

ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้
การสกัดลักษณะและการเรียนรู้ของเครื่อง

นายกฤติน ลีวนิชย์
นางสาวจิตาภา พงศ์หบดี

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2568
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Optical Satellite Identification and Tracking System from Astronomical
Images Using Feature Extraction and Machine Learning

Krittin Leewanich

Chidapha Phonkhahabodi

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 2025

Copyright of Chulalongkorn University

หัวข้อโครงการ	ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้
	การสกัดลักษณะและการเรียนรู้ของเครื่อง
โดย	นายกฤติน ลีวนิชย์
	นางสาวจิตาภา พงศ์คหบดี
สาขาวิชา	วิทยาการคอมพิวเตอร์
อาจารย์ที่ปรึกษาโครงการหลัก	รองศาสตราจารย์ ดร. รัชลิดา ลิปิกรณ์

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
อนุมัติให้นับโครงการฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา
2301499 โครงการวิทยาศาสตร์ (Senior Project)

..... หัวหน้าภาควิชาคณิตศาสตร์
(รองศาสตราจารย์ ดร.อธิปัติย์ อารังธัญลักษณ์) และวิทยาการคอมพิวเตอร์

คณะกรรมการสอบโครงการ

..... อาจารย์ที่ปรึกษาโครงการหลัก
(รองศาสตราจารย์ ดร.รัชลิดา ลิปิกรณ์)

..... กรรมการ
(ศาสตราจารย์ ดร.วัชรินทร์ วิจิธมาลา)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.วุฒิชัย จงจิตเมตต์)

นายกฤติน ลีวนิชย์, นางสาวจิตาภา พงศ์คหบดี: ชื่อโครงการภาษาไทย ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง. (Optical Satellite Identification and Tracking System from Astronomical Images Using Feature Extraction and Machine Learning) อ.ที่ปรึกษาโครงการหลัก: รองศาสตราจารย์ ดร. รัชลิดา ลิปิกรณ์, 72 หน้า.

โครงการวิจัยในชั้นเรียน เรื่อง “ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง” มีวัตถุประสงค์เพื่อพัฒนาระบบสำหรับการตรวจจับระบุ และติดตามวัตถุในภาพถ่ายทางดาราศาสตร์ โดยอาศัยเทคนิคการสกัดลักษณะร่วมกับการเรียนรู้ของเครื่องเพื่อทำการคัดแยกวัตถุออกเป็นประเภทต่าง ๆ ได้แก่ ดาวเคราะห์ในระบบสุริยะ ดาวฤกษ์ นอกกระบบสุริยะ ดาวเทียม และวัตถุอื่น ๆ นอกจากนี้ โครงการยังมุ่งพัฒนาวิธีการคำนวณตำแหน่งของวัตถุที่ตรวจพบจากภาพ และนำข้อมูลดังกล่าวไปเปรียบเทียบกับข้อมูลในบัญชีรายชื่อ เพื่อช่วยในการระบุชนิดของวัตถุได้อย่างแม่นยำมากยิ่งขึ้น โดยผลลัพธ์ที่ได้สามารถนำไปใช้สนับสนุนการวิเคราะห์และติดตามวัตถุทางดาราศาสตร์ในเชิงประยุกต์ได้

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต กฤติน ลีวนิชย์
 ลายมือชื่อนิสิต จิตาภา พงศ์คหบดี
 สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาโครงการหลัก รัชลิดา ลิปิกรณ์
 ปีการศึกษา 2568

6534402523, 6534408323: MAJOR COMPUTER SCIENCE

KEYWORDS: SATELLITE IDENTIFICATION / FEATURE EXTRACTION/ MACHINE LEARNING /
ASTRONOMICAL IMAGE

KRITTIN LEEWANICH, CHIDAPHA PHONGKHAHABODI: OPTICAL SATELLITE
IDENTIFICATION AND TRACKING SYSTEM FROM ASTRONOMICAL IMAGES USING
FEATURE EXTRACTION AND MACHINE LEARNING. ADVISOR: ASSOC. PROF. RAJALIDA
LIPIKORN, Ph.D., 72 pp.

The proposed project titled “Optical Satellite Identification and Tracking System from Astronomical Images Using Feature Extraction and Machine Learning” aims to develop a system for detecting, identifying, and tracking objects in astronomical images. The system employs feature extraction techniques in conjunction with machine learning to classify objects into various categories, including planets in the solar system, extrasolar stars, satellites, and other objects. In addition, the project seeks to develop methods for calculating the positions of detected objects from images and comparing such information with data in catalogs to enhance the accuracy of object identification. The resulting outcomes can be utilized to support the analysis and tracking of astronomical objects in applied contexts.

Department: Mathematics and Computer Science Student's Signature กฤตติน ล้วนนิชย์

Student's Signature Chidapha Phongkhabodi

Field of Study: Computer Science

Advisor's Signature Rajalida Lipikorn

Academic Year: 2025

กิตติกรรมประกาศ

โครงการวิจัยในชั้นเรียนเรื่อง “ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง” สำเร็จลุล่วงไปได้ด้วยดีเนื่องจากได้รับความอนุเคราะห์และการสนับสนุนจากคณาจารย์และบุคลากรต่าง ๆ หลายท่าน

ขอขอบพระคุณ นายกันต์ธนากร น้อยเสนา นายพีรวิชญ์ ปุกแก้วศรี นางสาวชนกพร สายวิวัฒน์ และนางสาวภัทรธร สรณธาวรกุล นักวิจัยจากสถาบันวิจัยดาราศาสตร์แห่งชาติที่ให้ความรู้ในส่วนขององค์ความรู้ทางดาราศาสตร์ที่เกี่ยวข้องกับโครงการ ให้ความอนุเคราะห์แก่คณะผู้จัดทำในการนำภาพถ่ายจากกล้อง ZWO ASI294MM Pro มาใช้ในการพัฒนาและทดสอบระบบ ตลอดจนเสนอแนะวิธีการในการแก้ปัญหาต่าง ๆ ที่โครงการต้องประสบพบเจอตลอดการพัฒนา

ขอขอบพระคุณ รองศาสตราจารย์ ดร.รัชลิดา ลิปิกรณ์ อาจารย์ที่ปรึกษาโครงการ ที่ให้ความช่วยเหลือตลอดช่วงเวลาการทำโครงการ ตลอดจนให้คำแนะนำและชี้แนะข้อบกพร่องต่าง ๆ จนโครงการนี้สำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณ ศาสตราจารย์ ดร.วัชรินทร์ วิชิรมาลา กรรมการสอบโครงการ ที่ช่วยชี้แนะข้อบกพร่องและเสนอแนะวิธีการในการแก้ไขทำให้โครงการมีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.วุฒิชัย จงจิตเมตต์ กรรมการสอบโครงการ ที่ช่วยชี้แนะข้อบกพร่องและเสนอแนะวิธีการในการแก้ไขทำให้โครงการมีความสมบูรณ์มากยิ่งขึ้น

ท้ายสุดนี้ คณะผู้จัดทำใคร่ขอขอบพระคุณบุคลากรทุกท่าน ทั้งที่ได้กล่าวไปข้างต้นและที่ยังไม่ได้กล่าวถึง ที่มีส่วนช่วยเหลือให้โครงการวิจัยชิ้นนี้สำเร็จลุล่วงไปได้ด้วยดีไว้ ณ ที่นี้ด้วย และหวังว่าโครงการวิจัยชิ้นนี้จะเป็นประโยชน์กับการสังเกตการณ์ท้องฟ้าต่อไป

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฅ
สารบัญภาพ	ญ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและเหตุผลการวิจัย	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตการวิจัย	2
1.4 ขั้นตอนการวิจัย	3
1.5 ประโยชน์ที่ได้รับ	4
1.6 โครงสร้างของรายงาน	5
บทที่ 2 งานวิจัยที่เกี่ยวข้อง	6
2.1 การตรวจหารอยทางของดาวเทียมโดยใช้โมเดล U-Net	6
2.2 การแบ่งละเอียดด้วยการแปลงเชิงน่าจะเป็นของฮัฟ	8
2.2 การวิเคราะห์เส้นขอบ และการสกัดลักษณะ	9
2.2 ชุดองค์ประกอบสองบรรทัด และการใช้งาน	10
บทที่ 3 วิธีการวิจัย	12
3.1 การรวบรวมข้อมูลที่ใช้ตรวจสอบ	12
3.2 การประมวลผลภาพก่อน (Preprocessing)	13
3.3 การเชื่อมเส้นและสกัดเส้นตรงของวิถีดาวเทียม	15

3.4	การคำนวณสมการเส้นและทิศทางการเคลื่อนที่.....	20
3.5	กระบวนการทำงานของระบบหลักในส่วนของการประมวลผลภาพ.....	23
3.6	การกำหนดขอบเขตการมองเห็นของระบบ.....	28
3.7	การจัดเตรียมข้อมูลดาวเทียมย้อนหลังจากบัญชีรายชื่อดาวเทียม.....	31
3.8	การกำหนดกรอบเวลาและตำแหน่ง.....	32
3.9	การคัดกรองและติดตามดาวเทียมในขอบเขตภาพ.....	34
3.10	การประเมินเปรียบเทียบข้อมูลวิถีดาวเทียมที่ได้จากการคำนวณและผลที่ได้จากการสกัดข้อมูลจากภาพ.....	36
บทที่ 4	ผลการวิจัย.....	38
4.1	ผลการตรวจจับเส้น.....	38
4.2	ผลการเชื่อมเส้น.....	39
4.3	ผลการวิเคราะห์เส้นวิถีและทิศทางการเคลื่อนที่ของวัตถุ.....	41
4.4	ผลจากการประเมินเปรียบเทียบข้อมูลวิถีดาวเทียมที่ได้จากการคำนวณและผลที่ได้จากการสกัดข้อมูลจากภาพบนข้อมูลที่มีอยู่.....	43
บทที่ 5	ข้อสรุปและข้อเสนอแนะ.....	50
5.1	ข้อสรุป.....	50
5.2	ข้อเสนอแนะ.....	51
	รายการอ้างอิง.....	52
	ภาคผนวก ก แบบเสนอหัวข้อโครงการ รายวิชา 2301399 Project Proposal ปีการศึกษา 2568..	55
	ประวัติผู้เขียน.....	61

สารบัญตาราง

	หน้า
ตารางที่ 1 แผนภูมิแกนต์แสดงขั้นตอนการวิจัยและช่วงเวลาของแต่ละขั้นตอน.....	4
ตารางที่ 2 ตารางเวกเตอร์การเคลื่อนที่และทิศทาง.....	41
ตารางที่ 3 ตารางสมการเส้นตรงและจุดปลายเส้น	42

สารบัญภาพ

หน้า

ภาพที่ 1 กลุ่มภาพเปรียบเทียบภาพเนกาทีฟสีตัวอย่างในผังซ้ายกับกลุ่มภาพหน้ากากจริงในผังขวา ...	7
ภาพที่ 2 ภาพเปรียบเทียบหน้ากากจริงกับผลลัพธ์ U-Net พร้อมแสดงความน่าจะเป็นของพิกเซลและรายละเอียดของเส้น	7
ภาพที่ 3 กราฟแสดงค่าประสิทธิภาพของ U-Net ที่ค่าขีดแบ่งต่าง ๆ	8
ภาพที่ 4 ภาพแสดงขั้นตอนการตรวจจับและปรับปรุงเส้นดาวเทียมจาก U-Net และการแปลงเชิงน่าจะเป็นของฮัพ	9
ภาพที่ 5 ตัวอย่างเฟรมภาพจากไฟล์วิดีโอที่ได้บันทึกด้วยกล้องโทรทรรศน์	12
ภาพที่ 6 โปรแกรมในส่วนการโหลดและเตรียมข้อมูลภาพ	13
ภาพที่ 7 โปรแกรมในส่วนการปรับค่าความเข้มของภาพ	13
ภาพที่ 8 โปรแกรมในส่วนการปรับขนาดภาพให้เหมาะสมกับโมเดล	14
ภาพที่ 9 โปรแกรมในส่วนการปรับปรุงผลลัพธ์ของการแบ่งส่วน	14
ภาพที่ 10 โปรแกรมในส่วนการลบวัตถุขนาดเล็ก	15
ภาพที่ 11 ภาพผลลัพธ์ที่ได้จากการสกัดเส้นด้วยโมเดล U-Net	16
ภาพที่ 12 โปรแกรมในการสกัดเส้นตรงหลักของวิถีดาวเทียม	19
ภาพที่ 13 โปรแกรมในส่วนการปรับปรุงผลลัพธ์ของการแบ่งส่วน	19
ภาพที่ 14 โปรแกรมในส่วนการปรับปรุงผลลัพธ์ของการแบ่งส่วน	20
ภาพที่ 15 โปรแกรมในส่วนการคำนวณจุดศูนย์กลางของพิกเซลสีขาวในแผนผังลักษณะฐานสอง	21
ภาพที่ 16 โปรแกรมในส่วนการกำหนดทิศทางของเส้นตามลำดับเวลา	21
ภาพที่ 17 โปรแกรมในส่วนการกำหนดทิศทางของเส้นตามลำดับเวลา	22
ภาพที่ 18 โปรแกรมในส่วนการแสดงผลเป็นกราฟ	23
ภาพที่ 19 โปรแกรมในส่วนการทำงานระบบหลักในการสกัดลักษณะภาพ	28
ภาพที่ 20 โปรแกรมในส่วนการแปลงพิกัดศูนย์กลางภาพให้อยู่ในหน่วยเชิงองศา	28
ภาพที่ 21 โปรแกรมในส่วนการคำนวณความละเอียดเชิงมุมต่อพิกเซล	29
ภาพที่ 22 โปรแกรมในส่วนการกำหนดระบบพิกัดเชิงโลก	29
ภาพที่ 23 โปรแกรมในส่วนการกำหนดระบบพิกัดเชิงโลก	30
ภาพที่ 24 โปรแกรมในส่วนการแปลงพิกัดมุมภาพจากพิกเซล	30
ภาพที่ 25 โปรแกรมในส่วนการคำนวณขอบเขตเชิงมุมของภาพ	31

ภาพที่ 26 โปรแกรมในส่วนการดึงข้อมูลบัญชีรายชื่อดาวเทียมจาก CelesTrak.....	31
ภาพที่ 27 โปรแกรมในส่วนการแปลงพิกัดภูมิศาสตร์เป็นรูปแบบเวกเตอร์สามมิติภายใต้กรอบอ้างอิงที่มีจุดกำเนิดที่ศูนย์กลางโลก.....	33
ภาพที่ 28 โปรแกรมในส่วนการแปลงตำแหน่งดาวเทียมสู่กรอบผู้สังเกต	33
ภาพที่ 29 โปรแกรมในส่วนการแปลงเป็นเป็นพิกัดเชิงมุม	34
ภาพที่ 30 โปรแกรมในส่วนการคำนวณตำแหน่งดาวเทียม ณ เวลาเริ่มต้น	34
ภาพที่ 31 โปรแกรมในส่วนดำเนินการคำนวณเพื่อคัดกรองดาวเทียมภายในขอบเขตเชิงพื้นที่.....	35
ภาพที่ 32 โปรแกรมในส่วนการคำนวณตำแหน่งดาวเทียม ณ เวลาสิ้นสุด.....	35
ภาพที่ 33 โปรแกรมในส่วนการจัดเก็บข้อมูลของดาวเทียมทั้งหมดที่พบ.....	36
ภาพที่ 34 โปรแกรมส่วนของการดึงข้อมูลขนาดภาพจากไฟล์ข้อมูลอภิปันธุ์.....	36
ภาพที่ 35 กลุ่มของหน้าภาพฐานสองที่ได้จากทำนายของโมเดลจากชุดข้อมูล GALILEO 11 เฟรมที่ 33-39	38
ภาพที่ 36 กลุ่มของหน้าภาพฐานสองที่ได้จากทำนายของโมเดลจากชุดข้อมูล NAVSTAR 85 เฟรมที่ 23-28	39
ภาพที่ 37 ภาพรวมของภาพฐานสองที่ได้จากหน้าภาพทำนายในชุดข้อมูล GALILEO.....	39
ภาพที่ 38 ภาพรวมของภาพฐานสองที่ได้จากหน้าภาพทำนายในชุดข้อมูล NAVSTAR 85	40
ภาพที่ 39 ภาพผลลัพธ์ที่ได้จากการรวมเส้นในชุดข้อมูล GALILEO.....	40
ภาพที่ 40 ภาพผลลัพธ์ที่ได้จากการรวมเส้นในชุดข้อมูล NAVSTAR 85.....	40
ภาพที่ 41 ผลการตรวจจับเส้นในภาพและการซ้อนทับกับเส้นทางที่คำนวณได้.....	43
ภาพที่ 42 กราฟการจัดอันดับดาวเทียมที่เป็นไปได้ของชุดข้อมูล NAVSTAR 85 (USA 581).....	43
ภาพที่ 43 ผลการตรวจจับเส้นและการเปรียบเทียบกับ trajectory ที่คำนวณได้	44
ภาพที่ 44 กราฟลำดับของดาวเทียมที่เป็นไปได้ในชุดข้อมูล GSAT0208 (GALILEO 11).....	45
ภาพที่ 45 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ โดยไม่มีดาวเทียมปรากฏ.....	45
ภาพที่ 46 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ เมื่อแก้ไขพิกัดกลางกล้องแล้ว...46	
ภาพที่ 47 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ โดยไม่มีดาวเทียม Gaofen-13 01 ปรากฏ	46
ภาพที่ 48 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ เมื่อแก้ไขพิกัดกลางกล้องแล้ว...47	
ภาพที่ 49 กราฟจำลองวิถีดาวเทียมทั้ง 4 ดวงที่ตรวจพบในชุดข้อมูล Zhongxing-12	47
ภาพที่ 50 รายการดาวเทียมที่เป็นไปได้ของดาวเทียมทั้ง 4 แต่ไม่สามารถแยกดาวเทียมเป้าหมายได้ชัดเจน.....	48
ภาพที่ 51 กราฟแสดงวิถีโคจรของเทียมที่คาดว่าจะโคจรผ่าน.....	49

บทที่ 1

บทนำ

1.1 ความเป็นมาและเหตุผลการวิจัย

ปัจจุบันวัตถุที่โคจรรอบโลก หรือ Resident Space Objects (RSOs) มีจำนวนเพิ่มขึ้นอย่างต่อเนื่อง อันเป็นผลมาจากการปล่อยดาวเทียมเพื่อการสื่อสาร การสำรวจทรัพยากรโลก การพยากรณ์อากาศ ตลอดจนภารกิจทางวิทยาศาสตร์และความมั่นคงของประเทศต่าง ๆ การเพิ่มขึ้นของวัตถุในวงโคจรดังกล่าวส่งผลให้การติดตาม ตรวจสอบ และระบุวัตถุเหล่านี้มีความสำคัญอย่างยิ่งต่อการดำเนินงานด้านการรับรู้สถานการณ์อวกาศ (Space Situational Awareness: SSA) ซึ่งเป็นกลไกสำคัญในการเฝ้าระวังตำแหน่ง การเคลื่อนที่ และพฤติกรรมของวัตถุในอวกาศอย่างเป็นระบบและแม่นยำ โดยนอกจากมิติด้านวิทยาศาสตร์และการใช้งานเชิงพาณิชย์แล้ว การรับรู้สถานการณ์อวกาศยังมีบทบาทสำคัญต่อความมั่นคงทางอวกาศของประเทศ โดยเฉพาะอย่างยิ่งในบริบทที่ดาวเทียมบางประเภท เช่น ดาวเทียมสอดแนม หรือดาวเทียมทางทหาร อาจมีภารกิจที่เกี่ยวข้องกับการเก็บรวบรวมข้อมูลเชิงยุทธศาสตร์ การมีขีดความสามารถในการตรวจสอบ ติดตาม และวิเคราะห์วัตถุที่ไม่ทราบแหล่งที่มา หรือมีพฤติกรรมผิดปกติในวงโคจร จึงเป็นองค์ประกอบหนึ่งของการเสริมสร้างอธิปไตยและความมั่นคงด้านอวกาศของประเทศ รวมถึงการลดความเสี่ยงจากภัยคุกคามที่อาจเกิดขึ้นทั้งในเชิงกายภาพและเชิงข้อมูลข่าวสาร

การสังเกตการณ์วัตถุในวงโคจรด้วยกล้องโทรทรรศน์เชิงแสงเป็นหนึ่งในวิธีการสำคัญที่ถูกนำมาใช้ในการตรวจจับและติดตามวัตถุเหล่านี้ โดยข้อมูลที่ได้จะอยู่ในรูปของภาพถ่ายหรือวิดีโอทางดาราศาสตร์ เพื่อใช้ในการชี้กล้องและบันทึกข้อมูล ข้อจำกัดสำคัญของกระบวนการในปัจจุบันคือความสามารถในการตรวจจับวัตถุที่ไม่ทราบมาก่อนนั้นยังทำได้ยาก เนื่องจากระบบส่วนใหญ่ถูกออกแบบมาเพื่อการติดตามวัตถุที่มีข้อมูลวงโคจรอยู่แล้ว การค้นพบหรือระบุวัตถุใหม่จึงต้องอาศัยหลายขั้นตอนและพึ่งพาการตรวจสอบโดยมนุษย์หรือระบบกึ่งอัตโนมัติเป็นหลัก ส่งผลให้ต้องใช้เวลาและทรัพยากรจำนวนมาก โดยเฉพาะอย่างยิ่งเมื่อปริมาณข้อมูลมีขนาดใหญ่และเพิ่มขึ้นอย่างรวดเร็ว ซึ่งเป็นอุปสรรคต่อการพัฒนาระบบเฝ้าระวังอวกาศแบบอัตโนมัติในระยะยาว

