089	-.284	.779

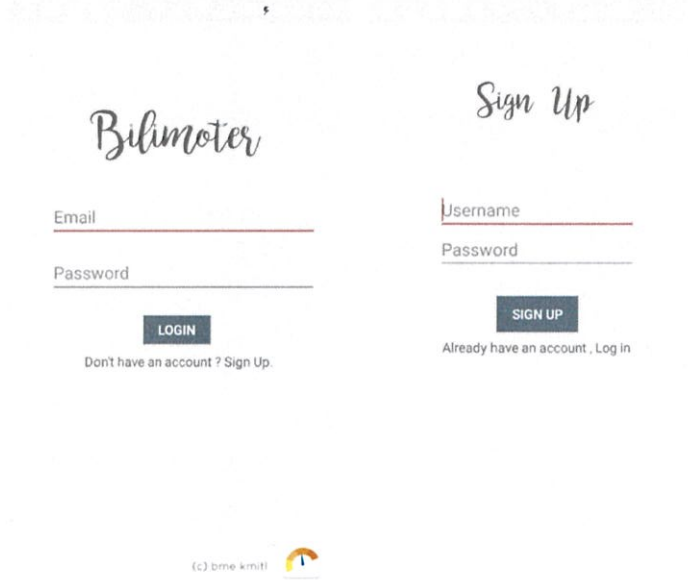
จากตารางที่ 4.5 สามารถเขียนสมการพยากรณ์ตัวแปร MB ได้ดังนี้

$$TcdB = 2.996 + 58.548(Y - M) \qquad (4.1)$$

ค่า B ของตัวแปรอิสระ YM มีค่า 58.548 หมายความว่า ค่าของตัวแปรอิสระ YM เพิ่มขึ้น 1 หน่วย ค่าของตัวแปร MB จะเพิ่มขึ้น 58.548 เมื่อทดสอบความมีนัยสำคัญพบว่า ไม่ยอมรับสมมติฐาน H_0 ($P > 0.05$) แสดงว่าตัวแปรอิสระ YM ส่งผลต่อตัวแปร MB อย่างมีนัยสำคัญทางสถิติ

4.3 หน้าต่างแสดงผลการทำงานของแอปพลิเคชัน

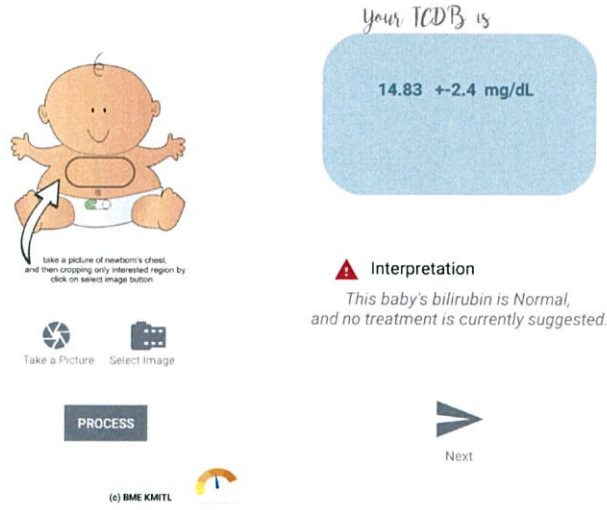
จากการออกแบบการทำงานของแอปพลิเคชัน ในส่วนของหน้าต่าง Login ดังภาพที่ 4.2 จะเป็นส่วนที่ใช้ในการ Login โดยใช้ Email และ Password เพื่อเข้าใช้งาน Application และหลังจากนั้นจะเข้าสู่หน้าต่างที่สามารถกรอกข้อมูลการใช้งานของผู้ใช้งานได้ ดังภาพที่ 4.3



ภาพที่ 4.2 หน้าต่าง Login และ Sign up

ภาพที่ 4.3 หน้าต่างสำหรับกรอกข้อมูลของเด็กแรกเกิด

หลังจากการกรอกข้อมูลส่วนตัวแล้วแอปพลิเคชันจะนำเข้าสู่การถ่ายภาพเพื่อนำประมวลผล ดังภาพที่ 4.4 ซึ่งจะมีฟังก์ชันการเลือกเฉพาะบริเวณหน้าอกของเด็กเท่านั้น แล้วจะทำการประมวณผลค่า Total transcutanoues digital bilirubin level และการแปรผลเบื้องต้น ภายหลังจากทราบค่าแล้วจะสามารถส่งข้อมูลเข้าสู่ฐานข้อมูลออนไลน์ได้ ดังภาพที่ 4.5



ภาพที่ 4.4 หน้าต่างการทดสอบและแสดงผลระดับบิลิรูบิน

ภาพที่ 4.5 หน้าต่างแสดงข้อมูลสำหรับการอัปโหลด

4.4 ผลการทดสอบและการวิเคราะห์ผลจากแอปพลิเคชัน

ทดสอบโดยการถ่ายภาพบริเวณผิวหนังหน้าอกเด็กทารกแรกเกิด จำนวน 30 ราย อายุ 0-7 วัน โดยเปรียบเทียบกับระดับบิลิรูบินในเลือด และนำผลที่ได้ไปวิเคราะห์ทางสถิติ โดยจะนำเสนอข้อมูลที่ได้จากการทดลองตามตารางที่ 4.6

ตารางที่ 4.6 แสดงข้อมูลเด็กทารกแรกเกิดที่ได้รับการทดสอบ

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	ระดับ Microbilirubin จากการทดสอบทาง Clinical Method (mg/dL)	ระดับ Microbilirubin จากการทดสอบจากแอปพลิเคชัน (mg/dL)			
				ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
1	48	2745	8.3	8.134	8.264	9.002	8.467
2	48	2220	9.7	9.954	9.754	9.625	9.778
3	48	2815	10.1	10.007	10.052	10.124	10.061
4	48	2920	9.8	12.177	11.053	9.321	10.850
5	48	2860	9.5	10.358	9.423	9.031	9.604
6	48	2530	11.0	12.422	11.543	11.422	11.795
7	48	3050	9.9	9.144	10.003	9.841	9.663
8	48	2985	10.0	11.359	10.867	9.921	10.715
9	48	2735	9.1	8.412	8.513	9.312	8.745
10	48	4355	9.7	10.292	9.582	9.341	9.738
11	48	2725	10.4	10.649	10.410	10.413	10.490
12	48	3145	9.4	10.567	9.245	9.312	9.708
13	48	3550	6.9	6.403	7.421	6.521	6.781
14	48	2360	5.5	6.412	5.042	5.913	9.045
15	48	2790	5.4	4.585	5.416	5.312	5.104
16	48	3045	8.9	8.775	9.002	8.942	8.906
17	48	2990	8.3	8.451	8.452	8.411	8.438
18	48	3775	9.0	9.555	8.941	9.423	9.306
19	48	3845	8.5	8.419	8.253	8.412	8.361
20	48	2875	6.4	6.412	7.512	6.413	6.779
21	48	2890	10.7	10.721	9.894	9.995	10.203
22	48	2561	4.8	4.234	5.005	4.931	4.723
23	48	2960	9.5	9.354	8.943	9.423	9.24

