บทที่ 3

วิธีดำเนินงานวิจัย

บทนี้จะกล่าวถึงขั้นตอนในการดำเนินงานของโครงงาน ซึ่งจะทำให้ทราบถึงการวิเคราะห์และการออกแบบระบบโดยละเอียด ว่ามีแนวทางในการทำงานหรือขั้นตอนในการทำงานของระบบอย่างไร โดยขั้นตอนในการดำเนินงานมีรายละเอียดดังนี้

3.1 กรอบการดำเนินงาน/ขั้นตอนการดำเนินงาน

3.2 การทำงานส่วนของการสร้าง Model การจำแนกกล่องนม

3.3 ขั้นตอนของการเรียนรู้และการทดสอบ

3.4 ขั้นตอนการเรียกใช้งาน Tesseract OCR

3.5 โมเดลการจำแนกยี่ห้อกล่องนม

3.6 การประเมินประสิทธิภาพ

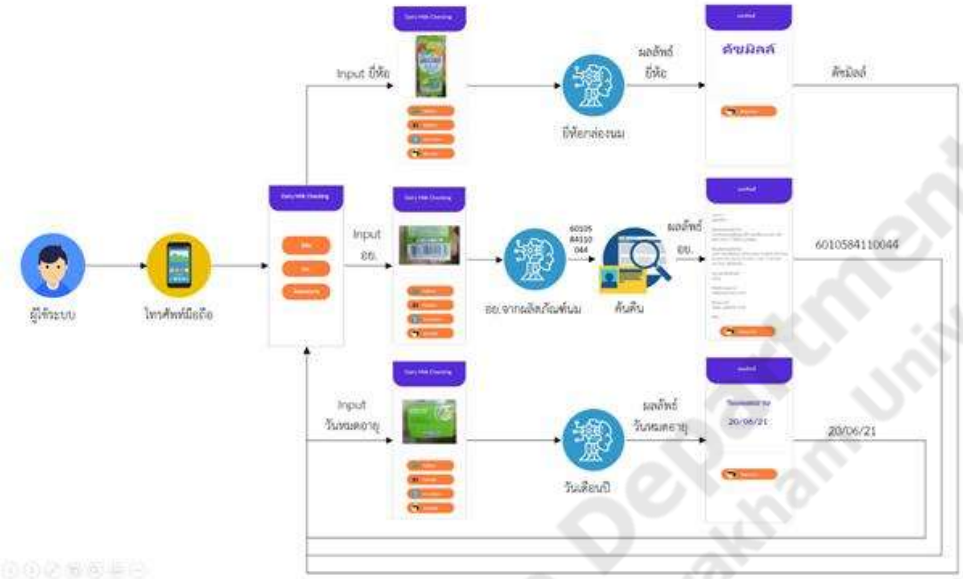
3.7 ชุดข้อมูลที่ใช้ในการทดลอง

3.8 Data Flow Diagram

- 3.8.1 Context Diagram
- 3.8.2 Data Flow Diagram Level 1
- 3.8.3 External Entity Description
- 3.8.4 Data Store Description
- 3.8.5 Data Structure Description

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.1 กรอบการดำเนินงาน



ภาพประกอบที่ 3.1 ขั้นตอนการทำงานของระบบ

จากภาพประกอบที่ 3.1 ผู้ใช้เข้าใช้งานผ่านแอพพลิเคชั่น ผู้ใช้เลือกได้ว่าจะจำแนกยี่ห้อ รู้จำวันเดือนปี หรือค้นคืน อย. ให้ผู้ใช้คลิกที่ปุ่มที่ต้องการ โดยหลังจากเลือกเสร็จ จะให้ผู้ใช้นำภาพเข้า การนำภาพเข้าสามารถนำเข้าได้ 2 แบบ 1. จากแกลลอรี่ 2. ถ่ายภาพ

โดยถ้าเลือกปุ่มที่ 1 ยี่ห้อผู้ใช้ต้องเลือกรูปจากแกลลอลี่หรือถ่ายภาพด้านหน้าของกล่องนม

ปุ่มที่ 2 อย. ผู้ใช้ต้องเลือกรูปจากแกลลอลี่หรือถ่ายภาพด้านข้างที่มีเลข อย.

ปุ่มที่ 3 วันหมดอายุ ผู้ใช้ต้องเลือกรูปจากแกลลอลี่หรือถ่ายภาพด้านบนที่มีวันหมดอายุ

โดยแต่ละรูปภาพจะเข้า Model ของแต่ละแบบเพื่อทำนายผล

1. รูปภาพโลโก้ เข้า Model จำแนกยี่ห้อเพื่อจำแนกยี่ห้อกล่องนม

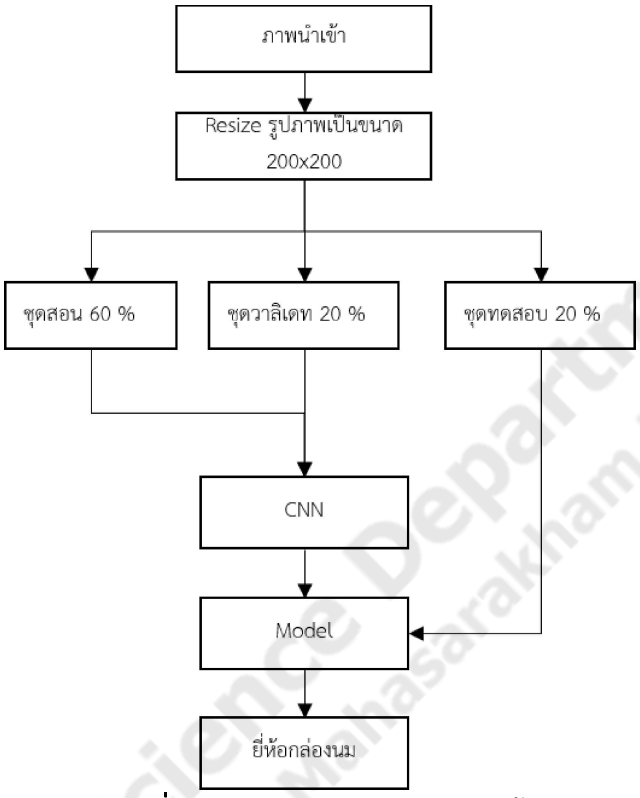
2.รูปภาพเลข อย. เข้า Model ค้นคืน อย. เพื่อค้นคืน อย.จากกล่องนม

3. รูปภาพวันเดือนปีหมดอายุเข้า Model รู้จำวันเดือนปีหมดอายุเพื่อรู้จำวันเดือนปีหมดอายุของกล่องนม

หลักจากเสร็จขั้นตอนการทำนายผล ระบบจะส่งผลคำตอบเป็นตัวหนังสือ ตัวเลขและอักษรพิเศษ ออกทางจอมือถือของผู้ใช้งานถ้าผู้ใช้เลือกยี่ห้อ คำตอบที่ออกจะมียี่ห้อของกล่องนมที่ผู้ใช้เลือกหรือถ่ายเข้าไป ถ้าผู้ใช้เลือก อย. คำตอบจะออกเป็น ประเภทอาหาร ชื่อผลิตภัณฑ์ (TH) ชื่อผลิตภัณฑ์ (EN) สถานะผลิตภัณฑ์ ชื่อผู้รับอนุญาต ชื่อสถานที่ และที่ตั้งถ้าผู้ใช้เลือก วันหมดอายุ คำตอบจะออกเป็นวันหมดอายุของกล่องนมสิ้นสุดขั้นตอน