ในขณะเดียวกัน ความก้าวหน้าทางด้านเทคนิคการประมวลผลภาพ (image processing) และการเรียนรู้ของเครื่อง (machine learning) ได้เปิดโอกาสให้สามารถสกัดข้อมูลเชิงลึกจากภาพถ่ายทางดาราศาสตร์ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น งานวิจัยนี้จึงมีแนวคิดในการพัฒนาระบบสำหรับการระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์ โดยอาศัยการสกัดลักษณะเฉพาะของ

วัตถุ (feature extraction) ร่วมกับแบบจำลองการเรียนรู้ของเครื่อง เพื่อให้สามารถตรวจหา ติดตาม และจำแนกวัตถุที่เคลื่อนที่ในวงโคจรรอบโลกได้โดยอัตโนมัติ นอกจากนี้ ระบบที่พัฒนาขึ้นยังสามารถ นำผลลัพธ์ที่ได้ไปเปรียบเทียบกับฐานข้อมูลบัญชีรายชื่อดาวเทียม (satellite catalog) เพื่อใช้ในการ ระบุวัตถุที่ตรวจพบ รวมถึงแยกแยะวัตถุที่อาจไม่ปรากฏอยู่ในฐานข้อมูล ซึ่งมีความสำคัญต่อทั้ง งานวิจัยทางดาราศาสตร์และการเฝ้าระวังด้านความมั่นคงทางอวกาศ โดยผลลัพธ์ของงานวิจัยนี้จะ ช่วยลดภาระในการวิเคราะห์ข้อมูลโดยมนุษย์ เพิ่มความรวดเร็วและความแม่นยำในการประมวลผล ข้อมูลจำนวนมาก และเป็นพื้นฐานสำคัญสำหรับการพัฒนาระบบรับรู้สถานการณ์อวกาศของประเทศ ในอนาคต

1.2 วัตถุประสงค์ของการวิจัย

1. พัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะแ การเรียนรู้ของเครื่อง เพื่อทำการคัดแยกวัตถุออกเป็นดาวเคราะห์ในระบบสุริยะ ดาวฤกษ์ นอกกระบบสุริยะ ดาวเทียม และวัตถุอื่น ๆ
2. พัฒนาวิธีการคำนวณตำแหน่งของวัตถุที่ตรวจพบ และเปรียบเทียบกับข้อมูลในบัญชีรายชื่อ

1.3 ขอบเขตการวิจัย

1. ระบบที่พัฒนาจะถูกออกแบบให้ทำงานกับข้อมูลภาพถ่ายที่ได้จากสถาบันวิจัยดาราศาสตร์ แห่งชาติ (NARIT) จัดหาให้เป็นหลัก โดยข้อมูลภาพถ่ายจะอยู่ในรูปแบบของไฟล์ Audio Video Interleave (AVI) ขนาด 3-5 GB
2. ระบบจะถูกพัฒนาในภาษา Python บนสภาพแวดล้อมสำหรับการคำนวณเชิงปฏิสัมพันธ์ (interactive computing environment)
3. ระบบใช้ข้อมูล TLE (Two-Line Element) จาก Celestrak ซึ่งอ้างอิงบัญชีรายชื่อ ดาวเทียม ของ NORAD (North American Aerospace Defense Command) โดย ดาวเทียมที่ใช้คือดาวเทียมที่ยังใช้งานอยู่ในวงโคจร (active satellites) ซึ่งมีข้อมูลดาวเทียม มากกว่า 13,000 ดวง และแต่ละรายการประกอบด้วยชื่อดาวเทียม NORAD ID และ TLE สำหรับคำนวณตำแหน่ง
4. ผลลัพธ์ที่ได้จากระบบนั้นจะประกอบด้วย 2 กรณี:
 - 4.1. กรณีที่วัตถุสามารถเปรียบเทียบกับบัญชีรายชื่อได้
 - 4.1.1. ข้อมูลการระบุดาวเทียม: ชื่อดาวเทียม และ NORAD ID
 - 4.1.2. พิกัดวัตถุ (RA/Dec)
 - 4.1.3. ระยะทาง (กิโลเมตร): ระยะห่างจากจุดสังเกตการณ์

- 4.1.4. สถิติความคลาดเคลื่อน: ความแตกต่างเชิงมุม และความแตกต่างเชิงตำแหน่งในรูปแบบร้อยละ
- 4.1.5. รายละเอียดตามจุด: ความคลาดเคลื่อนและพิกัดที่คำนวณได้ของแต่ละจุดใน track
- 4.1.6. ระดับความมั่นใจ (confidence level): กำหนดจากร้อยละสถิติความคลาดเคลื่อน
- 4.2. กรณีที่วัตถุไม่สามารถเปรียบเทียบกับบัญชีรายชื่อได้
 - 4.2.1. ข้อมูลการระบุดาวเทียม: ชื่อดาวเทียม และ NORAD ID
 - 4.2.2. พิกัดวัตถุ (RA/Dec):
 - 4.2.3. ระยะทาง (กิโลเมตร): ระยะห่างจากจุดสังเกตการณ์
 - 4.2.4. สถิติความคลาดเคลื่อน: ความแตกต่างเชิงมุม และเชิงตำแหน่งในรูปแบบร้อยละ
 - 4.2.5. รายละเอียดตามจุด: ความคลาดเคลื่อนและพิกัดที่คำนวณได้ของแต่ละจุดใน track
 - 4.2.6. ระดับความมั่นใจ (confidence level): กำหนดจากร้อยละสถิติความคลาดเคลื่อน

1.4 ขั้นตอนการวิจัย

1. กำหนดหัวข้อโครงการที่จะศึกษา
 - 1.1. ปรึกษากับนักวิจัยจากสถาบันวิจัยดาราศาสตร์แห่งชาติเพื่อให้ทราบถึงลักษณะของปัญหาที่พบเจอที่จะสามารถนำระบบคอมพิวเตอร์เข้ามาแก้ไขปัญหาได้
2. ศึกษาความรู้และเครื่องมือที่เกี่ยวข้องกับการพัฒนา
 - 2.1. ความรู้ทางดาราศาสตร์เกี่ยวกับการโคจร เทคโนโลยีดาวเทียม และบัญชีรายชื่อทางดาราศาสตร์
 - 2.2. ความรู้เกี่ยวกับเครื่องมือที่ใช้ในการพัฒนา เทคนิคการประมวลผลภาพ และแพ็คเกจที่เกี่ยวข้องกับการประมวลผลข้อมูลเชิงดาราศาสตร์
3. ออกแบบการทำงานของระบบ
 - 3.1. ส่วนการตรวจหา และคัดแยกวัตถุที่อยู่ในภาพ
 - 3.2. ส่วนการคำนวณตำแหน่งของวัตถุที่ตรวจพบ และเปรียบเทียบกับข้อมูลในบัญชีรายชื่อ
4. พัฒนาระบบตามที่ได้ออกแบบไว้
5. ทดสอบการทำงานของระบบ และปรับปรุงแก้ไขปัญหาของระบบที่พบระหว่างการทดสอบ
6. จัดทำรูปเล่มโครงการ

ตารางที่ 1 แผนภูมิแกนต์แสดงขั้นตอนการวิจัยและช่วงเวลาของแต่ละขั้นตอน

ขั้นตอนการดำเนินงาน	สิงหาคม 2568 - เมษายน 2569								
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
กำหนดหัวข้อโครงการที่จะศึกษา									
ศึกษาความรู้ที่เกี่ยวข้องของการพัฒนา									
ออกแบบการทำงานของระบบ									
พัฒนาระบบตามที่ออกแบบไว้									
ทดสอบและปรับปรุงการทำงานของระบบ									
จัดทำรูปแบบโครงการ									

1.5 ประโยชน์ที่ได้รับ

ประโยชน์ที่ได้รับจากการวิจัยในครั้งนี้มีดังนี้

1. ประโยชน์ต่อนิสิตที่ทำโครงการ

- 1.1. ได้ศึกษาการพัฒนาระบบคอมพิวเตอร์ด้วยภาษา Python และการใช้งานแพ็คเกจที่เกี่ยวข้องกับการประมวลผลภาพ
- 1.2. ได้ศึกษาความรู้ทางดาราศาสตร์ที่เกี่ยวข้องกับวัตถุบนท้องฟ้า วัตถุในวงโคจรโลก การจัดการข้อมูลภาพถ่าย ภาพเคลื่อนไหวทางดาราศาสตร์ และการใช้งานข้อมูลจากบัญชีรายชื่อทางดาราศาสตร์
- 1.3. ได้ประสบการณ์ในการพัฒนาระบบคอมพิวเตอร์เชิงดาราศาสตร์ สามารถนำไปประยุกต์ต่อในอนาคตได้

2. ประโยชน์ที่ได้จากโครงการที่พัฒนาขึ้น

- 2.1. ได้ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง สำหรับสถาบันวิจัยดาราศาสตร์แห่งชาติ
- 2.2. ช่วยลดภาระและทรัพยากรที่ต้องใช้ในการประมวลผลข้อมูลจากภาพถ่ายทางดาราศาสตร์ ทำให้บุคลากรสามารถดำเนินงานวิจัยได้รวดเร็วและถูกต้องแม่นยำมากยิ่งขึ้น
- 2.3. บันทึกวัตถุที่อาจค้นพบขึ้นใหม่ลงในบัญชีรายชื่อทางดาราศาสตร์

1.6 โครงสร้างของรายงาน

บทที่ 2 กล่าวถึงงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์

บทที่ 3 กล่าวถึงขั้นตอนการพัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์ ซึ่งจะประกอบไปด้วย ภาพถ่ายทางดาราศาสตร์ที่ใช้ในการทดสอบ การตรวจหาวัตถุในภาพ การแปลงพิกัดวัตถุในภาพเป็นพิกัดท้องฟ้า การเปรียบเทียบกับดาวเทียมที่มีอยู่ในบัญชีรายชื่อดาวเทียม การแสดงผลข้อมูล และการประเมินประสิทธิภาพของระบบ

บทที่ 4 กล่าวถึงผลการประเมินประสิทธิภาพของระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์

บทที่ 5 กล่าวถึงข้อสรุปของงานวิจัย และข้อเสนอแนะ

บทที่ 2

งานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับระบบตรวจหาและตรวจสอบดาวเทียมในบัญชีรายชื่อ เพื่อช่วยเพิ่มความแม่นยำในการระบุและติดตามดาวเทียม รวมถึงสนับสนุนการวิเคราะห์ข้อมูลภาพดาราศาสตร์ได้อย่างมีประสิทธิภาพ

2.1 การตรวจหารอยทางของดาวเทียมโดยใช้โมเดล U-Net

การตรวจจับเส้นของดาวเทียมในภาพดาราศาสตร์เป็นปัญหาที่มีความซับซ้อน เนื่องจากภาพประกอบไปด้วยวัตถุจำนวนมาก เช่น ดาวฤกษ์ กาแล็กซี และสัญญาณรบกวน ซึ่งอาจมีลักษณะคล้ายเส้น ทำให้การแยกเส้นของดาวเทียมออกจากพื้นหลังทำได้ยาก

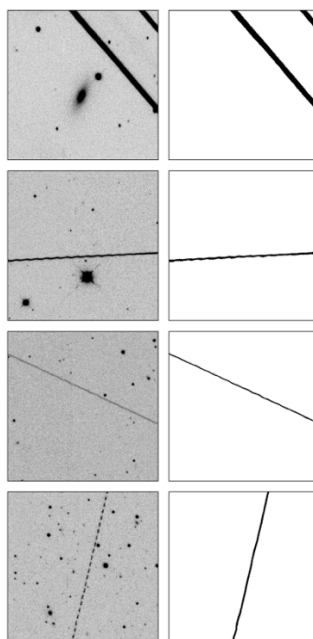
งานวิจัยของ Stoppa et al. (2024) ได้เสนอวิธีการตรวจจับเส้นดาวเทียมโดยใช้โครงสร้าง U-Net ซึ่งเป็น โครงข่ายประสาทเชิงสังวัตนาการ (convolutional neural network) สำหรับการแบ่งส่วนภาพเชิงความหมาย (semantic segmentation) โดยสามารถจำแนกพิกเซลแต่ละตำแหน่งในภาพว่าเป็นส่วนของเส้นดาวเทียมหรือไม่

ในเชิงเทคนิค U-Net ประกอบด้วยสองส่วนหลัก ได้แก่

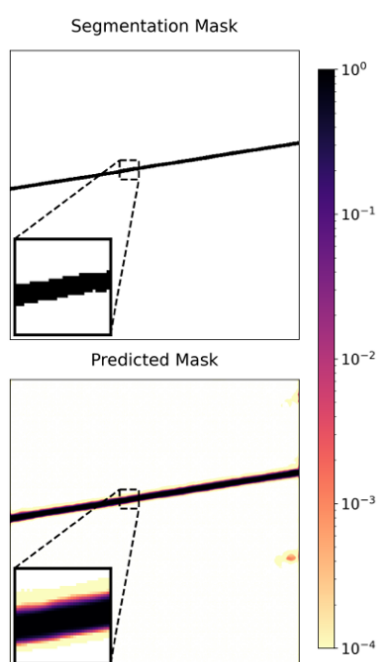
- Contracting path เป็นตัวเข้ารหัสสำหรับดึงลักษณะของภาพในหลายระดับ
- Expansive path เป็นตัวถอดรหัสสำหรับคืนรายละเอียดเชิงพื้นที่

พร้อมทั้งใช้ skip connections เพื่อรักษาข้อมูลเชิงตำแหน่งของวัตถุในภาพ ทำให้สามารถตรวจจับเส้นที่บางหรือไม่ต่อเนื่องได้อย่างมีประสิทธิภาพ

โมเดลถูกฝึกด้วยชุดข้อมูลที่มีข้อมูลประกอบแบบระดับพิกเซล โดยใช้เครื่องมือ LABKIT ซึ่งช่วยสร้างหน้ากากความจริงพื้นฐานสำหรับการแบ่งส่วนภาพที่มีคุณภาพสูง นอกจากนี้ยังมีการแบ่งภาพขนาดใหญ่ (full-field) ออกเป็นแพทช์ ขนาด 528×528 พิกเซล เพื่อเพิ่มประสิทธิภาพในการฝึกโมเดล และลดภาระหน่วยความจำ

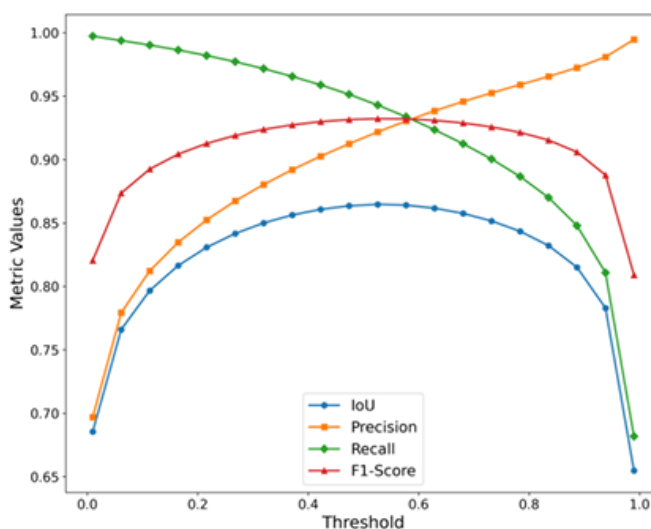


ภาพที่ 1 กลุ่มภาพเปรียบเทียบภาพเนกาทีฟสีตัวอย่างในฝั่งซ้ายกับกลุ่มภาพหน้ากากจริงในฝั่งขวา
(อ้างอิง: Automated detection of satellite trails
in ground-based observations using U-Net and Hough transform)



ภาพที่ 2 ภาพเปรียบเทียบหน้ากากจริงกับผลลัพธ์ U-Net พร้อมแสดงความน่าจะเป็นของพิกเซลและ
รายละเอียดของเส้น

(อ้างอิง: Automated detection of satellite trails
in ground-based observations using U-Net and Hough transform)



ภาพที่ 3 กราฟแสดงค่าประสิทธิภาพของ U-Net ที่ค่าขีดแบ่งต่าง ๆ

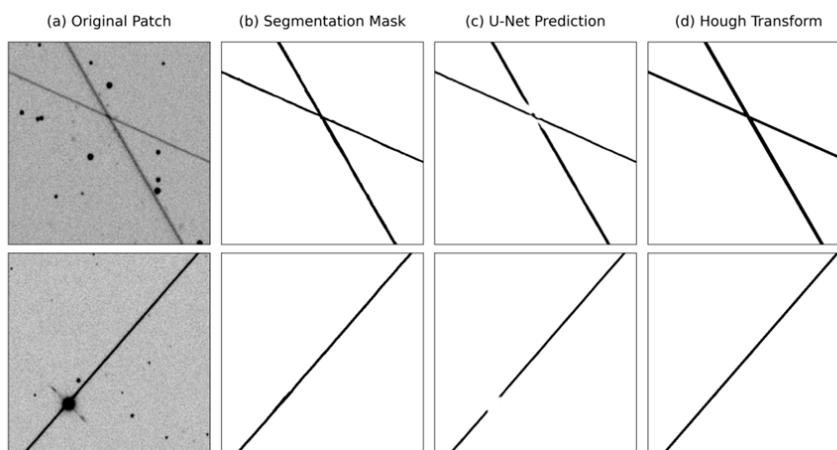
(อ้างอิง: Automated detection of satellite trails

in ground-based observations using U-Net and Hough transform)

โดยผลลัพธ์ของโมเดลอยู่ในรูปแบบ ภาพแปลงความน่าจะเป็น (probability map) ที่ค่า 0–1 ซึ่งจะถูกแปลงเป็นหน้ากากลักษณะฐานสอง (binary mask) ด้วยค่าขีดแบ่งที่ประมาณ 0.58 เพื่อแยกเส้นดาวเทียมออกจากพื้นหลัง โดยโมเดลสามารถให้ค่า precision และ recall ประมาณ 0.94 แสดงถึงประสิทธิภาพในการตรวจจับที่สูง

2.2 การแบ่งละเอียดด้วยการแปลงเชิงน่าจะเป็นของฮัฟ

เพื่อแก้ปัญหาเส้นที่ไม่ต่อเนื่องจากผลลัพธ์ของโมเดล U-Net งานวิจัยได้ใช้ การแปลงเชิงน่าจะเป็นของฮัฟ เป็นขั้นตอนการแบ่งละเอียดเพื่อเชื่อมเส้นที่ขาดและปรับให้เป็นเส้นตรงที่สมบูรณ์ ดังภาพที่ 4



ภาพที่ 4 ภาพแสดงขั้นตอนการตรวจจับและปรับปรุงเส้นดาวเทียมจาก U-Net และการแปลงเชิง
นจะเป็นของฮัฟ

(อ้างอิง: Automated detection of satellite trails

in ground-based observations using U-Net and Hough transform)

2.2 การวิเคราะห์เส้นขอบ และการสกัดลักษณะ

หลังจากได้ผลลัพธ์ที่ผ่านการแบ่งละเอียดแล้ว จะได้หน้าภาพลักษณะฐานสองที่ประกอบด้วยเส้นของดาวเทียมอย่างชัดเจน ขั้นตอนถัดไปคือการสกัดลักษณะ (feature extraction) เพื่อใช้ในการวิเคราะห์และระบุตัวตนของดาวเทียม งานวิจัยใช้วิธีการการตรวจหาเส้นขอบ โดยใช้ฟังก์ชัน `cv2.findContours` เพื่อระบุขอบเขตของเส้นดาวเทียมในภาพ จากนั้นสามารถคำนวณลักษณะต่าง ๆ ได้แก่

- ความยาวของเส้น
- ความกว้าง
- จุดเริ่มต้นและจุดสิ้นสุด
- มุมเอียงของเส้น
- ความสว่าง

ในกรณีที่มีเส้นหลายเส้นตัดกัน (intersecting trails) เส้นขอบที่ได้จะรวมหลายเส้นเข้าด้วยกัน งานวิจัยจึงใช้การจับกลุ่มแบบ DBSCAN บนมุมของเส้นที่ได้จากการแปลงของฮัฟ เพื่อแยกเส้นออกเป็นหลายกลุ่มอย่างถูกต้อง ขั้นตอนนี้ช่วยให้สามารถแยกเส้นที่ซ้อนกัน และสกัดลักษณะของแต่ละเส้นได้อย่างแม่นยำ

สำหรับโครงการนี้ ขั้นตอนการสกัดลักษณะ มีความสำคัญอย่างยิ่ง เนื่องจากข้อมูลที่ได้จะถูกนำไปใช้ในการคำนวณทิศทางการเคลื่อนที่ (motion vector) หาสมการเส้น เชื่อมโยงกับข้อมูลดาวเทียม เช่น TLE หรือข้อมูลจากบัญชีรายชื่อ

2.2 ชุดองค์ประกอบสองบรรทัด และการใช้งาน

งานของ Vallado และ Cefola (2012) ได้อธิบายถึงบทบาท ความสามารถ และข้อจำกัด ของชุดข้อมูลวงโคจรดาวเทียมแบบ Two-Line Element Sets หรือ TLE ซึ่งเป็นรูปแบบข้อมูล มาตรฐานที่ใช้กันอย่างแพร่หลายในงานด้านพลศาสตร์อวกาศมาเป็นเวลาหลายทศวรรษ โดย TLE ถูก ออกแบบให้มีลักษณะกะทัดรัดและสามารถใช้ร่วมกับแบบจำลองการเคลื่อนที่วงโคจรแบบ simplified general perturbations 4 (SGP4) เพื่อคำนวณตำแหน่งดาวเทียมได้อย่างรวดเร็วในระดับความ แม่นยำปานกลาง ในเชิงประวัติศาสตร์ แบบจำลอง SGP ถูกพัฒนาขึ้นตั้งแต่ช่วงทศวรรษ 1960 และ ได้รับการปรับปรุงต่อเนื่องจนกลายเป็น SGP4 ซึ่งเป็นมาตรฐานที่ใช้กันจริงในระบบติดตามวัตถุ อวกาศของหน่วยงานต่าง ๆ โดยเฉพาะระบบของกองทัพอากาศสหรัฐ (AFSPC/JSPOC) ซึ่งเป็นผู้ผลิต และเผยแพร่ TLE อย่างต่อเนื่องผ่านฐานข้อมูลสาธารณะ เช่น Space-Track และ Celestrak อย่างไรก็ตาม แบบจำลอง SGP4 ยังอิงกับทฤษฎีและสมมติฐานเกี่ยวกับสภาพแวดล้อมอวกาศในยุคแรก เช่น แบบจำลองแรงต้านบรรยากาศแบบง่าย และการละเลยแรงรบกวนบางประเภท ส่งผลให้เกิดข้อจำกัด ด้านความแม่นยำ

