# การตรวจจับภาพนู้ดด้วยวิธีการวิเคราะห์สีผิว

**Nudity Detection Using Skin Color Analysis Method**

นางสาวทิพนภา จันทร์หา รหัส 45380044

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 ม.ค. 2553
เลขทะเบียน 1500 9443
เลขเรียกหนังสือ ปร. ว4570 2548
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2548

# ใบรับรองโครงงานวิศวกรรม

| | |
|---|---|
| **หัวข้อโครงงาน** | การตรวจจับภาพนู้ดด้วยวิธีการวิเคราะห์สีผิว |
| **ผู้ดำเนินโครงงาน** | นางสาวทิพนภา จันทร์หา รหัส 45380044 |
| **อาจารย์ที่ปรึกษา** | ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2548 |

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

.....กรรมการ

( อาจารย์ศิริพร เดชะศิลารักษ์ )

.....กรรมการ

( ดร.ไพศาล มุณีสว่าง)

| | |
|---|---|
| หัวข้อโครงงาน | การตรวจจับภาพนู้ดด้วยวิธีการวิเคราะห์สีผิว |
| ผู้ดำเนินโครงงาน | นางสาวทิพนภา จันทร์หา รหัส 45380044 |
| อาจารย์ที่ปรึกษา | ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| ภาควิชา | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| ปีการศึกษา | 2548 |


## บทคัดย่อ

โครงงานนี้เกิดขึ้นเนื่องจาก ผู้จัดทำเล็งเห็นถึงผลเสียที่จะเกิดขึ้นกับเยาวชน ในการนำเสนอภาพนู้ดผ่านสื่อต่าง ๆ ที่เพิ่มมากขึ้น จึงต้องการศึกษาและพัฒนาโปรแกรมตรวจจับภาพนู้ดด้วยการประมวลผลภาพ (Image Processing) โดยวิธีการวิเคราะห์สีผิว (Skin Color Analysis Method) เพื่อวิเคราะห์ว่าภาพนั้นเป็นภาพนู้ดหรือไม่ การพัฒนานี้ทดสอบและใช้งานบนระบบปฏิบัติการวินโดวส์ XP ใช้ Visual Studio.NET 2005 ในการพัฒนา โดยโปรแกรมนี้จะสามารถทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคลหรือโน๊ตบุ๊คเท่านั้น

ผลที่ได้จากการทำโครงงานนี้คือ ได้โปรแกรมที่สามารถตรวจสอบและแยกแยะระหว่างพื้นผิวของวัตถุอื่นกับผิวของมนุษย์ เพื่อใช้ในการหาปริมาณพื้นที่สีผิวของมนุษย์ที่แท้จริง คุณภาพของการวิเคราะห์นั้นสามารถตรวจจับสีผิวได้ทั้งสีผิวขาว (ยุโรป), สีผิวเหลือง (เอเชีย) และสีผิวดำ (แอฟริกา) และสามารถวิเคราะห์ได้ว่าภาพนั้นเป็นภาพนู้ดหรือไม่

| | |
|---|---|
| **Project Title** | Nudity Detection Using Skin Color Analysis Method |
| **Name** | Miss. Tipnapa Chanha ID. 45380044 |
| **Project Advisor** | Assistant Professor Suchart Yammen, PhD |
| **Major** | Computer Engineering |
| **Department** | Electrical and Computer Engineering |
| **Academic Year** | 2005 |


## ABSTRACT

This project is motivated by the problems of nudity media which are happening to a number of juveniles. The nudity presents through many types of the media such as images and videos on internet and television. As such, we develop a software application for nudity detection from digital image. We employ the skin color analysis method to implement this application.

Nudity detection by skin color analysis is applied to differentiate the human skin color from the background image in order to obtain the area of skin. It can detect pale skin, yellowish skin, as well as dark skin. The percentage of the image pixels of the skin area is used for classification of the nudity in the input images. The software is the window application running on Microsoft Windows XP operating system and Visual Studio.Net 2005, and can be operated on a personal computer as well as notebook.

# กิตติกรรมประกาศ

โครงงานฉบับนี้สำเร็จลงได้ด้วยความอนุเคราะห์อันดียิ่งจากท่าน ดร.ไพศาล มุณีเสว่าง ที่กรุณาเป็นอาจารย์ที่ปรึกษาที่ได้ให้แนวคิดช่วยเหลือในการทำโครงงาน และขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น ให้ความเอาใจใส่ ตลอดจนสละเวลาอันแสนมีค่าเพื่อตรวจแก้ไขข้อบกพร่องต่างๆ จนสำเร็จลุล่วงไปด้วยดี อีกทั้งยังคอยอบรมสั่งสอนผู้จัดทำให้เป็นคนดีมีระเบียบวินัยผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ทั้งนี้ขอกราบขอบพระคุณอาจารย์ทุกท่าน ที่ได้อบรมสั่งสอน ให้ความรู้และคำแนะนำที่ดีเสมอมา

ท้ายนี้ขอกราบขอบพระคุณบิดา-มารดา ที่ให้กำลังใจและมอบสิ่งที่ดีในชีวิต ทั้งความอบอุ่นและคำปรึกษาที่มีค่ายิ่งเหนือสิ่งอื่นใด และขอบคุณเพื่อนๆ ทุกคนที่คอยเป็นกำลังใจให้ตลอดมาจนสามารถทำงานครั้งนี้ได้สำเร็จลุล่วงไปด้วยดี

นางสาวทิพนภา จันทร์หา

# สารบัญ

หน้า


บทคัดย่อภาษาไทย.....ก

บทคัดย่อภาษาอังกฤษ.....ข

กิตติกรรมประกาศ.....ค

สารบัญ.....ง

สารบัญตาราง.....ฉ

สารบัญรูป.....ช

บทที่ 1 บทนำ

1.1 วัตถุประสงค์.....1

1.2 ขอบข่ายของโครงงาน.....1

1.3 กิจกรรมการดำเนินงาน.....3

1.4 ผลที่คาดว่าจะได้รับ.....4

1.5 งบประมาณในการดำเนินโครงการ.....4

บทที่ 2 หลักการและทฤษฎี

2.1 ภาพดิจิตอล.....5

2.2 การประมวลผลภาพดิจิตอล (Digital Image Processing).....5

2.3 พื้นฐานทั่วไปที่เกี่ยวกับภาพ.....6

2.3.1 รูปภาพแบบเวกเตอร์ (Vector Graphic).....6

2.3.2 รูปภาพแบบบิตแมบ (Bitmap Image).....6

2.3.3 ความละเอียดของภาพ.....6

2.3.4 พิกเซลและดอท (Pixel and Dot).....6

2.3.5 ระบบ RGB.....6

2.3.6 ระบบสี HSV (Hue/Saturation/Intensity).....7

2.4 แบบจำลองสีผิวของมนุษย์ (Human Skin Color Model).....9

2.4.1 การตรวจสอบเกิดความผิดพลาดน้อย.....9

2.4.2 ตรวจจับจากความแตกต่างของชนิดของสีผิว.....9

2.4.3 การสัมผัสของความคลุมเครือระหว่างสีของผิวหนังและส่วนที่ไม่ใช่ผิวหนัง.....9

# สารบัญ(ต่อ)

หน้า

2.4.4 การทนต่อความแปรปรวนของสภาวะแสง ..... 9
2.5 การแยกข้อมูลภาพ (Image Segmentation) ..... 10
2.5.1 วิธีการแยกส่วนของภาพโดยอาศัยความเข้ม ..... 11
2.6 การอธิบายถึงภาพต่าง ๆ โดยใช้ฮีสโตรแกรมสี (Color Histogram) ..... 12

บทที่ 3 การดำเนินงาน
3.1 การหาค่าฮีสโตรแกรมของแม่แบบสีผิว ..... 14
3.2 การหาจำนวนพิกเซลของสีผิว ..... 18

บทที่ 4 ผลการทดลอง
4.1 การทดลองการตรวจจับสีผิว ..... 19

บทที่ 5 สรุปและวิเคราห์ผล
5.1 สรุปผลการทดลองที่ได้จากการพัฒนาโปรแกรม ..... 26
5.2 ข้อเสนอแนะ ..... 27
5.3 ปัญหาที่พบในการพัฒนาโครงงานวิจัย ..... 27
5.4 แนวทางในการพัฒนาโปรแกรม ..... 27

เอกสารอ้างอิง ..... 28

ภาคผนวก

ภาคผนวก ก ..... 29

ประวัติผู้เขียนโครงงาน ..... 42

# สารบัญตาราง

ตารางที่ หน้า

4.1 แสดงตัวอย่างผลการทดลอง.....22

# สารบัญรูป


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | แสดงการแทนค่าของแต่ละจุดสีของภาพระดับเทา 256 ระดับ | 5 |
| 2.2 | โหมดสีแบบ RGB | 7 |
| 2.3 | วงล้อของสี (Color Wheel) | 7 |
| 2.4 | แสดงโหมดสี HSV | 8 |
| 2.5 | แสดงตัวอย่างช่วงความเข้มของสี(Saturation) | 8 |
| 2.6 | แสดงตัวอย่างช่วงความสว่างของสี(Value) | 9 |
| 2.7 | ไบโมดัลอิมเมจฮีสโตแกรม (Bimodal image histogram) | 11 |
| 3.1 | แสดงการทำงานของโปรแกรม | 13 |
| 3.2 | แสดงรูปแบบจำลองของแม่แบบสี HSV | 14 |
| 3.3 | แสดงแบบจำลองของHue | 15 |
| 3.4 | แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวที่หลากหลาย | 16 |
| 3.5 | แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวขาว (ยุโรป) | 16 |
| 3.6 | แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวเหลือง (เอเชีย) | 17 |
| 3.7 | แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวดำ (แอฟริกา) | 17 |
| 4.1 | แสดงหน้าโปรแกรมก่อนการเรียกภาพ | 19 |
| 4.2 | การเรียกข้อมูลภาพจากเครื่องคอมพิวเตอร์ | 20 |
| 4.3 | แสดงภาพหลังจากเรียกข้อมูลภาพจากคอมพิวเตอร์ | 20 |
| 4.4 | แสดงผลการตรวจสอบว่าภาพไม่เป็นภาพนู้ด | 21 |
| 4.5 | แสดงผลการตรวจสอบว่าภาพเป็นภาพนู้ด | 22 |

# บทที่ 1
# บทนำ

คงไม่สามารถปฏิเสธได้ว่าอินเตอร์เนต (Internet) ได้เข้ามามีบทบาทในการดำรงชีวิตอย่างมาก ในปัจจุบัน ซึ่งกล่าวได้ว่าอินเตอร์เนตเป็นแหล่งความรู้และความบันเทิงขนาดใหญ่ซึ่งนับได้ว่าเป็นประโยชน์อย่างมาก แต่ในขณะเดียวกันก็มีการนำเสนอสื่อที่ไม่เหมาะสม เช่น ภาพนู้ด ภาพโป๊เปลือย ซึ่งส่งผลเสียต่อเยาวชนและสังคมอย่างมากดังนั้นหากเราสามารถสร้างเครื่องมือที่ตรวจสอบภาพนู้ดได้ก็จะสามารถพัฒนาเพื่อป้องกันไม่ให้ภาพเหล่านั้นออกมาเผยแพร่ทางอินเตอร์เนตและส่ง ผลเสียต่อเยาวชนได้ด้วยเหตุนี้ผู้จัดทำโครงงานจึงต้องการสร้างซอฟท์แวร์ที่สามารถตรวจจับภาพ นู้ดได้โดยอัตโนมัติ โดยการใช้ขนาดพื้นที่สีผิวเป็นเกณฑ์ ซึ่งจะถูกประมวลผลด้วยหลักการของการประมวลผลภาพด้วยสี (Color Image Processing) โดยการหาขอบเขตของภาพที่ต้องการและวิเคราะห์ลักษณะสีภายในเพื่อทำการหาจำนวนพิกเซลของสีผิวเทียบกับพื้นที่ภายในขอบเขตที่ต้อง การและทำการประมวลผลออกเป็นเปอร์เซ็นต์เพื่อวิเคราะห์ว่ารูปภาพนี้ควรจะเป็นภาพนู้ดหรือไม่ ในส่วนของการที่ติดต่อกับผู้ใช้ (Graphic User Interface) จะสร้างโดยโปรแกรม C#.NET เพื่อให้ผู้ใช้สามารถใช้งานได้ง่ายขึ้นโดยในโครงงานนี้จะมีการสาธิตการตรวจสอบภาพว่าภาพนั้นควรเป็น ภาพนู้ดหรือไมและมีความจะเป็นร้อยละเท่าไร ซึ่งผู้จัดทำโครงงานหวังว่าโครงงานนี้จะเป็นแนว ทางเพื่อก้าวไปสู่การพัฒนาเป็นซอฟท์แวร์ตรวจจับภาพนู้ดและเว็บไซต์ที่ไม่เหมาะสมได้

