โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา
Algorithm for analyzing roof types and counting crucial components from roof plan

ชื่อนิสิต นาย ธนวิน อาภาจรัส 603 36277 23

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา วิทยาการคอมพิวเตอร์

ปีการศึกษา 2563

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลัก และองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา

นาย ธนวิน อาภาจรัส

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

ALGORITHM FOR ANALYZING ROOF TYPES

AND COUNTING CRUCIAL COMPONENTS FROM ROOF PLAN

Thanawinn Arphacharas

A Project Submitted in Partial Fulfillment of the Requirements

for the Degree of Bachelor of Science Program in Computer Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2020

Copyright of Chulalongkorn University

หัวข้อโครงงาน	อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา
โดย	นาย ธนวิน อาภาจรัส
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงงานหลัก	รศ.ดร. นกุล คูหะโรจนานนท์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

..... หัวหน้าภาควิชาคณิตศาสตร์
(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี) และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงงาน

..... อาจารย์ที่ปรึกษาโครงงานหลัก
(รองศาสตราจารย์ ดร.นกุล คูหะโรจนานนท์)

..... กรรมการ
(อาจารย์ ดร.นฤมล ประทานวณิช)

..... กรรมการ
(อาจารย์ ดร.วุฒิชัย จงจิตเมตต์)

ธนวิน อาภาจรัส: อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา. (ALGORITHM FOR ANALZING ROOF TYPES AND COUNTING CRUCIAL COMPONENTS FROM ROOF PLAN) อ.ที่ปรึกษาโครงงานหลัก : รศ.ดร. นกุล คูหะโรจนานนท์, 66 หน้า.

โครงงานเรื่อง “อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา” มีวัตถุประสงค์ คือ เพื่อพัฒนาโปรแกรมทำหน้าที่ช่วยนับจำนวนองค์ประกอบที่สำคัญต่อการประเมินราคาและแปลงข้อมูลนำเข้าให้อยู่ในรูปแบบที่ใช้งานต่อได้ง่าย จุดมุ่งหมายของโครงงานคือเพื่อช่วยทำให้การเสนอราคาหลังคาเบื้องต้นให้กับลูกค้าได้อย่างรวดเร็วขึ้น และแบ่งเบาภาระงานในส่วนหนึ่งด้วยโปรแกรมคอมพิวเตอร์ ขอบเขตของโครงงานได้แก่ โปรแกรมทำงานได้เฉพาะระบบปฏิบัติการวินโดว์ มีตัวหลังคาเพียงหลังเดียวในแปลน รองรับหลังคาทรงปั้นหยา ทรงจั่ว และทรงเหยียบสิงห์ และองค์ประกอบสำคัญอื่น ประกอบด้วย ตะเข้สัน ตะเข้ราง เท่านั้น วิธีการดำเนินงานจะใช้วิธีการรวบรวมความต้องการ และข้อมูลที่จำเป็นจากพนักงานบริษัทที่ทำงานในส่วนนี้โดยตรง เพื่อนำไปออกแบบและวางแผนการพัฒนาโปรแกรมโดยใช้ Rule-Based Detection จากการรวมข้อมูลด้วยภาพย่อยในจุดต่าง ๆ ซึ่งผลการดำเนินงานที่ได้แสดงให้เห็นว่า โปรแกรมสามารถเป็นส่วนช่วยในการประเมินราคาหลังคาในเชิงธุรกิจได้

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต .....

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงงานหลัก .....

ปีการศึกษา 2563

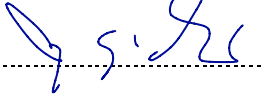
6033627723: MAJOR COMPUTER SCIENCE

KEYWORDS : ROOF / COST ESTIMATION / ALGORITHM

THNAWINN ARPHACHARAS: ALGORITHM FOR ANALZING ROOF TYPES AND COUNTING CRUCIAL COMPONENTS FROM ROOF PLAN. ADVISOR : ASSOC. PROF. NAGUL COOHAROJANANONE, Ph.D., 66 pp.

The objective of senior project "Algorithm for analzing roof types and counting crucial components from roof plan" are to develop a software that assist in detecting and counting crucial components in a roof plan and convert the output to the new format to improve its usability for future use. The goal of this project is to help deliver any roof construction cost to customer faster by using a specialized software. The limitations of this project are: software will only support Windows operation system, support a plan with only one roof, supported type of roof are: Hip, Gable and Dutch. Crucial components are: Hip valley, Raft valley. Methods of development are collecting requirements from employee that work in roof business then design and plan the development cycle to utilize sub-image for Rule-Based Detection. The results show that the software can gives output that is accurate enough to be beneficial to roof business in rough cost calculation.

Department: Mathematics and Computer Science Student's Signature

Field of Study: Computer Science Advisor's Signature 

Academic Year: 2020

กิตติกรรมประกาศ

โครงงานในชั้นเรียนในหัวข้อเรื่อง “อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา” ได้รับการสนับสนุนอย่างเต็มที่จากอาจารย์ที่ปรึกษา รศ.ดร. นกุล คูหะโรจนานนท์ และคุณปีย์ลักษณ์ วันทนาศิริ พนักงานแผนกหลังคาจากบริษัท ปูนซีเมนต์ไทย จำกัด (มหาชน) ทั้งในด้านองค์ความรู้ข้อมูลเฉพาะต่าง ๆ เกี่ยวกับหลังคา และเบี้ยเลี้ยงนักศึกษาฝึกงาน รวมถึงข้อมูลแปลนหลังคาตัวอย่างที่จำเป็นต้องใช้ในการพัฒนาโปรแกรม และจุฬาลงกรณ์มหาวิทยาลัยที่ให้งบประมาณในการทำวิจัย ซึ่งทำให้งานวิจัยนี้สำเร็จลุล่วงไปด้วยดี ข้าพเจ้าจึงใคร่ขอขอบพระคุณเป็นอย่างยิ่งสำหรับความช่วยเหลือในทุกๆ ด้าน ที่ได้รับจากผู้สนับสนุนทุกท่าน และหวังว่าผลลัพธ์จากโครงงานนี้จะเป็นประโยชน์ในการพัฒนาระบบการเรียนการสอน และพัฒนาธุรกิจหลังคาได้ต่อไป

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... จ

บทคัดย่อภาษาอังกฤษ ..... ฉ

กิตติกรรมประกาศ ..... ช

สารบัญ ..... ซ

สารบัญตาราง ..... ญ

สารบัญภาพ ..... ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและเหตุผล 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของโครงงาน 2

1.4 ขั้นตอนการพัฒนา 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 5

1.6 โครงสร้างของรายงาน 5

บทที่ 2 งานวิจัยที่เกี่ยวข้อง 6

2.1 Topological structural analysis of digitized binary images by border following 6

2.2 Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds from Aerial Imagery 6

2.3 A Fast Level Set Algorithm for Building Roof Recognition From High Spatial Resolution Panchromatic Images 7

บทที่ 3 วิธีการดำเนินงาน 8

3.1 บทนำ 8

3.2 สถาปัตยกรรมของโปรแกรม 8

3.3 การทำงานของอัลกอริธึม 10

หน้า

บทที่ 4 ผลการดำเนินงาน 19
4.1 ผลของการดำเนินการพัฒนา 19
4.1 ผลลัพธ์การทดสอบความแม่นยำของโปรแกรม 21

บทที่ 5 ข้อสรุปและข้อเสนอแนะ 25
5.1 ข้อสรุป 25
5.2 ข้อเสนอแนะ 25

รายการอ้างอิง 27
ภาคผนวก ก แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal ปีการศึกษา 2563 . 29
ภาคผนวก ข คู่มือการใช้งาน 33
ภาคผนวก ค ซอร์สโค้ด 40
ประวัติผู้เขียน 55

สารบัญตาราง

หน้า

ตารางที่ 3.1 ตัวอย่างตารางพิกเซล 5 x 5 11
ตารางที่ 3.2 ตารางตัวอย่างบริเวณที่นับจำนวนพิกเซลดำขนาด 10x10 12
ตารางที่ 3.3 ตัวอย่างภาพย่อยพร้อมจำนวนเส้นและตำแหน่งที่ระบุได้ 13

สารบัญภาพ

หน้า

ภาพที่ 1 Action plan ประกอบแผนการดำเนินงาน 4
ภาพที่ 3.1 Dataflow diagram 9
ภาพที่ 3.2 ตัวอย่างการครอบตัดภาพย่อยขอบหลังคา 10
ภาพที่ 3.3 ตัวอย่างของภาพย่อยที่ผ่านการทำ threshold 11
ภาพที่ 3.4 ตัวอย่างของภาพย่อยที่ผ่านการหมุนและลบเส้นที่นับแล้วออก 13
ภาพที่ 3.5 ตัวอย่างภาพย่อยซึ่งมีตะเข้สัน 4 แบบ 14
ภาพที่ 3.6 ตัวอย่างภาพย่อยซึ่งไม่มีมีตะเข้ 4 แบบ 14
ภาพที่ 3.7 ตัวอย่างภาพย่อยซึ่งเป็นองค์ประกอบของหลังคาเหยียบสิงห์ 8 แบบ 14
ภาพที่ 3.8 ตัวอย่างภาพย่อยซึ่งมีตะเข้ราง 4 แบบ 14
ภาพที่ 3.9 ตัวอย่างภาพย่อยซึ่งเป็นองค์ประกอบของหลังคาปั้นหยา 4 แบบ 15
ภาพที่ 3.10 แปลนหลังคาปั้นหยาที่มีเส้นรบกวนและภาพย่อยคู่มุมปั้นหยา 15
ภาพที่ 3.11 แปลนหลังคาจั่วที่ไม่มีเส้นรบกวนและภาพย่อยคู่มุมหลังคาจั่ว 16
ภาพที่ 3.12 หัวหลังคาปั้นหยาและพิกัดกึ่งกลางที่เกิดจากการเฉลี่ยพิกัดที่พบ4จุด 17
ภาพที่ 3.13 ตัวอย่างหลังคาเหยียบสิงห์ และภาพย่อยที่ใช้ในการจับคู่เพื่อระบุทรงหลังคา 18
ภาพที่ 3.14 ตะเข้สันที่ไม่เชื่อมกับมุมหลังคาและตะเข้สันที่เชื่อมกับมุม 18
ภาพที่ 4.1 ส่วนประสานผู้ใช้ของโปรแกรม 20
ภาพที่ 4.2 ตัวอย่างหลังคาเรียงตามความยาก 21
ภาพที่ 4.3 แผนภูมิแท่งของผลลัพธ์ Precision และ Recall ทั้งสามระดับความยาก 22
ภาพที่ 4.4 แผนภูมิแท่งของผลลัพธ์องค์ประกอบย่อย ทั้งสามระดับความยาก 23
ภาพที่ 4.5 แปลนหลังคาเหยียบสิงห์ระดับปานกลางที่ตรวจจับไม่ครบ 23
ภาพที่ 4.6 แปลนหลังคาเหยียบปั้นหยาผสมจั่วระดับปานกลางที่ตรวจจับครบถ้วน 24
ภาพที่ 5.1 ระยะของเส้นหลังคาที่เกิดจากการวาดเชื่อมจุด และแปลนที่มีการจุดสีลงไป 26

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุผล

การนำแบบแปลนหลังคาบ้านมาประเมินราคาในปัจจุบัน จะต้องนำแบบที่ลูกค้าส่งมาในรูปแบบของกระดาษหรือไฟล์ ไปขึ้นโครงในโปรแกรมเขียนแบบ ด้วยการให้พนักงานอ่านจากไฟล์ และกรอกแต่ละรายละเอียดลงไปในโปรแกรมร่างแบบ จากนั้นก็จะทำการนับองค์ประกอบที่จำเป็นในการประเมินราคาด้วยกำลังคน และอาศัยจำนวนองค์ประกอบและขนาดของหลังคา เพื่อให้ทราบถึงปริมาณวัสดุที่จะต้องใช้ที่จะนำไปประมาณเป็นราคา จนได้ผลลัพธ์เป็นใบเสนอราคาที่จะนำไปให้ลูกค้าพิจารณา

จากที่กล่าวมาข้างต้น จะเห็นว่ากระบวนการนี้ใช้กำลังคนเป็นส่วนใหญ่ แต่มีบางขั้นตอนที่สามารถพัฒนาให้มีความเร็วมากขึ้นและใช้กำลังคนน้อยลง โดยการนำโปรแกรมคอมพิวเตอร์เข้ามาสนับสนุนในการตรวจจับองค์ประกอบที่สำคัญเหล่านั้นด้วยการประมวลผลภาพ นอกจากนี้ยังสามารถขึ้นโครงเบื้องต้นจากแบบของลูกค้าแทนการร่างแบบขึ้นมาด้วยมือ โดยอาศัย Computer vision Library ที่เป็น open-source อย่าง OpenCV ประกอบกับขั้นตอนวิธีอ่านเส้นในภาพเพื่อระบุ และโครงสร้างหลังคาส่วนใหญ่เหมาะสมกับการใช้ rule-base detection ทั้งหมดนี้จะทำให้การเสนอราคาเบื้องต้นให้กับลูกค้าได้อย่างรวดเร็วขึ้น และสามารถแบ่งเบาภาระงานในส่วนหนึ่งได้

1.2 วัตถุประสงค์

1. เพื่อพัฒนาโปรแกรมทำหน้าที่ช่วยนับจำนวนองค์ประกอบที่สำคัญต่อการประเมินราคา
2. เพื่อพัฒนาโปรแกรมทำหน้าที่ช่วยระบุประเภทของหลังคาจากแปลนหลังคา
3. แปลงข้อมูลนำเข้าไฟล์แปลนหลังคา pdf (.pdf) ให้เป็นตัวเลขที่ใช้งานต่อได้ง่าย ในรูปแบบไฟล์ excel (.xlsx)

1.3 ขอบเขตของโครงงาน

1. โปรแกรมทำงานได้เฉพาะระบบปฏิบัติการวินโดว์
2. มีตัวหลังคาเพียงหลังเดียวในแปลน
3. ระบบแยกประเภทหลังคาทรงปั้นหยา ทรงจั่ว และทรงเหยียบสิงห์เท่านั้น
4. นับจำนวนองค์ประกอบสำคัญคือ ตะเข้สัน ตะเข้รางเท่านั้น
5. สามารถมีชนิดหลังคาย่อยหลายแบบในแปลนเดียว
6. แบบแปลนที่รองรับต้องไม่มีเส้นบอกระยะ หรือเส้นอื่นเส้นใดขีดทับหรือกีดขวางตัวโครงหลังคา
7. ข้อมูลนำเข้าเป็นไฟล์ .pdf เท่านั้น

1.4 ขั้นตอนการพัฒนา

การพัฒนาอัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา มีขั้นตอนการดำเนินการดังต่อไปนี้

1.รวบรวมความต้องการ และข้อมูลที่จำเป็น

- 1.1 ทำการสัมภาษณ์และสอบถามจากพนักงานบริษัทที่ทำงานในส่วนนี้โดยตรง
- 1.2 ถามหาลักษณะเฉพาะและข้อมูลที่เกี่ยวข้องของหลังคาแบบต่าง ๆ
- 1.3 คุณลักษณะที่ต้องการดึงออกมาจากแปลน
- 1.4 รวบรวมตัวอย่างแปลนหลังคา

2.ออกแบบและวางแผนการทำงาน

- 2.1 คาดคะเนเวลาที่ต้องใช้ในการทำโครงงาน เพื่อนำมาวางแผน
- 2.2 หาอัลกอริธึมที่จะใช้ดึงแปลนออกมาจากข้อมูลนำเข้า
- 2.3 หาอัลกอริธึมที่สามารถใช้ในการระบุตำแหน่งมุมของทรงปิด
- 2.4 ร่างโครงสร้างการทำงานของโปรแกรม

3.พัฒนาโปรแกรม

- 3.1 สร้างอัลกอริธึมระบุตำแหน่งทรงสำคัญในแปลน
- 3.2 สร้างอัลกอริธึมระบุประเภทของหลังคา
- 3.3 สร้างอัลกอริธึมระบุจำนวนส่วนประกอบที่สำคัญอื่น ๆ
- 3.4 สร้างส่วนประสานกับผู้ใช้

4.ทดสอบ

4.1 ทำสถิติ Accuracy และ recall ของโปรแกรม

4.2 ตรวจสอบการทำงานของอัลกอริธึมต่าง ๆ ที่ต้องปรับปรุง

4.3 รวบรวมจุดที่ต้องแก้ไขเพื่อนำกลับมาพัฒนาต่อไป

4.4 ตรวจสอบการแสดงผลของส่วนประสานผู้ใช้

5.นำเสนอให้กับผู้ใช้งาน

5.1 สาธิตการใช้งานโปรแกรมให้กับผู้ใช้

5.2 รวบรวมข้อเสนอแนะของผู้ใช้

5.3 เก็บความต้องการที่เกิดขึ้นใหม่

Task	ธันวาคม 2563				มกราคม 2564				กุมภาพันธ์ 2564				มีนาคม 2564			
	1-7	8-15	16-22	23-31	1-7	8-15	16-22	23-31	1-7	8-15	16-22	23-28	1-7	8-15	16-22	23-31
สัมภาษณ์,สอบถามจากพนักงานที่ทำงานในส่วนนี้โดยตรง	✓	✓														
ถามลักษณะเฉพาะ,ข้อมูลที่เกี่ยวข้องของหลังคาแบบต่าง ๆ	✓	✓														
คุณลักษณะที่ต้องการดึงออกมาจากแปลน		✓														
รวบรวมตัวอย่างแปลนหลังคา		✓														
คาดคะเนเวลาที่ต้องใช้ในการทำโครงงานเพื่อนำมาวางแผน		✓														
หาอัลกอริธึมที่จะใช้ดึงแปลนออกมาจากข้อมูลนำเข้า			✓	✓						✓						
หาอัลกอริธึมที่สามารถใช้ในการระบุตำแหน่งมุมของทรงปิด			✓	✓						✓						
ร่างโครงสร้างการทำงานของโปรแกรม				✓												
พัฒนาอัลกอริธึมระบุตำแหน่งทรงสำคัญในแปลน				✓	✓	✓				✓	✓				✓	
พัฒนาอัลกอริธึมระบุประเภทของหลังคา				✓	✓	✓				✓	✓				✓	
พัฒนาอัลกอริธึมระบุจำนวนส่วนประกอบที่สำคัญอื่น ๆ						✓	✓			✓					✓	
พัฒนาส่วนประสานกับผู้ใช้								✓	✓			✓				
ทำสถิติ Accuracy และ recall ของโปรแกรม							✓						✓		✓	
ตรวจสอบการทำงานของอัลกอริธึมต่างๆที่ต้องปรับปรุง						✓			✓			✓	✓			
รวบรวมจุดที่ต้องแก้ไขเพื่อนำกลับมาพัฒนาต่อไป						✓			✓							
ตรวจสอบการแสดงผลของส่วนประสานผู้ใช้								✓	✓			✓				
นำเสนอและสาธิตการใช้งานโปรแกรมให้กับผู้ใช้									✓					✓		✓
รวบรวมข้อเสนอแนะของผู้ใช้									✓					✓		✓
เก็บความต้องการที่เกิดขึ้นใหม่									✓					✓		

