



# Human Skeleton model

จัดทำโดย

นางสาว กนกวรรณ บัวภาคำ	6040200049
นาย ธรรมรัช ตันติปธรรม	6040201983
นาย อีระวัฒน์ ชรินทร์	6040202203
นางสาว รัตดาพร อักษรทอง	6040204010
นาย อนุชา ศรีลาแก้ว	6040205407

เสนอ

ดร.นิตยา เมืองนาค



## รายงาน

### เรื่อง Human Skeleton model

#### จัดทำโดย

นางสาว กนกวรรณ บัวภาคำ	6040200049
นาย ธรรมธัช ตันติปิธรรม	6040201983
นาย ชีระวัฒน์ ชรินทร์	6040202203
นางสาว รัตดาพร อักษรทอง	6040204010
นาย อนุชา ศรีลาแก้ว	6040205407

#### เสนอ

ดร.นิตยา เมืองนาค

รายวิชา 01204483 การประมวลผลสัญญาณภาพดิจิทัล

คณะวิทยาศาสตร์และวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตเฉลิมพระเกียรติ จังหวัดสกลนคร

พ.ศ. 2560

## บทคัดย่อ

การจัดทำโครงการนี้ได้จากการศึกษาการนำเสนอวิธีการประมาณท่าทางของมนุษย์เพื่อศึกษาการแบ่งส่วนและตรวจจับของรูปภาพในการวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ และเพื่อพัฒนาอินเทอร์เฟซคอมพิวเตอร์ (HCI) สำหรับโครงสร้างเสมือนจริง

ในเทคนิคที่นำเสนอแบบจำลองกราฟิก 3D นั้นถูกสร้างขึ้นครั้งแรก การฉายภาพบนระนาบภาพเสมือนจะใช้จับคู่กับเงาที่ได้ตรงกับภาพ ด้วยการปรับแบบจำลอง 3 มิติซ้ำ ๆ ของแบบจำลองกราฟิก 3 มิติ ด้วยข้อจำกัดทางกายภาพและกายวิภาคของการเคลื่อนไหวของมนุษย์ ท่าทางของมนุษย์และพารามิเตอร์การเคลื่อนไหว 3D ที่เชื่อมโยงสามารถระบุภาพได้หลากหลายและไม่ซ้ำกัน โดยผลการทดลองจะถูกนำเสนอด้วยภาพเสมือนจริง

## สารบัญ

บทคัดย่อ.....	
สารบัญ.....	
1. บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	1
2. การทบทวนวรรณกรรม.....	2
2.1 ขั้นตอนการทำงาน Human Skeleton model.....	2
3. วิธีการ	
จัดทำ.....	4
3.1 การได้มาซึ่งรูปภาพ.....	4
3.2 การเตรียมภาพ.....	4
3.3 การประมวลผลตามวัตถุประสงค์.....	5
3.4 การประเมินและวิเคราะห์ผล.....	7
4. สรุปผลและข้อเสนอแนะ.....	8
5. บรรณานุกรม.....	9
6. ภาคผนวก.....	10

# บทนำ

## 1. ที่มาและความสำคัญ

เนื่องจากในปัจจุบันนั้นมีการค้นคว้าและพัฒนาเทคโนโลยีต่างๆเกี่ยวกับการวิเคราะห์ความเคลื่อนไหวของมนุษย์และการปฏิสัมพันธ์ระหว่างมนุษย์กับคอมพิวเตอร์ (HCI) เพื่อนำมาประยุกต์ใช้ในการอำนวยความสะดวกหรือตอบสนองความต้องการของมนุษย์ ซึ่งระบบจัดการเคลื่อนไหวที่มีวงจำหน่ายส่วนใหญ่ต้องการให้ผู้ผู้ใช้ใส่เครื่องหมาย แทนข้อต่อหรือส่วนต่างๆ ของร่างกาย และเทคโนโลยีพื้นฐานที่ใช้นั้นจะมีราคาแพงและอาจไม่ตรงวัตถุประสงค์สำหรับการนำมาวิเคราะห์การเคลื่อนไหวของมนุษย์