ตารางที่ 4.6 (ต่อ)

ลำดับที่	อายุ (ชั่วโมง)	น้ำหนัก (กรัม)	ระดับ Microbilirubin จากการทดสอบทาง Clinical Method (mg/dL)	ระดับ Microbilirubin จากการทดสอบจากแอปพลิเคชัน (mg/dL)			
				ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ค่าเฉลี่ย
24	48	3490	9.8	9.227	10.011	10.002	9.747
25	48	2830	7.3	7.324	8.001	8.007	7.777
26	48	3490	10.0	11.307	10.315	10.031	10.551
27	48	3310	8.3	9.238	9.001	9.031	9.090
28	48	3415	10.6	11.612	10.312	10.312	10.7453
29	48	2475	12.4	12.555	11.521	11.841	11.972
30	48	3936	13.6	13.184	13.654	12.987	13.275

4.5 ผลการวิเคราะห์ข้อมูล

ตอนที่ 1 ข้อมูลทั่วไปของผู้ตอบแบบสอบถาม

ในการวิจัยแอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิด กลุ่มตัวอย่างที่วิจัยได้แก่ เด็กทารกแรกเกิด อายุ 0-7 วัน ซึ่งมีข้อมูลทั่วไป คือ อายุ น้ำหนัก โดยจะใช้การวิเคราะห์โดยการแจกแจงค่าความถี่ (Frequency) ค่าร้อยละ (Percentage) ซึ่งจะนำเสนอในรูปแบบตารางปรากฏดังตารางที่ 4.2-4.3

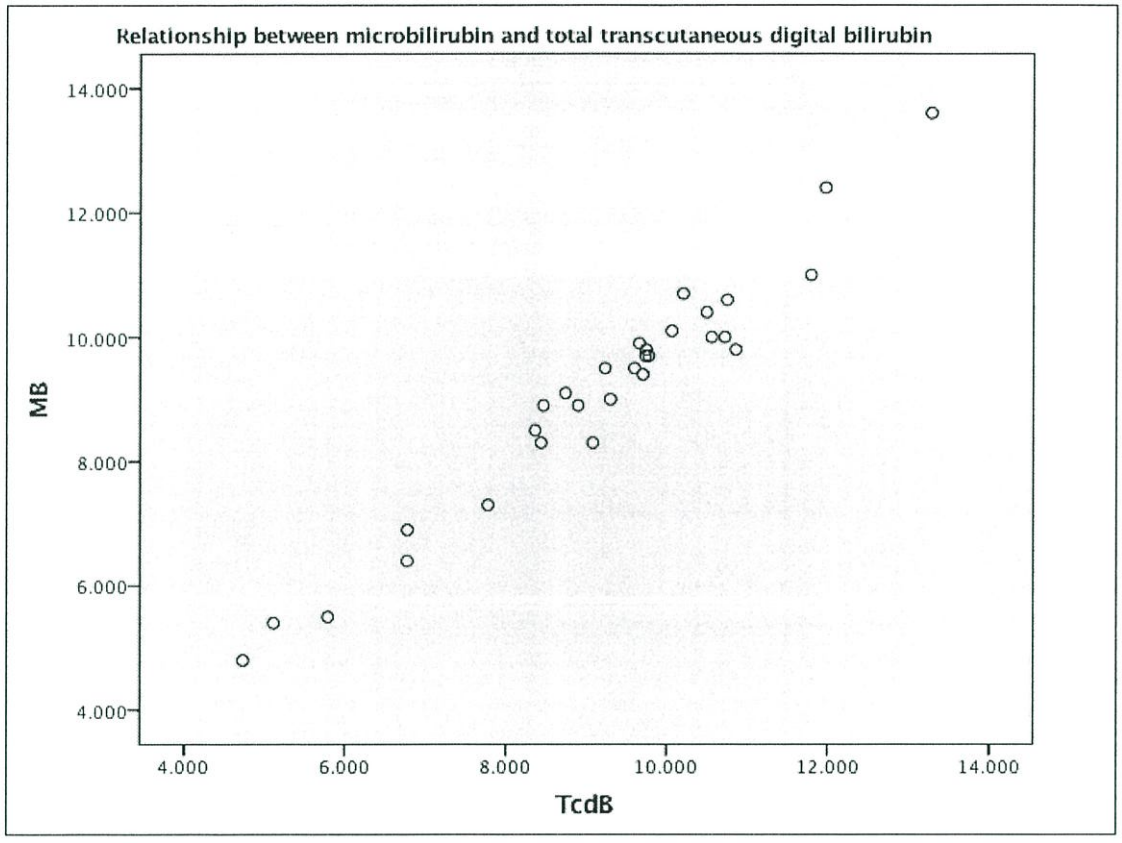
ตารางที่ 4.7 แสดงค่าความถี่ (Frequency) และค่าร้อยละ (Percentage) ข้อมูลทั่วไปของผู้ได้รับการทดลอง จำแนกตามน้ำหนัก

น้ำหนัก	จำนวน	ร้อยละ
น้อยกว่า 1250 กรัม	0	0
1251-1500 กรัม	0	0
1501-2000 กรัม	0	0
มากกว่า 2001 กรัม	30	100

จากตารางที่ 4.7 พบว่า ผู้ได้รับการทดลองที่มีน้ำหนักมากกว่า 2001 กรัม คิดเป็นร้อยละ 100 ของผู้ทดลอง

ตอนที่ 2 ค่าเฉลี่ยและความคลาดเคลื่อนของการทดลอง

ในการทดลองค่าเฉลี่ยจากการทดสอบจากแอปพลิเคชันมีค่า 9.213 mg/dl ค่าเฉลี่ยจากการทดสอบทางห้องปฏิบัติการมีค่า 9.113 mg/dL ค่าความคลาดเคลื่อนของการทดลองคือ1.169% และค่าความคลาดเคลื่อนมาตรฐานของการพยากรณ์คือ 1.35 ซึ่งพบว่ามีนัยสำคัญทางสถิติระหว่างระดับบิลิรูบินจากแอปพลิเคชันและระดับบิลิรูบินจากผลทางห้องปฏิบัติการ ($P<0.05$)