ภาพที่ 1 Action plan ประกอบแผนการดำเนินงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ก. ในด้านความรู้และประสบการณ์ต่อตัวนิสิตเอง

- ได้รับความรู้และประสบการณ์ด้านการประมวลผลภาพด้วย OpenCV
- ได้รับประสบการณ์ในการบริหารโครงงานด้วยตนเอง
- ประสบการณ์ในการทำงานร่วมกับบริษัทเอกชน

ข. ความรู้ ความเข้าใจที่นำไปสู่การแก้ไขปัญหาของสังคมหรือสภาพแวดล้อม

- บริษัทหลังคาได้โปรแกรมที่นำไปใช้ต่อให้เกิดประโยชน์ได้
- พัฒนากระบวนการทำงานของการเสนอราคาหลังคาบ้านให้มีประสิทธิภาพยิ่งขึ้น สามารถตอบรับความต้องการของลูกค้าได้อย่างรวดเร็ว ทำให้การดำเนินธุรกิจหลังคาคล่องตัวยิ่งขึ้น

1.6 โครงสร้างของรายงาน

บทที่ 2 จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับอัลกอริธึมวิเคราะห์แปลนหลังคา

บทที่ 3 จะกล่าวถึงวิธีการดำเนินงาน และการทำงานของอัลกอริธึม ซึ่งจะประกอบไปด้วย

- สถาปัตยกรรมของโปรแกรม
- ขั้นตอนการทำงาน และวิธีการทำงานในแต่ละขั้น
- ซอร์สโค้ดในแต่ละส่วนของโปรแกรมพร้อมคำอธิบาย

บทที่ 4 จะกล่าวถึงผลลัพธ์ของโครงงาน

บทที่ 5 จะกล่าวถึงข้อสรุป และข้อเสนอแนะ

บทที่ 2
งานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับอัลกอริธึมในการหาขอบของทรงปิดใด ๆ ในรูปภาพดิจิตอล[1] การระบุประเภทหลังคาจากภาพถ่ายทางอากาศด้วยโครงข่ายประสาทเทียม[2] และอัลกอริธึมในการตรวจจับหลังคาจากภาพทางอากาศขาวดำความละเอียดสูง[3]

2.1 Topological structural analysis of digitized binary images by border following

งานวิจัยนี้แสดงถึงวิธีการวิเคราะห์รูปทรงของรูปขาวดำจากการไล่ตามขอบรูปซึ่งเป็นส่วนเสริมของอัลกอริธึมที่แบ่งแยกระหว่างขอบนอกและขอบในของรูป ซึ่งตัว OpenCV ได้นำอัลกอริธึม [1] นี้มาใช้ใน findContour ซึ่งนำมาประยุกต์เพื่อหาขอบของรูปทรงปิด ซึ่งในที่นี้คือหลังคาได้

ที่มา

Satoshi Suzuki and others, “Topological structural analysis of digitized binary images by border following”, Computer Vision, Graphics, and Image Processing, 30(1):32–46, 1985.

2.2 Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds from Aerial Imagery

ในงานวิจัย [2] ที่เกี่ยวกับการระบุประเภทหลังคานี้ ผู้วิจัยได้ใช้โครงข่ายประสาทเทียมในการระบุและแยกแยะประเภทของหลังคา โดยมีข้อมูลนำเข้าเป็นภาพทางอากาศความละเอียดต่ำที่นำมาต่อกัน โดยผลลัพธ์พบว่าสามารถระบุประเภทของหลังคาได้แม่นยำถึงร้อยละ 96.65 และนอกจากประเภทของหลังคาแล้ว ยังสามารถประมาณความสูงของเสาอากาศและปล่องควันที่อยู่บนหลังคาบ้านได้อีกด้วย แต่ในกรณีของการอ่านจากแปลนหลังคาซึ่งได้นำมาใช้ในโครงงานนี้ ไม่มีปล่องควันและเสาอากาศเป็นส่วนหนึ่งขององค์ประกอบ และไม่ได้พิจารณาเรื่องความสูงของหลังคาด้วย

ที่มา

M. Axelsson, U. Soderman, A. Berg and T. Lithen, "Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds From Aerial Imagery," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 1293-1297, doi: 10.1109/ICASSP.2018.8461740.

2.3 A Fast Level Set Algorithm for Building Roof Recognition From High Spatial Resolution Panchromatic Images

งานวิจัย [3] นี้ได้เสนออัลกอริธึมในการตรวจจับหลังคาสิ่งปลูกสร้างจากภาพขนาดใหญ่ ซึ่งสามารถทำได้ในประสิทธิภาพที่สูง และมุ่งเน้นไปที่การเพิ่มประสิทธิภาพในการใช้พลังคำนวณ แต่ไม่ทำให้ผลลัพธ์ที่ได้มีความแม่นยำด้อยลง โดยการอาศัยเทคนิคเช่น Gaussian Kernel ซึ่งสุดท้ายแล้วประสิทธิภาพที่ได้มานี้ เหมาะกับการนำไปปรับใช้ในการตรวจจับและดึงวัตถุอย่างพาหนะหรือถนนได้อีกด้วย

ที่มา

Z. Li, Z. Liu and W. Shi, "A Fast Level Set Algorithm for Building Roof Recognition From High Spatial Resolution Panchromatic Images," in IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 4, pp. 743-747, April 2014, doi: 10.1109/LGRS.2013.2278342.

จากงานวิจัยที่เกี่ยวข้อง ทั้งหมดจะวิเคราะห์จากภาพถ่ายทางอากาศ ซึ่งมีความซับซ้อนกว่าการวิเคราะห์จากแปลนโดยตรง ซึ่งมีความซับซ้อนน้อยกว่า จึงต้องการพัฒนาวิธีตรวจจับแบบ rule based ขึ้นมาใหม่ ที่ใช้กำลังในการคำนวณของคอมพิวเตอร์ไม่มากนัก และไม่จำเป็นต้องพึ่งโครงข่ายประสาทเทียมและปัญญาประดิษฐ์ในการฝึกฝนโมเดล

บทที่ 3
วิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงวิธีการวิจัยอัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา สถาปัตยกรรมของโปรแกรมที่นำอัลกอริธึมนี้ไปใช้ และการทำงานของอัลกอริธึม

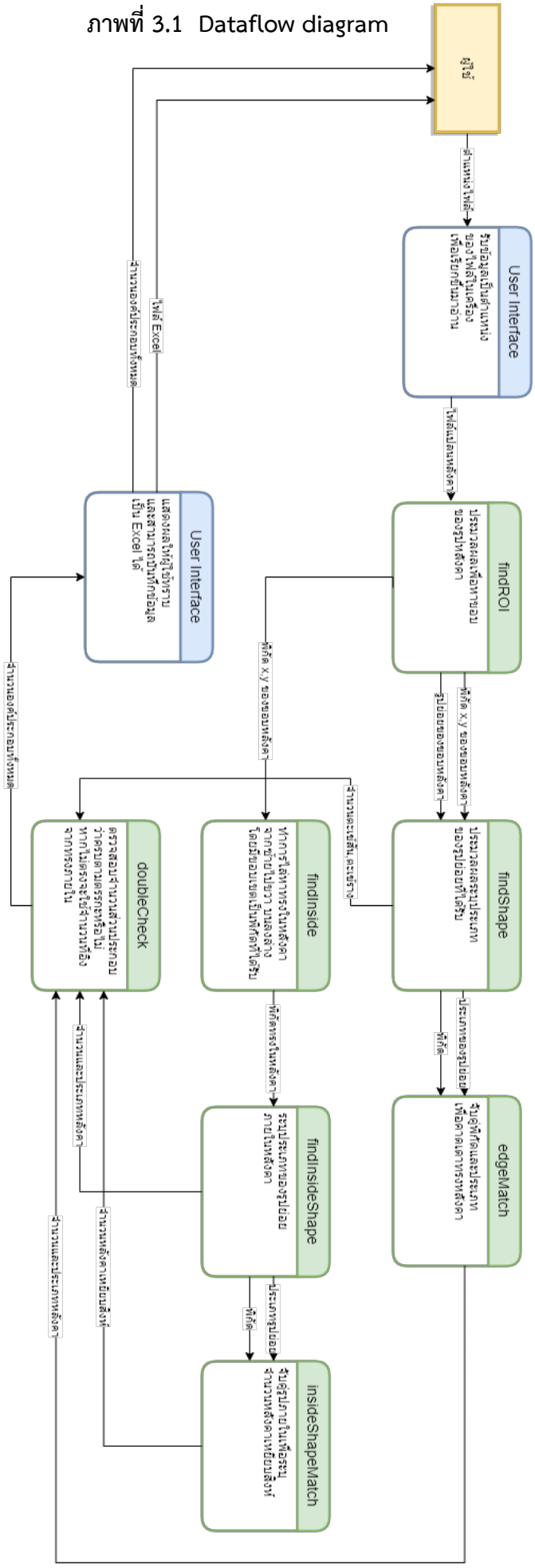
3.1 บทนำ

จากโจทย์ปัญหาที่บริษัท ปูนซีเมนต์ไทย จำกัด (มหาชน) ได้ให้มาว่า จะสามารถพัฒนากระบวนการประเมินราคาก่อสร้างให้ลูกค้าได้เร็วขึ้นได้อย่างไรนั้น พบว่า หากเป็นการนับจำนวนตะเข้สัน ตะเข้ราง และจำนวนหน้าหลังคา และประเภทหลังคา จะสามารถใช้ซอฟต์แวร์เข้ามาช่วยนับจำนวน เพื่อประมาณราคาอย่างคร่าว ๆ ได้รวดเร็วขึ้น และหลังจากการรวบรวมข้อมูลเฉพาะต่าง ๆ เกี่ยวกับหลังคา และได้เห็นรูปแบบของแปลนหลังคาจำนวนหนึ่งที่บริษัทให้มา จึงสามารถสร้างอัลกอริธึมที่ใช้ตรวจจับและตรวจสอบหลังคาเบื้องต้นได้

3.2 สถาปัตยกรรมของโปรแกรม

ตัวโปรแกรม ได้ออกแบบมาในรูปแบบ Dataflow ตามแผนภาพดังนี้

ภาพที่ 3.1 Dataflow diagram



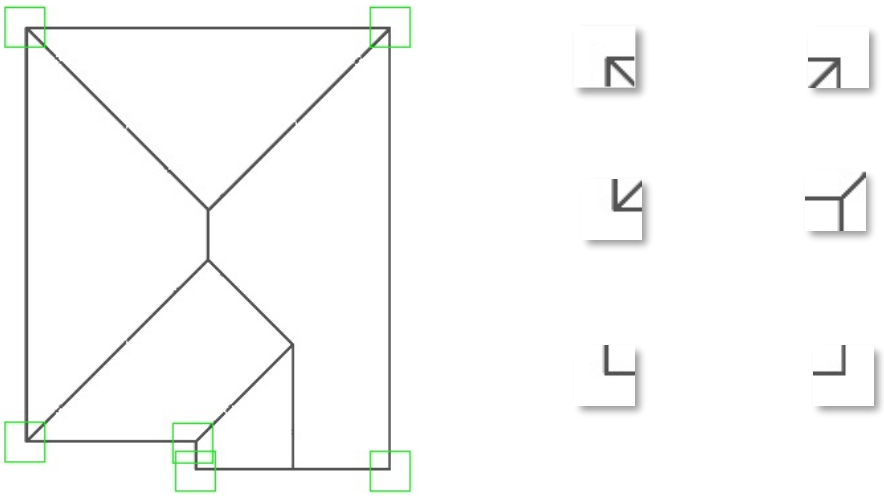
จากแผนภาพจะเห็นว่า ข้อมูลนำเข้าในรูปของที่อยู่ไฟล์ จะถูกผู้ใช้นำเข้าผ่าน user interface และโปรแกรมจะอ่านไฟล์แปลนหลังคาตามที่อยู่ และนำเข้าสู่โปรแกรมก่อนที่จะไปหาบริเวณที่สนใจในภาพ (ROI) จากขอบหลังคา ก่อนจะนำตำแหน่งที่ได้ไปประมวลหาประเภทภาพย่อยทั้งส่วนขอบและภายในตัวหลังคา ก่อนที่จะประมวลผลรวมข้อมูลเข้าด้วยกัน และแสดงผลลัพธ์ จำนวน และระบุตำแหน่งที่ตรวจจับได้ออกมาทางส่วนประสานผู้ใช้ และสุดท้ายจะบันทึกผลลัพธ์ทั้งหมดในรูปแบบไฟล์ excel

3.3 การทำงานของอัลกอริธึม

การทำงานจะสามารถแบ่งออกเป็นส่วนย่อยได้ 4 ส่วน คือ การหาขอบของหลังคา การระบุประเภทของภาพย่อย การจับคู่ประเภทของภาพย่อยเพื่อระบุหลังคา และการหาทรงหลังคาภายในหลังคา ซึ่งจะอธิบายการทำงานโดยละเอียดต่อไป

3.3.1 การหาขอบของหลังคา

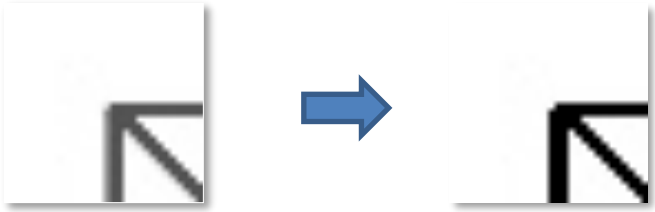
ใช้ฟังก์ชั่น findContour ของ OpenCV เพื่อทำการหาขอบ โดยมีอินพุตเป็นภาพหลังคาทั้งแปลน และมีเอาต์พุตคือตำแหน่ง x,y ของขอบหลังคาทั้งหมด จากนั้นจะนำตำแหน่งที่ได้เป็นศูนย์กลางเพื่อทำการครอบตัดภาพเป็นภาพย่อยจตุรัสขนาด 30 × 30 พิกเซล ก่อนจะส่งข้อมูลพิกัดพร้อมภาพย่อยนี้ไปทำการระบุประเภทต่อไป



ภาพที่ 3.2 ตัวอย่างการครอบตัดภาพย่อยขอบหลังคา

3.3.2 การระบุประเภทของภาพย่อย

ในการระบุประเภทของภาพย่อยนั้น จะต้องหาจำนวนเส้นและตำแหน่งของเส้น (บน,ล่าง,ซ้าย,ขวา) ในภาพ ส่วนเส้นเฉียง 45 องศาจะถูกนำไปหมุนเพื่อให้เป็นเส้นตรงและทำการหาตำแหน่งในลักษณะเดียวกัน โดยเริ่มจากการนำภาพย่อยไปทำ threshold เพื่อให้กลายเป็นภาพ Grayscale แบบไบนารี่ โดยตั้งค่าไว้ที่ 160 หากพิกเซลใดมีค่าเกิน 160 (สเกลระหว่าง 0-255) จะถูกปรับเป็นสีขาว (255) และพิกเซลที่มีค่าน้อยกว่าหรือเท่ากับ 160 จะถูกปรับเป็นสีดำ (0)



ภาพที่ 3.3 ตัวอย่างของภาพย่อยที่ผ่านการทำ threshold

เมื่อผ่านการทำ threshold แล้ว จะทำการไล่อ่านคอลัมน์พิกเซลจากซ้ายไปขวา หากพบพิกเซลสีดำจะทำการบันทึกพิกัดไว้ และนับจำนวน หากจำนวนที่นับได้มากกว่า ความกว้างของภาพหารด้วยตัวแปร sens แล้ว (มีขอบเขตเปิดระหว่าง 1 ถึงความกว้างของรูป หากค่าของตัวแปรนี้ยิ่งเข้าใกล้ 1 จะทำให้วัดผลว่ามีเส้นยากขึ้น) จะนับว่ามีเส้นในแถวของพิกเซลนั้น และหาตำแหน่งของเส้นโดยการเฉลี่ยจากค่าพิกัดพิกเซลดำในแถว บวกรวมกันหารด้วยจำนวนพิกเซลดำ หากมีค่าน้อยกว่าครึ่งหนึ่งของความกว้างภาพ จะนับว่าเป็นเส้นที่อยู่ด้านซ้าย แต่ถ้ามีค่ามากกว่า จะนับเป็นเส้นที่อยู่ทางขวา เมื่อเสร็จสิ้นจะนับจำนวนเส้นที่พบเก็บไว้ และเริ่มใหม่ในพิกเซลแถวถัดไป หากแถวถัดไปพบว่ามีเส้นเช่นกัน จะนับว่าเส้นที่พบติดกันนี้เป็นเส้นเดียวกัน และจะนับว่ามีเส้นเพิ่มขึ้น เมื่อแถวก่อนหน้าไม่พบว่ามีเส้น

0,0	1,0	2,0	3,0	4,0
0,1	1,1	2,1	3,1	4,1
0,2	1,2	2,2	3,2	4,2
0,3	1,3	2,3	3,3	4,3
0,4	1,4	2,4	3,4	4,4

ตารางที่ 3.1 ตัวอย่างตารางพิกเซล 5 x 5

ยกตัวอย่างประกอบคำอธิบายในตาราง 5×5 ข้างต้น กำหนดให้ sens = 2

หาก จำนวนจุดดำ > $\frac{\text{ความกว้างภาพย่อย}}{sens}$ คือ พบเส้น

ในแถวที่ 0 จะพบว่ามีจุดดำ 0 ช่อง = 0 < $\frac{5}{2}$ ไม่พบเส้น

ในแถวที่ 1 จะพบว่ามีจุดดำ 3 ช่อง = 3 > $\frac{5}{2}$ พบเส้น

หาตำแหน่งเส้น $\frac{2+3+4}{3}$ = 3 > $\frac{5}{2}$ เป็นเส้นทางขวา (จำนวนเส้นรวม = 1)