## 1.1 วัตถุประสงค์

1. เพื่อศึกษาถึงระบบการประมวลผลภาพด้วยสี (Color Image Processing) เพื่อใช้ในการตรวจสอบและแยกแยะระหว่างพื้นผิวของวัตถุอื่นกับผิวของมนุษย์เพื่อใช้ในการหาปริมาณพื้นที่สีผิวของมนุษย์ที่แท้จริง
2. สามารถประยุกต์ใช้หลักการประมวลผลภาพด้วยสี ตรวจจับภาพและประมวลผลเป็นเปอร์เซ็นต์เพื่อทำการวิเคราะห์ว่าเป็นภาพนู้ดหรือไม่
3. เพื่อพัฒนาซอฟท์แวร์ในการตรวจจับภาพนู้ดให้ได้ผลเป็นที่น่าพอใจ

## 1.2 ขอบข่ายของโครงงาน

1. ใช้วิธีการวิเคราะห์โดยการวิเคราะห์สีผิว (Skin Color Analysis Method)
2. วิเคราะห์ภาพโดยสามารถหาขอบเขตที่ต้องการ เพื่อหาขนาดในระนาบ 2 มิติ
3. วิเคราะห์ภาพโดยการหาฮีสโตรแกรมภาพเพื่อเปรียบเทียบช่วงค่าสีผิวกับส่วนที่ไม่ใช่สีผิว
4. วิเคราะห์ภาพโดยหาปริมาณพื้นที่สีผิว เพื่อบอกเปอร์เซ็นต์ส่วนของสีที่ไม่มีวัตถุหรือเสื้อผ้าปกคลุม
5. ใช้โปรแกรม Visual C#.NET ในการพัฒนาโปรแกรม
6. ซอฟท์แวร์ที่พัฒนาขึ้นมาใหม่นั้นจะสามารถตรวจจับภาพผู้ดได้ดีในระดับหนึ่ง ซึ่งซอฟท์แวร์ที่ได้นี้จะสามารถทำงานได้ที่เครื่องคอมพิวเตอร์ส่วนบุคคล (PC)

มหาวิทยาลัยนเรศวร

## 1.3 กิจกรรมการดำเนินงาน

| กิจกรรม | ปี 2548 | | | | | | | | | | | | ปี 2549 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ม.ค. | ก.พ. | มี.ค. | เม.ย. | พ.ค. | มิ.ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. |
| ศึกษาทฤษฎีทางด้านการประมวลผลภาพ (Image Processing) | ← | — | — | → | | | | | | | | | | | | |
| ศึกษาโปรแกรม Visual C#.NET | | | | ← | — | — | → | | | | | | | | | |
| ศึกษาอัลกอลิทึมในการวิเคราะห์ภาพแบบต่างๆ | | | | | | | ← | — | — | — | → | | | | | |
| ประยุกต์และพัฒนาแนวคิดที่จะนำมาใช้วิเคราะห์ภาพ | | | | | | | | | | ← | — | — | → | | | |
| ทดสอบอัลกอลิทึมกับภาพนู้ดและเพิ่มฟังก์ชันการทำงาน | | | | | | | | | | | | ← | — | — | → | |
| ทดสอบใช้งานจริง | | | | | | | | | | | | | ← | — | — | → |
| จัดทำรายงาน | | | | | | | | | | | | | | | ← | → |

## 1.4 ผลที่คาดว่าจะได้รับ

1. ได้รับความรู้เกี่ยวกับกระบวนการวิเคราะห์ภาพด้วยสี (Color Image Processing)
2. สามารถหาจำนวนพิกเซลของสีผิวเพื่อวิเคราะห์หาเปอร์เซ็นต์ความเป็นไปได้ที่ภาพนั้นจะเป็นภาพนู้ด
3. ได้รับความรู้ในการใช้โปรแกรม Visual C#.NET ในส่วนของการสร้างแอพพลิเคชั่น (Application)
4. ซอฟท์แวร์ที่ได้สามารถตรวจจับภาพนู้ดได้ดีในระดับหนึ่ง
5. เป็นแนวทางในการพัฒนาซอฟท์แวร์ในระดับที่สูงขึ้น

## 1.5 งบประมาณในการดำเนินโครงการ

| | | |
|---|---|---|
| 1. ค่าถ่ายเอกสารประกอบการทำโครงการ | 300 | บาท |
| 2. ค่ากระดาษในการพิมพ์งาน | 100 | บาท |
| 3. ค่าหมึกพิมพ์ | 500 | บาท |
| 4. ค่าจัดทำรูปเล่ม | 200 | บาท |
| รวมเป็นเงินทั้งสิ้น | 1000 | บาท (หนึ่งพันบาทถ้วน) |

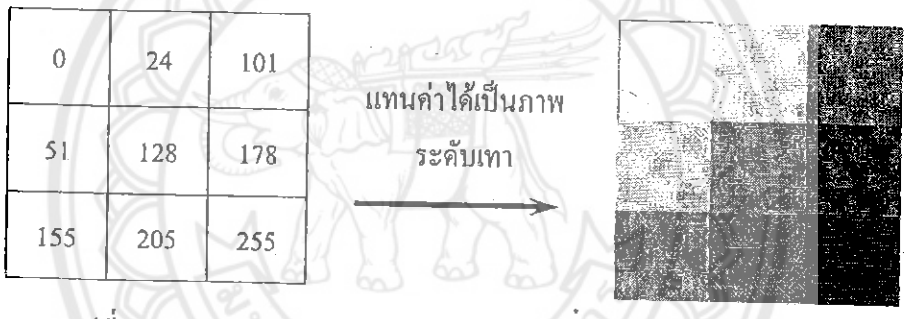
มหาวิทยาลัยนเรศวร

# บทที่ 2
# หลักการและทฤษฎี

ก่อนการดำเนินโครงงานนั้น เราควรศึกษาทฤษฎีที่เกี่ยวข้อง เพื่อเป็นการเพิ่มความรู้ความเข้าใจในโครงงานมากยิ่งขึ้น ซึ่งจะทำให้การดำเนินการง่ายและสะดวก

## 2.1 ภาพดิจิตอล

ภาพดิจิตอลคือการแสดงภาพรูปแบบของเมตริกซ์ X แถวและ Y หลักโดยที่แต่ละช่องของเมตริกซ์เรียกว่าจุดภาพ (Pixel) จะมีค่าสีของจุดภาพนั้นอยู่ โดยเมื่อนำจุดภาพที่มีค่าสีต่าง ๆกันจำนวนหนึ่งมาเรียงกันกว้าง X และ Y ก็สามารถแสดงภาพดิจิตอลได้




รูปที่ 2.1 แสดงการแทนค่าของแต่ละจุดสีของภาพระดับเทา 256 ระดับ
(กำหนดให้ 0 คือสีขาว 255 คือสีดำ)

## 2.2 การประมวลผลภาพดิจิตอล (Digital Image Processing)

การประมวลผลภาพดิจิตอล (Digital Image Processing) คือการนำภาพดิจิตอลที่เป็นลักษณะของเมตริกซ์มากระทำการใด ๆ ซึ่งอาจเป็นการกระทำเบื้องต้นทางคณิตศาสตร์ เช่น บวก, ลบ, คูณหรือหาร และอาจจะมีการกระทำเบื้องต้นทางตรรกศาสตร์ เช่น AND, OR, XOR เป็นต้น

## 2.3 พื้นฐานทั่วไปที่เกี่ยวกับภาพ

รูปภาพสามารถแบ่งตามวิธีการจัดเก็บได้เป็น 2 ชนิดใหญ่ ๆ ดังต่อไปนี้

### 2.3.1 รูปภาพแบบเวกเตอร์ (Vector Graphic)

รูปภาพชนิดนี้ถูกสร้างขึ้นจากสมการเชิงเส้นต่าง ๆ ไม่ว่าจะเป็นเส้นโค้ง เส้นตรง เมื่อเราย่อขยายรูปภาพเหล่านั้น คอมพิวเตอร์จะทำการคำนวณให้ภาพใหม่คมชัดอยู่เสมอ ดังนั้นรูปภาพชนิดนี้จะไม่ขึ้นอยู่กับความละเอียดของภาพ ตัวอย่างของภาพแบบนี้จะเห็นได้ชัด คือรูปภาพที่ได้จากโปรแกรม Adobe IIlustrator , CorelDRAW , MacroMedia Freehand เป็นต้น

### 2.3.2 รูปภาพแบบบิตแมบ (Bitmap Image)

ภาพชนิดนี้เกิดจากจุดเล็กๆ ประกอบกันจนเห็นเป็นภาพขึ้นมา คุณภาพของรูปจึงขึ้นอยู่กับความละเอียดของภาพ หากภาพมีความละเอียดมาก ก็จะมีความคมชัดมากขึ้น หากทำการขยายภาพชนิดนี้คอมพิวเตอร์จะทำการขยายภาพขึ้นด้วยความละเอียดที่มีอยู่ ทำให้ภาพที่ได้มาลักษณะหยาบ ตัวอย่างของรูปภาพแบบนี้คือ รูปภาพที่สร้างขึ้นจากโปรแกรม Adobe Photoshop , CorelPHOTO Paint เป็นต้น

### 2.3.3 ความละเอียดของภาพ

ความละเอียดของภาพเป็นสิ่งที่บอกถึงคุณภาพของภาพเหล่านั้น หน่วยที่เรานิยมใช้บอกถึงความความละเอียดของภาพ คือ พิกเซลต่อนิ้ว (Pixel/Inch) ค่านี้บอกให้ทราบว่าภาพนี้มีจำนวนพิกเซลในหนึ่งตารางนิ้วและสามารถคำนวณหาจำนวนจุดทั้งหมดของภาพได้อีกด้วยดังตัวอย่าง เช่น ภาพขนาด 1*1 ที่มีความละเอียดเท่ากับ 8 พิกเซลต่อนิ้ว ภาพนี้มีพิกเซลทั้งหมดเท่ากับ 64 พิกเซล

### 2.3.4 พิกเซลและด็อท (Pixel and Dot)

พิกเซล คือ จุดที่เล็กที่สุดของภาพ พิกเซลหนึ่งสามารถแสดงได้หลายสี ส่วนด็อทจะเป็นจุดที่ใช้กระบวนการพิมพ์ภาพ การสร้างพิกเซลหนึ่งพิกเซลจะต้องใช้ด็อทเพื่อทำให้เกิดความเข้มและสีต่าง ๆ กัน หน่วยด็อทต่อนิ้ว (dpi) จะใช้บอกความละเอียดของเครื่องพิมพ์ ส่วนหน่วยพิกเซลต่อนิ้ว (ppi) จะใช้บอกความละเอียดของเครื่องสแกนและจอภาพ

