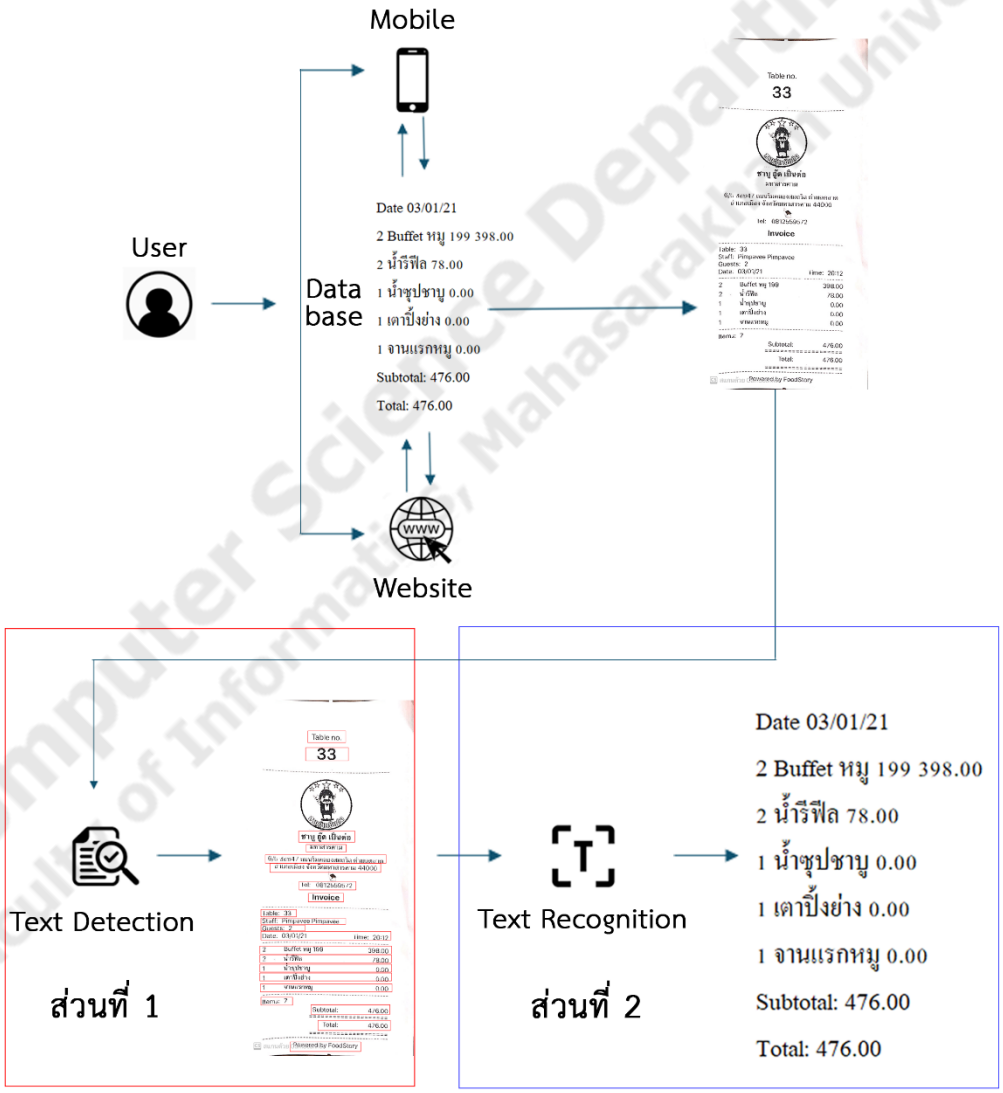
บทที่ 3

วิธีดำเนินงานวิจัย

ในบทนี้จะอธิบายถึงชุดข้อมูลใบเสร็จที่ใช้ในโครงงานนี้ และขั้นตอนในการดำเนินงานของการแปลงใบเสร็จรับเงินอัตโนมัติ ดังนี้

3.1 กรอบการดำเนินงาน

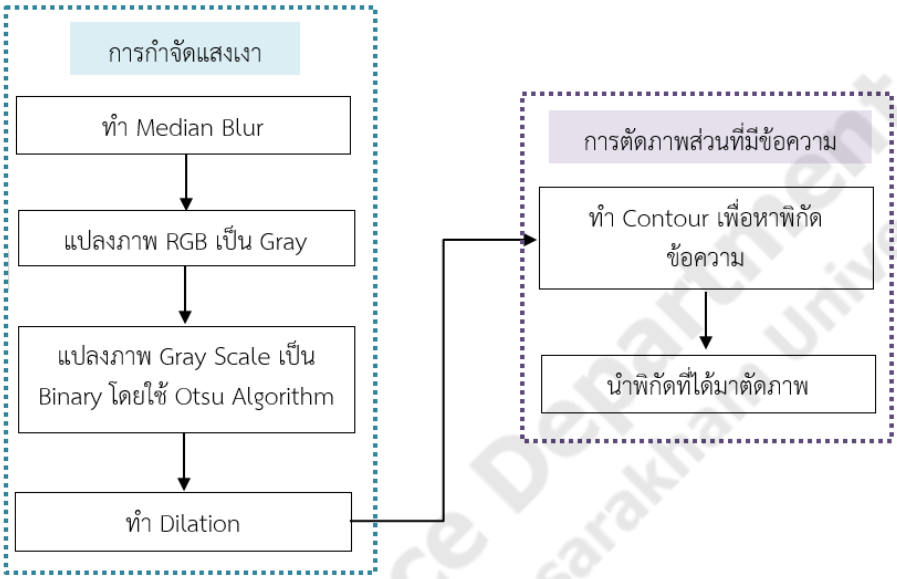


ภาพประกอบที่ 3.1 กรอบการดำเนินงานของระบบ

Computer Science Department Faculty of Informatics, Mahasarakham University

จากภาพประกอบที่ 3.1 จะเห็นว่าได้มีการแบ่งขั้นตอนการทำงานออกเป็น 2 ขั้นตอนหลัก ได้แก่

ส่วนที่ 1: การตรวจจับข้อความ (Text Detection)



ภาพประกอบที่ 3.2 ขั้นตอนการตรวจจับข้อความ

ขั้นตอนนี้คือขั้นตอนการเตรียมภาพก่อนนำไปรู้จำตัวอักษร โดยนำภาพที่เป็น Input ไปเข้าสู่กระบวนการ Pre-process เพื่อจำกัดแสงเงาและสิ่งรบกวนต่างๆบนรูปภาพที่อาจส่งผลต่อการรู้จำตัวอักษร โดยในขั้นตอนแรกของกระบวนการ Pre-process หลังจากอ่านไฟล์ภาพเข้ามาแล้วจะนำภาพไปทำ Median Blur เพื่อจำกัดแสงเงาและสิ่งรบกวนต่างๆ ทำให้ภาพ Smoot ขึ้น จากนั้นนำภาพที่ได้ไปแปลงเป็นภาพระดับเทา แล้วนำไปแปลงเป็นภาพสองระดับด้วย Thresholding จากนั้นนำภาพขาวดำไปทำ Dilation แล้วจึงนำภาพนั้นไปหาพิกัดของบรรทัดและนำไป Crop เพื่อเป็น Input ของกระบวนการรู้จำตัวอักษรต่อไป

ส่วนที่ 2: การรู้จำตัวอักษร (Text Recognition)

ขั้นตอนนี้คือระบบจะทำการรู้จำตัวอักษร โดยภาพที่ถูกนำเข้าจะเข้าสู่กระบวนการทำ Pre-processing เพื่อเตรียมข้อมูลก่อนเข้าสู่ Model ที่เป็นกระบวนการในการรู้จำ แล้วจึงได้ผลลัพธ์ออกมา

Computer Science Department
Faculty of Informatics Mahasarakham University

3.2 การตรวจจับข้อความ (Text Detection)

3.2.1 ภาพนำเข้า เป็นภาพใบเสร็จสี RGB ที่ได้มาจากการสแกน หรือการถ่ายรูปพื้นหลังของภาพไม่มีเส้นหรือบรรทัดลวดลายใดๆ

Table no.