ในแถวที่ 2 จะพบว่ามีเส้นทางขวา แต่แถวที่ 1 ซึ่งเป็นแถวก่อนหน้าก็พบเส้นทางขวาเช่นกัน จึงไม่นับจำนวนเส้นรวมเพิ่ม

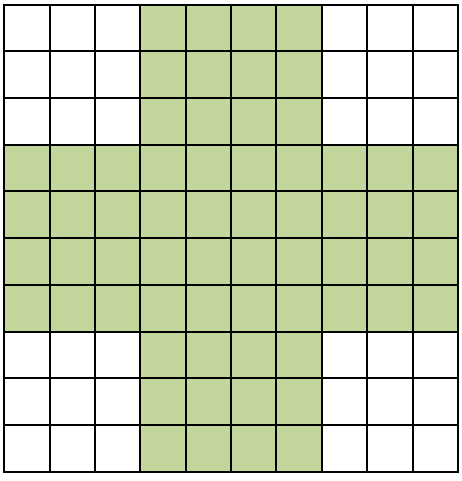
ในแถวที่ 3 จะพบว่ามีจุดดำ 1 ช่อง = 1 < $\frac{5}{2}$ ไม่พบเส้น

ในแถวที่ 4 จะพบว่ามีจุดดำ 3 ช่อง = 3 > $\frac{5}{2}$ พบเส้น

หาตำแหน่งเส้น $\frac{0+1+2}{3}$ = 1 < $\frac{5}{2}$ เป็นเส้นทางซ้าย (จำนวนเส้นรวม = 2)

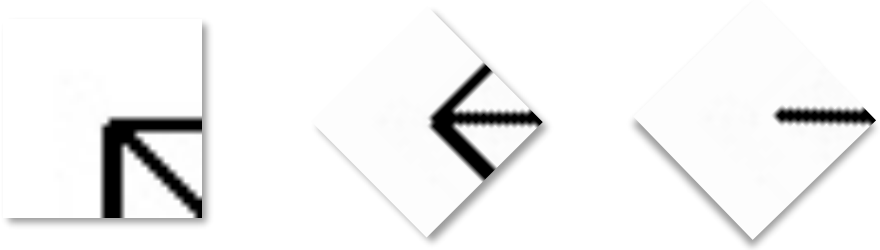
สรุปได้ว่า มีเส้นในภาพ 2 เส้น (ซ้าย 1 ,ขวา 1)

กระทำการในทำนองเดียวกันอีกครั้งในทิศทางบนลงล่างจะได้จำนวนเส้นในตำแหน่งบนและตำแหน่งล่าง จากนั้นจะทำการลบเส้นที่ถูกนับแล้วออกจากภาพโดยใช้พิกัดที่เคยบันทึกไว้ แล้วหมุนภาพ 45 องศา และทำการตรวจจับเส้นในแนวตั้งและแนวนอนอีกครั้ง



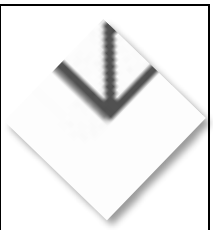
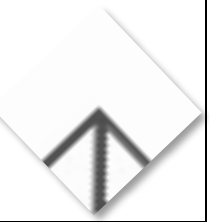
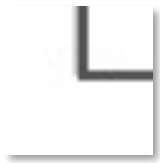
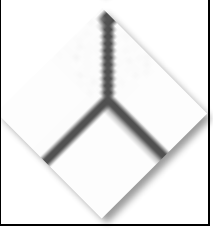
ตารางที่ 3.2 ตารางตัวอย่างบริเวณที่นับจำนวนพิกเซลดำขนาด 10x10

ในการทำงานจริงบริเวณที่ทำการนับจุดจะเป็นรูปบวก (+) ในตำแหน่งกึ่งกลางภาพ(บริเวณสีเขียวอ่อน) ซึ่งมีความกว้างทั้งสองฝั่งเป็น 4 ใน 10 ของความยาวกรอบย่อย ซึ่งเหตุผลที่ใช้ฟิลเตอร์เป็นรูปบวกนั้น เพราะสามารถช่วยลดจำนวนครั้งในการคำนวณได้ประมาณร้อยละ 36 และลดโอกาสที่จะนับเส้นใกล้เคียงที่ติดเข้ามาเนื่องจากตำแหน่งอยู่ใกล้กันเกินไป



ภาพที่ 3.4 ตัวอย่างของภาพย่อยที่ผ่านการหมุนและลบเส้นที่นับแล้วออก

จากในภาพที่ 3.4 ผลลัพธ์ของรอบก่อนหมุนคือ (ซ้าย:0 ,ขวา:1, บน:0, ล่าง:1)
และผลลัพธ์ในรอบหลังหมุน 45 องศาแล้วคือ (ซ้าย:0 ,ขวา:1, บน:0, ล่าง:0)

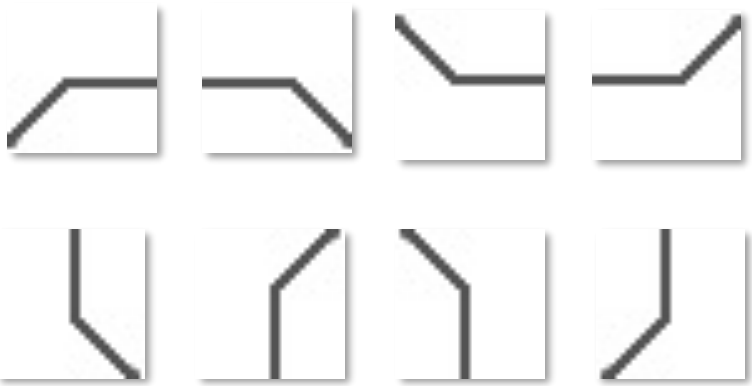
ก่อนหมุน (ซ้าย 0, ขวา 1, บน 1, ล่าง 0) หลังหมุน (ซ้าย 0, ขวา 0, บน 1, ล่าง 0)		
ก่อนหมุน (ซ้าย 1, ขวา 0, บน 0, ล่าง 1) หลังหมุน (ซ้าย 0, ขวา 0, บน 0, ล่าง 1)		
ก่อนหมุน (ซ้าย 0, ขวา 1, บน 1, ล่าง 0) หลังหมุน (ซ้าย 0, ขวา 0, บน 0, ล่าง 0)		
ก่อนหมุน (ซ้าย 1, ขวา 0, บน 0, ล่าง 1) หลังหมุน (ซ้าย 0, ขวา 0, บน 1, ล่าง 0)		

ตารางที่ 3.3 ตัวอย่างภาพย่อยพร้อมจำนวนเส้นและตำแหน่งที่ระบุได้

เมื่อแยกเก็บจำนวนของเส้นในตำแหน่งต่าง ๆ เป็นก่อนหมุนภาพและหลังหมุนภาพแล้ว จะสามารถระบุได้ว่าเป็นภาพย่อยทุกแบบที่ใช้ในแปลนหลังคาได้ เช่น ภาพ 3.5 คือขอบหลังคาที่มีตะเข้สัน

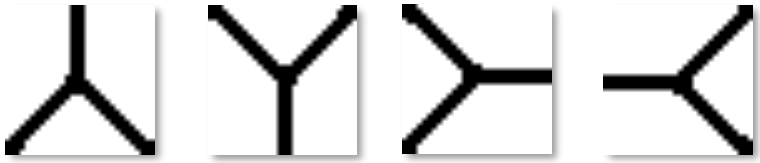
ภาพที่ 3.5 ตัวอย่างภาพย่อยซึ่งมีตะเข้สัน 4 แบบ

ภาพที่ 3.6 ตัวอย่างภาพย่อยซึ่งไม่มีมีตะเข้ 4 แบบ



ภาพที่ 3.7 ตัวอย่างภาพย่อยซึ่งเป็นองค์ประกอบของหลังคาเหยียบสิงห์ 8 แบบ

ภาพที่ 3.8 ตัวอย่างภาพย่อยซึ่งมีตะเข้ราง 4 แบบ

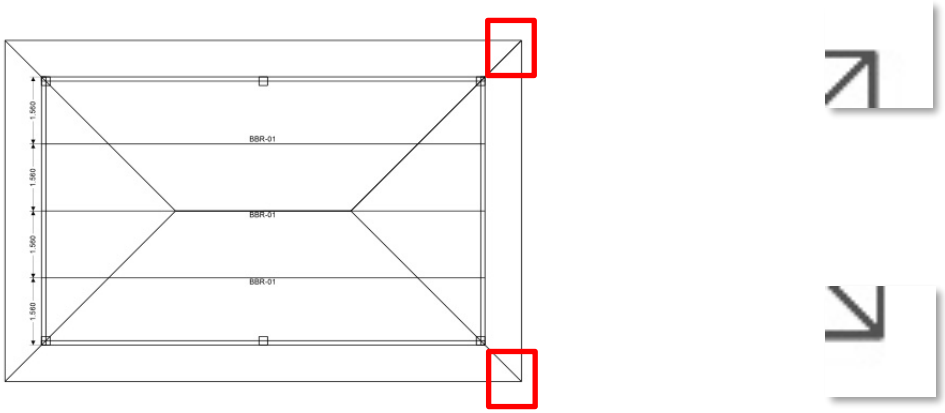


ภาพที่ 3.9 ตัวอย่างภาพย่อยซึ่งเป็นองค์ประกอบของหลังคาปั้นหยา 4 แบบ

ทั้งนี้ ประเภทของภาพย่อยที่สนใจและนำมาใช้งานในโครงงานนี้มีทั้งหมด 24 แบบ จากภาพที่ 3.5, 3.6, 3.7, 3.8 และ 3.9

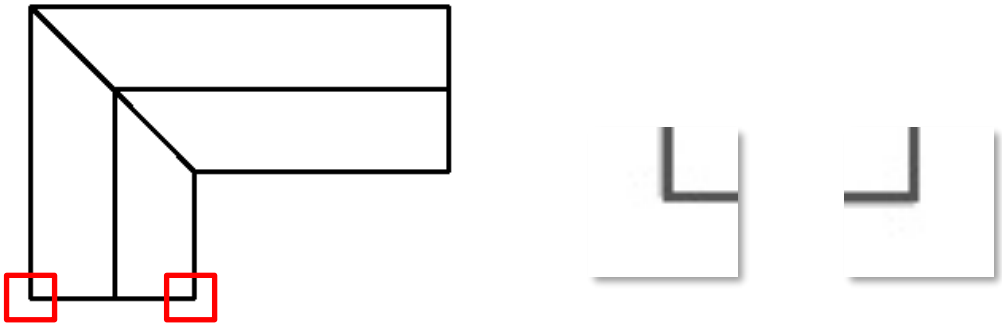
3.3.3 การจับคู่ประเภทของภาพย่อยเพื่อระบุหลังคา

จากการสังเกตประเภทหลังคาที่ต้องระบุทั้งสามประเภทคือ หลังคาปั้นหยา หลังคาจั่ว และหลังคาเหยียบสิงห์แล้ว จะพบว่าแต่ละแบบจะมีลักษณะเฉพาะที่สามารถตรวจจับได้จากวิธีในข้อ 3.3.2 ยกตัวอย่างในภาพ 3.7 จะเป็นหลังคาปั้นหยาซึ่งขอบของหลังคาจะมีลักษณะตามภาพ 3.5



ภาพที่ 3.10 แปลนหลังคาปั้นหยาที่มีเส้นรบกวน(ซ้าย) และภาพย่อยคู่มุมปั้นหยา (ขวา)

และเมื่อรู้พิกัดจากขั้นตอน 3.3.1 แล้ว ก็จะสามารถใช้ข้อมูลพิกัดที่ได้จับคู่ภาพที่มีตำแหน่งตรงข้ามกันได้ และหลังจากการระบุประเภทภาพย่อยของทั้งสองมุมดังกล่าวแล้ว ก็จะสามารถทราบทรงหลังคาเบื้องต้นได้



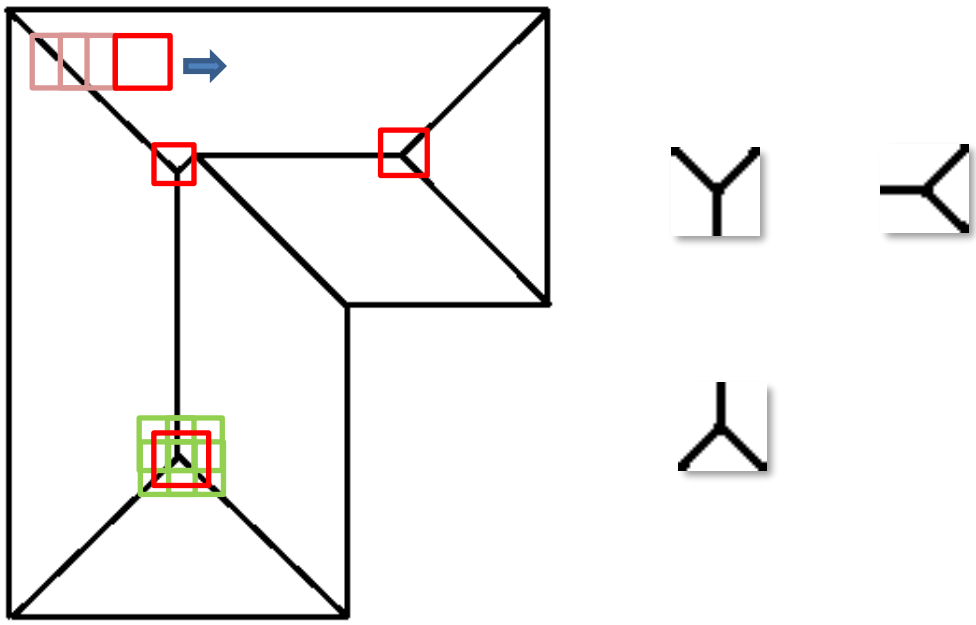
ภาพที่ 3.11 แปลนหลังคาจั่วที่ไม่มีเส้นรบกวน(ซ้าย) และภาพย่อยคู่มุมหลังคาจั่ว (ขวา)

ในกรณีหลังคาจั่ว จะมีลักษณะเฉพาะคือมุมหลังคาจะไม่มีตะเข้สันหรือตะเข้ราง และจะมีเส้นตรงลากตั้งฉากแนวหลังคาในตำแหน่งกึ่งกลางเสมอ ซึ่งสามารถคำนวณพิกัดของจุดที่ต้องตรวจสอบลักษณะนี้ได้ จากการรวมพิกัดของคู่มุมหลังคาแล้วหารสอง

สำหรับกรณีหลังคาเหยียบสิงห์และปั้นหยานั้น เบื้องต้นจะได้ข้อมูลจากมุมหลังคาว่าเป็นทรงปั้นหยา ซึ่งต้องนำไปตรวจสอบภายในรูปว่าเป็นหลังคาปั้นหยาหรือเหยียบสิงห์ ตามข้อที่ 3.3.4 ต่อไป

3.3.4 การหาหัวทรงหลังคาในหลังคา

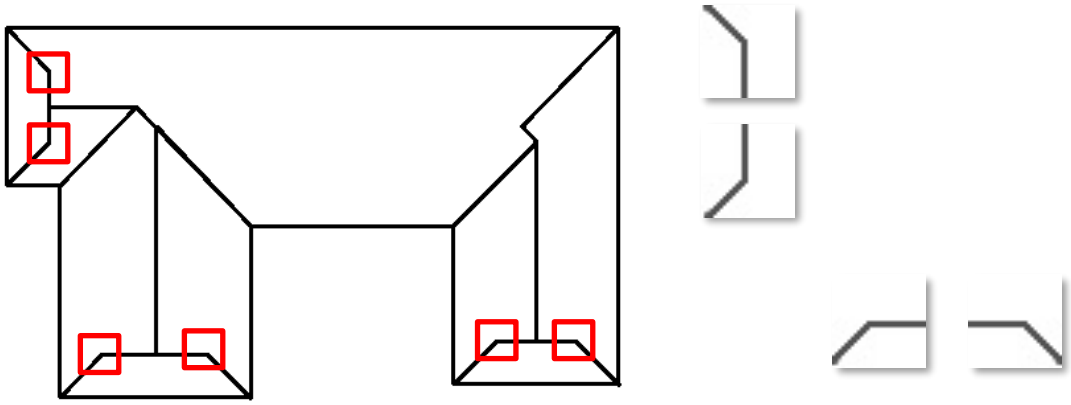
เมื่อได้พิกัดขอบหลังคาจากขั้นตอน 3.3.1 แล้ว ก็จะสามารถหาบริเวณที่สนใจในภาพได้ และเริ่มไล่หาทรงหลังคาภายใน ด้วยการสร้างกรอบขนาด 30×30 พิกเซล โดยมีตำแหน่งเริ่มต้นจากมุมบนซ้ายสุดของบริเวณสนใจ ระบุประเภทของรูปย่อยในกรอบนั้น หากตรงกับรูปแบบของทรงหลังคา จะบันทึกตำแหน่งพร้อมประเภทไว้ และจะเลื่อนตำแหน่งมาทางขวาครั้งละ 10 พิกเซล จนไปถึงด้านขวาสุดของหลังคา จากนั้นจะทำการย้ายกรอบไปฝั่งซ้ายสุดอีกครั้ง แต่ปรับพิกัดลงมาด้านล่าง 10 พิกเซล และไล่ตำแหน่งไปทางขวาเช่นเดิมจนครบทั้งพื้นที่หลังคา



ภาพที่ 3.12 หัวหลังคาปั้นหยาและพิกัดกึ่งกลาง(แดง)ที่เกิดจากการเฉลี่ยพิกัดที่พบ4จุด(เขียว)