Computer Science Department
Faculty of Informatics Mahasarakham University

3.2 การทำงานส่วนของการสร้าง Model การจำแนกกล่องนม

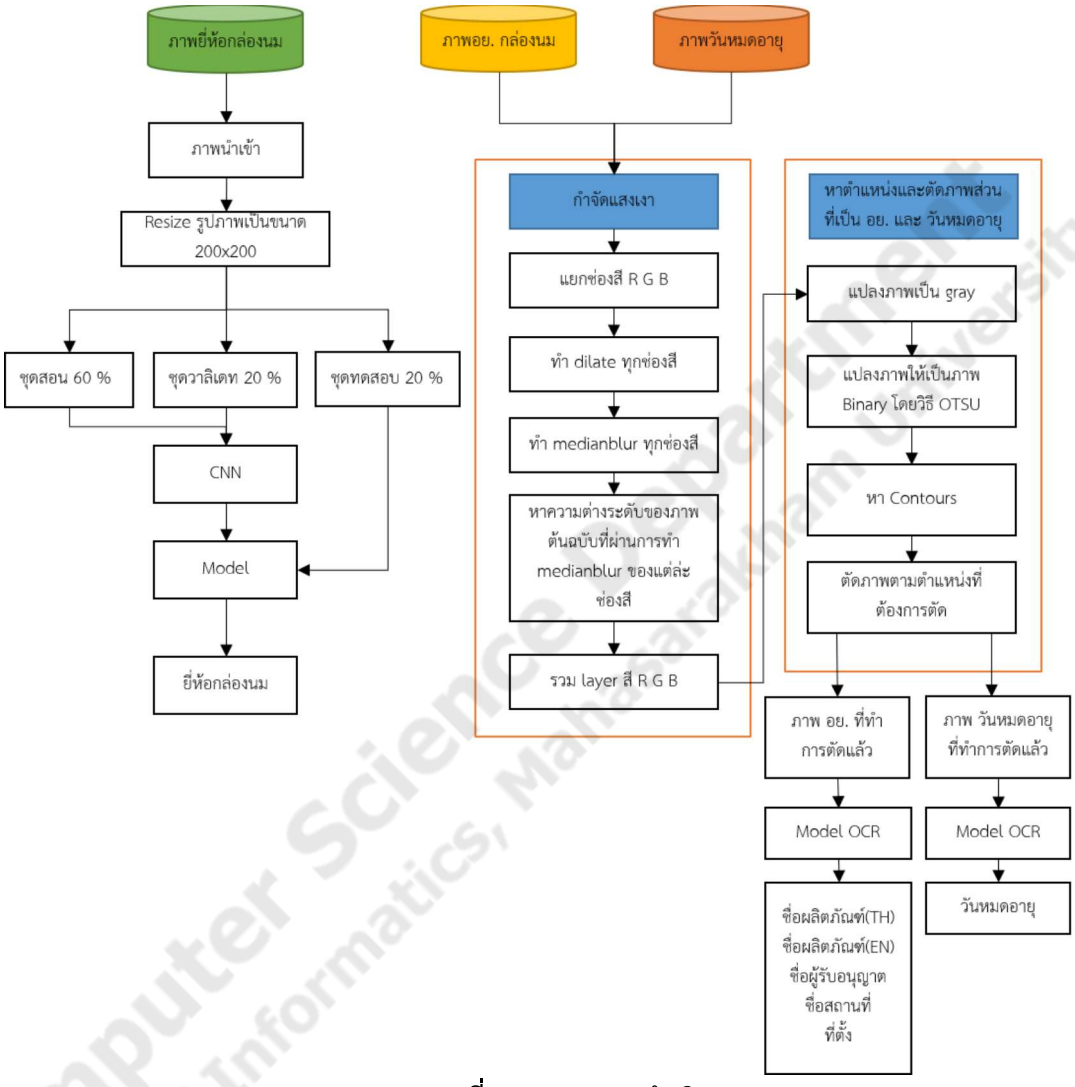


ภาพประกอบที่ 3.2 แผนผังการทำงานของการสร้าง Model

ภาพที่นำเข้ามาแบ่งเป็นรูปภาพโลโก้ รูปภาพอย. รูปภาพวันเดือนปีหมดอายุ จะเข้าสู่กระบวนการทำ Pre-Processing เพื่อเตรียมข้อมูลก่อนเข้าสู่ Model และนำข้อมูลที่ผ่านกระบวณการเข้าสู่ Model โดยประกอบด้วย CNN แล้วจึงจะได้เป็น ผลลัพธ์

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.3 ขั้นตอนของการเรียนรู้และการทดสอบ



ภาพประกอบที่ 3.3 กรอบการดำเนินงาน

Computer Science Department
Faculty of Informatics, Mahasarakham University

โดยมีการกำหนดขั้นตอนการดำเนินงานออกเป็นขั้นตอนดังนี้

3.3.1 ภาพนำเข้า ภาพนำเข้าเป็นไฟล์ที่ได้มาจากการถ่ายรูปหรือจากแกลลอรี่ เป็นภาพสี RGB

ภาพประกอบที่ 3.4 ตัวอย่างภาพนำเข้า

3.3.2 การกำจัดแสงเงา เป็นขั้นตอนในการกำจัดแสง หรือเงาที่ตกกระทบจากการถ่ายภาพ ซึ่งอาจจะทำให้เกิดปัญหาในการประมวลผล ดังนั้นจึงจะทำการกำจัดสิ่งรบกวนเหล่านี้ออก ก่อนนำไปประมวลผล โดยเริ่มจากการแยกช่องสีภาพ R G B แล้วทำ Dilation ของทุกช่องสีแล้วจึง นำมาทำเบลอโดยการใช้ Median Blur เพื่อให้ภาพมีความ Smooth แล้วนำมาหาความต่างระหว่างภาพต้นฉบับ กับภาพที่ทำ Blur ของแต่ละช่องสี แล้วจึงนำภาพที่เป็นความต่างของสี merge เข้าด้วยกันให้เป็นภาพสี RGB เหมือนเดิม และเพื่อให้เข้าใจการทำงานในแต่ละขั้นตอนให้มากขึ้น ซึ่งมีวิธีการคำนวณ ดังต่อไปนี้

-แยกช่องสี R G B ภาพสี RGB นั้น จะประกอบไปด้วยค่าสีเดียวของรูปภาพ 3 สี รวมกัน คือ สีแดง สีเขียว และสีน้ำเงินตามลำดับ ถูกเก็บให้อยู่ในรูปแบบของอาร์เรย์ 3 มิติ ซึ่งในการประมวลผลในขั้นถัดๆ ไปนั้น จำเป็นจะต้องแยกช่องสีทั้งสามนั้นออกให้เป็นรูปภาพ 3 รูป ที่ประกอบไปด้วยค่าสีเดียว แล้วจึงนำมาใช้ในการคำนวณ

ภาพประกอบที่ 3.5 ลักษณะของภาพสี R G B

- ทำ Dilation เป็นการขยายส่วนของวัตถุใน ภาพโดยเป็นการดำเนินงานระหว่างภาพที่อยู่ในรูปแบบ Binary และตัวดำเนินการซึ่งเป็นแบบ 4 ทิศทาง

0	1	0
1	1	1
0	1	0

ภาพประกอบที่ 3.6 ตัวดำเนินการ 4 ทิศทาง

วิธีดำเนินการ คือ นำตัวดำเนินการสแกนไปยังทุกข้อมูลของภาพตามลำดับไปทั่วทั้งภาพ โดยเมื่อพบว่าตำแหน่งกึ่งกลางของตัวดำเนินการและตำแหน่งของรูปภาพต้นฉบับที่ตรงกันมีค่า เท่ากับ 1 จะทำการยูเนียน (Union) ระหว่างภาพต้นฉบับและตัวดำเนินการ ดังสมการ

Computer Science Department
Faculty of Information Technology
Rangsit University

$$I \oplus H = \bigcup_{p \in I} H_p \quad (3.1)$$

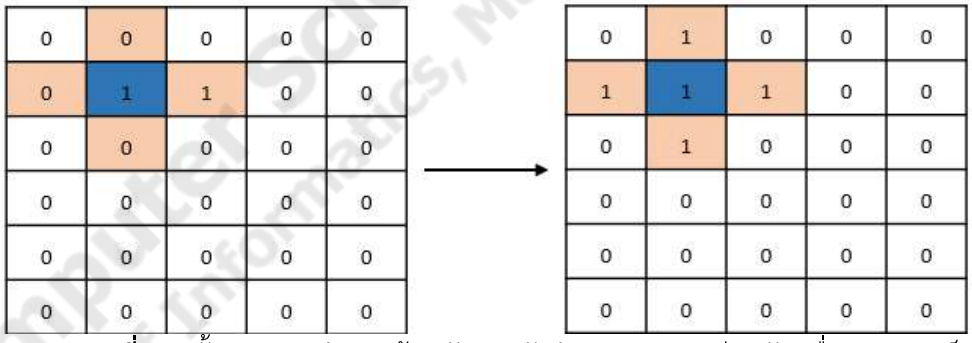
โดยที่ I คือภาพต้นฉบับ

H คือ ตัวดำเนินการ 4 ทิศทาง

ตัวอย่าง การขยายวัตถุในภาพ

0	0	0	0	0
0	1	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

ภาพประกอบที่ 3.7 ภาพต้นฉบับก่อนทำ Dilation

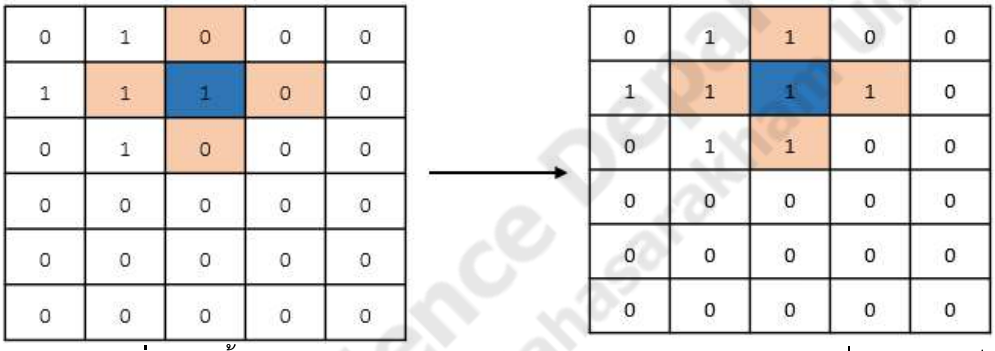


ภาพประกอบที่ 3.8 ขั้นตอนการนำภาพต้นฉบับและตัวดำเนินการมายูเนียนกัน เมื่อตรงกลางเป็นเลขเดียวกัน

Computer Science Department
Faculty of Informatics, Mahasarakham University

0	1	0	0	0
1	1	1	0	0
0	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

ภาพประกอบที่ 3.9 ภาพหลังจากการยูเนียนกันของภาพประกอบที่ 36



ภาพประกอบที่ 3.10 ขั้นตอนการนำภาพต้นฉบับและตัวดำเนินการมายูเนียนกัน เมื่อตรงกลางเป็นเลขเดียวกัน

0	1	1	0	0
1	1	1	1	0
0	1	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

ภาพประกอบที่ 3.11 ภาพหลังจากการยูเนียนกันของภาพประกอบที่ 38

Computer Science Department
Faculty of Informatics, Mahasarakham University

(ก) ภาพต้นฉบับ

(ข) ภาพผลลัพธ์การทำ Dilation

ภาพประกอบที่ 3.12 ภาพต้นฉบับและภาพผลลัพธ์จากการทำ Dilation

- การ Blur ภาพ โดยใช้ตัวกรอง Median Blur เพื่อกำจัดสัญญาณรบกวน วิธีการ ตัวอย่าง การกรองภาพขนาดภาพ 3 x 3 นำค่าในแต่ละช่องมาเรียงค่าจากน้อยไป หามากจากนั้นหาค่ากลาง

Computer Science Department
Faculty of Informatics Mahasarakham University

(ภาพประกอบที่ 3.13) แล้วนำค่ากลางไปแทนที่พิกเซล ตรงกลาง (ภาพประกอบ 3.14) จะทำแบบนี้ขยับไปเรื่อย ๆ จนครบทุกพิกเซล

20	15	3
6	88	99
199	15	4

ภาพประกอบที่ 3.13 ตัวอย่างภาพก่อนทำ Median

3	4	6	15	15	20	88	99	199

ภาพประกอบที่ 3.14 ตัวอย่างเรียงพิกเซลภาพจากน้อยไปหามาก

20	15	3
6	15	99
199	15	4

ภาพประกอบที่ 3.15 ตัวอย่างภาพหลังทำ Median

(ก) ภาพต้นฉบับ

ภาพประกอบที่ 3.16 ตัวอย่างภาพก่อนและหลังทำ Median Blur

Computer Science Department
Faculty of Information Management
Mahasarakham University

