

การวิเคราะห์ภาพถ่ายดาวเทียม เพื่อศึกษาการเปลี่ยนแปลงของ พื้นที่เมือง

โดย นายนครินทร์ จันทะโส รหัสนิสิต 6714650477



การวิเคราะห์ภาพถ่ายดาวเทียม
เพื่อศึกษาการเปลี่ยนแปลงของพื้นที่เมือง

รายวิชา 01159532

นิสิต

นายนครินทร์ ฉันทะโส รหัสนิสิต 6714650477

อาจารย์ผู้สอน

รศ.ดร.พงศ์ประพันธ์ พงษ์โสภณ

พ.ศ. ๒๕๕๖

วันที่ มีนาคม 2025

บทนำและความเป็นมา (Introduction & Background)

1. บริบทและเหตุผล

ในช่วงหลายทศวรรษที่ผ่านมา เมืองทั่วโลกมีการขยายตัวอย่างรวดเร็วเพื่อตอบสนองต่อการเติบโตของประชากรและเศรษฐกิจ อย่างไรก็ตาม การขยายตัวของเมืองส่งผลกระทบต่อสิ่งแวดล้อมอย่างมีนัยสำคัญ โดยเฉพาะ การลดลงของพื้นที่สีเขียวและการเพิ่มขึ้นของอุณหภูมิในเมือง อันเนื่องมาจาก Urban Heat Island Effect (UHI) ซึ่งเป็นปรากฏการณ์ที่พื้นที่เมืองมีอุณหภูมิสูงกว่าพื้นที่ชนบทโดยรอบเนื่องจากอาคาร คอนกรีต และพื้นผิวแข็งดูดซับและเก็บความร้อนมากขึ้นกว่าพื้นที่ธรรมชาติ

นอกจากปัญหาความร้อนแล้ว การลดลงของพื้นที่สีเขียวยังส่งผลกระทบต่อคุณภาพอากาศ ความสมดุลของระบบนิเวศ และความสามารถในการกักเก็บน้ำฝน การศึกษาการเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียวจึงมีความสำคัญต่อการพัฒนาเมืองอย่างยั่งยืน และช่วยให้นักเรียนตระหนักถึงผลกระทบของกิจกรรมมนุษย์ต่อสิ่งแวดล้อม

การศึกษาการเปลี่ยนแปลงของเมืองเป็นหัวข้อที่สำคัญ เนื่องจากมีผลต่อทั้ง สิ่งแวดล้อม เศรษฐกิจ และคุณภาพชีวิตของประชาชน การใช้ ภาพถ่ายดาวเทียม และการวิเคราะห์ภาพเชิงปริมาณสามารถช่วยให้เราติดตามแนวโน้มของการเปลี่ยนแปลงของเมืองได้อย่างแม่นยำ โครงการนี้เลือกใช้ Python เพื่อวิเคราะห์ภาพถ่ายดาวเทียมจากช่วงเวลาต่างๆ และคำนวณ อัตราการลดลงของพื้นที่สีเขียว และอัตราการเพิ่มขึ้นของพื้นที่เมือง ซึ่งจะช่วยให้สามารถออกแบบมาตรการแก้ไขปัญหาด้านสิ่งแวดล้อมได้อย่างเป็นระบบ

วัตถุประสงค์และผลลัพธ์การเรียนรู้ (Objectives and Learning Outcomes)

1. วัตถุประสงค์ของโครงการ

1 พัฒนาโปรแกรม Python สำหรับการวิเคราะห์การเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียว

โครงการนี้มีเป้าหมายหลักในการพัฒนา เครื่องมือดิจิทัลที่ช่วยให้นักเรียนสามารถตรวจสอบแนวโน้มการเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียว ผ่านการใช้ Python และ OpenCV โดยโปรแกรมที่พัฒนาจะสามารถนำเข้าภาพถ่ายดาวเทียมจากช่วงเวลาต่าง ๆ และทำการประมวลผลเพื่อตรวจจับการเปลี่ยนแปลงของสภาพแวดล้อม ซึ่งจะช่วยให้ผู้ใช้สามารถมองเห็นแนวโน้มของการขยายตัวของเมืองและการลดลงของพื้นที่สีเขียวได้อย่างแม่นยำ

2. ให้นักเรียนสามารถใช้ Machine Learning และ Image Processing ในการวิเคราะห์ภาพถ่ายดาวเทียม

เทคโนโลยีปัญญาประดิษฐ์และการประมวลผลภาพ (Image Processing) มีบทบาทสำคัญอย่างมากในปัจจุบัน โดยเฉพาะในการวิเคราะห์ข้อมูลด้านสิ่งแวดล้อมและการใช้ที่ดิน โครงการนี้มุ่งเน้นให้ผู้เรียนได้มีโอกาส เรียนรู้แนวคิดของ AI และการประมวลผลภาพในบริบทของปัญหาสิ่งแวดล้อม ผ่านการเขียนโค้ดและการใช้ข้อมูลจริง ทำให้นักเรียนได้ฝึกฝนการประยุกต์ใช้ความรู้ทางคอมพิวเตอร์และวิทยาศาสตร์สิ่งแวดล้อมควบคู่กัน

3. ช่วยให้นักเรียนเข้าใจแนวโน้มของการเปลี่ยนแปลงของเมืองและพื้นที่สีเขียว

การเติบโตของเมืองส่งผลกระทบต่อสิ่งแวดล้อมในหลายมิติ ไม่ว่าจะเป็น ภาวะโลกร้อน (Global Warming), ปัญหาฝุ่นละออง (PM2.5), และการสูญเสียทรัพยากรธรรมชาติ นักเรียนจะได้ศึกษาข้อมูลที่สะท้อนให้เห็นถึง อัตราการขยายตัวของเมืองในแต่ละปี และผลกระทบของการลดลงของพื้นที่สีเขียว การมีข้อมูลที่ชัดเจนและวิเคราะห์ได้จะช่วยให้ผู้เรียนสามารถ เข้าใจปัญหาด้านสิ่งแวดล้อมได้ลึกซึ้งขึ้น และสามารถนำไปใช้ประกอบการตัดสินใจเชิงนโยบายในอนาคต

4. เชื่อมโยงการศึกษาด้าน Data Science กับวิทยาศาสตร์สิ่งแวดล้อม

หนึ่งในเป้าหมายสำคัญของโครงการนี้คือการผสมผสาน Data Science และวิทยาศาสตร์สิ่งแวดล้อมเข้าด้วยกัน นักเรียนจะได้ฝึกการวิเคราะห์ข้อมูลขนาดใหญ่ (Big Data) และเข้าใจว่าการจัดเก็บและวิเคราะห์ข้อมูลสามารถใช้เป็น เครื่องมือสำคัญในการติดตามและคาดการณ์การเปลี่ยนแปลงของสิ่งแวดล้อม การเรียนรู้เหล่านี้จะเป็นประโยชน์ต่อผู้เรียนที่ต้องการศึกษาต่อในด้านวิทยาศาสตร์ข้อมูล (Data Science), ปัญญาประดิษฐ์ (AI), หรือการวางแผนพัฒนาเมือง

2. ผลลัพธ์การเรียนรู้ (Learning Outcomes)

1. พัฒนาทักษะด้านการเขียนโปรแกรม Python และการใช้เครื่องมือดิจิทัลเพื่อวิเคราะห์ข้อมูลสิ่งแวดล้อม