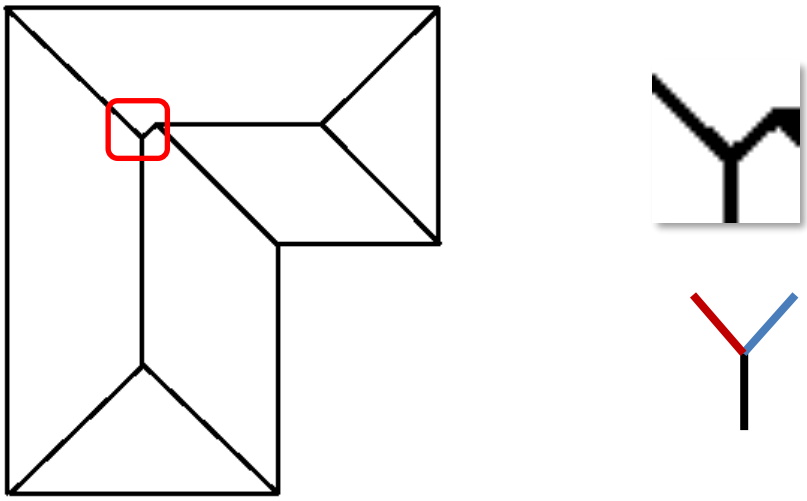
โดยทั่วไปแล้ว เมื่อพบรูปแบบหัวหลังคาในแปลนในตำแหน่งหนึ่งด้วยวิธีนี้ มักจะพบผลลัพธ์เดียวกันในบริเวณใกล้กัน เพราะระยะของกรอบเคลื่อนเพียงเล็กน้อยเพื่อเก็บรายละเอียดทุกจุดให้ได้มากที่สุด จึงต้องทำการจับกลุ่มพิกัดที่อยู่ใกล้กันเหล่านั้น และทำการเฉลี่ยพิกัดเพื่อหาพิกัดที่เป็นกึ่งกลางที่สุด และทำการบันทึกไว้

สำหรับกรณีหลังคาเหยียบสิงห์ จะพบว่ามีลักษณะเฉพาะคือ เส้นตะเข้สันที่เกิดขึ้นจากมุม จะหัก 135 องศาทั้งสองด้านมาบรรจบกัน ซึ่งสามารถใช้วิธีไล่หาแบบเดียวกันนี้เพื่อหาตำแหน่งของลักษณะดังกล่าว และจับคู่กันด้วยพิกัดเป็นหนึ่งหลังคาเหยียบสิงห์



ภาพที่ 3.13 ตัวอย่างหลังคาเหยียบสิงห์ และภาพย่อยที่ใช้ในการจับคู่เพื่อระบุทรงหลังคา

ทั้งนี้ตะเข้สันในแปลนบางแปลน ไม่ได้เชื่อมกับมุมของหลังคาเสมอไป ตามตัวอย่างในภาพ ฯ แต่จำนวนของทรงหลังคาปั้นหยาหรือหลังคาเหยียบสิงห์ 1 ทรง จะประกอบด้วยตะเข้สัน 2 ชิ้นเสมอ เมื่อจำนวนของตะเข้สันจากการวิเคราะห์ด้วยมุมหลังคาน้อยกว่าจำนวนตะเข้สันที่ได้จากการหาทรงหลังคาภายในแล้ว จะเปลี่ยนไปใช้จำนวนของตะเข้สันที่ได้จากทรงหลังคาภายในเป็นหลัก



ภาพที่ 3.14 ตะเข้สันที่ไม่เชื่อมกับมุมหลังคา(เส้นสีน้ำเงิน)และตะเข้สันที่เชื่อมกับมุม(แดง)

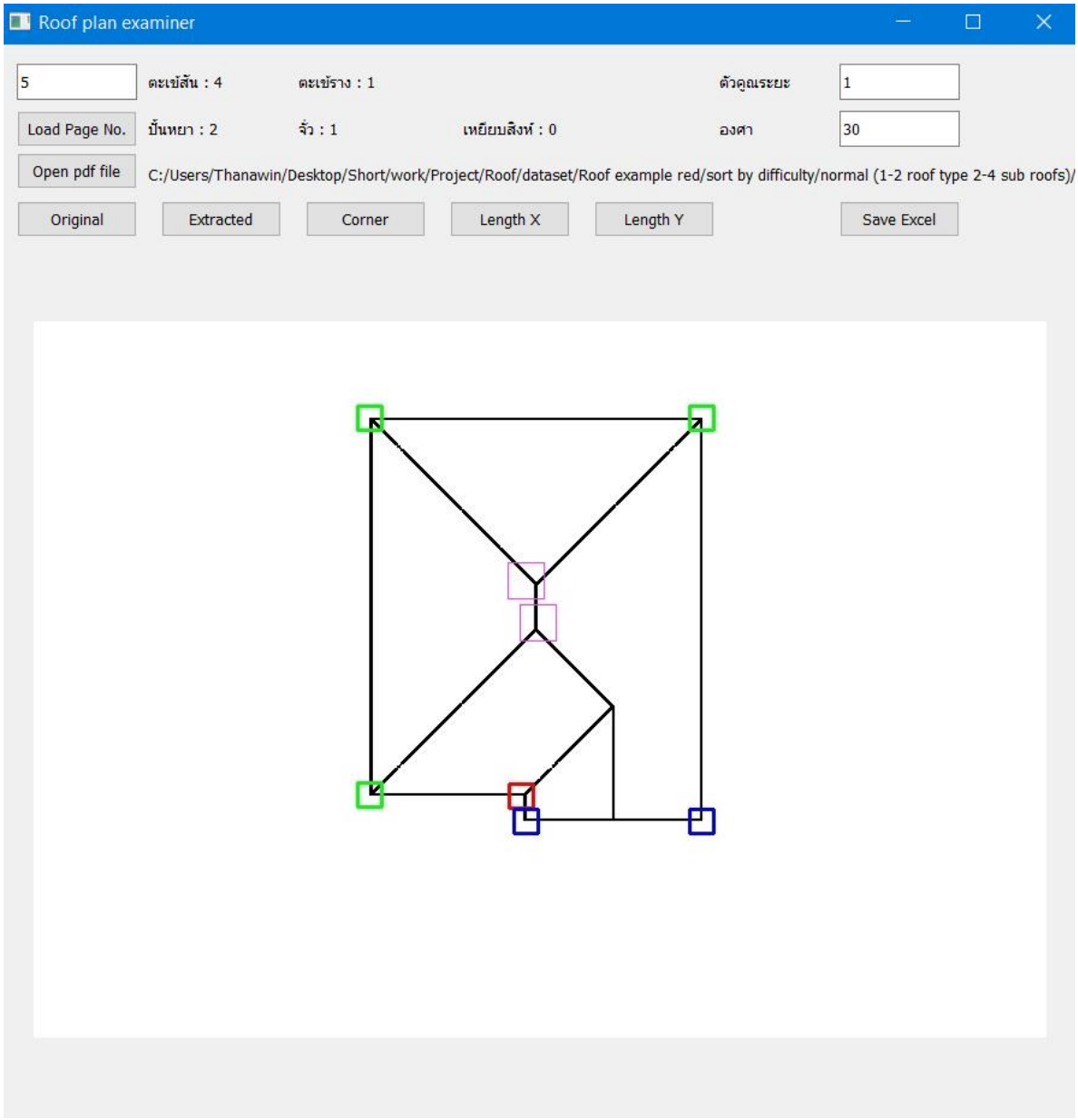
บทที่ 4
ผลการดำเนินงาน

ในบทนี้จะกล่าวถึง ผลของการดำเนินงานพัฒนาอัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา ประกอบด้วย ผลของการดำเนินการพัฒนา และผลลัพธ์การทดสอบความแม่นยำของโปรแกรมโดยแบ่งตามระดับความยากของแบบ

4.1 ผลของการดำเนินการพัฒนา

เมื่อปฏิบัติงานตามระยะเวลาของแผนการดำเนินงานในภาพที่ 1 แล้ว ได้ผลลัพธ์มาเป็นโปรแกรม Python ที่สามารถทำงานบนระบบปฏิบัติการวินโดวส์ และผ่านการทดสอบใช้งานบนคอมพิวเตอร์ของพนักงานจากบริษัท ปูนซีเมนต์ไทย จำกัด(มหาชน) โดยในด้านส่วนประสานผู้ใช้จะขับเคลื่อนด้วย Pyqt5 ใช้ไลบรารี OpenCv ในการประมวลผลภาพ และใช้ pandas-xlsxwriter ในการจัดการผลลัพธ์ที่ได้ให้อยู่ในรูปของไฟล์ excel จากภาพที่ 4.1 จะเห็นว่าผู้ใช้สามารถนำเข้าไฟล์ pdf และเลือกหน้าที่ต้องการทำการประมวลผลผ่านช่องตัวเลข ซึ่งในรูปตัวอย่างคือไฟล์ pdf หน้าที่ 5 เป็นต้น นอกจากนี้ผู้ใช้ยังสามารถดูภาพต้นฉบับและภาพที่มีการวัดความยาว จากปุ่ม Original, Extracted, Corner, Length X และ Length Y ซึ่งสามารถปรับตัวคูณในช่องตัวเลขมุมบนขวาให้ได้ตามขนาดที่ต้องการ และบันทึกไฟล์เป็น excel ต่อไป

ตัวโปรแกรมใช้เวลาทำงานระหว่าง 10 ถึง 18 วินาที ด้วยหน่วยประมวลผล Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz ซึ่งเวลาที่ใช้ประมวลผลจะเพิ่มขึ้นหากแบบมีความซับซ้อนมากขึ้น



ภาพที่ 4.1 ส่วนประสานผู้ใช้ของโปรแกรม

4.2 ผลลัพธ์การทดสอบความแม่นยำของโปรแกรม

ในส่วนของผลลัพธ์การทดสอบนั้นจะแสดงผลลัพธ์ในการหาองค์ประกอบและทรงหลังคาในแต่ละชนิดแบบภาพรวมทั้งแปลน ซึ่งจะวัดผลด้วย precision และ recall ซึ่งมีสูตรคำนวณดังนี้

$$Precision\% = \frac{True\ Positive}{True\ Positive + False\ Postive} \times 100$$

$$Recall\% = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100$$

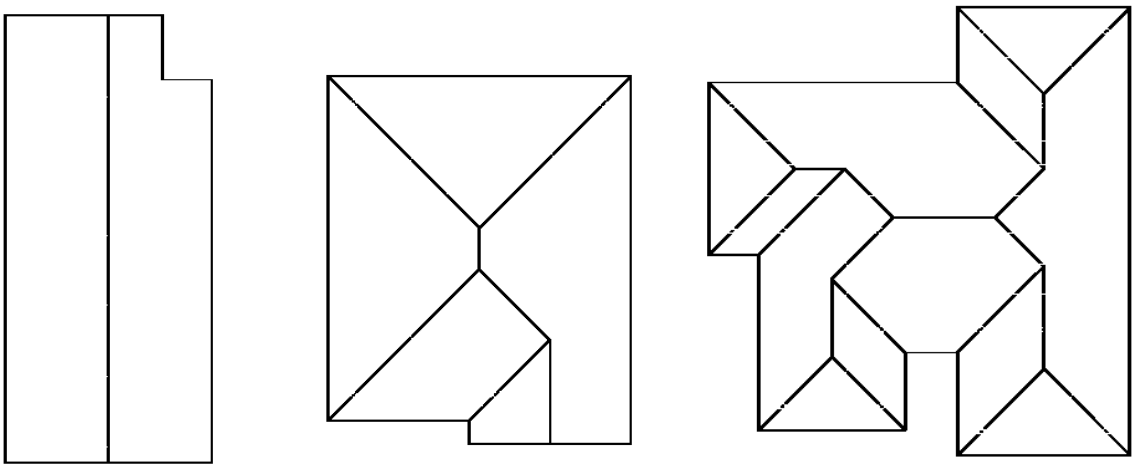
นอกจากนี้จะแสดงผลลัพธ์การตรวจจับองค์ประกอบในแต่ละประเภท ซึ่งจะวัดผลแบบไบนารี่ คือ ถูก-ผิด เท่านั้น โดยการแสดงผลลัพธ์ทั้งหมดจะจัดตามระดับความยากง่ายของแบบแปลน

ความยากของแบบจะแบ่งได้เป็น 3 ระดับ คือ ง่าย ปานกลาง และยาก โดยการพิจารณาระดับความยากจะใช้จำนวนของหลังคาย่อยและความหลากหลายของประเภทหลังคาย่อยเป็นตัวจำแนก ดังนี้

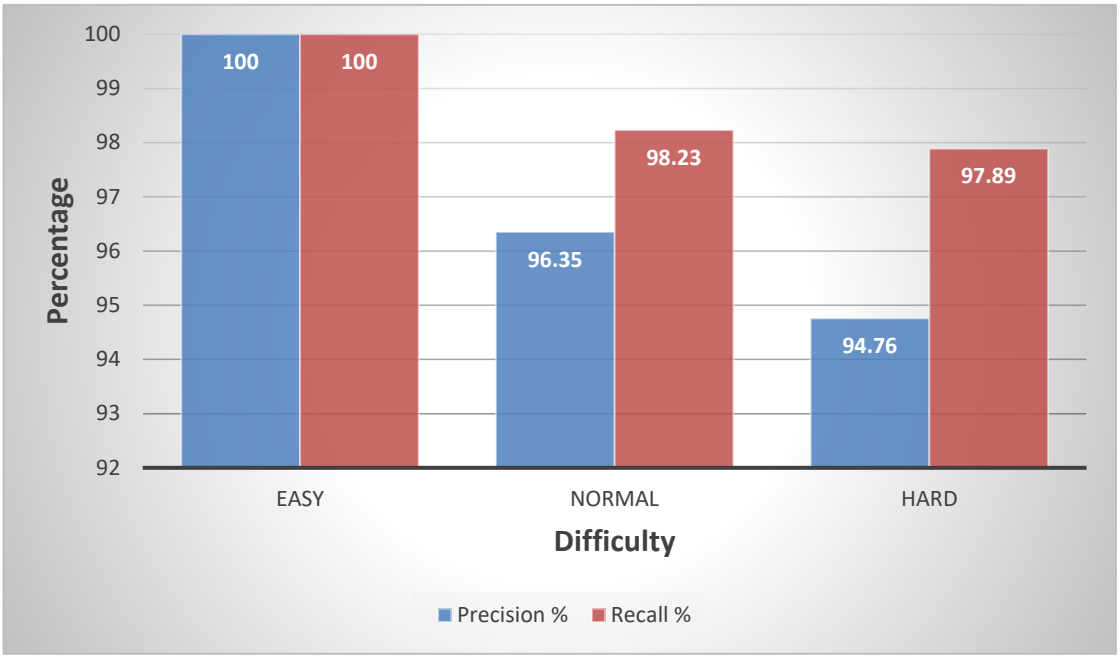
ง่าย : หลังคาย่อยในแบบมีเพียงประเภทเดียว และจำนวนหลังคาย่อยไม่เกินสองหลังคา

ปานกลาง : หลังคาย่อยในแบบมีหนึ่งถึงสองประเภท และจำนวนหลังคาย่อยมีสองถึงสี่หลังคา

ยาก : หลังคาย่อยในแบบมีหนึ่งประเภทขึ้นไป และจำนวนหลังคาย่อยมากกว่าสี่หลังคา



ภาพที่ 4.2 ตัวอย่างหลังคาเรียงตามความยาก (ง่าย-ซ้าย, ปานกลาง-กลาง, ยาก-ขวา)



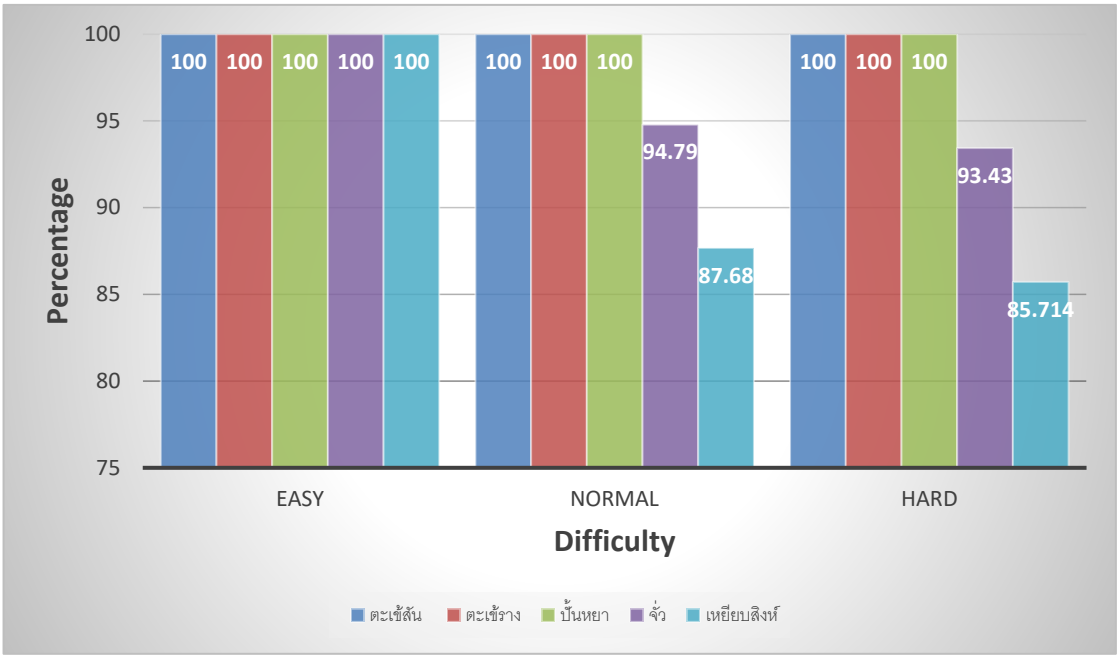
ภาพที่ 4.3 แผนภูมิแท่งของผลลัพธ์ Precision และ Recall ทั้งสามระดับความยาก

แบบง่าย มีตัวอย่างทดสอบทั้งหมด 19 แบบ
แบบปานกลาง มีตัวอย่างทดสอบทั้งหมด 36 แบบ
แบบยาก มีตัวอย่างทดสอบทั้งหมด 14 แบบ
แบบที่ใช้ทดสอบรวมทั้งหมด 69 แบบ

จากแผนภาพที่ 4.3 จะพบว่าผลลัพธ์ความแม่นยำเป็นดังนี้

- แบบง่ายได้ Precision ร้อยละ 100 และ Recall ร้อยละ 100
- แบบปานกลาง ได้ Precision ร้อยละ 96.35 และ Recall ร้อยละ 98.23
- แบบยากได้ Precision ร้อยละ 94.76 และ Recall ร้อยละ 97.89

