

## บทที่ 2

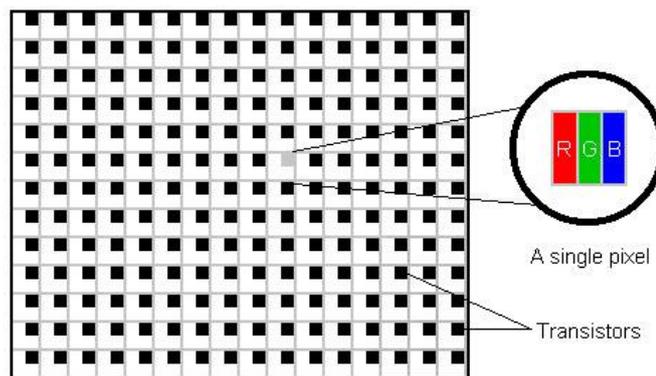
### ทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้อง

ในบทนี้กล่าวถึงทฤษฎีพื้นฐานและงานวิจัยที่เกี่ยวข้องกับการพัฒนาเทคนิคการเพิ่มประสิทธิภาพความถูกต้องของการอ่านชื่อจังหวัดบนป้ายทะเบียนรถยนต์ ซึ่งทฤษฎีพื้นฐานที่เกี่ยวข้องกับงานวิจัยประกอบด้วยเนื้อหา ดังนี้ การประมวลผลภาพดิจิทัล (Digital Image Processing), การรู้จำอักขระทางภาพ (Optical Character Recognition), กำหนดการพลวัต (Dynamic Programming) และป้ายทะเบียนรถยนต์ในประเทศไทย (Thai Vehicle License Plate)

#### การประมวลผลภาพดิจิทัล (Digital Image Processing)

การประมวลผลภาพดิจิทัลเป็นการนำภาพดิจิทัลมาประมวลผลเพื่อปรับปรุงหรือเปลี่ยนแปลงคุณภาพของภาพ โครงสร้างของภาพ จัดการข้อมูลภาพ และวิเคราะห์ภาพ เป็นต้น การประมวลผลภาพดิจิทัลประกอบไปด้วยเนื้อหา ดังนี้

ภาพดิจิทัล คือ ภาพที่เก็บอยู่ในรูปแบบของสัญญาณดิจิทัล (Digital Signal) เป็นการแสดงผลภาพในลักษณะสองมิติในหน่วยที่เรียกว่า พิกเซล ซึ่งเป็นหน่วยพื้นฐานของภาพ พิกเซล คือ ช่องหรือจุดภาพบนจอแสดงผลที่รวมกันเป็นภาพ และตัวเลขฐานสองจะถูกใช้ในการกำหนดค่ารายละเอียดของภาพ เช่น สี และ โทนของภาพ เป็นต้น



ภาพที่ 2.1 ภาพแสดงตารางพิกเซลของภาพดิจิทัล

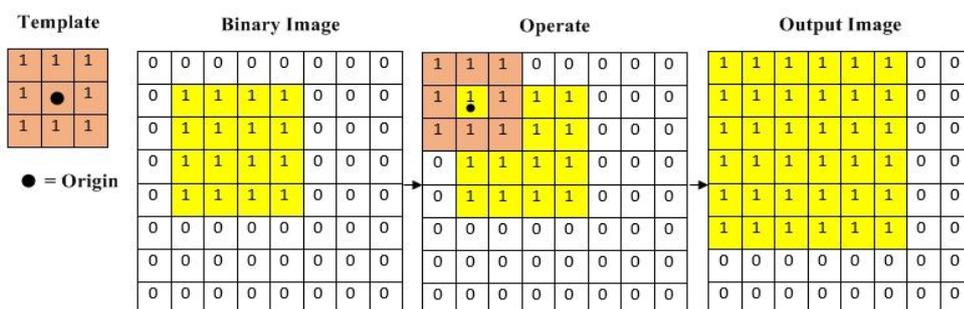
ที่มา : ([http://stou.ac.th/offices/rdec/nakorn/main/swf/DigitalTraining/digital01/digital01\\_04.html](http://stou.ac.th/offices/rdec/nakorn/main/swf/DigitalTraining/digital01/digital01_04.html))

## การประมวลผลภาพกับรูปร่างและโครงสร้างของภาพ (Morphological Processing)

การประมวลผลภาพกับรูปร่างและโครงสร้างของภาพ เป็นการประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ ดังนี้

### 1. การขยายภาพ (Dilation)

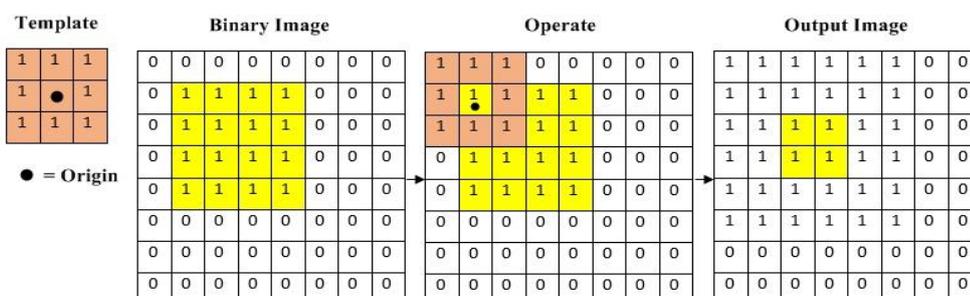
การขยายภาพเป็นการขยายโครงสร้างของภาพให้ใหญ่ขึ้น โดยพิจารณาข้อมูลภาพที่เป็นแบบไบนารี (Binary Image) โดยการสร้าง Template และนำ Template สแกนไปบนข้อมูลภาพตามลำดับตลอดทั้งภาพ เมื่อจุดเริ่ม (Origin) ของ Template ตรงกับตำแหน่งข้อมูลภาพที่พิกเซลมีค่าเท่ากับ 1 นั้นก็จะทำการยูเนียน (Union) Template นี้เข้ากับข้อมูลภาพ



ภาพที่ 2.2 การขยายภาพ (Dilation)

### 2. การย่อภาพ (Erosion)

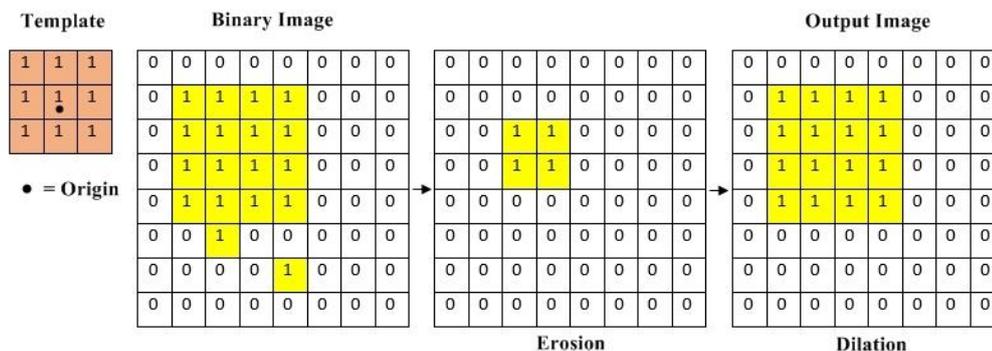
การย่อภาพเป็นของการลบข้อมูลภาพบริเวณขอบของภาพ การย่อภาพมีลักษณะคล้ายกับการขยายภาพ โดยการสร้าง Template แล้วนำ Template ไปสแกนบนข้อมูลภาพแบบไบนารีตามลำดับตลอดทั้งภาพซึ่งในขณะที่จุดเริ่มของ Template ตรงกับตำแหน่งข้อมูลภาพที่พิกเซลมีค่าเท่ากับ 1 ก็จะพิจารณา Template ว่าอยู่ในขอบเขตของข้อมูลภาพทั้งหมดหรือไม่ หากไม่ได้อยู่ในขอบเขตของข้อมูลภาพทั้งหมด ให้ทำการลบข้อมูลภาพตำแหน่งนั้น