(ข) ภาพผลลัพธ์

ภาพประกอบที่ 3.16 ตัวอย่างภาพก่อนและหลังทำ Median Blur (ต่อ)

- การคำนวณหาความแตกต่างของภาพ เป็นการคำนวณเพื่อหาความแตกต่างขององค์ประกอบภายในภาพระหว่างภาพต้นฉบับ กับภาพที่ทำการกำจัดแสงเงา ดังสมการ

$$dst(I) = saturate(|src1(I) - src2(I)|) \quad (3.2)$$

โดย dst คือ ผลลัพธ์จะอยู่ในรูปแบบ array

I คือ ภาพ โดยภาพจะอยู่ในรูปแบบ array

Src1 คือ ภาพแรกจะอยู่ในรูปแบบ array

Src2 คือ ภาพที่สองจะอยู่ในรูปแบบ array

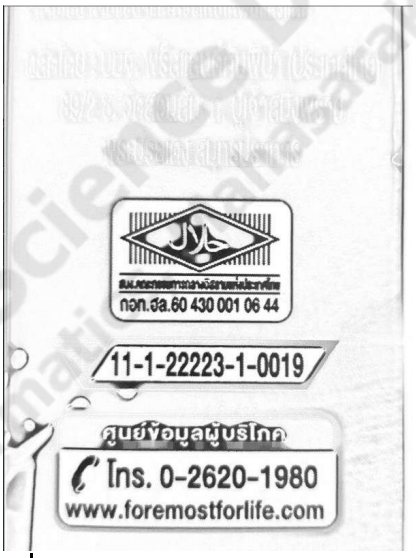
ตัวอย่างการคำนวณ ภาพขนาด 3x3 pixels

$$dst(I) = saturate(\left|\begin{bmatrix}574\\546\\489\end{bmatrix} - \begin{bmatrix}124\\124\\111\end{bmatrix}\right|) = \begin{bmatrix}430\\422\\378\end{bmatrix}$$

ภาพประกอบที่ 3.17 ตัวอย่างการคำนวณภาพขนาด 3x3 pixels

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.18 ตัวอย่างภาพหลังการหาความแตกต่าง



ภาพประกอบที่ 3.19 ตัวอย่างภาพหลังกลับสีของวัตถุกับพื้นหลัง

- การทำ Normalization การทำ Normalization ให้กับภาพเป็นการปรับปรุงเพื่อลดความซับซ้อนของข้อมูลโดยในจะทำการ Normalization ให้ค่าในภาพอยู่ระหว่าง 0 – 1 โดยหารด้วย 255 ตลอดทั้งภาพ

Computer Science Department
Faculty of Information Technology
Rajabhat Mahasarakham University

199	99	242	0	61
102	45	0	255	71
177	5	1	194	254
77	13	10	100	22
255	180	1	102	99

/255 =

0.78	0.38	0.94	0	0.23
0.4	0.17	0	1	0.27
0.69	0.01	0.00	0.76	0.99
0.30	0.05	0.03	0.39	0.08
1	0.70	0.00	0.4	0.38

ภาพประกอบที่ 3.20 ตัวอย่างตารางการทำ Normalization

- แปลงภาพสีเป็นภาพระดับเทา ขั้นตอนการแปลงภาพ RGB ไปเป็น Gray imageสามารถทำได้โดยอาศัยค่าของ R,G,B ที่อยู่ในแต่ละพิกเซลของรูปภาพต้นฉบับมาใช้ประมวลผลแปลงเป็น Gray image ซึ่งค่าของ R,G,B มีดังตารางต่อไปนี้

ภาพ RGB (1 pixel มีค่า 0-255 อยู่สามค่าคือ r,g,b ตามลำดับ)

	1	2	3	4	5
1					
2					
3					
4					
5					

	1	2	3	4	5
1					
2			0		
3		255	0	0	
4			0		
5					

ตารางค่าสี R

	1	2	3	4	5
1					
2			0		
3		0	0	250	
4			255		
5					

ตารางค่าสี G

	1	2	3	4	5
1					
2			0		
3		0	0	250	
4			255		
5					

ตารางค่าสี B

ภาพประกอบที่ 3.21 ตารางค่าสี RGB

ถ้านำพิกเซลตำแหน่งที่ (3,2) จากตารางมาแปลงเป็น Grayscale image จะสามารถทำได้ดังนี้

จากสูตร Gray=(0.299)R+(0.587)G+(0.114)B

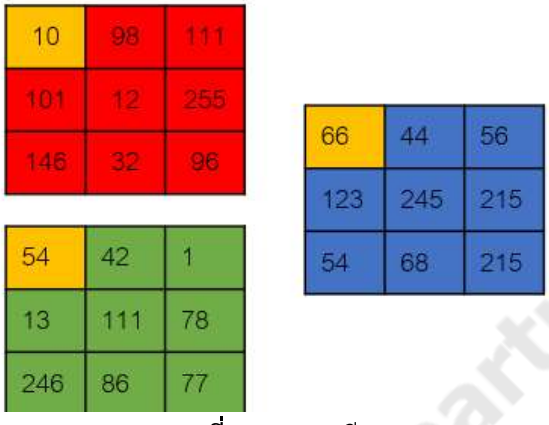
แทนค่า Gray=(0.299)255+(0.587)0+(0.114)0

จะได้ Gray=76.245+0+0

Gray เท่ากับ 76.245 จะถูกใช้แทนลงในตำแหน่ง (3,2) ของภาพ Grayscale image ซึ่งค่าพิกเซลในตำแหน่งอื่นๆจะใช้วิธีในการคำนวณที่เหมือนกัน โดยค่า R,G,B ของแต่ละตำแหน่งในรูปภาพ RGB แล้วจะได้ผลลัพธ์จากการแปลงภาพสีเป็นภาพระดับเทาดังภาพที่ 3.2

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตัวอย่างการแปลงภาพสี RGB ไปเป็น Gray Scale ของภาพที่มีขนาด 3 x 3 พิกเซล

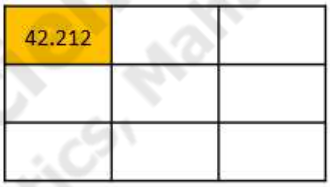


ภาพประกอบที่ 3.22 ภาพสี RGB (1,1)

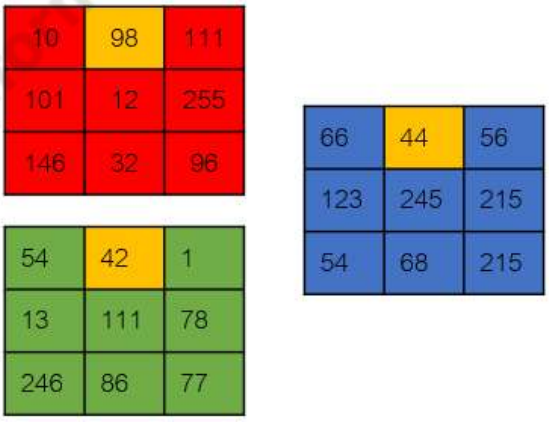
จากรูปภาพตารางมาแปลงเป็น Grayscale image จะสามารถทำได้ดังนี้

(0.299)10+(0.587)54+(0.114)66

2.99+31.698+7.524=42.212



ภาพประกอบที่ 3.23 ภาพหลังจากแปลงเป็นภาพระดับเทา1



ภาพประกอบที่ 3.24 ภาพสี RGB (1,2)

จากรูปภาพตารางมาแปลงเป็น Grayscale image จะสามารถทำได้ดังนี้

(0.299)98+(0.587)42+(0.114)44

Computer Science Department
Faculty of Informatics, Mahasarakham University

29.302+24.654+5.016=58.972

42.212	58.972	

ภาพประกอบที่ 3.25 ภาพหลังจากแปลงเป็นภาพระดับเทา2

ภาพประกอบที่ 3.26 ตัวอย่างภาพต้นฉบับที่ผ่านการลดแสงเงา

ภาพประกอบที่ 3.27 ตัวอย่างภาพพื้นหลังการแปลงภาพระดับเทา

- **แปลงภาพระดับเทาเป็นภาพขาวดำ** เป็นการแปลงค่าสีจากระดับเทาที่มีค่าอยู่ระหว่าง 0-255 ให้เป็นภาพแบบไบนารี (Binary) ที่มีค่า 0 และ 255 โดยที่ 0 เป็น ค่าสีดำ และ 255 เป็นค่าสีขาว การแปลงภาพระดับสีเทาให้ เป็นภาพขาว-ดำ จะต้อง กำหนดค่าความเข้มที่ต้องการอ้างอิงหรือ ค่าขีด

Computer Science Department
Faculty of Informatics, Mahasarakham University

แบ่ง (Threshold Value) โดยผู้ ใช้สามารถกำหนด ได้เองหรือใช้อัลกอริทึมในการหาค่า ซึ่งแสดงวิธีการคำนวณได้ ดังนี้

กำหนดให้ L มีค่าเท่ากับ 4

$$L = 4$$

และ n เท่ากับ 25, 28, 115, 21

$$n = \{25, 28, 15, 21\}$$

จากสูตร $P_i = \frac{n_j}{N} \quad , P_i \geq 0 \quad , \sum_{i=1}^{L} P_i = 1$ (3.3)

แทนค่าจะได้

$$P_i = \{0.280, 0.314, 0.168, 0.235\}$$

จากสูตร $\omega_0 = \sum_{i=1}^{K} P_i = \omega(k)$ (3.4)

โดยที่ ω_0 คือความน่าจะเป็นของกลุ่มที่ 1

$$k = 1, \qquad P_1 = \{0.280,\ 0.314,\ 0.168,\ 0.235\}$$

$$\omega_0 = \sum_{i=1}^{K} P_i = 0.280$$

จากสูตร $\omega_1 = \sum_{i=k+1}^{L} P_i$ (3.5)