หลังจากที่ได้เข้าร่วมโครงการนี้ นักเรียนจะมีทักษะในการเขียนโค้ดด้วยภาษา Python และสามารถใช้ไลบรารีสำคัญ เช่น OpenCV สำหรับการประมวลผลภาพ, NumPy สำหรับการคำนวณข้อมูล, และ Matplotlib สำหรับการแสดงผลข้อมูล นักเรียนจะสามารถพัฒนา โปรแกรมวิเคราะห์ภาพถ่ายดาวเทียมที่สามารถตรวจจับและคำนวณอัตราการเปลี่ยนแปลงของพื้นที่สีเขียวและพื้นที่เมืองได้ ซึ่งเป็นพื้นฐานสำคัญของการศึกษาด้าน Data Science และ AI

2. เสริมสร้างทักษะการคิดเชิงตรรกะ (Logical Thinking) และการแก้ปัญหาเชิงคำนวณ (Computational Thinking)

โครงการนี้จะช่วยให้นักเรียนได้ฝึกฝน การคิดวิเคราะห์และการแก้ปัญหาในเชิงคำนวณ โดยผ่านกระบวนการทำงานของโค้ด นักเรียนจะได้ ฝึกตั้งคำถาม สร้างสมมติฐาน และทดสอบผลลัพธ์จากข้อมูลจริง เช่น "พื้นที่สีเขียวลดลงกี่เปอร์เซ็นต์เมื่อเทียบกับปีที่แล้ว?" หรือ "การขยายตัวของเมืองส่งผลต่อคุณภาพอากาศอย่างไร?" การพัฒนาแนวคิดทางตรรกะเหล่านี้ช่วยเสริมสร้างความสามารถในการคิดวิเคราะห์และการตัดสินใจที่อ้างอิงจากข้อมูล (Data-Driven Decision Making)

3. เพิ่มความเข้าใจเกี่ยวกับแนวโน้มของปัญหาสิ่งแวดล้อมและการใช้ข้อมูลจริง (Real-world Environmental Data) ในการวิเคราะห์

นักเรียนจะได้เรียนรู้เกี่ยวกับ การเปลี่ยนแปลงของเมืองและพื้นที่สีเขียวจากข้อมูลจริง โดยการใช้ภาพถ่ายดาวเทียมจาก GISTDA, NASA และ Google Earth Engine ซึ่งเป็นแหล่งข้อมูลที่ใช้ในระดับวิจัยทางวิทยาศาสตร์ การนำข้อมูลเหล่านี้มาวิเคราะห์จะช่วยให้นักเรียนสามารถ เข้าใจแนวโน้มของปัญหาสิ่งแวดล้อมในระดับท้องถิ่นและระดับโลก และสามารถนำเสนอแนวทางการแก้ไขปัญหาได้อย่างมีประสิทธิภาพ

4. พัฒนาทักษะการทำงานเป็นทีมและการนำเสนอผลลัพธ์ในรูปแบบดิจิทัล

นอกเหนือจากการเขียนโค้ดและการวิเคราะห์ข้อมูล นักเรียนจะได้พัฒนาทักษะการทำงานเป็นทีม โดยสามารถใช้ Google Colab และ Jupyter Notebook เพื่อทำงานร่วมกันแบบออนไลน์ นักเรียนสามารถ แบ่งงานกันวิเคราะห์ข้อมูล สร้างกราฟแสดงผล และนำเสนอข้อมูลในรูปแบบของ Dashboard หรือ Interactive Visualization ซึ่งจะช่วยให้การนำเสนอข้อมูลมีความน่าสนใจและเข้าใจง่ายขึ้น

5. สามารถประยุกต์ใช้แนวคิดจากโครงการนี้ในการพัฒนาเครื่องมือเพื่อสนับสนุนการวางแผนนโยบายสิ่งแวดล้อมและการพัฒนาเมือง

ผลลัพธ์จากโครงการนี้สามารถนำไปใช้ เป็นพื้นฐานสำหรับการศึกษาต่อด้านการพัฒนาเมือง เทคโนโลยีดิจิทัล และการวางแผนนโยบายสิ่งแวดล้อม นักเรียนจะได้เรียนรู้ว่า การใช้เครื่องมือดิจิทัลช่วย

ให้นักวิทยาศาสตร์และนักวางแผนนโยบายสามารถวิเคราะห์ข้อมูลสิ่งแวดล้อมและออกแบบแนวทางแก้ไข ปัญหาต่าง ๆ ได้อย่างมีประสิทธิภาพ

กลุ่มเป้าหมายและการบูรณาการกับการสอนวิทยาศาสตร์ (Target Learners and Integration with Science Teaching)

1. ระดับชั้นหรือกลุ่มผู้เรียน

- สำหรับนักเรียนระดับชั้นมัธยมศึกษาตอนปลาย

2. หัวข้อทางวิทยาศาสตร์/สิ่งแวดล้อมที่ครอบคลุม

1. การเปลี่ยนแปลงของระบบนิเวศและการใช้ที่ดิน (Land Use Change & Ecosystem Impact)

การเปลี่ยนแปลงของพื้นที่เมืองมีผลโดยตรงต่อระบบนิเวศและการใช้ที่ดิน โดยเฉพาะอย่างยิ่งในบริเวณที่มีการเปลี่ยนแปลงจากพื้นที่ป่า พื้นที่เกษตรกรรม หรือพื้นที่ชุ่มน้ำให้กลายเป็นเขตเมืองหรือเขตอุตสาหกรรม โครงการนี้จะช่วยให้นักเรียนสามารถ ติดตามและคำนวณอัตราการเปลี่ยนแปลงของพื้นที่สีเขียวและพื้นที่เมือง ซึ่งเป็นข้อมูลสำคัญในการวางแผนด้านสิ่งแวดล้อมและการใช้ทรัพยากรธรรมชาติอย่างยั่งยืน

- การลดลงของพื้นที่สีเขียว → ส่งผลต่อความหลากหลายทางชีวภาพ (Biodiversity Loss) และระบบนิเวศโดยรวม
- การเพิ่มขึ้นของพื้นที่เมือง → มีผลต่อการบริโภคน้ำ ทรัพยากร น้ำเสีย และมลพิษทางอากาศ

การนำเทคโนโลยี Remote Sensing และ Image Processing มาประยุกต์ใช้ในการศึกษานี้จะช่วยให้สามารถตรวจจับการเปลี่ยนแปลงของพื้นที่ได้แม่นยำยิ่งขึ้น

2. ผลกระทบของการเปลี่ยนแปลงภูมิอากาศ (Climate Change & Urban Heat Island Effect)

การขยายตัวของเมืองเป็นหนึ่งในปัจจัยที่ส่งผลกระทบต่อสภาพภูมิอากาศ โดยเฉพาะการเพิ่มขึ้นของปรากฏการณ์เกาะความร้อนในเมือง (Urban Heat Island - UHI) ซึ่งเกิดจากการที่โครงสร้างพื้นฐานของเมือง เช่น ถนน คอนกรีต และอาคารสูง ดูดซับและกักเก็บความร้อนมากกว่าพื้นที่ธรรมชาติ

- อุณหภูมิในเมืองสูงขึ้น → ส่งผลต่อสุขภาพของประชาชน โดยเฉพาะในช่วงฤดูร้อน
- ปัญหาสิ่งแวดล้อมที่เกิดจาก UHI → รวมถึงการเพิ่มขึ้นของการใช้พลังงานเพื่อทำความเย็น เช่น เครื่องปรับอากาศ
- การลดลงของพื้นที่สีเขียว → ทำให้เมืองขาดร่มเงาและพื้นที่ที่สามารถช่วยลดอุณหภูมิได้ตามธรรมชาติ