ภาพที่ 2.3 การย่อภาพ (Erosion)

### 3. การเปิดช่องว่าง (Opening)

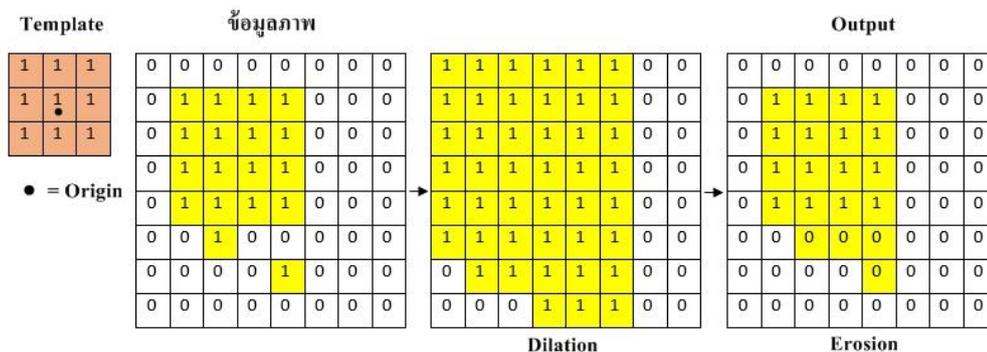
การเปิดช่องว่างเป็นการตัดหรือทำลายส่วนที่เกินมาของวัตถุในภาพ ซึ่งทำให้วัตถุในภาพมีขนาดเล็กกลง โดยทำการย่อภาพ (Erosion) ก่อนจากนั้นจึงทำการขยายภาพ (Dilation)



ภาพที่ 2.4 การเปิดช่องว่าง (Opening)

### 4. การปิดช่องว่าง (Closing)

การปิดช่องว่างเป็นการเติมเต็มส่วนที่โผล่หรือยื่นออกมาของวัตถุในภาพ ซึ่งทำให้ภาพมีการเชื่อมต่อกันมากขึ้น โดยทำการขยายภาพ (Dilation) ก่อนจากนั้นจึงทำการย่อภาพ (Erosion)



ภาพที่ 2.5 การปิดช่องว่าง (Closing)

การแบ่งส่วนภาพเป็นการแยกองค์ประกอบของรูปภาพออกจากกันตามลักษณะที่สำคัญของภาพ ในการแยกองค์ประกอบของภาพส่วนใหญ่จะทำหลังจากการปรับปรุงคุณภาพของภาพให้ดีขึ้นแล้ว ลักษณะการแยกจะขึ้นอยู่กับวัตถุประสงค์ที่จะนำไปใช้งานต่อ เช่น วิเคราะห์ลักษณะวัตถุ คำนวณขนาดของวัตถุ นับจำนวนวัตถุภายในภาพ เป็นต้น โดยมีหลักการพื้นฐานที่ใช้ในการแบ่งส่วนภาพ คือ เทรชโฮลด์ (Threshold) และการหาขอบภาพ (Edge Detection)

## 1. เทรชโฮลด์ (Threshold)

เทรชโฮลด์เป็นวิธีการแบ่งส่วนของภาพเกรย์สเกล (Gray Scale Image) โดยใช้คุณสมบัติของค่าความเข้มแสงหรือสีมาเป็นตัวตัดสิน เช่น กำหนดค่า Threshold ให้มีค่าเท่ากับ 120 แล้วค่าพิกเซลในภาพที่มีค่าน้อยกว่าหรือเท่ากับ 120 จะถูกปรับค่าเป็น 0 ในทางกลับกันถ้าค่าพิกเซลไหนมากกว่า 120 จะถูกปรับค่าเป็น 255

230	255	210	231	255	170	200	150	230	255
240	230	199	222	255	240	230	230	240	230
245	240	0	150	170	20	120	119	245	240
244	244	20	230	240	15	150	90	244	244
210	231	20	200	150	16	230	80	210	231
199	222	20	230	230	89	100	100	199	222
200	150	15	17	170	90	210	231	200	150
230	230	16	18	240	95	199	222	230	230
255	170	255	170	210	231	255	170	255	170
255	240	255	240	199	222	255	240	255	240

input image size 10\*10

255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	0	255	255	0	0	0	255	255
255	255	0	255	255	0	255	0	255	255
255	255	0	255	255	0	255	0	255	255
255	255	0	255	255	0	0	0	255	255
255	255	0	0	255	0	255	255	255	255
255	255	0	0	255	0	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255

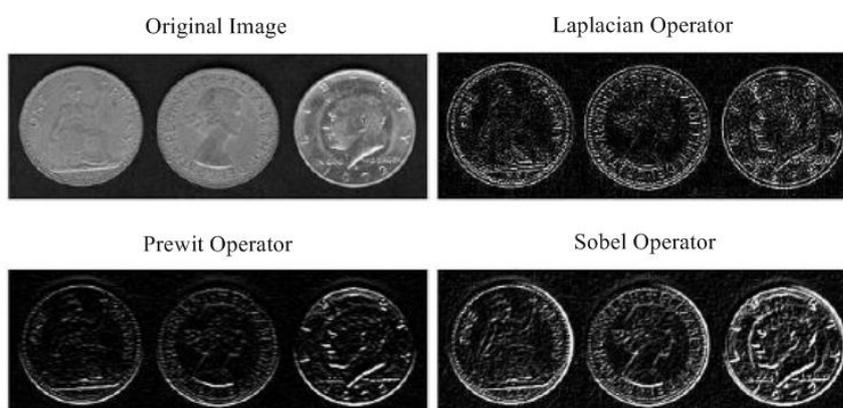
output image size 10\*10

ภาพที่ 2.6 ตัวอย่างการทำเทรชโฮลด์จากการคำนวณด้วยมือ

ที่มา : (<http://kengzer.blogspot.com/2013/01/emgu-cv-lab-6-threshold.html>)

## 2. การหาขอบภาพ (Edge Detection)

การหาขอบภาพเป็นวิธีการหาเส้นรอบวัตถุที่อยู่ในภาพ ซึ่งอาศัยค่าความแตกต่างของความเข้มแสงของค่าสีในพื้นที่ของภาพ หรือลายเส้นที่มีระดับความแตกต่างของความเข้มแสง ซึ่งมีวิธีที่ใช้ในการหาหลายวิธี เช่น Sobel Operator, Prewit Operator และ Laplacian Operator เป็นต้น