23

ชาบู อู๊ต เป็นต่อ

มหาสารคาม

6/5 ซอย47 ถนนริมคลองสมยวิน ตำบลตลาด
อำเภอเมือง จังหวัดมหาสารคาม 44000

Tel: 0812559672

Receipt

Table: 23
Staff: Pimpavee Pimpavee
Guests: 2 ID: L27S9
Date: 19/12/20 Time: 21:23

1	จานแรกหมู	0.00
1	น้ำซุปน้ำต๋ำญี่ปุ่น	0.00
1	น้ำซุปต้มยำน้ำข้น	0.00
2	Buffet หมูทะเล 249	498.00
2	น้ำรีฟิล	78.00

Items: 7

Subtotal: 576.00

Total: 576.00

Cash 576.00

Change 0.00

Powered by FoodStory

ภาพประกอบที่ 3.3 ตัวอย่างภาพนำเข้า

3.2.2 การเตรียมพร้อมภาพ (Pre-Processing)

เป็นขั้นตอนในการเตรียมภาพนำเข้าเพื่อให้พร้อมต่อการนำไปประมวลผลต่อไป โดยจะประกอบไปด้วยขั้นตอนในการกำจัดแสงเงา การแปลงภาพให้เป็นไบนารี ขาว - ดำ และตัดรูปภาพให้มีเฉพาะตรงส่วนที่มีข้อความอยู่ โดยขั้นตอนและวิธีการต่างๆ สามารถอธิบายได้ ดังต่อไปนี้

(1) การกำจัดแสงเงา

เป็นขั้นตอนในการกำจัดแสง หรือเงาที่ตกกระทบจากการถ่ายภาพ โดยเริ่มจากการทำ Median blur เพื่อทำให้ภาพมีความ Smooth จากนั้นแยกช่องสีของภาพ RGB และส่งแต่ละช่องสีเข้าสู่กระบวนการทำ ไดเลชัน (Dilation) เพื่อให้เข้าใจการทำงานในแต่ละขั้นตอนให้มากขึ้น จะอธิบายขั้นตอนวิธีการคำนวณ ดังต่อไปนี้

- การทำ Median Blur เพื่อกำจัดสัญญาณรบกวนและทำให้ภาพมีความ Smooth ขึ้น

วิธีการ ตัวอย่างการกรองภาพขนาด 5 x 5 โดยหาค่าที่ตำแหน่ง (3,3) จะนำค่าโดยรอบตำแหน่ง (3,3) มาเรียงค่ากันจากค่าน้อยไปหาค่ามาก (ภาพประกอบที่ 3.5) แล้วเลือกค่าที่อยู่กึ่งกลางไปแทนที่พิกเซลตรงกลาง (ภาพประกอบที่ 3.6) จะทำแบบนี้และขยับไปเรื่อยๆ จนครบทุกพิกเซล

Computer Science Department
Faculty of Informatics, Mahasarakham University

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

ภาพประกอบที่ 3.4 ตัวอย่างข้อมูลภาพก่อนทำ Median ที่ตำแหน่ง (3,3)

115	119	120	123	124	125	126	127	150

ภาพประกอบที่ 3.5 ตัวอย่างเรียงพิกเซลภาพจากค่าน้อยไปหาค่ามาก ที่ตำแหน่ง (3,3)

123	125	126	130	140
122	124	126	127	135
118	120	124	125	134
119	115	119	123	133
111	116	110	120	130

ภาพประกอบที่ 3.6 ตัวอย่างหลังทำ Median ที่ตำแหน่ง (3,3)

123	125	126	130	140
122	124	126	127	135
118	120	124	125	134
119	115	119	123	133
111	116	110	120	130

ภาพประกอบที่ 3.7 ตัวอย่างข้อมูลภาพก่อนทำ Median ที่ตำแหน่ง (3,4)

Computer Science Department
Faculty of Informatics Mahasarakham University

119	123	124	125	126	127	133	134	135

ภาพประกอบที่ 3.8 ตัวอย่างเรียงพิกเซลภาพจากค่าน้อยไปหาค่ามาก ที่ตำแหน่ง (3,4)

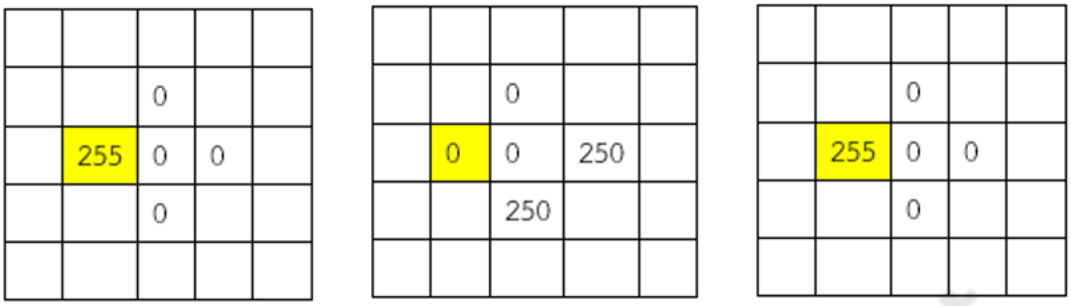
123	125	126	130	140
122	124	126	127	135
118	120	124	126	134
119	115	119	123	133
111	116	110	120	130

ภาพประกอบที่ 3.9 ตัวอย่างหลังทำ Median ที่ตำแหน่ง (3,4)

ภาพประกอบที่ 3.10 ภาพก่อนละหลังการทำ Median Blur

- **การแปลงภาพสี RGB ไปเป็นภาพระดับเทา (Gray Scale)** ขั้นตอนการแปลงภาพทำได้โดยอาศัยค่าของ RGB ที่อยู่ในแต่ละพิกเซลของภาพต้นฉบับมาแปลงเป็นภาพระดับเทา โดยช่องสี R จะมีค่าอยู่ที่ 0 – 255 ช่องสี G มีค่าอยู่ที่ 0 – 255 และช่องสี B จะมีค่าอยู่ที่ 0 – 255

Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.11 ตารางค่าสี R G B ตามลำดับ

วิธีการ ถ้านำพิกเซล ตำแหน่งที่ (3,4) จากตารางมาแปลงเป็นภาพระดับเทาสามารถทำได้ดังนี้

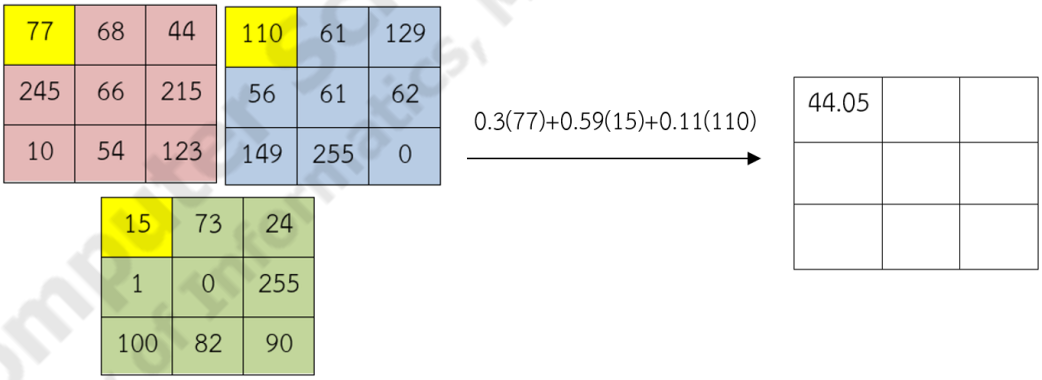
จากสูตร Gray = $0.3R + 0.59G + 0.11B$ (3.1)