ภาพที่ 4.6 Scatter plot ระหว่างค่าบิลิรูบินที่ได้จากวิธีทางห้องปฏิบัติการกับแอปพลิเคชัน

บทที่ 5
บทสรุป

5.1 สรุป

ภาวะตัวเหลืองเป็นปัญหาที่พบบ่อยที่สุดในทารกแรกเกิด มากกว่า 2 ใน 3 ของทารกแรกเกิดจะมีอาการตัวเหลืองที่สามารถเห็นได้ด้วยตาเปล่า ทารกบางรายอาจมีระดับบิลิรูบินที่สูงมากกว่าปกติ (Pathologic Jaundice) ระดับบิลิรูบินที่สูงมากสามารถทำให้เกิดความผิดปกติของสมองอย่างถาวรได้ การวินิจฉัยภาวะตัวเหลืองจะทำการตรวจเลือดเพื่อวัดระดับบิลิรูบินซึ่งทำให้เกิดความเจ็บปวดและใช้เวลานาน แอปพลิเคชันสำหรับวินิจฉัยภาวะตัวเหลืองในทารกแรกเกิดนี้สามารถเพิ่มความสะดวกและลดเวลาให้กับบุคคลากรทางการแพทย์ในการวินิจฉัยสภาวะตัวเหลืองในเด็กทารกแรกเกิด ซึ่งหลักการทำงานของแอปพลิเคชันคือการนำภาพถ่ายบริเวณอกของเด็กทารกแรกเกิดมาประมวลผลผ่านกระบวนการทางภาพเพื่อที่จะหาระดับ Total transcutaneous digital Bilirubin (TcdB) โดยใช้พิกเซลสีจากภาพที่ถ่ายคำนวณโดยสมการ $TcdB = 2.996 + 58.548(Y - M)$ อีกทั้งยังสามารถส่งข้อมูลของเด็กทารกและระดับ Total transcutaneous digital Bilirubin (TcdB) เข้าสู่ฐานข้อมูลออนไลน์ (Database) ได้

ในการทดลอง ได้ทำการทดสอบแอปพลิเคชันกับเด็กทารกแรกเกิดโดยทำการถ่ายภาพผ่านสมาร์ทโฟนที่บริเวณอกของเด็กทารกจำนวน 30 ราย แล้วนำภาพมาผ่านการประมวลผลผ่านแอปพลิเคชันเพื่อที่จะแสดงผลระดับบิลิรูบินและคำแนะนำในการรักษาเบื้องต้น จากนั้นนำมาเปรียบเทียบกับค่า Microbilirubin (MB) ที่ได้จากการทำการทดสอบทางห้องปฏิบัติการ

จากการทดลองในการถ่ายภาพเด็กทารกแรกเกิดบริเวณอก แล้วนำมาประมวลผลผ่านแอปพลิเคชัน พบว่าค่าเฉลี่ยของผลการทดสอบจากแอปพลิเคชันมีค่า 9.213 mg/dL ค่าเฉลี่ยจากการทดสอบทางห้องปฏิบัติการมีค่า 9.113 mg/dL ค่าความคลาดของการทดลองคือ 1.169% และค่าความคลาดเคลื่อนมาตรฐานของการพยากรณ์คือ 1.35 ซึ่งพบว่ามีนัยสำคัญทางสถิติระหว่างระดับบิลิรูบินจากแอปพลิเคชันและระดับบิลิรูบินจากผลทางห้องปฏิบัติการ ($P<0.05$)

5.2 ข้อเสนอแนะ

แนวทางในการพัฒนาแอปพลิเคชันตรวจสอบสภาวะตัวเหลืองในเด็กทารกแรกเกิดเพื่อให้เกิดประสิทธิภาพที่ดียิ่งขึ้นและให้บุคคลทั่วไปสามารถที่นำไปใช้งานได้ง่าย

1. ควรพัฒนาแอพพลิเคชั่นบนระบบ IOS เพื่อให้ผู้ใช้งานที่ไม่ได้ใช้สมาร์ทโฟนบนระบบ Android สามารถใช้งานได้
2. ระบบฐานข้อมูลออนไลน์ที่ใช้มีพื้นที่จำกัดเนื่องจากฐานข้อมูลที่ใช้ในการทดลองนี้เป็นการจำลองฐานข้อมูลของโรงพยาบาลในเบื้องต้นเท่านั้น
3. การประมวลผลจากค่าพิกเซลสีนั้นยังพบว่ามีความคลาดเคลื่อนที่อาจจะเกิดจากปัจจัยจากสภาวะแวดล้อมภายนอกซึ่งส่งผลต่อการคำนวณระดับบิลิรูบิน
4. ควรเพิ่มจำนวนเด็กทารกแรกเกิดในการทดสอบ เพื่อเพิ่มลดความคลาดเคลื่อนในการวิเคราะห์การถดถอย

บรรณานุกรม

[1] ประสงค์ ธูสรานนท์. 2555. "การซ้อนทับภาพและการจัดเรียงให้ตรงกันของภาพใบหน้าสามมิติขณะแสดงอารมณ์สีหน้าที่แตกต่างกัน."วิทยานิพนธ์วิศวกรรมศาสตร์ดุษฎีบัณฑิต สาขาวิศวกรรม ไฟฟ้า,สถาบันเทคโนโลยีพระจอมเจ้าคุณทหารลาดกระบัง.

[2] ชูชาติ ปิณฑวิรุจน์. 2555. **การประมวลผลภาพดิจิตอลด้วย Matlab.** ครั้งที่พิมพ์ 2. กรุงเทพฯ : มีน เซอร์วิส ซัพพลาย.

[3] Alagille D, Odievre M. 1979. **Liver and biliary tract disease in children.** New York: Wiley.

[4] Engle WD, 2005. **Evaluation of a transcutaneous jaundice meter following hospital discharge in term and near term neonates.** J Perinatol.; 25(7): pp. 49-90

[5] Hansen T,December, 2017. **Neonatal Jaundice.** [Online]. Available : https://emedicine.medscape.com/article/974786-overview#a4

[6] Jirawatee, July, 2016. **รู้จัก Firebase Authentication ตั้งแต่ Zero จนเป็น Hero.** [Online].Available : https://developers.ascendcorp.com/รู้จัก Firebase Authenticationตั้งแต่ Zero จนเป็น Hero

[7] MarcusCode. August, 2016. **ภาษาJava.**[Online]. Available : http://marcuscode.com/lang/java.