### 2.3.5 ระบบ RGB

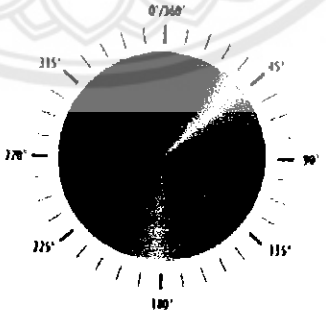
ระบบสี RGB เป็นโหมดที่ใช้แชนแนลสีจำนวน 3 สี คือ แดง เขียว น้ำเงิน โดยแต่ละสีจะมีการไล่ลำดับสีได้ถึง 256 ระดับ เมื่อรวมกันทั้ง 3 สีจะสามารถแสดงได้สูงถึง 16.7 ล้านสี (2 ยกกำลัง 24 บิต)โหมดสีนี้เหมาะสำหรับการตกแต่งสีเพราะสามารถแทนสีได้มาก และยังเป็นโหมดเดียวกับที่ใช้ในโมนิเตอร์ด้วย ซึ่งผู้ใช้คอมพิวเตอร์ส่วนใหญ่คุ้นเคย เป็นระบบที่ใช้ในการกำหนดค่าเปอร์เซ็นต์ของสีแดง เขียว และน้ำเงิน เบื้องต้น ที่นำมารวมเข้าด้วยกัน เพื่อสร้างสีที่ต้องการ ซึ่งระบบการกำหนดค่าสีแบบนี้เป็นระบบที่สามารถกำหนดสีที่สามารถถูกสร้างออกมาได้อย่างชัดเจนที่สุด และเป็นระบบการแทนค่าที่ดีสำหรับการใช้งานในระบบฮาร์ดแวร์แต่เป็นการยากในการรับรู้และกำหนดค่าด้วยสัญชาติญาณของมนุษย์เพื่อจะสร้างสีที่ต้องการขึ้น ตัวอย่างเช่นต้องการใช้สีแดง เขียว

และน้ำเงินเท่าไรในการสร้างสีส้มซึ่งจะเห็นได้ว่าเป็นการยากที่มนุษย์จะสามารถกำหนดค่าขึ้นมาด้วยตัวเอง ระบบ RGB เป็นระบบที่แยกสัญญานขององค์ประกอบของสีออกจากกันโดยสิ้นเชิงซึ่งเป็นระบบการกำหนดค่าที่ใช้ในระบบการแสดงภาพทั้งในระบบจอภาพคอมพิวเตอร์และโทรทัศน์ สีเพียงแต่สัญญาณของสีที่ใช้ในการแสดงออกมานั้นจะมีการเข้ารหัสเป็นความถี่คลื่นในมาตรฐานที่ต่างกันซึ่งกระบวนการในการเปลี่ยนแปลงค่าสีเป็นความถี่และจากความถี่กลับมาเป็นค่าสีจะทำให้เกินความผิดเพี้ยนของสี ไปได้

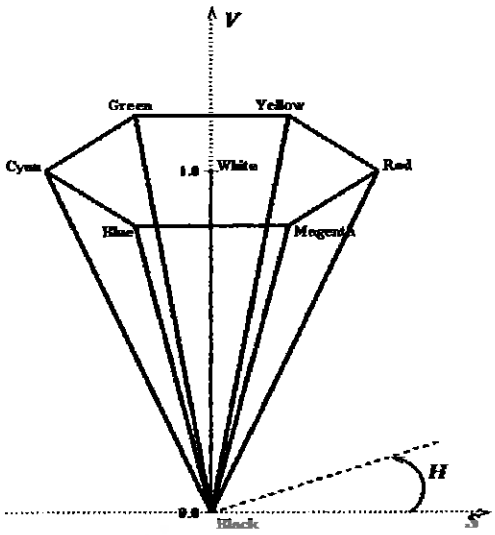
รูปที่ 2.2 โหมดสีแบบ RGB

### 2.3.6 ระบบสี HSV (Hue/Saturation/Intensity)

ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue, Saturation และ Value ซึ่ง Hue คือค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดงและเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อย ๆ สีก็จะเปลี่ยนแปลงไปตามสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้งซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา ซึ่งจะถูกอ้างอิงโดยมุมจากวงล้อของสี (Color Wheel)



รูปที่ 2.3 วงล้อของสี (Color Wheel)




**รูปที่ 2.4** แสดงโหมดสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี (ค่าสี) มีค่าเป็นไปตามสีที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเองตัวอย่างเช่น จอภาพขาว-ดำถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

S - ค่าความเข้มของสี (Saturation) ในความเข้มระดับต่ำตั้งแต่ 0 – 20 % จะทำให้ผลลัพธ์สีที่ออกมาเป็นสีเทา ความเข้มระดับกลาง 40 – 60 % ผลลัพธ์จะเป็นสีอ่อน และความสูง 80 – 100 % ผลลัพธ์ได้สีที่จัดหรือสดใส

**รูปที่ 2.5** แสดงตัวอย่างความช่วงเข้มของสี(Saturation)

V – ความสว่างของสี (Value) ซึ่งมีช่วงตั้งแต่ 0 % (มืดหรือดำ) จนถึง 100 % (สว่างหรือขาว) หรือสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกัน

**รูปที่ 2.6** แสดงตัวอย่างช่วงความสว่างของสี(Value)

โดยปกติในซอฟท์แวร์หรือเครื่องมือช่วยในการสร้างสีนั้นมักมีการใช้ทั้งระบบ RGB และ HSV ร่วมกัน เราอาจจะเห็นการเรียกชื่อระบบลักษณะ HIS ร่วมกัน โดยเราจะเห็นการเรียกชื่อระบบลักษณะ HIS ในชื่ออื่น เช่น Hue/Saturation/Intensity(HSI) , Luminance/Hue/Saturation(LHS หรือ HSL)

## 2.4 แบบจำลองสีผิวของมนุษย์ (Human Skin Color Model)

การใช้โมเดลสีผิวของมนุษย์นั้นเป็นตัวที่จะใช้ในการตัดสินใจว่าสีนั้น เป็นสีผิวของมนุษย์หรือไม่ใช่สีผิว สิ่งที่จำเป็นของโมเดลสีผิวมีดังนี้

### 2.4.1 การตรวจสอบเกิดความผิดพลาดน้อย

การตรวจสอบสีผิวเป็นขั้นตอนแรกในการแบ่งส่วนสีผิวซึ่งจำเป็นมากในการตรวจจับสีผิว ดังนั้นหากเลี่ยงไม่ได้ว่าสีผิวทั้งหมดต้องมีการตรวจสอบขณะที่ความสอดคล้องอัตราการตรวจจับที่ต่ำค่าความผิดพลาดจะสามารถถูกจัดการได้เมื่อมีความเข้าใจในสิ่งที่สนใจมากกว่าที่มีให้ตอนแรกที่หามาได้

### 2.4.2 ตรวจจับจากความแตกต่างของชนิดของสีผิว

กล่าวคือ ชนิดของสีผิวมีความหลากหลายทั้งผิวขาว ผิวเหลืองถึงคล้ำและผิวสีน้ำตาล จะต้องแบ่งออกให้ได้ว่าเป็นผิวแบบไหนจากทั้งหมด

### 2.4.3 การสัมผัสของความคลุมเครือระหว่างสีของผิวหนังและส่วนที่ไม่ใช่ผิวหนัง

การมีสีของสิ่งแวดล้อมที่หลากหลายและคล้ายกับสีผิวหนังที่สนใจนั้น ทำให้ไม่สามารถตรวจจับผิวหนังได้เสมอ ถ้าไม่มีการชี้เฉพาะส่วนของพื้นที่หรือขอบเขตของสีผิวหรือส่วนที่ไม่ใช่สีผิว จะทำให้เกิดความคลุมเครือระหว่างสีผิวและไม่ใช่สีผิวเกิดขึ้น

### 2.4.4 การทนต่อความแปรปรวนของสภาวะแสง

สีผิวจะแตกต่างกันอย่างเห็นได้ชัดภายใต้สภาวะแสงที่ต่างกันการสร้างโมเดลสีผิวจะทำงานไม่ได้หากอยู่ภายใต้สภาวะความแปรปรวนของแสง อย่างไรก็ตามโมเดลสีผิวที่ดีจะต้องทนต่อสภาวะความแปรปรวนของแสงได้ โมเดลสีผิวของมนุษย์นั้นต้องการวิธีในการแบ่งส่วนของสีผิวและพื้นที่ของสีแต่ละสีว่ามีสีไหนบ้างจากที่ให้มาทั้งหมด

วิธีการแบ่งส่วนเป็นการรวบรวมความเข้าใจหลาย ๆ อย่างเข้าด้วยกัน เป็นแนวทางในการจัดการ ในการตัดสินใจแบ่งขอบเขตและแบ่งส่วนการประมาณค่าพื้นฐานของความหนาแน่น ซึ่งตัวเลือกประเภทพื้นที่ของสีที่ต่างกันได้ เช่น RGB , YCbCr , HSV เป็นต้น

ขั้นตอนหลักในการแบ่งส่วนพิกเซลระหว่างส่วนที่เป็นสีผิวและส่วนที่ไม่ใช่สีผิว คือการคำนวณ P(skin/cs) โดยใช้กฎของเบย์ (Bayes rule)

**กฎของเบย์ (Bayes rule)**

$$P(skin/cs) = \frac{P(cs/skin)P(skin)}{P(cs/skin)P(skin) + P(cs/nskin)P(nskin)}$$

โดยให้ cs เป็นค่าของสีผิว

เมื่อ P (skin/cs) $\geq 0$

P (skin) คือ ค่าความน่าจะเป็นของสีผิว

P (nskin) คือ ค่าความน่าจะเป็นของส่วนที่ไม่ใช่สีผิว

เนื่องจาก P (skin) + P(nskin) = 1 เราจะพบว่า 1 เป็นค่าเริ่มต้น การเลือกค่าความเป็นไปได้ใน 1 ตามอัตราที่เหมาะสม ของผลรวมในส่วนที่เป็นพิกเซลของผิวในฮิสโตแกรมจากผลรวมของจำนวนพิกเซลทั้งหมด

$$P(skin) = \frac{Ts}{Ts + Tn}$$

เมื่อ Ts เป็นค่าผลรวมของจำนวนพิกเซลสีผิว

Tn เป็นค่าผลรวมของจำนวนพิกเซลส่วนที่ไม่ใช่สีผิว

## 2.5 การแยกข้อมูลภาพ (Image Segmentation)

การแยกข้อมูลภาพเป็นการแยกภาพออกเป็นส่วนๆซึ่งจะทำให้สามารถเลือกส่วนที่ต้องการได้วิธีการพื้นฐานสำหรับการแยกข้อมูลภาพออกเป็นส่วน ๆ คือการพิจารณาภาพแอมปริจูด (Image amplitude) ได้แก่การพิจารณาความสว่างของภาพสำหรับภาพระดับขาวเทาและลักษณะของพื้นผิว ก็เป็นองค์ประกอบหนึ่งที่สามารถทำการเซคเมนเตชัน (Segmentation) ได้สะดวกยิ่งขึ้น วิธีการแยกข้อมูลภาพ วิธีนี้คือ

**2.5.1 วิธีการแยกส่วนของภาพโดยอาศัยความเข้ม (Amplitude segmentation mathoding)**

สำหรับการแยกข้อมูลภาพออกเป็นส่วน ๆ ในหัวข้อนี้จะพิจารณาความเข้มของจุดภาพต่างๆ ภายในภาพ ซึ่งผลหารแยกข้อมูลภาพจะขึ้นอยู่กับวิธีการหาค่าขีดแบ่งของส่วนประกอบที่เป็นความเข้มหรือสีของภาพซึ่งมีอยู่หลายวิธีด้วยกันดังนี้คือ

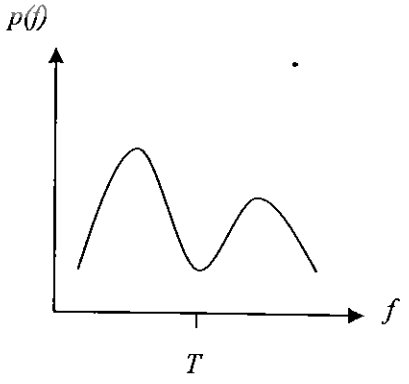
**2.5.1.1 ค่าขีดแบ่งความเข้มสว่างระดับสอง (Bilevel Luminance Thresholding)**