แทนค่า Gray = 0.3(255)+0.59(0)+0.11(255)

จะได้ค่า Gray = 76.5+0+28.05 = 104.55

โดยค่าคำตอบของ Gray จะถูกแทนลงในตำแหน่ง (3,4) ของภาพระดับเทา

ตัวอย่างการแปลงภาพสี RGB เป็นภาพระดับเทา ของภาพที่มีขนาด 3 x 3 พิกเซล



ภาพประกอบที่ 3.12 การแปลงภาพสี RGB ไปเป็นภาพระดับเทา

44.05	70.18	39.78
80.25	26.51	221.77
78.39	92.63	90

ภาพประกอบที่ 3.13 ตัวอย่างหลังจากการแปลงเป็นภาพระดับเทา

Computer Science Department
Faculty of Informatics, Mahasarakham University

ภาพประกอบที่ 3.14 ตัวอย่างภาพต้นฉบับก่อนการแปลงเป็นภาพระดับเทา

ภาพประกอบที่ 3.15 ตัวอย่างภาพหลังการแปลงเป็นภาพระดับเทา

- **การแปลงภาพระดับเทา ไปเป็นภาพขาวดำ (Binary image)**

เป็นการแปลงค่าสีจากระดับเทาที่มีค่าอยู่ระหว่าง 0 – 255 ให้เป็นภาพแบบไบนารีที่มีค่า 0 และ 255 โดยจะหาค่าขีดแบ่ง (Threshold) ที่เหมาะสม โดยใช้ Otsu จะคำนวณหาความแปรปรวนระหว่างกลุ่มเพื่อหาค่าขีดแบ่งที่ดีที่สุดในการแยกวัตถุออกจากพื้นหลัง โดยพิจารณาจากค่า t ที่เป็นไปได้ทั้งหมด โดยมีวิธีการคำนวณดังนี้

ขั้นตอนที่ 1 กำหนดค่า L ให้เท่ากับ 4 และ n_i = {29,21,37,18}

Computer Science Department
Faculty of Informatics Mahasarakham University

ขั้นตอนที่ 2 คำนวณหาค่าความน่าจะเป็น ดังสมการ

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=0}^{L-1} p_i = 1 \quad (3.2)$$

โดยที่ p_i คือ จำนวนจุดภาพที่ระดับความเข้มที่ i

n_i คือ จำนวนพิกเซล ณ ระดับที่ i

N คือ ผลรวมของพิกเซลทั้งหมด ในแต่ละระดับ

L คือ ระดับความเข้มของภาพระดับสีเทา

- คำนวณความน่าจะเป็น p_i

$$p_0 = \frac{29}{105} = 0.2761$$
$$p_1 = \frac{21}{105} = 0.2$$
$$p_2 = \frac{37}{105} = 0.3524$$
$$p_3 = \frac{18}{105} = 0.1714$$

จะได้ $p_i = \{0.2761, 0.2, 0.3524, 0.1714\}$

ขั้นตอนที่ 3 คำนวณหาค่าน้ำหนักของ ω_0 และ ω_1 โดยค่าน้ำหนักคำนวณได้ตามสมการดังนี้

$$\omega_0 = \sum_{i=1}^{K} p_i = \omega(k) \quad (3.3)$$

โดยที่ ω_0 คือ ความน่าจะเป็นของกลุ่มที่ 1

$k = 1$

$p = 0.2761$

$\omega_0 = \sum_{i=1}^{K} p_i = 0.2761$

$$\omega_1 = \sum_{i=k+1}^{L} p_i \quad (3.4)$$

โดยที่ ω_1 คือ ความน่าจะเป็นของกลุ่มที่ 2

$k = 1$

$p = \{0.2 + 0.3524 + 0.1714\}$

$\omega_1 = \sum_{i=k+1}^{L} p_i = 0.7238$

- คำนวณหาค่า ω_0 และ ω_1

$\omega_0 = \{0.2761, 0.3524, 0.1714\}$

$\omega_1 = \{0.7238, 0.3524, 0.1714\}$

ขั้นตอนที่ 4 คำนวณหาเฉลี่ย ดังสมการ

$$\mu_0 = \frac{\mu(k)}{\omega(k)} \quad (3.5)$$

โดยที่ μ_0 คือ ค่าเฉลี่ยของกลุ่มที่ 1

$\mu_k = \sum_{i=1}^{K} i * p_i$ คือ ผลรวมของ $i * p_i$ ตั้งแต่ $i = k + 1$ ถึง L

$$u_1 = \sum_{i=k+1}^{L} \frac{i * p_i}{\omega_1} \quad (3.6)$$

โดยที่ μ_1 คือ ค่าเฉลี่ยของกลุ่มที่ 2

Computer Science Department
Faculty of Informatics, Mahasarakham University

$u_1 = \sum_{i=k+1}^{L} \frac{i*p_i}{\omega_1}$ คือ ผลรวมของ $\frac{i*p_i}{\omega_1}$ ตั้งแต่ $i = k+1$ ถึง L

- คำนวณหาค่า μ_0 μ_1และ μ_k แทนค่าในสูตร กำหนด $k = 3$

$$\mu_k = (1 * 0.2761) = 0.2761$$
$$\mu_0 = (1 * 0.2761) + (2 * 0.2) + (3 * 0.3524) = 1.7333$$
$$\mu_1 = \left(\frac{2*0.2}{0.7238}\right) + \left(\frac{3*0.3524}{0.7238}\right) + \left(\frac{4*0.1714}{0.7238}\right) = 2.9604$$

- คำนวณหาค่า μ_0 μ_1และ μ_k ทุกตำแหน่ง โดยกำหนด $k = 1$

$$\mu_k = \{0.2761, 0.4, 1.0572\}$$
$$\mu_0 = \{1, 0.8402, 1.2760\}$$
$$\mu_1 = \{0.5526, 2.0132, 2.9604\}$$

ขั้นตอนที่ 5 ทำการคำนวณหาค่าความแปรปรวน ดังสมการ

$$\sigma_B^2 = \omega_0\omega_1(\mu_0 - \mu_1)^2 \quad (3.7)$$

โดยที่ σ_B^2 คือ ค่าความแปรปรวนของกลุ่มตั้งแต่รอบที่ 1 – 255

$u_1 = \sum_{i=k+1}^{L} \frac{i*p_i}{\omega_1}$ คือ ผลรวมของ $\frac{i*p_i}{\omega_1}$ ตั้งแต่ $i = k+1$ ถึง L

- คำนวณหาค่า σ_B^2 จะได้

$$\sigma_B^2 = \{0.04, 0.6545, 2.3492\}$$

ขั้นตอนที่ 6 คำนวณหาค่าความแปรปรวนสูงสุด ดังสมการ

$$\sigma_B^2(k^*) = \max_{1 \le k < L} \sigma_B^2(k) \quad (3.8)$$

โดยที่ $\sigma_B^2(k^*)$ คือ ค่าความแปรปรวนสูงสุดของกลุ่มตั้งแต่รอบที่ k = 1 – 255