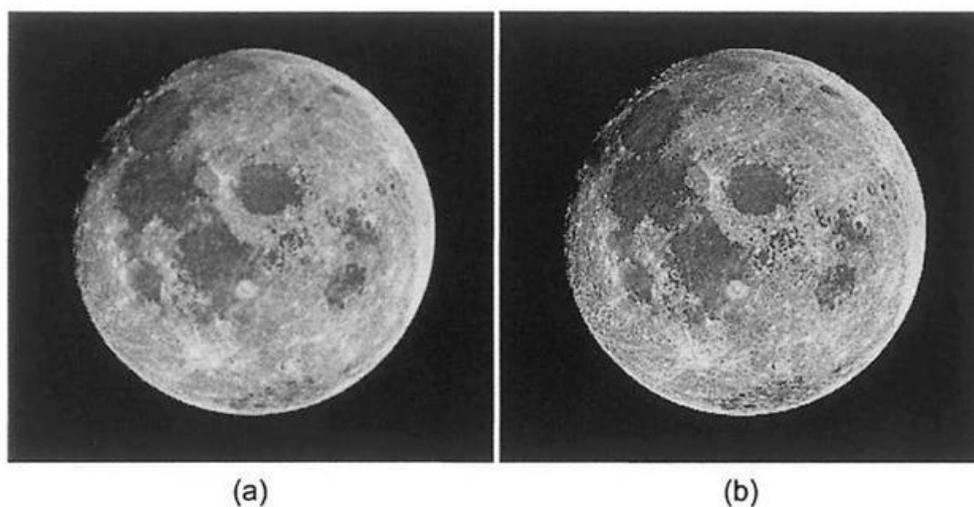


ภาพที่ 2.7 ตัวอย่างภาพผลลัพธ์ที่ได้จากการหาขอบภาพ

ที่มา : ([https://app.enit.kku.ac.th/mis/administrator/doc\\_upload/20120312103907.pdf](https://app.enit.kku.ac.th/mis/administrator/doc_upload/20120312103907.pdf))

การกรองข้อมูลภาพ คือการนำภาพไปผ่านตัวกรองสัญญาณเพื่อให้ได้ภาพผลลัพธ์ที่มีคุณสมบัติของภาพที่ดีขึ้น วัตถุประสงค์ของการกรองข้อมูลภาพ คือ การเน้นรายละเอียดของภาพ (Image Enhancement) เนื่องจากในการใช้งานจริงภาพที่ได้มาอาจมีความคมชัดน้อย หรือในกรณีที่ภาพมีความไม่สมบูรณ์ เช่น มีสิ่งรบกวน (Noise) สามารถใช้เทคนิคการกรองสัญญาณภาพเพื่อกำจัดสิ่งรบกวน หรือลดทอน (Attenuate) คุณสมบัติบางประการเพื่อให้ได้ภาพที่มีคุณสมบัติที่ดีขึ้น ประเภทของตัวกรองข้อมูลภาพแบ่งออกเป็น 4 ประเภท ตามลักษณะการเลือกความถี่ของตัวกรองข้อมูลภาพ ดังนี้

1. ตัวกรองความถี่ต่ำผ่าน (Low-pass Filter)
2. ตัวกรองความถี่สูงผ่าน (High-pass Filter)
3. ตัวกรองแถบความถี่ผ่าน (Band-pass Filter)
4. ตัวกรองหยุดแถบความถี่ (Band-stop Filter)

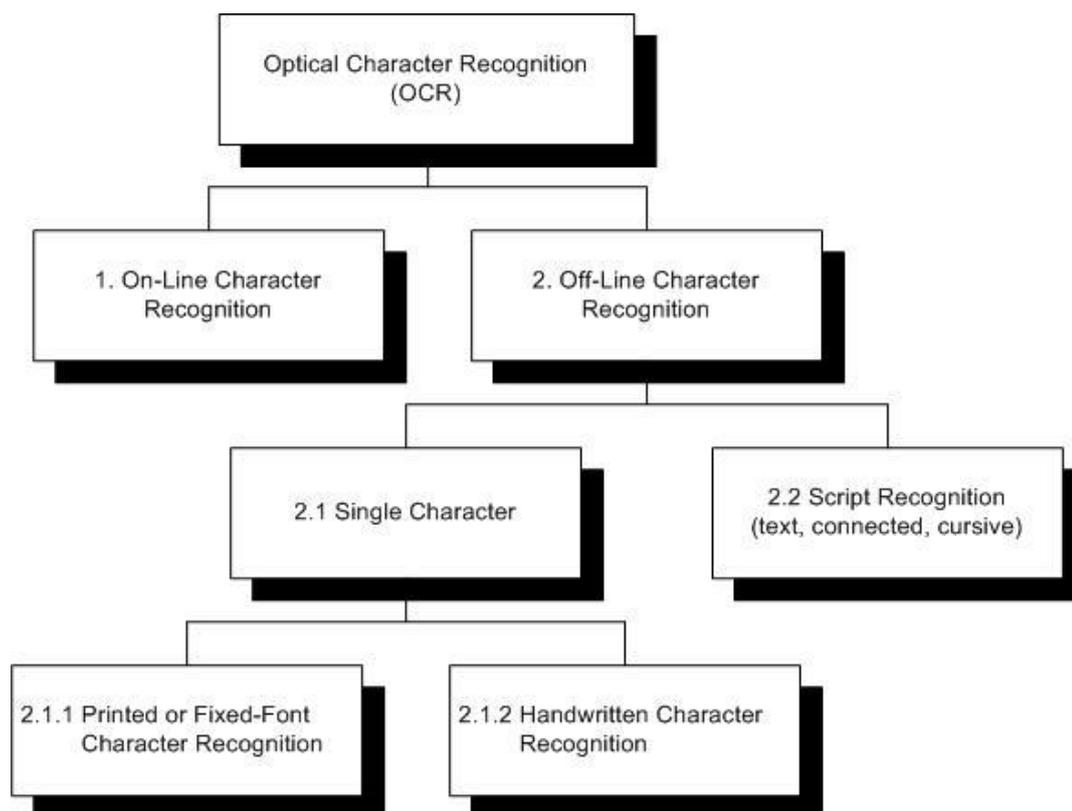


ภาพที่ 2.8 ภาพการกรองความถี่ต่ำผ่าน (Low-pass Filter) (a) ภาพต้นฉบับ, (b) ผลลัพธ์  
ที่มา : ([http://what-when-how.com/wp-content/uploads/2011/09/tmp6150\\_thumb2\\_thumb.jpg](http://what-when-how.com/wp-content/uploads/2011/09/tmp6150_thumb2_thumb.jpg))

## การรู้จำอักขระทางภาพ (Optical Character Recognition: OCR)

การรู้จำอักขระทางภาพ คือ การแปลงข้อมูลที่อยู่ในรูปแบบของเอกสารรูปภาพ ให้อยู่ในรูปแบบของเอกสารข้อความ (Text File) ทำให้สะดวกในการปรับแต่ง แก้ไขเอกสาร และประหยัดพื้นที่ในการจัดเก็บข้อมูล การรู้จำอักขระทางภาพสามารถแบ่งประเภทตามคุณลักษณะตามแหล่งที่มาของตัวอักษรได้ดังนี้

1. การรู้จำตัวอักษรแบบออนไลน์ (On-Line Character Recognition)
2. การรู้จำตัวอักษรแบบออฟไลน์ (Off-Line Character Recognition)
  - 2.1. ตัวอักษร โคด (Single Character)
    - 2.1.1. การรู้จำตัวพิมพ์ฟอนต์เฉพาะ (Printed or Fixed-Font Character Recognition)
    - 2.1.2. การรู้จำลายมือเขียนแบบตัว โคด (Handwritten Character Recognition)
  - 2.2. การรู้จำลายมือแบบเขียนต่อเนื่อง (Script Recognition)



ภาพที่ 2.9 แผนผังประเภทของ OCR

ที่มา : (<http://thaiocr.phaisarn.com/category.html>)

โครงสร้างของการรู้จำอักขระทางภาพประกอบไปด้วยขั้นตอนการทำงานหลัก 3 ขั้นตอนดังนี้



ภาพที่ 2.10 ขั้นตอนการทำงานของ OCR

ที่มา : ([https://app.enit.kku.ac.th/mis/administrator/doc\\_upload/20120312103907.pdf](https://app.enit.kku.ac.th/mis/administrator/doc_upload/20120312103907.pdf))