[8] Mindphp. January, 2016. **Firebase (ไฟร์เบส) คืออะไร.** [Online]. Available : http://www.mindphp.com/คู่มือ/73-คืออะไร/3921-what-is-firebase-backend.html

[9] Paphatsirinatthi S. May, 2017. **สิ่งที่ควรรู้เกี่ยวกับ Android NDK.** [Online]. Available : https://medium.com/@sirinatoampaphatsirinatthi.

[10] Phonbopit C. April, 2014. **เซฟข้อมูลด้วย SharedPreferences.** [Online]. Available : https://devahoy.com/posts/android-shared-preferences-tutorial/

[11] Pitas, I. 2000 . **DIGITAL IMAGE PROCESSING ALGORITHMS AND APPLICATIONS.** New York : John Wiley and Sons, Inc.

บรรณานุกรม (ต่อ)

[12] PoolsawatBlogs. September, 2016. **Firebase คืออะไร ขั้นตอนการเริ่มใช้งาน พร้อมโค๊ดตัวอย่างแบบง่าย ๆ.** [Online]. Available :https://www.poolsawat.com/firebase-tutorial

[13] Sleeping For Less. July, 2017. **[Android Code] รู้จักและเรียกใช้งาน Camera API v1 บนแอนดรอยด์ แบบง่ายๆ [ตอนที่ 1].** [Online]. Available : http://www.akexorcist.com/2017/07/android-camera-api-v1-part-1.html.

[14] Sleeping For Less. February, 2013 **. [Android Code] การส่งข้อมูลผ่าน Intent ระหว่าง Activity.** [Online]. Available : http://www.akexorcist.com/2013/02/android-code-intent-with-bundle-extra.html.

[15] Sleeping For Less. April, 2014 **. [Android Code] เก็บค่าตัวแปรให้ถาวรแบบง่ายๆได้ด้วย SharedPreferences.** [Online]. Available : http://www.akexorcist.com/2014/09/android-shared-preferences.html

[16] Smola, A.J. and Scholkopf, B. "A tutorial on support vector regression". **Statistics and computing 14**, 3 (2004), pp. 199–222.

[17] Somsak Leartveravat. M.D. 2544. **Transcutaneous bilirubin measurement in full term neonate by digital camera.**

[18] Surapong. February, 2017. **มาลองใช้ OpenCV กับ Android.[Online].** Available : http://ajgo.blogspot.com/2017/02/opencv-android.html.

[19] Tutorialspoint."**Java – Overview.**" [Online]. Available : https://www.tutorialspoint.com/java/java_overview.htm . 2017

[20] Walker HK, Hall WD, Hurst JW, editors. 1990. **Clinical Methods: The History, Physical, and Laboratory Examinations.** 3rd edition. Boston: Butterworths.

[21] Wongsuphasawat K., 2017. **API คืออะไร? อธิบายแบบคนไม่เขียนโปรแกรมรู้เรื่องได้มั้ย?.** [Online].Available : https://medium.com/skooldio/api.

[22] K.N.Plataniotis,A.N.Venetsanopoulos, 2000. **Color Image Processing and Applications.** New York : Springer-Verlag Berlin Heidelberg

ภาคผนวก

MainActivity

```
public class MainActivity extends AppCompatActivity {
    LinearLayout l1;            // เป็นประกาศ Linearlayout โดยมี l1
    Animation uptodown;     // เป็นการประกาศ Animation โดยเป็นชนิดจากบนลงล่าง
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // การกำหนด Content ของ Activityนี้

        l1 =(LinearLayout) findViewById(R.id.l1); //เป็นการประกาศว่า l1 คือ
Linearlayout
        uptodown = AnimationUtils.loadAnimation(this,R.anim.uptodown);
        l1.setAnimation(uptodown);

        final Intent i =new Intent(this,Main2Activity.class); // กำหนด intent ในการสลับ
            Activity
        Thread timer = new Thread(){
            public void run (){
                try {
                    sleep(3000); // กำหนดช่วงเวลาของการแสดงผลในการสลับไปยัง Activity
ต่อไป
                }catch (InterruptedException e) {
                    e.printStackTrace();
                } finally {
                    startActivity(i);
                    finish();
                }
            }
        }; timer.start();
    }
}
```

Activity Login

```
public class LoginActivity extends AppCompatActivity implements
View.OnClickListener {
    FirebaseAuth mAuth;
    EditText editTextEmail,editTextPassword;
    ProgressBar progressBar;
    TextView logo,bme;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        mAuth = FirebaseAuth.getInstance(); //กำหนดกหารใช้ Firebase
        editTextEmail = (EditText) findViewById(R.id.etUsername); //รับค่าตัวแปร
        editTextPassword =(EditText) findViewById(R.id.etPassword); //รับค่าตัวแปร
        progressBar = (ProgressBar) findViewById(R.id.pgb1);
        bme=(TextView) findViewById(R.id.tvBME);
        Typeface myCustomFont1 =
Typeface.createFromAsset(getAssets(),"fonts/Quicksand-Regular.otf");
        bme.setTypeface(myCustomFont1);
        findViewById(R.id.tvsignup).setOnClickListener(this); //กำหนดฟังก์ชั่น Button
        findViewById(R.id.bt).setOnClickListener(this); //กำหนดฟังก์ชั่น Button
    }
// กำหนดฟังก์ชันการทำงานในการตรวจสอบ EditText ในการ Login
private void userLogin(){
        String email = editTextEmail.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();
```

(ต่อ)

```
        if(email.isEmpty()){
            editTextEmail.setError("Email is required");
            editTextEmail.requestFocus();
            return; }
        if (password.isEmpty()){
            editTextPassword.setError("Password is required");
            editTextPassword.requestFocus();
            return;  }
mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(ne
w OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(task.isSuccessful()){
                    Intent intent = new Intent(LoginActivity.this,Main5Activity.class);
                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);
                }else{
Toast.makeText(getApplicationContext(),task.getException().getMessage(),Toast.LEN
GTH_SHORT).show();
                } } });  }
    @Override
    public void onClick(View view) {
        switch (view.getId()){
            case R.id.tvsignup:
                startActivity(new Intent(this,SignupActivity.class));
                break;
            case R.id.bt:
                userLogin();
                break;
        }   } }
```