ใช้สำหรับภาพบางชนิดจะมีลักษณะของวัตถุที่เราสนใจซึ่งความเข้มคงที่เมื่อเปรียบเทียบกับพื้นหลังตัวอย่างได้แก่ ภาพของอักษร (Text) เป็นต้น ภาพเหล่านี้จะมีความเข้มของวัตถุที่เราสามารถแยกออกจากพื้นหลังได้อย่างชัดเจน (มีความเข้มสองระดับได้แก่ความเข้มของวัตถุและความเข้มของพื้นหลัง) การแยกข้อมูลภาพออกเป็นส่วน ๆ สามารถทำได้โดยกำหนดค่าขีดแบ่งซึ่งเป็นค่าความเข้ม ให้มีที่ไม่สามารถแยกความแตกต่างของวัตถุและพื้นหลังได้อย่างชัดเจน เช่น ภาพของตัวอักษรที่มีความเข้มของตัวอักษรเป็น 0 (สีดำ) และมีความเข้มของพื้นหลังเป็น 255 (สีขาว) ดังนั้นค่าความขัดแย้งจึงควรจะมีค่าเท่ากับ 128 เพื่อที่จะให้สามารถแยกวัตถุออกจากพื้นหลังได้ โดยปกติแล้วการเลือกค่าขีดบางจะขึ้นอยู่กับฮีสโตแกรมที่จุดต่ำสุดระหว่างจุดยอด (peaks)

$$\text{เมื่อ } g(x,y) = \begin{cases} 1 \text{ เมื่อ } (x,y) > T \\ 0 \text{ เมื่อ } (x,y) \text{เป็นค่าอื่น} \end{cases}$$

เมื่อ g ( x,y) เป็นข้อมูลภาพ ณ ตำแหน่ง x , y

T เป็นค่าขีดแบ่ง




**รูปที่ 2.7** ไบโมดัลอิมเมจฮีสโตแกรม (Bimodal image histogram)

**2.5.1.2 ค่าขีดแบ่งความสว่างหลายระดับ (Multilevel Luminance Thresholding)**

สำหรับภาพที่จะประกอบด้วยหลาย ๆ วัตถุสามารถทำการแยกข้อมูลภาพออกเป็นส่วน ๆ ได้โดยใช้ค่าขีดแบ่งหลาย ๆ ค่าสำหรับภาพที่ N วัตถุโดยที่แต่ละวัตถุจะมีช่วงกว้างของความเข้มเท่ากับ Ri (กำหนดได้ด้วยค่าขีดแบ่ง 2 ค่าคือ $T_{i-1}, T_i$) สามารถทำการแยกได้ดังนี้

$$g(x,y) = R, \text{ เมื่อ } T_{i-1} \leq f(x,y) \leq T_i , \qquad i = 1,2,3\ldots\ldots,N$$

ค่าขีดแบ่งสามารถหาได้จากฮีสโตรแกรมภาพ แต่ในบางครั้งการเปลี่ยนแปลงฮีสโตรแกรมไม่สามารถบอกการเปลี่ยนแปลงของวัตถุได้อย่างชัดเจนนัก ซึ่งวิธีที่ง่ายที่สุดในการที่จะทำ ฮีสโตรแกรมให้สามารถหาค่าขีดแบ่งได้ง่ายขึ้นนั้น คือ การหาขอบของภาพ (Edge Detection) เพื่อพิจารณาจุดภาพต่าง ๆ ของภาพว่าเป็นขอบเขตของวัตถุหรือไม่

## 2.6 การอธิบายถึงภาพต่าง ๆ โดยใช้ฮีสโตรแกรมของสี (Color Histogram)

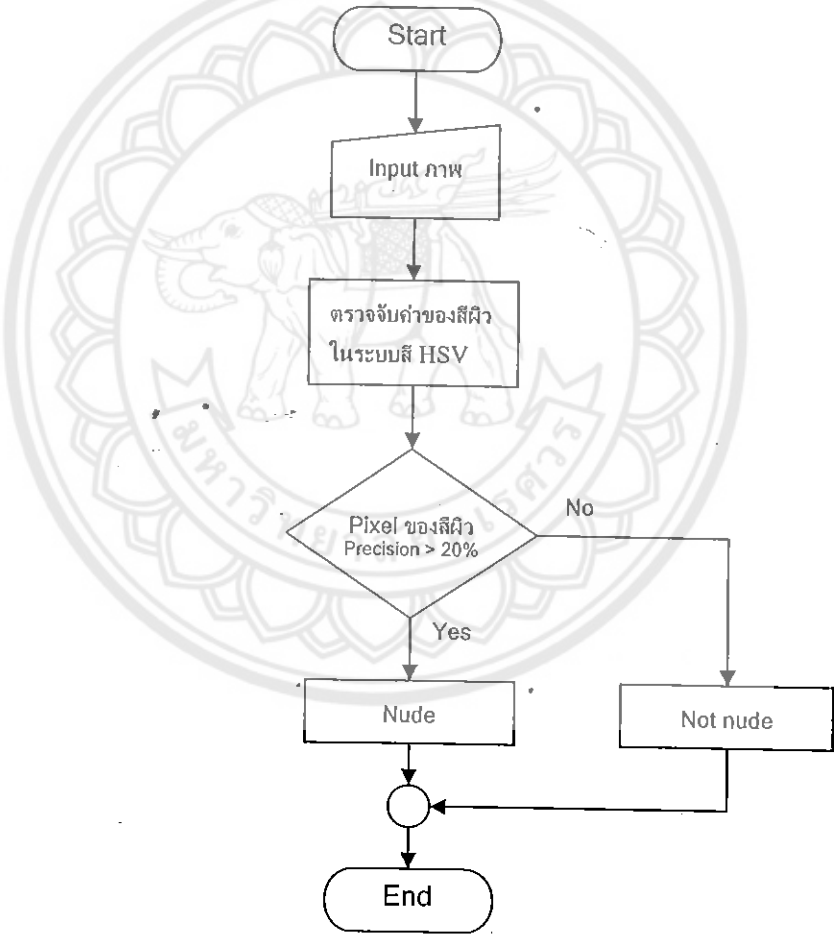
ภาพโดยทั่วไปนั้นจะประกอบขึ้นด้วยจุดของเม็ดสี (Pixel) ต่าง ๆ เป็นจำนวนมากมารวมกันจนเกิดภาพ โดยที่แต่ละจุดเม็ดสีนั้นจะอธิบายว่าเป็นสีอะไร เช่นในระบบสี RGB ก็จะอธิบายว่าแต่ละจุดสีมีค่าของสีแดง, เขียวและน้ำเงินเป็นเท่าไร ในการทำฮีสโตรแกรมนั้นสามารถทำได้โดยเริ่มจากการสร้างกราฟที่มีจำนวนช่วงเท่ากับจำนวนของช่องที่ต้องการกำหนดค่าลำดับชั้นของสีที่ต้องการจากค่าสูงสุดของแต่ละช่วงการแบ่งเป็นช่วง ๆ ตามจำนวนของลำดับชั้นที่ต้องการ จากนั้นจึงนำจุดของเม็ดทุกจุดในภาพเปรียบเทียบว่าอยู่ในช่วงใดของกราฟ เมื่อมีจุดของเม็ดสีตกอยู่ในช่วงใดก็จะทำการเพิ่มค่าในช่วงดังกล่าวเป็นจำนวนหนึ่งค่าโดยทำตั้งแต่จุดของเม็ดสีแรกของภาพไปจนถึงจุดของเม็ดสีสุดท้ายของภาพ โดยทั่วไปแล้วการทำฮีสโตรแกรมนั้นจะนำค่าที่ได้ในแต่ละช่วงมาหาอัตราส่วนระหว่างค่าของจำนวนสีในช่วง ๆ นั้นกับจำนวนจุดของเม็ดสีหนึ่งๆ ซึ่งจะทำให้ทราบว่ามีช่วงสีใดมากในภาพดังกล่าว

ในการทำฮีสโตรแกรม ของภาพในระบบสี HSV นั้น จะเป็นการแสดงถึงค่าองค์ประกอบของแต่ละสีในรูปของระบบสี HSV ถ้าภาพมีสีที่อยู่ในช่วงนั้นก็จะได้กราฟที่มีความสูงมาก ค่าของฮีสโตรแกรมในช่วงของระบบสี HSV จะมีจำนวนช่วงสีที่ต้องการคือ ค่าผลคูณระหว่างค่าของจำนวนช่วงในแกน Hue , Saturation และ Value ซึ่งจะได้ช่วงจำนวนสีทั้งหมดที่ต้องการ จากนั้นนำค่าของสีที่อยู่ในระบบของ HSV มาเปรียบเทียบว่าอยู่ในช่วงใดโดยการหาค่าของความกว้างของช่วงในแต่ละแกนต่อไป

# บทที่ 3
# การดำเนินงาน

จากที่ได้กล่าวมาข้างต้น จะพบปัญหาของการตรวจจับภาพผู้ดเนื่องจากข้อจำกัดของความแตกต่างของสีผิว ดังนั้นจึงต้องมีการหาค่าฮีสโตรแกรมของผิวในแต่ละแบบ เช่นค่าฮีสโตรแกรมของผิว ขาว (ยุโรป) , ผิวเหลือง (เอเชีย) และผิวดำ (แอฟฟริกา) ซึ่งจะช่วยให้การตรวจจับและวิเคราะห์ภาพง่ายยิ่งขึ้น

ขั้นตอนการทำงานของโปรแกรมสามารถสรุปได้ดังนี้




**รูปที่ 3.1** แสดงการทำงานของโปรแกรม

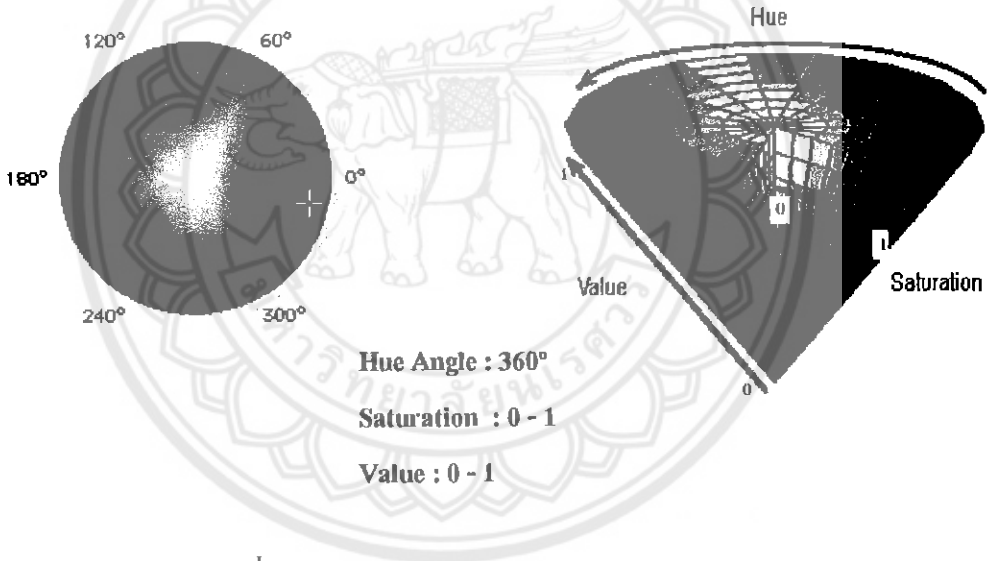
เริ่มต้นโดยการป้อนภาพนั้นเนื่องจากข้อจำกัดของโปรแกรมที่สามารถทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) ดังนั้นภาพที่สามารถนำมากระทำการวิเคราะห์นั้นจะต้องอยู่บนเครื่อง

คอมพิวเตอร์ส่วนบุคคล และเป็นภาพถ่ายที่เป็นสีเท่านั้นจากนั้นอัลกอลิทึม จะทำการวิเคราะห์ค่าของสีผิวในระบบสี HSV เปรียบเทียบกับช่วงค่าของสีผิวที่กำหนดไว้ในแต่ละช่วงของสีผิว หากภาพที่นำมาวิเคราะห์ตรวจพบพื้นที่เป็นสีผิวของมนุษย์มากกว่าร้อยละ 20 ของจำนวนพิกเซลทั้งหมดจะถือว่าภาพนั้นเป็นภาพนู้ด