ในด้านความแม่นยำ งานวิจัยหลายฉบับพบว่า TLE มีความน่าเชื่อถือในช่วงเวลาสั้น ๆ โดยทั่วไปไม่เกินไม่กี่วัน เนื่องจากข้อผิดพลาดจากการประมาณค่าและแรงรบกวนที่ไม่ได้ถูกจำลอง เช่น แรงดันรังสีดวงอาทิตย์ การเปลี่ยนแปลงของบรรยากาศ และโดยเฉพาะอย่างยิ่งการเปลี่ยนวง โคจร (maneuver) ของดาวเทียม ซึ่งไม่สามารถตรวจจับได้ล่วงหน้าในระบบแบบพาสซีฟ (passive) นอกจากนี้ ความคลาดเคลื่อนมักเด่นชัดในทิศทางตามรอบทาง และอาจมีขนาดระดับกิโลเมตรเมื่อทำ การพยากรณ์ระยะยาว อีกข้อจำกัดสำคัญของ TLE คือการไม่มีข้อมูลความแปรปรวนร่วม (covariance) ซึ่งจำเป็นต่อการวิเคราะห์ความไม่แน่นอน เช่น การประเมินความเสี่ยงการชนกัน (conjunction analysis) แม้ว่าจะมีความพยายามจากหลายองค์กร เช่น ESA, CNES และ Aerospace Corporation ในการประมาณความแปรปรวนร่วมจากชุด TLE แต่ผลลัพธ์ยังมีข้อจำกัด เนื่องจากข้อมูล TLE ผ่านกระบวนการการกำหนดวงโคจร (orbit determination) มาแล้ว ทำให้ สูญเสียข้อมูลเชิงสถิติจากการสังเกตจริงบางส่วน

ในด้านการปรับปรุงคุณภาพของ TLE มีการเสนอแนวทางหลายรูปแบบ ได้แก่ การปรับปรุง แบบจำลองทางคณิตศาสตร์ของ SGP4 โดยเพิ่มแรงรบกวนเพิ่มเติม เช่น tesseral harmonics และ higher-order zonal terms รวมถึงการใช้วิธีการกำหนดวงโคจร โดยอาศัยข้อมูลเอเฟเมอริส (ephemeris) หรือแม้แต่ TLE เองเป็นการสังเกต เพื่อสร้างค่าประมาณใหม่ที่มีความแม่นยำสูงขึ้น งานวิจัยพบว่าวิธีการกำหนดวงโคจรสามารถช่วยปรับปรุงความแม่นยำได้ โดยเฉพาะเมื่อมีข้อมูล จำนวนมากและครอบคลุมช่วงเวลาที่เพียงพอ อย่างไรก็ตาม ประสิทธิภาพของวิธีดังกล่าวขึ้นอยู่กับ ประเภทวงโคจร ความถี่ของข้อมูล และคุณภาพของข้อมูลที่ใช้

นอกจากนี้ ยังมีการศึกษาการแปลงเอเฟเมอริส แบบเชิงตัวเลขไปเป็น TLE ซึ่งสามารถทำได้ทั้งแบบจุดเดียว และการเลือกตัวอย่างวิถีโคจร โดยพบว่าวิธีการเลือกตัวอย่างวิถีโคจร ให้ผลลัพธ์ที่แม่นยำกว่าในภาพรวม แต่ยังคงมีการสูญเสียความแม่นยำเมื่อเทียบกับเอเฟเมอริส ดังเดิมโดยเฉพาะในวงโคจรระดับต่ำ (low earth orbit: LEO) ในเชิงการใช้งาน TLE ยังคงถูกนำมาใช้ในหลายบริบท เช่น การคัดกรองเบื้องต้นของการชนกันของวัตถุอวกาศ การศึกษาบรรยากาศ และการตรวจจับการเปลี่ยนวงโคจรของดาวเทียม อย่างไรก็ตาม การใช้งานในระดับปฏิบัติการที่ต้องการความแม่นยำสูง จำเป็นต้องใช้ข้อมูลหรือแบบจำลองเพิ่มเติมร่วมด้วย

โดยสรุป แม้ว่า TLE จะเป็นเครื่องมือที่มีความสำคัญและใช้งานอย่างแพร่หลาย เนื่องจากความสะดวกและการเข้าถึงข้อมูล แต่ยังมีข้อจำกัดสำคัญด้านความแม่นยำและความไม่แน่นอน งานวิจัยส่วนใหญ่จึงมุ่งเน้นไปที่การปรับปรุงความแม่นยำ การสร้างความแปรปรวนร่วม และการผสานข้อมูลจากแหล่งอื่น เพื่อให้สามารถรองรับความต้องการของระบบปฏิบัติการอวกาศยุคใหม่ได้อย่างมีประสิทธิภาพ

บทที่ 3

วิธีการวิจัย

ในบทนี้จะกล่าวถึงวิธีการวิจัยระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์ตั้งแต่การดำเนินการที่เกี่ยวข้องกับภาพถ่าย ได้แก่ ภาพที่ใช้ในการทดสอบ การประมวลผลภาพก่อน การเชื่อมเส้นและสกัดเส้นตรงของวิถีดาวเทียม การคำนวณสมการเส้นและทิศทางการเคลื่อนที่ การคำนวณเวกเตอร์การเคลื่อนที่ และการแสดงผลจากการประมวลผลทางภาพ และยังรวมไปถึงขั้นตอนการดำเนินการที่เกี่ยวข้องกับวิธีวงโคจรของดาวเทียม และการดึงข้อมูลที่เกี่ยวข้อง อันประกอบไปด้วยการกำหนดขอบเขตการมองเห็นของระบบ การจัดเตรียมข้อมูลดาวเทียมย้อนหลังจากบัญชีรายชื่อการกำหนดกรอบเวลาและตำแหน่ง การคัดกรองและติดตามดาวเทียมในขอบเขตภาพ และการประเมินเปรียบเทียบข้อมูลวิถีดาวเทียมที่ได้จากการคำนวณและผลที่ได้จากการสกัดข้อมูลจากภาพ

3.1 การรวบรวมข้อมูลที่ใช้ตรวจสอบ

ภาพที่ใช้ในการทดสอบในงานวิจัยนี้ได้มาจากวิดีโอที่บันทึกด้วยกล้องโทรทรรศน์ โดยทำการบันทึกในวันที่ 15 กุมภาพันธ์ 2026, 22 กุมภาพันธ์ 2026 และ 19 มีนาคม 2026 ซึ่งไฟล์วิดีโอทั้งหมดอยู่ในรูปแบบ AVI มีความละเอียดสูงที่ 8288×5644 พิกเซลต่อเฟรม ภายในวิดีโอประกอบด้วยวัตถุท้องฟ้า ได้แก่ ดาวฤกษ์ (stars) ซึ่งปรากฏเป็นจุดแสง และเส้นวิถีของดาวเทียม (satellite trails) ซึ่งปรากฏเป็นเส้นสว่างต่อเนื่องในภาพ เนื่องจากการเคลื่อนที่ของดาวเทียมระหว่างช่วงเวลาการเปิดรับแสง กล้องที่ใช้ในการบันทึกข้อมูลคือ ZWO ASI294MM Pro ซึ่งเป็นกล้องประเภท Monochrome CMOS โดยตั้งค่าการบันทึกภาพในรูปแบบ RAW8 ดังภาพที่ 5



ภาพที่ 5 ตัวอย่างเฟรมภาพจากไฟล์วิดีโอที่บันทึกด้วยกล้องโทรทรรศน์

3.2 การประมวลผลภาพก่อน (Preprocessing)

เนื่องจากข้อมูลที่ใช้เป็นไฟล์วิดีโอประเภท Audio Video Interleave (AVI) ที่มีขนาดใหญ่ถึง 5–20 กิกะไบต์ต่อไฟล์ โดยขึ้นอยู่กับระยะเวลาการบันทึก จึงมีการดำเนินการคัดเลือกเฉพาะเฟรมที่มีเส้นวิถีของดาวเทียมปรากฏอยู่ เพื่อลดปริมาณข้อมูลและเพิ่มประสิทธิภาพในการประมวลผล โดยเฉพาะภายใต้ข้อจำกัดด้านฮาร์ดแวร์ดังต่อไปนี้

3.2.1 การโหลดและเตรียมข้อมูลภาพ

ในขั้นตอนแรก ทำการอ่านภาพจากไฟล์ที่ได้จากการแยกเฟรมของวิดีโอ โดยแปลงภาพให้อยู่ในรูปแบบสเกลสีเทา (grayscale) เพื่อให้เหมาะสมกับการประมวลผลเชิงภาพ ใช้ฟังก์ชัน `load_grayscale_image()` ดังภาพที่ 6 เพื่ออ่านภาพและแปลงเป็นสเกลสีเทาร่วมแปลงชนิดข้อมูลเป็น float สำหรับการคำนวณ

```
def load_grayscale_image(img_path):
    img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
    if img is None:
        raise ValueError(f"Cannot read image: {img_path}")
    return img.astype(np.float32)
```

ภาพที่ 6 โปรแกรมในส่วนการโหลดและเตรียมข้อมูลภาพ

3.2.2 การปรับค่าความเข้มของภาพ

เนื่องจากภาพแต่ละเฟรมอาจมีความสว่างแตกต่างกันจากสภาพแวดล้อมและการตั้งค่ากล้อง จึงมีการปรับค่าความเข้มของภาพให้อยู่ในรูปแบบมาตรฐานโดยการทำให้เป็นบรรทัดฐาน (normalization) โดยใช้ฟังก์ชัน `normalize_image()` ดังภาพที่ 7 ซึ่งทำการปรับค่าความเข้มของจุดภาพให้มีค่าเฉลี่ยเท่ากับ 0 และส่วนเบี่ยงเบนมาตรฐาน เท่ากับ 1 ขั้นตอนนี้ช่วยให้โมเดลสามารถเรียนรู้และประมวลผลได้อย่างเสถียรมากขึ้น

```
def normalize_image(img):
    mean = np.mean(img)
    std = np.std(img)
    if std < 1e-8:
        std = 1.0
    return (img - mean) / std
```

ภาพที่ 7 โปรแกรมในส่วนการปรับค่าความเข้มของภาพ

3.2.3 การปรับขนาดภาพให้เหมาะสมกับโมเดล

เนื่องจากโมเดลทำงานในรูปแบบแบ่งภาพเป็นแพทช์ (patch) ขนาดคงที่ จึงจำเป็นต้องปรับขนาดภาพให้สามารถแบ่งได้ลงตัว โดยใช้ฟังก์ชัน `pad_to_patch_size()` ดังภาพที่ 8 เพื่อเป็นการเติมการเสริมเต็ม (padding) ให้ขนาดของภาพสามารถหารด้วยขนาดของแพทช์ได้พอดี โดยไม่เกิดการสูญเสียข้อมูลเดิม

```
def pad_to_patch_size(img, patch_size):
    h, w = img.shape
    new_h = ((h + patch_size - 1) // patch_size) * patch_size
    new_w = ((w + patch_size - 1) // patch_size) * patch_size

    padded = np.zeros((new_h, new_w), dtype=img.dtype)
    padded[:h, :w] = img
    return padded, h, w
```

ภาพที่ 8 โปรแกรมในส่วนการปรับขนาดภาพให้เหมาะสมกับโมเดล

3.2.4 การปรับปรุงผลลัพธ์ของการแบ่งส่วน

โดยขั้นตอนนี้เป็นกระบวนการหลังการทำหน้ากาก (mask post-processing) หลังจากโมเดลทำการทำนายแล้ว จะได้ผลลัพธ์ในรูปแบบภาพแปลงความน่าจะเป็น ซึ่งยังไม่สามารถนำไปใช้งานได้โดยตรง จำเป็นต้องมีการแปลงและปรับปรุงให้เป็นหน้ากากฐานสอง ที่ชัดเจนโดยใช้ฟังก์ชัน `postprocess_mask()` ดังภาพที่ 9

```
def postprocess_mask(pred, threshold=0.25, min_area=10):
    prob = cv2.GaussianBlur(pred, (5,5), 0)
    binary = (prob > threshold).astype(np.uint8) * 255

    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (9,9))
    binary = cv2.morphologyEx(binary, cv2.MORPH_CLOSE, kernel)

    binary = remove_small_components(binary, min_area)
    return binary
```

ภาพที่ 9 โปรแกรมในส่วนการปรับปรุงผลลัพธ์ของการแบ่งส่วน

ผลลัพธ์ที่ได้จึงเป็นหน้ากากฐานสองที่มีเฉพาะโครงสร้างของเส้นวิถีดาวเทียมที่ชัดเจนและพร้อมสำหรับขั้นตอนถัดไป ซึ่งการดำเนินการดังกล่าวประกอบไปด้วยขั้นตอนย่อยดังต่อไปนี้

- การลดสัญญาณรบกวน (noise) ด้วยการเบลอแบบเกาส์เซียน (Gaussian blur) เพื่อลดสัญญาณรบกวนที่ปรากฏในภาพ
- การทำขีดแบ่ง (thresholding) โดยดำเนินการแปลงภาพแปลงความน่าจะเป็น ซึ่งเป็นภาพลักษณะฐานสอง (binary image) โดยใช้ค่าขีดแบ่งที่กำหนดไว้
- การเชื่อมเส้นด้วยการปิดเชิงสัณฐาน (morphological closing) โดยใช้การดำเนินการเชิงสัณฐานของภาพ (morphological operations) เพื่อเชื่อมเส้นที่ขาด และทำให้เส้นต่อเนื่องมากขึ้น
- การลบวัตถุขนาดเล็กโดยใช้ฟังก์ชัน `remove_small_components()` ดังภาพที่ 10 เพื่อลบสัญญาณรบกวน หรือวัตถุที่มีขนาดเล็กเกินไป ซึ่งไม่ใช่เส้นวิถีดาวเทียม

```
def remove_small_components(binary, min_area=20):
    num_labels, labels, stats, _ = cv2.connectedComponentsWithStats(binary)
    cleaned = np.zeros_like(binary)
    for i in range(1, num_labels):
        if stats[i, cv2.CC_STAT_AREA] >= min_area:
            cleaned[labels == i] = 255
    return cleaned
```

ภาพที่ 10 โปรแกรมในส่วนการลบวัตถุขนาดเล็ก

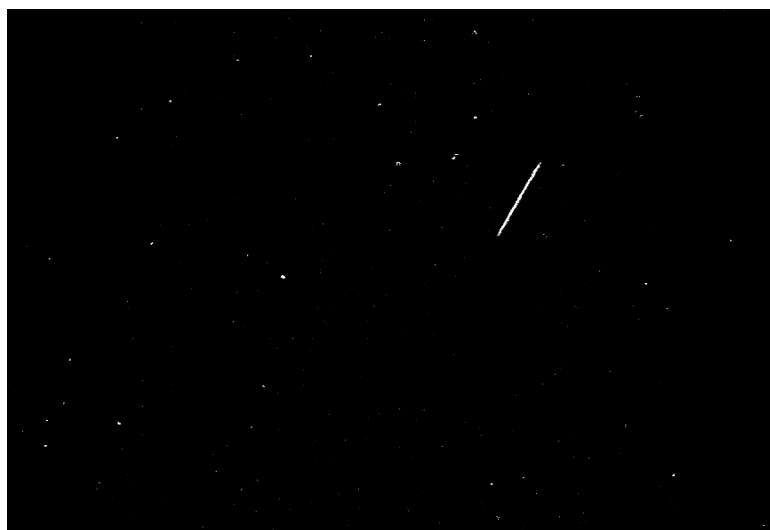
3.3 การเชื่อมเส้นและสกัดเส้นตรงของวิถีดาวเทียม

ในขั้นตอนนี้มีวัตถุประสงค์เพื่อสกัดเส้นวิถีของดาวเทียมจากภาพที่ผ่านการประมวลผลภาพก่อนแล้ว โดยอาศัยทั้งการเรียนรู้ของเครื่องในรูปแบบโมเดลการเรียนรู้เชิงลึก U-Net เพื่อให้สามารถตรวจจับเส้นที่มีลักษณะบาง ต่อเนื่อง และมีความแปรผันของความสว่างได้อย่างมีประสิทธิภาพ

3.3.1 การสกัดเส้นด้วยโมเดล U-Net

เมื่อได้ชุดภาพที่ต้องการตรวจสอบจากวิดีโอแล้ว จะนำภาพแต่ละเฟรมเข้าสู่โมเดล U-Net เพื่อทำการแบ่งส่วน (segmentation) ของเส้นวิถีดาวเทียม โดยโมเดล U-Net ที่ใช้ในงานนี้ถูกฝึกด้วยข้อมูลภาพถ่ายจริงจากกล้องโทรทรรศน์เชิงแสง MeerLICHT ในประเทศแอฟริกาใต้จำนวน 178 ภาพ โดยทำการทำป้ายด้วยมือ (manual labeling) อย่างละเอียด เพื่อให้ครอบคลุมลักษณะของเส้นในหลายรูปแบบ ได้แก่ เส้นที่มีความสว่างแตกต่างกัน (จางและสว่าง) เส้นที่มีความยาวแตกต่างกัน (สั้นและยาว) เส้นที่มีการบิดเบี้ยวจากสภาพบรรยากาศ (atmospheric distortion) ซึ่งการใช้ข้อมูลที่มีความหลากหลายนี้ช่วยให้โมเดลมีลักษณะทั่วไปที่ดี และสามารถตรวจจับเส้นดาวเทียมในสภาพแวดล้อมที่แตกต่างกันได้ ผลลัพธ์จากโมเดลจะอยู่ในรูปของภาพแปลงความน่าจะเป็นซึ่งแสดง

ความน่าจะเป็นที่แต่ละพิกเซลจะเป็นส่วนหนึ่งของเส้น จากนั้นจะถูกแปลงเป็นภาพลักษณะฐานสองผ่านขั้นตอนการดำเนินการภายหลัง เพื่อให้ได้เส้นที่มีความต่อเนื่อง และชัดเจน



ภาพที่ 11 ภาพผลลัพธ์ที่ได้จากการสกัดเส้นด้วยโมเดล U-Net

3.3.2 การรวมหน้ากาก (Mask Merging)

หลังจากได้หน้ากากฐานสอง ขอแต่ละเฟรมแล้ว จึงดำเนินการรวมภาพทั้งหมดเข้าด้วยกันโดยใช้วิธีการหาค่าสูงสุดรายพิกเซล (pixel-wise maximum) หรือการรวมด้วยค่าสูงสุด (max merging) ซึ่งแนวคิดของขั้นตอนนี้คือหากเส้นวิถีดาวเทียมปรากฏอยู่ในหลายเฟรม การรวมค่าจะทำให้เส้นสะสมกันจนมีความยาวมากขึ้น ดังนั้นผลลัพธ์คือภาพรวมหน้ากากที่มีเส้นวิถีดาวเทียมที่ยาวและต่อเนื่องมากขึ้น ซึ่งช่วยลดปัญหาเส้นขาดตอน และทำให้สามารถวิเคราะห์แนวเส้นได้ง่ายขึ้น

3.3.3 การเชื่อมเส้นและหาเส้นหลัก

เมื่อได้ภาพรวมหน้ากากแล้ว จากนั้นทำการสกัดเส้นตรงหลักของวิถีดาวเทียม โดยใช้ฟังก์ชัน `extract_straight_line()` โดยมีขั้นตอนย่อยดังต่อไปนี้

- การเชื่อมเส้น โดยการเชื่อมเส้นด้วยการปิดเชิงสัณฐาน ร่วมกับการแปลงระยะทาง (distance transform) เพื่อเชื่อมส่วนของเส้นที่อยู่ใกล้กันให้กลายเป็นโครงสร้างเดียวกัน
- การเลือกเส้นหลัก โดยใช้การวิเคราะห์ส่วนประกอบที่เชื่อมต่อกัน (connected component analysis) เพื่อแยกวัตถุออกเป็น ส่วน ๆ และเลือกส่วนประกอบที่มีลักษณะคล้ายเส้นมากที่สุด โดยพิจารณาจากพื้นที่ และความยาวต่อความกว้าง (elongation)