Activity Sign Up

```
// กำหนดฟังก์ชันการทำงานในหน้า Signup Activity
public class SignupActivity extends AppCompatActivity implements
View.OnClickListener{
    private FirebaseAuth mAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        editTextEmail = (EditText) findViewById(R.id.et3);
        editTextPassword =(EditText) findViewById(R.id.et4);
        progressBar = (ProgressBar) findViewById(R.id.pgb);
        mAuth = FirebaseAuth.getInstance();
        signup=(TextView) findViewById(R.id.tvSign);
        signup.setTypeface(myCustomFont2);
        findViewById(R.id.bt3).setOnClickListener(this);
        findViewById(R.id.tvLogin).setOnClickListener(this);
    }
// กำหนดฟังก์ชันในการ Registration เข้าในระบบ Firebase เพื่อเข้าใช้งานฐานข้อมูลออนไลน์
( Database) และ แอปพลิเคชั่น
    private void registerUser(){
        String email = editTextEmail.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();
        if(email.isEmpty()){
            editTextEmail.setError("Email is required");
            editTextEmail.requestFocus();
            return;
        }
```

(ต่อ)

```
if (password.isEmpty()){
            editTextPassword.setError("Password is required");
            editTextPassword.requestFocus();
            return;  }
mAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListene
r(new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(task.isSuccessful()){
                    Intent intent = new Intent(SignupActivity.this,LoginActivity.class);
                    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                    startActivity(intent);  }
                else {
                    if (task.getException() instanceof
FirebaseAuthUserCollisionException){
                        Toast.makeText(getApplicationContext(),"You are already
registered",Toast.LENGTH_SHORT).show();   }
                    else {
Toast.makeText(getApplicationContext(),task.getException().getMessage(),Toast.LEN
GTH_SHORT).show();
                    } } } }); }
    @Override
    public void onClick(View view) {
        switch (view.getId()){
            case R.id.bt3:
                registerUser();
                break;
            case R.id.tvLogin:
                startActivity(new Intent(this,LoginActivity.class));
                break; }  } }
```

Activity 2

```
public class Main2Activity extends AppCompatActivity {
    Uri uri;            // เป็นการส่งภาพในรูปแบบคลาส Uri โดยไม่ต้องส่งข้อมูลภาพโดยตรง
    Bitmap result;  // ประกาสตัวแปรข้อมูล result ในรูปแบบของ Bitmap
    DecimalFormat num = new DecimalFormat("0.000"); // การกำหนดรูปแบบของเลขทศนิยมโดยมีรูปแบบเป็นทศนิยม 3 ตำแหน่ง
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        // เป็นการกำหนด id ให้กับตัวแปร
        Button btnCamera = (Button) findViewById(R.id.btnCamera);
        Button btncrop = (Button) findViewById(R.id.btncrop);
        imageView = (ImageView) findViewById(R.id.imageView);
        btnCamera.setOnClickListener(new View.OnClickListener() { //กำหนดการทำงานให้ Button เมื่อมีการกดปุ่มจะเริ่มการทำงานตามชุดคำสั่งต่อไปนี้
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE); // เรียกใช้ intent ถ่ายภาพจากกล้องโทรศัพท์
                startActivityForResult(intent, 0);
            }
        });
```

(ต่อ)

```
btncrop.setOnClickListener(new View.OnClickListener() { ////กำหนดการทำงานให้
Button เมื่อมีการกดปุ่มจะเริ่มการทำงานตามชุดคำสั่งต่อไปนี้
        @Override
        public void onClick(View view) {
            Intent CropIntent = new Intent(Intent.ACTION_PICK,
         MediaStore.Images.Media.EXTERNAL_CONTENT_URI); // เป็นการเรียกใช้
งานการดึงภาพจาก Mediastore ในเครื่องเพื่อนำข้อมูลจากภาพมาในรูปแบบ Uri
            CropIntent.setDataAndType(uri, "image/*"); //กำหนด Intent การ crop
ภาพจากข้อมูล uri
            CropIntent.putExtra("crop", "true"); // กำหนดฟังก์ชันการ Crop ภาพ
            CropIntent.putExtra("outputX", 180); // กำหนด output ของแกน X มีขนาด
180
            CropIntent.putExtra("outputY", 240); //กำหนดการ output ของแกน Y มี
ขนาด 240
            CropIntent.putExtra("aspectX", 3); //กำนหนดอัตราส่วน ของแกน X คือ 3
            CropIntent.putExtra("aspectY", 4); ); //กำหนดอัตราส่วน ของแกน X คือ 4
            CropIntent.putExtra("return-data", true);//ส่งข้อมูลเพื่อนำไปใช้
            startActivityForResult(CropIntent, 2); // จะเป็นการเรียกใช้ Intent ที่มีชื่อว่า
CropIntent และส่งค่า 2 ไปยัง intent ต่อไป
        }
    });
  }
```

(ต่อ)

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == 2 && resultCode == RESULT_OK)  //เช็คเงื่อนไขเมื่อค่าที่ส่ง
มา มีค่าเท่ากับ 2 และ สถานะของ Code สามารถทำงานได้ จะทำงานคำสั่งต่อไปนี้
        {
            Bundle extras = data.getExtras(); //รับค่าข้อมูลมาเก็บในตัวแปร extra
            Bitmap bitmap = extras.getParcelable("data");//รับค่าที่สามารถนำมาแก้ไขได้
โดยเก็บไว้ที่ตัวแปร bitmap โดยใช้ชื่อว่า data
            imageView.setImageBitmap(bitmap); // ImageView แสดงภาพที่เป็นข้อมูลจาก
Bitmap ที่ผ่านการ Crop มาแล้ว
            int redColors = 0; // กำหนดตัวแปร redColors เป็นจำนวนเต็มเพื่อเก็บข้อมูลสี
แดงใน pixel จากภาพ โดยมีค่าเริ่มต้นเท่ากับ 0
            int greenColors = 0; // กำหนดตัวแปร greenColors เป็นจำนวนเต็มเพื่อเก็บข้อมูล
สีแดงใน pixel จากภาพ โดยมีค่าเริ่มต้นเท่ากับ 0
            int blueColors = 0; // กำหนดตัวแปร blueColors เป็นจำนวนเต็มเพื่อเก็บข้อมูลสี
แดงใน pixel จากภาพ โดยมีค่าเริ่มต้นเท่ากับ 0
            int pixelCount = 0;// กำหนดตัวแปร pixelCount เป็นจำนวนเต็มเพื่อเก็บข้อมูล
การนับ pixel โดยมีค่าเริ่มต้นเท่ากับ 0
```

(ต่อ)