- คำนวณหาค่า $\sigma_B^2(k^*)$ จะได้

$$\sigma_B^2(k^*) = 2.3492$$

โดยจะเลือกค่า k ที่ให้ค่าความแปรปรวนสูงสุด คือ 2 ที่เท่ากับ 2.3492 กำหนดให้เป็นค่า Threshold จะได้ t = 2 จะได้ผลลัพธ์ดังภาพประกอบที่ 3.16

ภาพประกอบที่ 3.16 ตัวอย่างการแปลงภาพระดับเทา เป็นภาพ Binary

Computer Science Department
Faculty of Informatics, Mahasarakham University

- **การทำ Dilation** เป็นการขยายส่วนของวัตถุในภาพ โดยจะพิจารณาข้อมูลภาพที่อยู่ในรูปแบบไบนารี (Binary) และส่วนประกอบโครงสร้างซึ่งเป็นแบบ 4 ทิศทาง

1	1	1
1	1	1
1	1	1

ภาพประกอบที่ 3.17 ส่วนประกอบโครงสร้าง 4 ทิศทาง

วิธีดำเนินการ คือ จะกำหนดส่วนประกอบโครงสร้าง ดังภาพประกอบที่ 3.17 และนำส่วนประกอบโครงสร้างสแกนไปยังทุกข้อมูลของภาพตามลำดับตลอดทั้งภาพ โดยเมื่อจุดเริ่มต้นของส่วนประกอบโครงสร้างตรงกับตำแหน่งข้อมูลภาพต้นฉบับที่ตรงกันมีค่าเท่ากับ 1 จะทำการยูเนียน (Union) ส่วนประกอบโครงสร้างเข้ากับภาพต้นฉบับ

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

ภาพประกอบที่ 3.18 ภาพต้นฉบับก่อนการทำ Dilation

0⊕1	0⊕1	0⊕1	0	0	0	0	0
0⊕1	1⊕1	1⊕1	0	1	1	1	0
0⊕1	1⊕1	1⊕1	1	1	1	1	0
0	1	0	0	1	1	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

ภาพประกอบที่ 3.19 ขั้นตอนการนำภาพต้นฉบับและส่วนประกอบโครงสร้างมายูเนียนกัน

Computer Science Department
Faculty of Informatics, Mahasarakham University

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

ภาพประกอบที่ 3.20 ภาพหลังการยูเนียนกันของ ภาพประกอบที่ 3.19

ภาพประกอบที่ 3.21 ภาพผลลัพธ์การทำ Dilation

3.2.3 การตัดภาพส่วนที่มีข้อความ

ในขั้นตอนนี้จะทำการตัดภาพให้เหลือเฉพาะส่วนที่มีข้อความอยู่เท่านั้น โดยทำการหาพิกัด (x,y) เริ่มต้นของภาพ ความกว้าง และความสูงของภาพตรงส่วนที่มีข้อความปรากฏ แล้ว crop ตัดรูปภาพนั้น ตามพิกัดที่ได้มา โดยมีรายละเอียดและขั้นตอนดังต่อไปนี้

- **การ Contour**

วิธีดำเนินการ กำหนดตัวแปรแต่ละตัว หมายถึง

f_{ij} คือ ค่าของพิกเซลที่ตำแหน่ง (i,j)

i คือ ค่าชองพิกเซลที่ตำแหน่งแถวบนสุด และแถวล่างสุด

Computer Science Department Faculty of Informatics Mahasarakham University

j คือ ค่าของพิกเซลที่ตำแหน่งคอลัมน์ซ้ายสุด และคอลัมน์ขวาสุด

เงื่อนไขของการ Contour คือจะกำหนดหมายเลขเฉพาะให้กับเส้นขอบใหม่ที่พบเป็น NBD

เงื่อนไขที่ 1 if เส้นขอบด้านนอกของวัตถุ (f_{ij} = 1 และ $f_{i,j-1}$ = 0)

ให้นำ NBD เข้า และตั้งค่า i2,j2 เป็น i,j-1

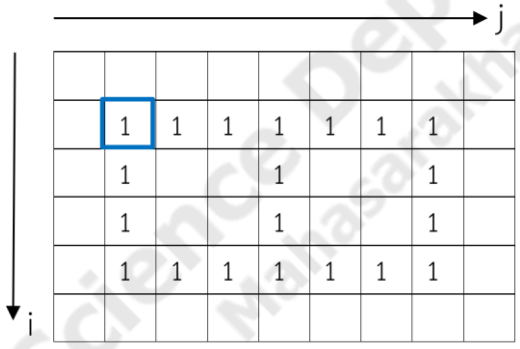
else if เส้นขอบด้านในของวัตถุ

ให้นำ NBD เข้า และตั้งค่า i2,j2 เป็น i,j+1

else if f_{ij} > 1

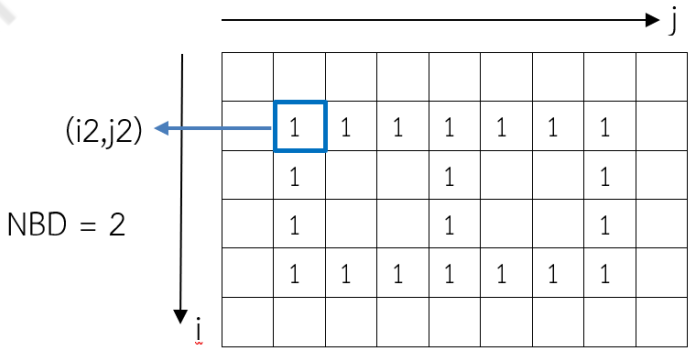
ให้ NBD = NBD

ขั้นตอนที่ 1 เริ่มสแกนรูปภาพจากซ้ายไปขวาจนกว่าจะพบพิกเซลวัตถุ โดยหากค่าพิกเซลที่พบมีค่ามากกว่าหรือเท่ากับ 1 แปลว่าพิกเซลนี้อาจเป็นขอบภาพของวัตถุดังภาพ



ภาพประกอบที่ 3.22 ตัวอย่างการสแกนรูปภาพจากซ้ายไปขวา

ขั้นตอนที่ 2 หลังจากได้จุดที่น่าจะเป็นขอบภาพแล้วจะทำตามขั้นตอนดังเงื่อนไขที่1 และทุกครั้งที่เริ่มสแกนแถวใหม่ จะต้องรีเซ็ต NBD ให้เป็น 1



ภาพประกอบที่ 3.23 ตัวอย่างการ Contour ขั้นตอนที่ 2

Computer Science Department
Faculty of Informatics, Mahasarakham University

ขั้นตอนที่ 3 เริ่มหาเส้นขอบ

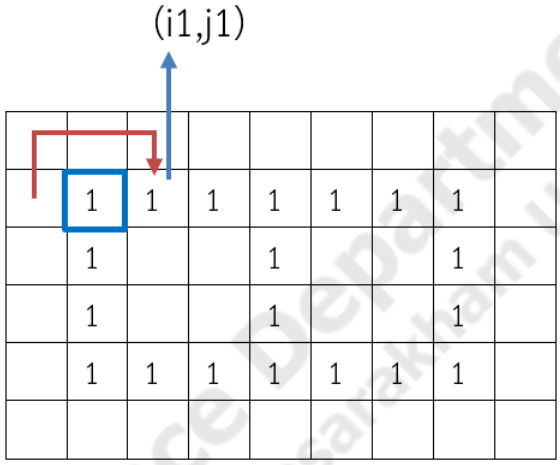
- เริ่มหาเส้นขอบจาก (i2,j2) โดยจะดูรอบๆตามเข็มนาฬิกาของพิกเซลในพื้นที่ใกล้เคียงของ (i,j) โดยมีเงื่อนไขการหาดังนี้