3. คุณภาพอากาศและผลกระทบจากมลพิษ (Air Quality & Environmental Pollution)

การขยายตัวของเมืองมักมาพร้อมกับ การเพิ่มขึ้นของปริมาณยานพาหนะ โรงงานอุตสาหกรรม และกิจกรรมที่ทำให้เกิดมลพิษทางอากาศ โครงการนี้สามารถช่วยให้นักเรียนวิเคราะห์ว่า พื้นที่ที่มีการพัฒนาเป็นเมืองมากขึ้นนั้นมีคุณภาพอากาศที่เปลี่ยนแปลงไปอย่างไร

- ปริมาณฝุ่นละออง PM2.5 และ NO₂ ในเขตเมือง → มักจะสูงกว่าพื้นที่ชนบท
- การเพิ่มขึ้นของก๊าซเรือนกระจก (Greenhouse Gas Emission) → ส่งผลต่อภาวะโลกร้อน
- ผลกระทบต่อสุขภาพของประชาชน → มลพิษทางอากาศเป็นสาเหตุสำคัญของโรคระบบทางเดินหายใจ

3. แนวทางทางวิชาการ/แนวปฏิบัติ

แนวทาง Inquiry-Based Learning (IBL) ส่งเสริมให้ผู้เรียนเป็นผู้แสวงหาความรู้ด้วยตนเอง ผ่านการตั้งคำถาม การทดลอง และการค้นหาคำตอบผ่านกระบวนการทางวิทยาศาสตร์ ในกรณีของโครงการนี้ นักเรียนจะถูกกระตุ้นให้ตั้งคำถามเกี่ยวกับการเปลี่ยนแปลงของพื้นที่เมือง เช่น

คำถามนำร่องสำหรับการเรียนรู้เชิงสืบสอบ

- "พื้นที่สีเขียวในเมืองของเราลดลงมากน้อยเพียงใดในช่วง 10 ปีที่ผ่านมา?"
- "การขยายตัวของเมืองมีผลกระทบต่ออุณหภูมิและคุณภาพอากาศหรือไม่?"
- "เราสามารถใช้อ AI และ Image Processing เพื่อตรวจจับการเปลี่ยนแปลงของพื้นที่เมืองได้อย่างไร?"

กระบวนการบูรณาการเครื่องมือโค้ดกับ IBL

1. ตั้งคำถาม → ให้นักเรียนตั้งคำถามเกี่ยวกับปัญหาสิ่งแวดล้อมที่เกี่ยวข้องกับการเปลี่ยนแปลงของพื้นที่เมือง
2. วิเคราะห์ข้อมูล → ใช้โค้ด Python และ OpenCV เพื่อวิเคราะห์ภาพถ่ายดาวเทียมและเปรียบเทียบแนวโน้มการเปลี่ยนแปลง
3. แปลผลและอภิปราย → นักเรียนตีความผลลัพธ์ที่ได้จากการวิเคราะห์ข้อมูล และสรุปความหมายของการเปลี่ยนแปลงที่เกิดขึ้น
4. นำเสนอผลการศึกษา → แสดงผลผ่าน Dashboard, กราฟ หรือแผนที่เชิงโต้ตอบ

การออกแบบโครงงานและอัลกอริทึม (Project Design and Algorithm)

1. ภาพรวมของโปรแกรม/เครื่องมือ

โปรแกรมนี้เป็นเครื่องมือที่ใช้ Python และ OpenCV สำหรับ วิเคราะห์ภาพถ่ายดาวเทียม เพื่อตรวจจับการเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียวในช่วงเวลาต่าง ๆ โปรแกรมจะทำงานโดย

- อ่านภาพถ่ายดาวเทียม จากไฟล์ภาพที่เป็นข้อมูลอินพุต
- ประมวลผลภาพ ด้วย Image Processing และ AI เพื่อตรวจจับพื้นที่ที่เป็นเมืองและพื้นที่สีเขียว
- คำนวณอัตราการเปลี่ยนแปลง ของแต่ละประเภทพื้นที่
- แสดงผลในรูปแบบของกราฟและแผนที่ความหนาแน่น เพื่อช่วยให้เข้าใจแนวโน้มการเปลี่ยนแปลงของเมือง

2. อัลกอริทึมหรือผังงาน (Flowchart)

กระบวนการทำงานของโปรแกรมสามารถแบ่งเป็น 4 ขั้นตอนหลัก ดังนี้

1. รับข้อมูลอินพุต (Input Processing)

- โปรแกรมรับ ไฟล์ภาพถ่ายดาวเทียม ที่เป็นภาพจากปีต่าง ๆ
- ใช้ OpenCV โหลดภาพและแปลงเป็น โหมดสี HSV เพื่อแยกพื้นที่สีเขียวและพื้นที่เมือง

2. การประมวลผลภาพ (Image Processing & Analysis)

- ใช้ Thresholding และ Masking เพื่อแยกพื้นที่ที่เป็นสีเขียวและพื้นที่เมือง
- ใช้ Morphological Operations เพื่อลบสัญญาณรบกวนและทำให้ข้อมูลแม่นยำขึ้น
- คำนวณ เปอร์เซ็นต์ของพื้นที่สีเขียวและพื้นที่เมือง จากจำนวนพิกเซลของแต่ละประเภท

3. วิเคราะห์และเปรียบเทียบข้อมูล (Data Analysis & Comparison)

- คำนวณอัตราการเปลี่ยนแปลงของแต่ละพื้นที่ในช่วงเวลาที่แตกต่างกัน
- สร้าง กราฟแสดงแนวโน้มของพื้นที่สีเขียวและพื้นที่เมือง

4. แสดงผลลัพธ์ (Output & Visualization)

- แสดง ภาพที่ผ่านการวิเคราะห์ โดยระบายสีพื้นที่เมืองและพื้นที่สีเขียว
- สร้าง กราฟและ Dashboard เพื่อแสดงแนวโน้มการเปลี่ยนแปลงของพื้นที่

3. ฟังก์ชันสำคัญ (Key Functions)

- ฟังก์ชันโหลดและแปลงภาพถ่ายดาวเทียมเป็นโหมด HSV

```
import cv2
```

```
import numpy as np
```



```
def load_and_convert_image(image_path):
    image = cv2.imread(image_path)
    if image is None:
        raise ValueError(f"ไม่สามารถโหลดภาพ: {image_path}")
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    return hsv_image
```

- ฟังก์ชันแยกพื้นที่สีเขียวและพื้นที่เมืองโดยใช้ Thresholding

```
def extract_green_and_urban_areas(hsv_image):
    # กำหนดขอบเขตสีของพื้นที่สีเขียว
    lower_green = np.array([35, 40, 40])
    upper_green = np.array([85, 255, 255])
    green_mask = cv2.inRange(hsv_image, lower_green, upper_green)

    # กำหนดขอบเขตสีของพื้นที่เมือง (เทา-น้ำตาล)
    lower_urban = np.array([0, 0, 110])
    upper_urban = np.array([180, 60, 255])
    urban_mask = cv2.inRange(hsv_image, lower_urban, upper_urban)

    return green_mask, urban_mask
```

- ฟังก์ชันคำนวณเปอร์เซ็นต์ของพื้นที่สีเขียวและพื้นที่เมือง

```
def calculate_area_percentage(mask):
    total_pixels = mask.size
    green_pixels = np.sum(mask > 0)
    return (green_pixels / total_pixels) * 100
```