## 3.1 การหาค่าฮีสโตรแกรมของโมเดลสีผิวในระบบ HSV

เริ่มแรกคือการหาโมเดลสีผิวของแต่ละประเภทสีผิวในระดับที่หลากหลายและนำมาหาค่าฮีสโตรกรม และหาช่วงค่าของฮีสโตรแกรม ออกมา จากนั้นจะนำภาพที่ได้จากที่กล่าวมาแล้วในข้างต้นมาหาค่าฮีสโตรแกรม โดยจะใช้ โมเดลสีแบบ HSV ซึ่งโมเดลแบบ HSV นั้นเป็นโมเดลที่มีความใกล้เคียงกับการมองเห็นของมนุษย์มาก โดยโมเดลสีแบบ HSV จะประกอบไปด้วย

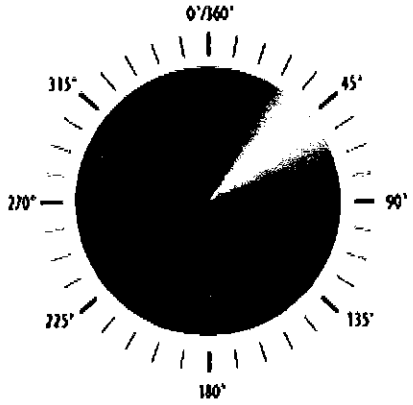
โดยจะใช้ โมเดลสีแบบ HSV ซึ่งโมเดลแบบ HSV นั้นเป็นโมเดลที่มีความใกล้เคียงกับการมองเห็นของมนุษย์มาก โดยโมเดลสีแบบ HSV จะประกอบไปด้วย




**รูปที่ 3.2** แสดงรูปแบบจำลองของโมเดล (Model) สีแบบ HSV

**Hue** คือวงกลมที่ปากของกรวยโดยจะเป็นส่วนของสีเทียบกับมุมองศาของรูปวงกลม ที่ 0° จะเป็นแดง (Red) เมื่อมุมเปลี่ยนองศาไปเรื่อย ๆ จนมีค่ามากขึ้น สีก็จะเปลี่ยนไปดังรูปที่ 3.3

ดังนั้นจะเห็นได้ว่า Hue จะมีค่ามุมองศาของรูปวงกลม คือตั้งแต่ 0° - 360°




**รูปที่ 3.3** แสดงแบบจำลองของ Hue

**Saturation** คือส่วนของพื้นที่วงกลมหรือมองในทางปฏิบัติก็คือ การแบ่งพื้นที่ภายในของวงกลมเป็นวงกลมที่ซ้อนทับกัน โดยจะแบ่งให้อยู่ในช่วงตั้งแต่ 0 – 250 โดยเมื่อรวมกับ Hue แล้วถ้า Saturation = 0 (จุดศูนย์กลางของวงกลม) จะไม่มี Hue อยู่ซึ่งนั่นจะเป็นสีขาว แต่ในทางกลับกันถ้า Saturation = 1 จะไม่มีแสงสีขาวอยู่เลย

**Value** ถ้าพิจารณาจากรูปกรวยก็จะหมายถึงแกนกลางของรูปกรวยนั้นเอง Value จะบอกถึงความสว่างที่มีอยู่ในช่วง 0 – 1

โดยค่าของ Hue (H) , Saturation(S) และ Value(V) สามารถคำนวณได้จากค่าโมเดลสี แบบ RGB ได้ดังนี้

**การหาค่า Hue** สามารถคำนวณได้ดังนี้

$$Red_h = red - \min(red, green, blue)$$
$$Green_h = green - \min(red, green, blue)$$
$$Blue_h = bule - \min(red, greed, blue)$$

$$Hue = \frac{(240 x blue_h) + (120 x green_h)}{blue_h + green_h}$$

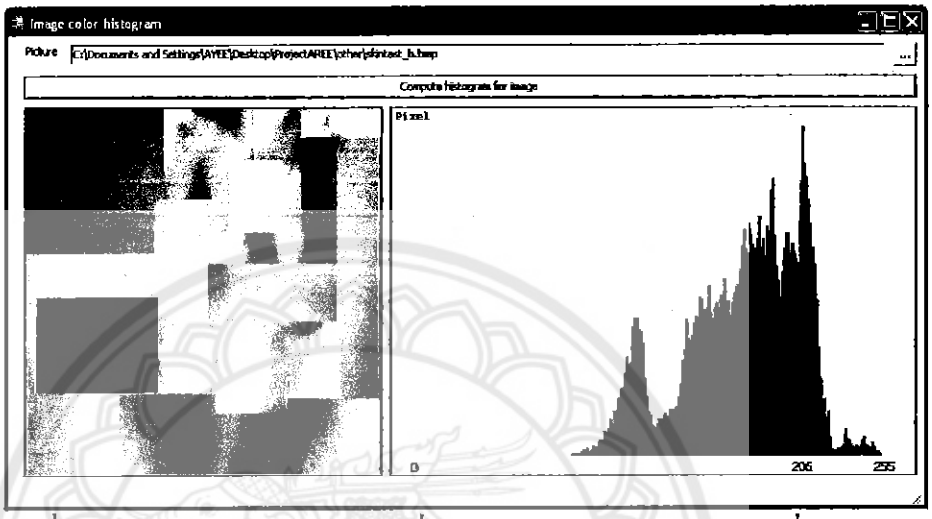
**การหา Saturation** สามารถคำนวณได้ดังนี้

$$\text{Saturation} = \frac{\text{Max(red, green, blue)} - \text{Min(red, green, blue)}}{\text{Max(red, green, blue)}}$$

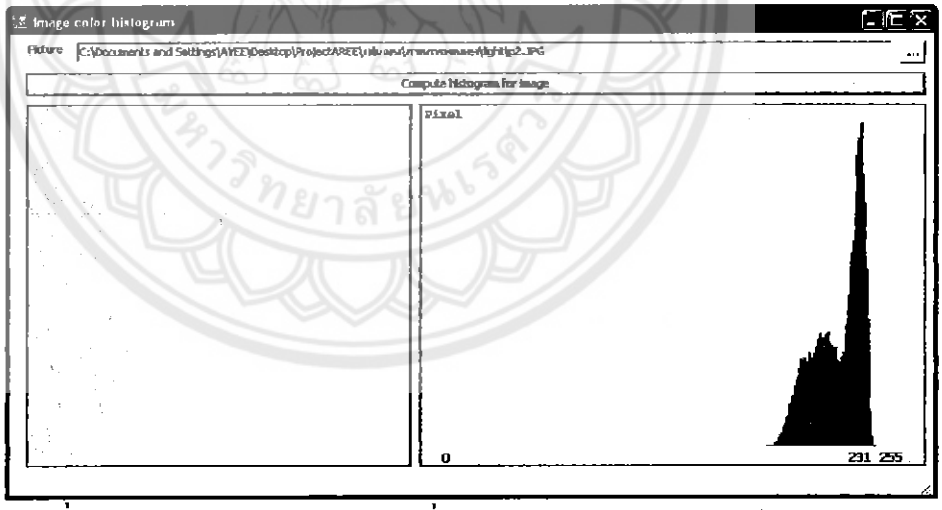
**การหา Value** สามารถคำนวณได้ดังนี้

Value = max (red , green , blue)

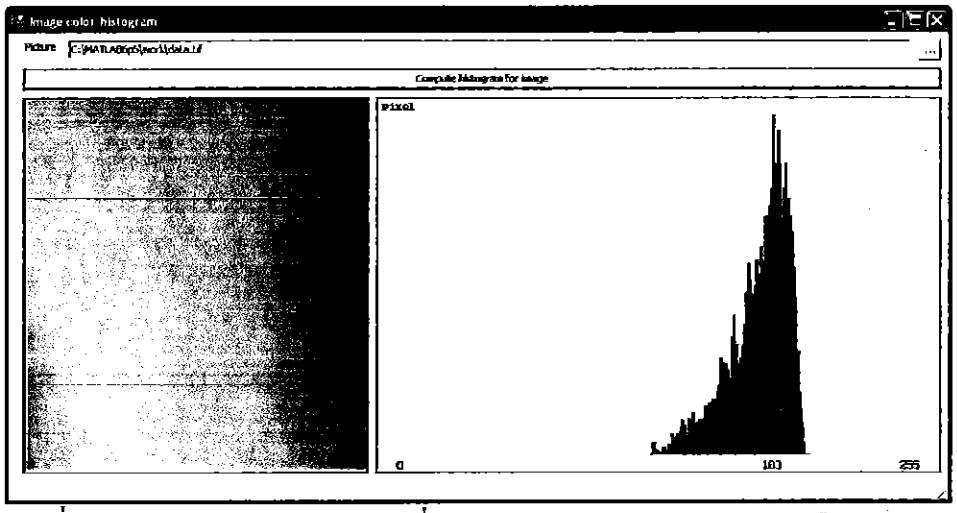
ตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิว



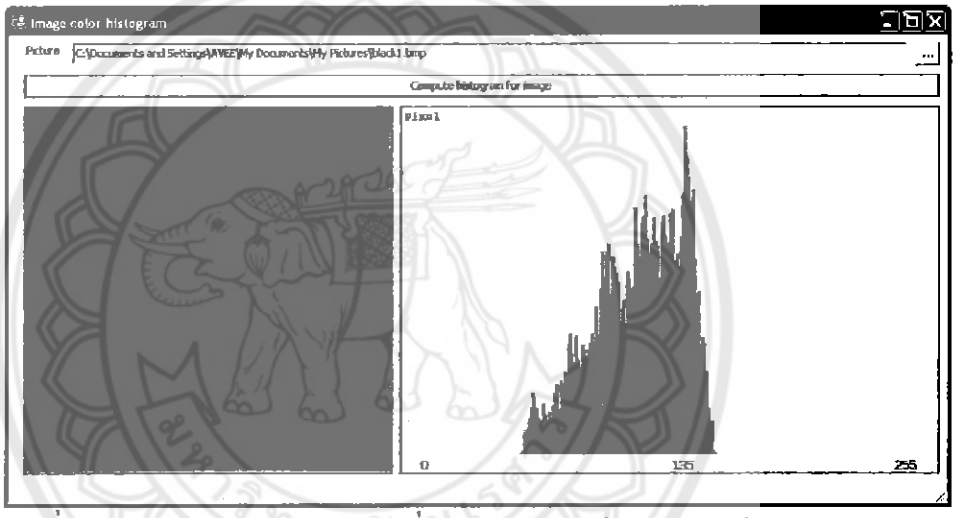

**รูปที่ 3.4** แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวที่หลากหลาย




**รูปที่ 3.5** แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวขาว (ยุโรป)




**รูปที่ 3.6** แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวเหลือง (เอเชีย)




**รูปที่ 3.7** แสดงตัวอย่างของโปรแกรมที่ช่วยในการหาค่าฮีสโตรแกรมสีผิวดำ (แอฟริกา)

เมื่อได้ค่าสีของโมเดลในระบบ HSV ซึ่งมีค่า H มีค่าอยู่ในช่วง 0 ถึง 360 , S และ V อยู่ในช่วง 0 ถึง 1 ดังนั้นเก็บค่าดังกล่าวไว้ และกำหนดให้ค่าสีผิวเป็นสองช่วงคือ ช่วงแรกค่า Hue มีค่ามากกว่าหรือเท่ากับ 0 แต่น้อยกว่า 40 , ค่า Saturation มีค่ามากกว่าหรือเท่ากับ 0 แต่น้อยกว่า 0.8 และค่า Value มีค่ามากกว่าหรือเท่ากับ 0.55 แต่น้อยกว่า 0.95 ส่วนในช่วงที่สองให้ค่า Hue มีค่ามากกว่าหรือเท่ากับ 230 แต่น้อยกว่า 360 ,ค่า Saturation มีค่ามากกว่าหรือเท่ากับ 0 แต่น้อยกว่า 0.8 และค่า Value มีค่ามากกว่าหรือเท่ากับ 0.55 แต่น้อยกว่า 0.95 และเก็บค่าไว้เพื่อเป็นต้นแบบในการเปรียบเทียบค่าสีผิว