เงื่อนไขที่ 2 if pixel != 0

ให้ระบุว่าเป็น (i1,j1)

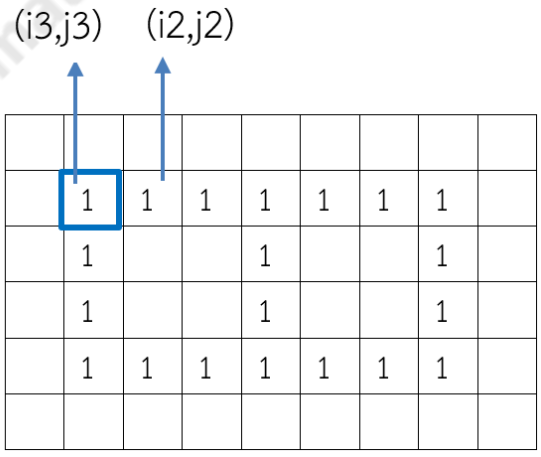
else if หากไม่พบ pixel ที่ไม่ใช่ 0

ให้ f_{ij} = -NBD



ภาพประกอบที่ 3.24 ตัวอย่างการหาเส้นขอบตามเข็มนาฬิกา

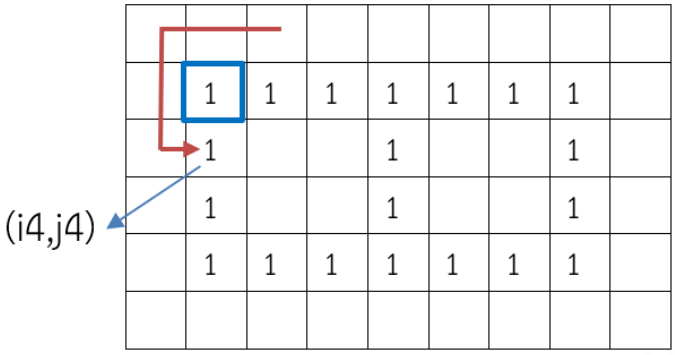
- จากนั้นกำหนดให้ (i2,j2) เท่ากับ (i1,j1) และให้ (i3,j3) เท่ากับ (i,j)



ภาพประกอบที่ 3.25 ตัวอย่างการกำหนดค่าให้ (i2,j2) เท่ากับ (i1,j1) และให้ (i3,j3) เท่ากับ (i,j)

- เริ่มหาเส้นขอบจากพิกเซลถัดไปของ (i2,j2) โดยจะดูแบบทวนเข็มนาฬิกาของพิกเซลในบริเวณใกล้เคียงของ (i3,j3) เพื่อหาพิกเซลที่ไม่ใช่ 0 แรก และตั้งค่าเป็น (i4,j4) ดังภาพประกอบที่ 3.26

Computer Science Department
Faculty of Informatics
Mahasarakham University



ภาพประกอบที่ 3.26 ตัวอย่างการหาเส้นขอบแบบทวนเข็มนาฬิกา

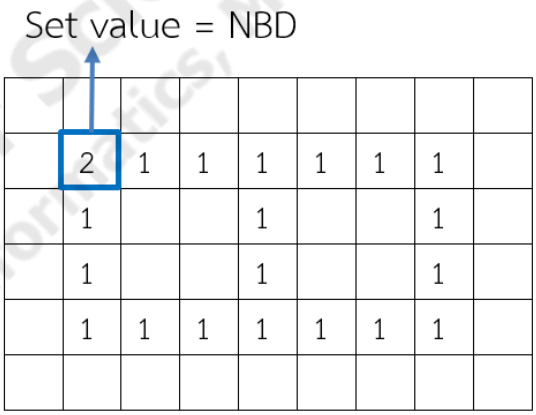
- เปลี่ยนค่าของพิกเซล (i3,j3) ตามเงื่อนต่อไปนี้

เงื่อนไขที่ 3 if เส้นขอบด้านนอกของวัตถุ (pixel (i3,j3+1) = 0)

ให้ตั้งค่าพิกเซลปัจจุบันเป็น -NBD

else if หาก pixel (i3,j3+1) != 0 และ ค่าพิกเซลปัจจุบัน = 1

ให้ตั้งค่าพิกเซลปัจจุบันเป็น NBD



ภาพประกอบที่ 3.27 ตัวอย่างการเปลี่ยนค่าพิกเซลตามเงื่อนไขที่ 3

เงื่อนไขที่ 4 if f_{ij} != 1

ให้ตั้งค่า NBD = $|f_{ij}|$ และเริ่มสแกนพิกเซลถัดไป (i,j+1)

โดยจะทำซ้ำขั้นตอนข้างต้นไปเรื่อยๆจนครบทุกพิกเซลและทำการหยุดเมื่อถึงมุมขวาล่างของภาพซึ่งจะแสดงผลลัพธ์ดังภาพประกอบที่ 3.28

Computer Science Department
Faculty of Informatics, Mahasarakham University

	2	2	2	2	2	2	-2	
	-3			-4			-2	
	-3			-4			-2	
	2	2	2	2	2	2	-2	

ภาพประกอบที่ 3.28 ภาพผลลัพธ์หลังการทำ Contour จนครบ

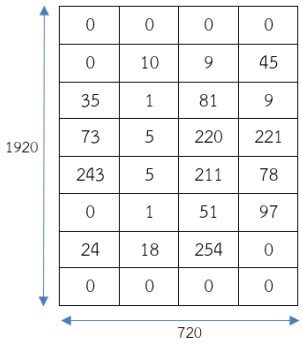
ภาพประกอบที่ 3.29 ตัวอย่างก่อนการทำ Contour

ภาพประกอบที่ 3.30 ตัวอย่างภาพผลลัพธ์หลังการทำ Contour

- **การหาพิกัดที่ต้องการตัด**

เป็นการหาค่าความกว้างและความสูงเพื่อระบุตำแหน่งที่จะเอามาตัด crop รูปภาพให้เหลือเฉพาะส่วนที่ต้องการ โดยมีขั้นตอนและวิธีดำเนินการ ดังนี้

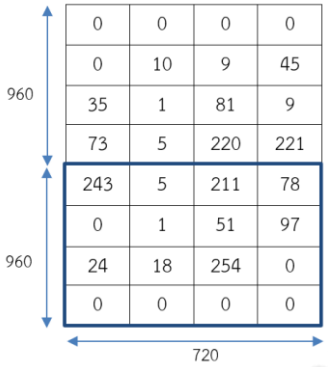
ขั้นตอนที่ 1 เมื่อ Input ภาพด้วยการใช้ shape จะได้ค่าความกว้างและความสูง ซึ่งจากภาพประกอบที่ 3.31 จะมีค่าความกว้างและความสูงคือ (720,1920)



ภาพประกอบที่ 3.31 ขั้นตอนการหาค่าความกว้างและความสูง

Computer Science Department
Faculty of Informatics Mahasarakham University

ขั้นตอนที่ 2 หลังจากได้ค่าความกว้างและความสูงแล้ว ให้นำความสูงมาหารสอง โดยนำ 1920 หาร ด้วย 2 จะได้ 960 แสดงดังภาพประกอบที่ 3.32



0	0	0	0
0	10	9	45
35	1	81	9
73	5	220	221
243	5	211	78
0	1	51	97
24	18	254	0
0	0	0	0