- ฟังก์ชันสร้างกราฟแสดงแนวโน้มของพื้นที่สีเขียวและพื้นที่เมือง

```
import matplotlib.pyplot as plt
```

```
def plot_area_trends(years, green_areas, urban_areas):
    plt.figure(figsize=(8, 5))
    plt.plot(years, green_areas, marker='o', linestyle='-', color='green', label="พื้นที่สีเขียว")
    plt.plot(years, urban_areas, marker='o', linestyle='-', color='red', label="พื้นที่เมือง")
    plt.xlabel("ปี")
```

```
plt.ylabel("เปอร์เซ็นต์ของพื้นที่")
plt.title("การเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียว")
plt.legend()
plt.grid(True)
plt.show()
```

4. ไบเบรารีที่ใช้ (Libraries Used)

ไลเบรารี	หน้าที่
cv2 (OpenCV)	ใช้สำหรับการประมวลผลภาพ เช่น โหลดภาพ, แปลงสี, และสร้าง Mask สำหรับพื้นที่เมืองและพื้นที่สีเขียว
numpy	ใช้จัดการข้อมูลในรูปแบบอาร์เรย์และคำนวณเปอร์เซ็นต์ของพื้นที่ที่น่าสนใจ
matplotlib	ใช้สร้างกราฟและแผนภาพแสดงผลแนวโน้มของพื้นที่เมืองและพื้นที่สีเขียว
pandas	ใช้จัดการข้อมูลเชิงตาราง (DataFrame) และวิเคราะห์สถิติของพื้นที่ที่เปลี่ยนแปลง
os	ใช้จัดการไฟล์ภาพ เช่น โหลดไฟล์จากโฟลเดอร์ต่าง ๆ

การพัฒนาโค้ด (Coding Implementation)

1. โค้ด (Code Snippets)

1. โหลดภาพและแปลงเป็นโหมดสี HSV

- โหลดภาพถ่ายดาวเทียมโดยใช้ cv2.imread()
- แปลงภาพเป็นโหมดสี HSV ซึ่งช่วยให้การแยกสีพื้นที่เมืองและพื้นที่สีเขียวง่ายขึ้น

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def load_and_convert_image(image_path):
    """ โหลดภาพถ่ายดาวเทียมและแปลงเป็นโหมดสี HSV """
    image = cv2.imread(image_path)
    if image is None:
        raise ValueError(f"ไม่สามารถโหลดภาพ: {image_path}")
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    return hsv_image
```

2. แยกพื้นที่สีเขียวและพื้นที่เมือง

- ใช้ Thresholding เพื่อตรวจจับสีในช่วงที่กำหนด
- แยกพื้นที่สีเขียวออกโดยกำหนดขอบเขตของค่าสี HSV
- แยกพื้นที่เมืองออกโดยใช้ช่วงสีที่เหมาะสมกับอาคารและโครงสร้างพื้นฐาน

```
def extract_green_and_urban_areas(hsv_image):
    """ ใช้ Thresholding เพื่อตรวจจับพื้นที่สีเขียวและพื้นที่เมือง """

    # ตรวจจับพื้นที่สีเขียว
    lower_green = np.array([35, 40, 40])
    upper_green = np.array([85, 255, 255])
    green_mask = cv2.inRange(hsv_image, lower_green, upper_green)

    # ตรวจจับพื้นที่เมือง (เทา-น้ำตาล)
```

```

lower_urban = np.array([0, 0, 110])
upper_urban = np.array([180, 60, 255])
urban_mask = cv2.inRange(hsv_image, lower_urban, upper_urban)

return green_mask, urban_mask

```

3. คำนวณเปอร์เซ็นต์ของพื้นที่ที่เปลี่ยนแปลง

- คำนวณจำนวนพิกเซลทั้งหมดของภาพ
- คำนวณจำนวนพิกเซลที่ถูกตรวจนับว่าเป็นพื้นที่สีเขียวหรือพื้นที่เมือง
- คำนวณเปอร์เซ็นต์ของพื้นที่ที่ตรวจพบ

```

def calculate_area_percentage(mask):
    """ คำนวณเปอร์เซ็นต์ของพื้นที่สีเขียวหรือพื้นที่เมือง """
    total_pixels = mask.size
    detected_pixels = np.sum(mask > 0)
    return (detected_pixels / total_pixels) * 100

```

4. แสดงผลข้อมูลผ่านกราฟ

- นำข้อมูลเปอร์เซ็นต์ของพื้นที่เมืองและพื้นที่สีเขียวที่คำนวณได้มาสร้าง กราฟแสดงแนวโน้ม
- ใช้ matplotlib ในการสร้างกราฟเส้น (Line Graph)
- ใช้สีเขียวแทนพื้นที่สีเขียว และสีแดงแทนพื้นที่เมือง