### 1. การประมวลผลขั้นต้น (Pre-Processing)

การประมวลผลขั้นต้นเป็นขั้นตอนแรกของกระบวนการรู้จำ เพราะก่อนที่โปรแกรมจะสามารถวิเคราะห์ลักษณะของตัวอักษรว่าเป็นตัวอักษรใดได้นั้น จะต้องมีการปรับปรุงภาพเพื่อให้ความเหมาะสมในการตรวจสอบและวิเคราะห์ตัวอักษร ซึ่งในบางครั้งภาพที่นำมารู้จำอาจมีสิ่งรบกวนจากสภาพแวดล้อม เช่น แสงสะท้อน เป็อนโคลน ตัวอักษรเอียง เป็นต้น ซึ่งมีผลต่อความถูกต้องของการรู้จำ โดยมีขั้นตอนดังต่อไปนี้

1.1 การกรองข้อมูลแทรกซ้อน (Noise Filtering) เป็นวิธีการลดทอนสิ่งที่ไม่พึงประสงค์ที่ปรากฏอยู่บนภาพ ในการลดทอนสิ่งรบกวนของภาพ

1.2 การปรับแต่งข้อมูล (Normalization) เป็นวิธีการปรับแต่งข้อมูลเพื่อให้ภาพที่จะนำไปประมวลผลนั้น มีความเหมาะสมในการประมวลผลภาพ เช่น การปรับขนาดภาพตัวอักษร การปรับความเอียงของภาพตัวอักษร เป็นต้น

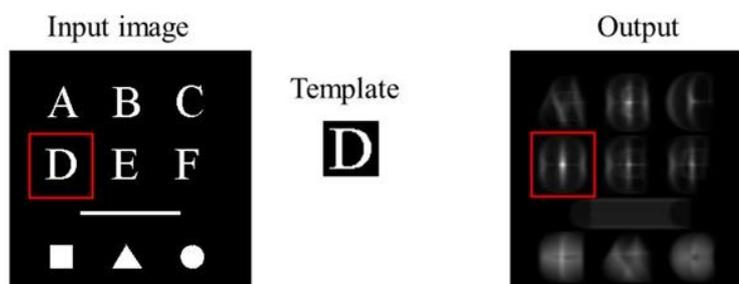
1.3 การตัดแบ่งพื้นที่ใช้งาน (Cropping) เป็นการแยกเอาเฉพาะภาพตัวอักษรส่วนที่ต้องการนำไปวิเคราะห์ เพื่อความถูกต้องของข้อมูลและลดจำนวนข้อมูลของรูปภาพที่ไม่จำเป็นในการวิเคราะห์

1.4 การสกัดลักษณะสำคัญ (Feature Extraction) เป็นการนำเอาโครงสร้างพื้นฐานที่สำคัญของตัวอักษรแต่ละตัวออกมา เพื่อส่งต่อเป็นอินพุตสำหรับขั้นตอนการรู้จำ ซึ่งจะนำมาเปรียบเทียบกับโครงสร้างพื้นฐานที่กำหนดไว้ ตัวอย่างเช่น สำหรับตัวอักษรภาษาไทยเราอาจกำหนดว่าตัวอักษรภาษาไทยทั้งหมดประกอบด้วยโครงสร้างพื้นฐานคือ เส้นตรงทั้งแนวตั้งและแนวนอน เส้นเอียง เส้นโค้ง วงกลม วงรี จุดตัด เป็นต้น

## 2. การรู้จำ (Recognition)

การรู้จำเป็นขั้นตอนที่สำคัญของการรู้จำอักขระทางภาพ เพราะเป็นขั้นตอนที่ทำหน้าที่ตัดสินใจว่าภาพที่ส่งเข้าไปวิเคราะห์นั้นเป็นตัวอักษรอะไร วิธีการรู้จำได้แบ่งออกได้ 4 วิธีดังนี้

2.1 วิธีการเข้าคู่รูปแบบ (Template Matching) เป็นวิธีการนำรูปภาพที่ต้องการไปเปรียบเทียบกับรูปแบบตัวอย่าง (Template) ซึ่งจะเก็บค่าและกำหนดลักษณะสำคัญที่สามารถแยกความแตกต่างของตัวอักษรได้ เพื่อตรวจสอบความคล้ายคลึงของภาพทั้งสอง



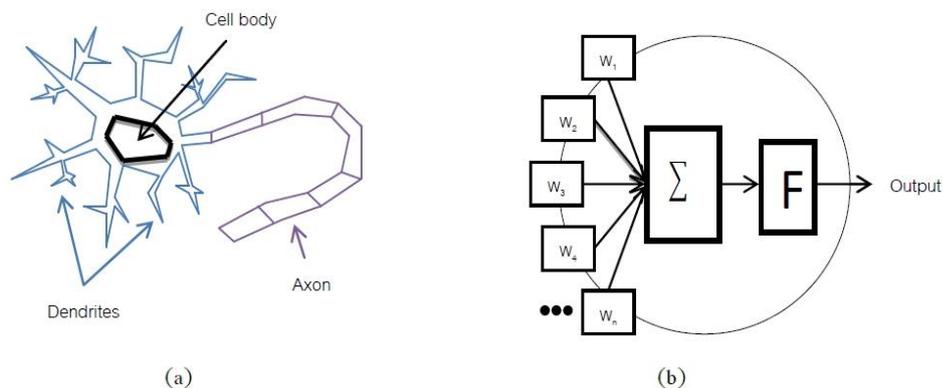
ภาพที่ 2.11 ตัวอย่างการเข้าคู่รูปแบบ (Template Matching)

ที่มา : (<http://slideplayer.com/slide/5147244/>)

2.2 วิธีทางสถิติ (Statistical Approach) เป็นวิธีที่อาศัยผลลัพธ์ที่ได้จากการสกัดลักษณะสำคัญ เมื่อนำภาพเข้าสู่ขั้นตอนการรู้จำเพื่อตรวจสอบลักษณะเฉพาะของตัวอักษรแต่ละตัว ซึ่งใช้การตัดสินใจตามหลักการทางสถิติจากค่าความน่าจะเป็น หรือฟังก์ชันการแจกแจงความน่าจะเป็น

2.3 วิธีการวิเคราะห์ทางโครงสร้าง (Structural Analysis) เป็นวิธีที่อาศัยหลักการทางโครงสร้างของตัวอักษร เพราะตัวอักษรหรือตัวเลขแต่ละตัวนั้นล้วนมีองค์ประกอบเฉพาะ ในขั้นตอนการรู้จำลักษณะของตัวอักษรนั้นจะถูกตรวจสอบลักษณะเฉพาะหรือวิเคราะห์กฎการเขียนตัวอักษร เพื่อที่จะสามารถระบุได้ว่าเป็นตัวอักษรใด

2.4 วิธีทางโครงข่ายประสาทเทียม (Artificial Neural Network: ANN) เป็นวิธีการจำลองการทำงานโครงข่ายประสาทของมนุษย์ (Biological Neurons) ในรูปแบบโมเดลทางคณิตศาสตร์ หรือ โมเดลทางคอมพิวเตอร์สำหรับประมวลผลสารสนเทศด้วยการคำนวณแบบคอนเนกชันนิสต์ (Connectionist) ซึ่งประกอบด้วยส่วนของการประมวลผลที่เรียกว่านิวรอน (Neuron) โดยทุกนิวรอนสามารถมีอินพุตได้หลายอินพุต แต่มีเอาต์พุตเพียงเอาต์พุตเดียว และทุกเอาต์พุตจะแยกไปยังอินพุตของนิวรอนอื่นภายในโครงข่าย การติดต่อกันภายในระหว่างนิวรอนจะมีน้ำหนัก (Weight) เป็นตัวกำหนดกำลังของการติดต่อและช่วยในการตัดสินใจ