### 3.2 การหาจำนวนพิกเซล (Pixel) ของสีผิว

เริ่มจากการนำเข้าข้อมูลภาพ (input) ขึ้นมาแล้วนับจำนวนพิกเซลโดยการใช้อัลกอลิทึมในการหาค่าของสีทั้งหมดแล้วหาเปอร์เซ็นต์ความน่าจะเป็นของจำนวนพิกเซลของสีผิวเทียบกับจำนวนพิกเซลทั้งหมด

$$\text{Precision} = \frac{b1 \times 100}{\text{pixels}}$$

เมื่อ b1 คือจำนวนพิกเซลของสีผิว

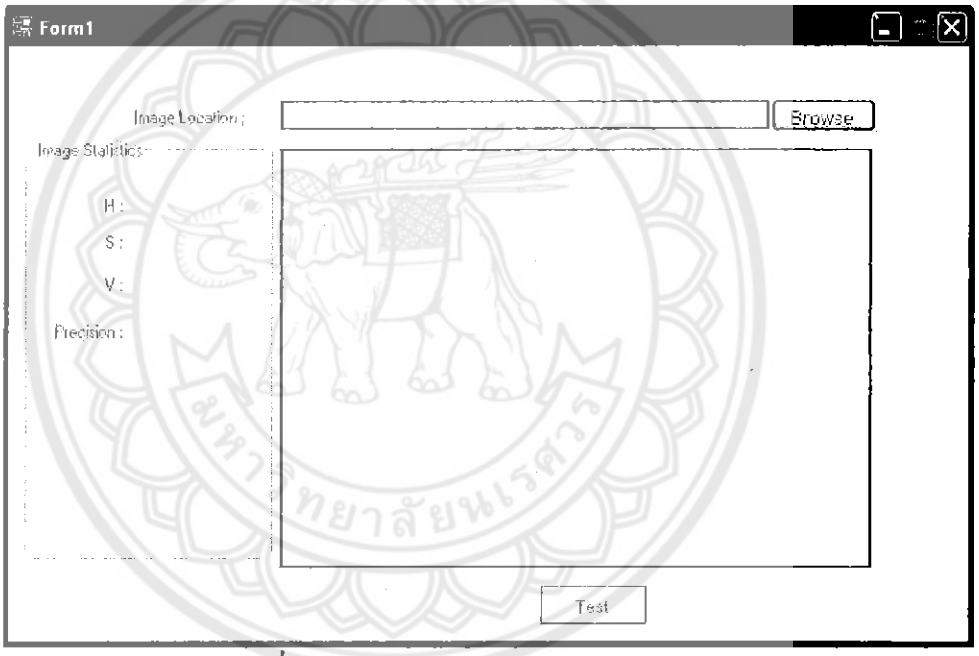
pixels คือจำนวนของพิกเซลทั้งหมดของภาพนั้น

# บทที่ 4
# ผลการทดลอง

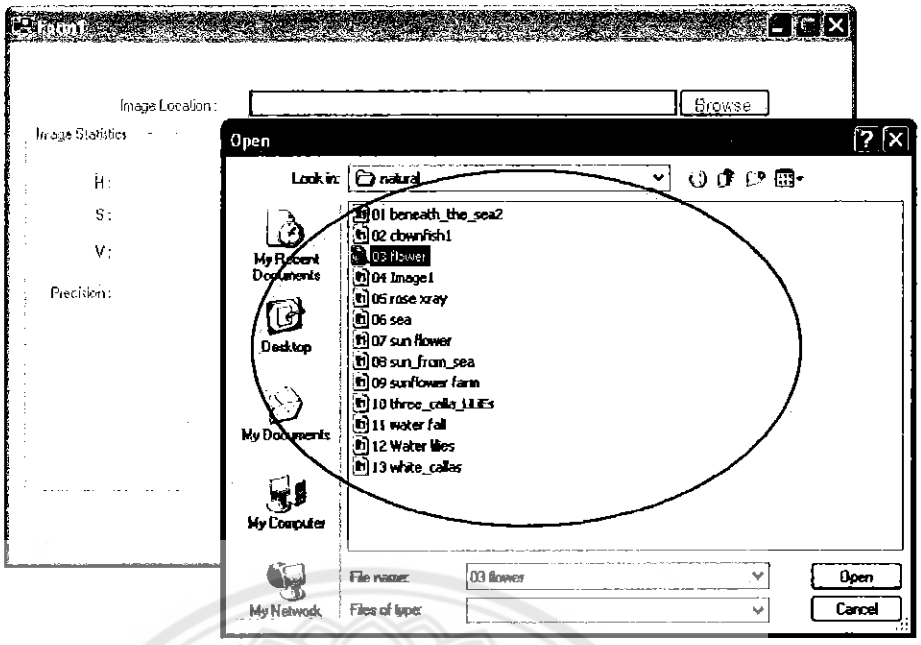
จากการที่ได้ศึกษาทฤษฎีทำการออกแบบการวิเคราะห์ภาพด้วยการวิเคราะห์จากสีผิวโดยสามารถวิเคราะห์ได้ดังนี้

## 4.1 การทดลองการตรวจจับสีผิว

ก่อนการวิเคราะห์นั้นจะต้องมีการนำเข้าข้อมูลภาพจากเครื่องคอมพิวเตอร์จึงจะทำการวิเคราะห์ได้



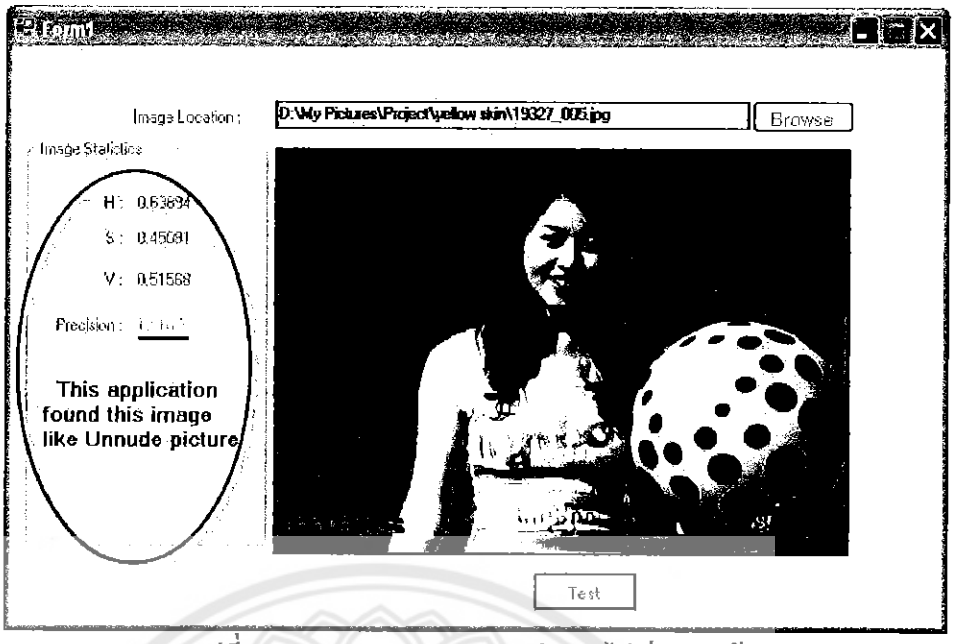

รูปที่ 4.1 แสดงหน้าโปรแกรมก่อนการเรียกภาพ




**รูปที่ 4.2** การเรียกข้อมูลภาพจากเครื่องคอมพิวเตอร์

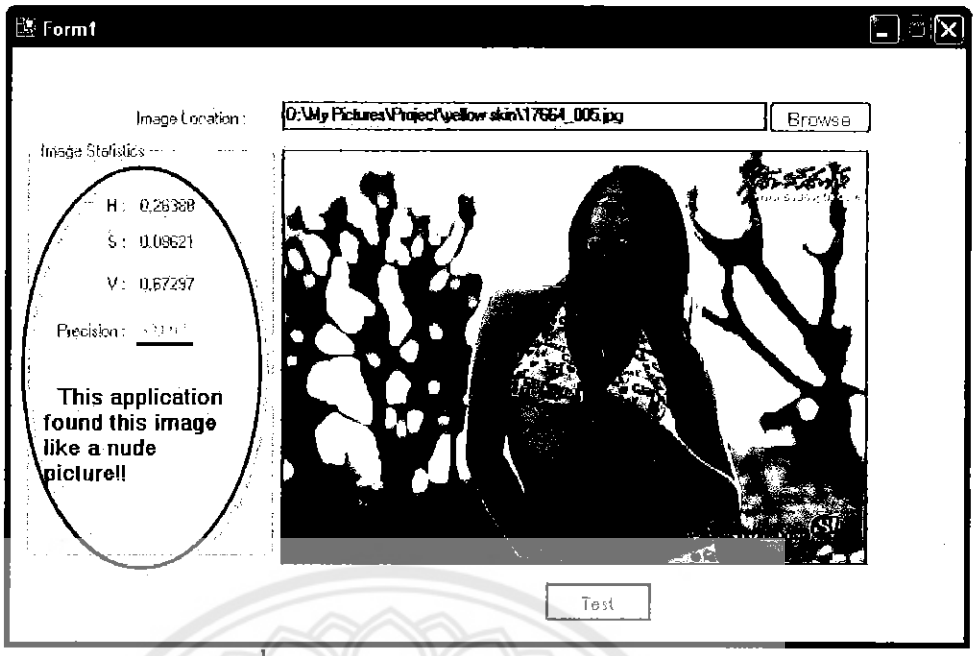
**รูปที่ 4.3** แสดงภาพหลังจากเรียกข้อมูลภาพจากคอมพิวเตอร์

จากรูปที่ 4.2 เมื่อทำการ Browse เพื่อนำเข้าข้อมูลภาพเข้ามาแล้วเมื่อกดปุ่มทดสอบ (Test) โปรแกรมจะทำการทดสอบ เพื่อตรวจจับจำนวนเปอร์เซ็นต์พิกเซลสีผิวและทำการวิเคราะห์ว่าเป็นภาพนู้ดหรือไม่




**รูปที่ 4.4** แสดงผลการตรวจสอบว่าภาพไม่เป็นภาพนู้ด

ตัวอย่างรูปที่ 4.4 มีค่าเปอร์เซ็นต์ของสีผิวเป็น 12.16 ซึ่งยังน้อยกว่าเกณฑ์**ที่ตั้งไว้** (20 เปอร์เซ็นต์) ดังนั้นโปรแกรมจะทำการวิเคราะห์ว่าภาพนี้ไม่เป็นภาพนู้ด




**รูปที่ 4.5** แสดงผลการตรวจสอบว่าภาพเป็นภาพนู้ด

ตัวอย่างรูปที่ 4.5 มีค่าเปอร์เซ็นต์ของสีผิวเป็น 39.60 ซึ่งมากกว่าเกณฑ์ที่ตั้งไว้ (20 เปอร์เซ็นต์) ดังนั้นโปรแกรมจะทำการวิเคราะห์ว่าภาพนี้เป็นภาพนู้ด

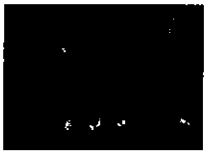
**ตารางที่ 4.1** แสดงตัวอย่างผลการทดลอง

| ชื่อภาพ | ภาพ | HSV | Precistion (%) | ผลการทดสอบ |
|---|---|---|---|---|
| 01 dark skin | | H : 0.29520<br>S : 0.27006<br>V : 0.34070 | 5.45 | Unnude picture |
| 02 dark skin | | H : 0.35592<br>S : 0.56287<br>V : 0.36398 | 5.90 | Unnude picture |