จากการที่ได้ศึกษาเกี่ยวกับเทคนิคต่างๆ ในการปรับปรุงคุณภาพของภาพ ทางคณะผู้จัดทำ ได้เลือกเทคนิค Color Image Segmentation by Thresholding เพื่อศึกษาการแบ่งส่วนและตรวจจับของรูปภาพในการวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ โดยเริ่มจากเขียนโปรแกรมใน ภาษา Python ใช้ opencv-python เข้ามาช่วยในการใช้งานโปรแกรม การทำงานของโปรแกรม จะมีการกำหนดลักษณะต่างๆของร่างกายมนุษย์โดยทั้งหมด 19 องค์ประกอบที่ไล่จากศีรษะจนถึงเท้า และลักษณะท่าทางที่เป็นส่วนขยับของร่างกายเช่น คอ มือ แขน เป็นต้น

ซึ่งจากเหตุผลที่กล่าวมาข้างต้นทางคณะผู้จัดทำจึงมีความคิดริเริ่มที่อยากจะนำเอาความรู้ทาง Digital Image Processing และ Algorithm มาประยุกต์ใช้ เพื่อศึกษาผลลัพธ์ ซึ่งอาจจะเป็นแนวทางในการพัฒนาต่อยอดและนำไปประยุกต์ใช้ให้เข้ากับเทคโนโลยีต่างๆ ได้

## 2. วัตถุประสงค์

1. เพื่อศึกษาเกี่ยวกับ เทคนิค Color Image Segmentation by Thresholding
2. เพื่อศึกษาการแบ่งส่วนและตรวจจับของรูปภาพ
3. โปรแกรมสามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้

## การทบทวนวรรณกรรม

จากการศึกษาเนื้อหาจาก เอกสารและโครงการที่เกี่ยวข้องที่นิตยสารสืบค้นจากอินเทอร์เน็ต

### ขั้นตอนการทำงาน Human Skeleton model

#### 1. แบบจำลองและวิธีการอ้างอิง

อัลกอริทึมการประมวลผลภาพ 3D ที่นำเสนอขึ้นอยู่กับการเปรียบเทียบแบบจำลอง 3 มิติ แบบมนุษย์กับกราฟิกที่คาดการณ์ไว้อย่างชัดเจนและภาพที่ถ่าย กราฟิกแบบเชื่อมต่อแบบจำลองของมนุษย์ถูกสร้างขึ้นและปรับกับภาพซ้ำๆ เพื่อให้สอดคล้องกับภาพอินพุตขึ้นอยู่ข้อมูลของภาพ เงาม และสีของขอบเขตวัตถุ

##### 1.1 การสร้างแบบจำลองร่างกายมนุษย์

เนื่องจากการขาดข้อมูล 3D จากภาพอินพุต a จึงต้องสร้างแบบจำลองกราฟิก 3 มิติ ของร่างกายมนุษย์สำหรับการประมวลผลแบบ 3D ควรมีความสามารถในการแสดงการเคลื่อนไหวของมนุษย์ได้หลากหลาย ขนาดใหญ่และง่ายต่อการระบุจากเงา แบบจำลอง 3 มิติของมนุษย์ส่วนใหญ่สร้างขึ้นมีจำนวนชิ้นส่วนและข้อต่อที่แข็งแรง จำนวนองศาอิสระจึงเป็นปัจจัยสำคัญในการสร้างแบบจำลอง 3 มิติของมนุษย์ กราฟิกในงานนี้รูปแบบมนุษย์ 3D ถูกสร้างขึ้นโดยใช้ห้องสมุด OpenGL ประกอบด้วยส่วนต่างๆ ของร่างกาย 10 ข้อต่อต่อ 9 ข้ออิสรภาพ 22 องค์ประกอบต่างๆของร่างกายจะถูกแทนด้วยทรงกลมทรงรีและกระบอกสูบ สีที่แตกต่างกันจะถูกกำหนดให้กับส่วนต่างๆของร่างกายเพื่ออำนวยความสะดวกในกระบวนการจดจำท่าทาง เนื่องจากแบบจำลองกราฟิก 3 มิติได้ถูกฉายเป็นระนาบรูปภาพเสมือนสำหรับการจับคู่แม่แบบและการจัดแนวด้วยภาพฉากจริง พื้นที่วัตถุในภาพทั้ง 2 ควรมีขนาดและทิศทางเหมือนกัน ดังนั้นโมเดลมนุษย์ 3D จึงเป็นที่ยอมรับถูกสร้างขึ้นก่อนและการเริ่มต้นโมเดลในพื้นที่กระบวนการถูกดำเนินการสำหรับคนในฉาก