ภาพที่ 2.12 Neural Model (a) ในสมองมนุษย์, (b) ในคอมพิวเตอร์

ที่มา : ([https://app.enit.kku.ac.th/mis/administrator/doc\\_upload/20120312103907.pdf](https://app.enit.kku.ac.th/mis/administrator/doc_upload/20120312103907.pdf))

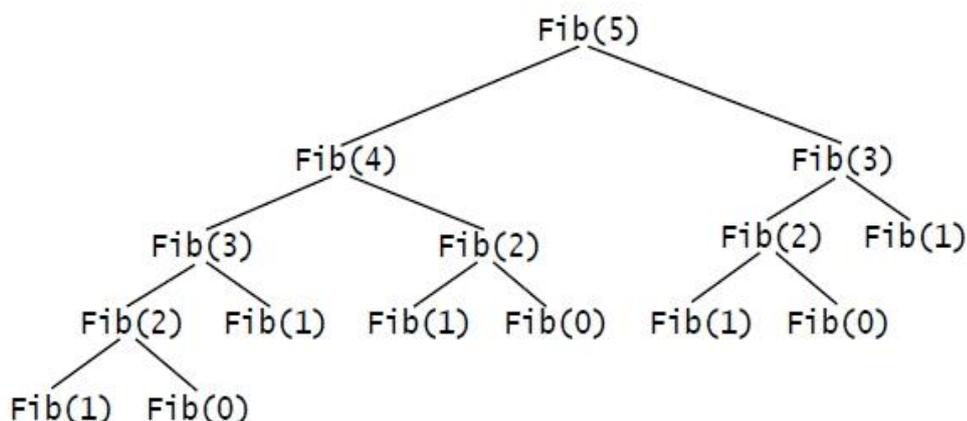
### กำหนดการพลวัต (Dynamic Programming)

กำหนดการพลวัตเป็นกลวิธีการแก้ปัญหาคำนวณที่มีลักษณะซับซ้อน แนวคิดของกลวิธีนี้คือการแบ่งปัญหาใหญ่ออกเป็นปัญหาย่อยหลายปัญหาที่มีขนาดเล็กกว่า และสามารถแก้ปัญหานั้นได้ง่ายกว่า โดยหาคำตอบของแต่ละปัญหาย่อย แล้วนำคำตอบของปัญหาย่อยที่ได้มารวมกันเพื่อกลายเป็นคำตอบของปัญหาใหญ่ โดยทั่วไปจะใช้กำหนดการพลวัตเพื่อแก้ปัญหามันที่เรียกว่า ปัญหาการหาค่าที่เหมาะสมที่สุด (Optimization Problems) โดยผลลัพธ์ที่ต้องการจากปัญหามันก็มีตัวประเมินคุณภาพว่าต้องดีที่สุด น้อยสุด หรือมากที่สุดเป็นต้น เรียกรวมว่าเป็นค่าที่เหมาะสมที่สุด (Optimal) โดยคุณภาพของผลเฉลยนั้นวัดจากค่าของฟังก์ชันจุดประสงค์ (Objective Function) นอกจากนี้สิ่งที่กำหนดไว้ในปัญหาการหาค่าที่เหมาะสมที่สุดก็คือเซตของเงื่อนไขบังคับ (Constraints) เรียกผลเฉลยใดที่อยู่ภายใต้เงื่อนไขบังคับที่กำหนดว่าเป็นผลเฉลยที่เป็นไปได้ (Feasible Solution) และเรียกผลเฉลยที่เป็นไปได้ตัวที่ได้ค่าของฟังก์ชันจุดประสงค์เหมาะสมที่สุดว่าผลเฉลยที่เหมาะสมที่สุด (Optimal Solution) ลักษณะเฉพาะของปัญหาการหาค่าที่เหมาะสมที่สุดที่ใช้กำหนดการพลวัตหาผลเฉลยที่เหมาะสมที่สุดและมีประสิทธิภาพนั้น จะต้องมิลักษณะเฉพาะของปัญหาดังต่อไปนี้

1. ผลเฉลยที่เหมาะสมที่สุดของปัญหาใหญ่ประกอบด้วยผลเฉลยที่เหมาะสมที่สุดของปัญหาย่อย คุณสมบัตินี้เรียกว่าโครงสร้างย่อยเหมาะสมที่สุด (Optimal Substructure) ซึ่งเป็นลักษณะเฉพาะของปัญหาการหาค่าที่เหมาะสมที่สุด ซึ่งผลเฉลยที่เหมาะสมที่สุดประกอบด้วยผลเฉลยที่เหมาะสมที่สุดของปัญหาย่อย ทำให้สามารถประกันได้ว่าผลเฉลยที่หาได้ด้วยการพลวัตจะเป็นผลเฉลยที่เหมาะสมที่สุด

2. ปัญหาย่อยที่ต้องการหาผลเฉลยมีการซ้อนเหลื่อมกัน (Overlapping Subproblems) เพราะเมื่อมีการแบ่งปัญหาใหญ่เป็นปัญหาย่อย และในตัวปัญหาย่อยนั้นเมื่อถูกแบ่งต่อจะประกอบด้วยปัญหาย่อยที่เหมือนกัน ส่งผลให้ต้องแก้ปัญหาย่อยที่ซ้อนเหลื่อมกันนี้ซ้ำแล้วซ้ำอีก เช่น เจียนโปรแกรม

เวียนเกิดเพื่อหาค่าของจำนวนฟีโบนัชชี (Fibonacci) คือ ลำดับของจำนวนเต็ม โดยมีนิยามของความสัมพันธ์ว่า จำนวนถัดไปเท่ากับผลบวกของจำนวนสองจำนวนก่อนหน้า และสองจำนวนแรกก็คือ 0 และ 1 ตามลำดับ ดังตัวอย่างในภาพที่ 2.13



ภาพที่ 2.13 ตัวอย่างปัญหาย่อยที่ซ้อนเหลื่อมกันจากการหาค่าฟีโบนัชชี

ที่มา : (<http://bus.rmutt.ac.th/~suwat/Computer%20Programming%201/Document/algorithms.pdf>)

จากภาพที่ 2.13 การหาค่าฟีโบนัชชี  $Fib(5)$  แสดงด้วยต้นไม้จำลองการเวียนเกิด ซึ่งทำให้เกิดการเรียก  $Fib(n)$  ที่มีค่าของ  $n$  เหมือนกับที่เคยเรียกซ้ำแล้วซ้ำอีก จากภาพเรียก  $Fib(3)$  ซ้ำ 2 ครั้ง เรียก  $Fib(2)$  ซ้ำ 3 ครั้ง เป็นต้น ซึ่งเรียกว่าเป็นการซ้อนเหลื่อมของปัญหาย่อย โดยมีวิธีการแก้ปัญหาคือการซ้อนเหลื่อมกันของปัญหาย่อย ดังนี้

1. ใช้กลวิธีแบ่งแยกและเอาชนะ (Divide and Conquer) คือการแบ่งปัญหาใหญ่ออกเป็นปัญหาย่อย แต่มีการจำคำตอบของปัญหาย่อยที่เคยแก้ไขมาแล้ว เมื่อพบปัญหาที่รู้คำตอบแล้วจากผลในอดีตก็คืนคำตอบได้เลยไม่ต้องหาคำตอบใหม่ ลักษณะเช่นนี้จะเป็นการทำงานแบบบนลงล่าง (Top-Down) เช่น การเรียงลำดับแบบเร็ว (Quick Sort) การค้นแบบทวิภาค (Binary Search) การเรียงลำดับแบบผสาน (Merge Sort) เป็นต้น