**ตารางที่ 3.1 (ต่อ)** แสดงตัวอย่างผลการทดลอง

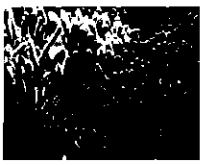
| ชื่อภาพ | ภาพ | HSV | Precistion (%) | ผลการทดสอบ |
|---|---|---|---|---|
| 03 dark skin | | H : 0.39832<br>S : 0.36117<br>V : 0.46894 | 14.64 | Unnude picture |
| 04 dark skin | | H : 0.20499<br>S : 0.68504<br>V : 0.21417 | 4.74 | Unnude picture |
| 05 dark skin | | H : 0.59498<br>S : 0.07153<br>V : 0.49338 | 28.68 | Nude Picture |
| 06 Pale skin | | H : 0.34004<br>S : 0.34316<br>V : 0.52491 | 10.37 | Unnude picture |
| 07 Pale skin | | H : 0.52251<br>S : 0.31275<br>V : 0.77655 | 27.37 | Nude Picture |
| 08 Pale skin | | H : 0.36555<br>S : 0.49042<br>V : 0.53552 | 5.60 | Unnude picture |
| 09 Pale skin | | H : 0.31669<br>S : 0.45836<br>V : 0.58922 | 37.20 | Nude Picture |

**ตารางที่ 3.1 (ต่อ)** แสดงตัวอย่างผลการทดลอง

| ชื่อภาพ | ภาพ | HSV | Precistion (%) | ผลการทดสอบ |
|---|---|---|---|---|
| 10 Pale skin | | H : 0.64394<br>S : 0.34473<br>V : 0.46739 | 6.39 | Unnude Picture |
| 11 Yellowish skin | | H : 0.63667<br>S : 0.09892<br>V : 0.44166 | 14.58 | Unnude Picture |
| 12 Yellowish skin | | H : 0.14764<br>S : 0.29382<br>V : 0.20628 | 10.27 | Unnude Picture |
| 13 Yellowish skin | | H : 0.16639<br>S : 0.39880<br>V : 0.39430 | 7.63 | Unnude Picture |
| 14 Yellowish skin | | H : 0.48842<br>S : 0.39252<br>V : 0.75509 | 46.23 | Nude Picture |
| 15 Yellowish skin | | H : 0.46927<br>S : 0.30626<br>V : 0.72226 | 14.28 | Unnude picture |
| 16 Nature |  | H : 0.90030<br>S : 0.55983<br>V : 0.36205 | 0.47 | Unnude Picture |

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร


**ตารางที่ 3.1 (ต่อ)** แสดงตัวอย่างผลการทดลอง

| ชื่อภาพ | ภาพ | HSV | Precistion (%) | ผลการทดสอบ |
|---|---|---|---|---|
| 17 Nature | | H : 0.61068<br>S : 0.52066<br>V : 0.50639 | 2.60 | Unnude Picture |
| 18 Nature |  | H : 0.61242<br>S : 0.69713<br>V : 0.31221 | 14.35 | Unnude picture |
| 19 Nature | | H : 0.41147<br>S : 0.20870<br>V : 0.41479 | 9.02 | Unnude picture |
| 20 Nature | | H : 0.20952<br>S : 0.38882<br>V : 0.30933 | 5.75 | Unnude picture |

จากผลการทดสอบพบว่า ในการทดสอบภาพสีจาก 20 ภาพ ตรวจสอบได้ถูกต้อง 17 ภาพ และไม่ถูกต้อง 3 ภาพ ซึ่งมีค่าความถูกต้องคิดเป็น 85 เปอร์เซ็นต์ จากภาพทั้งหมด

# บทที่ 5
# สรุปและวิเคราะห์ผล

โครงงานนี้จัดทำขึ้นเพื่อศึกษาและ พัฒนาโปรแกรมโดยใช้หลักการของการประมวลผลภาพด้วยสี (Color Image Processing) โดยทำการหาช่วงค่าฮีสโตรแกรมของสีผิว เพื่อหาค่าเปอร์เซ็นต์ส่วนของสีผิวเทียบกับจำนวนพิกเซลทั้งหมดของภาพ เกณฑ์ที่ในการตัดสินคือ ถ้าตรวจพบจำนวนพิกเซลของสีผิวมากกว่า 20 เปอร์เซ็นต์ถือว่าเป็นภาพนู้ด ถ้าน้อยกว่า 20 เปอร์เซ็นต์จะไม่ถือว่าเป็นภาพนู้ด ซึ่งผลที่ได้จากการทำงานของโปรแกรมคือ การวิเคราะห์ค่าของสีผิวในระบบ HSV และแสดงค่าเปอร์เซ็นต์ของสีผิวเทียบกับจำนวนพิกเซลทั้งหมด

ในโครงงานนี้ใช้โปรแกรม Visual C# . NET ในการพัฒนาในส่วนของแอบพลิเคชั่น ซึ่งผลจากโปรแกรมที่ได้รับดังแสดงในบทที่ผ่านมา

## 5.1 สรุปผลการทดลองที่ได้จากการพัฒนาโปรแกรม

จากการทดลองนั้นจะเห็นได้ว่า ในการทำงานของโปรแกรมนั้นสามารถตรวจจับและบอกเปอร์เซ็นต์สีผิวเทียบกับจำนวนพิกเซลของภาพทั้งหมดได้ การทำงานของโปรแกรมสามารถวิเคราะห์หาค่าสีในระบบ HSV ของภาพได้ คุณภาพของการตรวจจับนั้นสามารถตรวจจับสีผิวได้ทั้งสีผิวขาว (ยุโรป), สีผิวเหลือง (เอเชีย) และสีผิวดำ (แอฟริกา) และสามารถวิเคราะห์ได้ว่าภาพนั้นเป็นภาพนู้ดหรือไม่ แต่ในการวิเคราะห์นั้นยังเกิดค่าความผิดพลาดอันเนื่องมา จากสภาวะแสง ซึ่งจะทำให้ช่วงฮีสโตรแกรมของค่าสีผิวเปลี่ยนไป และความผิดพลาดที่เกิดจากขนาดของตัวบุคคล และ พื้นที่สีผิวที่ทำให้ไม่สามารถบ่งชี้ได้ชัดว่าเป็นมนุษย์หรือเป็นเพียงส่วนหนึ่งของผิวมนุษย์เท่านั้น ซึ่งเนื่องมาจากข้อจำกัดของหลักการที่ใช้ในการดำเนินโครงงาน

ในการทำโครงงานครั้งนี้ผู้จัดทำได้รับความรู้ในส่วนของหลักการการประมวลผลภาพ (Image Processing) ซึ่งนำมาประยุกต์ใช้ในการสร้างโปรแกรมตรวจจับภาพนู้ด ได้รับความรู้ในการเขียนโปรแกรมด้วย Visual C#.NET ซึ่งโปรแกรมนี้สามารถวิเคราะห์ผลได้เป็นที่น่าพอใจตามวัตถุประสงค์

## 5.2 ข้อเสนอแนะ

เนื่องจากโครงงานนี้เป็นการเริ่มการวิจัยโครงงานในขั้นเริ่มต้นถ้าหากมีผู้ที่ต้องการที่จะพัฒนาโครงงานนี้ต่อไป ทางผู้จัดทำมีข้อแนะนำดังต่อไปนี้

5.2.1 เนื่องจากการทำโครงงานครั้งนี้ใช้ทฤษฎีหลักการวิเคราะห์หาพื้นที่สีผิวด้วยฮีสโตรแกรมภาพเท่านั้น ผลการตรวจจับภาพนู้ดนั้นจึงได้ผลดีในระดับหนึ่ง

5.2.2 หาต้องการตรวจจับสีผิวมนุษย์ควรจำกัดในเรื่องของสภาวะแสง หรือแก้ปัญญาในเรื่องของแสงที่มากเกินไปซึ่งจะทำให้ค่าฮีสโตรแกรมของสีผิวผิดเพี้ยนไป

5.2.3 เพื่อการตรวจจับภาพนู้ดที่สมบูรณ์ ควรใช้ทฤษฎีลักษณะพื้นฐานของภาพทฤษฎีอื่นด้วยเช่นลักษณะพื้นผิว (Texture) และรูปร่างของภาพ (Shape) เพื่อการทำงานที่มีประสิทธิภาพของโปรแกรม

## 5.3 ปัญหาที่พบในการพัฒนาโครงงานวิจัย

5.3.1 ในการหาค่าพิกเซลสีผิวอย่างเดียวนั้น ยังไม่เพียงพอต่อการตรวจจับภาพนู้ด ทั้งนี้ควรใช้หลักการและทฤษฎีอื่นด้วย

5.3.2 ในการบ่งชี้ว่าเป็นภาพนู้ดหรือไม่นั้นยังไม่สามารถแยกแยะระหว่างภาพที่เป็นรูปร่างมนุษย์หรือพื้นที่สีผิวบางส่วนของร่างกายมนุษย์เท่านั้น

5.3.3 ขนาดของรูปร่างมนุษย์ที่อยู่บนภาพมีส่วนทำให้เกิดความผิดพลาดในการวิเคราะห์ว่าเป็นภาพนู้ดหรือไม่ เมื่อเทียบกับจำนวนพิกเซลทั้งหมดของภาพ

5.3.4 เป็นการยากที่จะหาค่าฮีสโตรแกรมภาพภายใต้ข้อจำกัดของสภาวะแสง

5.3.5 ยังขาดประสบการณ์ในการเขียนโปรแกรมด้วยภาษา C#.NET

## 5.4 แนวทางในการพัฒนาโปรแกรม

5.4.1 พัฒนาการตรวจจับที่ถูกต้องและมีประสิทธิภาพมากขึ้นโดยการใช้ทฤษฎีอื่นมาประกอบด้วย

5.4.2 ใช้หลักการและทฤษฎีลักษณะพื้นฐานรูปร่างของภาพ (Shape) เพื่อแก้ปัญหาในการแยกแยะส่วนที่เป็นรูปร่างของมนุษย์กับพื้นที่บางส่วนของผิวมนุษย์ หรือรูปร่างของวัตถุอื่นที่มีค่าสีใกล้เคียงกับสีผิวของมนุษย์

5.4.2 พัฒนาโครงงานนี้ให้เป็นโปรแกรมที่สามารถตรวจจับหรือป้องกันเว็บไซด์หรือสื่อลามกบนเครื่องคอมพิวเตอร์ส่วนบุคคล (PC) หรือบนโครงข่ายอินเตอร์เนต (Internet)

# เอกสารอ้างอิง

[1] นิรันดร์ ประวิทย์ธนา. **เก่ง C# ให้ครบสูตร 1.** กรุงเทพมหานคร : บิ๊วตี้ กรุ๊ป จำกัด. 2545.

[2] ณัช ภู่วรวรรณ. **เริ่มต้นเรียนรู้ C# 1.** กรุงเทพมหานคร : ซีเอ็ดยูเคชั่นจำกัด (มหาชน). 2545.

[3] Maher A. Sid-Ahmed. **Image Processing.** Singapore : McGraw-Hill Book Co. 1995.

[4] S.Kawato,J.Ohya . **"Automatic Skin-Color Distribution Extraction for Face Detection and Tracking"** . [Online] Available : Http://www.mis.atr.jp /~skawato/ pdfs/ICSP2000.pdf .

[5] P. Kuchi, P. Gabbur, P. Susbbnna Bhat, S. David . **"Human Face Detection and Tracking using Skin Color Modeling andConnected Component Operators "** . [Online] Available : Http://www.ece.arizona.edu/~pgsangam/ietepaper.pdf .

[6] Michael J. Jones, James M. Rehg . **"Statistical Color Models with Application to Skin Detection"** . [Online] Available : Http://wwwstatic.cc.gatech.edu/~rehg/Papers/ SkinDetect - IJCV-lowres.pdf .