ภาพประกอบที่ 3.32 ขั้นตอนการนำค่าความสูงมาหารสอง

ขั้นตอนที่ 3 นำพิกัดที่ได้จากขั้นตอนที่ 1 – 2 มาใช้ในการตัดภาพต้นฉบับ

Table no.
23

ชาบู อู๊ด เป็นต่อ
มหาสารคาม

6/5 ซอย47 ถนนริมคลองสมถวิล ตำบลตลาด
อำเภอเมือง จังหวัดมหาสารคาม 44000

Tel: 0812559672

Receipt

Table: 23
Staff: Pimpavee Pimpavee
Guests: 2 ID: L27S9
Date: 19/12/20 Time: 21:23

1	จานแรกหมู	0.00
1	น้ำซุปน้ำดำญี่ปุ่น	0.00
1	น้ำซุปต้มยำน้ำข้น	0.00
2	Buffet หมูทะเล 249	498.00
2	น้ำรีฟิล	78.00

Items: 7
Subtotal: 576.00
Total: 576.00

Cash 576.00
Change 0.00

Powered by FoodStory

ภาพประกอบที่ 3.33 ตัวอย่างภาพใบเสร็จที่เป็น Input

Computer Science Department
Faculty of Informatics Mahasarakham University

Table: 23
Staff: Pimpavee Pimpavee
Guests: 2 ID: L27S9
Date: 19/12/20 Time: 21:23

1	จานแรกหมู	0.00
1	น้ำซุปน้ำดำญี่ปุ่น	0.00
1	น้ำซุปต้มยำน้ำข้น	0.00
2	Buffet หมูทะเล 249	498.00
2	น้ำรีฟิล	78.00

Items: 7
Subtotal: 576.00
Total: 576.00
Cash 576.00
Change 0.00

ภาพประกอบที่ 3.34 ตัวอย่างภาพใบเสร็จหลังจากนำพิกัดมาตัดภาพเอาเฉพาะส่วนที่ต้องการ

3.3 ขั้นตอนการเรียกใช้งาน Tesseract OCR

ในกระบวนการ Optical Character Recognition (OCR) นั้นจำเป็นจะต้องมีข้อมูลลักษณะ (Feature) ของตัวอักขระนั้นๆ ก่อน เพื่อนำมาประมวลผลเทียบเคียงกับข้อมูลที่ได้จากภาพ ข้อมูล Feature ที่ได้มาจากการฝึกฝน (Train) ซึ่งค่อนข้างมีความซับซ้อนในการพัฒนา ดังนั้นในโครงงานนี้จะใช้เครื่องมือที่ช่วยลดความยุ่งยาก ที่มีชื่อว่า Tesseract OCR เพื่อนำมาเปรียบเทียบผลลัพธ์ที่ได้

วิธีการดำเนินการ การสกัดตัวหนังสือภาษาไทยและภาษาอังกฤษ โดยในขั้นตอนแรกนำภาพใบเสร็จเข้าไป จากนั้นเขียน Code เพื่อทดสอบการทำงานและตรวจสอบผลลัพธ์

Table: 20
Staff: Pimpavee Pimpavee
Guests: 2 ID: 5C5XQ
Date: 10/01/21 Time: 19:42

2	Buffet หมู 199	398.00
2	น้ำรีฟิล	78.00
1	น้ำซุปชาบู	0.00
1	เตาปิ้งย่าง	0.00
1	จานแรกหมู	0.00

Items: 7
Subtotal: 476.00
Total: 476.00
Cash 476.00
Change 0.00
Powered by FoodStory

ภาพประกอบที่ 3.35 ตัวอย่างภาพใบเสร็จ

Computer Science Department
Faculty of Informatics Mahasarakham University

```
104
105   #Text from Image
106   data_Img = pytesseract.image_to_string(cropX, lang = 'Tha+Eng')
107   print(data_Img)
108
```

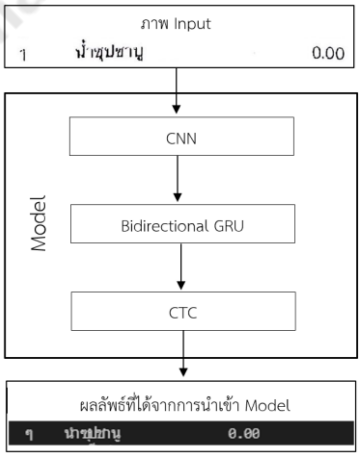
ภาพประกอบที่ 3.36 ตัวอย่าง Code และ การสกัดตัวอักษรภาษาไทยและภาษาอังกฤษ

โดยใช้ pytesseract สกัดข้อความจากภาพ ซึ่ง parameter lang= ‘tha+eng’ หมายถึง การใช้ Training Dataset ในการสกัดตัวอักษรไทยและอังกฤษจากภาพ จะพบว่าสามารถสกัดข้อความออกมาได้ถูกต้องบางคำ

```
Table: 20
Staff: Pimpavee Pimpavee
Guests: 2              ID: 5COXQ
Date: 10/01/21           lime: 19:42
2       Buffet หมู 199                  398.00
2    ทาร์ทิล                78.00
ๆ    น้ำซุปกบู              0.00
1    เตาปิ้งย่าง             0.00
4    จานแรกหมู              0.00
Items: 7                     /
Subtotal:                  476.00
Total       ave nts
Cash                     476.00
Change                     0.00
ջ
```

ภาพประกอบที่ 3.37 ผลลัพธ์หลังจากทดลองใช้งาน Tesseract

3.4 การรู้จำข้อความ (Text Recognition)



ภาพประกอบที่ 3.38 ขั้นตอนการทำงานของกระบวนการรู้จำตัวอักษร

เมื่อผ่านกระบวนการ Pre-Processing แล้ว ผลลัพธ์ที่ได้จากขั้นตอนดังกล่าวจะถูกส่งเข้าสู่กระบวนการรู้จำตัวอักษร โดยจะมีรายละเอียดการทำงาน ดังนี้

Computer Science Department
Faculty of Informatics, Mahasarakham University

3.5 การเรียนรู้เชิงลึก (Deep Larning)

(1) โครงข่ายประสาทเทียมแบบคอนโวลูชัน Convolution Neural Network (CNN)

- ชั้น Convolution เป็นขั้นตอนในการสกัดคุณลักษณะของภาพ โดยการทำงานของ Convolution จะทำการ Sliding Windows (Filter) มาสแกนเพื่อทำการแยกองค์ประกอบของภาพ เช่น สี หรือรูปร่าง

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

ภาพประกอบที่ 3.39 ภาพต้นฉบับก่อนทำ Convolution ขนาด 5 x 5

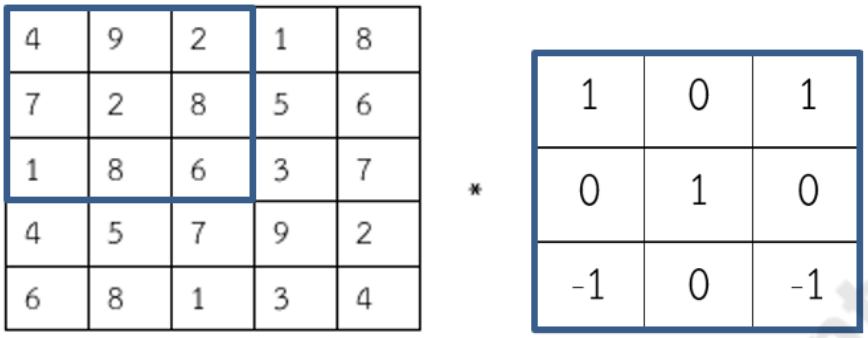
1	0	1
0	1	0
-1	0	-1

ภาพประกอบที่ 3.40 Filter สำหรับการทำ Convolution ขนาด 3 x 3

วิธีการคำนวณ

รอบที่ 1 คำนวณหาค่าใหม่ โดยเอาค่าในภาพมาคูณกับค่าใน Filter ที่ตำแหน่งตรงกัน ทำจนครบขนาดของ Filter แล้วหาผลรวมของทุกตำแหน่ง (ตำแหน่งที่ 1 + ... + ตำแหน่งที่ 4) และแทนค่าในตำแหน่งที่ 1