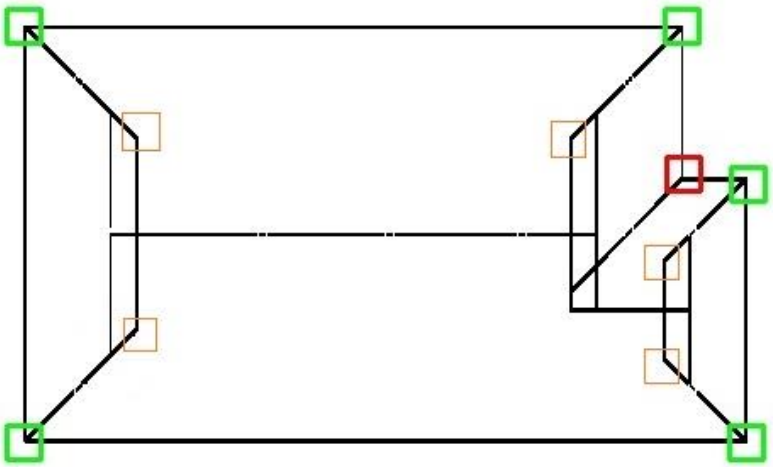
จะสังเกตเห็นได้ว่า เมื่อความยากเพิ่มขึ้น จะทำให้ Precision ลดลง แต่ Recall ยังเข้าใกล้ร้อยละ 100 เช่นเดิม แสดงว่าการตรวจพลาดส่วนใหญ่ มาจากการตรวจจับแล้วได้จำนวนมากเกินความเป็นจริง นอกจากนี้จะสังเกตเห็นว่า ในความยากระดับยากที่สุด ผลลัพธ์มีประสิทธิภาพน้อยที่สุด คาดว่าเนื่องจากจำนวนแบบหลังคาเหยียบสิงห์ และจำนวนหลังคาย่อยที่มีอยู่มาก ซึ่งทำให้เกิดข้อผิดพลาดในการตรวจจับได้ง่าย มีมากที่สุดจากทั้งสามระดับ



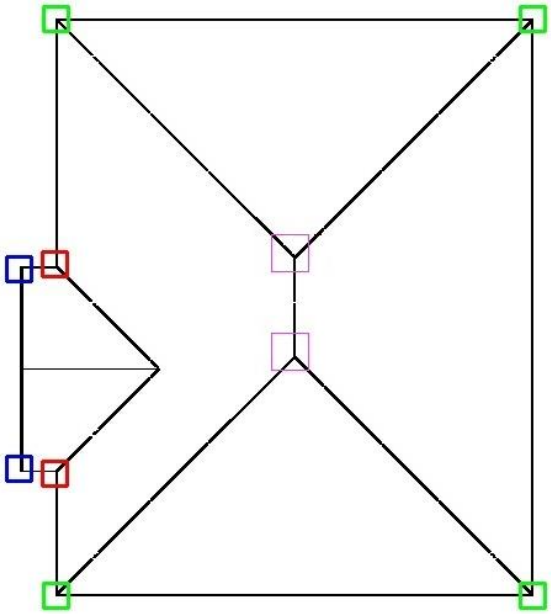
ภาพที่ 4.4 แผนภูมิแท่งของผลลัพธ์องค์ประกอบย่อย ทั้งสามระดับความยาก

จากแผนภาพที่ 4.4 จะพบว่าส่วนที่ตรวจจับผิดพลาดส่วนใหญ่จะเป็นหลังคาหลังคาเหยียบสิงห์ เนื่องจากเป็นชนิดหลังคาที่ตรวจจับยากที่สุด เพราะต้องนำภาพย่อยสองจุดมาประกอบกัน แลพหลังคาจั่วจะผิดพลาดรองลงมา แต่องค์ประกอบอื่นนั้น อัลกอริธึมสามารถทำงานได้อย่างสมบูรณ์

ความผิดพลาดในส่วนของหลังคาจั่วและหลังคาเหยียบสิงห์ในระดับปานกลาง หนึ่งในนั้นเกิดจากการตรวจจับหลังคาเหยียบสิงห์ไม่ครบ 1 แปลน และตรวจจับหลังคาจั่วและเหยียบสิงห์เกิน อีก 2 แปลน ส่วนความผิดพลาดในระดับยากเกิดจากหลังคาเหยียบสิงห์เกิน 1 แปลน



ภาพที่ 4.5 แปลนหลังคาเหยียบสิงห์ระดับปานกลางที่ตรวจจับไม่ครบ



ภาพที่ 4.6 แปลนหลังคาเหยียบปั้นหยาผสมจั่วระดับปานกลางที่ตรวจจับครบถ้วน

ตัวโปรแกรมจะแสดงตำแหน่งที่พบลักษณะของหลังคาด้วยการตีกรอบสีต่าง ๆ โดยมีความหมายดังนี้

สีเขียว: มุมหลังคาที่มีตะเข้สัน(เป็นมุมของหลังคาปั้นหยาหรือหลังคาเหยียบสิงห์)

สีน้ำเงิน: มุมหลังคาจั่วหรือหลังคาที่ไม่มีตะเข้

สีแดง: มุมหลังคาที่มีตะเข้ราง

สีม่วง: ทรงหัวหลังคาปั้นหยาภายใน

สีส้ม: ทรงขอบหลังคาเหยียบสิงห์ภายใน

จากภาพที่ 4.5 ซึ่งเป็นหนึ่งในแปลนที่หลังคาเหยียบสิงห์ตรวจจับได้น้อยกว่าความเป็นจริงนั้น จะเห็นว่ามีหลังคาเหยียบสิงห์หนึ่งทรง ที่มีทรงขอบหลังคาเหยียบสิงห์ภายในไม่ครบคู่ จึงไม่เกิดการนับว่าจุดนั้นเป็นหลังคาเหยียบสิงห์ ซึ่งเป็นผลเสียของการทำงานแบบ Rule-based กล่าวคือ ไม่ยืดหยุ่นพอที่จะรองรับทรงหลังคาที่มีการผิดเพี้ยน แต่ภาพที่ 4.6 แสดงให้เห็นว่าหากตำแหน่งและลักษณะไม่โดนทับหรือบดบัง ทั้งหลังคาจั่วและหลังคาปั้นหยาสามารถตรวจจับได้อย่างครบถ้วนและตรงตำแหน่ง แม้ว่าจะมีหลังคาหลายแบบอยู่ในแปลนเดียวกัน

หมายเหตุ:เนื่องจากจำนวนแปลนหลังคาเหยียบสิงห์ที่นำมาทดสอบมีกลุ่มตัวอย่างขนาดเล็ก จึงอาจทำให้ผลลัพธ์เฉลี่ยคาดเคลื่อนกับการนำไปใช้จริงค่อนข้างมาก

บทที่ 5
ข้อสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึง สรุปผลจากข้อมูลผลการตรวจจับองค์ประกอบของหลังคา ซึ่งประกอบด้วยตะเข้สัน และตะเข้ราง และหลังคาย่อยทรงปั้นหยา ทรงจั่ว และทรงเหยียบสิงห์ ในบทที่ 4 รวมถึงข้อเสนอแนะอื่นๆ ที่ได้จากการพัฒนาโปรแกรม

5.1 ข้อสรุป

อัลกอริธึมสามารถทำงานได้อย่างมีประสิทธิภาพสมบูรณ์ในทุกกรณีที่ยกเว้นหลังคาเหยียบสิงห์และหลังคาจั่วในบางแบบ อาจจะไม่เหมาะสมในการนำอัลกอริธึมนี้ไปใช้ เนื่องจากอาจเกิดความคาดเคลื่อนเนื่องจากจำนวนที่ตรวจจับมากกว่าความเป็นจริง

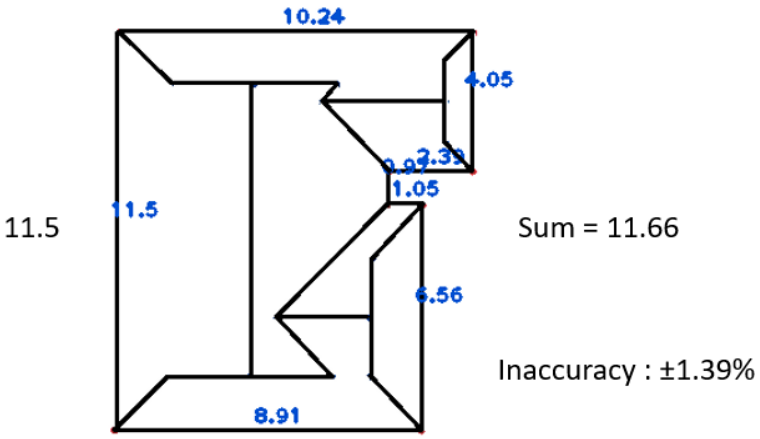
อย่างไรก็ตาม การใช้งานจริงในกรณีนี้ให้ความสำคัญกับ Recall มากกว่า เพราะหากตรวจไม่ครบทั้งหมดจะทำให้ราคาที่ประเมินน้อยกว่าความเป็นจริง ซึ่งจะทำให้บริษัทเสียรายได้

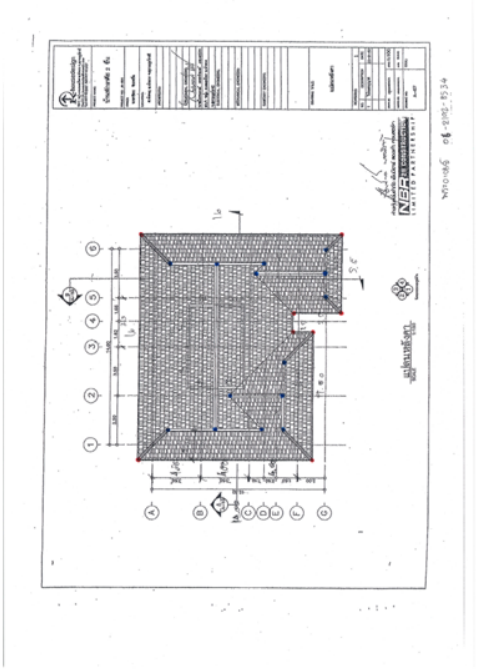
สาเหตุที่ผลลัพธ์ของหลังคาเหยียบสิงห์ยังไม่ดีพอ ทั้งนี้เป็นเพราะมีกลุ่มตัวอย่างน้อย และเป็นหลังคาที่ตรวจจับได้ซับซ้อนที่สุดจากสามประเภทหลังคา

5.2 ข้อเสนอแนะ

วิธีตรวจสอบหลังคาเหยียบสิงห์ ยังไม่ดีพอ ทั้งนี้เป็นเพราะมีกลุ่มตัวอย่างน้อย และพบไม่ได้บ่อยนัก และการตรวจจับหลังคาจั่วยังไม่รัดกุมพอ ทำให้จำนวนเกิน ซึ่งต้องหาวิธีตรวจจับที่ดีกว่านี้เพื่อลดการเกิด False Positive และเพิ่ม True Positive

การนำโปรแกรมนี้ไปใช้จริงยังทำได้ยาก เนื่องจากแบบของลูกค้ามักมีเส้นบอกระยะและเส้นอื่น ๆ กีดขวางการตรวจจับ ซึ่งหากนำไปใช้งานได้ง่ายกว่านี้ จะต้องมีวิธีช่วยให้ผู้ใช้นำแบบที่มีเส้นรบกวนมาใช้งานได้ โดยได้ทดลองให้ผู้ใช้จุดสีแต้มลงไปบนกระดาษ ก่อนจะนำมาแสกนเป็นไฟล์ภาพเพื่อจะนำจุดเหล่านั้นมาเป็นพิกัดอ้างอิงเพื่อลากเชื่อมต่อกันเป็นแบบหลังคา แต่ผลลัพธ์ที่ได้กลับพบว่าสามารถใช้ได้กับแบบที่ไม่ซับซ้อนมากเท่านั้น หรือใช้ machine learning เข้ามาเรียนรู้ว่าเส้นใดเป็นโครงหลังคา เส้นใดเป็นเส้นรบกวน และช่วยลบเส้นที่ไม่พึงประสงค์ออก ก่อนจะนำแบบที่ไม่มีเส้นรบกวนเข้าโปรแกรมนี้ต่อไป





ภาพที่ 5.1 ระยะของเส้นหลังคาที่เกิดจากการวาดเชื่อมจุด และแปลนที่มีการจุดสีลงไป

หากจะดึงข้อมูลที่เป็นประโยชน์ต่อการประเมินราคามามากกว่านี้ควรจะมีการคำนวณระยะความยาวขององค์ประกอบ ซึ่งได้ทดลองคูณค่าความยาวของหลังคาแต่ละด้านด้วยตัวแปรระยะเพื่อแปลงเลขที่ได้จากหน่วยพิกเซลเป็นเมตร พบว่ามีความคาดเตลื่อนจากแบบจริงเฉลี่ยไม่เกิน 0.2 เมตร ซึ่งสามารถนำค่านี้ไปคำนวณความยาวขององค์ประกอบต่าง ๆ รวมถึงพื้นที่หลังคา แต่พบว่า ตัวเลขที่ได้ค่อนข้างห่างจากความเป็นจริงอยู่บ้าง เนื่องจากลักษณะทางปริมาตรและพื้นที่หลายอย่างของหลังคาไม่สามารถสื่อผ่านภาพสองมิติได้อย่างครบถ้วน อาทิ หลังคาอาจมีการซ้อนทับกันในระดับความสูงที่ต่างกัน ทำให้ใช้เลขที่ได้จากแบบ ไม่ใช่พื้นที่หลังคาทั้งหมด จึงควรหาวิธีอื่นที่สามารถนำมาประยุกต์ใช้ได้อย่างมีประสิทธิภาพมากกว่านี้

รายการอ้างอิง

[1] Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32–46, 1985.

[2] M. Axelsson, U. Soderman, A. Berg and T. Lithen, "Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds From Aerial Imagery," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 1293-1297, doi: 10.1109/ICASSP.2018.8461740.

[3] Z. Li, Z. Liu and W. Shi, "A Fast Level Set Algorithm for Building Roof Recognition From High Spatial Resolution Panchromatic Images," in IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 4, pp. 743-747, April 2014, doi: 10.1109/LGRS.2013.2278342.

ภาคผนวก

ภาคผนวก ก
แบบเสนอหัวข้อโครงงาน รายวิชา 2301399 Project Proposal
ปีการศึกษา 2563

ชื่อโครงงาน (ภาษาไทย)	อัลกอริธึมสำหรับวิเคราะห์ประเภทและจำนวนขององค์ประกอบหลักและองค์ประกอบย่อยของหลังคาบ้านจากแบบแปลนหลังคา
ชื่อโครงงาน (ภาษาอังกฤษ)	Algorithm for analzing roof types and counting crucial components from roof plan
อาจารย์ที่ปรึกษา	รศ.ดร. นกุล คูหะโรจนานนท์
ผู้ดำเนินการ	1. นาย ธนวิน อาภาจรัส เลขประจำตัวนิสิต 6033627723 **สาขาวิชา วิทยาการคอมพิวเตอร์** **ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์** คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หลักการและเหตุผล

การนำแบบแปลนหลังคาบ้านมาประเมินราคาในปัจจุบัน จะต้องนำแบบที่ลูกค้าส่งมาในรูปแบบของกระดาษหรือไฟล์ ไปขึ้นโครงในโปรแกรมเขียนแบบ ด้วยการให้พนักงานอ่านจากไฟล์ และกรอกแต่ละรายละเอียดลงไปในโปรแกรมร่างแบบ จากนั้นก็จะทำการนับองค์ประกอบที่จำเป็นในการประเมินราคาด้วยกำลังคน และอาศัยจำนวนองค์ประกอบและขนาดของหลังคา เพื่อให้ทราบถึงปริมาณวัสดุที่จะต้องใช้ที่จะนำไปประมาณเป็นราคา จนได้ผลลัพธ์เป็นใบเสนอราคาที่จะนำไปให้ลูกค้าพิจารณา

จากที่กล่าวมาข้างต้น จะเห็นว่ากระบวนการนี้ใช้กำลังคนเป็นส่วนใหญ่ แต่มีบางขั้นตอนที่สามารถพัฒนาให้มีความเร็วมากขึ้นและใช้กำลังคนน้อยลง โดยการนำโปรแกรมคอมพิวเตอร์เข้ามาสนับสนุนในการตรวจจับองค์ประกอบที่สำคัญเหล่านั้นด้วยการประมวลผลภาพ นอกจากนี้ยังสามารถขึ้นโครงเบื้องต้นจากแบบของลูกค้าแทนการร่างแบบขึ้นมาด้วยมือ โดยอาศัย Computer vision Library ที่เป็น open-source อย่าง OpenCV ประกอบกับขั้นตอนวิธีอ่านเส้นในภาพเพื่อระบุ และโครงสร้างหลังคาส่วนใหญ่เหมาะสม

กับการใช้ rule-base detection ทั้งหมดนี้จะทำให้การเสนอราคาเบื้องต้นให้กับลูกค้าได้อย่างรวดเร็วขึ้น และสามารถแบ่งเบาภาระงานในส่วนหนึ่งได้

วัตถุประสงค์

เพื่อพัฒนาโปรแกรมทำหน้าที่ช่วยนับจำนวนองค์ประกอบที่สำคัญต่อการประเมินราคาหลังคาและแปลงข้อมูลนำเข้าให้อยู่ในรูปแบบที่ใช้งานต่อได้ง่าย

ขอบเขตของโครงงาน

1. โปรแกรมทำงานได้เฉพาะระบบปฏิบัติการวินโดว์
2. มีตัวหลังคาเพียงหลังเดียวในแปลน
3. รองรับหลังคาทรงปั้นหยา ทรงจั่ว และทรงเหยียบสิงห์
4. องค์ประกอบสำคัญอื่น ประกอบด้วย ตะเข้สัน และตะเข้ราง
5. สามารถมีชนิดหลังคาย่อยหลายแบบในแปลนเดียว
6. แบบแปลนที่รองรับต้องไม่มีเส้นบอกระยะ หรือเส้นอื่นเส้นใดขีดทับหรือกีดขวางตัวโครงหลังคา
7. ข้อมูลนำเข้าสามารถใช้ได้ทั้งจากภาพถ่าย ภาพแสกน หรือไฟล์ .pdf

วิธีการดำเนินงาน

1.รวบรวมความต้องการ และข้อมูลที่จำเป็น

- 1.1 ทำการสัมภาษณ์และสอบถามจากพนักงานบริษัทที่ทำงานในส่วนนี้โดยตรง
- 1.2 ถามหาลักษณะเฉพาะและข้อมูลที่เกี่ยวข้องของหลังคาแบบต่าง ๆ
- 1.3 คุณลักษณะที่ต้องการดึงออกมาจากแปลน
- 1.4 รวบรวมตัวอย่างแปลนหลังคา