```

def plot_area_trends(years, green_areas, urban_areas):
    """ สร้างกราฟแสดงแนวโน้มของพื้นที่สีเขียวและพื้นที่เมือง """
    plt.figure(figsize=(8, 5))
    plt.plot(years, green_areas, marker='o', linestyle='-', color='green', label="พื้นที่สีเขียว")
    plt.plot(years, urban_areas, marker='o', linestyle='-', color='red', label="พื้นที่เมือง")
    plt.xlabel("ปี")
    plt.ylabel("เปอร์เซ็นต์ของพื้นที่")
    plt.title("การเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียว")
    plt.legend()
    plt.grid(True)
    plt.show()

```


2. คำอธิบายขั้นตอนหลัก

1. การโหลดและจัดการข้อมูล

- โปรแกรมเริ่มต้นด้วยการ โหลดภาพถ่ายดาวเทียม จากไฟล์
- ใช้ `cv2.imread()` เพื่อโหลดภาพ และ `cv2.cvtColor()` เพื่อแปลงเป็นโหมดสี HSV

2. การใช้ Thresholding เพื่อแยกประเภทของพื้นที่

- ใช้ `cv2.inRange()` ตรวจสอบสีที่อยู่ในช่วงของพื้นที่สีเขียวและพื้นที่เมือง
- ใช้ ค่าพิกเซลที่ตรงกับช่วงสีที่กำหนด เพื่อสร้าง Mask ของแต่ละประเภทพื้นที่

3. การใช้ Loops และ Conditionals

- ใช้ Loop เพื่อประมวลผลภาพในแต่ละปี

4. การแสดงผลข้อมูล

- โปรแกรมแสดงผลลัพธ์โดยใช้ `matplotlib` เพื่อสร้าง กราฟแนวโน้มของการเปลี่ยนแปลงพื้นที่เมืองและพื้นที่สีเขียว
- ใช้ `cv2.imshow()` (ในกรณีใช้งานในเครื่องที่รองรับ) หรือใช้ `matplotlib` แสดงภาพที่มีการระบายสีพื้นที่ที่ตรวจพบ

3. การทดสอบและแก้ไข (Testing and Debugging)

ปัญหา: ไม่สามารถโหลดภาพบางไฟล์ได้

การแก้ไข:

- ตรวจสอบว่าพารามิเตอร์ของไฟล์ถูกต้อง
- ใช้ `if image is None:` ตรวจสอบและแจ้งเตือนหากภาพโหลดไม่สำเร็จ

ปัญหา: Thresholding ไม่สามารถแยกพื้นที่สีเขียวได้ชัดเจน

การแก้ไข:

- ทดลองปรับค่าขอบเขตของสี HSV
- ใช้ **Morphological Transformations** เพื่อลบจุดรบกวน

ปัญหา: โปรแกรมแสดงผลช้าเมื่อใช้กับภาพขนาดใหญ่

การแก้ไข:

- ใช้ `cv2.resize()` ลดขนาดภาพก่อนประมวลผล
- ใช้ `numpy` เพื่อทำการคำนวณแบบเวกเตอร์ไลซ์ ลดเวลาประมวลผล

แผนการนำไปใช้ในชั้นเรียน (Classroom Implementation Plan)

1. โครงสร้างบทเรียน (Lesson Outline)

รูปแบบการเรียนรู้: ผสมผสานระหว่าง การสอนเชิงบรรยาย การทดลองปฏิบัติ และการทำโครงการ
ระยะเวลา: 4 ชั่วโมง แบ่งเป็น 2 ช่วง (ช่วงละ 2 ชั่วโมง)

กิจกรรม:

- มี ใบงานให้นักเรียนทำแบบฝึกหัดเกี่ยวกับการวิเคราะห์ข้อมูล
- มีกิจกรรม ให้เขียนโค้ด และอภิปรายผลลัพธ์ที่ได้จากการวิเคราะห์ภาพถ่ายดาวเทียม
- มี การนำเสนอผลการวิเคราะห์ ในรูปแบบของรายงาน หรือ dashboard

2. ลำดับขั้นตอนการสอน (Instructional Sequence)

ชั่วโมงที่ 1-2: การเรียนรู้พื้นฐานเกี่ยวกับปัญหาสิ่งแวดล้อมและการใช้โค้ดในการวิเคราะห์ข้อมูล

1. แนะนำหัวข้อและเกริ่นปัญหาสิ่งแวดล้อม (20 นาที)

- แสดงตัวอย่างการขยายตัวของเมืองผ่านภาพถ่ายดาวเทียม
- ถามคำถามกระตุ้น เช่น "การขยายตัวของเมืองส่งผลกระทบต่อสิ่งแวดล้อมอย่างไร?"
- นำเสนอเป้าหมายของบทเรียนและโครงการ

2. อธิบายโค้ดพื้นฐานและการใช้เครื่องมือ (40 นาที)

- อธิบายหลักการของ Image Processing และการวิเคราะห์ภาพถ่ายดาวเทียม
- แนะนำไลบรารีที่ใช้ เช่น OpenCV, NumPy และ Matplotlib
- สอนวิธีโหลดภาพถ่ายดาวเทียมและการแปลงสีเป็น HSV

3. ฝึกให้นักเรียนทดลองเขียนโค้ดพื้นฐาน (1 ชั่วโมง)

- นักเรียนทำใบงานฝึกหัด โดยให้เขียนโค้ดสำหรับ แยกพื้นที่สีเขียวและพื้นที่เมืองจากภาพ