2. ใช้การจำคำตอบเหมือนกันแต่จะใช้วิธีการหาและจำคำตอบของปัญหาเล็กทั้งหมดที่มี จากนั้นหาคำตอบของปัญหาที่ใหญ่ขึ้นทั้งหมด โดยนำคำตอบของปัญหาที่รู้และนำมาใช้สร้างคำตอบของปัญหาที่ใหญ่ขึ้น กระทำเช่นนี้กับปัญหาที่ใหญ่ขึ้นอีกทั้งหมด จนกระทั่งพบคำตอบของปัญหาที่ต้องการแก้ไข ลักษณะเช่นนี้เป็นการทำงานแบบล่างขึ้นบน (Bottom-Up)

ขั้นตอนการออกแบบด้วยกำหนดการพลวัต

1. จำแนกโครงสร้างย่อยที่เหมาะสมที่สุดของปัญหาในรายละเอียดว่าจะนำผลเฉลยที่เหมาะสมที่สุดของปัญหาย่อยมาประกอบกันอย่างไร จึงได้ผลเฉลยที่เหมาะสมที่สุดของปัญหา
2. เขียนความสัมพันธ์เวียนเกิดของค่าของผลเฉลยที่เหมาะสมที่สุดจากโครงสร้างย่อยที่เหมาะสมที่สุดที่ได้ในขั้นตอนแรก
3. หาวิธีจำกัดขอบ เช่น ใช้ตารางสองมิติ หรือใช้โครงสร้างข้อมูลแบบอื่น เขียนอัลกอริทึมที่ใช้คำตอบจากที่จำมาคำนวณและเติมคำตอบใหม่ โดยไล่หาจากปัญหามิติเล็กไปสู่ปัญหามิติใหญ่ที่ต้องการ
4. หาวิธีสร้างผลเฉลยที่เหมาะสมที่สุดจากสิ่งที่ได้คำนวณในขั้นตอนที่ 3 เพื่อใช้เป็นข้อมูลในการสร้างผลเฉลยที่เหมาะสมที่สุด

ลำดับย่อยร่วมยาวสุด (Longest Common Subsequence: LCS) เป็นวิธีสำหรับเปรียบเทียบของสายอักขระสองสายเพื่อแก้ปัญหาลำดับย่อยร่วมยาวสุด ซึ่งเป็นขั้นตอนวิธีการเขียนโปรแกรมเชิงพลวัต โดยใช้ปริภูมิเชิงเส้น (Linear Space) และตารางสองมิติในการจำกัดขอบของลำดับย่อยเพื่อหาลำดับย่อยร่วมยาวสุดรวมถึงหาการเรียงตัวของลำดับย่อย โดยนิยามของแอลซีเอส คือ กำหนดให้ลำดับ  $X = (x_1, x_2, \dots, x_m)$  และลำดับ  $Y = (y_1, y_2, \dots, y_n)$  โดยเรียก  $Z = (z_1, z_2, \dots, z_k)$  ว่าเป็นลำดับย่อยร่วม (Common Subsequence) ก็ต่อเมื่อ  $Z$  เป็นลำดับย่อยของทั้ง  $X$  และ  $Y$  ตัวอย่างเช่น  $X = (S, O, M, C, H, A, I)$  และ  $Y = (C, H, U, A, N)$  มีลำดับย่อยร่วมคือ  $Z = (C), (H), (A), (C, H), (C, A), (H, A)$  และ  $(C, H, A)$  เป็นต้น โดยสามารถเขียนเป็นขั้นตอนการออกแบบด้วยกำหนดการพลวัตได้ดังนี้

1. จำแนกโครงสร้างย่อยที่เหมาะสมที่สุดของปัญหาลำดับย่อยร่วมยาวสุด เป็นการหาลำดับย่อยร่วมที่ยาวสุดของสองลำดับ เรียกว่า LCS โดยสิ่งที่ต้องการหาคือ LCS ของลำดับ  $X$  และ  $Y$  ซึ่งเป็นปัญหาที่มีโครงสร้างย่อยที่เหมาะสมที่สุดจากนิยามข้างต้นกำหนดให้  $Z$  เป็น LCS ของลำดับ  $X$  และ  $Y$  เขียนได้ว่า  $LCS(X, Y)$  โครงสร้างย่อยที่เหมาะสมที่สุดของ LCS สามารถแสดงได้ด้วยการแยกพิจารณาเป็นสองกรณีดังนี้

$$1.1 \text{ ถ้า } x_m = y_n \text{ จะได้ว่า } z_k = x_m \text{ และ } Z_{k-1} = LCS(X_{m-1}, Y_{n-1})$$

1.2 ถ้า  $x_m \neq y_n$  มีสองกรณีย่อยต้องพิจารณาคือ

$$1.2.1 \text{ ถ้า } z_k \neq x_m \text{ จะได้ว่า } Z = LCS(X_{m-1}, Y)$$

$$1.2.2 \text{ ถ้า } z_k \neq y_n \text{ จะได้ว่า } Z = LCS(X, Y_{n-1})$$

2. เขียนความสัมพันธ์เวียนเกิดของค่าผลเฉลยเหมาะที่สุดจากโครงสร้างย่อยเหมาะที่สุด โดยกำหนดให้  $c(i, j)$  เก็บความยาวของ LCS และสามารถเขียนความสัมพันธ์เวียนเกิดของความยาวลำดับย่อยร่วมยาวสุด ซึ่งแบ่งออกได้เป็น 3 กรณีดังนี้ กรณีแรกให้  $c(i, j) = 0$  เมื่อตำแหน่ง  $i = 0$  หรือ  $j = 0$  และกรณีที่สองเมื่อข้อมูลตำแหน่ง  $X_i = Y_j$  ให้  $c(i, j) = 1 + c(i-1, j-1)$  และกรณีที่สามหากข้อมูลที่ตำแหน่ง  $X_i \neq Y_j$  จะพิจารณาว่าตำแหน่งของข้อมูลไหนมีค่าความยาวมากที่สุดจะเป็นค่าของ  $c(i, j)$  ระหว่าง  $c(i-1, j)$  กับ  $c(i, j-1)$  ซึ่งจะเห็นการซ้อนเหลื่อมของปัญหาย่อยตรงกรณีสุดท้ายของความสัมพันธ์เวียนเกิดนี้จากการหาค่าความยาวที่มากที่สุดซ้ำแล้วซ้ำอีก

$$c(i, j) = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ 1 + c(i-1, j-1) & \text{if } x_i = y_j \\ \max(c(i-1, j), c(i, j-1)) & \text{if } x_i \neq y_j \end{cases}$$

ภาพที่ 2.14 ความสัมพันธ์เวียนเกิดของความยาว LCS

(ที่มา: <http://bus.rmutt.ac.th/~suwat/Computer%20Programming%201/Document/algorithms.pdf>)

3. การจำคำตอบและที่มาของคำตอบลำดับย่อยร่วม โดยใช้ตารางอาร์เรย์ 2 มิติ เพื่อเก็บค่าความยาวของลำดับย่อยร่วมไว้ในตาราง  $c[m,n]$  และที่มาของลำดับย่อยร่วมไว้ใน  $b[m,n]$

```

LCS-LENGTH(X, Y)
1  m ← length[X]
2  n ← length[Y]
3  for i ← 1 to m
4      do c[i, 0] ← 0
5  for j ← 0 to n
6      do c[0, j] ← 0
7  for i ← 1 to m
8      do for j ← 1 to n
9          do if  $x_i = y_j$ 
10             then  $c[i, j] \leftarrow c[i-1, j-1] + 1$ 
11                  $b[i, j] \leftarrow \text{"↖"}$ 
12             else if  $c[i-1, j] \geq c[i, j-1]$ 
13                 then  $c[i, j] \leftarrow c[i-1, j]$ 
14                      $b[i, j] \leftarrow \text{"↑"}$ 
15                 else  $c[i, j] \leftarrow c[i, j-1]$ 
16                      $b[i, j] \leftarrow \text{"←"}$ 
17  return c and b