2.ออกแบบและวางแผนการทำงาน

- 2.1 คาดคะเนเวลาที่ต้องใช้ในการทำโครงงาน เพื่อนำมาวางแผน
- 2.2 หาอัลกอริธึมที่จะใช้ดึงแปลนออกมาจากข้อมูลนำเข้า
- 2.3 หาอัลกอริธึมที่สามารถใช้ในการระบุตำแหน่งมุมของทรงปิด
- 2.4 ร่างโครงสร้างการทำงานของโปรแกรม

3.พัฒนาโปรแกรม

- 3.1 สร้างอัลกอริธึมระบุตำแหน่งทรงสำคัญในแปลน

3.2 สร้างอัลกอริธึมระบุประเภทของหลังคา

3.3 สร้างอัลกอริธึมระบุจำนวนส่วนประกอบที่สำคัญอื่น ๆ

3.4 สร้างส่วนประสานกับผู้ใช้

4.ทดสอบ

4.1 ทำสถิติ Accuracy และ recall ของโปรแกรม

4.2 ตรวจสอบการทำงานของอัลกอริธึมต่าง ๆ ที่ต้องปรับปรุง

4.3 รวบรวมจุดที่ต้องแก้ไขเพื่อนำกลับมาพัฒนาต่อไป

4.4 ตรวจสอบการแสดงผลของส่วนประสานผู้ใช้

5.นำเสนอให้กับผู้ใช้งาน

5.1 สาธิตการใช้งานโปรแกรมให้กับผู้ใช้

5.2 รวบรวมข้อเสนอแนะของผู้ใช้

5.3 เก็บความต้องการที่เกิดขึ้นใหม่

ประโยชน์ที่คาดว่าจะได้รับ

ค. ในด้านความรู้และประสบการณ์ต่อตัวนิสิตเอง

- ได้รับความรู้และประสบการณ์ด้านการประมวลผลภาพด้วย OpenCV
- ได้รับประสบการณ์ในการบริหารโครงงานด้วยตนเอง
- ประสบการณ์ในการทำงานร่วมกับบริษัทเอกชน

ง. ความรู้ ความเข้าใจที่นำไปสู่การแก้ไขปัญหาของสังคมหรือสภาพแวดล้อม

- บริษัทหลังคาได้โปรแกรมที่มีประสิทธิภาพไปใช้งานจริง
- พัฒนากระบวนการทำงานของการเสนอราคาหลังคาบ้านให้มีประสิทธิภาพยิ่งขึ้น
- สามารถตอบรับความต้องการของลูกค้าได้อย่างรวดเร็ว ทำให้การดำเนินธุรกิจหลังคาคล่องตัวยิ่งขึ้น

อุปกรณ์และเครื่องมือที่ใช้

คอมพิวเตอร์พกพา

Processor : Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz

RAM : 16.0 GB

System: 64-bit Operating System, x64-based processor

ซอฟต์แวร์

- Windows 10 Home Single Language
- Python
- Spyder (Anaconda) IDE

งบประมาณ

- 1 TB SSD M.2 PCIe SAMSUNG 970 EVO ราคา 4,990 บาท

เอกสารอ้างอิง

Big, A. 2019. Top 15 Roof Types, Plus Their Pros & Cons. Roof Cost Estimator[Online]. Available from: https://www.roofcostestimator.com/top-15-roof-types-and-their-pros-cons/#hip%20roof[2020, October 30]

OpenCV. 2020. Structural Analysis and Shape Descriptors. Open Source Computer Vision[Online]. Available from: https://docs.opencv.org/4.5.0/d3/dc0/group__imgproc__shape.html#gadf1ad6a0b82947fa1fe3c3d497f260e0[2020, October 12]

Jorj, X. M. and Ruikai, L. 2020. Python binding for MuPDF “a lightweight PDF and XPS viewer”. PyMuPDF[Online]. Available from: https://pypi.org/project/PyMuPDF[2020, October 31]

Riverbank Computing Limited, 2020. Python bindings for the Qt cross platform application toolkit. PyQt5[Online]. Available from: https://pypi.org/project/PyQt5[2020, October 31]

ภาคผนวก ข
คู่มือการใช้งาน

ขั้นตอนการติดตั้งโปรแกรม

1. ติดตั้ง python 3.7.8
2. ติดตั้งไฟล์โปรแกรม
3. ติดตั้ง dependencies library
4. สร้างไฟล์ shortcut เพื่อเปิดโปรแกรม

1.ติดตั้ง python 3.7.8 (ต้องเชื่อมต่ออินเทอร์เน็ต)

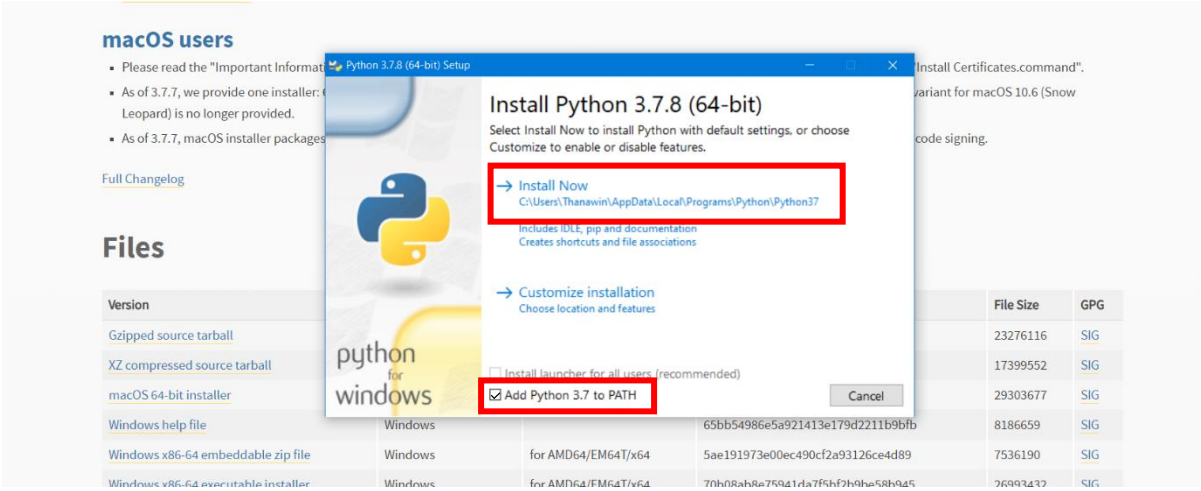
ไปที่ https://www.python.org/downloads/release/python-378/

เลือกดาวน์โหลด Windows x86-64 web-based installer

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		4d5b16e8c15be38eb0f4b8f04eb68cd0	23276116	SIG
XZ compressed source tarball	Source release		a224ef2249a18824f48fba9812f4006f	17399552	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	2819435f3144fd973d3dea4ae6969f6d	29303677	SIG
Windows help file	Windows		65bb54986e5a921413e179d2211b9bfb	8186659	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	5ae191973e00ec490cf2a93126ce4d89	7536190	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	70b08ab8e75941da7f5bf2b9be58b945	26993432	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	b07dbb998a4a0372f6923185ebb6bf3e	1363056	SIG
Windows x86 embeddable zip file	Windows		5f0f83433bd57fa55182cb8ea42d43d6	6765162	SIG
Windows x86 executable installer	Windows		4a9244c57f61e3ad2803e900a2f75d77	25974352	SIG
Windows x86 web-based installer	Windows		642e566f4817f118abc38578f3cc4e69	1324944	SIG

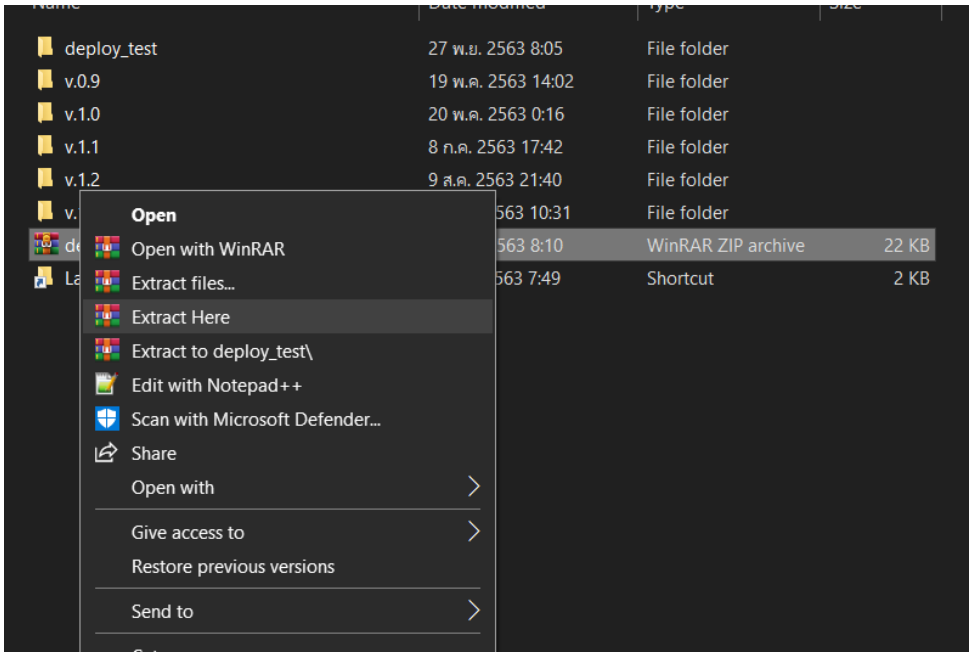
รันไฟล์ที่ดาวน์โหลดมา คลิกกล่อง Add Python 3.7 to PATH และเลือก Install Now



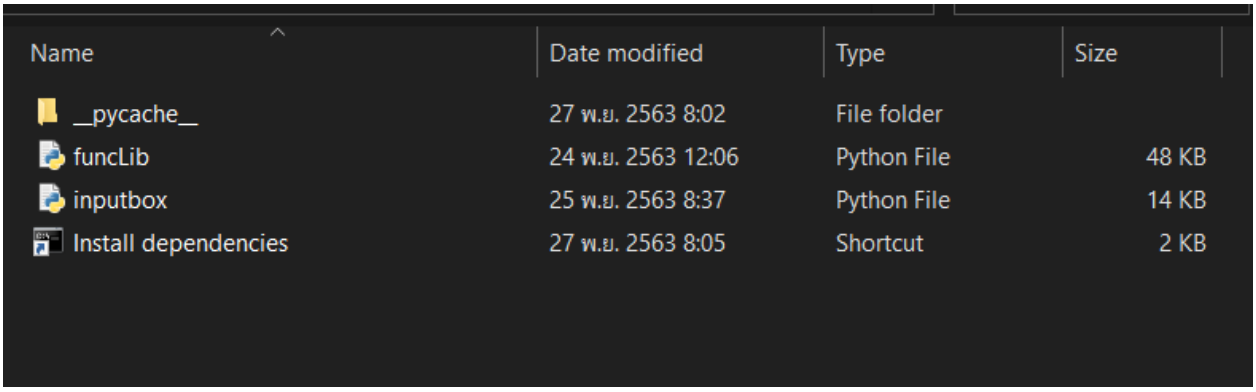
รอจนการติดตั้ง python เสร็จสิ้น

2.ติดตั้งไฟล์โปรแกรม

ทำการแตกไฟล์ .zip ไว้ที่ใดก็ได้ในคอมพิวเตอร์



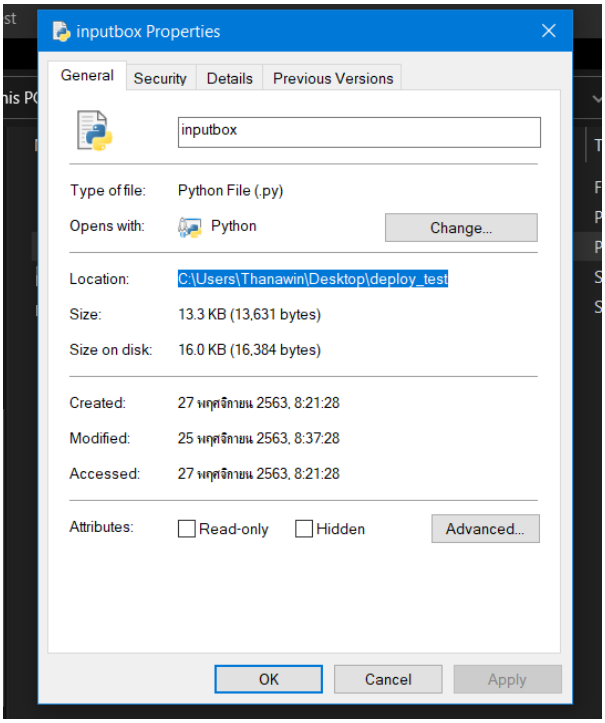
3.ติดตั้ง dependencies library (ต้องเชื่อมต่ออินเทอร์เน็ต)



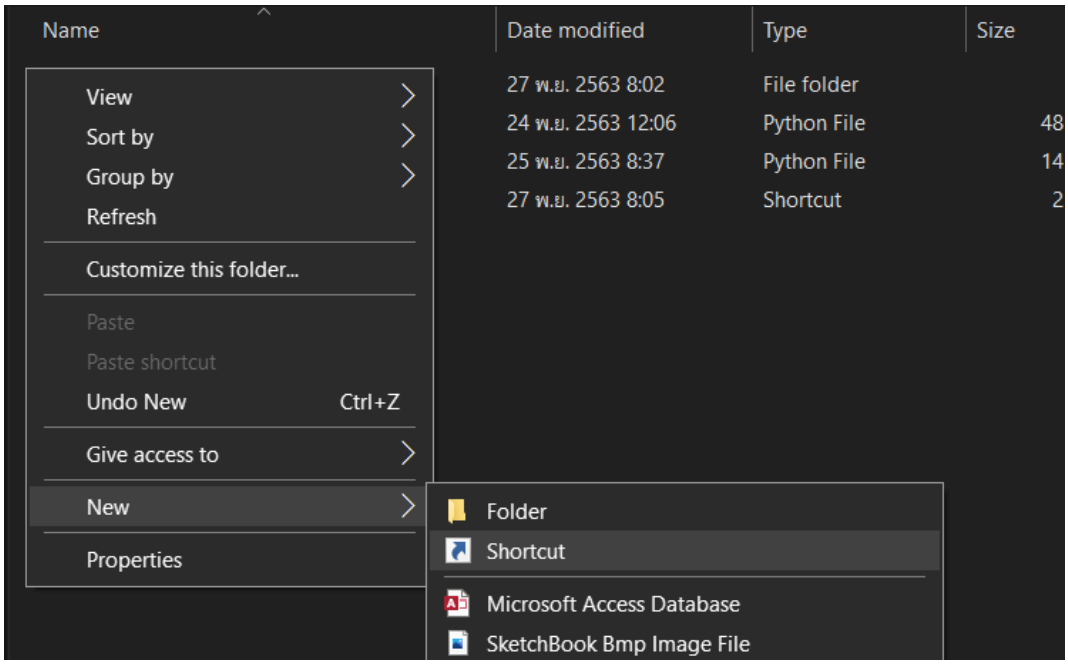
- เข้าไปในโฟลเดอร์ที่แตกไฟล์มาในขั้นที่ 2
- เปิด shortcut “Install dependencies”
- รอจนการติดตั้งเสร็จสิ้น

4. สร้างไฟล์ shortcut เพื่อเปิดโปรแกรม

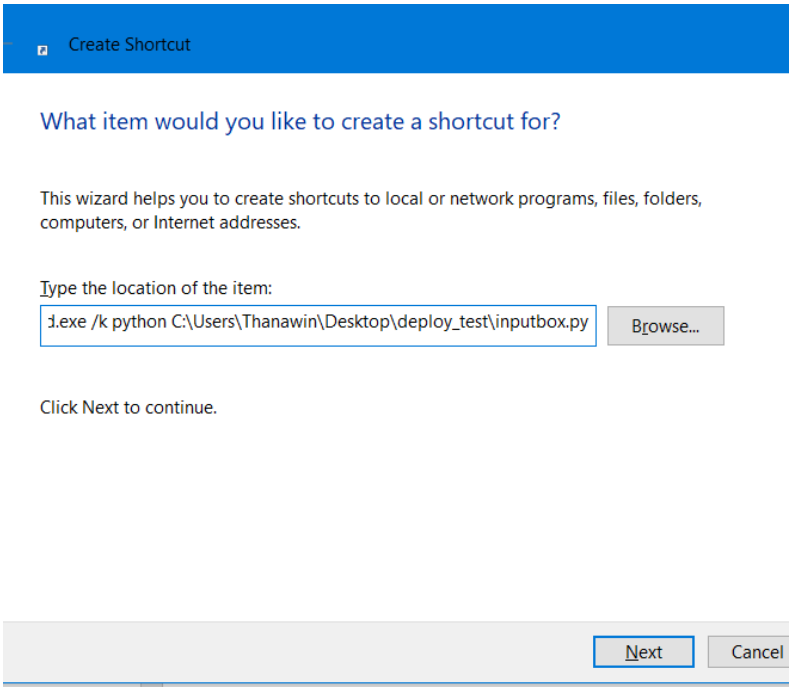
คลิกขวาบนไฟล์ inputbox.py แล้วเลือก properties และ Copy ที่อยู่ของไฟล์โปรแกรม



คลิกขวาในจุดที่ไม่ใช่ไฟล์ แล้วเลือก New -> Shortcut



กรอก C:\Windows\System32\cmd.exe /k python ***ที่อยู่ไฟล์***\inputbox.py



ตัวอย่าง

ที่อยู่ไฟล์ C:\Users\Thanawin\Desktop\deploy_test

ที่ต้องกรอกคือ

C:\Windows\System32\cmd.exe /k python C:\Users\Thanawin\Desktop\deploy_test\inputbox.py