- ทดลองปรับค่า HSV เพื่อให้สามารถตรวจจับพื้นที่ได้แม่นยำขึ้น
- ใช้ Loop เพื่อประมวลผลภาพจากหลายช่วงปี และสรุปผลลัพธ์

ชั่วโมงที่ 3-4: การทำโครงการวิเคราะห์ข้อมูลและการนำเสนอ

4. การวิเคราะห์และเปรียบเทียบข้อมูล (40 นาที)

- นักเรียนใช้โค้ดที่พัฒนาแล้ว เพื่อวิเคราะห์การเปลี่ยนแปลงของพื้นที่เมืองและพื้นที่สีเขียวในช่วงหลายปี
- คำนวณอัตราการเปลี่ยนแปลงของพื้นที่ และแสดงผลในรูปแบบของกราฟ

5. การอภิปรายและสะท้อนผลลัพธ์ (40 นาที)

- นักเรียนอภิปรายถึงผลการวิเคราะห์ เช่น
 - "พื้นที่เมืองขยายตัวเร็วแค่ไหน?"
 - "ผลกระทบต่อสิ่งแวดล้อมมีอะไรบ้าง?"
 - "สามารถนำข้อมูลนี้ไปใช้วางแผนนโยบายการพัฒนาเมืองได้หรือไม่?"

6. การนำเสนอและสรุปบทเรียน (40 นาที)

- นักเรียนสร้าง Dashboard หรือรายงานสรุป การวิเคราะห์ของตนเอง
- นำเสนอผลการวิเคราะห์ให้เพื่อนร่วมชั้น และสะท้อนความคิดเห็น
- ครูสรุปบทเรียน และอภิปรายถึงแนวทางการใช้ AI และ Machine Learning ในการศึกษาสิ่งแวดล้อม

3. การประเมิน (Assessment Strategy)

1. การวัดความเข้าใจด้านการเขียนโค้ด

- นักเรียนต้องสามารถ โหลดและแปลงภาพถ่ายดาวเทียม
- ต้องสามารถ ใช้ OpenCV และ Thresholding เพื่อแยกพื้นที่สีเขียวและพื้นที่เมือง
- ต้องสามารถ คำนวณเปอร์เซ็นต์ของพื้นที่เมืองและพื้นที่สีเขียวในแต่ละปี

2. การวัดความเข้าใจในปัญหาสิ่งแวดล้อม

- ใช้คำถามเชิงอภิปราย เช่น
 - "พื้นที่เมืองขยายตัวส่งผลกระทบต่ออุณหภูมิและมลพิษทางอากาศอย่างไร?"
 - "หากต้องการเพิ่มพื้นที่สีเขียว ควรใช้แนวทางใด?"
 - "เราสามารถใช้อ AI และ Machine Learning เพื่อช่วยแก้ปัญหาสิ่งแวดล้อมได้อย่างไร?"

3. รูปแบบการประเมิน

- แบบทดสอบสั้น ๆ (10 คำถาม) เพื่อวัดความเข้าใจด้านโค้ดและปัญหาสิ่งแวดล้อม
- ใบงานการทดลอง (Lab Worksheet) เพื่อดูว่านักเรียนสามารถวิเคราะห์ข้อมูลได้ถูกต้องหรือไม่
- การนำเสนอ (Presentation Assessment) เพื่อวัดความสามารถในการสื่อสารข้อมูลที่วิเคราะห์ได้

ผลลัพธ์และข้อสังเกต (Results and Observations)

1. ความสนใจของนักเรียน (Student Engagement)

การใช้ Python และ Image Processing ในการวิเคราะห์ภาพถ่ายดาวเทียมเป็นแนวทางที่ทำให้การเรียนรู้เกี่ยวกับปัญหาสิ่งแวดล้อมมีความน่าสนใจมากขึ้น นักเรียนสามารถเห็น การเปลี่ยนแปลงของพื้นที่เมือง และพื้นที่สีเขียวได้จริงผ่านข้อมูลภาพถ่ายดาวเทียม ซึ่งช่วยให้พวกเขาเข้าใจประเด็นสิ่งแวดล้อมในมิติที่ลึกขึ้น

สิ่งที่กระตุ้นความสนใจของนักเรียน

- นักเรียนส่วนใหญ่ให้ความสนใจ เมื่อเห็นผลลัพธ์การวิเคราะห์ที่มาจากข้อมูลจริง
- นักเรียนรู้สึกตื่นเต้นที่ได้ ใช้เครื่องมือ AI และ Machine Learning ในการศึกษาปัญหาสิ่งแวดล้อม
- การทำ โครงการวิเคราะห์ข้อมูลแบบทีม ทำให้นักเรียนมีความกระตือรือร้นและแข่งขันกันในการ สร้างแบบจำลองที่แม่นยำ

2. พัฒนาการด้านการเรียน (Learning Gains)

การเรียนรู้ผ่านการ เขียนโค้ดเพื่อวิเคราะห์ปัญหาสิ่งแวดล้อม ทำให้นักเรียนพัฒนาทักษะในหลายด้าน โดยมีหลักฐานที่แสดงให้เห็นถึงการเพิ่มขึ้นของทักษะด้าน โค้ดดิ้งและความเข้าใจเกี่ยวกับวิทยาศาสตร์ สิ่งแวดล้อม

พัฒนาการด้านการเขียนโค้ด

- นักเรียนสามารถ เขียนโค้ดเพื่อโหลดและวิเคราะห์ภาพถ่ายดาวเทียม ได้ด้วยตนเอง
- การเรียนรู้ผ่านการ เขียนโค้ดเพื่อวิเคราะห์ปัญหาสิ่งแวดล้อม ทำให้นักเรียนพัฒนาทักษะในหลายด้าน โดยมีหลักฐานที่แสดงให้เห็นถึงการเพิ่มขึ้นของทักษะด้าน โค้ดดิ้งและความเข้าใจเกี่ยวกับ วิทยาศาสตร์สิ่งแวดล้อม