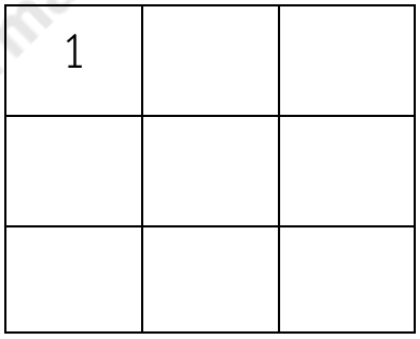
Computer Science Department
Faculty of Informatics, Mahasarakham University



ภาพประกอบที่ 3.41 การทำ Convolution ในรอบแรก

- การคำนวณผลลัพธ์ของการ Stride ในรอบที่ 1

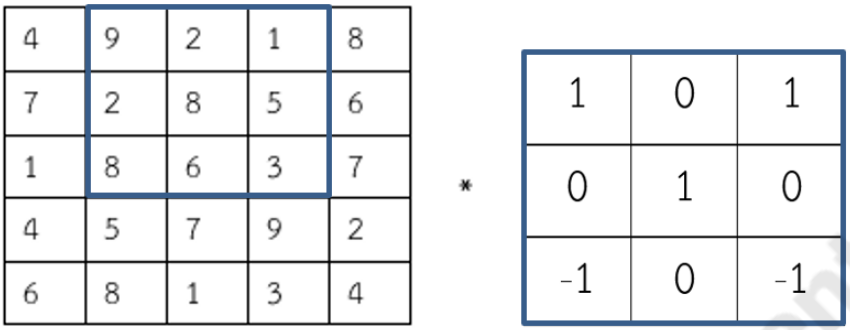
ตำแหน่งที่ 1 = 4 x 1 = 4
ตำแหน่งที่ 2 = 9 x 0 = 0
ตำแหน่งที่ 3 = 2 x 1 = 2
ตำแหน่งที่ 4 = 7 x 0 = 0
ตำแหน่งที่ 5 = 2 x 1 = 2
ตำแหน่งที่ 6 = 8 x 0 = 0
ตำแหน่งที่ 7 = 1 x (-1) = -1
ตำแหน่งที่ 8 = 8 x 0 = 0
ตำแหน่งที่ 9 = 6 x (-1) = -6
ผลรวมของทุกตำแหน่ง = 4+0+2+0+2+0+(-1)+0+(-6) = 1



ภาพประกอบที่ 3.42 ค่าใหม่หลังจากการ Stride ในรอบที่ 1

รอบที่ 2 คำนวณหาค่าใหม่โดยเอาค่าในภาพมาคูณกับค่าใน Filter ที่ตำแหน่งตรงกัน ทำจนครบขนาดของ Filter แล้วหาผลรวมของทุกตำแหน่ง (ตำแหน่งที่ 1 + ... + ตำแหน่งที่ 4) และแทนค่าในตำแหน่งที่ 1

Computer Science Department
Faculty of Informatics, Mahasarakham University



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

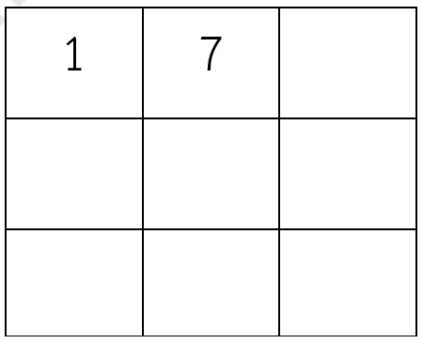
*

1	0	1
0	1	0
-1	0	-1

ภาพประกอบที่ 3.43 การทำ Convolution ในรอบที่ 2

- การคำนวณผลลัพธ์ของการ Stride ในรอบที่ 2

ตำแหน่งที่ 1 = 9 x 1 = 9
ตำแหน่งที่ 2 = 2 x 0 = 0
ตำแหน่งที่ 3 = 1 x 1 = 1
ตำแหน่งที่ 4 = 2 x 0 = 0
ตำแหน่งที่ 5 = 8 x 1 = 8
ตำแหน่งที่ 6 = 5 x 0 = 0
ตำแหน่งที่ 7 = 8 x (-1) = -8
ตำแหน่งที่ 8 = 6 x 0 = 0
ตำแหน่งที่ 9 = 3 x (-1) = -3
ผลรวมของทุกตำแหน่ง = 9+0+1+0+8+0+(-8)+0+(-3) = 7



1	7	

ภาพประกอบที่ 3.44 ค่าใหม่หลังจากการ Stride ในรอบที่ 2

Computer Science Department
Faculty of Informatics, Mahasarakham University

1	7	2
12	-1	8
5	7	17

ภาพประกอบที่ 3.45 ผลลัพธ์จากการทำ Convolution จนครบทุกช่อง

- **ชั้นพูล (Pooling layer)** หลังจากที่ทำ Convolution เสร็จแล้ว จะทำการลดขนาดของข้อมูล แต่รายละเอียด Input ยังคงเดิม ส่งผลให้ความเร็วในการคำนวณ ซึ่งจะใช้ Max Pooling ในการคำนวณ

วิธีดำเนินการ จะทำการหนดค่า Stride = 1 Padding = 1 และ Filter ขนาด 2 x 2 โดย Max Pooling เป็นการนำค่ามากที่สุดที่อยู่ในบริเวณของ Filter มาเป็นค่าของรูปแบบใหม่ จากนั้นขยับ Filter ตามค่า Stride ดังตัวอย่าง

4	9	2	1	8	0
7	6	17	7	7	8
1	15	8	11	12	6
4	6	15	15	5	7
6	12	6	10	13	2
0	6	8	1	3	4

ภาพประกอบที่ 3.46 ภาพก่อนการทำ Max Pooling

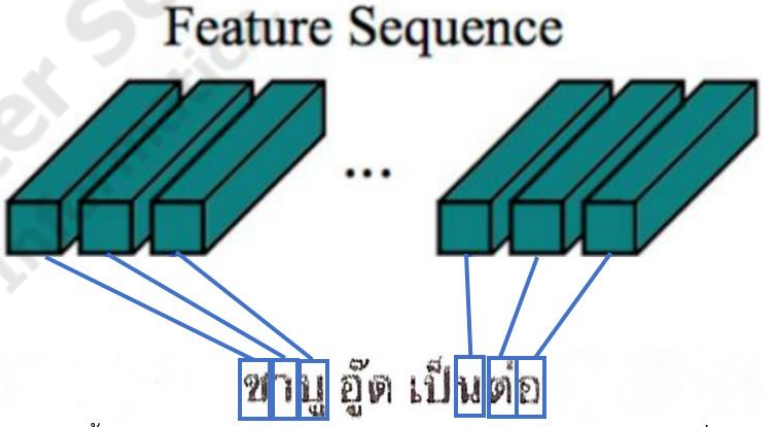
Computer Science Department
Faculty of Informatics, Mahasarakham University