โดยที่ ω_1คือความน่าจะเป็นของกลุ่มที่ 2

$$k = 1, \qquad P = \{0.280,\ 0.314,\ 0.168,\ 0.235\}$$

$$\omega_1 = \sum_{i=k+1}^{L} P_i = 0.717$$

คำนวณหาค่า $\omega_0,\ \omega_1$

จะได้ $\omega_0 = \{0.280, 0.594, 0.762\}$

$\omega_1 = \{0.717, 1.031, 1.199\}$

จากสูตร $\mu_0 = \frac{\mu(k)}{\omega(k)}$ (3.6)

Computer Science Department
Faculty of Informatics Mahasarakham University

โดยที่ $\mu_k = \sum_{i=1}^{K} ip_i$

$$\mu_k = \{0.280, 0.628, 0.504\}$$

$$\omega_0 = \{0.280, 0.594, 0.762\}$$

ดังนั้น $\mu_0 = \frac{\mu(k)}{\omega(k)} = \{1, 1.0572, 0.661\}$

จากสูตร $\mu_1 = \frac{\sum_{i=K+1}^{L} ip_i}{\omega_1}$ (3.7)

โดยที่ $\omega_1 = \{0.717, 1.031, 1.199\}$

ดังนั้น $\mu_1 = \frac{\sum_{i=K+1}^{L} ip_i}{\omega_1} = \{0.875, 1.577, 2.888\}$

คำนวณหาค่า $\mu_0,\ \mu_1$

จะได้ $\mu_0 = \{1, 1.0572, 0.661\}$

$\mu_1 = \{0.875, 1.577, 2.888\}$

จากสูตร $\sigma_B^2 = \omega_0\omega_1(\mu_0 - \mu_1)^2$ (3.8)

โดยที่ $\omega_0 = \{0.280, 0.594, 0.762\}$

$\omega_1 = \{0.717, 1.031, 1.199\}$

$\mu_0 = \{1, 1.0572, 0.661\}$

$\mu_1 = \{0.875, 1.577, 2.888\}$

จะได้ $\sigma_B^2 = \{0.0031, 0.1653, 4.528\}$

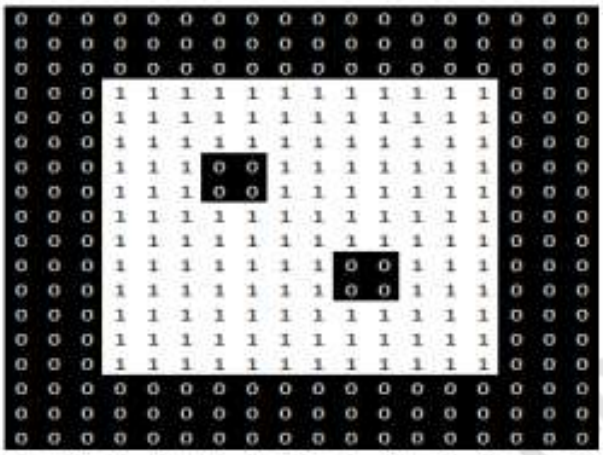
จากสูตร $\sigma_B^2(k^*) = \max_{1 \le k < L} \sigma_B^2(k)$ (3.9)

จะได้ $\sigma_B^2 = \{0.0031, 0.1653, 4.528\}$

เลือก k ที่ให้ค่าความแปรปรวนสูงสุดคือ 3 ที่เท่ากับ 4.528 กำหนดให้เป็นค่า threshold

จะได้ $T = 3$

Computer Science Department
Faculty of Informatics Mahasarakham University



ภาพประกอบที่ 3.28 ภาพขาว-ดำ

ตัวอย่างแสดงภาพแปลงภาพระดับสีเทาให้เป็นขาว-ดำ

ภาพประกอบที่ 3.29 ภาพผลลัพธ์จากการทำ Grayscale

Computer Science Department
Faculty of Informatics Mahasarakham University

ภาพประกอบที่ 3.30 ตัวอย่างภาพหลังการแปลงเป็นภาพระดับเทา เป็นภาพ Binary

- การ contour

Definitions

กำหนด P เป็นตำแหน่งของขอบภาพจุดนั้นๆ

กำหนดให้ P1 เป็นจุดที่อยู่ทางซ้ายของ P2

กำหนดให้ P2 เป็นจุดที่อยู่ข้างหน้า

กำหนดให้ P3 เป็นจุดที่อยู่ทางขวาของ P2

กำหนดให้ B เป็น vector ที่ว่าง

เงื่อนไขที่ 1 if pixel P1 เป็นสีดำ

ให้นำ P1 เข้า B

แล้วให้ P=P1

เงื่อนไขที่ 2 else if pixel P2 เป็นสีดำ

ให้นำ P2 เข้า B

แล้วให้ P=P2

เงื่อนไขที่ 3 else if pixel P3 เป็นสีดำ

ให้นำ P3 เข้า B

แล้วให้ P=P3

เงื่อนไขที่ 4 else if เมื่อหมุนครบ 3 ทิศ

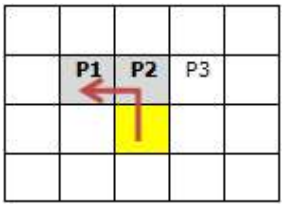
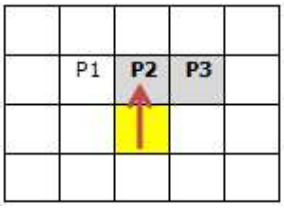
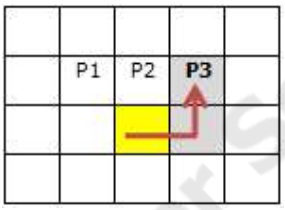
เมื่อหมุนครบ 3 ทิศ แล้วหากไม่พบจุดสีดำให้ จบการทำงาน

เงื่อนไขที่ 5 else

หมุน 90 องศา

Computer Science Department
Faculty of Informatics Mahasarakham University

ตารางที่ 3.1 การหา contour

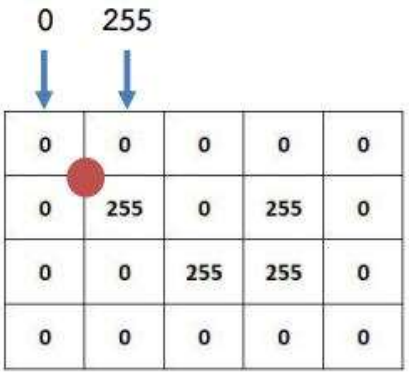
รูปภาพ	วิธีการ
	ขั้นตอนแรกให้ทำการตรวจสอบพิกเซล P1 หาก P1 เป็นสีดำให้ทำการประกาศให้ P1 เป็นพิกเซล จุดเริ่มต้นปัจจุบัน และทำการเคลื่อนที่ไปข้างหน้า โดยย้าย ขั้นตอนที่หนึ่งในปัจจุบัน เคลื่อนที่ไปทางซ้ายไปจอดที่จุด P1
	ถ้า P1 เป็นสีขาว ให้ดำเนินการตรวจสอบ P2 ถ้า P2 เป็นสีดำให้ประกาศ P2 เป็นพิกเซลจุดเริ่มต้นปัจจุบัน และทำการเคลื่อนที่ไปข้างหน้าแล้วหยุดที่ P2
	ในกรณีที่ P1 และ P2 เป็นสีขาว ให้ดำเนินการตรวจสอบ P3 ถ้า P3 เป็นสีดำให้ประกาศ P3 เป็นพิกเซลเริ่มต้นปัจจุบัน และทำการเคลื่อนที่ไปข้างหน้า โดยย้ายขั้นตอนที่หนึ่งใน ปัจจุบัน เคลื่อนที่ไปทางขวาไปจอดที่จุด P1

- การหาพิกัดที่ต้องการตัด ในขั้นตอนนี้จะเป็นการหาจุดเริ่มต้น และจุดสิ้นสุดที่จะใช้ในการระบุตำแหน่งที่จะเอามาตัด crop รูปภาพให้เหลือเฉพาะส่วนที่มีข้อความที่ต้องการ

วิธีการดำเนินการ

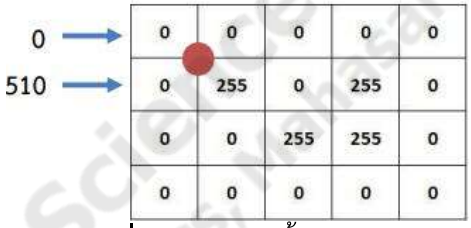
ขั้นตอนที่ 1 หาจุดเริ่มต้นในแนวแกน x ทำการหาผลรวมของภาพโดยเริ่มจากบนลงล่าง (แนวแกน y) และเริ่มหาจากแนวแกน x ที่ x = 0 ถ้าผลรวมเป็น 0 แสดงว่าไม่ใช่จุดที่มีเส้นเค้าโครงอยู่ ดังนั้นจะไปหาผลรวมของแกน x ต่อไป คือ x = 1 ทำไปเรื่อย ๆ จนกระทั่งพบแกนที่มีค่าผลรวมมากกว่า 0 แสดงว่าเป็นจุดที่มีเส้นเค้าโครงอยู่ ก็จะได้ค่าจุดเริ่มต้นในแนวแกน x

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.31 ภาพขั้นตอนในการหาพิกัด x เริ่มต้น

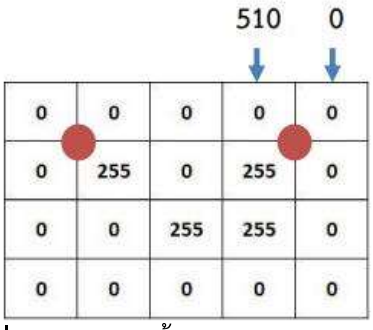
ขั้นตอนที่ 2 หาจุดเริ่มต้นในแนวแกน y เช่นเดียวกันกับการหาในแนวแกน x แต่ให้สลับแนวในการหาในแกน y นั้นจะหาผลรวมโดยเริ่มจากซ้ายไปขวา (แนวแกน x) และเริ่มต้นที่ y = 0 ค่าจุดเริ่มต้นแกน y ที่จะได้ก็คือจุดที่ค่าผลรวมมากกว่า 0