- พัฒนาการด้านการเขียนโค้ด
- นักเรียนสามารถ เขียนโค้ดเพื่อโหลดและวิเคราะห์ภาพถ่ายดาวเทียม ได้ด้วยตนเอง

พัฒนาการด้านความเข้าใจในปัญหาสิ่งแวดล้อม

- นักเรียนสามารถ ระบุปัจจัยที่ส่งผลต่อการลดลงของพื้นที่สีเขียว ผ่านการวิเคราะห์ข้อมูล
- นักเรียนสามารถ อธิบายผลกระทบของ Urban Heat Island Effect และเสนอแนวทางแก้ไขที่อิง ข้อมูลจริง
- นักเรียนสามารถใช้ข้อมูลจากการวิเคราะห์เป็นหลักฐานประกอบการอภิปรายเกี่ยวกับแนวทางการ พัฒนาเมืองอย่างยั่งยืน

3. อุปสรรคหรือปัญหา (Challenges)

ข้อจำกัดด้านอุปกรณ์และการประมวลผล

- นักเรียนบางคนอาจไม่มี คอมพิวเตอร์ที่มีประสิทธิภาพสูงพอ สำหรับการประมวลผลภาพขนาดใหญ่
- การทำงานกับ ภาพถ่ายดาวเทียมที่มีขนาดใหญ่ อาจทำให้โปรแกรมทำงานช้าบนเครื่องคอมพิวเตอร์ที่มี RAM น้อย

ข้อจำกัดด้านเวลา

- การเรียนรู้เกี่ยวกับ การเขียนโค้ดและการวิเคราะห์ข้อมูล อาจต้องใช้เวลามากกว่าการสอนเชิงทฤษฎี
- นักเรียนบางคนต้องใช้เวลาเพิ่มในการ เรียนรู้แนวคิดพื้นฐานของ Image Processing

ปัญหาด้านการเชื่อมต่ออินเทอร์เน็ต

- หากใช้ Google Colab สำหรับการรันโค้ด อาจพบปัญหาเกี่ยวกับ การเชื่อมต่ออินเทอร์เน็ตที่ไม่เสถียร



สรุปผลและแนวทางในอนาคต (Conclusion and Future Directions)

1. สรุปวัตถุประสงค์และผลลัพธ์ของโครงการ

โครงการนี้มีเป้าหมายเพื่อพัฒนา เครื่องมือวิเคราะห์ภาพถ่ายดาวเทียมโดยใช้ Python และ OpenCV เพื่อให้นักเรียนสามารถ ติดตามแนวโน้มของการเปลี่ยนแปลงพื้นที่เมืองและพื้นที่สีเขียว โปรแกรมที่พัฒนาขึ้นช่วยให้สามารถ

- แยกแยะพื้นที่สีเขียวและพื้นที่เมืองจากภาพถ่ายดาวเทียม โดยใช้เทคนิค Thresholding และ Image Processing
- คำนวณเปอร์เซ็นต์ของพื้นที่ที่เปลี่ยนแปลง ในช่วงเวลาที่แตกต่างกัน
- แสดงผลในรูปแบบกราฟและ Dashboard ที่ช่วยให้สามารถมองเห็นแนวโน้มของการเปลี่ยนแปลงของเมือง
- กระตุ้นการเรียนรู้ของนักเรียน โดยการใช้ข้อมูลจริงและการลงมือทำผ่านโครงการ

2. แนวทางขยายผลและการปรับปรุงในอนาคต

1. เพิ่มฟีเจอร์การวิเคราะห์ข้อมูลที่ซับซ้อนขึ้น

- ใช้ Machine Learning และ Deep Learning ในการวิเคราะห์แนวโน้มของพื้นที่เมือง

2. เชื่อมโยงกับข้อมูลจากโลกจริงมากขึ้น

- ใช้ข้อมูลจาก Google Earth Engine, GISTDA, NASA, และ OpenAQ API เพื่อนำข้อมูลจริงมาวิเคราะห์
- ผสานข้อมูลจาก อุณหภูมิและคุณภาพอากาศ เพื่อวิเคราะห์ผลกระทบของการเปลี่ยนแปลงของพื้นที่เมืองต่อสิ่งแวดล้อม
- ใช้ข้อมูลจาก เซ็นเซอร์ IoT เพื่อตรวจสอบอุณหภูมิและคุณภาพอากาศแบบเรียลไทม์

3. พัฒนาเป็น Web Application หรือ Dashboard แบบโต้ตอบ

- อาจพัฒนาเป็น Web App ที่สามารถ ให้ผู้ใช้สามารถอัปโหลดภาพถ่ายดาวเทียมและดูผลลัพธ์แบบอินเตอร์แอคทีฟ
- ทำให้โปรแกรมสามารถ แสดงผลการวิเคราะห์ได้แบบเรียลไทม์ผ่านเว็บเบราว์เซอร์

3. แนวทางขยายผลด้านการสอน

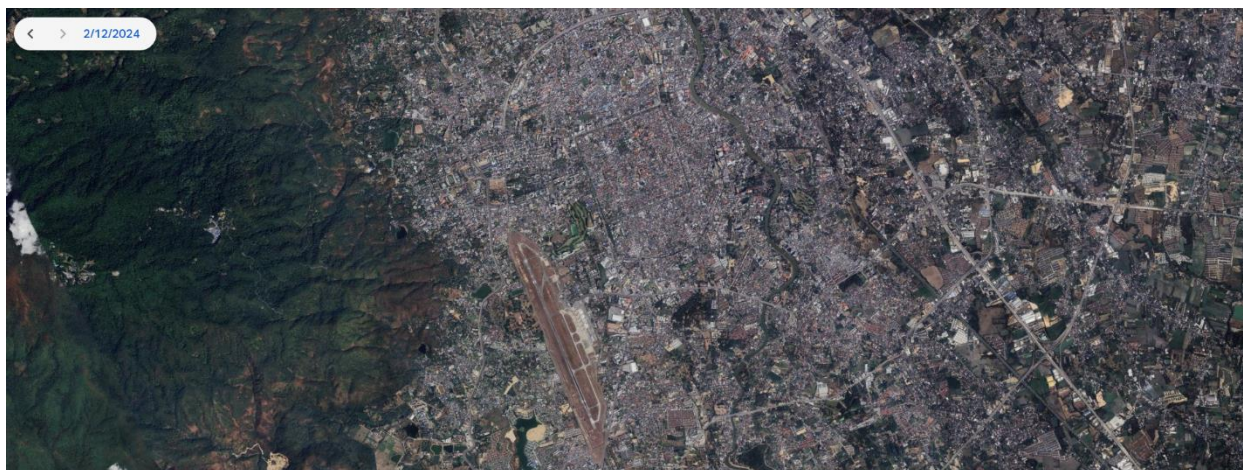
- นำไปใช้เป็น กิจกรรมในหลักสูตร STEM หรือวิชา Data Science สำหรับสิ่งแวดล้อม
- ขยายโครงการให้ นักเรียนทำโครงการแบบกลุ่มที่ใช้ข้อมูลจากพื้นที่ของตนเอง
- ปรับปรุงใบงานให้มีความท้าทายมากขึ้น เช่น การให้ นักเรียนพัฒนาโมเดล AI ของตนเอง

บรรณานุกรม
(References)

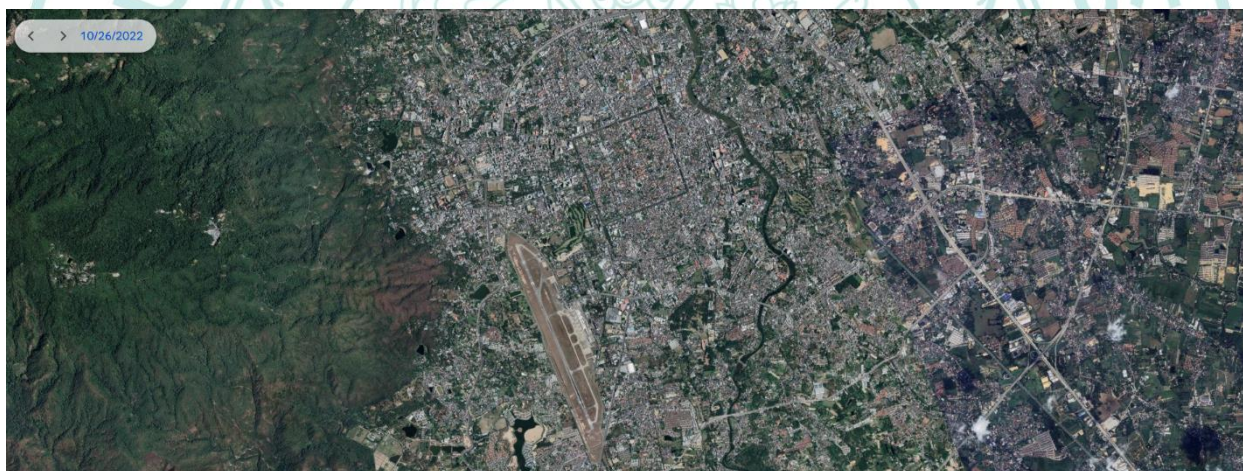
- NASA Earth Observations (NEO). (n.d.). *Global Satellite Observations for Climate and Environmental Monitoring*. Retrieved from <https://neo.sci.gsfc.nasa.gov>