##### 1.2 การประมาณแบบ Pose จาก Silhouette

ด้วยภาพเงาเบื้องหน้าของร่างกายมนุษย์ ท่าทางที่เกี่ยวข้องนั้นถูกประเมินโดยการลดความแตกต่างระหว่างเงาในภาพฉากจริงและประมาณการของโมเดล 3 มิติบนระนาบภาพเสมือน การค้นหาโพสท่าที่ดีที่สุดของโมเดล 3 มิติกราฟิกที่เข้ากับท่าทางของมนุษย์ควรมีฟังก์ชันการวัดและราคาที่เหมาะสม ในงานแรกเดิมและคณะ ได้นำเสนอวิธีการแปลงระยะทางแบบยุคลิดเพื่อคำนวณระยะทางพิกเซลที่ชาญฉลาดระหว่างภาพเงาที่แท้จริงและเสมือนจริง ฟังก์ชันค่าใช้จ่ายที่กำหนดโดยผลรวมของระยะทางพิกเซลใช้ในการปรับแบบจำลอง 3 มิติ เนื่องจากรูปภาพทั้งหมดใช้สำหรับการเปรียบเทียบต้นทุนการคำนวณค่อนข้างสูงและผลลัพธ์มีแนวโน้มที่จะรวมกันเป็นท้องถิ่นขึ้นต่ำแตกต่างจากวิธีการจับคู่เงาทั้งหมด โดยเสนอเทคนิคการจัดตำแหน่งแบบหลายส่วน ส่วนต่างๆ ของร่างกายในภาพเงาที่เกิดขึ้นจริงและแบบจำลองจะถูกนำมาเปรียบเทียบและปรับทีละภาพโดยใช้ XOR ที่มีน้ำหนักหลักการทำงาน ความแตกต่างของพิกเซลจะได้รับการ

ประมวลผลภายในสำหรับแต่ละอันส่วนของร่างกายเพื่อให้ได้ผลการจัดตำแหน่งที่ดีขึ้นด้วยการคำนวณสามารถทำได้ นอกจากนี้ยังเหมาะสำหรับการเชื่อมต่อโมเดล 3 มิติที่มีข้อต่อและชิ้นส่วนที่แข็งแรงหากต้องการดำเนินการประมาณท่าทางแบบหลายส่วน ส่วนที่สำคัญที่สุด คือ ลำตัวจะถูกระบุก่อน มันคือส่วนกลางของภาพเบื้องหน้าเชื่อมต่อส่วนที่เหลือของส่วนต่างๆของร่างกาย เมื่อแตกลำตัวพื้นที่แล้วของหัวแขนขาส่วนบนและส่วนล่างสามารถหาได้ง่ายในการระบุลำตัวก่อนวนซ้ำเพื่อลบแขนขาออกในภาพเบื้องหน้าโมเดล 3 มิติที่ฉายจะถูกวางซ้อนบนกึ่งกลางของรูปเงาดำตามด้วยการหมุน 3 มิติ เพื่อลดความแตกต่างระหว่างส่วนท้ายของภาพเงาและ 2 มิติ

การฉายภาพของแบบจำลอง 3 มิติ หลังจากที่ได้รับแบบ 3D ของลำตัวขึ้นและบนแขนขาที่ต่ำกว่าจะถูกประมวลผลในลำดับของแขนข้อมือต้นขาและขา การจำแนกแขนขานั้นดำเนินการโดยการเปรียบเทียบอัตราส่วนพื้นที่หน้ากับพื้นหลังของแบบจำลองกราฟิก สำหรับส่วนต่างๆ ของร่างกายจะกำหนด 2 DOF สำหรับการหมุน (ไม่มีการหมุนตามแกนหลัก) ดังที่แสดงในรูป 1 (a) แขนมีความสามารถในการหมุน $360^{\circ}$ บนระนาบภาพ (แสดงเป็นมุม  $\theta$ ) และ $180^{\circ}$ จากระนาบภาพ (แทนด้วยมุม  $\phi$ ) เมื่อค้นหาโพสท่าของ แขนขา, มุม  $\theta$  ถูกระบุก่อนโดยหมุนส่วนของร่างกายที่สอดคล้องกันในแบบจำลอง 3 มิติ การหมุนเริ่มต้นหลายครั้ง ค้นด้วย $45^{\circ}$  ใช้เพื่อหลีกเลี่ยงการค้นหาแบบเต็มและเร่งกระบวนการจัดตำแหน่ง จากนั้นคำนวณมุม  $\phi$  โดยการตรวจจับการเปลี่ยนแปลงขนาดของชิ้นส่วนร่างกายที่คาดการณ์ไว้เนื่องจาก foreshortening ดังแสดงในรูปที่ 1 (b)