- การปรับเส้นตรง (line fitting) เมื่อได้ส่วนประกอบหลักแล้ว จะใช้แนวคิดของการวิเคราะห์องค์ประกอบหลัก (principal component analysis: PCA) เพื่อหาแนวแกนหลักของกลุ่มพิกเซล ซึ่งให้ทิศทางของเส้นตรง ดังภาพที่ 12

```
def extract_straight_line(mask, debug_prefix=None):

    dist_transform = cv2.distanceTransform(mask, cv2.DIST_L2, cv2.DIST_MASK_PRECISE)
    if debug_prefix:
        cv2.imwrite(f"{debug_prefix}_02_distance_transform.png", np.uint8(np.clip(dist_transform /
(dist_transform.max() + 1e-6) * 255, 0, 255)))

    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (DILATION_KERNEL, DILATION_KERNEL))
    connected = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel,
iterations=DILATION_ITERATIONS)
    if debug_prefix:
        cv2.imwrite(f"{debug_prefix}_03_connected.png", connected)
    # 2. find the most line-like connected component
    num_labels, labels, stats, _ = cv2.connectedComponentsWithStats(connected)
    best_label = None
    best_score = 0
    total_pixels = np.count_nonzero(mask)
    for i in range(1, num_labels):
        area = stats[i, cv2.CC_STAT_AREA]
        width = stats[i, cv2.CC_STAT_WIDTH]
        height = stats[i, cv2.CC_STAT_HEIGHT]
        elongation = max(width, height) / (min(width, height) + 1)
        score = area * elongation
        if score > best_score:
            best_score = score
            best_label = i
    if best_label is None:
        best_line = _hough_longest_line(mask)
        if best_line is not None:
            return _vector_from_line(best_line)
    return None
```

```

component = (labels == best_label).astype(np.uint8) * 255
if debug_prefix:
    cv2.imwrite(f"{debug_prefix}_05_selected_component.png", component)

# If the selected component is too small or not a clear line, fallback to Hough
if best_score < MIN_COMPONENT_PIXELS or (total_pixels > 0 and stats[best_label,
cv2.CC_STAT_AREA] / total_pixels < MIN_COMPONENT_RATIO):
    full_line = _hough_longest_line.connected)
    if full_line is not None:
        return _vector_from_line(full_line)

points = np.column_stack(np.where(component > 0))
if points.shape[0] >= 20:
    points = points[:, :-1]
    centroid = points.mean(axis=0)
    centered = points - centroid
    cov = np.cov(centered, rowvar=False)
    eigvals, eigvecs = np.linalg.eigh(cov)
    direction = eigvecs[:, np.argmax(eigvals)]
    vx, vy = direction / np.linalg.norm(direction)
    x0, y0 = float(centroid[0]), float(centroid[1])
    if vx < 0 or (vx == 0 and vy < 0):
        vx = -vx
        vy = -vy
    return vx, vy, x0, y0

# Not enough points, use Hough on the connected mask
full_line = _hough_longest_line.connected)
if full_line is not None:
    return _vector_from_line(full_line)

edges = cv2.Canny(component, CANNY_LOW, CANNY_HIGH, apertureSize=3)
if debug_prefix:
    cv2.imwrite(f"{debug_prefix}_06_canny_edges.png", edges)

```

```

lines = cv2.HoughLinesP(edges, 1, np.pi / 180, threshold=HOUGH_THRESHOLD,
minLineLength=HOUGH_MIN_LENGTH, maxLineGap=HOUGH_MAX_GAP)

if lines is not None and len(lines) > 0:
    best_line = max(lines[:, 0], key=lambda line: (line[2] - line[0]) ** 2 + (line[3] - line[1]) ** 2)
    return _vector_from_line(best_line)

return None

```

ภาพที่ 12 โปรแกรมในการสกัดเส้นตรงหลักของวิถีดาวเทียม

3.3.4 การใช้การแปลงแบบฮัฟ

เป็นวิธีสำรองในกรณีที่ไม่สามารถสกัดเส้นได้จากส่วนประกอบที่เลือก หรือข้อมูลมีสัญญาณรบกวนสูง จึงจะเลือกใช้วิธีดังกล่าวโดยใช้ฟังก์ชัน `_hough_longest_line()` เพื่อค้นหาเส้นตรงที่ยาวที่สุดในภาพ จากนั้นแปลงเป็นเวกเตอร์ทิศทางด้วยฟังก์ชัน `_vector_from_line()` เพื่อให้ได้พารามิเตอร์ของเส้นในรูปแบบมาตรฐานดังภาพที่ 13 และภาพที่ 14 ตามลำดับ

```

def _hough_longest_line(image):
    edges = cv2.Canny(image, CANNY_LOW, CANNY_HIGH, apertureSize=3)
    lines = cv2.HoughLinesP(edges, 1, np.pi / 180, threshold=max(8, HOUGH_THRESHOLD
// 2), minLineLength=max(80, HOUGH_MIN_LENGTH // 2), maxLineGap=max(80,
HOUGH_MAX_GAP))
    if lines is None or len(lines) == 0:
        return None
    return max(lines[:, 0], key=lambda line: (line[2] - line[0]) ** 2 + (line[3] - line[1]) ** 2)

```

ภาพที่ 13 โปรแกรมในส่วนการปรับปรุงผลลัพธ์ของการแบ่งส่วน

```
def _vector_from_line(line):
    x1, y1, x2, y2 = line
    dx = x2 - x1
    dy = y2 - y1
    length = np.hypot(dx, dy)
    if length == 0:
        return None
    vx = dx / length
    vy = dy / length
    if vx < 0 or (vx == 0 and vy < 0):
        vx = -vx
        vy = -vy
    return vx, vy, float((x1 + x2) / 2.0), float((y1 + y2) / 2.0)
```

ภาพที่ 14 โปรแกรมในส่วนการปรับปรุงผลลัพธ์ของการแบ่งส่วน

3.4 การคำนวณสมการเส้นและทิศทางการเคลื่อนที่

หลังจากได้เส้นตรงหลักของวิถีดาวเทียมแล้ว ขั้นตอนถัดไปคือการนำเส้นดังกล่าวมาวิเคราะห์เพื่อหาสมการเส้น จุดเริ่มต้น จุดสิ้นสุด และทิศทางการเคลื่อนที่ตามลำดับเวลา โดยฟังก์ชันที่เกี่ยวข้องได้แก่ฟังก์ชัน `get_component_center()`, `orient_line_by_frame_order()` และ `draw_overlay()` ขณะที่ข้อมูลเวกเตอร์เริ่มต้นมาจากฟังก์ชัน `extract_straight_line()`

3.4.1 การหาศูนย์กลางของวัตถุในแต่ละเฟรม

ฟังก์ชัน `get_component_center(binary_mask)` ใช้สำหรับคำนวณจุดศูนย์กลางของพิกเซลสีขาวในหน้ากากรฐานสองของแต่ละเฟรม โดยหาค่าเฉลี่ยของพิกัด x และ y ของทุกพิกเซลที่เป็นส่วนหนึ่งของวัตถุ ผลลัพธ์คือจุดศูนย์กลาง (cx, cy) ซึ่งใช้แทนตำแหน่งของเส้นดาวเทียมในเฟรมนั้น ๆ และเมื่อคำนวณจุดศูนย์กลางของทุกเฟรมแล้ว จะสามารถนำศูนย์กลางของเฟรมแรกและเฟรมสุดท้ายมาใช้ออกแนวโน้มการเคลื่อนที่ตามเวลาได้ ดังภาพที่ 15

```
def get_component_center(binary_mask):
    if binary_mask.dtype != np.uint8:
        binary_mask = binary_mask.astype(np.uint8)
    ys, xs = np.where(binary_mask > 0)
    if len(xs) == 0: return None
    cx = float(np.mean(xs))
    cy = float(np.mean(ys))
    return (cx, cy)
```

ภาพที่ 15 โปรแกรมในส่วนการคำนวณจุดศูนย์กลางของฟิกเชลสีขาวในแผ่นบังลักษณะฐานสอง

3.4.2 การคำนวณเวกเตอร์การเคลื่อนที่

เมื่อได้จุดเริ่มต้น (x_1, y_1) และจุดสิ้นสุด (x_2, y_2) แล้ว สามารถคำนวณเวกเตอร์การเคลื่อนที่ได้ดังสมการที่ 3.1

$$\begin{aligned} dx &= x_2 - x_1 \\ dy &= y_2 - y_1 \end{aligned} \quad (3.1)$$

จากค่า dx และ dy สามารถตีความทิศทางได้ เช่น

$dx > 0$ หมายถึงเคลื่อนที่จากซ้ายไปขวา

$dx < 0$ หมายถึงเคลื่อนที่จากขวาไปซ้าย

$dy > 0$ หมายถึงเคลื่อนที่จากบนลงล่างในระบบพิกัดภาพ

$dy < 0$ หมายถึงเคลื่อนที่จากล่างขึ้นบนในระบบพิกัดภาพ

3.4.3 การกำหนดทิศทางของเส้นตามลำดับเวลา

ถึงแม้จะสามารถหาแนวเส้นตรงได้แล้ว แต่เพียงจากรูปทรงของเส้นเพียงอย่างเดียวจะยังไม่สามารถระบุได้ว่าดาวเทียมเคลื่อนที่จากปลายด้านใดไปยังอีกด้านหนึ่ง จึงจำเป็นต้องกำหนดทิศทางของเส้นให้ตรงกับลำดับเวลา โดยใช้ฟังก์ชัน `orient_line_by_frame_order()` โดยมีหลักการคือสร้างเวกเตอร์ของเส้นจากจุดปลาย p_1 ไปยัง p_2 และสร้างเวกเตอร์เวลา จากจุดศูนย์กลางของเฟรมแรกไปยังเฟรมสุดท้าย จากนั้นคำนวณค่าผลคูณจุด (dot product) ระหว่างเวกเตอร์ทั้งสอง หากค่าเป็นลบ แสดงว่าทิศทางของเส้นสวนทางกับการเคลื่อนที่จริง จึงต้องสลับจุดปลายของเส้น หากค่าเป็นบวก แสดงว่าทิศทางของเส้นถูกต้องอยู่แล้ว วิธีนี้ช่วยให้สามารถระบุได้ว่าจุดใดคือจุดเริ่มต้น และ จุดใดคือจุดสิ้นสุดของการเคลื่อนที่ ดังภาพที่ 17

```
def orient_line_by_frame_order(p1, p2, first_center, last_center):
    p1 = np.array(p1, dtype=np.float32)
    p2 = np.array(p2, dtype=np.float32)
    first_center = np.array(first_center, dtype=np.float32)
    last_center = np.array(last_center, dtype=np.float32)
    line_vec = p2 - p1
    time_vec = last_center - first_center
    dot = np.dot(line_vec, time_vec)
    if dot < 0:
        return tuple(map(int, p2)), tuple(map(int, p1))
    else:
        return tuple(map(int, p1)), tuple(map(int, p2))
```

ภาพที่ 17 โปรแกรมในส่วนการกำหนดทิศทางของเส้นตามลำดับเวลา

3.4.4 การคำนวณสมการเส้นตรง

เมื่อกำหนดจุดเริ่มต้นและจุดสิ้นสุดของเส้นได้แล้ว จึงสามารถคำนวณสมการเส้นตรงได้ 2 รูปแบบ ได้แก่ รูปความชัน-จุดตัดแกนดังสมการที่ 3.2

$$y = \frac{y_2 - y_1}{x_2 - x_1} x + b \quad (3.2)$$

สามารถคำนวณจุดตัดแกน y ได้จากสมการที่ 3.3

$$b = y_1 - mx_1 \quad (3.3)$$

โดยมีรูปทั่วไปดังสมการที่ 3.4

$$Ax + By + C = 0 \quad (3.4)$$

โดยกำหนดให้ $A = y_1 - y_2$

$$B = x_1 - x_2$$

$$C = x_1 y_2 - x_2 y_1$$

สมการทั้งสองรูปแบบนี้ถูกใช้เพื่ออธิบายแนวเส้นวิถีของดาวเทียมในระบบพิกัดภาพ และสามารถนำไปใช้ต่อในการวิเคราะห์เชิงเรขาคณิตหรือการเปรียบเทียบกับข้อมูลวงโคจรในขั้นตอนต่อไป

3.4.5 การแสดงผลลัพธ์

ฟังก์ชัน `draw_overlay(base_img, merged_binary, endpoint_1, endpoint_2, dx, dy, motion_x, motion_y)` ถูกใช้สำหรับสร้างภาพซ้อนทับ (overlay) บนภาพต้นฉบับ เพื่อแสดงผลลัพธ์ของการตรวจจับเส้นวิถีดาวเทียม โดยภายในภาพจะแสดงองค์ประกอบสำคัญ ได้แก่ เส้นตรงหลักของวิถีดาวเทียม จุดเริ่มต้นและจุดสิ้นสุดของเส้น รวมถึงข้อมูลเชิงปริมาณ เช่น ค่า dx , dy และทิศทางการเคลื่อนที่ ดังภาพที่ 18

```
def draw_overlay(base_img, merged_binary, endpoint_1, endpoint_2, dx, dy, motion_x, motion_y):
    overlay = cv2.cvtColor(base_img.astype(np.uint8), cv2.COLOR_GRAY2BGR)
    overlay[merged_binary > 0] = [0, 255, 255]
    cv2.line(overlay, endpoint_1, endpoint_2, (0, 255, 0), 2)
    cv2.circle(overlay, endpoint_1, 5, (0, 0, 255), -1)
    cv2.circle(overlay, endpoint_2, 5, (255, 0, 0), -1)
    text1 = f"Start: {endpoint_1} End: {endpoint_2}"
    text2 = f"dx = {dx}, dy = {dy}"
    text3 = f"Motion: {motion_x}"
    text4 = f"Motion: {motion_y}"
    cv2.putText(overlay, text1, (20, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
    cv2.putText(overlay, text2, (20, 60), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
    cv2.putText(overlay, text3, (20, 90), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
    cv2.putText(overlay, text4, (20, 120), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
    return overlay
```

ภาพที่ 18 โปรแกรมในส่วนการแสดงผลลัพธ์เป็นกราฟ

3.5 กระบวนการทำงานของระบบหลักในส่วนของการประมวลผลภาพ

ฟังก์ชัน `main()` เป็นส่วนควบคุมการทำงานของระบบ โดยทำหน้าที่รวมทุกขั้นตอนตั้งแต่การอ่านข้อมูลภาพ การประมวลผลด้วยโมเดล การสกัดเส้นวิถี ไปจนถึงการคำนวณผลลัพธ์และการบันทึกข้อมูล โดยลำดับการทำงานสามารถอธิบายได้ดังภาพที่ 19

```

def main():
    ensure_dir(OUTPUT_FOLDER)
    model = ASTA(MODEL_PATH)

    image_files = sorted([
        os.path.join(INPUT_FOLDER, f)
        for f in os.listdir(INPUT_FOLDER)
        if is_image_file(f)
    ])

    if not image_files:
        print(f"No image files found in folder: {INPUT_FOLDER}")
        return

    predicted_masks = []
    frame_centers = []
    first_img = None

    for img_path in image_files:
        print("Processing:", img_path)

        img = load_grayscale_image(img_path)
        if first_img is None:
            first_img = img.copy()

        norm = normalize_image(img)
        pred = predict_full_mask(model, norm, PATCH_SIZE)
        predicted_masks.append(pred)

        # Save per-frame predicted mask
        filename = os.path.splitext(os.path.basename(img_path))[0]
        pred_save = (pred * 255).clip(0, 255).astype(np.uint8)
        cv2.imwrite(os.path.join(OUTPUT_FOLDER, f"{filename}_predicted_mask.png"), pred_save)

```

```

# Get frame center
    frame_binary = postprocess_mask(pred, THRESHOLD, MIN_COMPONENT_AREA)
    center = get_component_center(frame_binary)
    frame_centers.append(center)

merged_pred = np.max(np.stack(predicted_masks), axis=0)
merged_binary = postprocess_mask(merged_pred, THRESHOLD, MIN_COMPONENT_AREA)

# Save merged predicted mask
satellite_name = os.path.basename(INPUT_FOLDER.rstrip('/\\'))
merged_pred_save = (merged_pred * 255).clip(0, 255).astype(np.uint8)
cv2.imwrite(os.path.join(OUTPUT_FOLDER, f"{satellite_name}_merged_predicted_mask.png"),
merged_pred_save)

# Extract line with debug prefix
debug_prefix = os.path.join(OUTPUT_FOLDER, satellite_name)
result = extract_straight_line(merged_binary, debug_prefix=debug_prefix)

if result is None:
    raise ValueError("No line detected")

vx, vy, x0, y0 = result

h, w = merged_binary.shape

# Calculate line endpoints at image boundaries
# Line: (x - x0) / vx = (y - y0) / vy
endpoints = []

# Check intersection with left edge (x = 0)
if abs(vx) > 1e-6:
    t = -x0 / vx
    y_int = y0 + t * vy
    if 0 <= y_int <= h:
        endpoints.append((0, int(round(y_int))))

```

```

# Check intersection with right edge (x = w-1)
if abs(vx) > 1e-6:
    t = (w - 1 - x0) / vx
    y_int = y0 + t * vy
    if 0 <= y_int <= h:
        endpoints.append((w - 1, int(round(y_int))))

# Check intersection with top edge (y = 0)
if abs(vy) > 1e-6:
    t = -y0 / vy
    x_int = x0 + t * vx
    if 0 <= x_int <= w:
        endpoints.append((int(round(x_int)), 0))

# Check intersection with bottom edge (y = h-1)
if abs(vy) > 1e-6:
    t = (h - 1 - y0) / vy
    x_int = x0 + t * vx
    if 0 <= x_int <= w:
        endpoints.append((int(round(x_int)), h - 1))

# Remove duplicates and keep only the two most distant points
endpoints = list(set(endpoints))
if len(endpoints) >= 2:
    # Sort by distance from (x0, y0)
    endpoints.sort(key=lambda p: (p[0] - x0)**2 + (p[1] - y0)**2)
    ep1, ep2 = endpoints[0], endpoints[-1]
else:
    # Fallback: use large extending points
    ep1 = (int(x0 - vx * 10000), int(y0 - vy * 10000))
    ep2 = (int(x0 + vx * 10000), int(y0 + vy * 10000))

# Orient line by frame centers for correct temporal direction
valid_centers = [c for c in frame_centers if c is not None]
if len(valid_centers) >= 2:

```

```

    first_center = valid_centers[0]
    last_center = valid_centers[-1]
    start_pt, end_pt = orient_line_by_frame_order(ep1, ep2, first_center, last_center)
else:
    start_pt, end_pt = ep1, ep2
x1, y1 = start_pt
x2, y2 = end_pt

# Calculate line equation
if x2 != x1:
    m = (y2 - y1) / (x2 - x1)
    b = y1 - m * x1
    slope_form = f"y = {m:.6f}x + {b:.6f}"
else:
    slope_form = f"x = {x1}"

# Calculate general form: Ax + By + C = 0
A = y1 - y2
B = x2 - x1
C = x1 * y2 - x2 * y1
general_form = f"{A}x + {B}y + {C} = 0"

# Calculate motion vector
dx = x2 - x1
dy = y2 - y1
line_length = np.hypot(dx, dy)

# Determine motion direction
if dx > 0:
    motion_x = "left -> right"
elif dx < 0:
    motion_x = "right -> left"
else:
    motion_x = "none"

if dy > 0:
    motion_y = "top -> bottom (image coords)"

```

```

elif dy < 0:
    motion_y = "bottom -> top (image coords)"
else:
    motion_y = "none"

```

ภาพที่ 19 โปรแกรมในส่วนการทำงานระบบหลักในการสกัดลักษณะภาพ

3.5.1 การแสดงผลและบันทึกผลลัพธ์

ในขั้นตอนสุดท้าย ระบบจะทำการสร้างภาพซ้อนทับ เพื่อแสดงเส้นวิถีที่ตรวจพบบนภาพต้นฉบับ พร้อมทั้งบันทึกผลลัพธ์ในรูปแบบต่าง ๆ ได้แก่:

- ภาพหน้ากากการทำนาย (predicted mask) ของแต่ละเฟรม
- ภาพหน้ากากที่รวม (merged mask)
- ภาพเส้นตรงสุดท้าย
- ภาพซ้อนทับสำหรับการตรวจสอบ

รวมถึงบันทึกข้อมูลเชิงตัวเลข เช่น สมการเส้น จุดปลาย และเวกเตอร์การเคลื่อนที่ (motion vector) ลงในไฟล์ข้อความ (.txt) เพื่อใช้ในการวิเคราะห์เพิ่มเติมและเปรียบเทียบกับบัญชีรายชื่อดาวเทียม

3.6 การกำหนดขอบเขตการมองเห็นของระบบ

ในกระบวนการเริ่มต้นของระบบตรวจจับดาวเทียมจากภาพทางดาราศาสตร์ ข้อมูลเริ่มต้นที่มีอยู่มักจำกัดอยู่เพียง พิกัดศูนย์กลางของภาพในระบบพิกัดศูนย์กลางสูตร ซึ่งเป็นระบบพิกัดทรงกลมท้องฟ้า ที่อยู่ในรูปแบบที่ประกอบไปด้วยไรต์แอสเซนชัน (right ascension: RA) และเดคลิเนชัน (declination: Dec) อย่างไรก็ตาม ขอบเขตเชิงพื้นที่ของภาพ (field of view: FOV) ยังไม่ถูกนิยามอย่างชัดเจน ดังนั้นจึงจำเป็นต้องมีขั้นตอนการแปลงข้อมูลดังกล่าวให้กลายเป็นขอบเขตเชิงมุมบนทรงกลมท้องฟ้า ซึ่งสามารถนำไปใช้ในกระบวนการคัดกรองวัตถุในขั้นตอนถัดไปได้

ในขั้นตอนแรก พิกัดศูนย์กลางของภาพจะถูกป้อนเข้าสู่ระบบในรูปแบบฐานหกสิบ (sexagesimal) ที่ไรต์แอสเซนชันอยู่ในรูป ชั่วโมง:นาที:วินาที และเดคลิเนชันอยู่ในรูป องศา:ลิปดา:ฟิลิปดา จากนั้นค่าพิกัดดังกล่าวจะถูกแปลงให้อยู่ในหน่วยองศา เพื่อให้เหมาะสมต่อการคำนวณเชิงตัวเลข โดยอาศัยคุณสมบัติของคลาสวัตถุ SkyCoord ใน Astropy ดังภาพที่ 20

```

center = SkyCoord(ra=ra_str, dec=dec_str, unit=(u.hourangle, u.deg))
ra_center = center.ra.degree
dec_center = center.dec.degree