ภาพประกอบที่ 3.32 ภาพในขั้นตอนการหาพิกัด y เริ่มต้น

ขั้นตอนที่ 3 หาความกว้างของภาพตรงส่วนที่มีเส้นเค้าโครงจากการหาขั้นตอนที่ 1 และขั้นตอนที่ 2

- ให้ทำการสแกนหาผลรวมของภาพในแนวแกน y โดยเริ่มจาก x ที่หาได้จากขั้นตอนที่ 1 ถ้าผลรวมมากกว่า 0 แสดงว่ายังมีเส้นเค้าโครง ดังนั้นให้เลื่อน x ออกไป จนกระทั่งพบจุดที่ผลรวมมีค่าเท่ากับ 0 ก็จะได้ความกว้างของภาพ

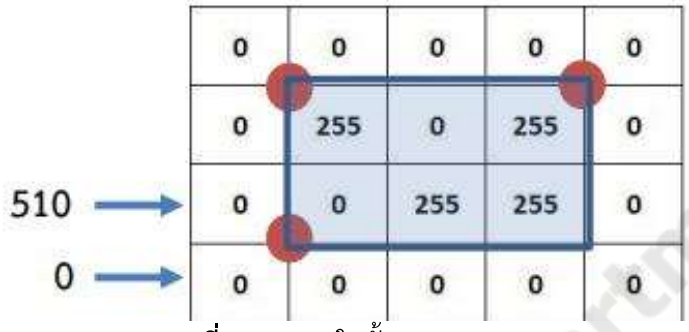


ภาพประกอบที่ 3.33 ภาพในขั้นตอนการหาความกว้างของภาพ

Computer Science Department
Faculty of Informatics, Mahasarakham University

ขั้นตอนที่ 4 หาความสูงของภาพตรงส่วนที่มีเส้นเค้าโครง

- ทำเหมือนขั้นตอนที่ 3 แต่สลับแกน



ภาพประกอบที่ 3.34 ภาพในขั้นตอนการหาความสูงของภาพ

ขั้นตอนที่ 5 นำค่าที่ได้จากขั้นตอนที่ 1-4 มาใช้ในการตัดภาพต้นฉบับ

ภาพประกอบที่ 3.35 ตัวอย่างภาพต้นฉบับก่อนทำการตัดภาพ

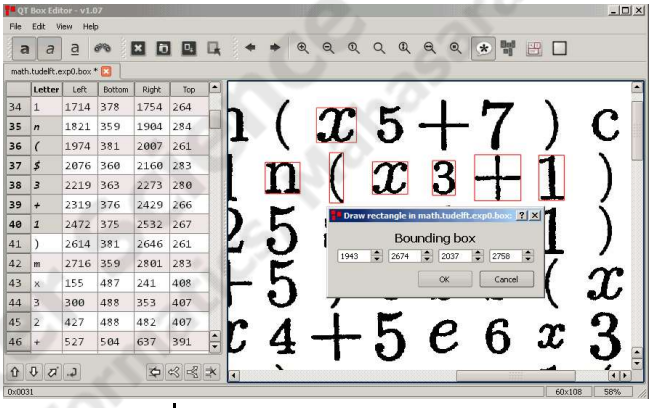
ภาพประกอบที่ 3.36 ตัวอย่างภาพที่ผ่านการตัดให้เหลือเฉพาะตรงส่วนที่ต้องการ

3.4 ขั้นตอนการเรียกใช้งาน Tesseract OCR

สร้าง Traindata สำหรับ OCR ด้วย Tesseract

ในกระบวนการ Optical Character Recognition (OCR) จำเป็นจะต้องมีข้อมูลลักษณะ Feature ของตัวอักขระก่อน เพื่อนำมาประมวลผลเทียบเคียงกับข้อมูลที่ได้จากภาพ ข้อมูล Feature ได้มาจากการ Train ซึ่งจะซับซ้อนในการพัฒนา ดังนั้นในบทความนี้จะใช้ Tool ที่ช่วยลดความซับซ้อนลง โดยมีชื่อว่า Tesseract OCR ใช้สำหรับการรู้จำอักขระทางภาพ พัฒนาขึ้นโดยบริษัท HP ระหว่างปี 1984 -1985 โดยเริ่มต้นมาจากโปรเจควิจัยระดับปริญญาเอกในห้องปฏิบัติการณ์ HP โดยมีความตั้งใจเพื่อนำไปใช้กับงานเครื่องสแกนเป็นหลัก ซึ่งในปี 2005 HP ก็ได้ปล่อยให้เป็น Open Source โดยมี google เป็นผู้สนับสนุน ซึ่งนักพัฒนาสามารถนำชุดคำสั่งนี้มาใช้งานได้ Tesseract นั้นถือว่าเป็นหนึ่งใน OCR Open Source

Step1: สร้างไฟล์ box ซึ่งเป็นไฟล์ที่ใช้ระบุตำแหน่งของอักขระต่างๆ ที่อยู่ในรูปภาพที่จะนำมาใช้ฝึกฝน โดยตำแหน่งของอักขระจะถูกระบุเป็นพิกัด



ภาพประกอบที่ 3.37 หน้าตาโปรแกรมของ Qt-Box Editor

ที่มา http://zdenop.github.io/qt-box-editor/

Step2: เมื่อมีไฟล์ box และรูปภาพที่ต้องการ ให้นำไฟล์ทั้งสองเข้าสู่ Tesseract-OCR โดยเรียกการใช้งานผ่าน command prompt โดยใช้คำสั่ง

tesseract tha.test_font.exp0.png tha.test_font.exp0.box nobatch box.train

Step3: ใช้ utility unicharset_extractor เพื่อสร้างไฟล์ unicharset โดยใช้คำสั่ง

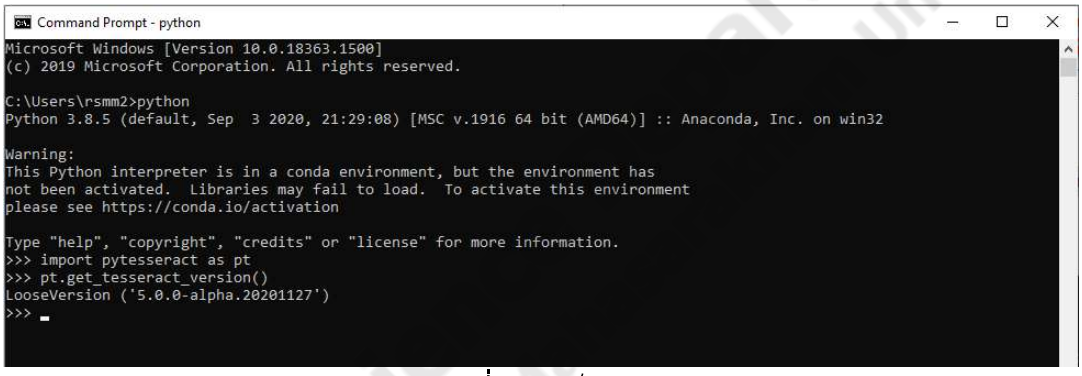
unicharset_extractor tha.test_font.exp0.box

Step4: สร้างไฟล์ font_properties ให้อยู่ใน directory เดียวกัน โดยใช้โปรแกรม Notepad ให้ save ชื่อว่า tha.font_properties โดยข้อความภายในให้มีรูปแบบดังนี้

<fontname> <italic> <bold> <fixed> <serif> <fraktur>

โดย fontname คือ ชื่อของ font ที่เราต้องการจะ train ซึ่งในที่นี้คือ test_font เพียงแค่ระบุลักษณะตามต้องการเพียงแค่ใส่ 0 หรือ 1 เช่น test_font ที่ได้นำมา train นั้น ไม่ได้มีลักษณะพิเศษใดๆ เลย จึงมีเพียงแค่ test font 0 0 0 0 0 ใช้คำสั่ง **shapeclustering -F font_properties -U unicharset tha.test_font.exp0.tr** ใช้คำสั่ง **mftraining font_properties -U unicharset -O tha.unicharset tha.test_font.exp0.tr** ใช้คำสั่ง **cntraining tha.test_font.exp0.tr** ใช้คำสั่ง **combine_tessdata tha.** เป็นการสิ้นสุดของกระบวนการเมื่อเสร็จเรียบร้อยแล้ว จะได้ไฟล์ tha.traineddata มา ซึ่งสามารถนำไปใช้เป็นไฟล์ข้อมูลสำหรับทำกระบวนการ OCR ได้

เช็คว่า pytesseract สามารถใช้งานได้แล้วหรือยังโดยการทดลองใช้ method ของ pytesseract



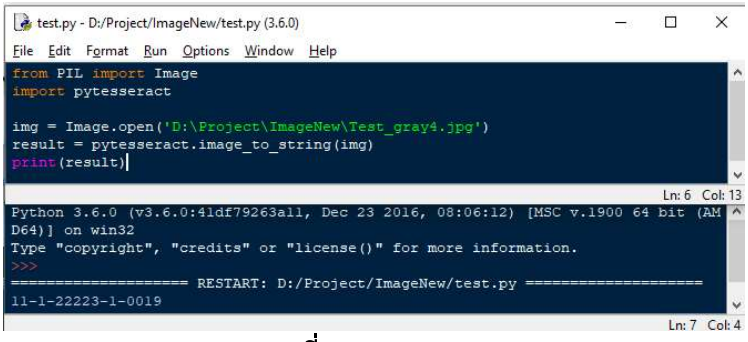
ภาพประกอบที่ 3.38 เช็ค pytesseract

ทำตามขั้นตอนเสร็จแล้วให้ทำการ install tesseract โดยเข้าไปที่หน้า Command Prompt แล้วพิมพ์คำสั่ง **pip install tesseract** ได้เลย

ตัวอย่างการใช้งาน

ภาพประกอบที่ 3.39 รูปตัวอย่าง อย.