### 1.3 ข้อ จำกัดลักษณะที่ปรากฏ

เป็นที่ทราบกันดีว่าเงาจากหน้าไม่ได้ให้ข้อมูลการบดบังตัวเองของวัตถุที่จะทำให้ก่อให้เกิดอัลกอริธึมการประเมินที่แข็งแกร่งยิ่งขึ้นซึ่งใช้กันทั่วไป วิธีการคือ ใช้ข้อมูลสีและขอบของวัตถุโดยการแยกแต่ละส่วนของวัตถุ มีข้อจำกัดทางกายภาพและจลนศาสตร์บังคับใช้กับการเคลื่อนไหวของโมเดลมนุษย์เริ่มต้น 3 มิติ ดังนั้นการปิดกั้นตนเองของอวัยวะต่างๆ จึงไม่จำเป็นสักก่อนกระบวนการประเมินแบบจำลอง หนึ่งสามารถระบุจุดสิ้นสุดของแต่ละแขนร่วมกับข้อจำกัดข้างต้นเพื่อประมาณค่ามนุษย์ 3D ก่อให้เกิดความกำกวม projective ในกรณีนี้แต่ละส่วนของร่างกายถือเป็นลิงค์ของแบบจำลองโครงกระดูกมนุษย์และตำแหน่งของมือและเท้าควรระบุไว้ในภาพเบื้องหน้า



(a) 2 rotation DOF of a limb.



(b) Changes due to foreshortening.

รูปที่ 1 ภาพจากเปเปอร์

## วิธีการจัดทำ

จากการศึกษาเนื้อหาจาก เอกสารและโครงงานที่เกี่ยวข้องที่นิสิตสืบค้นจากอินเทอร์เน็ต และจากการที่ได้ศึกษาเกี่ยวกับเทคนิคต่างๆ ในการปรับปรุงคุณภาพของภาพ ทางคณะผู้จัดทำ ได้เลือกเทคนิค Color Image Segmentation by Thresholding เพื่อศึกษาการแบ่งส่วนและตรวจจับของรูปภาพในการวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์โดยเริ่มจากเขียนโปรแกรมใน ภาษา Python ใช้ opencv-python เข้ามาช่วยในการใช้งานโปรแกรม การทำงานของโปรแกรม จะมีการกำหนดลักษณะต่างๆ ของร่างกายมนุษย์โดยทั้งหมด 19 องค์ประกอบที่ได้จากศีรษะจนถึงเท้า และลักษณะท่าทางที่เป็นส่วนขยับของร่างกายเช่น คอ มือ แขน เป็นต้น

### 1. การได้มาซึ่งรูปภาพ

ภาพที่ใช้ในการทดสอบโปรแกรม คือ ภาพที่นำมาจากอินเทอร์เน็ต ซึ่งใช้ทั้งหมด 5 ภาพ และภาพที่ถ่ายเองอีก 1 ภาพ

### 2. การเตรียมภาพ

นำภาพที่ได้มาจากอินเทอร์เน็ต และภาพที่ถ่ายเอง มาปรับปรุงให้คุณสมบัติของภาพนั้นๆ เด่นขึ้น