```

ภาพที่ 20 โปรแกรมในส่วนการแปลงพิกัดศูนย์กลางภาพให้อยู่ในหน่วยเชิงองศา

ซึ่งในเชิงคณิตศาสตร์แล้ว การแปลงไรต์แอสเซนชัน จากหน่วยชั่วโมงเป็นองศาสามารถเขียนได้ตามสมการที่ 3.5 โดยที่ค่าคงที่ 15 นั้นมาจากหาร 360 องศา กับ 24 ชั่วโมง ได้เป็น 15 องศาต่อชั่วโมง

$$RA_{deg} = RA_{hour} \times 15 \quad (3.5)$$

โดยกำหนดให้

RA_{deg} คือค่าพิกัด RA ในหน่วยชั่วโมง

RA_{hour} คือค่าพิกัด RA ในหน่วยของชั่วโมง

ต่อมาเป็นการคำนวณความละเอียดเชิงมุมต่อพิกเซล โดยสามารถคำนวณจากข้อมูลขนาดของขอบเขตเชิงพื้นที่ของภาพ และจำนวนพิกเซลในแนวนอนของภาพ สามารถคำนวณค่าความละเอียดเชิงมุมต่อพิกเซลได้ ดังภาพที่ 21 และสมการที่ 3.6

$$\text{pixel_scale} = \text{fov_deg} / \text{img_w}$$

ภาพที่ 21 โปรแกรมในส่วนการคำนวณความละเอียดเชิงมุมต่อพิกเซล

$$s = \frac{FoV}{N_x} \quad (3.6)$$

โดยกำหนดให้

S คือค่าความละเอียดเชิงมุมต่อพิกเซล

FoV คือขนาดมุมมองภาพในแนวนอน (องศา)

N_x คือจำนวนพิกเซลในแนวนอนของภาพ

จากนั้นโปรแกรมจะทำการกำหนดระบบพิกัดเชิงโลก (world coordinate system: WCS) เพื่อให้สามารถแปลงพิกัดระหว่างปริภูมิพิกเซล (pixel space) และพิกัดท้องฟ้าได้ ระบบใช้แบบจำลอง WCS ซึ่งกำหนดผ่านพารามิเตอร์หลัก ดังภาพที่ 22 และสมการที่ 3.7

$$\begin{aligned} \text{w.wcs.crpix} &= [(\text{img_w} + 1) / 2, (\text{img_h} + 1) / 2] \\ \text{w.wcs.crval} &= [\text{ra_center}, \text{dec_center}] \\ \text{w.wcs.ctype} &= ["\text{RA---TAN}", "\text{DEC--TAN}"] \end{aligned}$$

ภาพที่ 22 โปรแกรมในส่วนการกำหนดระบบพิกัดเชิงโลก

$$\begin{bmatrix} \Delta RA \\ \Delta Dec \end{bmatrix} = \begin{bmatrix} x - x_{ref} \\ y - y_{ref} \end{bmatrix} \cdot \begin{bmatrix} -S & 0 \\ 0 & S \end{bmatrix} \quad (3.7)$$

โดยกำหนดให้

(x, y) คือพิกัดพิกเซล
 (x_{ref}, y_{ref}) คือจุดอ้างอิงพิกเซลจากพิกเซลกลาง
 (RA, Dec) คือพิกัดจริงของจุดอ้างอิง

โดยสังเกตว่ามีการคูณเชิงสเกลาร์กับเมทริกซ์แปลงสเกลและการหมุน เนื่องจากเครื่องหมายลบในแกน RA เกิดจากธรรมชาติของระบบพิกัดท้องฟ้าที่มีทิศทางกลับด้านกับแกน X ของภาพ

ในขั้นตอนถัดไปโปรแกรมจะพิจารณาการหมุนของภาพ ซึ่งมีไว้ในกรณีที่กล้องมีการหมุน จะมีการใช้เมทริกซ์การหมุนเข้ามาปรับแกนพิกัด โดยเมทริกซ์นี้จะถูกนำไปคูณร่วมกับมาตราส่วนพิกเซลเพื่อสร้าง เมทริกซ์การแปลง (transformation matrix) ที่สมบูรณ์สำหรับ WCS ดังภาพที่ 23 และสมการที่ 3.8

```
theta = np.deg2rad(rotation_deg)
w.wcs.pc = [[np.cos(theta), -np.sin(theta)], [np.sin(theta), np.cos(theta)]]
```

ภาพที่ 23 โปรแกรมในส่วนการกำหนดระบบพิกัดเชิงโลก

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (3.8)$$

โดยกำหนดให้

θ คือมุมหมุนในหน่วยเรเดียน

จากนั้นทำการแปลงพิกัดมุมภาพจากพิกเซลเพื่อหาขอบเขตของภาพ จำเป็นต้องพิจารณาพิกัดของมุมทั้ง 4 ของภาพ และทำการแปลงเป็นพิกัดท้องฟ้า โดยในเชิงแนวคิดคือการนำสมการ WCS ในข้อก่อนหน้าไปใช้กับทุกจุดมุม ดังภาพที่ 24

```
corners_pixels = np.array([[0, 0], [img_w, 0], [0, img_h], [img_w, img_h]])
ra_list, dec_list = w.pixel_to_world_values(x, y)
```

ภาพที่ 24 โปรแกรมในส่วนการแปลงพิกัดมุมภาพจากพิกเซล

และในขั้นตอนสุดท้ายเป็นการคำนวณขอบเขตเชิงมุมของกรอบขอบเขต (bounding box) โดยเมื่อได้ค่าพิกัดของมุมทั้งสี่แล้ว สามารถคำนวณขอบเขต ดังภาพที่ 25

```
min_ra = np.min(ra_list)
max_ra = np.max(ra_list)
min_dec = np.min(dec_list)
max_dec = np.max(dec_list)
```

ภาพที่ 25 โปรแกรมในส่วนการคำนวณขอบเขตเชิงมุมของภาพ

ผลลัพธ์ที่ได้คือขอบเขตสี่เหลี่ยมเชิงมุมบนทรงกลมท้องฟ้า ซึ่งจะถูกนำไปใช้โดยตรงในขั้นตอนถัดไปสำหรับการตรวจสอบว่าดาวเทียมดวงใดตกอยู่ภายในพื้นที่การมองเห็นของกล้อง โดยสรุปแล้ว ขั้นตอนนี้เป็นกระบวนการแปลงข้อมูลตั้งต้นที่มีเพียงจุดศูนย์กลางภาพ ให้กลายเป็นพื้นที่เชิงมุม ที่มีความหมายเชิงกายภาพบนท้องฟ้า และเป็นโครงสร้างพื้นฐานที่ทำให้ขั้นตอนการตรวจจับและติดตามวัตถุในลำดับถัดไปสามารถดำเนินการได้อย่างเป็นระบบและเชิงคำนวณครบถ้วน

3.7 การจัดเตรียมข้อมูลดาวเทียมย้อนหลังจากบัญชีรายชื่อดาวเทียม

ภายหลังจากที่ได้กำหนดขอบเขตการมองเห็นของระบบในเชิงมุมแล้ว ขั้นตอนถัดไปคือการจัดเตรียมข้อมูลวัตถุในวงโคจร (resident space objects: RSOs) ซึ่งในบริบทของงานวิจัยนี้ มุ่งเน้นไปที่ ดาวเทียมที่ยังคงปฏิบัติการกิจอยู่ โดยใช้ข้อมูลวงโคจรในรูปแบบมาตรฐาน TLE ซึ่งมีลักษณะเป็นข้อมูลเชิงเวลา กล่าวคือ ค่าพารามิเตอร์วงโคจรจะมีความถูกต้องสูงสุดในช่วงเวลาที่ใกล้กับเวลาอ้างอิงของข้อมูล ดังนั้นจึงต้องกำหนดวันที่อ้างอิงสำหรับการดึงข้อมูลให้สอดคล้องกับช่วงเวลาการสังเกตการณ์ ระบบใช้ฐานข้อมูลบัญชีรายชื่อดาวเทียมจาก CelesTrak ซึ่งเป็นแหล่งข้อมูลสาธารณะสำหรับ TLE โดยมีการประกอบ URL ดังภาพที่ 26

```
tle_date = f"{year:04d}-{month:02d}-{day:02d}"
tle_url = (f"https://celestrak.org/NORAD/elements/archive/gp.php"
           f"?GROUP=active&DATE={tle_date}&FORMAT=tle")
```

ภาพที่ 26 โปรแกรมในส่วนการดึงข้อมูลบัญชีรายชื่อดาวเทียมจาก CelesTrak

แม้ว่าในโค้ดจะไม่ได้แสดงการคำนวณพารามิเตอร์เหล่านี้โดยตรง แต่ไลบรารี Skyfield จะใช้ข้อมูลดังกล่าวในการคำนวณตำแหน่งของดาวเทียมผ่านแบบจำลอง SGP4

3.8 การกำหนดกรอบเวลาและตำแหน่ง

หลังจากการจัดเตรียมข้อมูลวงโคจรของดาวเทียมในขั้นตอนก่อนหน้าแล้ว ขั้นตอนถัดไปของระบบคือการกำหนดกรอบอ้างอิงทั้งในเชิงเวลาและเชิงพื้นที่ เพื่อใช้เป็นพื้นฐานในการคำนวณตำแหน่งปรากฏของดาวเทียมเมื่อมองจากผู้สังเกตการณ์บนพื้นผิวโลก ขั้นตอนนี้มีความสำคัญเนื่องจากเป็นกระบวนการแปลงข้อมูลวงโคจรซึ่งมีลักษณะเชิงนามธรรม ให้กลายเป็นพิกัดที่สามารถตีความและสังเกตได้จริงบนทรงกลมท้องฟ้า

ในมิติของเวลา ระบบใช้มาตรฐานเวลาสากลเชิงพิกัด (Coordinated Universal Time: UTC) เป็นกรอบอ้างอิงเชิงเวลาหลักในการคำนวณ เนื่องจากเป็นมาตรฐานสากลที่ใช้ในการประสานข้อมูลเชิงดาราศาสตร์และการคำนวณวงโคจร อย่างไรก็ตาม เวลาท้องถิ่นของประเทศไทยมีค่าเร็วกว่า UTC อยู่ 7 ชั่วโมง ดังนั้นข้อมูลเวลาที่ป้อนเข้าสู่ระบบจึงจำเป็นต้องถูกแปลงให้อยู่ในรูปแบบ UTC ก่อนดำเนินการคำนวณทุกครั้ง เพื่อให้เกิดความสอดคล้องกับข้อมูลวงโคจรและแบบจำลองทางดาราศาสตร์ที่ใช้งาน และลดความสับสนที่อาจเกิดขึ้นได้ นอกจากนี้ ระบบได้กำหนดช่วงเวลาของการสังเกตการณ์ในรูปแบบที่มีความละเอียดระดับวินาที โดยระบุเป็น ปี เดือน วัน ชั่วโมง นาที และวินาที ครอบคลุมทั้งเวลาเริ่มต้นและเวลาสิ้นสุดของการสังเกตการณ์ ช่วงเวลาดังกล่าวทำหน้าที่เป็นหน้าต่างเวลา สำหรับการติดตามการเคลื่อนที่ของดาวเทียม โดยจะถูกนำไปใช้ในการคำนวณตำแหน่งของดาวเทียม ณ จุดเวลาที่แตกต่างกัน เพื่อสร้างเวกเตอร์การเคลื่อนที่ ซึ่งเป็นพื้นฐานสำคัญสำหรับการวิเคราะห์ในขั้นตอนถัดไป

ในส่วนของการอ้างอิงเชิงพื้นที่ ตำแหน่งของผู้สังเกตการณ์ถูกกำหนดในรูปแบบพิกัดภูมิศาสตร์ ได้แก่ ละติจูด (latitude) ลองจิจูด (longitude) และระดับความสูงจากระดับน้ำทะเล (altitude) โดยพิกัดดังกล่าวอ้างอิงอยู่บนแบบจำลองทรงรีโลกมาตรฐาน World Geodetic System 1984 (WGS84) ซึ่งเป็นมาตรฐานสากลที่ใช้กันอย่างแพร่หลายในงานภูมิสารสนเทศและดาราศาสตร์เชิงตำแหน่ง การกำหนดพิกัดของผู้สังเกตในกรอบอ้างอิงนี้ ช่วยให้สามารถคำนวณตำแหน่งสัมพัทธ์ระหว่างผู้สังเกตและดาวเทียมได้อย่างถูกต้อง และสอดคล้องกับระบบพิกัดอื่นที่เกี่ยวข้องในกระบวนการทั้งหมด อย่างไรก็ตาม พิกัดภูมิศาสตร์ดังกล่าวยังไม่สามารถนำไปใช้ร่วมกับตำแหน่งของดาวเทียมได้โดยตรง เนื่องจากตำแหน่งของดาวเทียมที่ได้จากแบบจำลองวงโคจรผ่านกระบวนการ SGP4 จะอยู่ในรูปของเวกเตอร์สามมิติภายใต้กรอบอ้างอิงที่มีจุดกำเนิดที่ศูนย์กลางโลก ดังนั้นก่อนดำเนินการคำนวณตำแหน่งสัมพัทธ์ จำเป็นต้องทำการแปลงพิกัดของผู้สังเกตจากรูปแบบภูมิศาสตร์ (ϕ, λ, h) ให้อยู่ในรูปเวกเตอร์คาร์ทีเซียนสามมิติ (x, y, z) ภายใต้กรอบอ้างอิงเดียวกันโดยสามารถทำได้โดยใช้ API ของ SkyField ที่ทำการแปลงพิกัดภูมิศาสตร์ เป็นรูปแบบเวกเตอร์สามมิติภายใต้กรอบอ้างอิงที่มีจุดกำเนิดที่ศูนย์กลางโลก ดังภาพที่ 27

```
observer = wgs84.latlon(OBS_LAT, OBS_LON, elevation_m=OBS_ELEV)
```

ภาพที่ 27 โปรแกรมในส่วนการแปลงพิกัดภูมิศาสตร์เป็นรูปแบบเวกเตอร์สามมิติภายใต้
กรอบอ้างอิงที่มีจุดกำเนิดที่ศูนย์กลางโลก

โดยหากเรียกดูวิธีการทำงานของ API ดังกล่าวนั้นมีวิธีการคำนวณดังสมการที่ 3.9

$$\begin{aligned}x &= (R + h) \cos \phi \cos \lambda \\y &= (R + h) \cos \phi \sin \lambda \\z &= (R + h) \sin \phi\end{aligned}\quad (3.9)$$

โดยกำหนดให้

- R คือจำนวนรัศมีเฉลี่ยของโลกในระบบพิกัด WGS84
- h คือความสูงจากระดับน้ำทะเล
- ϕ คือค่าละติจูดบนระบบพิกัด WGS84
- λ คือค่าลองจิจูดบนระบบพิกัด WGS84

เมื่อมีพิกัดดาวเทียมและพิกัดผู้สังเกตอยู่ภายใต้กรอบอ้างอิงที่มีระบบพิกัดเดียวกันใน
แบบจำลอง SGP4 แล้ว สามารถดำเนินการต่อได้ เมื่อมีทั้งข้อมูลดาวเทียม และตำแหน่งผู้สังเกตแล้ว
สามารถคำนวณตำแหน่งสัมพัทธ์ของดาวเทียม ดังภาพที่ 28 และสมการที่ 3.10

```
diff = sat - observer
```

ภาพที่ 28 โปรแกรมในส่วนการแปลงตำแหน่งดาวเทียมสู่กรอบผู้สังเกต

$$r_{relative} = r_{sat} - r_{obs} \quad (3.10)$$

โดยกำหนดให้

- $r_{relative}$ คือตำแหน่งสัมพัทธ์ระหว่างดาวเทียมและผู้สังเกต
- r_{sat} คือตำแหน่งดาวเทียมในกรอบโลก
- r_{obs} คือตำแหน่งผู้สังเกต

จากนั้นเมื่อได้ตำแหน่งสัมพัทธ์ระหว่างดาวเทียมกับผู้สังเกตในรูปแบบเวกเตอร์สาม
มิติ แล้วจึงทำการแปลงเป็นพิกัดท้องถิ่นฟ้าเชิงมุม และระยะทางดังภาพที่ 29 และสมการที่ 3.11

```
ra, dec, dist = topo.radec()
```

ภาพที่ 29 โปรแกรมในส่วนการแปลงเป็นเป็นพิกัดเชิงมุม

$$\begin{aligned} RA &= \tan^{-1}\left(\frac{y}{x}\right) \\ Dec &= \sin^{-1}\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \end{aligned} \quad (3.11)$$

โดยกำหนดให้

(x, y, z) คือองค์ประกอบของเวกเตอร์ตำแหน่งสัมพัทธ์

3.9 การคัดกรองและติดตามดาวเทียมในขอบเขตภาพ

ภายหลังจากที่ได้กำหนดขอบเขตการมองเห็นของภาพ และคำนวณตำแหน่งเชิงมุมของดาวเทียมจากมุมมองผู้สังเกตแล้ว ขั้นตอนถัดไปคือการคัดกรองดาวเทียมที่ปรากฏอยู่ในขอบเขตดังกล่าว และติดตามการเคลื่อนที่ของดาวเทียมในช่วงเวลาที่กำหนด ซึ่งเป็นแกนหลักของกระบวนการตรวจจับในระบบนี้ เริ่มจากการคำนวณตำแหน่งดาวเทียม ณ เวลาเริ่มต้น โดยสำหรับดาวเทียมแต่ละดวงที่ดึงข้อมูลมาจากบัญชีรายชื่อ จะมีการคำนวณตำแหน่งสัมพัทธ์ที่เวลาเริ่มต้นที่ได้กำหนดไว้ โดยมีการคำนวณในลักษณะคล้ายคลึงกับบทก่อนหน้านี้ ดังภาพที่ 30

```
for sat in satellites:
    diff = sat - observer
    topo = diff.at(t_start)
    ra, dec, dist = topo.radec()
    ra_deg = ra.hours * 15
    dec_deg = dec.degrees
```

ภาพที่ 30 โปรแกรมในส่วนการคำนวณตำแหน่งดาวเทียม ณ เวลาเริ่มต้น

จากนั้นจึงดำเนินการคำนวณเพื่อคัดกรองดาวเทียมภายในขอบเขตเชิงพื้นที่ โดยการดำเนินการดังกล่าวเป็นเงื่อนไขหลักของโครงงานนี้ คือการตรวจสอบว่าดาวเทียมใดบ้างที่อยู่ภายในขอบเขตเชิงมุมที่ได้คำนวณไว้โดยมีเงื่อนไข ดังภาพที่ 31 และสมการที่ 3.12

if (min_ra <= ra_deg <= max_ra) and (min_dec <= dec_deg <= max_dec):

ภาพที่ 31 โปรแกรมในส่วนดำเนินการคำนวณเพื่อคัดกรองดาวเทียมภายในขอบเขตเชิงพื้นที่

$$\begin{aligned} RA_{min} &\leq RA_i^{start} \leq RA_{max} \\ Dec_{min} &\leq Dec_i^{start} \leq Dec_{max} \end{aligned} \quad (3.12)$$

โดยกำหนดให้

$RA_{min}, RA_{max}, Dec_{min}, Dec_{max}$ คือขอบเขตของ FoC
 $RA_i^{start}, Dec_i^{start}$ คือพิกัดเริ่มต้นของดาวเทียมแต่ละดวง

โดยเงื่อนไขดังกล่าวอาจตีความได้ว่า เป็นการตรวจสอบถึงการเป็นสมาชิกของจุดเริ่มต้นที่อยู่ภายในกรอบขอบเขตเชิงพื้นที่ของภาพ ดังสมการที่ 3.13

$$(RA_i^{start}, Dec_i^{start}) \in FoV \quad (3.13)$$

โดยกำหนดให้

$RA_i^{start}, Dec_i^{start}$ คือพิกัดเริ่มต้นของดาวเทียมแต่ละดวง

และเมื่อได้เวลาตำแหน่งเริ่มต้นของดาวเทียม ณ เวลาเริ่มต้น จากนั้นโปรแกรมจึงคำนวณตำแหน่งสุดท้าย ณ เวลาสิ้นสุดของดาวเทียมที่อยู่ภายในขอบเขตเชิงพื้นที่นั้น โดยมีการคำนวณในลักษณะเดิม ดังภาพที่ 32

```
topo2 = diff.at(t_end)
ra2, dec2, _ = topo2.radec()
```

ภาพที่ 32 โปรแกรมในส่วนการคำนวณตำแหน่งดาวเทียม ณ เวลาสิ้นสุด

และเมื่อทราบถึงตำแหน่งของดาวเทียมทั้งหมดภายในขอบเขตเชิงพื้นที่ ณ เวลาเริ่มต้น และเวลาสิ้นสุดแล้ว จึงสามารถนิยามเวกเตอร์การเคลื่อนที่ของดาวเทียมได้ ดังสมการที่ 3.14

$$\vec{v}_i = \begin{bmatrix} RA_i^{end} - RA_i^{start} \\ Dec_i^{end} - Dec_i^{start} \end{bmatrix} \quad (3.14)$$

โดยกำหนดให้

$RA_i^{start}, Dec_i^{start}$ คือพิกัดเริ่มต้นของดาวเทียมแต่ละดวง ณ เวลาเริ่มต้น

RA_i^{end}, Dec_i^{end} คือพิกัดสุดท้ายของดาวเทียมแต่ละดวง ณ เวลาสิ้นสุด

จากนั้นจึงจัดเก็บข้อมูลของดาวเทียมทั้งหมดที่พบในลักษณะโครงสร้างข้อมูลแบบพจนานุกรม (dictionary) ซึ่งประกอบไปด้วยชื่อดาวเทียม พิกัดเริ่มต้น พิกัดสิ้นสุด และระยะห่างของดาวเทียมจากผู้สังเกตดังภาพที่ 33

```
found_sats.append({
    "name": sat.name,
    "ra_start": ra_deg,
    "dec_start": dec_deg,
    "ra_end": ra2.hours * 15,
    "dec_end": dec2.degrees,
    "distance_km": dist.km})
```

ภาพที่ 33 โปรแกรมในส่วนการจัดเก็บข้อมูลของดาวเทียมทั้งหมดที่พบ

3.10 การประเมินเปรียบเทียบข้อมูลวิถีดาวเทียมที่ได้จากการคำนวณและผลที่ได้จากการสกัดข้อมูลจากภาพ