รูปภาพที่ทดสอบ



ภาพประกอบที่ 3.40 รูปภาพการทดสอบ

3.5 โมเดลการจำแนกยี่ห้อกล่องนม

เมื่อผ่านกระบวนการ Pre-Processing แล้ว ภาพเหล่านั้นจะถูกส่งไปยังกระบวนการ การจำแนกกล่องนม โดยมีสถาปัตยกรรมการทำงานของ CNN

-Convolution Neural Network (CNN) เป็นขั้นตอนในการสกัดคุณลักษณะของภาพ โดยการทำ convolution โดยใช้ Sliding Windows (Filter) มาสแกนภาพเพื่อทำการแยกองค์ประกอบ วิธีดำเนินการ นำ filter สแกนไปบนภาพ และทำการคำนวณโดยการนำเอาตำแหน่ง ที่ตรงกันของภาพต้นฉบับ และ filter มาคูณกัน แล้วจึงนำผลรวมของทุกตำแหน่งมาบวกกัน (ตำแหน่งที่ 1 + ตำแหน่งที่ 2 + ... + ตำแหน่งที่ n) ตัวอย่างการคำนวณ

$$\frac{N-F+2P}{S}+1 \quad (3.10)$$

โดยที่ N = ขนาดของภาพ

F = จำนวนของ Filter

P = จำนวนของ Padding

S = จำนวนของ Stride

2	3	5	8	9	9
5	4	6	8	1	3
2	3	1	5	4	7
4	5	2	6	3	1
4	5	6	8	7	8
1	2	3	4	8	9

ภาพประกอบที่ 3.41 Input ขนาด 6x6

1	0
0	1

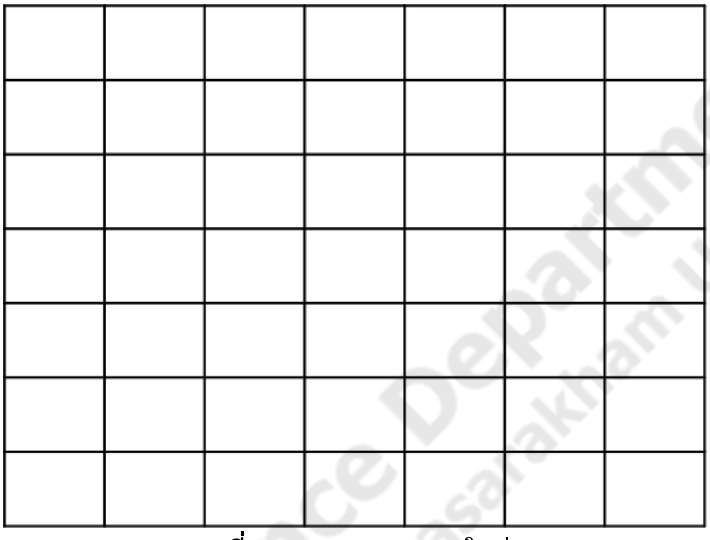
ภาพประกอบที่ 3.42 Filter หรือ Kernel ขนาด 3x3

การคำนวณหาค่าใหม่ทำได้โดยการกำหนด Filter ขนาด 2x2 ตามภาพ ประกอบที่3.42 จากนั้นกำหนด Stride = 1 (Stride คือ จำนวณของการขยับ Filter) และกำหนด Paddinng = 1

Computer Science Department Faculty of Informatics, Mahasarakham University

จากนั้นแทนค่าในสมการ Output of size เพื่อหาขนาดใหม่หลักจากการทำ Convolution แล้วแทนค่าตามสมการ (3.10)

$$\text{Output of size} = \frac{6\text{-}2\text{+}2(1)}{1} + 1 = 7$$



ภาพประกอบที่ 3.43 ขนาดของภาพใหม่ ขนาด 7x7

เนื่องจาก Padding เป็น 1 จะต้องแทนค่าขอบให้เป็น 0 จึงจะสามารถทำให้ Filter ว่างและสามารถ Stride ไปได้

0	0	0	0	0	0	0	0
0	2	3	5	8	9	9	0
0	5	4	6	8	1	3	0
0	2	3	1	5	4	7	0
0	4	5	2	6	3	1	0
0	4	5	6	8	7	8	0
0	1	2	3	4	8	9	0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.44 ภาพ Input ที่ทำการเพิ่มขอบและใส่ค่าเป็น 0 ที่ขอบ

Computer Science Department
Faculty of Informatics, Mahasarakham University

0	0	0	0	0	0	0	0
0	2	3	5	8	9	9	0
0	5	4	6	8	1	3	0
0	2	3	1	5	4	7	0
0	4	5	2	6	3	1	0
0	4	5	6	8	7	8	0
0	1	2	3	4	8	9	0
0	0	0	0	0	0	0	0

X

1	0
0	1

ภาพประกอบที่ 3.45 การหาภาพใหม่โดยการคูณ Filter 2x2

จากภาพที่ประกอบที่ 3.45 ให้นำ Filter 2x2 วางทับตามตำแหน่งดังตัวอย่าง ให้เอาตัวเลขที่ตรงช่องมาคูณกันแล้วนำผลคูณแต่ล่ะตัวมาบวกกัน ผลลัพธ์ที่ได้จากการบวกจะเป็นค่าใหม่ที่จะใช้แทนค่าในภาพใหม่ขนาด 7x7

รอบที่ 1

ตำแหน่งที่ 1 คือ 0 x 1 = 0

ตำแหน่งที่ 2 คือ 0 x 0 = 0

ตำแหน่งที่ 3 คือ 0 x 0 = 0

ตำแหน่งที่ 4 คือ 2 x 1 = 2

หลังจากหาค่าครบ 4 ตำแหน่งแล้วให้นำค่าแต่ล่ะตำแหน่งมาบวกกัน (ตำแหน่งที่ 1+ตำแหน่งที่ 2+ตำแหน่งที่ 3+ตำแหน่งที่ 4) จะได้ค่าเท่ากับ (0+0+0+2) = 2 และนำค่าที่ได้ไปแทนค่าในตารางขนาด 7x7 ในตำแหน่งที่ 1 ดังรูปภาพประกอบที่ 3.46 (รูปด้านล่าง)

2						

ภาพประกอบที่ 3.46 แทนค่ารอบที่ 1

Computer Science Department
Faculty of Informatics, Mahasarakham University

หลังจากแทนค่าลงแล้วจะขยับ Filter ไปด้านขวา1 ช่องดังภาพประกอบที่ 3.47

0	0	0	0	0	0	0	0
0	2	3	5	8	9	9	0
0	5	4	6	8	1	3	0
0	2	3	1	5	4	7	0
0	4	5	2	6	3	1	0
0	4	5	6	8	7	8	0
0	1	2	3	4	8	9	0
0	0	0	0	0	0	0	0

X

1	0
0	1

ภาพประกอบที่ 3.47 ขยับ Filter 1 ช่อง

รอบที่ 2

ตำแหน่งที่ 1 คือ 0 x 1 = 0

ตำแหน่งที่ 2 คือ 0 x 0 = 0

ตำแหน่งที่ 3 คือ 2 x 0 = 0

ตำแหน่งที่ 4 คือ 3 x 1 = 3

จะได้ (0+0+0+3) = 3 นำค่าที่หามาได้ใส่ในตาราง 7x7 ช่องที่2 ดังภาพประกอบที่ 3.48

2	3					

ภาพประกอบที่ 3.48 แทนค่าในช่องที่ 2 ในตาราง 7x7

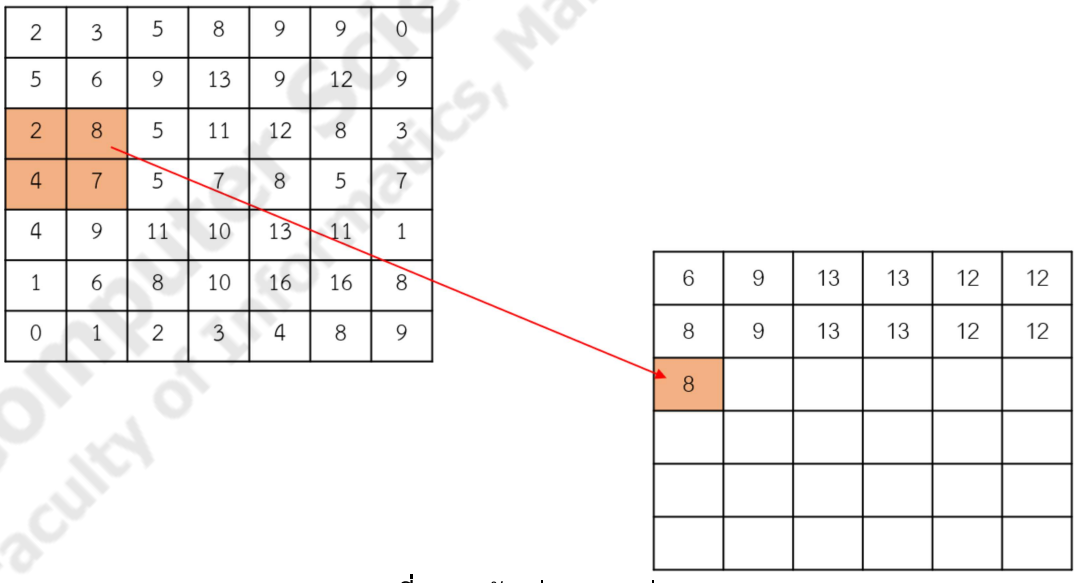
Computer Science Department
Faculty of Informatics Mahasarakham University

หลังจากนั้นทำให้ครบทุกช่องจะได้ผลลัพธ์ดังภาพประกอบที่ 3.49

2	3	5	8	9	9	0
5	6	9	13	9	12	9
2	8	5	11	12	8	3
4	7	5	7	8	5	7
4	9	11	10	13	11	1
1	6	8	10	16	16	8
0	1	2	3	4	8	9