รูปที่ 2 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



รูปที่ 3 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต





รูปที่ 4 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



รูปที่ 5 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



รูปที่ 6 ภาพที่ใช้ในการทดสอบจากอินเทอร์เน็ต



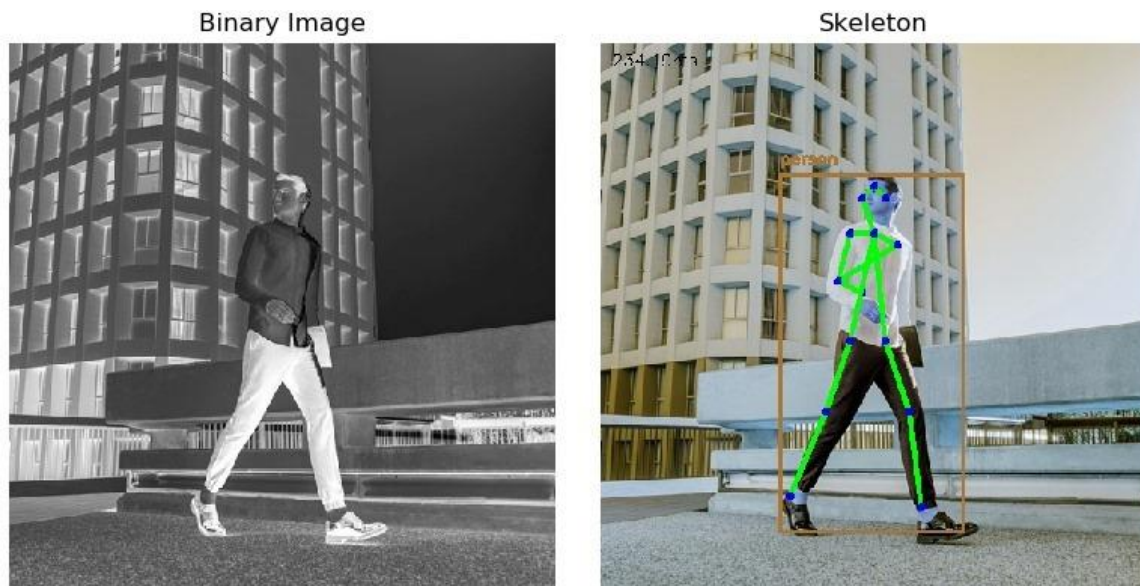
รูปที่ 7 ภาพที่ใช้ในการทดสอบจากการถ่ายเอง

### 3. การประมวลผลตามวัตถุประสงค์

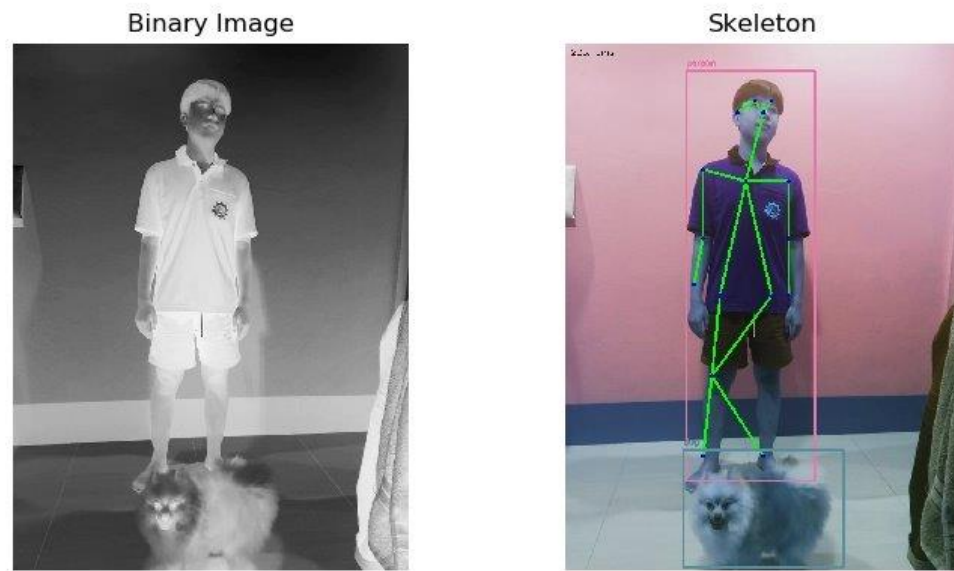
โดยเริ่มแรก การทำงานโปรแกรมจะมีการตรวจจับรูปภาพเข้ามาทาง input เพื่อทำการ scan ว่ารูปภาพที่เข้ามามีลักษณะอย่างไร (ซึ่งใช้ Ai ในการตรวจจับมันเร็วมาก) ถ้าหากการตรวจสอบผ่าน โปรแกรมก็จะเอาภาพ input ที่ได้มานำไปประมวลผลต่อและแสดงออกมาเป็น output ที่ detect 19 องค์ประกอบ ของ