- GISTDA. (n.d.). *Geo-Informatics and Space Technology Development Agency - Thailand*. Retrieved from <https://www.gistda.or.th>
- Google Earth Engine. (n.d.). *A planetary-scale platform for Earth science data & analysis*. Retrieved from <https://earthengine.google.com>



ภาคผนวก
(Appendices)

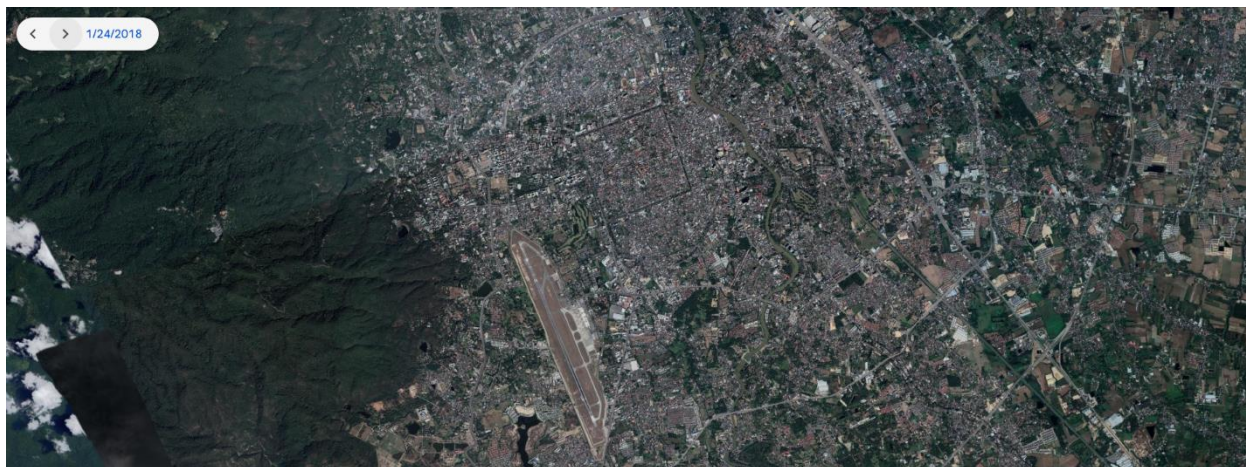


ภาพถ่ายดาวเทียมจังหวัดเชียงใหม่ ปี 2024

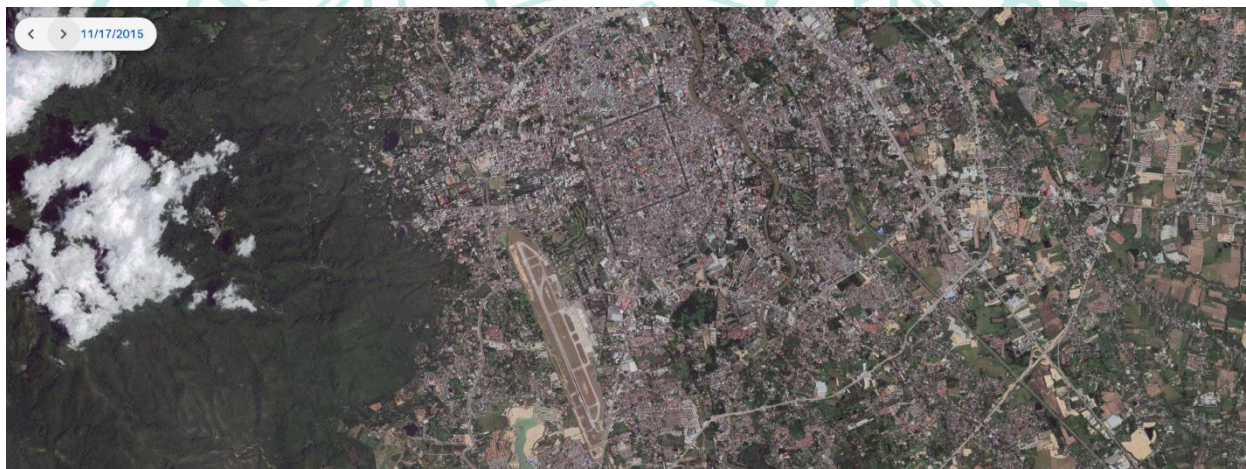


ภาพถ่ายดาวเทียมจังหวัดเชียงใหม่ ปี 2022

พ.ศ. ๒๕๕๖

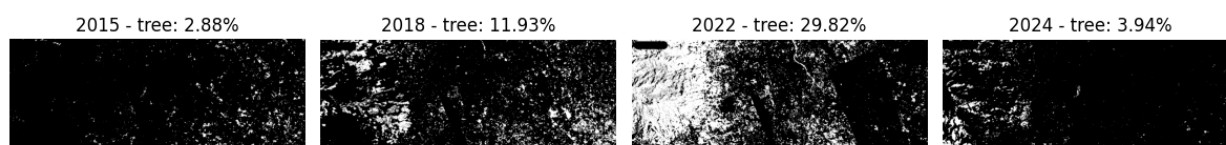
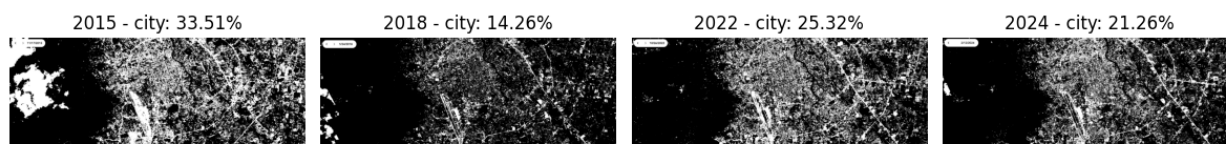


ภาพถ่ายดาวเทียมจังหวัดเชียงใหม่ ปี 2018

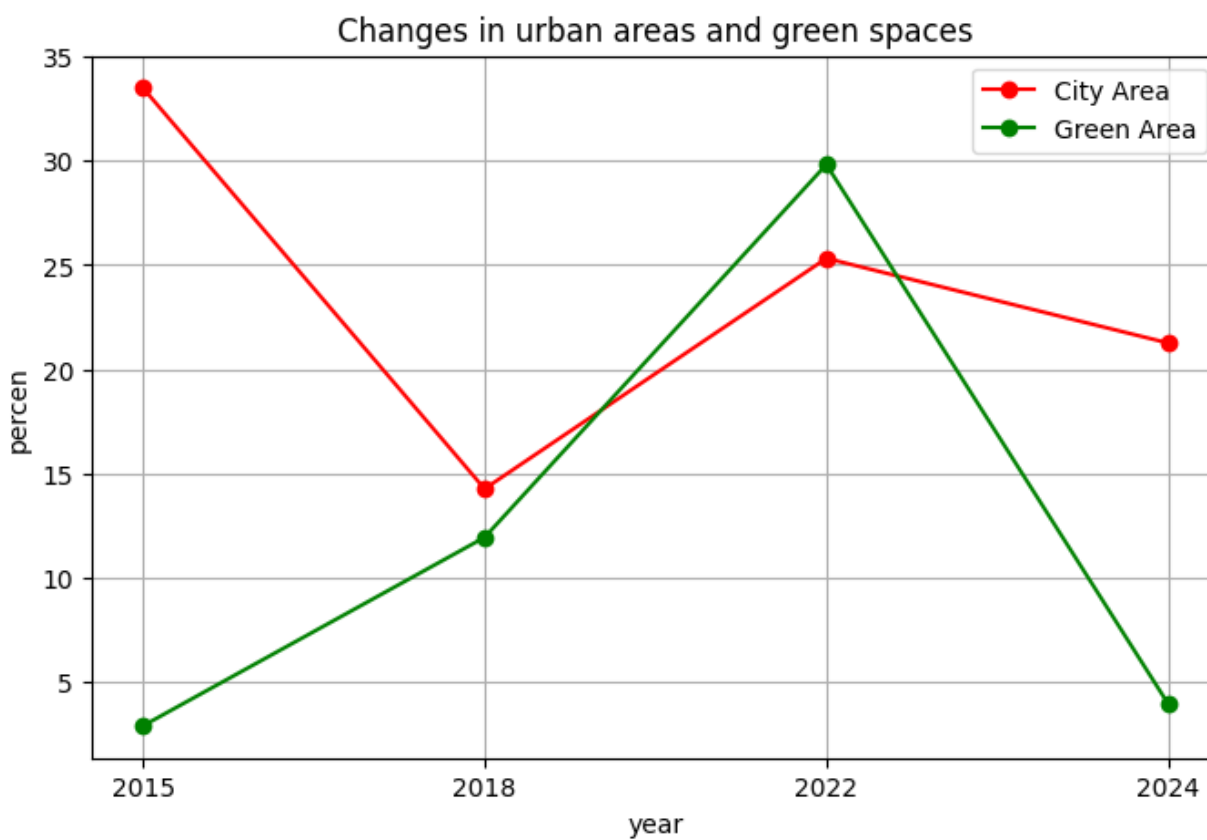


ภาพถ่ายดาวเทียมจังหวัดเชียงใหม่ ปี 2015





เปรียบเทียบภาพถ่ายดาวเทียมจังหวัดเชียงใหม่ ปี 2015 2018 2022 2024



กราฟเปรียบเทียบภาพถ่ายดาวเทียมจังหวัดเชียงใหม่ ปี 2015 2018 2022 2024

โค้ดทั้งหมด

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image_paths = {
    "2015": "/content/image_2015.png",
    "2018": "/content/image_2018.png",
    "2022": "/content/image_2022.png",
    "2024": "/content/image_2024.png",
}

import cv2
import numpy as np
import matplotlib.pyplot as plt

urban_percentages = {}
green_percentages = {}

plt.figure(figsize=(12, 8))

for i, (year, path) in enumerate(image_paths.items()):
    # โหลดภาพ
    image = cv2.imread(path)
    if image is None:
        print(f"Error: โหลดภาพ {year} ไม่ได้!")
        continue

    # แปลงเป็น HSV
    hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)

    # ตรวจสอบพื้นที่เมือง (สีเทา-น้ำตาล)
    lower_urban = np.array([0, 0, 110])
    upper_urban = np.array([180, 60, 255])
    urban_mask = cv2.inRange(hsv, lower_urban, upper_urban)
```

```

# ตรวจจับพื้นที่สีเขียว
lower_green = np.array([35, 40, 40])
upper_green = np.array([85, 255, 255])
green_mask = cv2.inRange(hsv, lower_green, upper_green)

# คำนวณเปอร์เซ็นต์ของพื้นที่เมืองและพื้นที่สีเขียว
urban_ratio = np.sum(urban_mask > 0) / urban_mask.size * 100
green_ratio = np.sum(green_mask > 0) / green_mask.size * 100

urban_percentages[year] = urban_ratio
green_percentages[year] = green_ratio

# แสดงผล
plt.subplot(2, 4, i+1)
plt.imshow(urban_mask, cmap='gray')
plt.title(f"{year} - city: {urban_ratio:.2f}%")
plt.axis("off")

plt.subplot(2, 4, i+5)
plt.imshow(green_mask, cmap='gray')
plt.title(f"{year} - tree: {green_ratio:.2f}%")
plt.axis("off")

plt.tight_layout()
plt.show()

# แสดงกราฟเปรียบเทียบ
years = list(urban_percentages.keys())
urban_values = list(urban_percentages.values())
green_values = list(green_percentages.values())

plt.figure(figsize=(8, 5))
plt.plot(years, urban_values, marker='o', linestyle='-', color='red', label="City Area")
plt.plot(years, green_values, marker='o', linestyle='-', color='green', label="Green Area")

```



```
plt.xlabel("year")  
plt.ylabel("percen")  
plt.title("Changes in urban areas and green spaces")  
plt.legend()  
plt.grid(True)  
plt.show()
```