```

ภาพที่ 2.15 Algorithm LCS-LENGTH

(ที่มา: <http://staff.ustc.edu.cn/~csl/graduate/algorithms/book6/chap16.htm>)

จากภาพที่ 2.15 Algorithm LCS-LENGTH เป็นการจำกัดค่าตอบและการจำกัดที่มาของคำตอบ ลำดับย่อยร่วม เริ่มด้วยการสร้างตาราง  $c[m, n]$  และตาราง  $b[m, n]$  (บรรทัดที่ 1 และ 2) กำหนดให้  $m$  เป็นตำแหน่งตัวอักษรในลำดับ  $X$  และให้  $n$  เป็นตำแหน่งตัวอักษรในลำดับ  $Y$  โดยการเพิ่มความยาวของลำดับย่อยกรณี  $i = 0$ , และ  $j = 0$  (บรรทัดที่ 3 ถึง 6) ที่แถวบนและแถวซ้ายสุดของตาราง และเริ่มตรวจสอบตัวร่วมของลำดับทั้งสอง (บรรทัดที่ 7 ถึง 8) เมื่อตำแหน่ง  $m$  และ  $n$  เท่ากับ 1 ถ้าข้อมูล  $x_i = y_j$  (บรรทัดที่ 9 ถึง 11) หมายถึงพบตัวร่วมของลำดับย่อย แล้วนำค่าความยาวของ  $c[i-1, j-1]+1$  เก็บไว้ในตำแหน่ง  $c[i, j]$  ของตาราง  $c[m, n]$  และจดจำลูกศร “↖” ซึ่งเป็นที่มาของคำตอบไว้ในตำแหน่ง  $b[i, j]$  ของตาราง  $b[m, n]$  แต่ถ้าข้อมูล  $x_i \neq y_j$  และค่าความยาวของ  $c[i-1, j] \geq c[i, j-1]$  (บรรทัดที่ 12 ถึง 14)ให้นำค่าความยาวของ  $c[i-1, j]$  มาเก็บไว้ในตำแหน่ง  $c[i, j]$  ของตาราง  $c[m, n]$  และจดจำลูกศร “↑” ไว้ในตำแหน่ง  $b[i, j]$  ของตาราง  $b[m, n]$  ในทางกลับกันถ้าหากค่าความยาวของ  $c[i-1, j] < c[i, j-1]$  (บรรทัดที่ 15 ถึง 16)ให้นำค่าความยาวของ  $c[i, j-1]$  มาเก็บไว้ในตำแหน่ง  $c[i, j]$  ของตาราง  $c[m, n]$  และจดจำลูกศร “←” ไว้ในตำแหน่ง  $b[i, j]$  ของตาราง  $b[m, n]$  เมื่อทำการตรวจสอบตัวร่วมของลำดับย่อยครบแล้ว (บรรทัดที่ 17) จึงนำข้อมูลในตาราง  $c[m, n]$  และข้อมูลในตาราง  $b[m, n]$  มาแสดง

4. การสร้างผลเฉลยเหมาะสมที่สุดเป็นการหาคำตอบของลำดับย่อยร่วมยาวสุดจากการจำกัดค่าตอบและที่มาของคำตอบ ซึ่งเป็นการหาคำตอบโดยการย้อนตามทิศลูกศร เริ่มจากตำแหน่งสุดท้ายของตาราง  $b[m, n]$  โดยจะสิ้นสุดการหาลำดับย่อยร่วมยาวสุดและแสดงคำตอบ (บรรทัดที่ 1 และ 2) เมื่อตำแหน่ง  $[i, j]$  มีค่า  $i = 0$  หรือ  $j = 0$  ซึ่งหากพบทิศลูกศร “↖” (บรรทัดที่ 3 ถึง 5) ในตำแหน่ง  $b[i, j]$  หมายถึงตัวร่วมของลำดับย่อยให้แสดงข้อมูลของ  $X_i$  แล้วให้  $b[i-1, j-1]$  แต่ถ้าหากพบทิศลูกศร “↑” (บรรทัดที่ 6 และ 7) ให้  $b[i-1, j]$  และถ้าหากไม่ใช่ทั้ง 2 กรณี (บรรทัดที่ 8 และ 9) ให้เปลี่ยนค่าเป็น  $b[i, j-1]$

```

PRINT-LCS( $b, X, i, j$ )
1  if  $i = 0$  or  $j = 0$  then
2    return
3  if  $b[i, j] = \text{“}\swarrow\text{”}$  then
4    PRINT-LCS( $b, X, i - 1, j - 1$ )
5    print  $x_i$ 
6  else if  $b[i, j] = \text{“}\uparrow\text{”}$  then
7    PRINT-LCS( $b, X, i - 1, j$ )
8  else
9    PRINT-LCS( $b, X, i, j - 1$ )

```

ภาพที่ 2.16 Algorithm PRINT-LCS

ที่มา : (<http://i.stack.imgur.com/eTB4S.png>)

		$j$	0	1	2	3	4	5	6
		$y_j$		<b>B</b>	<i>D</i>	<b>C</b>	<i>A</i>	<b>B</b>	<b>A</b>
$i$	$x_i$		0	0	0	0	0	0	0
0	<i>A</i>		0	↑	↑	↑	↖1	←1	↖1
1	<b>B</b>		0	↖1	←1	←1	↑1	↖2	←2
2	<b>C</b>		0	↑1	↑1	↖2	←2	↑2	↑2
3	<b>B</b>		0	↖1	↑1	↑2	↑2	↖3	←3
4	<i>D</i>		0	↑1	↖2	↑2	↑2	↖3	↑3
5	<b>A</b>		0	↑1	↑2	↑2	↖3	↑3	↖4
6	<b>B</b>		0	↖1	↑2	↑2	↑3	↖4	↑4

ภาพที่ 2.17 ตัวอย่างการหาลำดับย่อยร่วมยาวสุดของลำดับ X และ Y

ที่มา : (<http://staff.ustc.edu.cn/~csl/graduate/algorithms/book6/chap16.htm>)

จากภาพที่ 2.17 ตัวอย่างการหาลำดับย่อยร่วมยาวสุดของลำดับ X และ Y โดยกำหนดให้ลำดับ  $X = (A, B, C, B, D, A, B)$  และลำดับ  $Y = (B, D, C, A, B, A)$  ซึ่งคำตอบของลำดับย่อยร่วมยาวสุด คือ  $(B, C, B, A)$  และมีค่าความยาวของลำดับย่อยร่วมยาวสุดคือ 4