ร่างกายมนุษย์นั้นคือ skeleton แต่จะตรวจจับถูกต้องหรือไม่ จึงต้องใช้ฟังก์ชัน assert เข้ามาช่วยในการทดสอบว่าองค์ประกอบที่ตรวจสอบ เข้ามานั้นถูกต้องและผิดพลาดน้อยที่สุดหลังจากผ่านการทดสอบจะเก็บค่าไว้ที่ Point เพื่อรอคำสั่งค้นหาชิ้นส่วนของร่างกายนั้นก็คือลักษณะท่าทางของมนุษย์ว่า ณ รูปภาพนั้นกำลังทำท่าทางอะไรอยู่ และถ้าหากมีความสอดคล้องกับตัวแปรที่กำหนดไว้ข้างต้นจะทำการเช็คค่าของ threshold ในท่าทางและลักษณะของร่างกายมนุษย์ทั้งหมดแต่ละส่วนเพื่อค่าสูงสุดและต่ำสุดเพื่อทำการโยกหากัน ถ้าหากค่า threshold ที่ได้มีค่าสูงกว่าปกติ การทำ Skeleton นั้นก็จะไม่เกิดขึ้นเพราะเนื่องจากค่า threshold สูงเกินไป และหากทำการ threshold เรียบร้อย จะเข้ามาใน loop ของการค้นหาองค์ประกอบและลักษณะท่าทางที่มีความสัมพันธ์กัน และเกิดการทดสอบขึ้นอีกครั้ง จากนั้นเมื่อทำการทดสอบเสร็จสมบูรณ์ก็จะเก็บค่าไว้ที่ตัวแปรตัวถัดไปก็คือ idFrom และ idTo

หมายความว่า ai จะเริ่ม detect ว่าทั้งหมดของร่างกายมนุษย์ตรงไหนสัมพันธ์กันมั้งและเริ่มโยงเส้น Skeleton จากตรงนี้ ถึงตรงนั้น จากนั้นจะใช้คำสั่ง if เพื่อทำการตรวจสอบว่า จุดนี้ไปตรงนั้นและจุดนั้นมาตรงนี้ ถูกหรือไม่ จากนั้นเขียนหน้าต่างหรือที่เรียกว่า Figure แบบใน matlab เพื่อจะแสดง output ออกมาให้เห็นก็จะกำหนดค่าต่างๆ จากนั้นก็จะแสดง Output ออกมา 2 รูปแบบ gray-scale นั่นก็คือ รูปแรกเป็นรูปปกติ รูปที่สองเป็นรูปที่ผ่านการตรวจสอบมาทุกวิธี



รูปที่ 8 เป็นภาพผลลัพธ์จากการทำ color image segmentation โดยภาพที่ได้จากอินเทอร์เน็ต



รูปที่ 9 เป็นภาพผลลัพธ์จากการทำ *color image segmentation* โดยภาพที่ถ่ายเอง

#### 4. การประเมินและวิเคราะห์ผล

output ออกมาให้เห็นก็จะกำหนดค่าต่างๆ จากนั้นก็จะแสดง Output ออกมา 2 รูปแบบ gray-scale นั่นก็คือ รูปแรกเป็นรูปปกติ รูปที่สองเป็นรูปที่ผ่านการตรวจสอบมาทุกวิธี

ซึ่ง output ที่ได้ สามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้ตรงตามที่คณะผู้จัดทำได้ศึกษามา

## สรุปผลและข้อเสนอแนะ

### 1. สรุปผล

จากการศึกษาการแบ่งส่วนและตรวจจับของรูปภาพในการวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ โดยเริ่มจากเขียนโปรแกรมใน ภาษา Python ใช้ opencv-python เข้ามาช่วยในการใช้งานโปรแกรม การทำงานของโปรแกรม และการทดลองใช้โปรแกรม ผลที่ได้คือ ตรงตามวัตถุประสงค์ของผู้จัดทำ เนื่องจาก โปรแกรมที่ทางผู้จัดทำได้ทำขึ้นนั้น สามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้

### 2. ข้อเสนอแนะ

เนื่องจาก เป็นการเรียกใช้การทำของ Image Segmentation โดยแบ่งหลักการให้ส่งข้อมูลเป็นรูปภาพที่มีร่างกายเป็นมนุษย์และใช้ opencv-python ในการตรวจจับเป็นภาพ Skeleton ขึ้นมา ยังมีปัญหาติดขัดอยู่ เนื่องจากเพิ่งได้ศึกษาการเขียน Opencv-python ส่วนใหญ่จะใช้ Video Capture ในการตรวจจับวัตถุและแสดงออกมาในลักษณะร่างของ Skeleton

ควรใช้ Video Capture ในการตรวจจับวัตถุและแสดงออกมาในลักษณะร่างของ Skeleton จะสามารถวิเคราะห์หารูปร่างและความเป็นไปได้ของโครงสร้างร่างกายมนุษย์ได้ดีกว่า

## บรรณานุกรม

1. Skeleton Viewer for Kinect V2 Skeletal Data :

<https://www.mathworks.cn/help/supportpkg/kinectforwindowsruntime/examples/plot-skeletons-with-the-kinect-v2.html?fbclid=IwAR3OK4eX55qPTydOAPMh--B9G0ez7n83OLsig2ZCLKuCPcUEwpX6kqf5zcs#d117e260>