ภายหลังจากที่ระบบสามารถคำนวณเส้นทางการเคลื่อนที่ของดาวเทียมในเชิงทฤษฎีจากข้อมูล TLE ได้แล้ว ขั้นตอนถัดไปคือการตรวจสอบความสอดคล้องระหว่างเส้นทางที่คำนวณได้ กับเส้นทางที่ตรวจพบจริงจากการประมวลผลภาพ ซึ่งถือเป็นกระบวนการสำคัญในการระบุว่าดาวเทียมดวงใดเป็นต้นกำเนิดของเส้นรอยทางที่ปรากฏในภาพ

ข้อมูลจากกล้องถูกอ่านในรูปแบบคีย์-ค่า (key-value) โดยมีการดึงค่าขนาดภาพและเวลาการถ่ายภาพ ดังภาพที่ 34

```
image_width = int(dims[0].strip())
image_height = int(dims[1].strip())
```

ภาพที่ 34 โปรแกรมส่วนของการดึงข้อมูลขนาดภาพจากไฟล์ข้อมูลอภิปันธุ์

เส้นที่ตรวจพบจากภาพ จะถูกนิยามในรูปแบบสมการเชิงเส้น $y = mx + b$ จากนั้นจึงนำมาเปรียบเทียบกับข้อมูลดาวเทียมที่คำนวณได้โดยคำนวณจากความแตกต่างเชิงมุม และความแตกต่างเชิงตำแหน่ง ดังสมการที่ 3.15 - 3.17

$$\theta = \tan^{-1}(m) \quad (3.15)$$

$$\Delta\theta = |\tan^{-1}(m_c) - \tan^{-1}(m_d)| \quad (3.16)$$

$$\Delta b = |b_c - b_d| \quad (3.17)$$

จากนั้นให้เป็นคะแนน ดังสมการที่ 3.18

$$Score_i = f(\Delta\theta, \Delta b) \quad (3.18)$$

บทที่ 4

ผลการวิจัย

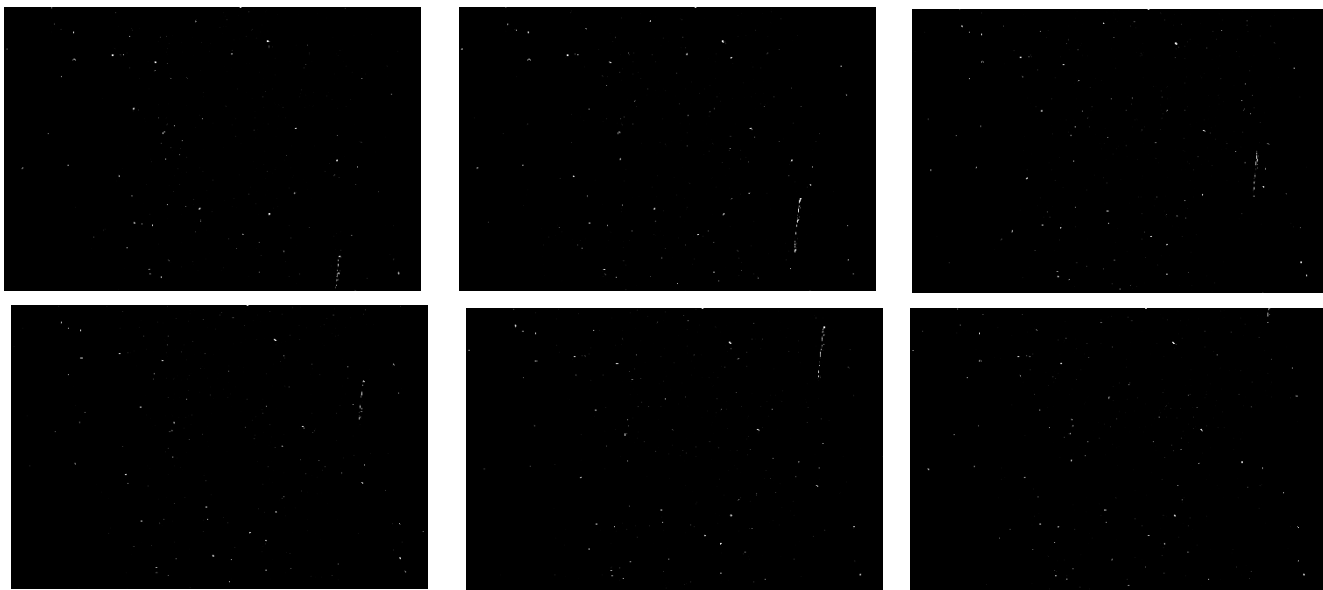
ในบทนี้จะกล่าวถึงผลการประเมินประสิทธิภาพของระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง โดยมีวัตถุประสงค์เพื่อเปรียบเทียบผลการเส้นตรงของรอยทางจากผลลัพธ์การสกัดจากภาพ และจากการคำนวณวิถีโคจรย้อนหลังจากบัญชีรายชื่อดาวเทียม เพื่อช่วยในการระบุชนิดของวัตถุและสนับสนุนการสังเกตการณ์ทางดาราศาสตร์ การทดลองดำเนินการบนเครื่องคอมพิวเตอร์ที่มีหน่วยประมวลผลกลาง (Central Processing Unit: CPU) รุ่น Intel Core i7-1280P หน่วยความจำ (memory unit) ขนาด 32 GB และหน่วยจัดเก็บข้อมูลแบบ SSD ความจุ 1 TB ภายใต้ระบบปฏิบัติการ Windows 11

4.1 ผลการตรวจจับเส้น

จากผลการทดลองพบว่าโมเดลสามารถตรวจจับเส้นที่มีลักษณะมองเห็นได้ และต่อเนื่องได้อย่างมีประสิทธิภาพ แม้ในกรณีที่เส้นมีความสว่างต่ำหรือมีสัญญาณรบกวนจากพื้นหลังของภาพ ก็ยังสามารถตรวจจับเส้นได้ อย่างไรก็ตาม ในบางกรณีที่เส้นมีความจางมากหรือได้รับผลกระทบจากสัญญาณรบกวนสูง อาจทำให้การตรวจจับเส้นนั้นไม่สมบูรณ์ ดังภาพที่ 35 แสดงให้เห็นถึงรอยทางที่ตรวจจับได้ มีลักษณะเส้นที่ชัดเจน และ ภาพที่ 36 แสดงให้เห็นถึงรอยทางที่ตรวจจับได้ แต่ลักษณะของเส้นนั้นไม่ชัดเจน



ภาพที่ 35 กลุ่มของหน้ากากระนาบสองที่ได้จากทำนายของโมเดลจากชุดข้อมูล GALILEO 11 เฟรมที่ 33-39



ภาพที่ 36 กลุ่มของหน้ากาฐานสองที่ได้จากทำนายของโมเดลจากชุดข้อมูล NAVSTAR 85 เฟรมที่ 23-28

4.2 ผลการเชื่อมเส้น

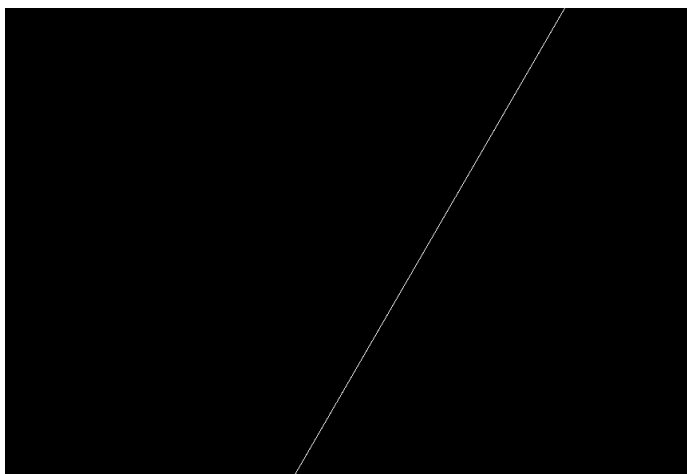
ผลลัพธ์ในส่วนนี้แสดงให้เห็นถึงการรวมข้อมูลจากหลายเฟรม (mask merging) ซึ่งช่วยให้เส้นวิถีมีความต่อเนื่องและยาวมากขึ้น รวมถึงการเชื่อมส่วนของเส้นที่ขาดให้กลายเป็นเส้นตรงที่สมบูรณ์และชัดเจนยิ่งขึ้น ทำให้สามารถนำไปใช้ในการวิเคราะห์ในขั้นตอนถัดไป ในชุดข้อมูล GALILEO และ NAVSTAR 85 ผลจากการรวมภาพฐานสองมีลักษณะ ดังภาพที่ 37 และภาพที่ 38 ตามลำดับ สังเกตได้ว่าเส้นของชุดข้อมูล GALILEO ที่สังเกตได้มีลักษณะเป็นเส้นตรงที่เห็นได้ชัด มีความขาดช่วงของเส้นน้อยกว่าเมื่อเทียบกับเส้นที่สังเกตได้ของชุดข้อมูล NAVSTAR 85 แต่เมื่อดำเนินการเชื่อมเส้นด้วยการแปลงของฮัฟแล้ว ผลลัพธ์ที่ได้ออกมาเป็นเส้นตรงที่มีความต่อเนื่องอย่างชัดเจน ดังภาพที่ 39 และ ภาพที่ 40



ภาพที่ 37 ภาพรวมของภาพฐานสองที่ได้จากหน้ากาการทำนายในชุดข้อมูล GALILEO



ภาพที่ 38 ภาพรวมของภาพฐานสองที่ได้จากหน้ากากทำนายในชุดข้อมูล NAVSTAR 85



ภาพที่ 39 ภาพผลลัพธ์ที่ได้จากการรวมเส้นในชุดข้อมูล GALILEO



ภาพที่ 40 ภาพผลลัพธ์ที่ได้จากการรวมเส้นในชุดข้อมูล NAVSTAR 85

4.3 ผลการวิเคราะห์เส้นวิถีและทิศทางการเคลื่อนที่ของวัตถุ

ผลลัพธ์ทั้งหมดถูกจัดเก็บในรูปแบบไฟล์ข้อความ (.txt) เพื่อความสะดวกในการตรวจสอบ และนำไปใช้ในขั้นตอนการเปรียบเทียบกับบัญชีรายชื่อดาวเทียมในลำดับถัดไป

จากผลการทดลองพบว่าสามารถคำนวณสมการเส้นตรงและทิศทางการเคลื่อนที่ได้อย่างชัดเจนในทุกกรณี โดยเวกเตอร์การเคลื่อนที่สามารถใช้ระบุแนวทางการเคลื่อนที่ของวัตถุในแกน x และ y ได้อย่างถูกต้อง และสามารถคำนวณความยาวของเส้นวิถีได้

เพื่อให้เห็นภาพรวมของผลลัพธ์ จึงสรุปข้อมูลในรูปแบบตาราง ได้แก่ ค่าการเปลี่ยนแปลงและทิศทางแกน X และแกน Y และความยาวของเวกเตอร์ในแต่ละชุดข้อมูล ดังตารางที่ 2 และพิกัดของจุดปลายทั้งสอง สมการเส้นตรง และรูปทั่วไปของสมการเส้นตรง ดังตารางที่ 3

ตารางที่ 2 ตารางเวกเตอร์การเคลื่อนที่และทิศทาง

ชุดข้อมูล	dx	dy	ทิศทางแกน X	ทิศทางแกน Y	ความยาว(px)
GALILEO 11	-6683	3255	Right → Left	Top → Bottom	7433.54
GAOFEN	-3260	5643	Right → Left	Top → Bottom	6516.98
NAVSTAR 85	594	-5643	Left → Right	Bottom → Top	5674.18
O3B MPOWER	-7280	-5094	Right → Left	Bottom → Top	8885.23
ONEWEB	-5918	-4791	Right → Left	Bottom → Top	7614.22
UNKNOWN 1	5766	-5571	Left → Right	Bottom → Top	8017.66
UNKNOWN 2	5893	-5643	Left → Right	Bottom → Top	8159.10
UNKNOWN 3	5796	-5409	Left → Right	Bottom → Top	7927.86
UNKNOWN 4	5893	-5643	Left → Right	Bottom → Top	8159.10

ตารางที่ 3 ตารางสมการเส้นตรงและจุดปลายเส้น

ชุดข้อมูล	พิกัดปลาย 1	พิกัดปลาย 2	สมการเส้นตรง	รูปทั่วไป
GALILEO 11	(8287, 2388)	(1604, 5643)	$y = -0.487057x + 6424.238965$	$-3255x - 6683y + 42933189 = 0$
GAOFEN	(6714, 0)	(3454, 5643)	$y = -1.730982x + 11621.810429$	$-5643x - 3260y + 37887102 = 0$
NAVSTAR 85	(6579, 5643)	(7173, 0)	$y = -9.500000x + 68143.500000$	$5643x + 594y - 40477239 = 0$
O3B MPOWER	(8287, 5094)	(1007, 0)	$y = 0.699725x - 704.623352$	$5094x - 7280y - 5129658 = 0$
ONEWEB	(8287, 4791)	(2369, 0)	$y = 0.809564x - 1917.857215$	$4791x - 5918y - 11349879 = 0$
UNKNOWN 1	(0, 5571)	(5766, 0)	$y = -0.966181x + 5571.000000$	$5571x + 5766y - 32122386 = 0$
UNKNOWN 2	(234, 5643)	(6127, 0)	$y = -0.957577x + 5867.072968$	$5643x + 5893y - 34574661 = 0$
UNKNOWN 3	(0, 5409)	(5796, 0)	$y = -0.933230x + 5409.000000$	$5409x + 5796y - 31350564 = 0$
UNKNOWN 4	(1064, 5643)	(6957, 0)	$y = -0.957577x + 6661.861700$	$5643x + 5893y - 39258351 = 0$

4.4 ผลจากการประเมินเปรียบเทียบข้อมูลวิถีดาวเทียมที่ได้จากการคำนวณและผลที่ได้จากการสกัดข้อมูลจากภาพบนข้อมูลที่มีอยู่

ในการศึกษานี้ ได้ดำเนินการทดสอบระบบตรวจจับและระบุดาวเทียมจากภาพถ่ายทางดาราศาสตร์ โดยใช้ชุดข้อมูลทดสอบจำนวนทั้งสิ้น 6 ชุดข้อมูล ซึ่งในแต่ละชุดข้อมูล ผู้ปฏิบัติงานเก็บข้อมูลจากกล้อง ได้มีการบันทึกข้อมูลล่วงหน้าว่าจะมีดาวเทียมเป้าหมายเคลื่อนที่ผ่านบริเวณท้องฟ้าที่กล้องทำการบันทึกภาพ ณ ช่วงเวลาที่กำหนด และตั้งชื่อชุดข้อมูลตามดาวเทียมเป้าหมายนั้น อย่างไรก็ตามจากการทดสอบจริงพบว่าผลลัพธ์ของระบบมีความแปรผันตามความถูกต้องและความครบถ้วนของข้อมูลภูมิพจน์ รวมถึงเงื่อนไขของการถ่ายภาพในแต่ละชุดข้อมูล

ชุดข้อมูล NAVSTAR 85 (USA 581)

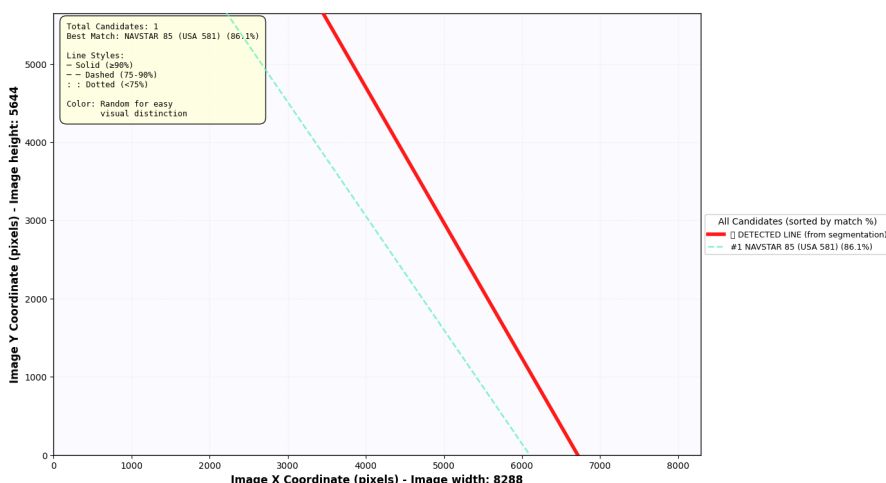
ในชุดข้อมูลนี้ ระบบสามารถตรวจจับเส้นทางการเคลื่อนที่ของวัตถุในภาพได้อย่างชัดเจน และเมื่อทำการเปรียบเทียบกับเส้นทางที่ได้จากการคำนวณตำแหน่งดาวเทียมโดยอาศัยข้อมูลวงโคจร พบว่ามีความสอดคล้องกันในระดับสูง โดยลำดับของดาวเทียมที่เป็นไปได้ (candidate satellite) ที่ได้จากการคัดกรองสามารถจัดอันดับให้ดาวเทียมเป้าหมายอยู่ในลำดับต้นได้อย่างถูกต้อง โดยผลลัพธ์ที่ได้เป็นรายการดาวเทียมที่เป็นไปได้ โดยปรากฏดาวเทียม NAVSTAR 85 (USA 581) มาเพียงดวงเดียว โดยมีความคลาดเคลื่อนเชิงมุมไปราว 12.49 องศา และคิดคะแนนประเมินเป็นร้อยละ 86.1 ดังภาพที่ 41 โดยมีแสดงข้อมูลในรูปกราฟ ดังภาพที่ 42

Total candidates: 1

RANK	SATELLITE NAME	MATCH %	ANGLE °
1	NAVSTAR 85 (USA 581)	86.1%	12.49° ✓

🏆 BEST MATCH: NAVSTAR 85 (USA 581) (86.1%)
 Calculated line: $y = -1.455681x + 8877.673273$
 Distance: 20435.9 km

ภาพที่ 41 ผลการตรวจจับเส้นในภาพและการซ้อนทับกับเส้นทางที่คำนวณได้



ภาพที่ 42 กราฟการจัดอันดับดาวเทียมที่เป็นไปได้ของชุดข้อมูล NAVSTAR 85 (USA 581)

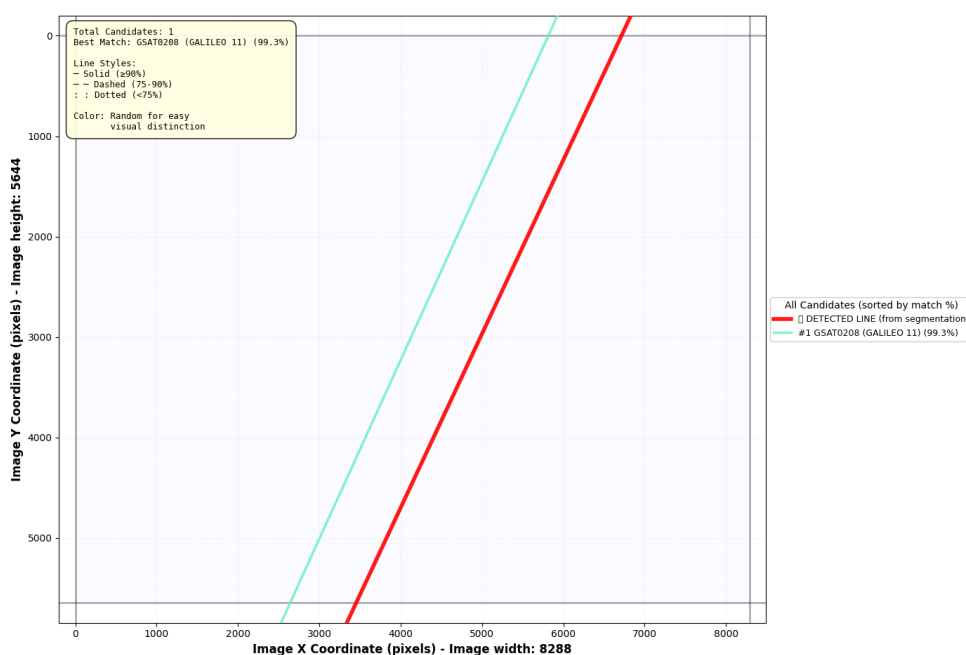
ผลลัพธ์ดังกล่าวสะท้อนให้เห็นว่าข้อมูลอภิปันธุ์ที่ใช้ในชุดนี้มีความถูกต้อง โดยเฉพาะพิกัดจุดกลางภาพที่บันทึกและเวลาที่ใช้ในการบันทึกภาพ มีความถูกต้อง แต่จากภาพที่ 42 จะเห็นได้ว่าเส้นที่คำนวณออกมานั้นมีการเลื่อนไปทางด้านซ้าย เนื่องจากข้อมูล TLE ที่มีนั้นเป็นข้อมูล TLE ณ เดือนมีนาคม 2569 แต่ข้อมูลที่นำมาคำนวณนั้นเป็นจากเดือนกุมภาพันธ์ในปีเดียวกัน ซึ่งระยะเวลาที่แตกต่างกันนั้น รายละเอียดดวงโคจรของดาวเทียมที่ปรากฏใน TLE นั้นจึงเปลี่ยนไปเล็กน้อย และเกิดเป็นความคลาดเคลื่อนอย่างที่ปรากฏ ซึ่งลักษณะดังกล่าวจะปรากฏในชุดข้อมูลอื่นเช่นกัน เนื่องจากทุกชุดข้อมูลนั้นล้วนเป็นข้อมูลในอดีต

ชุดข้อมูล GSAT0208 (GALILEO 11)