```
        for (int y = 0; y < bitmap.getHeight(); y++) {  // เมื่อตัวแปรy มีค่าน้อยกว่าความสูงของข้อมูล bitmap และ y จะมีค่าเพิ่มขึ้น 1 เมื่อทำงานในเงื่อนไขแล้ว
        for (int x = 0; x < bitmap.getWidth(); x++) {  // เมื่อตัวแปร x มีค่าน้อยกว่าความกว้างของข้อมูล bitmap และ x จะมีค่าเพิ่มขึ้น 1 เมื่อทำงานในเงื่อนไขแล้ว
        int c = bitmap.getPixel(x, y); // ดึงค่า pixel จากข้อมูล bitmap
            pixelCount++; // ค่า pixelCount มีค่าเพิ่มขึ้น1 ทุกครั้งที่มีการทำงานครบตามเงื่อนไข
            redColors += Color.red(c); //  ตัวแปร redColors มีการเก็บค่าของสีแดงแต่ละ pixel เอาไว้
            greenColors += Color.green(c); // ตัวแปร greenColors มีการเก็บค่าของสีเขียวแต่ละ pixel เอาไว้
            blueColors += Color.blue(c); // ตัวแปร blueColors มีการเก็บค่าของสีน้ำเงินแต่ละ pixel เอาไว้
          } }
        double red = (redColors / pixelCount); // หาค่าเฉลี่ยของสีแดงจากตัวแปร redColors และเก็บค่าเฉลี่ยไว้ในตัวแปร red
        double green = (greenColors / pixelCount); // หาค่าเฉลี่ยของสีเขียวจากตัวแปร greenColors และเก็บค่าเฉลี่ยไว้ในตัวแปร green
        double blue = (blueColors / pixelCount); // หาค่าเฉลี่ยของสีน้ำเงินจากตัวแปร blueColors และเก็บค่าเฉลี่ยไว้ในตัวแปร blue
        double dev = 255;
        double rn = (red / dev); // rn คือ R'
        double gn = (green / dev); //  gn คือ G'
        double bn = (blue / dev); // bn คือ B'
```

(ต่อ)

```
            double mK = max(rn, gn);
            double mK2 = max(mK, bn);
            double K = (1 - mK2);
            final double C = (1 - rn - K) / (1 - K);
            final double M = (1 - gn - K) / (1 - K);
            final double Y = (1 - bn - K) / (1 - K);
// ส่วนที่ใช้ในการส่งข้อมูลจาก Activity นี้ไปยัง Activity ต่อไป
Button btnPass = (Button) findViewById(R.id.btnPass);
            btnPass.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(getApplicationContext(),
Main3Activity.class);
                    intent.putExtra("data", tv_color.getText().toString()); // ส่งข้อมูล data
ที่บรรจุค่า RGB เอาไว้
                    intent.putExtra("ydata",Y); // ส่งข้อมูล Y โดยผ่านตัวแปร ydata
                    intent.putExtra("mdata",M); // ส่งข้อมูล M โดยผ่านตัวแปร mdata
                    startActivity(intent);
                }
            })
        }
    }
}
```

Activity 3

```
public class Main3Activity extends AppCompatActivity {
    Button prc; // กำหนดให้ prc เป็นButton
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main3);
        Bundle bundle = getIntent().getExtras(); // การรับค่าจาก Intent อื่น
        TextView tvRecieved = (TextView) findViewById(R.id.tvRecieved); // กำหนด id
ให้กับ ImageView
        String data = bundle.getString("data"); // เก็บข้อมูลเป็น String จากตัวแปร data
         // กำหนด id ให้กับ textView ที่จะนำมาแสดงค่า RGB
        TextView tvYdata = (TextView) findViewById(R.id.tvYdata) ;
        TextView tvMdata = (TextView) findViewById(R.id.tvMdata) ;
        // รับข้อมูลที่เป็นข้อมูลรูปแบบ double
        double yvalue = bundle.getDouble("ydata");
        double mvalue = bundle.getDouble("mdata");
        tvRecieved.setText(data); // แสดงข้อมูลจากตัวแปร Data ที่เก็บข้อมูลค่า RGB ผ่าน
ทางหน้าจอ
         final double tcdbP = 2.58 + ( 58.711 * (yvalue-mvalue) ); // สมการ
Transcutaneous bilirubin
        // การส่งค่า TcdB เมื่อมีการกดปุ่ม Button
        prc = (Button) findViewById(R.id.prc);
        prc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Main3Activity.this, Main4Activity.class);
                intent.putExtra("tcdbP",tcdbP); // ส่งข้อมูลของ tcdbP ในตัวแปร tcdbP
                startActivity(intent)
            } }); } } } }
```

Activity 4

```
public class Main4Activity extends AppCompatActivity {
    Button btn; //กำหนด btn เป็น Button
    DecimalFormat num = new DecimalFormat ("0.00"); //กำหนดรูปแบบการแสดงผล
ของตัวเลขให้เป็นทศนิยม 2 ตำแหน่ง
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main4);
// ฟังก์ชันในการรับค่าจาก Shared Preferences ที่มาจาก Activity อื่นๆ
        final SharedPreferences sp = getSharedPreferences("P_NAME",
Context.MODE_PRIVATE);
        String wth = sp.getString("WEIGTH","Null"); // กำหนดตัวแปรในการรับค่า
        Double wei = Double.valueOf(wth);
        Bundle bundle = getIntent().getExtras();
// กำหนดการตรวจสอบเงื่อนของค่า Transcutaneous Bilirubin ดังเงื่อนไขต่อไปนี้ และ
แสดงผล
   final double valueP = bundle.getDouble("tcdbP");
        if (wei < 1250) {
            if (valueP > 12){
                TextView tvAbnormal = (TextView) findViewById(R.id.tvAbnormal);
                tvAbnormal.setText("This baby's bilirubin is Abnormal, \n above
the phototherapy threshold ");
            } else {
                TextView jaundice = (TextView) findViewById(R.id.jaundice);
                jaundice.setText("This baby's bilirubin is Normal, \n and no
treatment is currently suggested. ");
            } }
```

(ต่อ)