2. Background Substraction With Python And OpenCV :

[http://grauonline.de/wordpress/?page\\_id=3065&fbclid=IwAR2j\\_Yi0drXbVgowHilyciqOTiMkPhiqiKolApsETeh7vgcGhdGdADaUTlg](http://grauonline.de/wordpress/?page_id=3065&fbclid=IwAR2j_Yi0drXbVgowHilyciqOTiMkPhiqiKolApsETeh7vgcGhdGdADaUTlg)

3. Skeletonize :

[https://scikit-image.org/docs/dev/auto\\_examples/edges/plot\\_skeleton.html?fbclid=IwAR0u2cZU3d1n0LwrHs3xupzYDH6r5fxcTP2PqQVOTc4ORTT6JBtoRpziWng](https://scikit-image.org/docs/dev/auto_examples/edges/plot_skeleton.html?fbclid=IwAR0u2cZU3d1n0LwrHs3xupzYDH6r5fxcTP2PqQVOTc4ORTT6JBtoRpziWng)

4. Object Detection of Code Using Python :

<https://towardsdatascience.com/object-detection-with-less-than-10-lines-of-code-using-python-2d28eebc5b11>

## ภาคผนวก

โค้ดแสดงการทำงานของโปรแกรม :

```
Terminal  Help  skeletonrgb.py - Project Image - Visual Studio Code

ai.py  skeletonrgb.py X

skeletonrgb.py > ...
1  import cv2
2  import numpy as np
3  import argparse #เป็น module ที่กำหนด argument ที่จะเรียกใช้งานได้จาก console
4  from skimage.morphology import skeletonize
5  import matplotlib.pyplot as plt
6  import cvlib as cv
7  from cvlib.object_detection import draw_bbox
8  parser = argparse.ArgumentParser() #ตัวกำหนดเรียกใช้ที่ console
9  parser.add_argument('--input') #เพิ่มรูปภาพโดยใช้ argument สิ่งที่ console
10 parser.add_argument('--thr', default=0.2, type=float) #ปรับระดับค่า threshold
11 parser.add_argument('--width', default=368, type=int) #กำหนดความกว้างของ figure
12 parser.add_argument('--height', default=368, type=int) #กำหนดความสูงของ figure
13 im = cv2.imread("make.jpg", 0) # ภาพ .jpg ไวท์เลส
14 args = parser.parse_args() #ตัวกำหนดเรียกใช้ argument ทั้งหมด
15
16 #กำหนดส่วนลักษณะต่างๆของร่างกายมนุษย์
17 BODY_PARTS = { "Nose": 0, "Neck": 1, "RShoulder": 2, "RElbow": 3, "RWrist": 4,
18               "LShoulder": 5, "LElbow": 6, "LWrist": 7, "RHip": 8, "RKnee": 9,
19               "RAnkle": 10, "LHip": 11, "LKnee": 12, "LAnkle": 13, "REye": 14,
20               "LEye": 15, "REar": 16, "LEar": 17, "Background": 18 }
21
22 #กำหนดท่าทางให้คล้องกับร่างกายมนุษย์
23 POSE_PAIRS = [ ["Neck", "RShoulder"], ["Neck", "LShoulder"], ["RShoulder", "RElbow"],
24               ["RElbow", "RWrist"], ["LShoulder", "LElbow"], ["LElbow", "LWrist"],
25               ["Neck", "RHip"], ["RHip", "RKnee"], ["RKnee", "RAnkle"], ["Neck", "LHip"],
26               ["LHip", "LKnee"], ["LKnee", "LAnkle"], ["Neck", "Nose"], ["Nose", "REye"],
27               ["REye", "REar"], ["Nose", "LEye"], ["LEye", "LEar"] ]
28
29
30
31 #แสดงหน้าต่าง gui
32 inWidth = args.width
33 inHeight = args.height
34 #เรียกใช้ไฟล์ graph_opt.pb
35 net = cv2.dnn.readNetFromTensorflow("graph_opt.pb")
36
37 #ai ตรวจสอบรูปภาพที่เข้ามาทาง input
38 cap = cv2.VideoCapture(args.input if args.input else 0)
39
40 #หลังจากตรวจสอบและผ่านการตรวจสอบจะเข้า loop while เพื่อหาไฟล์รูปภาพขึ้นมาให้ชื่อว่า frame ถ้าหากไม่ผ่านก็อาจจะ error หรือการตรวจสอบต่อไป