ภาพประกอบที่ 3.49 ผลลัพธ์จากการทำ Convolution จนครบทุกช่อง

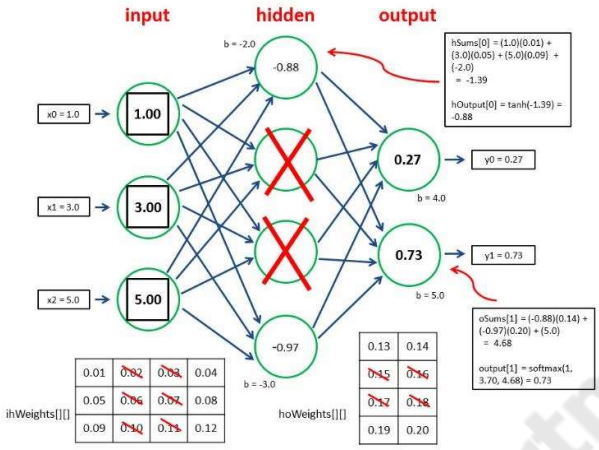
-Pooling Layer จะเป็นการ Resize ข้อมูลให้มีขนาดเล็กลงแต่จะมีความละเอียด Input ยังคงมีค่าเท่าเดิมแต่ความเร็วในการคำนวณจะเร็วขึ้น โดยจะใช้ Max-Pooling ในการหา โดยกำหนด Stride= 1 Paddin = 1 Filter มีขนาด 2x2 การหาค่า Max-Pooling จะเป็นการหาค่ามากที่สุดใน Filter และ Stride ในแต่ล่ะช่อง



ภาพประกอบที่ 3.50 ตัวอย่างการหาค่า Max-Pooling

-Dropout ขั้นตอนนี้จะช่วยลดเปอร์เซ็นต์การเกิด Over Fitting ของข้อมูล โดยจะให้ Network มีความยากขึ้น จะนำ Node ออกตามจำนวน Dropout ที่กำหนด เช่น 0.2 หมายถึง 20 % ของ Node ทั้งหมดจะไม่มีการ Update Weight ทำให้ช่วยลด Over Fitting ของข้อมูล

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.51 ตัวอย่างการทำงานของ Dropout

ที่มา: https://datascience.stackexchange.com/questions/44293/how-dropout-work-during-testing-in-neural-Network

การทำงานของ Dropout จะคูณน้ำหนักของการเชื่อมต่อ Input ของ Layer สุดท้าย โดยตัวแปรแต่ล่ะตัวที่แสดงในภาพประกอบที่ 3.51 มีความหมายดังนี้

x คือ Input

ihWeights คือ ค่าน้ำหนักใน Input Layer ขนาด 3x4

hoWeights คือ ค่าน้ำหนักใน hidden Layer ขนาด 4x2

hSum คือ ค่า input แต่ล่ะตัวคูณกับค่าน้ำหนัก ihWeights แต่ล่ะช่อง แล้วนำมาบวกกัน

hOutput คือ ค่าที่ได้จาก hSum มาหาค่าใน ไฮเปอร์โบลิกฟังก์ชัน

oSums คือ ค่าที่ได้จาก hOutput มาคูณกับ hoWeights แล้วนำมาบวกกัน

3.6 การประเมินประสิทธิภาพ

3.6.1 Confusion Matrix คือการประเมินผลลัพธ์การทำนาย (หรือผลลัพธ์จากโปรแกรม) เปรียบเทียบกับผลลัพธ์จริงๆ

True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่า จริง และผลการทำนายบอกว่า จริง

False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่า จริง และผลการทำนายบอกว่า ไม่จริง

True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่า ไม่จริง และผลการทำนายบอกว่า ไม่จริง

False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่า ไม่จริง และผลการทำนายบอกว่า จริง

Computer Science Department Faculty of Informatics Mahasarakham University

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

ภาพประกอบที่ 3.52 รูปตาราง Confusion Matrix

ตารางที่ 3.2 ตัวอย่างการ Predicted

Expected	Predicted	Expected	Predicted
DuchMilk	DuchMilk	SchoolMilk	SchoolMilk
DuchMilk	DuchMilk	SchoolMilk	SchoolMilk
DuchMilk	DuchMilk	SchoolMilk	SchoolMilk
DuchMilk	Foremost	SchoolMilk	DuchMilk
Foremost	Foremost	Vitamilk	Vitamilk
Foremost	Foremost	Vitamilk	Vitamilk
Foremost	SchoolMilk	Vitamilk	Vitamilk
Foremost	Vitamilk	Vitamilk	DuchMilk

ตารางที่ 3.3 การคำนวณ Confusion Matrix

	DuchMilk	Foremost	Foremost	Vitamilk
DuchMilk	3	0	1	1
Foremost	1	2	0	0
Foremost	0	1	3	0
Vitamilk	0	1	0	3

Computer Science Department
Faculty of Information Technology Rangsit University

1. Precision เป็นการวัดความแม่นยำของข้อมูล โดยพิจารณาแยกทีละคลาส

$$Precision = \frac{tp}{tp+fp} \quad (3.11)$$

จะได้ Precision (DuchMilk)= 3/(3+1+1) =0.75

Precision (Foremost)= 2/(2+1) =0.5

Precision (SchoolMilk)= 3/(3+1) =0.75

Precision (Vitamilk)= 3/(3+1) =0.75

2. Recall เป็นการวัดความถูกต้องของ Model โดยพิจารณาแยกทีละคลาส

$$Recall = \frac{tp}{tp+fn} \quad (3.12)$$

จะได้ Recall (DuchMilk) = 3/(3+1) = 0.75

Recall (Foremost) = 2/(2+1+1) = 0.5

Recall (SchoolMilk) = 3/(3+1) = 0.75

Recall (Vitamilk) = 3/(3+1) = 0.75

3. Accuracy เป็นการวัดความถูกต้องของ Model โดยพิจารณารวมทุกคลาส

$$Accuracy = \frac{tp+tn}{tp+tn+fp+fn} \quad (3.13)$$

จะได้ Accuracy (DuchMilk)= 3+8/(3+8+2+1) = 0.785714

Accuracy (Foremost)= 2+9/(2+9+1+2) = 0.785714

Accuracy (SchoolMilk)= 3+8/(3+8+1+1) = 0.846153

Accuracy (Vitamilk)= 3+8/(3+8+1+1) = 0.846153

3.6.2 การคำนวณ CER ขึ้นอยู่กับแนวคิดของระยะทาง โดยใช้ Levenshtein edit distance ซึ่ง เป็นขั้นตอนวิธีการวัดหาค่าความต่างกันของสายอักขระสองชุด ระหว่างชุดผลเฉลยกับผลการทำนาย โดยค่าความต่างกันจะวัดจากจำนวนของประมวลผลข้อความโดยการตัดออก การแทรก และการแทนที่ จะวัดค่า Character Error Rate (CER)

ภาพประกอบที่ 3.53 ตัวอย่างรูปภาพ อย. ในการทำนาย

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตารางที่ 3.4 ตารางผลการทำนาย

/	1	3	-	1	-	1	1	1	8	0	-	6	-	0	0	8	0

1. ตัด / ออก 1 ครั้ง

จะได้ผลลัพธ์ ดังตารางนี้

ตารางที่ 3.5 ตารางผลการทำนายที่ทำการแก้ไข

1	3	-	1	-	1	1	1	8	0	-	6	-	0	0	8	0

2.แทรกตัวเลข 0 ครั้ง

3.แทนที่ เลข 8 เป็นเลข 3 2 ครั้ง

4.แทนที่ เลข 0 เป็นเลข 5 1 ครั้ง

จะได้ผลลัพธ์ ดังตารางนี้

ตารางที่ 3.6 ตารางผลการทำนายที่ถูกต้อง

1	3	-	1	-	1	1	1	3	5	-	6	-	0	0	3	0

$$\text{จะได้ CER} = \frac{(\text{ผลรวมของจำนวนที่ตัดออก แทรก และ แทนที่})}{\text{จำนวนของอักขระที่ถูกต้อง}} \times 100$$

$$\text{แทนค่า CER} = \frac{4}{14} \times 100 = 28.5714$$

ดังนั้น Error rate ของรูปภาพที่ มีค่าผิดพลาดอยู่ที่ 28.5714 %

3.7 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลที่ใช้ในการทดสอบ Model แบ่งออกเป็น 3 ชุด ได้แก่

1. ชุดข้อมูลรูปภาพยี่ห้อกล่องนม มีทั้งหมด 4 ยี่ห้อ ยี่ห้อละ 300 กล่อง

(ก) ยี่ห้อ Foremost (ข) ยี่ห้อ School Milk

ภาพประกอบที่ 3.54 ยี่ห้อกล่องนม

Computer Science Department
Faculty of Informatics Mahasarakham University

(ค) ยี่ห้อ Duch Milk

(ง) ยี่ห้อ Vitamilk

ภาพประกอบที่ 3.54 ยี่ห้อกล่องนม (ต่อ)