```
else if (wei > 1251 && wei < 1500){
        if (valueP > 14){
            TextView tvAbnormal = (TextView) findViewById(R.id.tvAbnormal);
            tvAbnormal.setText("This baby's bilirubin is Abnormal, \n above
the phototherapy threshold ")
        } else {
            TextView jaundice = (TextView) findViewById(R.id.jaundice);
            jaundice.setText("This baby's bilirubin is Normal, \n and no
treatment is currently suggested. "); }
    } else if (wei > 1501 && wei < 2000){
        if (valueP > 16){
            TextView tvAbnormal = (TextView) findViewById(R.id.tvAbnormal);
            tvAbnormal.setText("This baby's bilirubin is Abnormal, \n above
the phototherapy threshold ");
        } else {
            TextView jaundice = (TextView) findViewById(R.id.jaundice);
            jaundice.setText("This baby's bilirubin is Normal, \n and no
treatment is currently suggested. ");   }
    } else if (wei > 2001 ) {
        if (valueP > 20){
            TextView tvAbnormal = (TextView) findViewById(R.id.tvAbnormal);
            tvAbnormal.setText("This baby's bilirubin is Abnormal, \n above
the phototherapy threshold ");
```

ต่อ

```
        } else {
            TextView jaundice = (TextView) findViewById(R.id.jaundice);
            jaundice.setText("This baby's bilirubin is Normal, \n and no
treatment is currently suggested. "); }
    } else if (wei > 2001 ) {
        if (valueP > 20){
            TextView tvAbnormal = (TextView) findViewById(R.id.tvAbnormal);
            tvAbnormal.setText("This baby's bilirubin is Abnormal, \n above
the phototherapy threshold ");
        } else {
            TextView jaundice = (TextView) findViewById(R.id.jaundice);
            jaundice.setText("This baby's bilirubin is Normal, \n and no
treatment is currently suggested. ")
        }
    }
final String TcdBvalue = Double.toString(valueP);
    btn = (Button) findViewById(R.id.nxt);
    btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Main4Activity.this, Main6Activity.class);
            vt = getSharedPreferences("P_NAME", Context.MODE_PRIVATE);
            editor = vt.edit();
            editor.putString("TCDB",TcdBvalue);
            editor.commit();
            startActivity(intent);
        }
    }); } }
```

Activity 5

```
    public class Main5Activity extends AppCompatActivity {
    EditText name,brtd,wth; // กำหนดตัวแปร name,brtd,wth ให้เป็นตัวแปรของ
EditText
    Button btn; // กำหนดตัวแปร btn ให้เป็นตัวแปรของ Button
   // กำหนดการับค่า SharedPreferneces
    SharedPreferences sp;
    SharedPreferences.Editor editor;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main5);
        //กำหนด id ให้กับ EditText ของแต่ละตัว
        name = (EditText) findViewById(R.id.name);
        gest = (EditText) findViewById(R.id.gest);
        wth  =  (EditText) findViewById(R.id.wth);
        date = (EditText) findViewById(R.id.date);
        hospital = (EditText) findViewById(R.id.hospital);
        // กำหนดให้ Button ทำงานเมื่อมีการกดปุ่ม Button
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
             // กำหนดให้ตัวแปรของ edittext แต่ละตัวรับค่าจาก edittext แล้วเก็บไว้ในตัวแปร
แต่ละตัว
                String namevalue = name.getText().toString();
                String datevalue = gest.getText().toString();
                String wthvalue  = wth.getText().toString();
                String agevalue = date.getText().toString();
                String hospitalvalue = hospital.getText().toString();
                Intent intent = new Intent(Main5Activity.this, Main2Activity.class);
```

ต่อ

```
//กำหนดให้ตัวแปรแต่ละตัวและส่งค่าใน SharedPreference
sp = getSharedPreferences("P_NAME", Context.MODE_PRIVATE);
            editor = sp.edit();
            editor.putString("NAME",namevalue);
            editor.putString("GEST",datevalue);
            editor.putString("WEIGTH",wthvalue);
            editor.putString("DATE",datevalue);
            editor.putString("HOSPITAL",hospitalvalue);
            editor.commit();
            startActivity(intent);
        } }); } }
```

Activity 6

```
public class Main6Activity extends AppCompatActivity {
TextView tv,tv1,tv2,tv3,tv4,tv5;// กำหนดตัวแปรของ TextView
    private Button send, back;
    private Firebase mRootRef;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main6);
 //กำหนดที่อยู่ URL ของ Database ที่จะนำข้อมูลไปเก็บไว้
        mRootRef = new Firebase("https://bilimoter1.firebaseio.com/Users");
        send = (Button) findViewById(R.id.btFb);
        back = (Button) findViewById(R.id.btReset);
        back.setOnClickListener(new View.OnClickListener() {
```

(ต่อ)

```
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(Main6Activity.this, Main5Activity.class);
            startActivity(intent); } });
    SharedPreferences sp = getSharedPreferences("P_NAME",
Context.MODE_PRIVATE);
    SharedPreferences vt = getSharedPreferences("P_NAME",
Context.MODE_PRIVATE);
    final String name = sp.getString("NAME","0");
    final String gest = sp.getString("GEST","0");
    final String wth = sp.getString("WEIGTH","0");
    final String date = sp.getString("DATE","0");
    final String hospital = sp.getString("HOSPITAL","0");
    final String tcdb = vt.getString("TCDB","0");
    tv = (TextView) findViewById(R.id.textView3);
    tv1 = (TextView) findViewById(R.id.textView4);
    tv2 = (TextView) findViewById(R.id.textView5);
    tv3 = (TextView) findViewById(R.id.textView17);
    tv4 = (TextView) findViewById(R.id.textView18);
//แสดงค่าของแต่ละตัวแปรผ่านทาง TextView
    tv.setText(name);
    tv1.setText(gest);
    tv2.setText(wth);
    tv3.setText(date);
    tv4.setText(hospital);
```

(ต่อ)

```
//กำหนดฟังก์ชันในการส่งข้อมูลเข้าสู่ฐานข้อมูลออนไลน์ (Database)
send.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            progress = new ProgressDialog(Main6Activity.this);
            progress.setMax(100);
            progress.setMessage("Data is Uploading . . .");
            progress.setTitle("Upload Data");
            progress.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            progress.show();
            new Thread(new Runnable() {
                @Override
                public void run() {
                    try {
                        while (progress.getProgress() <= progress.getMax()){
                            Thread.sleep(50);
                            handle.sendMessage(handle.obtainMessage());
                            if (progress.getProgress() == progress.getMax())
                            {
                                progress.dismiss(); } }
                    }catch (Exception e){
                        e.printStackTrace();
                    } }  }).start();
        Firebase childRef = mRootRef.child(name);
        childRef.child("Birthday").setValue(gest);
        childRef.child("Weight").setValue(wth);
        childRef.child("AGE").setValue(date);
        childRef.child("Hospital").setValue(hospital);
        childRef.child("TcdB").setValue(tcdb);
```

(ต่อ)