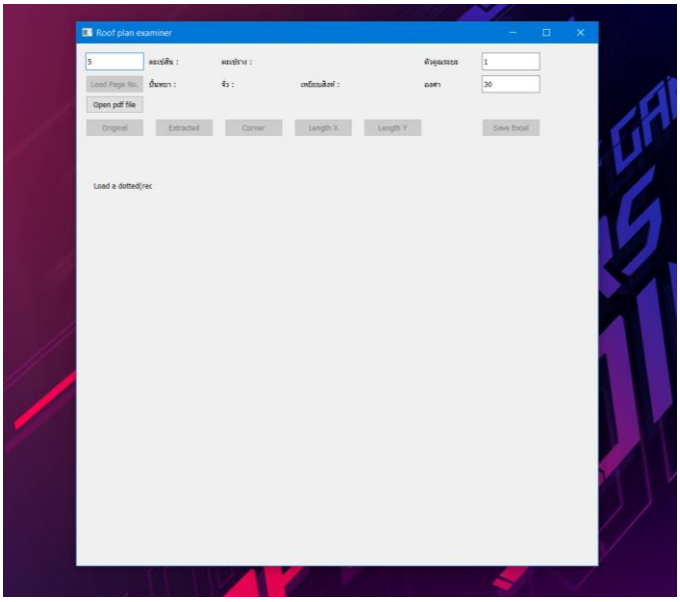
- เมื่อกรอกแล้ว กด next เพื่อตั้งชื่อ shortcut
- กด finish

เมื่อจะใช้งานโปรแกรม ให้เข้าจาก shortcut ที่สร้าง

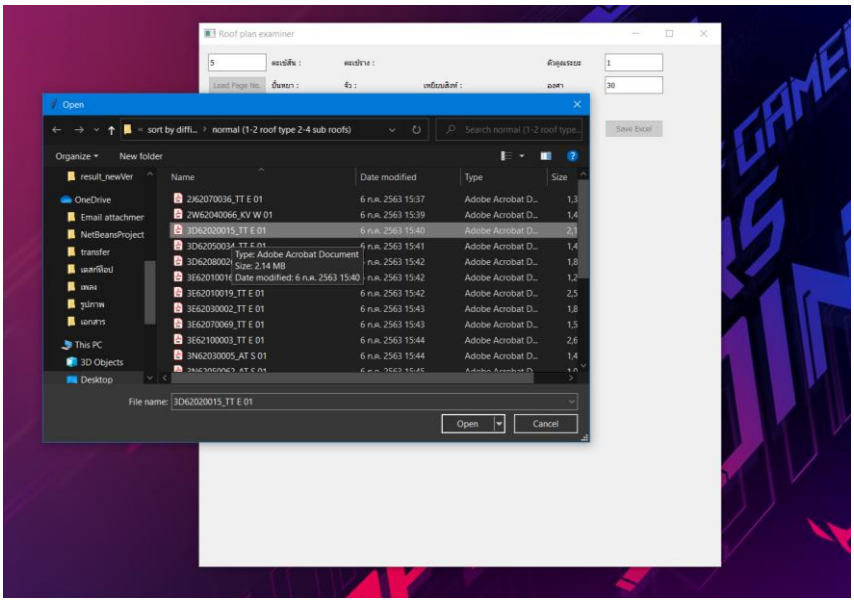
หากเปลี่ยนที่อยู่ของไฟล์โปรแกรม ให้ทำขั้นตอนนี้ใหม่อีกครั้ง

ขั้นตอนการใช้งานโปรแกรม

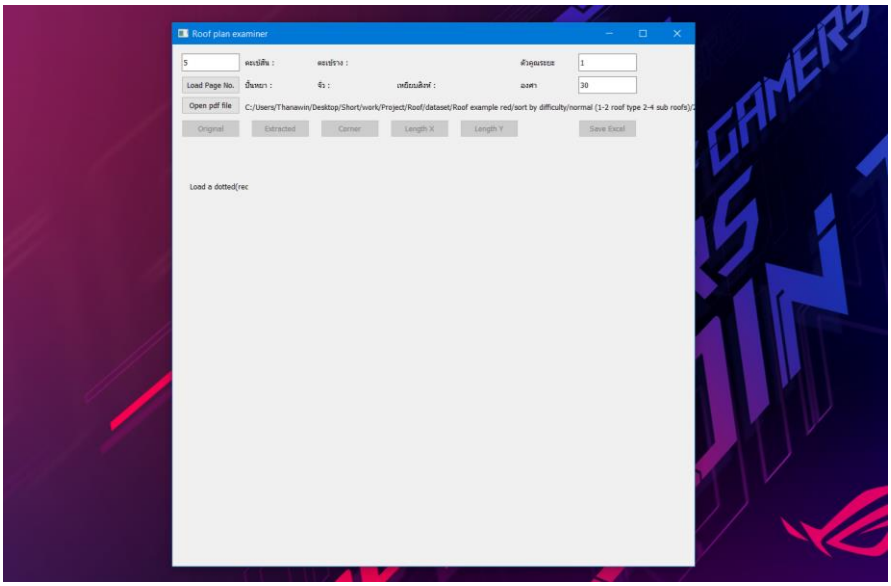
เมื่อเปิดโปรแกรมขึ้นมาจะพบหน้าต่างลักษณะนี้



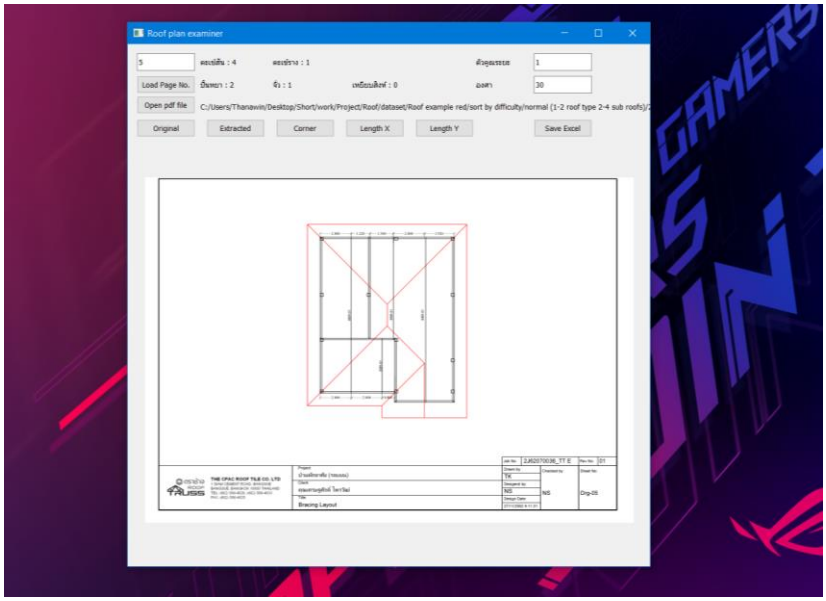
ให้กดปุ่ม Open pdf file เพื่อดำเนินการเลือกไฟล์ต่อ



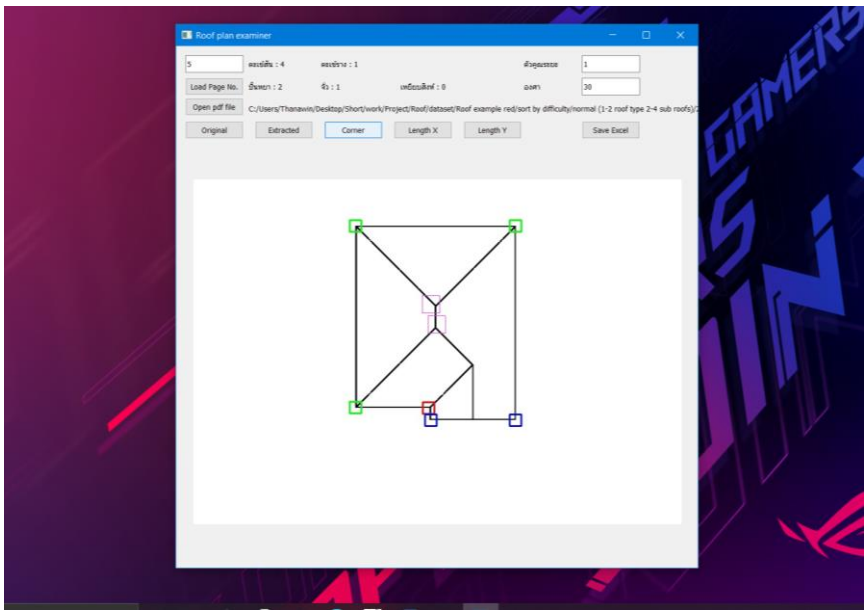
เมื่อกดแล้วจะทำการเปิดหน้าต่างย่อยใหม่ขึ้นมา ให้ทำการเลือกไฟล์ pdf ที่ต้องการ แล้วกด open



หลังจากนำเข้าไฟล์แล้ว จะพบชื่อไฟล์ข้างปุ่ม open pdf file แสดงว่านำเข้าไฟล์สำเร็จ ต่อมาให้กรอกเลขหน้าในไฟล์ pdf ที่ต้องการประมวลผล ในช่องบนซ้ายสุด เมื่อกรอกแล้วให้กดปุ่ม load page no แล้วรอการประมวลผลสักครู่



เมื่อทำการประมวลผลเสร็จ จะพบว่าที่ว่างด้านล่างจะมีรูปไฟล์ pdf ในหน้าที่เลือกปรากฏขึ้นมา และจะมีจำนวนขององค์ประกอบต่าง ๆแสดงอยู่ด้านบน



หากอยากทราบว่าโปรแกรมได้จับตำแหน่งใดมาใช้ ให้กดปุ่ม corner เพื่อแสดงภาพที่มีการตีกรอบบอกบริเวณที่ตรวจจับองค์ประกอบได้

หากอยากบันทึกเป็นไฟล์ excel ให้กดปุ่ม save excel หากบันทึกสำเร็จ จะมีหน้าต่างขึ้นมาแจ้งเตือน และสามารถเปิดไฟล์อื่นมาประมวลผลต่อโดยการกดปุ่ม open pdf file

ภาคผนวก ค
ซอร์สโค้ด

จากขั้นตอนการทำงานในข้อ 3.3 จะทำการแสดงซอร์สโค้ดจากฟังก์ชั่นที่สำคัญทั้งหมด 4 ฟังก์ชั่น ซึ่งประกอบด้วย การหามุมหลังคา การระบุประเภทของภาพย่อย การจับคู่ประเภทของภาพย่อยเพื่อระบุหลังคา และการหาทรงหลังคาภายในหลังคา ตามลำดับ

ซอร์สโค้ดการหาขอบหลังคา

```
def findRoi(img,sensitivity): #default 0.004
    size = img.shape
    #print(size)
    #crop_img = img[35:size[0]-35, 35:size[1]-35] #scg's standard
border length 35
    crop_img = img
    gray = cv2.cvtColor(crop_img,cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (5, 5), 0)
    edged = cv2.Canny(blurred, 50, 200, 255)

    cnts = cv2.findContours(edged.copy(),
cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)
    cnts = sorted(cnts, key=cv2.contourArea, reverse=True)

    for c in cnts:
        # approximate the contour
        peri = cv2.arcLength(c, True)
        approx = cv2.approxPolyDP(c, sensitivity * peri, True)
#default 0.004 less value more sensitive.
        #print(len(approx))

    # biggest zone should be the plan
    c = cnts[0]
    peri = cv2.arcLength(c, True)
    approx = cv2.approxPolyDP(c, sensitivity * peri, True) #default
0.004 less value more sensitive.

    edgelist = []
    for i in approx:
        x = i[0][0]
        y = i[0][1]
        fsize = 15

        tmp2 = x
        tmp3 = y

        isDuped = False
        for pic,x1,y1 in edgelist:
            if matchInexact(x1,tmp2,8) and matchInexact(y1,tmp3,8):
```

```
                isDuped = True

        if not isDuped:
            tmp1 = crop_img[y-fsize:y+fsize, x-fsize:x+fsize]
#standard size v1. 40*40  v2. 20*20
            #cv2.rectangle(crop_img,(x-fsize,y-
fsize),(x+fsize,y+fsize),(0,255,0),1)
            #cv2.imwrite(name+str(namec)+'.jpg',tmp1)
            #namec += 1
            edgelist.append((tmp1,tmp2,tmp3)) # image,coor x, coor y


    print("No. of corner : ",len(edgelist))
    return edgelist
```

ซอร์สโค้ดการระบุประเภทของภาพย่อย

```
def scanline(imgur,sens1 = 3.33,sens2 = 3.4):   #sens 1 = (2.35-3.33)
sens 2 = (3.4-5.2)#
    data = []

    step2 = []
    img = cv2.cvtColor(imgur,cv2.COLOR_BGR2GRAY)

    (h, w) = img.shape[:2]
    #img = sharpen(img)

    ret,th1 = cv2.threshold(img,160,255,cv2.THRESH_BINARY) #127-255
    #cv2.imwrite('testFunc0.jpg',th1)
    data.append(scanCol(th1,sens1)) ##2.35 , 3.35 red
    data.append(scanRow(th1,sens1))

    step1 = data.copy()

    img2 = rotate(th1,45)
    img2 = removeAllEdge(img2,int(h/4))
    img2 = sharpen(img2)

    # what is size again? size of deletion?  = dim/4
    blurred = cv2.GaussianBlur(img2, (3, 3), 0)
    ret,th2 = cv2.threshold(blurred,160,255,cv2.THRESH_BINARY)

    #cv2.imwrite('testFunc.jpg',th2)

    colz = scanCol(th2,sens2)  #3.4 - 5.2 red
    rowz = scanRow(th2,sens2)

    data.append(colz)
    data.append(rowz)

    step2.append(colz)
    step2.append(rowz)


    n_top = 0
    n_bot = 0
    n_left = 0
    n_right = 0

    passed = []
    #blacklist shape

    stepz1 = []
    stepz2 = []
    for s in step1:
        if s[0] != 0:
            for direction in s[1]:
                stepz1.append(direction)
    for s in step2:
        if s[0] != 0:
            for direction in s[1]:
                stepz2.append(direction)
```

```
    # Double check v.1.4

    #inside shape
    if len(stepz1) == 1 and stepz1[0] == 'bot':
        if len(stepz2) == 2:
            if (stepz2[0] == 'top' and stepz2[1] == 'right') or
(stepz2[0] == 'right' and stepz2[1] == 'top'):
                passed.append('roof seal')
                passed.append(0)
            elif(stepz2[0] == 'bot' and stepz2[1] == 'left') or
(stepz2[0] == 'left' and stepz2[1] == 'bot'):
                passed.append('roof seal')
                passed.append(1)
            elif(stepz2[0] == 'top' and stepz2[1] == 'left') or
(stepz2[0] == 'left' and stepz2[1] == 'top'):
                passed.append('roof hip')
                passed.append(1)

        elif len(stepz2) == 1 and stepz2[0] == 'left':
            passed.append('roof dutch')
            passed.append(4)
        elif len(stepz2) == 1 and stepz2[0] == 'top':
            passed.append('roof dutch')
            passed.append(6)

    elif len(stepz1) == 1 and stepz1[0] == 'top':
        if len(stepz2) == 2:
            if (stepz2[0] == 'top' and stepz2[1] == 'right') or
(stepz2[0] == 'right' and stepz2[1] == 'top'):
                passed.append('roof seal')
                passed.append(2)
            elif(stepz2[0] == 'bot' and stepz2[1] == 'left') or
(stepz2[0] == 'left' and stepz2[1] == 'bot'):
                passed.append('roof seal')
                passed.append(3)
            elif(stepz2[0] == 'bot' and stepz2[1] == 'right') or
(stepz2[0] == 'right' and stepz2[1] == 'bot'):
                passed.append('roof hip')
                passed.append(0)
        elif len(stepz2) == 1 and stepz2[0] == 'bot':
            passed.append('roof dutch')
            passed.append(5)
        elif len(stepz2) == 1 and stepz2[0] == 'right':
            passed.append('roof dutch')
            passed.append(7)

    elif len(stepz1) == 1 and stepz1[0] == 'right':
        if len(stepz2) == 2:
            if(stepz2[0] == 'bot' and stepz2[1] == 'left') or
(stepz2[0] == 'left' and stepz2[1] == 'bot'):
                passed.append('roof hip')
                passed.append(2)
            if(stepz2[0] == 'top' and stepz2[1] == 'left') or
(stepz2[0] == 'left' and stepz2[1] == 'top'):
                passed.append('roof seal')
                passed.append(6)
            if(stepz2[0] == 'bot' and stepz2[1] == 'right') or
(stepz2[0] == 'right' and stepz2[1] == 'bot'):
                passed.append('roof seal')
                passed.append(7)
```

```
        elif len(stepz2) == 1 and stepz2[0] == 'bot':
            passed.append('roof dutch')
            passed.append(0)
        elif len(stepz2) == 1 and stepz2[0] == 'left':
            passed.append('roof dutch')
            passed.append(2)
    elif len(stepz1) == 1 and stepz1[0] == 'left':
        if len(stepz2) == 2:
            if (stepz2[0] == 'top' and stepz2[1] == 'right') or
(stepz2[0] == 'right' and stepz2[1] == 'top'):
                passed.append('roof hip')
                passed.append(3)
            elif (stepz2[0] == 'bot' and stepz2[1] == 'right') or
(stepz2[0] == 'right' and stepz2[1] == 'bot'):
                passed.append('roof seal')
                passed.append(4)
            elif (stepz2[0] == 'top' and stepz2[1] == 'left') or
(stepz2[0] == 'left' and stepz2[1] == 'top'):
                passed.append('roof seal')
                passed.append(5)

        elif len(stepz2) == 1 and stepz2[0] == 'right':
            passed.append('roof dutch')
            passed.append(1)
        elif len(stepz2) == 1 and stepz2[0] == 'top':
            passed.append('roof dutch')
            passed.append(3)


    #edge shape
    elif len(stepz1) == 2:
        if (stepz1[0] == 'top' and stepz1[1] == 'right') or
(stepz1[0] == 'right' and stepz1[1] == 'top'):
            if len(stepz2) == 1 and stepz2[0] == 'top':
                passed.append('sun')
                passed.append(0)
            elif len(stepz2) == 1 and stepz2[0] == 'bot':
                passed.append('rang')
                passed.append(0)
            elif len(stepz2) == 0:
                passed.append('jua')
                passed.append(0)

        if (stepz1[0] == 'bot' and stepz1[1] == 'right') or
(stepz1[0] == 'right' and stepz1[1] == 'bot'):
            if len(stepz2) == 1 and stepz2[0] == 'right':
                passed.append('sun')
                passed.append(1)
            elif len(stepz2) == 1 and stepz2[0] == 'left':
                passed.append('rang')
                passed.append(1)
            elif len(stepz2) == 0:
                passed.append('jua')
                passed.append(1)

        if (stepz1[0] == 'top' and stepz1[1] == 'left') or (stepz1[0]
== 'left' and stepz1[1] == 'top'):
            if len(stepz2) == 1 and stepz2[0] == 'left':
                passed.append('sun')
                passed.append(2)
```

```
            elif len(stepz2) == 1 and stepz2[0] == 'right':
                passed.append('rang')
                passed.append(2)
            elif len(stepz2) == 0:
                passed.append('jua')
                passed.append(2)

        if (stepz1[0] == 'bot' and stepz1[1] == 'left') or (stepz1[0]
== 'left' and stepz1[1] == 'bot'):
            if len(stepz2) == 1 and stepz2[0] == 'bot':
                passed.append('sun')
                passed.append(3)
            elif len(stepz2) == 1 and stepz2[0] == 'top':
                passed.append('rang')
                passed.append(3)
            elif len(stepz2) == 0:
                passed.append('jua')
                passed.append(3)




    ####

    for d in data:
        #print(d)
        if d[0] != 0:
            for direction in d[1]:
                if direction == 'top':
                    n_top += 1
                elif direction == 'bot':
                    n_bot +=1
                elif direction == 'left':
                    n_left +=1
                elif direction == 'right':
                    n_right +=1

    n_dir = (n_top,n_bot,n_left,n_right)
    return n_dir,passed
```