# ภาคผนวก ก

## 1. โปรแกรมวิเคราะห์ภาพ (Form1.cs)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Diagnostics;
using System.IO;

namespace WindowsApplication1

{
    public partial class Form1 : Form
    {
        private string fileName = string.Empty;
        private int pixels = 0;
        private Bitmap his = null;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {

                if (openFileDialog1.ShowDialog() == DialogResult.OK)
                {
                    fileName = openFileDialog1.FileName;
                    System.IO.FileStream op =
(System.IO.FileStream)openFileDialog1.OpenFile();
                    this.pictureBox1.Image = Image.FromStream(op);
                    this.textBox1.Text = fileName;

                    op.Close();

                }

            }
            catch { File.Delete("D:\\Image\test.jpg");}

        }

        protected void OpenFile(string fileName)
        {
            try
            {
                Stream stream = File.OpenRead(fileName);
                using (StreamReader read = new StreamReader(stream))
                {
```

```
                    read.Read();
                }
            }
            catch (IOException ex)
            {
                MessageBox.Show(ex.Message, "Fail to load image",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            }
        }

        private bool ThumbnailCallback()
        {
            return false;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                Random random = new Random();
                int num = random.Next(10000);
                Image.GetThumbnailImageAbort myCallback = new
Image.GetThumbnailImageAbort(ThumbnailCallback);
                Image img_thumb = Image.FromFile(fileName);
                Image myThumbnail =
img_thumb.GetThumbnailImage((140), (90), myCallback, IntPtr.Zero);
                myThumbnail.Save("D:\\Image\\test" + num+ ".jpg");
                his = new Bitmap("D:\\Image\\test" + num + ".jpg");
                int x = 0;
                int y = 0;
                double b1 = 0;
                double Hue = 0;
                double HueValue = 0;
                double Saturation = 0;
                double Value = 0;
                int height = his.Height;
                int width = his.Width;
                pixels = width * height;
                double precision = 0;
                double Hmean = 0;
                double Smean = 0;
                double Vmean = 0;
                double HH = 0; double HS = 0; double HV = 0;
                double resultH = 0;double resultS = 0;double resultV
= 0;

                for (y = 0; y < height; y++)
                {
                    for (x = 0; x < width; x++)
                    {

                        HH = his.GetPixel(x, y).GetHue();
                        double hpoint = (Convert.ToDouble(HH));
                        resultH = resultH + hpoint;


                        HS = his.GetPixel(x, y).GetSaturation();
                        double spoint = (Convert.ToDouble(HS));
                        resultS = resultS + spoint;
```

```
                        HV = his.GetPixel(x, y).GetBrightness();
                        double vpoint = (Convert.ToDouble(HV));
                        resultV = resultV + vpoint;


//************************************************************

                        Hue = his.GetPixel(x, y).GetHue();
                        HueValue = (Hue);
                        Saturation = his.GetPixel(x,
y).GetSaturation();
                        Value = his.GetPixel(x, y).GetBrightness();

                        if (HueValue >= 0 && HueValue < 40 &&
Saturation >= 0 && Saturation < 0.8 && Value >= 0.55 && Value < 0.95)
                        {
                            b1 = b1 + 1;
                        }
                        if (HueValue >= 230 && HueValue < 360 &&
Saturation >= 0 && Saturation < 0.8 && Value >= 0.55 && Value < 0.95)
                        {
                            b1 = b1 + 1;
                        }

                    }
                }
                Hmean = ((resultH / pixels) / 360);
                Smean = (resultS / pixels);
                Vmean = (resultV / pixels);
                string hVal = Hmean.ToString("F5");
                string sVal = Smean.ToString("F5");
                string vVal = Vmean.ToString("F5");
                precision = (b1 * 100) / pixels;
                this.label7.Text = hVal;
                this.label8.Text = sVal;
                this.label9.Text = vVal;
                this.label3.Text = precision.ToString("F2")+" %";
                if (precision > 20)
                {
                    this.textBox2.ForeColor = Color.Red;
                    this.textBox2.Text = "  This application found
this image like a nude picture!!";
                }
                else
                {
                    this.textBox2.ForeColor = Color.Blue;
                    this.textBox2.Text = "  This application found
this image like Unnude picture";
                }
                //MessageBox.Show("This precision:" +
precision.ToString("F2") + "%", "Output");
            }
            catch { }
        }

       /* private void button3_Click(object sender, EventArgs e)
        {
            this.pictureBox1.Image = Image.FromFile("C:\\Documents
and Settings\\MAKO RYO\\My Documents\\Visual Studio
2005\\Projects\\WindowsApplication1\\WindowsApplication1\\Resources\\
unexcept.JPG");
```

```
            MessageBox.Show("T555");
            File.Delete(fileName);
        }*/

        private void Form1_Load(object sender, EventArgs e)
        {
            Directory.CreateDirectory("D:\\Image");
        }

        private void Form1_FormClosed(object sender,
FormClosedEventArgs e)
        {
            try
            {
                Directory.Delete("D:\\Image");
            }
            catch
            {
            }
        }

        private void label1_Click(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {

        }

        private void button4_Click(object sender, EventArgs e)
        {

        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {

        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {

        }

    }
}
```

## 2. โปรแกรมการออกแบบแอพพลิเคชั่น (From1.design)

```
namespace WindowsApplication1
{
   partial class Form1
   {
      /// <summary>

      /// Required designer variable.
      /// </summary>
      private System.ComponentModel.IContainer components = null;

      /// <summary>
      /// Clean up any resources being used.
      /// </summary>
      /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
      protected override void Dispose(bool disposing)
      {
         if (disposing && (components != null))
         {
            components.Dispose();
         }
         base.Dispose(disposing);
      }

      #region Windows Form Designer generated code

      /// <summary>
      /// Required method for Designer support - do not modify
      /// the contents of this method with the code editor.
      /// </summary>
```

```
private void InitializeComponent()
{
    this.pictureBox1 = new System.Windows.Forms.PictureBox();
    this.button1 = new System.Windows.Forms.Button();
    this.openFileDialog1 = new System.Windows.Forms.OpenFileDialog();
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.label1 = new System.Windows.Forms.Label();
    this.button2 = new System.Windows.Forms.Button();
    this.label2 = new System.Windows.Forms.Label();
    this.label3 = new System.Windows.Forms.Label();
    this.groupBox1 = new System.Windows.Forms.GroupBox();
    this.textBox2 = new System.Windows.Forms.TextBox();
    this.label9 = new System.Windows.Forms.Label();
    this.label8 = new System.Windows.Forms.Label();
    this.label7 = new System.Windows.Forms.Label();
    this.label6 = new System.Windows.Forms.Label();
    this.label5 = new System.Windows.Forms.Label();
    this.label4 = new System.Windows.Forms.Label();
    ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
    this.groupBox1.SuspendLayout();
    this.SuspendLayout();
    //
    // pictureBox1
    //
    this.pictureBox1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
    this.pictureBox1.Location = new System.Drawing.Point(194, 73);
    this.pictureBox1.Name = "pictureBox1";
    this.pictureBox1.Size = new System.Drawing.Size(419, 295);
    this.pictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage;
    this.pictureBox1.TabIndex = 0;
    this.pictureBox1.TabStop = false;
```

```
            //
            // button1
            //
            this.button1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.button1.ForeColor = System.Drawing.SystemColors.GradientActiveCaption;
            this.button1.Location = new System.Drawing.Point(542, 38);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(74, 23);
            this.button1.TabIndex = 1;
            this.button1.Text = "Browse";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Click += new System.EventHandler(this.button1_Click);
            //
            // openFileDialog1
            //
            this.openFileDialog1.FileName = "openFileDialog1";
            //
            // textBox1
            //
            this.textBox1.BackColor = System.Drawing.SystemColors.Control;
            this.textBox1.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
            this.textBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.5F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.textBox1.ForeColor = System.Drawing.SystemColors.ControlText;
            this.textBox1.Location = new System.Drawing.Point(194, 39);
            this.textBox1.Name = "textBox1";
            this.textBox1.Size = new System.Drawing.Size(346, 20);
            this.textBox1.TabIndex = 2;
            this.textBox1.TextChanged += new System.EventHandler(this.textBox1_TextChanged);
```

```
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.label1.ForeColor = System.Drawing.SystemColors.GradientActiveCaption;
            this.label1.Location = new System.Drawing.Point(86, 44);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(86, 13);
            this.label1.TabIndex = 3;
            this.label1.Text = "Image Location :";
            this.label1.Click += new System.EventHandler(this.label1_Click);
            //
            // button2
            //
            this.button2.FlatStyle = System.Windows.Forms.FlatStyle.Flat;
            this.button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.button2.ForeColor = System.Drawing.SystemColors.GradientActiveCaption;
            this.button2.Location = new System.Drawing.Point(379, 381);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(75, 27);
            this.button2.TabIndex = 4;
            this.button2.Text = "Test";
            this.button2.UseVisualStyleBackColor = true;
            this.button2.Click += new System.EventHandler(this.button2_Click);
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.ForeColor = System.Drawing.SystemColors.GradientActiveCaption;
```

```
this.label2.Location = new System.Drawing.Point(18, 128);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(56, 13);
this.label2.TabIndex = 5;
this.label2.Text = "Precision :";
this.label2.Click += new System.EventHandler(this.label2_Click);
//
// label3
//
this.label3.AutoSize = true;
this.label3.ForeColor = System.Drawing.SystemColors.Desktop;
this.label3.Location = new System.Drawing.Point(77, 128);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(0, 13);
this.label3.TabIndex = 6;
//
// groupBox1
//
this.groupBox1.Controls.Add(this.textBox2);
this.groupBox1.Controls.Add(this.label9);
this.groupBox1.Controls.Add(this.label8);
this.groupBox1.Controls.Add(this.label7);
this.groupBox1.Controls.Add(this.label6);
this.groupBox1.Controls.Add(this.label5);
this.groupBox1.Controls.Add(this.label4);
this.groupBox1.Controls.Add(this.label2);
this.groupBox1.Controls.Add(this.label3);
this.groupBox1.ForeColor = System.Drawing.SystemColors.GradientActiveCaption;
this.groupBox1.Location = new System.Drawing.Point(12, 67);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(176, 295);
this.groupBox1.TabIndex = 7;
}
```

```
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Image Statistics";
            this.groupBox1.Enter += new System.EventHandler(this.groupBox1_Enter);
            //
            // textBox2
            //
            this.textBox2.BackColor = System.Drawing.SystemColors.Control;
            this.textBox2.BorderStyle = System.Windows.Forms.BorderStyle.None;
            this.textBox2.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.textBox2.ForeColor = System.Drawing.SystemColors.ControlText;
            this.textBox2.Location = new System.Drawing.Point(12, 172);
            this.textBox2.Multiline = true;
            this.textBox2.Name = "textBox2";
            this.textBox2.Size = new System.Drawing.Size(144, 101);
            this.textBox2.TabIndex = 8;
            this.textBox2.TextChanged += new System.EventHandler(this.textBox2_TextChanged);
            //
            // label9
            //
            this.label9.AutoSize = true;
            this.label9.Location = new System.Drawing.Point(77, 95);
            this.label9.Name = "label9";
            this.label9.Size = new System.Drawing.Size(0, 13);
            this.label9.TabIndex = 12;
            //
            // label8
            //
            this.label8.AutoSize = true;
            this.label8.Location = new System.Drawing.Point(77, 39);
            this.label8.Name = "label8";
            this.label8.Size = new System.Drawing.Size(0, 13);
```

```
this.label8.TabIndex = 11;
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(77, 65);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(0, 13);
this.label7.TabIndex = 10;
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(54, 95);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(20, 13);
this.label6.TabIndex = 9;
this.label6.Text = "V :";
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(54, 65);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(20, 13);
this.label5.TabIndex = 8;
this.label5.Text = "S :";
```

```
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Location = new System.Drawing.Point(53, 39);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(21, 13);
            this.label4.TabIndex = 7;
            this.label4.Text = "H :";
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(682, 420);
            this.Controls.Add(this.groupBox1);
            this.Controls.Add(this.button2);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.button1);
            this.Controls.Add(this.pictureBox1);
            this.ForeColor = System.Drawing.SystemColors.Desktop;
            this.MaximizeBox = false;
            this.Name = "Form1";
            this.Text = "Form1";
            this.FormClosed += new
System.Windows.Forms.FormClosedEventHandler(this.Form1_FormClosed);
            this.Load += new System.EventHandler(this.Form1_Load);
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.ResumeLayout(false);
```

```
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.PictureBox pictureBox1;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.OpenFileDialog openFileDialog1;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.TextBox textBox2;
    }
}
```

# ประวัติผู้เขียนโครงงาน

ชื่อ นางสาวทิพนภา จันทร์หา

ภูมิลำเนา 59 ม.4 ต.ป่าพุทรา อ. ขาณุวรลักษบุรี จ. กำแพงเพชร

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนขาณุวิทยาจังหวัดกำแพงเพชร
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยนเรศวร

E-mail : ayeedunk@hotmail.com