41 while cv2.waitKey(1) < 0:
42     hasFrame, frame = cap.read()
43     if not hasFrame:
44         cv2.waitKey()
45         break
46
47 #แสดงรูปร่างที่ได้ในหน้าต่าง gui
48 frameWidth = frame.shape[1]
49 frameHeight = frame.shape[0]
50
```

```

51 #เป็น output ที่เรียกใช้ 19 องค์ประกอบของร่างกายมนุษย์
52 net.setInput(cv2.dnn.blobFromImage(frame, 1.0, (inWidth, inHeight), (127.5, 127.5, 127.5), swapRB=True, crop=False))
53 out = net.forward()
54 out = out[:, :19, :, :]
55 #assert เป็นฟังก์ชันที่ไว้ทดสอบองค์ประกอบร่างกายของมนุษย์ตรงตาม output ที่เช็คเข้ามาใหม่
56 assert(len(BODY_PARTS) == out.shape[1])
57
58 #กำหนดตัวแปร points ขึ้นมาค่าว่างเปล่า
59 points = []
60
61 #คำสั่งวนซ้ำให้ค้นหาชิ้นส่วนของร่างกายให้ตรงตามองค์ประกอบ 19 ส่วน
62 for i in range(len(BODY_PARTS)):
63     heatMap = out[0, i, :, :] #ตรวจสอบชิ้นส่วนร่างกายที่มีความสอดคล้องกัน
64
65     #กำหนดตัวแปรหาค่าสูงสุดและค่าต่ำสุดของ input
66     _, conf, _, point = cv2.minMaxLoc(heatMap)
67     x = (frameWidth * point[0]) / out.shape[3]
68     y = (frameHeight * point[1]) / out.shape[2]
69
70     #จุดที่มีการเช็คค่า ถ้าหากค่าของ threshold มีค่าที่สูงกว่าปกติ
71     points.append((int(x), int(y)) if conf > args.thr else None)
72
73 #คำสั่ง loop ที่จะค้นหาองค์ประกอบใดครบสมบูรณ์ที่สุด
74 for pair in POSE_PAIRS:
75     partFrom = pair[0]
76     partTo = pair[1]
77     assert(partFrom in BODY_PARTS)
78     assert(partTo in BODY_PARTS)
79
80     #กำหนดตัวแปรเพิ่มเติม
81     idFrom = BODY_PARTS[partFrom]
82     idTo = BODY_PARTS[partTo]
83     #ใช้คำสั่ง if เพื่อตรวจสอบข้อมูลของรูปภาพที่เข้ามาและเช็คแต่ละเฟรม
84     if points[idFrom] and points[idTo]:
85         cv2.line(frame, points[idFrom], points[idTo], (0, 255, 0), 3)
86         cv2.ellipse(frame, points[idFrom], (3, 3), 0, 0, 360, (0, 0, 255), cv2.FILLED)
87         cv2.ellipse(frame, points[idTo], (3, 3), 0, 0, 360, (0, 0, 255), cv2.FILLED)
88
89 #กำหนดค่าหน้าต่างของ figure แบบใน matlab
90 fig, axes = plt.subplots(1, 2, figsize=(8, 4), sharex=True, sharey=True)
91 ax = axes.ravel()
92 skeleton = skeletonize(frame)
93
94 #แสดงรูปภาพที่ถูกประมวลผล
95 ax[0].imshow(im)
96 ax[0].set_title('threshold')
97 ax[0].axis('off')
98
99 ax[1].imshow(frame, cmap=plt.cm.gray)
100 ax[1].set_title('skeleton')
101 ax[1].axis('off')
102
103 #ตัวแปรที่พาเข้าไลบรารีที่จะทำการอ่านภาพจากหน่วยเก็บข้อมูลทำการตรวจสอบวัตถุบนภาพและแสดงออกมาเป็นกล่องข้อความ
104 bbox, label, conf = cv.detect_common_objects(im)
105 output_image = draw_bbox(frame, bbox, label, conf)
106
107 #คำนวณเวลาในการประมวลผลรูปภาพ
108 t, _ = net.getPerfProfile()
109 freq = cv2.getTickFrequency() / 1000
110 cv2.putText(frame, '%.2fms' % (t / freq), (10, 20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 0))
111 print('Success')
112 cv2.imshow('Human Skeleton using OpenCV', frame) #ตัวแสดงทั้งกล่องและ gui
113
114 fig.tight_layout()
115 plt.imshow(output_image)
116 plt.show()

```