ในกรณีนี้ ระบบให้ผลลัพธ์ในลักษณะเดียวกับชุดข้อมูล NAVSTAR 85 (USA 581) กล่าวคือสามารถตรวจจับเส้นทางของดาวเทียมในภาพและจับคู่กับผลการคำนวณได้อย่างถูกต้อง เนื่องจากในขั้นตอนการสกัดข้อมูลจากภาพนั้น รอยทางดาวเทียมที่ปรากฏในชุดข้อมูลนี้มีลักษณะเป็นเส้นที่ตรงต่อเนื่องชัดเจน ส่งผลให้ผลลัพธ์จากการเปรียบเทียบนั้นสอดคล้องกัน โดยดาวเทียมเป้าหมายถูกจัดอยู่ในลำดับแรกของรายการดาวเทียมที่เป็นไปได้อย่างชัดเจน โดยระบบแสดงผลดาวเทียมที่เป็นไปได้เพียงดวงเดียว โดยมีความคลาดเคลื่อนเชิงมุมไปราว 0.67 องศา และคิดคะแนนประเมินเป็นร้อยละ 99.3 ดังภาพที่ 43

Total candidates: 1			
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	GSAT0208 (GALILEO 11)	99.3%	0.67° 
 BEST MATCH: GSAT0208 (GALILEO 11) (99.3%) Calculated line: $y = -1.778736x + 10343.367816$ Distance: 23228.1 km =====			

ภาพที่ 43 ผลการตรวจจับเส้นและการเปรียบเทียบกับ trajectory ที่คำนวณได้



ภาพที่ 44 กราฟลำดับของดาวเทียมที่เป็นไปได้ในชุดข้อมูล GSAT0208 (GALILEO 11)

ชุดข้อมูล O3B MPOWER F2

แม้ว่าผู้ปฏิบัติงานจะระบุว่าดาวเทียมดังกล่าวควรปรากฏในภาพ แต่ผลการทดสอบในขั้นต้นพบว่าระบบไม่สามารถจับคู่เส้นทางได้อย่างถูกต้อง เนื่องจากค่าพิกัดของกล้องที่บันทึกไว้ในข้อมูลอภิปันธุ์มีความคลาดเคลื่อนจากค่าจริง ส่งผลให้วิถีโคจรที่คำนวณได้ไม่สอดคล้องกับเส้นที่สกัดมาจากภาพ อย่างไรก็ตาม เมื่อทำการปรับแก้ประมาณค่าพิกัดของกล้องให้ถูกต้องตามสภาพวงโคจรของดาวเทียมจริง และทำการประมวลผลใหม่ พบว่าระบบสามารถระบุดาวเทียมได้ถูกต้อง โดยผลลัพธ์มีลักษณะสอดคล้องกับชุดข้อมูลที่ ดังภาพที่ 45 และภาพที่ 46

📡 Loading Historical TLE...
📦 Loaded 14794 satellites
🛰️ Satellites found in FOV: 0
📋 Found satellite list: none

ภาพที่ 45 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ โดยไม่มีดาวเทียมปรากฏ

RANK	SATELLITE NAME	MATCH %	ANGLE °
1	03B MPOWER F2	64.9%	31.58° 🟡
2	SYRACUSE 3A	56.1%	39.52° 🟡
3	SHIJIAN-17 (SJ-17)	54.7%	40.73° 🟡
4	YAOGAN-46	37.5%	56.24° 🔴

🏆 BEST MATCH: 03B MPOWER F2 (64.9%)
 Calculated line: $y = 0.059362x + 2503.158825$
 Distance: 10522.7 km

ภาพที่ 46 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ เมื่อแก้ไขพิกัดกลางกล่องแล้ว

จากผลการทดลองเพิ่มเติม พบว่าความถูกต้องของพิกัดศูนย์กลางของกล่อง มีผลโดยตรงต่อความแม่นยำของการคำนวณวิถีดาวเทียม โดยในกรณีที่ใช้พิกัดศูนย์กลางที่ไม่ได้รับการแก้ไข การคำนวณวิถีดาวเทียมที่คาดว่าจะเคลื่อนผ่านกรอบภาพไม่สอดคล้องกับข้อมูลภาพจริง อย่างไรก็ตามเมื่อทำการปรับแก้พิกัดศูนย์กลางให้ถูกต้องแล้ว พบว่าสามารถตรวจพบดาวเทียมได้ตามช่วงเวลาที่คาดการณ์ไว้ ผลลัพธ์ดังกล่าวจึงเป็นหลักฐานเชิงประจักษ์ที่ยืนยันได้ว่าค่าพิกัดที่บันทึกไว้ในขั้นต้นมีความคลาดเคลื่อน และส่งผลกระทบต่อประสิทธิภาพของกระบวนการตรวจจับและระบุตำแหน่งดาวเทียม

ชุดข้อมูล Gaofen-13 01

ผลการทดลองในชุดข้อมูลนี้มีลักษณะคล้ายคลึงกับชุดข้อมูลที่ 3 กล่าวคือเกิดความคลาดเคลื่อนของข้อมูลพิกัดกล่อง ทำให้ผลการคำนวณตำแหน่งดาวเทียมไม่ตรงกับข้อมูลภาพ ภายหลังจากการปรับแก้พิกัดกล่อง พบว่าระบบสามารถกลับมาระบุดาวเทียมเป้าหมายได้อย่างถูกต้อง สะท้อนให้เห็นถึงความไวของระบบต่อความถูกต้องของข้อมูลเชิงภูมิศาสตร์

Total candidates: 14			
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	USA 285	79.7%	18.23° ✅
2	INTELSAT 906 (IS-906)	78.2%	19.59° ✅
3	ARCTURUS	74.5%	22.95° 🟡
4	EWS-G2 (GOES 15)	72.6%	24.69° 🟡
5	TJS-2	72.0%	25.19° 🟡
6	EUTELSAT 708	71.3%	25.80° 🟡
7	INTELSAT 22 (IS-22)	71.3%	25.80° 🟡
8	INTELSAT 20 (IS-20)	71.3%	25.80° 🟡
9	INTELSAT 36 (IS-36)	71.3%	25.83° 🟡
10	AMOS-4	71.3%	25.83° 🟡
11	COSMOS 2533	71.3%	25.83° 🟡
12	INTELSAT 17 (IS-17)	71.3%	25.86° 🟡
13	GSAT-N2 (GSAT-20)	71.2%	25.91° 🟡
14	TJS-22	69.2%	27.69° 🟡

🏆 BEST MATCH: USA 285 (79.7%)
 Calculated line: $y = -0.135885x + 5274.981818$
 Distance: 36809.6 km

ภาพที่ 47 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ โดยไม่มีดาวเทียม Gaofen-13 01 ปรากฏ

Total candidates: 8

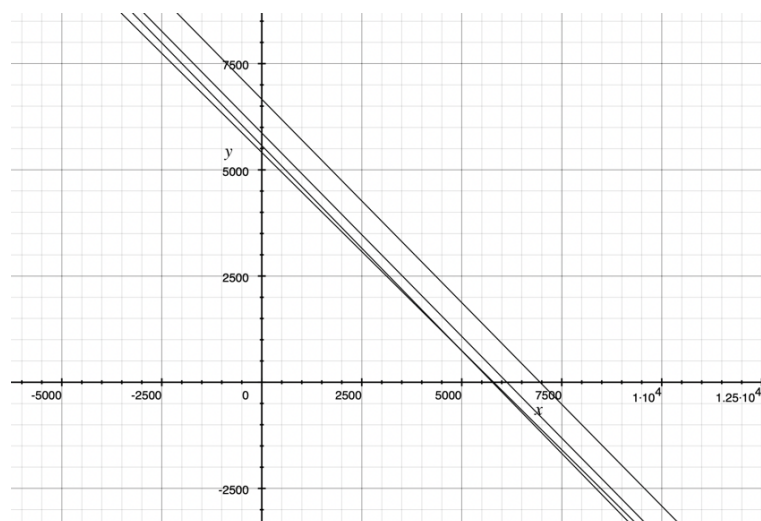
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	GAOFEN-13 01	71.5%	25.68° ▲
2	DIRECTV 10	71.4%	25.74° ▲
3	TELKOM 3S	71.3%	25.84° ▲
4	THAICOM 4	71.3%	25.86° ▲
5	BANGABANDHUSAT-1	71.3%	25.86° ▲
6	ASIASAT 6	71.3%	25.87° ▲
7	ZHONGXING-4A	69.7%	27.27° ▲
8	TJS-11	65.8%	30.77° ▲

🏆 BEST MATCH: GAOFEN-13 01 (71.5%)
 Calculated line: $y = -0.005095x + 2837.333333$
 Distance: 36491.5 km

ภาพที่ 48 ผลการตรวจจับเส้นและการเปรียบเทียบกับที่คำนวณได้ เมื่อแก้ไขพิกัดกลางกล้องแล้ว

ชุดข้อมูล Zhongxing-12

ในกรณีนี้ แม้ว่าจะมีการบันทึกว่าดาวเทียมเป้าหมายจะเคลื่อนที่ผ่าน แต่ผลการวิเคราะห์พบข้อจำกัดสำคัญ กล่าวคือ ระบบตรวจจับได้หลายวัตถุในเวลาที่แตกต่างกัน ในขั้นตอนการสกัดข้อมูลจากภาพสามารถบันทึกวัตถุได้ 4 รายการ แต่หากย้อนดูไฟล์วิดีโอที่ได้จากการกล้องแล้ว พบว่ากล้องมีการเปลี่ยนตำแหน่งถ่าย จึงทำให้จุดกลางภาพเปลี่ยนไป โดยไม่มีการบันทึกข้อมูลการเคลื่อนที่ดังกล่าวไว้ในข้อมูลอภิปันธุ์ ทำให้ระบบไม่สามารถคำนวณอย่างถูกต้องได้ และนอกจากนี้ บริเวณที่กล้องบันทึกภาพมีดาวเทียมโคจรผ่านเป็นจำนวนมาก หากนำผลการที่ได้จากการการสกัดข้อมูลจากภาพทั้ง 4 รายการมาวาดเป็นกราฟ จะเป็นไปดังภาพที่ 49



ภาพที่ 49 กราฟจำลองวิถีดาวเทียมทั้ง 4 ดวงที่ตรวจพบในชุดข้อมูล Zhongxing-12

Total candidates: 18			
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	THEMIS E	63.4%	32.98° ▲
2	LDPE-2	57.3%	38.48° ▲
3	SHIJIAN-20 (SJ-20)	57.1%	38.65° ▲
4	LUCH-5V	56.7%	38.96° ▲
5	ZHONGXING-9	52.1%	43.07° ▲
6	ST-2	51.6%	43.59° ▲
7	WGS F4 (USA 233)	51.6%	43.60° ▲
8	CHINASAT 9C	51.6%	43.60° ▲
9	GSAT-17	51.5%	43.61° ▲
10	KAZSAT-2	51.5%	43.61° ▲
11	QZS-6 (MICHIBIKI-6)	51.5%	43.62° ▲
12	MEASAT-3B	51.5%	43.62° ▲
13	YAMAL 401	51.5%	43.62° ▲
14	MEASAT 3D	51.5%	43.63° ▲
15	ZHONGXING-12	51.5%	43.63° ▲
16	GSAT-15	51.4%	43.72° ▲
17	LUCAS (JDRS-1)	51.4%	43.74° ▲
18	AEHF-5 (USA 292)	50.1%	44.88° ▲
📌 BEST MATCH: THEMIS E (63.4%) Calculated line: $y = -0.190369x + 4439.316027$ Distance: 52689.4 km			

Total candidates: 18			
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	THEMIS E	63.1%	33.24° ▲
2	LDPE-2	57.0%	38.66° ▲
3	SHIJIAN-20 (SJ-20)	56.8%	38.91° ▲
4	LUCH-5V	56.4%	39.22° ▲
5	ZHONGXING-9	51.9%	43.33° ▲
6	ST-2	51.3%	43.85° ▲
7	WGS F4 (USA 233)	51.3%	43.86° ▲
8	CHINASAT 9C	51.3%	43.86° ▲
9	GSAT-17	51.3%	43.87° ▲
10	KAZSAT-2	51.3%	43.87° ▲
11	QZS-6 (MICHIBIKI-6)	51.2%	43.88° ▲
12	MEASAT-3B	51.2%	43.88° ▲
13	YAMAL 401	51.2%	43.88° ▲
14	MEASAT 3D	51.2%	43.88° ▲
15	ZHONGXING-12	51.2%	43.88° ▲
16	GSAT-15	51.1%	43.98° ▲
17	LUCAS (JDRS-1)	51.1%	44.00° ▲
18	AEHF-5 (USA 292)	49.9%	45.13° ✖
📌 BEST MATCH: THEMIS E (63.1%) Calculated line: $y = -0.190369x + 4439.316027$ Distance: 52689.4 km			

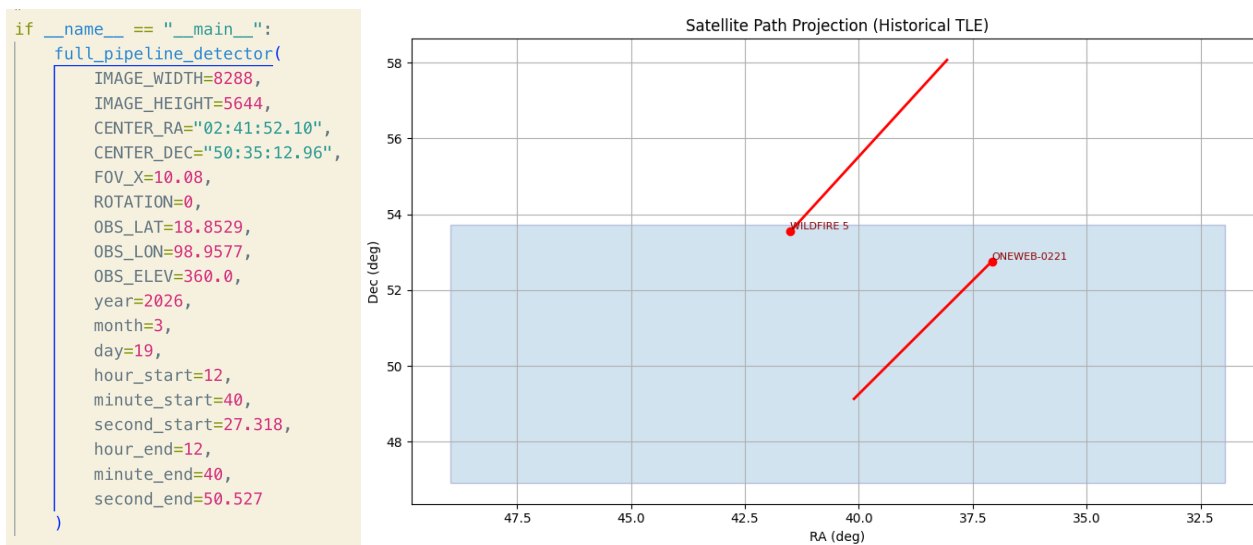
Total candidates: 18			
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	THEMIS E	63.4%	32.98° ▲
2	LDPE-2	57.3%	38.48° ▲
3	SHIJIAN-20 (SJ-20)	57.1%	38.65° ▲
4	LUCH-5V	56.7%	38.96° ▲
5	ZHONGXING-9	52.1%	43.07° ▲
6	ST-2	51.6%	43.59° ▲
7	WGS F4 (USA 233)	51.6%	43.60° ▲
8	CHINASAT 9C	51.6%	43.60° ▲
9	GSAT-17	51.5%	43.61° ▲
10	KAZSAT-2	51.5%	43.61° ▲
11	QZS-6 (MICHIBIKI-6)	51.5%	43.62° ▲
12	MEASAT-3B	51.5%	43.62° ▲
13	YAMAL 401	51.5%	43.62° ▲
14	MEASAT 3D	51.5%	43.63° ▲
15	ZHONGXING-12	51.5%	43.63° ▲
16	GSAT-15	51.4%	43.72° ▲
17	LUCAS (JDRS-1)	51.4%	43.74° ▲
18	AEHF-5 (USA 292)	50.1%	44.88° ▲
📌 BEST MATCH: THEMIS E (63.4%) Calculated line: $y = -0.190369x + 4439.316027$ Distance: 52689.4 km			

Total candidates: 18			
RANK	SATELLITE NAME	MATCH %	ANGLE °
1	THEMIS E	64.2%	32.24° ▲
2	LDPE-2	58.1%	37.67° ▲
3	SHIJIAN-20 (SJ-20)	57.9%	37.91° ▲
4	LUCH-5V	57.5%	38.22° ▲
5	ZHONGXING-9	53.0%	42.33° ▲
6	ST-2	52.4%	42.85° ▲
7	WGS F4 (USA 233)	52.4%	42.86° ▲
8	CHINASAT 9C	52.4%	42.86° ▲
9	GSAT-17	52.4%	42.87° ▲
10	KAZSAT-2	52.4%	42.87° ▲
11	QZS-6 (MICHIBIKI-6)	52.4%	42.88° ▲
12	MEASAT-3B	52.4%	42.88° ▲
13	YAMAL 401	52.4%	42.88° ▲
14	MEASAT 3D	52.3%	42.89° ▲
15	ZHONGXING-12	52.3%	42.89° ▲
16	GSAT-15	52.2%	42.98° ▲
17	LUCAS (JDRS-1)	52.2%	43.00° ▲
18	AEHF-5 (USA 292)	51.0%	44.14° ▲
📌 BEST MATCH: THEMIS E (64.2%) Calculated line: $y = -0.190369x + 4439.316027$ Distance: 52689.4 km			

ภาพที่ 50 รายการดาวเทียมที่เป็นไปได้ของดาวเทียมทั้ง 4 แต่ไม่สามารถแยกดาวเทียมเป้าหมายได้ชัดเจน

ชุดข้อมูล ONEWEB-0221

สำหรับชุดข้อมูลนี้ ปัญหาหลักเกิดจากความไม่สอดคล้องกันของข้อมูลเวลา (timestamp misalignment) ระหว่างเวลาที่บันทึกภาพจากระบบกล้องกับเวลาที่ใช้ในการคำนวณตำแหน่งดาวเทียมจากข้อมูลวงโคจร (TLE) ส่งผลให้การจับคู่เชิงเวลา (temporal alignment) ระหว่างข้อมูลภาพและวิถีโคจรที่คำนวณได้เกิดความคลาดเคลื่อน ทำให้ตำแหน่งที่คาดการณ์ไม่ตรงกับตำแหน่งจริงของวัตถุในภาพ โดยพบว่าในช่วงเวลาที่เริ่มบันทึกภาพ ดาวเทียมได้เคลื่อนที่ผ่านขอบเขตการมองเห็นของกล้องไปแล้ว หรือยังไม่เข้าสู่กรอบภาพในช่วงเวลาที่ระบบทำการตรวจจับ ส่งผลให้ระบบสามารถตรวจพบวัตถุได้เพียงในบางเฟรมที่ไม่สอดคล้องกับช่วงเวลาที่เราควรปรากฏตามการคำนวณดังภาพที่ 51



ภาพที่ 51 กราฟแสดงวิถีโคจรของเทียมที่คาดว่าจะโคจรผ่าน

หากสังเกตจะพบว่าระบบสามารถตรวจพบวัตถุได้เฉพาะในเฟรมที่ 2–3 ซึ่งสอดคล้องกับช่วงเวลาต้นของการบันทึก (ประมาณ 19:38 น.) ขณะที่ผลการคำนวณระบุว่าดาวเทียมควรปรากฏในภายหลัง (ประมาณ 19:40 น.) ความไม่ตรงกันนี้บ่งชี้ถึงความคลาดเคลื่อนของการอ้างอิงเวลาอย่างชัดเจน ทำให้วิถีโคจรที่คำนวณได้ไม่สอดคล้องกับลำดับภาพจริง และส่งผลให้ระบบตรวจจับไม่สามารถระบุวัตถุได้ในช่วงเวลาที่เราควรปรากฏตามแบบจำลอง

โดยสรุป ผลการทดลองทั้ง 6 ชุดข้อมูลแสดงให้เห็นอย่างชัดเจนว่าปัจจัยสำคัญที่ส่งผลต่อประสิทธิภาพของระบบ ได้แก่ ความถูกต้องของพิกัดกล้อง ความสอดคล้องของเวลาในการบันทึกข้อมูล และความครบถ้วนของข้อมูลอวกาศที่เกี่ยวข้อง รวมถึงปัจจัยทางกายภาพของการถ่ายภาพ เช่น การหมุนของกล้องและความหนาแน่นของวัตถุในท้องฟ้า ซึ่งล้วนมีผลโดยตรงต่อความสามารถในการระบุและยืนยันตัวตนของดาวเทียมจากข้อมูลภาพถ่าย

บทที่ 5

ข้อสรุปและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงข้อสรุปและข้อเสนอแนะในการพัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง

5.1 ข้อสรุป

งานวิจัยนี้นำเสนอการพัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง ซึ่งประกอบไปด้วย การดำเนินการที่เกี่ยวข้องกับภาพถ่าย ได้แก่ ภาพที่ใช้ในการทดสอบ การประมวลผลภาพก่อน การเชื่อมเส้นและสกัดเส้นตรงของวิถีดาวเทียม การคำนวณเวกเตอร์การเคลื่อนที่ การคำนวณสมการเส้นและทิศทางการเคลื่อนที่ และการแสดงผลจากการประมวลผลทางภาพ และยังรวมไปถึงขั้นตอนการดำเนินการที่เกี่ยวข้องกับการทำนายวิถีวงโคจรของดาวเทียม และการดึงข้อมูลที่เกี่ยวข้อง ที่ประกอบไปด้วย การกำหนดขอบเขตการมองเห็นของระบบ การจัดเตรียมข้อมูลดาวเทียมย้อนหลังจากบัญชีรายชื่อ การกำหนดกรอบเวลาและตำแหน่ง การคัดกรองและติดตามดาวเทียมในขอบเขตภาพ และการประเมินเปรียบเทียบความแม่นยำข้อมูลที่ได้จากการทำนาย และจากการสกัดข้อมูลวิถีดาวเทียมจากภาพ