```
        }
        Handler handle = new Handler(){
            @Override
            public void handleMessage (Message msg){
                super.handleMessage(msg);
                progress.incrementProgressBy(1);
            }};
    }); }}
```

Senddata Activity

```
public class Sendata extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Firebase.setAndroidContext(this);
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.mochutima.BiliMoTer">
    <uses-permission android:name="android.permission.CAMERA"></uses-
permission>
    <uses permission
android:name="android.permission.READ_EXTERNAL_STORAGE"></uses-
permission>
//เป็นการขอการเข้าถึงการอ่านไฟล์ ใน Storage
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
//เป็นการขอการเข้าถึงการเขียนไฟล์ ใน Storage
    <uses-permission
android:name="android.permission.MEDIA_CONTENT_CONTROL"></uses-
permission>
//เป็นการขอการเข้าถึง Media Content
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
```

(ต่อ)

```
<activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".Main2Activity" />
        <activity android:name=".Main3Activity" />
        <activity android:name=".Main4Activity" />
        <activity android:name=".Main5Activity" />
        <activity android:name=".Main6Activity" />
        <activity android:name=".LoginActivity" />
        <activity android:name=".SignupActivity"></activity>
    </application>
</manifest>
```

Build Gradle (Module:app)

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 26
    buildToolsVersion "26.0.2"
    defaultConfig {
        applicationId "com.example.mochutima.BiliMoTer"
        minSdkVersion 16
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
}
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.pro' } }
    sourceSets {
        main
                {   jni.srcDirs = ['src/main/jni', 'src/main/jniLibs/']
                    jni.srcDirs = [] } } }
dependencies {
    implementation 'com.google.firebase:firebase-database:11.8.0'
    implementation 'com.google.firebase:firebase-auth:11.8.0'
    compile 'com.google.firebase:firebase-core:11.8.0'
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
```

(ต่อ)

```
compile project(':openCVLibrary249')
    compile 'com.android.support:appcompat-v7:26.+'
    compile 'com.android.support.constraint:constraint-layout:1.0.1'
    compile 'com.android.support.constraint:constraint-layout-solver:1.0.1'
    compile 'com.android.volley:volley:1.0.0'
    compile 'com.firebase:firebase-client-android:2.5.2'
    testCompile 'junit:junit:4.12'
    implementation 'com.android.support.constraint:constraint-layout:1.0.1'
}
apply plugin: 'com.google.gms.google-services'
```

Matlab (Simple Color Balance)

```
%% import image
clc; clear all; close all;
im_orig = cbimread('/Users/MoChutima/Desktop/Project/DATA
resize/30_3936_(2).jpg');
figure
imshow(im_orig)
title('Original Image')

% full width histogram method
satLevel = .01; %percentage of the image to saturate to black or white, tweakable
param
q = [satLevel/2 1-satLevel/2];
plots=1;
imRGB_orig = cbreshape(im_orig)*255;
imRGB = zeros(size(imRGB_orig));
N = size(imRGB_orig,2);
color = {'r','g','b'};
for ch = 1:3
    if plots
        figure
        subplot(211)
        hist(imRGB_orig(ch,:),256)
        set(findobj(gca,'Type','patch'),'FaceColor',color{ch},'EdgeColor',color{ch})
        xlim([0 255])
        title('Original Histogram')
    end
```

(ต่อ)

```
tiles = quantile(imRGB_orig(ch,:),q);
    imRGB(ch,:) = cbsaturate(imRGB_orig(ch,:),tiles); %saturate at the appropriate
pts. in distribution
    bottom = min(imRGB(ch,:)); top = max(imRGB(ch,:));
    imRGB(ch,:) = (imRGB(ch,:)-bottom)*255/(top-bottom);

    if plots
        subplot(212)
        hist(imRGB(ch,:),256)
        set(findobj(gca,'Type','patch'),'FaceColor',color{ch},'EdgeColor',color{ch})
        xlim([0 255])
        title('Corrected Histogram')
    end
end
figure
imshow(cbunshape(imRGB,size(im_orig))/255)
title('Simplest Color Balance Corrected')
```

Matlab (Color CCT)

```
%% Color CCT
clc; clear all; close all;
f=imread('/Users/MoChutima/Desktop/WORK1:2560/Project/Pictures/cct.jpg');


R=f(:,:,1);
G=f(:,:,2);
B=f(:,:,3);
ANS=[R(1500,1500) G(1500,1500) B(1500,1500)];


X = (-0.14282)*ANS(:,1) + (1.54924)*ANS(:,2) + (-0.95641)*ANS(:,3);
Y = (-0.32466)*ANS(:,1) + (1.57837)*ANS(:,2) + (-0.73191)*ANS(:,3);
Z = (-0.68202)*ANS(:,1) + (0.77073)*ANS(:,2) + (0.56332)*ANS(:,3);


X=double(X);
Y=double(Y);
Z=double(Z);
lux = Y;
x=X/(X+Y+Z);
y=Y/(X+Y+Z);


n = (x - 0.3320) / (0.1858 - y);
CCT=449*(n^3) + 3525*(n^2) + 6823.3*n + 5520.33;
```

Matlab Function (cbreshape)

```
function out = cbreshape(im)
%out = cbreshape(im)
% Takes a width x height x 3 RGB image and returns a matrix where each column
is an RGB

s = size(im);
m = s(1); n = s(2);
out = reshape(permute(im,[3 1 2]),[3 m*n 1]);
 end
```

Matlab Function (cbsaturate)

```
function y = cbsaturate(x,bounds)
%y = cbsaturate(x,bounds)
% Saturates the vector x given the bounds pair [low high]

low = bounds(1); high = bounds(2);
y=zeros(size(x));
for i = 1:length(x)
    if x(i) < low
        y(i) = low;
    elseif x(i) > high
        y(i) = high;
    else
        y(i) = x(i);
    end
end
end
```

Matlab Function (cbunshape)

```
function out = cbunshape(mat,s)
%out = cbunshape(im,[width,height])
% Takes a 3xn matrix of RGB pixels and returns a height x width x 3 RGB
% image

width = s(1); height = s(2),
out = reshape(mat,[3,width,height]);
out = permute(out,[2 3 1]);
end
```