2 .ชุดข้อมูลรูปภาพ อย. มีทั้งหมด 4 ยี่ห้อ ยี่ห้อละ 300 กล่อง

(ก) อย. Foremost

(ข) อย. School Milk

(ค) อย. Duch Milk

(ง) อย. Vitamilk

ภาพประกอบที่ 3.55 อย.กล่องนม

Computer Science Department
Faculty of Informatics Mahasarakham University

3. ชุดข้อมูลรูปภาพวันหมดอายุ มีทั้งหมด 4 ยี่ห้อ ยี่ห้อละ 300 กล่อง

(ก) วันหมดอายุ Foremost

(ข) วันหมดอายุ School Milk

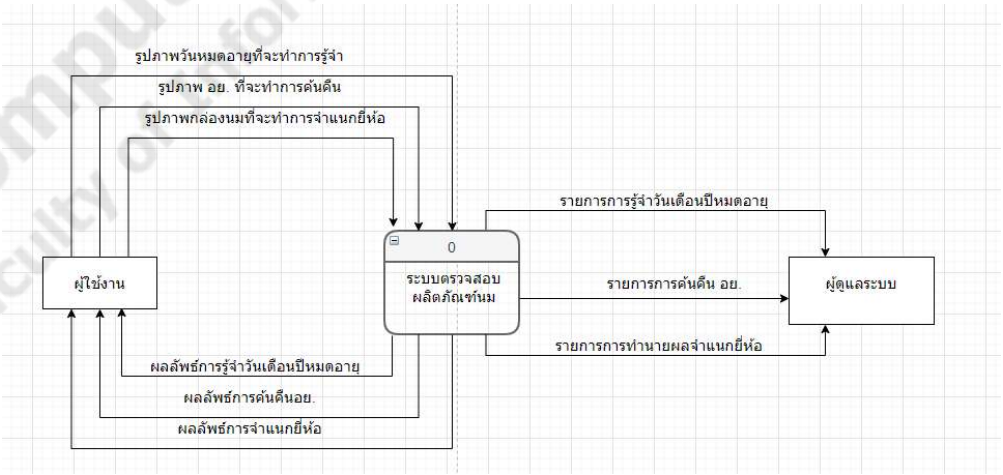
(ค) วันหมดอายุ Dutch Milk

(ง) วันหมดอายุ Vitamilk

ภาพประกอบที่ 3.56 วันหมดอายุกล่องนม

3.8 Data Flow Diagram

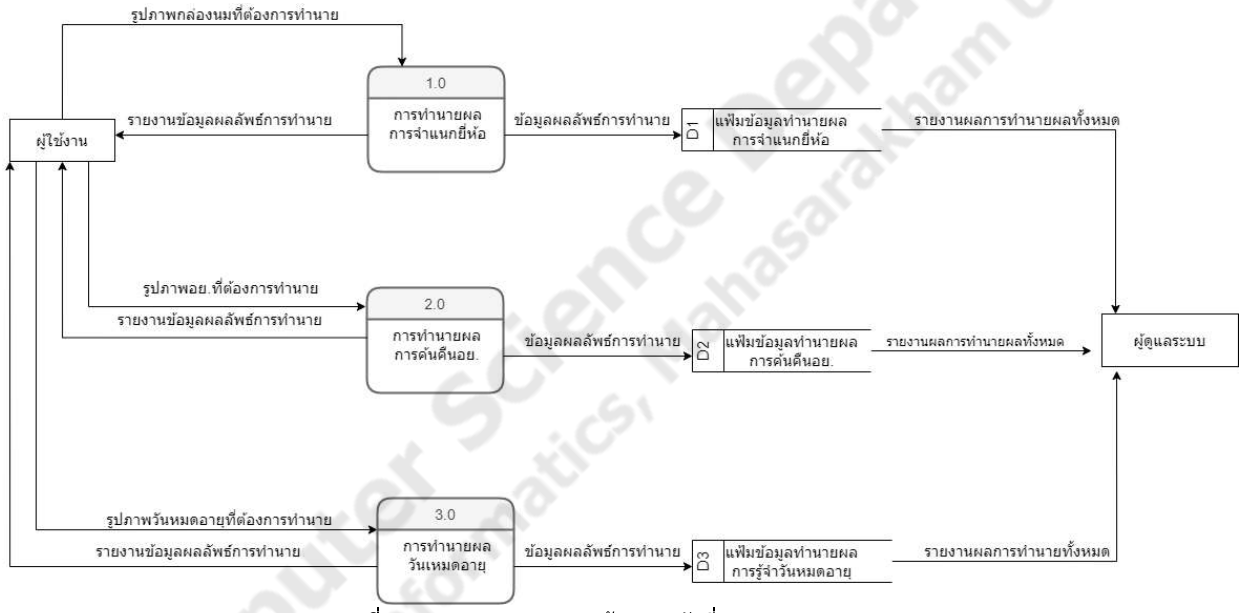
3.8.1 Context Diagram



ภาพประกอบที่ 3.57 ภาพ Context Diagram

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.8.2 Data Flow Diagram Level 1



ภาพประกอบที่ 3.58 ภาพแผนภาพกระแสข้อมูล ระดับที่1 (Data Flow Diagram Level)

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.8.3 External Entity Description

ตารางที่ 3.7 ตารางแสดง External Entity Description

Name	Description	Input Data Flow	Output Data Flow
ผู้ดูแลระบบ	ผู้ที่ดูแลระบบตรวจสอบผลิตภัณฑ์นม โดยผู้ดูแลระบบสามารถดูผลการทำนายที่ผู้ใช้งานได้ทำการประมวลผลไว้ได้	- รายการการรู้จำวันเดือนปีหมดอายุ - รายการการค้นคืนอย. - รายการการทำทายผลจำแนกยี่ห้อ	
ผู้ใช้งาน	ผู้ใช้บริการระบบสามารถนำรูปภาพเพื่อเข้ามาประมวลผลได้ สามารถรู้ข้อมูลการทำทายได้	-ผลลัพธ์การรู้จำวันเดือนปีหมดอายุ -ผลลัพธ์การค้นคืนอย. -ผลลัพธ์การจำแนกยี่ห้อ	-รูปภาพวันหมดอายุที่จะทำการรู้จำ -รูปภาพ อย. ที่จะทำการค้นคืน -รูปภาพกล่องนมที่จะทำการจำแนกยี่ห้อ

3.8.4 Data Store Description

ตารางที่ 3.8 ตารางแสดง Data Store Description

ID	Name	Description	Data Structure
D1	แฟ้มข้อมูลทำนายผลการจำแนกยี่ห้อ	ฐานข้อมูลเก็บผลลัพธ์จากการทำนายผลการจำแนกยี่ห้อ	รหัสการทำนายผล+วันที่ทำนาย+เวลาทำนาย+ผลของการทำนาย

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตารางที่ 3.8 ตารางแสดง Data Store Description (ต่อ)

ID	Name	Description	Data Structure
D2	แฟ้มข้อมูลทำนายผลการค้นคืนอย.	ฐานข้อมูลเก็บผลลัพธ์จากการทำนายผลการค้นคืนอย.	รหัสการทำนายผล+วันที่ทำนาย+เวลาทำนาย+ผลของการทำนาย
D3	แฟ้มข้อมูลทำนายผลการรู้จำวันหมดอายุ	ฐานข้อมูลเก็บผลลัพธ์จากการทำนายผลการรู้จำวันหมดอายุ	รหัสการทำนายผล+วันที่ทำนาย+เวลาทำนาย+ผลของการทำนาย

3.8.5 Data Structure Description

ตารางที่ 3.9 ตารางแสดง Data Structure Description

Name	Description	Source	Destination	Data Structure
รูปภาพกล่องนมที่ต้องการทำนาย	ไฟล์รูปภาพที่ผู้ใช้ส่งเข้ามาประมวลผล	ผู้ใช้งาน	Process 1.0 การทำนายผลการจำแนกยี่ห้อ	รหัสของผลการทำนาย+ชื่อไฟล์การทำนาย+วันที่ทำนาย+เวลาทำนาย
		Process 1.0 การทำนายผลการจำแนกยี่ห้อ	D1 แฟ้มข้อมูลทำนายผลการจำแนกยี่ห้อ	
รายงานข้อมูลการทำนาย	เป็นผลลัพธ์จากการทำนาย	Process 1.0 การทำนายผลการจำแนกยี่ห้อ	ผู้ใช้งาน	รหัสของผลการทำนาย+ชื่อไฟล์การทำนาย+วันที่ทำนาย+เวลาทำนาย

Computer Science Department Faculty of Informatics, Mahasarakham University

ตารางที่ 3.9 ตารางแสดง Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
รายการการทำนายผลทั้งหมด	รายการการประมวลผลทั้งหมดของผู้ใช้ทุกคนที่เคยประมวลผลไว้	D1 แฟ้มข้อมูลทำนายผลการจำแนกยี่ห้อ	ผู้ดูแลระบบ	ชื่อไฟล์ + เวลาที่ทำนาย + วันที่ทำนาย + ผลลัพธ์ที่ได้จากการทำนาย
รูปภาพอย. ที่ต้องการทำนาย	ไฟล์รูปภาพที่ผู้ใช้ส่งเข้ามาประมวลผล	ผู้ใช้งาน	Process 2.0 การทำนายผลการค้นคืนอย.	รหัสของผลการทำนาย+ชื่อไฟล์การทำนาย+ วันที่ทำนาย+ เวลาทำนาย
		Process 2.0 การทำนายผลการค้นคืนอย.	D2 แฟ้มข้อมูลทำนายผลการค้นคืนอย.	
รายงานข้อมูลการทำนาย	เป็นผลลัพธ์จากการทำนาย	Process 2.0 การทำนายผลการค้นคืนอย.	ผู้ใช้งาน	รหัสของผลการทำนาย+ชื่อไฟล์การทำนาย+ วันที่ทำนาย+ เวลาทำนาย
รายการการทำนายผลทั้งหมด	รายการการประมวลผลทั้งหมดของผู้ใช้ทุกคนที่เคยประมวลผลไว้	D2 แฟ้มข้อมูลทำนายผลการค้นคืนอย.	ผู้ดูแลระบบ	ชื่อไฟล์ + เวลาที่ทำนาย + วันที่ทำนาย + ผลลัพธ์ที่ได้จากการทำนาย
รายงานข้อมูลการทำนาย	เป็นผลลัพธ์จากการทำนาย	Process 3.0 การทำนายผลวันหมดอายุ	ผู้ใช้งาน	รหัสของผลการทำนาย+ชื่อไฟล์การทำนาย

Computer Science Department Faculty of Informatics, Mahasarakham University

ตารางที่ 3.9 ตารางแสดง Data Structure Description (ต่อ)

Name	Description	Source	Destination	Data Structure
รูปภาพวันหมดอายุที่ต้องการทำนาย	ไฟล์รูปภาพที่ผู้ใช้ส่งเข้ามาประมวลผล	ผู้ใช้งาน	Process 3.0 การทำนายผลวันหมดอายุ	รหัสของผลการทำนาย+ชื่อไฟล์การทำนาย+วันที่ทำนาย+เวลาทำนาย
		Process 3.0 การทำนายผลวันหมดอายุ	D3 แฟ้มข้อมูลทำนายผลการรู้จำวันหมดอายุ	
รายการการทำนายผลทั้งหมด	รายการการประมวลผลทั้งหมดของผู้ใช้	D3 แฟ้มข้อมูลทำนายผลการรู้จำวันหมดอายุ	ผู้ดูแลระบบ	ชื่อไฟล์ + เวลาที่ทำนาย + วันที่ทำนาย

Computer Science Department
Faculty of Informatics, Mahasarakham University