จากกระบวนการประมวลผลภาพถ่ายที่ได้ดำเนินการ สามารถสกัดลักษณะเชิงเส้นออกจากภาพได้สำเร็จ แม้ว่าเส้นที่ตรวจพบอาจมีความไม่ต่อเนื่องหรือมีช่องว่างบางส่วน อย่างไรก็ตาม เมื่อพิจารณาในภาพรวม เส้นดังกล่าวยังคงสามารถเชื่อมโยงเป็นเส้นเดียวกันได้อย่างมีนัยสำคัญ จากนั้นจึงนำข้อมูลเชิงเส้นที่ได้ไปแปลงให้อยู่ในรูปสมการเส้นตรง รวมถึงเวกเตอร์สามารถระบุทิศทางการเคลื่อนที่ เพื่อใช้เป็นข้อมูลนำเข้าสำหรับกระบวนการประมวลผลในขั้นตอนถัดไป

จากผลการประเมินความแม่นยำเชิงเปรียบเทียบระหว่างข้อมูลที่สกัดได้จากภาพ และข้อมูลอ้างอิงจากการบันทึกของกล้องโทรทรรศน์ พบว่า ในจำนวนทั้งหมด 6 กรณี มี 2 กรณีที่ข้อมูลมีความถูกต้องและสอดคล้องกัน ระบบสามารถคัดเลือก ระบุ และยืนยันดาวเทียมที่ได้เพียงหนึ่งดวง และตรงกับดาวเทียมที่ถูกต้องตามชุดข้อมูลฝึก (training dataset) อย่างไรก็ตาม ในกรณีที่เหลือ พบว่าชุดข้อมูลฝึกมีความคลาดเคลื่อนในหลายประเด็น เช่น ความผิดพลาดของพิกัดจุดศูนย์กลางภาพ การเบี่ยงเบนของแนวกล้องระหว่างการบันทึกภาพ ตลอดจนความไม่สอดคล้องกันของเวลาที่ใช้ในการถ่ายภาพเมื่อเทียบกับเวลาการผ่านของดาวเทียมตามบัญชีรายชื่อ ทั้งนี้ในกรณีที่ความคลาดเคลื่อนเกิด

จากพิกัดศูนย์กลางภาพ หากมีการปรับแก้พิกัดให้มีความใกล้เคียงกับค่าจริงโดยอ้างอิงจากบัญชีรายชื่อดาวเทียม พบว่าระบบสามารถเพิ่มความสามารถในการระบุดาวเทียมที่มีแนวโน้มถูกต้องได้

5.2 ข้อเสนอแนะ

1. ควรดำเนินการจัดเก็บข้อมูลภาพถ่ายเพิ่มเติมให้มีปริมาณมากขึ้น เนื่องจากในปัจจุบันชุดข้อมูลที่มีอยู่ยังมีจำนวนจำกัด ส่งผลให้การทดสอบระบบยังไม่ครอบคลุม และความเชื่อมั่นอาจจะยังไม่มากพอ
2. หากสามารถใช้อุปกรณ์ที่มีสมรรถนะด้านการประมวลผลสูงขึ้น จะเอื้อให้สามารถดำเนินการสกัดข้อมูลจากภาพได้ครบทุกเฟรมที่มีอยู่ ซึ่งจะช่วยเพิ่มความละเอียดและความต่อเนื่องของข้อมูลนำไปใช้ในการวิเคราะห์ได้
3. หากใช้อุปกรณ์กล้องที่มีค่าความไวแสงสูงขึ้น จะช่วยเพิ่มความสามารถในการรับสัญญาณแสงจากวัตถุเป้าหมายได้ดียิ่งขึ้น ส่งผลให้ภาพที่ได้มีความสว่างและรายละเอียดเพิ่มขึ้น ซึ่งเอื้อต่อการตรวจจับและวิเคราะห์วิถีการเคลื่อนที่ของดาวเทียมได้อย่างมีประสิทธิภาพมากยิ่งขึ้น
4. ระบบมีการใช้งานข้อมูล TLE ซึ่งโดยทั่วไปเป็นข้อมูลที่สะท้อนสถานะวงโคจรในปัจจุบัน อย่างไรก็ตาม ข้อมูล TLE ย้อนหลังมีข้อจำกัดด้านการเข้าถึงและความสมบูรณ์ ทำให้จำเป็นต้องอาศัยการคำนวณวิถีดาวเทียมย้อนหลัง ซึ่งยังย้อนกลับไปในอดีตเป็นระยะเวลานาน ความคลาดเคลื่อนของผลลัพธ์ก็ยิ่งเพิ่มสูงขึ้น ดังนั้น หากสามารถจัดหาข้อมูล TLE ย้อนหลังที่มีความครบถ้วนและเชื่อถือได้ จะช่วยเพิ่มความแม่นยำและความถูกต้องของระบบได้อย่างมีนัยสำคัญ

รายการอ้างอิง

A. De Vittori, R. Cipollone, P. Di Lizia, M. Massari, “Real-time space object tracklet extraction from telescope survey images with machine learning”, *Astrodynamics*, vol. 6, no. 2, pp. 205–218, 2022.

C. Jeffries, R. Acuña, “Detection of streaks in astronomical images using machine learning”, *Journal of Artificial Intelligence and Technology*, vol. 4, no. 1, pp. 1–8, 2024.

D. Kyselica, S. Krajčovič, J. Šilha, R. Ďurikovič, “Advanced algorithms for segmentation of space debris astronomical images”, *Proceeding of the 26th International Conference Information Visualisation (IV)*, IEEE, pp. 336–342, 2022.

D.A. Vallado, P.J. Cefola, “Two-line element sets – Practice and use”, *Proceeding of the 63rd International Astronautical Congress*, International Astronautical Federation, pp. 1–13, 2012

F. Samadzadegan, F. Alidoost, “The design and implementation of an optical astronomical satellite tracking system”, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-1/W3, pp. 25–30, 2013.

F. Stoppa, P.J. Groot, R. Stuik, P. Vreeswijk, S. Bloemen, D.L.A. Pieterse, P.A. Woudt, “Automated detection of satellite trails in ground-based observations using U-Net and Hough transform”, *Astronomy & Astrophysics*, vol. 692, A199, pp. 1–10, 2024

J. Osborn, L. Blacketer, M.J. Townson, O.J.D. Farley, “Astrosat: forecasting satellite transits for optical astronomical observations”, *Monthly Notices of the Royal Astronomical Society*, vol. 509, pp. 1848–1853, 2022.

J. Virtanen, J. Poikonen, M. Näränen, M. Lehtihet, H. Kaasalainen, “Streak detection and analysis pipeline for space-debris optical observations”, *Advances in Space Research*, vol. 57, no. 8, pp. 1607–1617, 2016

K. Balachandran, K. Subbarao, “Classification of resident space objects by shape and spin motion using neural networks and photometric light curves”, *Proceeding of the 8th European Conference on Space Debris*, ESA Space Debris Office, vol. 8, no. 1, 2021.

M.D. Danarianto, A.M. Maharani, B.M. Falah, F. Rohmah, “Prototype of automatic satellite streak detection, identification and initial orbit determination pipeline from optical observation”, *Journal of Physics: Conference Series*, vol. 2214, no. 1, 012018, pp. 1–8, 2022.

ภาคผนวก

ภาคผนวก ก

แบบเสนอหัวข้อโครงการ รายวิชา 2301399 Project Proposal

ปีการศึกษา 2568

ชื่อโครงการ (ภาษาไทย)	ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้ การสกัดลักษณะและการเรียนรู้ของเครื่อง
ชื่อโครงการ (ภาษาอังกฤษ)	Optical Satellite Identification and Tracking from Astronomical Images Using Feature Extraction and Machine Learning
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. รัชลิดา ลิปิกรณ์
ผู้ดำเนินการ	1. กฤติน ลีวนิชย์ เลขประจำตัวนิสิต 6534402523 2. จิตภา พงศ์หบดี เลขประจำตัวนิสิต 6534408323 สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการ คอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หลักการและเหตุผล

การติดตามวัตถุที่เคลื่อนที่ในวงโคจรรอบโลก หรือ Resident Space Objects (RSOs) มีความสำคัญต่อความมั่นคงทางอวกาศ ซึ่งการติดตาม และการพยากรณ์วัตถุที่เคลื่อนที่ในวงโคจรรอบโลกมีความจำเป็นเพื่อประเมินความถูกต้องของการเคลื่อนที่ของวัตถุในวงโคจรรอบโลกที่มีอยู่ในบัญชีรายชื่อทางดาราศาสตร์ (astronomical catalog) และติดตามวัตถุที่อยู่นอกบัญชีรายชื่อ [1]

ทั้งนี้ปัจจุบันในประเทศไทย ยังไม่มีระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์ โดยการใช้การสกัดลักษณะและการเรียนรู้ของเครื่องที่ออกแบบมาสำหรับระบบที่มีอยู่ ทำให้การติดตามวัตถุที่เคลื่อนที่ในวงโคจรรอบโลกต้องใช้บุคลากรในการตรวจหาดำเนินการ ทำให้ต้องใช้เวลานาน และทรัพยากรเป็นจำนวนมาก

คณะผู้จัดทำจึงมีแนวคิดที่จะออกแบบและพัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง โดยใช้เทคนิคการประมวลผลภาพ (image processing) กับองค์ความรู้ทางดาราศาสตร์ เพื่อที่จะสามารถตรวจหาและติดตามวัตถุที่เคลื่อนที่ในวงโคจรรอบโลกได้อย่างอัตโนมัติ

วัตถุประสงค์

1. พัฒนาระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง เพื่อทำการคัดแยกวัตถุออกเป็นดาวเคราะห์ในระบบสุริยะ ดาวฤกษ์นอกระบบสุริยะ ดาวเทียม และวัตถุอื่น ๆ
2. พัฒนารูปแบบการคำนวณตำแหน่งของวัตถุที่ตรวจพบ และเปรียบเทียบกับข้อมูลในบัญชีรายชื่อ

ขอบเขตของโครงการ

1. ระบบที่พัฒนาจะถูกออกแบบให้ทำงานกับข้อมูลภาพถ่ายที่ได้จากสถาบันวิจัยดาราศาสตร์แห่งชาติ (NARIT) จัดหาให้เป็นหลัก โดยข้อมูลภาพถ่ายจะอยู่ในรูปแบบของไฟล์ Audio Video Interleave (AVI) ขนาดตั้งแต่ 3-5 GB
2. ระบบจะถูกพัฒนาในภาษา Python บนสภาพแวดล้อมสำหรับการคำนวณเชิงปฏิสัมพันธ์ (interactive computing environment)
3. ระบบใช้ข้อมูล TLE (Two-Line Element) จาก Celestrak ซึ่งอ้างอิงบัญชีรายชื่อดาวเทียมของ NORAD (North American Aerospace Defense Command) โดยดาวเทียมที่ใช้คือดาวเทียมที่ยังใช้งานอยู่ในวงโคจร (active satellites) ซึ่งมีข้อมูลดาวเทียมมากกว่า 13,000 ดวง และแต่ละรายการประกอบด้วยชื่อดาวเทียม NORAD ID และ TLE สองบรรทัด (Line 1 และ Line 2) สำหรับคำนวณตำแหน่ง [10]
4. ผลลัพธ์ที่ได้จากระบบนั้นจะประกอบด้วย 2 กรณี:
 - 4.1. กรณีที่วัตถุสามารถเปรียบเทียบเข้ากับบัญชีรายชื่อได้
 - 4.1.1. ข้อมูลการระบุดาวเทียม: ชื่อดาวเทียม และ NORAD ID
 - 4.1.2. พิกัดวัตถุ (altitude-azimuth):
 - 4.1.2.1. Altitude (องศา): มุมเงยจากขอบฟ้า
 - 4.1.2.2. Azimuth (องศา): มุมทิศจากทิศเหนือ
 - 4.1.3. ระยะทาง (กิโลเมตร): ระยะห่างจากจุดสังเกตการณ์
 - 4.1.4. สถิติความคลาดเคลื่อน: ค่าเฉลี่ย (mean) ค่าสูงสุด (max) ค่าต่ำสุด (min) และส่วนเบี่ยงเบนมาตรฐาน (standard deviation) ทั้งหมดเป็นหน่วยองศา
 - 4.1.5. รายละเอียดตามจุด: ความคลาดเคลื่อนและพิกัดที่คำนวณได้ของแต่ละจุดใน track
 - 4.1.6. ระดับความมั่นใจ (confidence level): กำหนดจากค่าเฉลี่ยความคลาดเคลื่อน (เช่น very high 99%+ เมื่อ error < 0.1°, high 95-99% เมื่อ error < 0.5°)

4.2. กรณีที่วัตถุไม่สามารถเปรียบเทียบเข้ากับบัญชีรายชื่อได้

4.2.1. พิกัดวัตถุ (altitude-azimuth):

4.2.1.1. Altitude (องศา): มุมเงยจากขอบฟ้า

4.2.1.2. Azimuth (องศา): มุมทิศจากทิศเหนือ

4.2.2. รายละเอียดตามจุด: ความคลาดเคลื่อนและพิกัดที่คำนวณได้ของแต่ละจุดใน track

4.2.3. ระดับความมั่นใจ (confidence level): กำหนดจากค่าเฉลี่ยความคลาดเคลื่อน (เช่น Very High 99%+ เมื่อ error < 0.1°, High 95-99% เมื่อ error < 0.5°)

4.2.4. สถิติความคลาดเคลื่อน: ค่าเฉลี่ย (mean) ค่าสูงสุด (max) ค่าต่ำสุด (min) และส่วนเบี่ยงเบนมาตรฐาน (standard deviation) ทั้งหมดเป็นหน่วยองศา

วิธีการดำเนินงาน

7. กำหนดหัวข้อโครงการที่จะศึกษา

7.1. พุดคุยกับนักวิจัยจากสถาบันวิจัยดาราศาสตร์แห่งชาติเพื่อให้ทราบถึงลักษณะของปัญหาที่พบเจอที่จะสามารถนำระบบคอมพิวเตอร์เข้ามาแก้ไขปัญหาได้

8. ศึกษาความรู้และเครื่องมือที่เกี่ยวข้องกับการพัฒนา

8.1. ความรู้ทางดาราศาสตร์เกี่ยวกับการโคจร เทคโนโลยีดาวเทียม และบัญชีรายชื่อทางดาราศาสตร์

8.2. ความรู้เกี่ยวกับเครื่องมือที่ใช้ในการพัฒนา เทคนิคการประมวลผลภาพ และแพ็คเกจที่เกี่ยวข้องกับการประมวลผลข้อมูลเชิงดาราศาสตร์

9. ออกแบบการทำงานของระบบ

9.1. ส่วนการทำงานที่ตรวจหา และคัดแยกวัตถุที่อยู่ในภาพ

9.2. คำนวณตำแหน่งของวัตถุที่ตรวจพบ และเปรียบเทียบกับข้อมูลในบัญชีรายชื่อ

10. พัฒนาระบบตามที่ได้ออกแบบไว้

11. ทดสอบการทำงานของระบบ และปรับปรุงแก้ไขปัญหาของระบบที่พบระหว่างการทดสอบ

12. จัดทำรูปเล่มโครงการ

ตารางเวลาการดำเนินงาน

ขั้นตอนการดำเนินงาน	สิงหาคม 2568 - พฤษภาคม 2569									
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
กำหนดหัวข้อโครงการที่จะศึกษา										
ศึกษาความรู้ที่เกี่ยวข้องกับการพัฒนา										
ออกแบบการทำงานของระบบ										
พัฒนาระบบตามที่ออกแบบไว้										
ทดสอบและปรับปรุงการทำงานของระบบ										
จัดทำรูปเล่มโครงการ										

ประโยชน์ที่ได้รับ

3. ประโยชน์ต่อตัวนิสิตที่ทำโครงการ
 - 3.1. ได้ศึกษาการพัฒนาระบบคอมพิวเตอร์ด้วยภาษา Python และการใช้งานแพ็คเกจที่เกี่ยวข้องกับการประมวลผลภาพ
 - 3.2. ได้ศึกษาความรู้ทางดาราศาสตร์ที่เกี่ยวข้องกับวัตถุบนท้องฟ้า วัตถุในวงโคจรโลก การจัดการข้อมูลภาพถ่าย ภาพเคลื่อนไหวทางดาราศาสตร์ และการใช้งานข้อมูลจากบัญชีรายชื่อทางดาราศาสตร์
 - 3.3. ได้ประสบการณ์ในการพัฒนาระบบคอมพิวเตอร์เชิงดาราศาสตร์ สามารถนำไปประยุกต์ต่อในอนาคตได้
4. ประโยชน์ที่ได้จากโครงการที่พัฒนาขึ้น
 - 4.1. ได้ระบบระบุและติดตามดาวเทียมจากภาพทางดาราศาสตร์โดยใช้การสกัดลักษณะและการเรียนรู้ของเครื่อง สำหรับสถาบันวิจัยดาราศาสตร์แห่งชาติ
 - 4.2. ช่วยลดภาระและทรัพยากรที่ต้องใช้ในการประมวลผลข้อมูลจากภาพถ่ายทางดาราศาสตร์ ทำให้บุคลากรสามารถดำเนินงานวิจัยได้รวดเร็วและถูกต้องแม่นยำมากยิ่งขึ้น
 - 4.3. บันทึกวัตถุที่อาจค้นพบขึ้นใหม่ลงในบัญชีรายชื่อทางดาราศาสตร์

อุปกรณ์และเครื่องมือที่ใช้

1. ฮาร์ดแวร์
 - 1.1. เครื่องคอมพิวเตอร์พกพาที่มีหน่วยประมวลผล Intel Core i7-1280P มีหน่วยประมวลผลกราฟิก Nvidia RTX 3060 Max-q ขนาด 6 GB ที่มีหน่วยความจำ 32 GB มี SSD ความจุ 1 TB และใช้ระบบปฏิบัติการ Window 11 Home Single Language
 - 1.2. เครื่องคอมพิวเตอร์พกพาที่มีหน่วยประมวลผล Apple M1 Pro ที่มีหน่วยความจำ 16 GB มี SSD ความจุ 1 TB และใช้ระบบปฏิบัติการ MacOS 26
2. ซอฟต์แวร์
 - 2.1. โปรแกรม ASIVideoStack ใน ASISudio เพื่อการดำเนินการไฟล์ AVI ที่ได้จากกล้อง [2]
 - 2.2. The Astropy Project เป็นการรวมกลุ่มแพ็คเกจในภาษา Python ที่เกี่ยวข้องกับดาราศาสตร์เข้าไว้ด้วยกัน แพ็คเกจย่อยที่ถูกนำมาใช้งาน ได้แก่
 - 2.2.1. Astropy เป็นแพ็คเกจหลัก ใช้สำหรับการทำงานเชิงดาราศาสตร์โดยทั่วไป เช่นการแปลงจากตำแหน่งจุดภาพไปเป็นตำแหน่งพิกัดบนท้องฟ้า [3]
 - 2.2.2. Astroquery ใช้ในการดึงข้อมูลบัญชีรายชื่อวัตถุบนท้องฟ้า [4]
 - 2.2.3. Photutils ใช้ในการประมวลผลภาพถ่ายเชิงดาราศาสตร์ [5]
 - 2.3. Matplotlib Pyplot ใช้สำหรับการแสดงผลเป็นภาพ (visualization) [6]
 - 2.4. NumPy ใช้สำหรับการจัดการข้อมูลหลายมิติ และการประมวลผลหลายมิติ [7]
 - 2.5. Imageio สำหรับการอ่านไฟล์ภาพในรูปแบบ JPG กับ PNG และใช้ในการประมวลผลเฟรมทีละภาพเพื่อทำงานด้าน object tracking [8]
 - 2.6. Scikit-image สำหรับการดำเนินการทางการประมวลผลภาพที่เกี่ยวข้อง [9]

งบประมาณ

1. คีย์บอร์ด Keychron K8 HE Wireless Magnetic Switch Custom Keyboard ราคา 4,990 บาท
2. แตร็กแพด Apple Magic Trackpad (USB-C) - พื้นผิว Multi-Touch สีขาว ราคา 4,190 บาท
3. ที่เก็บข้อมูล Kingston 240 Gb SSD Sata A400 ราคา 820 บาท

หมายเหตุ : งบประมาณที่ตั้งไว้ถัวจ่ายทุกรายการ

เอกสารอ้างอิง

- [1] K. Balachandran and K. Subbarao. Classification of resident space objects by shape and spin motion using neural networks and photometric light curves [Online]. ESA Space Debris Office, Volume 8, Issue 1, 2021. Available from: <https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/311/SDC8-paper311.pdf> [2025, November 4]
- [2] ZWO Astro. Software - Discovery Astrophotography [Online]. Available from: <https://www.zwoastro.com/software/> [2025, November 4]
- [3] The Astropy Project. About The Astropy Project [Online]. (n.d.). Available from: <https://www.astropy.org/about.html> [2025, November 4]
- [4] The Astroquery Developers. Astroquery [Online]. (n.d.). Available from: <https://astroquery.readthedocs.io/en/latest/> [2025, November 4]
- [5] Photutils Developers. photutils: An Astropy Package for Photometry [Online]. (n.d.). Available from: <https://photutils.readthedocs.io/en/stable/index.html> [2025, November 4]
- [6] The Matplotlib development team. Pyplot tutorial [Online]. (n.d.). Available from: <https://matplotlib.org/stable/tutorials/pyplot.html> [2025, November 4]
- [7] NumPy Developers. NumPy documentation [Online]. (n.d.). Available from: <https://numpy.org/doc/stable/> [2025, November 4]
- [8] The imageio contributors. Welcome to imageio's documentation! — imageio 2.13.3 documentation [Online]. Available from: <https://imageio.readthedocs.io/en/stable/> [2025, November 4]
- [9] The scikit contributor team. scikit-learn: Machine Learning in Python [Online]. Available from: <https://scikit-learn.org/stable/> [2025, November 4]
- [10] Space-Track.org. TLE Format Documentation [Online]. Space-Track.org. Available from: <https://www.space-track.org/documentation#/tle>

ประวัติผู้เขียน



นายกฤติน ลีวนิชย์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิทยาการคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย



นางสาวจิตาภา พงศ์หบดี
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิทยาการคอมพิวเตอร์
จุฬาลงกรณ์มหาวิทยาลัย