### ป้ายทะเบียนรถยนต์ในประเทศไทย (Thai Vehicle License Plate)

ลักษณะแผ่นป้ายทะเบียนรถยนต์ในประเทศไทยตามกฎหมายกระทรวงของกระทรวงคมนาคม ว่าด้วยการกำหนดขนาด ลักษณะ และสีของแผ่นป้ายทะเบียนรถและการแสดงแผ่นป้ายทะเบียนรถ และเครื่องหมายแสดงการเสียภาษีประจำปี พ.ศ. 2554 กำหนดให้แผ่นป้ายทะเบียนให้มีขนาดกว้าง 15 เซนติเมตร ยาว 34 เซนติเมตร ขอบแผ่นป้ายอัดเป็นรอยคูน ในแผ่นป้ายมีเครื่องหมายเป็นตัวอักษร ขส อยู่ในวงกลมอัดเป็นรอยคูนที่มุมล่างด้านขวาของแผ่นป้ายที่สามารถมองเห็นได้ชัดเจนด้วยตาเปล่าตามขนาดที่อธิบดีกำหนด แผ่นป้ายแบ่งออกเป็นสองบรรทัด บรรทัดที่หนึ่งประกอบด้วยตัวอักษรประจำหมวดตัวที่หนึ่ง ตัวอักษรประจำหมวดตัวที่สอง และหมายเลขทะเบียนไม่เกินสี่หลัก บรรทัดที่สองเป็นตัวอักษรแสดงชื่อกรุงเทพมหานครหรือจังหวัดที่จดทะเบียน เว้นแต่กรณีจดทะเบียนที่อำเภอเบตง จังหวัดยะลา ให้ใช้คำว่า เบตง ทั้งนี้ ตัวอักษรให้ใช้ตัวอักษรไทย และหมายเลขทะเบียนให้ใช้ตัวเลขอารบิก และทั้งตัวเลขและตัวอักษรให้อัดเป็นรอยคูนในกรณีที่ใช้ตัวอักษรประจำหมวดตัวที่หนึ่งและตัวที่สองจนครบทุกตัวอักษรแล้วในบรรทัดที่หนึ่ง ให้เพิ่มตัวเลขอารบิกตั้งแต่ 1 ถึง 9 ตามลำดับ ไว้ด้านบนของตัวอักษรประจำหมวดตัวที่หนึ่ง ตัวเลขและตัวอักษร ให้อัดเป็นรอยคูน ในการนี้ตัวเลขในบรรทัดที่หนึ่งให้สูงไม่น้อยกว่า 5.8 เซนติเมตร และกว้างไม่น้อยกว่า 3.1 เซนติเมตร ตัวอักษรประจำหมวดในบรรทัดที่หนึ่ง ให้สูงไม่น้อยกว่า 5.8 เซนติเมตร และกว้างไม่น้อยกว่า 3.8 เซนติเมตร ตัวอักษรในบรรทัดที่สอง ให้สูงไม่น้อยกว่า 2.1 เซนติเมตร และกว้างไม่น้อยกว่า 1.8 เซนติเมตร ทั้งนี้ เว้นแต่กรณีตัวอักษร ข ง ช ซ ฉ ณ ญ ฎ ฏ ฐ ฒ ฒ ป ฝ ฟ ศ ย ส พ ฮ สระ วรรณยุกต์และหมายเลข 1 ให้อธิบดีกำหนดความสูงและความกว้างของตัวอักษรหรือตัวเลขดังกล่าว



ภาพที่ 2.18 ภาพจำลองแผ่นป้ายทะเบียนรถยนต์

ที่มา : ([http://202.44.34.144/nccitedoc/admin/nccit\\_files/NCCIT-2011220204.pdf](http://202.44.34.144/nccitedoc/admin/nccit_files/NCCIT-2011220204.pdf))

## ชื่อจังหวัดบนป้ายทะเบียนรถยนต์ในประเทศไทย

1. กรุงเทพมหานคร	21. นครราชสีมา	41. เพชรบูรณ์	61. สมุทรสงคราม
2. กระบี่	22. นครศรีธรรมราช	42. ภูเก็ต	62. สมุทรสาคร
3. กาญจนบุรี	23. นครสวรรค์	43. มหาสารคาม	63. สระแก้ว
4. กาลสินธุ์	24. นนทบุรี	44. มุกดาหาร	64. สระบุรี
5. กำแพงเพชร	25. นครราชสีมา	45. แม่ฮ่องสอน	65. สิงห์บุรี
6. ขอนแก่น	26. น่าน	46. ยะลา	66. สุโขทัย
7. จันทบุรี	27. บึงกาฬ	47. ยโสธร	67. สุพรรณบุรี
8. ฉะเชิงเทรา	28. บุรีรัมย์	48. ร้อยเอ็ด	68. สุราษฎร์ธานี
9. ชลบุรี	29. ปทุมธานี	49. ระนอง	69. สุรินทร์
10. ชัยนาท	30. ประจวบคีรีขันธ์	50. ระยอง	70. หนองคาย
11. ชัยภูมิ	31. ปราจีนบุรี	51. ราชบุรี	71. หนองบัวลำภู
12. ชุมพร	32. ปัตตานี	52. ลพบุรี	72. อ่างทอง
13. เชียงราย	33. พระนครศรีอยุธยา	53. ลำปาง	73. อุตรธานี
14. เชียงใหม่	34. พังงา	54. ลำพูน	74. อุทัยธานี
15. ตรัง	35. พัทลุง	55. เลย	75. อุตรดิตถ์
16. ตรัง	36. พิจิตร	56. ศรีสะเกษ	76. อุบลราชธานี
17. ตาก	37. พิษณุโลก	57. สกลนคร	77. อำนาจเจริญ
18. นครนายก	38. เพชรบุรี	58. สงขลา	78. เบตง
19. นครปฐม	39. เพชรบูรณ์	59. สตูล	
20. นครพนม	40. แพร่	60. สมุทรปราการ	

### งานวิจัยที่เกี่ยวข้อง

โปรแกรมประยุกต์รู้จำป้ายทะเบียนรถยนต์ภาษาไทยสำหรับอุปกรณ์เคลื่อนที่ จากการศึกษางานวิจัยของนายพิรวัส กลิ่นเฟื่อง และนายอิสระพงศ์ จิโรจน์วงศ์ สามารถสรุปงานวิจัยรวมถึงขั้นตอนและวิธีการหลัก ดังนี้

งานวิจัยนี้เป็นการพัฒนาโปรแกรมประยุกต์รู้จำป้ายทะเบียนรถยนต์ภาษาไทยสำหรับอุปกรณ์เคลื่อนที่ระบบปฏิบัติการ Android งานวิจัยนี้เป็นระบบที่ใช้สำหรับวิเคราะห์ภาพถ่ายของป้ายทะเบียนรถยนต์และแปลงภาพของป้ายทะเบียนเหล่านั้นออกมาเป็นตัวอักขระสามารถนำไปใช้ร่วมกับงานอื่นได้ งานวิจัยนี้ประกอบด้วยการทำงาน 3 ส่วน คือ 1. การประมวลผลตอนต้น ใช้สำหรับการเตรียมรูปป้ายทะเบียนรถยนต์โดยการแยกป้ายทะเบียนออกเป็นส่วนของเลขทะเบียนและส่วนของจังหวัด แล้วทำเทรซโฮลด์เพื่อลดปัญหาด้านแสงเงาและสิ่งรบกวน 2. ส่วนการรู้จำป้ายทะเบียนรถยนต์ จะเป็นการสร้างไฟล์ Box จากโปรแกรม QT Box Editor ที่เก็บลักษณะเฉพาะของตัวอักษรโดยระบุตำแหน่งของตัวอักขระแต่ละตัว เพื่อใช้ในการรู้จำอักขระบนป้ายทะเบียน โดยใช้ Tesseract-OCR ฝึกฝนให้ระบบรู้จำตัวอักขระ 3. ทำการวิเคราะห์ลักษณะของตัวอักษรและแสดงผลให้ผู้ใช้งานทราบ

จากการศึกษางานวิจัยข้างต้นนี้ ผลการทดสอบความแม่นยำในการรู้จำป้ายทะเบียนภาษาไทย โดยอาศัยตัวอย่างป้ายทะเบียนจำนวน 86 ป้าย มีความถูกต้องเฉลี่ย 60.46% โดยมีความผิดพลาดของอักขระ 39.54% ซึ่งเกิดจากปัญหาด้านแสงเงา ปัญหาจากสิ่งแปลแป้น และปัญหาจากชนิดของป้ายทะเบียน