ซอร์สโค้ดการจับคู่ประเภทของภาพย่อยเพื่อระบุหลังคา

```
def matchTakeiX(a,b,fullimg):  #adding check inside image
    imga = a[0]
    imgb = b[0]
    x1 = a[1]
    y1 = a[2]
    x2 = b[1]
    y2 = b[2]

    tmp1,tmp2 = scanline(imga)
    takeiA,typeA = findTakei(tmp1,tmp2)

    tmp1,tmp2 = scanline(imgb)
    takeiB,typeB = findTakei(tmp1,tmp2)

    fsize = 15
    dist = math.hypot(x2 - x1, y2 - y1)
    #0.126x - 26.014 = y
    offset = int((0.126*dist) - 26.014)

    if takeiA == takeiB :
        if (typeA == 0 and typeB == 2) or (typeA == 1 and typeB == 3)
or (typeA == 2 and typeB == 0) or (typeA == 3 and typeB == 1):
            if takeiA == 'sun':
                #added v1.2 -- should collect inside box for further
checking
                #check hip
                if (typeA == 0 and typeB == 2) or (typeA == 2 and
typeB == 0):

                    middleX = int((x1+x2)/2)
                    middleY = y1 - int(dist/2)
                elif (typeA == 1 and typeB == 3) or (typeA == 3 and
typeB == 1):

                    middleX = int((x1+x2)/2)
                    middleY = y1 + int(dist/2)
                insidePic = fullimg[middleY-fsize:middleY+fsize,
middleX-fsize:middleX+fsize]
                #cv2.rectangle(fullimg,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(200,100,0),2)
                #cv2.imwrite("test_middlehip.jpg",insidePic)
                tmp1,tmp2 = scanline(insidePic)
                out = findTakei(tmp1,tmp2)
                print("check for hip")
                print(out)
                print("")

                #check yeabsingh(dutch)
                if (typeA == 0 and typeB == 2) or (typeA == 2 and
typeB == 0):

                    middleX = int((x1+x2)/2)
                    middleX = int((middleX + x1)/2) + offset
                    middleY = y1 - int(dist/2)
                    middleY = int((y1 + middleY)/2) - offset
```

```
                elif (typeA == 1 and typeB == 3) or (typeA == 3 and
typeB == 1):

                    middleX = int((x1+x2)/2)
                    middleX = int((middleX + x1)/2) + offset
                    middleY = y1 + int(dist/2)
                    middleY = int((y1 + middleY)/2) + offset

                aa = 0
                insidePic = fullimg[middleY-fsize:middleY+fsize,
middleX-fsize:middleX+fsize]
                #cv2.rectangle(fullimg,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(200,100,0),2)
                #cv2.imwrite("test_matchtakei.jpg",fullimg)

                tmp1,tmp2 = scanline(insidePic)
                out = findTakei(tmp1,tmp2)
                print("check hip low x")
                #print(scanline(insidePic))
                #print(findTakei(scanline(insidePic)))
                print()

                ######## another side
                if (typeA == 0 and typeB == 2) or (typeA == 2 and
typeB == 0):

                    middleX = int((x1+x2)/2)
                    middleX = int((middleX + x2)/2) - offset
                    middleY = y1 - int(dist/2)
                    middleY = int((y1 + middleY)/2) - offset
                elif (typeA == 1 and typeB == 3) or (typeA == 3 and
typeB == 1):

                    middleX = int((x1+x2)/2)
                    middleX = int((middleX + x2)/2) - offset
                    middleY = y1 + int(dist/2)
                    middleY = int((y1 + middleY)/2) + offset

                insidePic = fullimg[middleY-fsize:middleY+fsize,
middleX-fsize:middleX+fsize]
                #cv2.rectangle(fullimg,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(200,100,0),2)

                tmp1,tmp2 = scanline(insidePic)
                out2 = findTakei(tmp1,tmp2)

                print("check hip low x2")
                #print(scanline(insidePic))
                #print(findTakei(scanline(insidePic)))
                print()

                #may need special line no. evaluation
                if out[0] == 'rang' or out2[0] == 'rang':
                    return True,'hip-dutch',dist



                #check hip
                if (typeA == 0 and typeB == 2) or (typeA == 2 and
typeB == 0):
```

```
                    middleX = int((x1+x2)/2)
                    middleY = y1 - int(dist/2)
                elif (typeA == 1 and typeB == 3) or (typeA == 3 and
typeB == 1):
                    middleX = int((x1+x2)/2)
                    middleY = y1 + int(dist/2)
                #cv2.rectangle(fullimg,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(0,255,0),2)
                fsize2 = 10
                insidePic = fullimg[middleY-fsize2:middleY+fsize2,
middleX-fsize2:middleX+fsize2]
                tmp1,tmp2 = scanline(insidePic)
                out = findTakei(tmp1,tmp2)
                print("check dutch x")
                #print(scanline(insidePic))
                #print(findTakei(scanline(insidePic)))
                print(out)
                #dont need further editing
                if out[0]== 'rang':
                    return True,'hip-dutch',dist
                elif out[0] == 'line' :
                    #change line for higher threshold?
                    return True,'dutch',dist
                else:
                    return False,'invalid',dist
            elif takeiA == 'jua':
                middleX = int((x1+x2)/2)
                middleY = y1
                insidePic = fullimg[middleY-fsize:middleY+fsize,
middleX-fsize:middleX+fsize]
                #cv2.rectangle(fullimg,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(0,255,0),2)
                tmp1,tmp2 = scanline(insidePic)
                out = findTakei(tmp1,tmp2)
                print("check jua x")
                #print(scanline(insidePic))
                print(out)
                print(tmp1)
                print()
                #scan the line itself
                limX1 = int(abs(x1-x2)/2)
                limX2 = int(abs(x1-x2)/2)
                middleX1 = middleX
                middleX2 = middleX

                lim = int(limX1/4)

                if out[0] != 'jua' :
                    while (limX1 > lim and limX2 > lim):
                        middleX1 = middleX1-5
                        middleX2 = middleX2+5

                        insidePic1 = fullimg[middleY-
fsize:middleY+fsize, middleX1-fsize:middleX1+fsize]
                        insidePic2 = fullimg[middleY-
fsize:middleY+fsize, middleX2-fsize:middleX2+fsize]

                        tmpa1,tmpa2 = scanline(insidePic1)
                        outa = findTakei(tmpa1,tmpa2)
                        tmpb1,tmpb2 = scanline(insidePic2)
```

```
                        outb = findTakei(tmpb1,tmpb2)

                        if outa[0] == 'jua' or outb[0] == 'jua':
                            #cv2.rectangle(fullimg,(middleX1-
fsize,middleY-fsize),(middleX1+fsize,middleY+fsize),(255,0,0),2)
                            #cv2.rectangle(fullimg,(middleX2-
fsize,middleY-fsize),(middleX2+fsize,middleY+fsize),(255,0,255),2)
                            #return True,'gable',dist
                            print("found jua X")
                            return True,'gable',dist
                            break

                        if x2 > x1:
                            limX1 = abs(middleX1 - x1)
                            limX2 = abs(middleX2 - x2)
                        elif x1 > x2:
                            limX1 = abs(middleX2 - x1)
                            limX2 = abs(middleX1 - x2)

                #neednt further editing
                if out[0] == 'jua' or (out[0] == 'wtf' and 4 <=
out[1] <7):
                    #cv2.rectangle(fullimg,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(0,255,0),2)
                    print("found jua x")
                    return True,'gable',dist
                else:
                    return False,'invalid',dist

            elif takeiA == 'rang':
                return True,'hmm',dist
        else:
            return False,'invalid',dist
    else:
        if(takeiA == 'sun' and takeiB == 'rang'):


            #sun_mismatch += 1
            return False,'sun_mismatch',dist

        elif(takeiA == 'rang' and takeiB == 'sun'):
            return False,'sun_mismatch',dist
        return False,'invalid',dist
```

ซอร์สโค้ดการหาทรงหลังคาภายในหลังคา

```
def findInside(edgelist,fsize,img,draw_img,checkSingha = True):
    minx = 9999
    miny = 9999
    maxx = 0
    maxy = 0

    millis1 = int(round(time.time() * 1000))


    for e in edgelist:
        print("x:",e[1]," y:",e[2])
        if e[1] < minx:
            minx = e[1]
        if e[1] > maxx:
            maxx = e[1]
        if e[2] < miny:
            miny = e[2]
        if e[2] > maxy:
            maxy = e[2]



    x = minx + (fsize+10)   #sometime too close to border
    y = miny + (fsize+10)
    half = int(fsize/3) #v.1.3 scan per fsize, v.1.4 scan double to
fsize/2 more number more accurate
    chunkNo = 0
    insideCoor = []
    roofseal = []
    insideDutch = []

    while y < maxy - (fsize+10):
        while x < maxx - (fsize+10):
            tmp1 = img[y-fsize:y+fsize, x-fsize:x+fsize]
            if checkSingha:
                tmp_dir,tmp_pass = scanline(tmp1)
            else:
                tmp_dir,tmp_pass = scanline(tmp1,3,4)
            out = findTakei(tmp_dir,tmp_pass)
            isEdge = False
            for e in edgelist:
                if getDist(x,y,e[1],e[2]) < (fsize): #fsize*2
                    isEdge = True
                    break
            if not(isEdge) and (out[0]== 'roof hip'):
                print("found hip")
                if len(insideCoor) == 0:
                    insideCoor.append((x,y))
                    cv2.rectangle(draw_img,(x-fsize,y-
fsize),(x+fsize,y+fsize),(255,50,255),1)
                    print(out[0],out[1])
                    print(tmp_dir)
                    print(tmp_pass)
                else:
                    isDuped = False

                    for shape in insideCoor:
```

```
                        if getDist(x,y,shape[0],shape[1]) <
(fsize*2):
                            isDuped = True
                    if not isDuped:
                        insideCoor.append((x,y))
                        cv2.rectangle(draw_img,(x-fsize,y-
fsize),(x+fsize,y+fsize),(255,50,255),1)
                        print(out[0],out[1])
                        print(tmp_dir)
                        print(tmp_pass)
                    else:
                        print("reject duplicate")

            elif not(isEdge) and (out[0]== 'roof seal'):
                if len(roofseal) == 0:
                    roofseal.append((x,y))
                    #cv2.rectangle(draw_img,(x-fsize,y-
fsize),(x+fsize,y+fsize),(100,50,255),1)
                else:
                    isDuped = False

                    for shape in roofseal:
                        if getDist(x,y,shape[0],shape[1]) <
(fsize*2):
                            isDuped = True
                    if not isDuped:
                        roofseal.append((x,y))
                        #cv2.rectangle(draw_img,(x-fsize,y-
fsize),(x+fsize,y+fsize),(100,50,255),1)
                        print(out[0],out[1])
                    else:
                        print("reject duplicate")

            elif not(isEdge) and (out[0]== 'roof dutch'):
                if len(insideDutch) == 0:
                    insideDutch.append((x,y,out[1]))
                    #cv2.rectangle(draw_img,(x-fsize,y-
fsize),(x+fsize,y+fsize),(100,100,255),1)
                else:
                    isDuped = False

                    insideDutch.append((x,y,out[1]))

            x = x + half
            chunkNo+=1

        x = minx
        y = y + int(half)

    # END of while loop
    #### EDIT THSI SEGMENT FOR MORE ACCURATE COORDINATES ####
    distance = fsize*2
    tmpBlue = []
    blueCluster = []

    for point in insideDutch:
        #base case
        if len(tmpBlue) == 0:
            tmpBlue.append(point)
            continue
```

```
        else:
            tmp_avgCoor = getAvgCoor(tmpBlue)
            if
getDist(point[0],point[1],tmp_avgCoor[0],tmp_avgCoor[1]) < distance:
                tmpBlue.append(point)
                continue
            else:
                if len(blueCluster) == 0:
                    blueCluster.append(tmpBlue)
                    tmpBlue = []
                    tmpBlue.append(point)
                    continue
                else:
                    #search recorded cluster
                    added = False
                    for i in range(len(blueCluster)):
                        tmp_avgCoor2 = getAvgCoor(blueCluster[i])
                        if
getDist(point[0],point[1],tmp_avgCoor2[0],tmp_avgCoor2[1]) <
distance:
                            blueCluster[i].append(point)
                            added = True
                            break

                    #not found, create new cluster
                    if not added:
                        blueCluster.append(tmpBlue)
                        tmpBlue = []
                        tmpBlue.append(point)
                        continue
                    if added:
                        continue

    blueCluster.append(tmpBlue)
    tmp_insideDutch = []
    try:
        for cluster in blueCluster:

            point_x,point_y,typez = cluster[0]
            tmp_coor_x,tmp_coor_y = getAvgCoor(cluster)

            tmp_insideDutch.append((tmp_coor_x,tmp_coor_y,typez))
    except Exception as exc:
        print(exc)
    insideDutch = tmp_insideDutch

    ## FIX DONE ##

    #identify dutch from inside
    dutch = 0
    taken = []
    print(" yeetttttt inside dutch",len(insideDutch))
    if len(insideDutch) > 0 and checkSingha:
        for dude in insideDutch:
            cv2.rectangle(draw_img,(dude[0]-fsize,dude[1]-
fsize),(dude[0]+fsize,dude[1]+fsize),(0,0,0),1)
            for lass in insideDutch:
                x1 = dude[0]
                y1 = dude[1]
                x2 = lass[0]
```

```
                y2 = lass[1]

                if x1 == x2 and y1 == y2:
                    continue

                breakCall = 'go'
                for t in taken:
                    if (x1 == t[0] and y1 == t[1]):
                        breakCall = 'change dude'
                        break
                    if (x2 == t[0] and y2 == t[1]):
                        breakCall = 'change lass'
                        break
                if breakCall == 'change dude':
                    break
                elif breakCall == 'change lass':
                    continue

                #Check Vertical
                if matchInexact(x1,x2,4):  #defualt 4
                    middleX = x1
                    middleY = int((y1+y2)/2)
                    insidePic = img[middleY-fsize:middleY+fsize,
middleX-fsize:middleX+fsize]
                    #cv2.rectangle(draw_img,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(0,255,0),4)
                    tmp1,tmp2 = scanline(insidePic)
                    out = findTakei(tmp1,tmp2)


                    if (dude[2] == 4 and lass[2] == 5) or (dude[2] ==
5 and lass[2] == 4):

                        print("umm",out[0])
                        if out[0] == 'jua' or out[0] == "line":  # or
line added 8 DEC 2020 for more recall
                            cv2.rectangle(draw_img,(x1-fsize,y1-
fsize),(x1+fsize,y1+fsize),(100,100,255),1)
                            cv2.rectangle(draw_img,(x2-fsize,y2-
fsize),(x2+fsize,y2+fsize),(100,100,255),1)
                            taken.append(dude)
                            taken.append(lass)
                            dutch += 1


                    elif (dude[2] == 6 and lass[2] == 7) or (dude[2]
== 7 and lass[2] == 6):

                        if out[0] == 'jua' or out[0] == "line": # or
line added 8 DEC 2020 for more recall
                            cv2.rectangle(draw_img,(x1-fsize,y1-
fsize),(x1+fsize,y1+fsize),(100,100,255),1)
                            cv2.rectangle(draw_img,(x2-fsize,y2-
fsize),(x2+fsize,y2+fsize),(100,100,255),1)
                            taken.append(dude)
                            taken.append(lass)
                            dutch += 1
                #Check Horizontal
                if matchInexact(y1,y2,4): #default 4
                    middleX = int((x1+x2)/2)
```

```
                    middleY = y1
                    insidePic = img[middleY-fsize:middleY+fsize,
middleX-fsize:middleX+fsize]
                    #cv2.rectangle(draw_img,(middleX-fsize,middleY-
fsize),(middleX+fsize,middleY+fsize),(0,255,0),2)
                    tmp1,tmp2 = scanline(insidePic)
                    out = findTakei(tmp1,tmp2)

                    if (dude[2] == 0 and lass[2] == 1) or (dude[2] ==
1 and lass[2] == 0):
                        if out[0] == 'jua':
                            cv2.rectangle(draw_img,(x1-fsize,y1-
fsize),(x1+fsize,y1+fsize),(100,100,255),1)
                            cv2.rectangle(draw_img,(x2-fsize,y2-
fsize),(x2+fsize,y2+fsize),(100,100,255),1)
                            taken.append(dude)
                            taken.append(lass)
                            dutch += 1
                    elif (dude[2] == 2 and lass[2] == 3) or (dude[2]
== 3 and lass[2] == 2):
                        if out[0] == 'jua':
                            cv2.rectangle(draw_img,(x1-fsize,y1-
fsize),(x1+fsize,y1+fsize),(100,100,255),1)
                            cv2.rectangle(draw_img,(x2-fsize,y2-
fsize),(x2+fsize,y2+fsize),(100,100,255),1)
                            taken.append(dude)
                            taken.append(lass)
                            dutch += 1

    print(chunkNo)
    print("inside hip",len(insideCoor))
    print("inside dutch",dutch)
    millis2 = int(round(time.time() * 1000))
    mill = millis2-millis1
    print("time used in process :",mill)

    cv2.imwrite("takei_test.jpg",draw_img)

    return insideCoor,roofseal,dutch
```

ประวัติผู้เขียน

นาย ธนวิน อาภาจรัส
คณะวิทยาศาสตร์ สาขาวิชาวิทยาการคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย
โทรศัพท์ : 081-487-9807
อีเมล์ : Thanawin25000@gmail.com