9	17	17	8	8
15	17	17	12	12
15	15	15	15	12
12	15	15	15	13
12	12	10	13	13

ภาพประกอบที่ 3.47 ภาพหลังการทำ Max Pooling

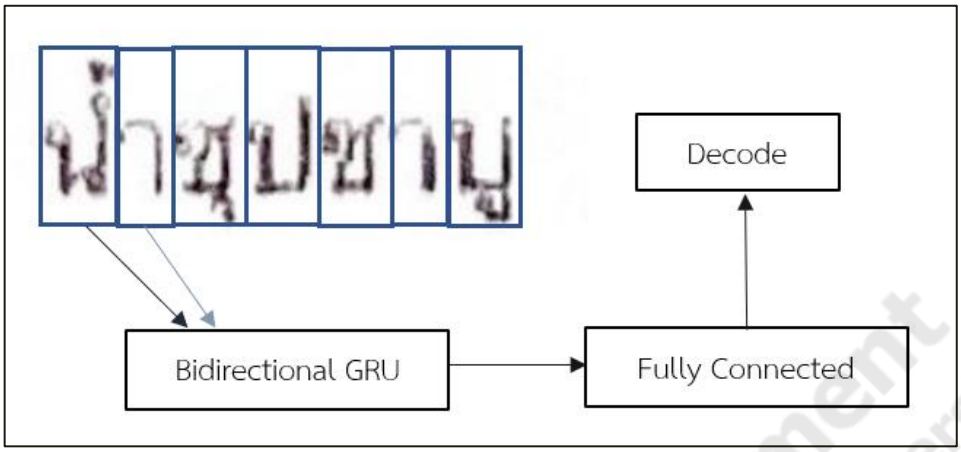
- **Dropout** เป็นขั้นตอนในการลดจำนวน Node ออกตามจำนวน Dropout ที่กำหนด เพื่อทำให้ Network มีความยากขึ้น เพื่อลดการเกิด Overfitting ของข้อมูล โดยค่าที่ใช้กำหนด Dropout เช่น 0.5 หมายความว่า 50% ของจำนวน Node ทั้งหมด จะไม่มีการ Update ค่า Weight ซึ่งค่าที่ใช้ใน Dropout เกิดจากการสุ่มความน่าจะเป็นขึ้นมา

(2) **Reshape** เป็นการแปลงรูปภาพ โดยจะส่งข้อมูลเข้าทีละ Time-Step เอามาเรียงต่อกันเป็นแถวยาว จากนั้นส่งข้อมูลให้ Bidirectional GRU เพื่อทำการประมวลผลในขั้นตอนต่อไป



ภาพประกอบที่ 3.48 ตัวอย่าง Time-Step ของข้อมูล

(3) **Bidirectional GRU** จากการ Reshape ผลลัพธ์ที่ได้จะโดนส่งข้อมูลเข้า GRU ทีละ Time-Step โดยจะเข้าสู่การทำงานตาม Alogorithm GRU จนครบทุก Time-Step



ภาพประกอบที่ 3.49 ตัวอย่างการ Feed ข้อมูลเข้าทีละ Time Step

ตัวอย่างการทำงานของ GRU โดยจะกำหนดตัวแปรแต่ละตัว หมายถึง

x_t คือ ค่า Input Data ที่เวลา t

U_z คือ Transition Matrix

b_z คือ Bias unit สำหรับ Update Gate และ Reset Gate ตามลำดับ

w_z คือ Weight Matrix ของ Hidden Layer

$h_{(t-1)}$ คือ Hidden State ที่เวลา t

ขั้นตอนที่ 1 Input “วรรณ”

กำหนด Character Dictionary : {‘ว:0’, ‘ร:1’, ‘ณ:2’,}

Parameter : Output(O) = 4

Batch size(B) = 2

Sequence Length(S) = 2

ขั้นตอนที่ 2 แปลงอักขระให้เป็น Sequence และ Batch size ตามจำนวนที่กำหนด

“วรรณ” = [0, 1, 1 ,2]

ขั้นตอนที่ 3 สร้าง mini batch และ Transpose

$[[0,1],[1,2]]$ จะได้ $\left(\begin{bmatrix}0\\1\end{bmatrix}\begin{bmatrix}1\\2\end{bmatrix}\right)$

ขั้นตอนที่ 4 สร้าง One-Hot Encode

One-Hot Encode คือ การเข้ารหัสข้อมูลให้อยู่ในรูป Binary โดยถ้าสนใจตัวนั้น ให้ตัวอักษรนั้นเป็น 1 นอกนั้นเป็น 0

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตารางที่ 3.1 ตารางการสร้าง One-Hot Encode

One – Hot Encode				
Number and Character Dictionary		0	1	2
		ว	ร	ณ
0	ว	1	0	0
1	ร	0	1	0
1	ร	0	1	0
2	ณ	0	0	1

- นำ Array ที่ทำการ Transpose มาแปลงให้เป็น One-Hot Encode

$$x_i = \left[\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}, \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\right]$$

ขั้นตอนที่ 5 กำหนดค่าเริ่มต้นให้

$$w_2 = \begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}, U_2 = \begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}, b_z = [0 \quad 0], e \approx 2.71828$$

ในการทำงานจะมีขั้นตอนของการทำ Forward และ Backward โดยจะยกตัวอย่างการทำงานดังนี้

$Forward = 0$

Z_t คือ Update Gate

$$z_t = \sigma\left(w_z x_t + U_z h_{(t-1)} + b_z\right) \tag{3.9}$$

$$Sigmoid\ \sigma(a) = \frac{1}{1+e^{-a}} \tag{3.10}$$

- Batch 1 Sequence 1

$$z_0 = \sigma\left(\begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix} + \begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + [0 \quad 0]\right)$$

$$= \begin{bmatrix}0.6251 & 0.462\\ 0.6137 & 0.5858\end{bmatrix}$$

- Batch 1 Sequence 2

$$z_0 = \sigma\left(\begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} + \begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + [0 \quad 0]\right)$$

$$= \begin{bmatrix}0.5 & 0.6251\\ 0.5 & 0.6137\end{bmatrix}$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

จะได้ $Z = \left(\begin{bmatrix}0.6251 & 0.462\\ 0.6137 & 0.5858\end{bmatrix}\begin{bmatrix}0.5 & 0.6251\\ 0.5 & 0.6137\end{bmatrix}\right) = \begin{bmatrix}1.1251 & 1.0871\\ 1.1137 & 1.1995\end{bmatrix}$

r_t คือ Reset Gate

$$r_t = \sigma\left(w_r x_t + U_r h_{(t-1)} + b_r\right) \tag{3.11}$$

- Batch 1 Sequence 1

$$r_0 = \sigma\left(\begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix} + \begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + [0 \quad 0]\right)$$

$$= \begin{bmatrix}0.6251 & 0.462\\ 0.6137 & 0.5858\end{bmatrix}$$

- Batch 1 Sequence 2

$$r_0 = \sigma\left(\begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} + \begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + [0 \quad 0]\right)$$

$$= \begin{bmatrix}0.5 & 0.6251\\ 0.5 & 0.6137\end{bmatrix}$$

จะได้ $r = \left(\begin{bmatrix}0.6251 & 0.462\\ 0.6137 & 0.5858\end{bmatrix}\begin{bmatrix}0.5 & 0.6251\\ 0.5 & 0.6137\end{bmatrix}\right) = \begin{bmatrix}1.1251 & 1.0871\\ 1.1137 & 1.1995\end{bmatrix}$

$\tilde{h}$ คือ State การทำงาน (Intermediate Memory)

$$\tilde{h} = tanh\left(w_h x_t + r * U_h h_{(t-1)} + b_z\right) \tag{3.12}$$

$$tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{3.13}$$

- Batch 1 Sequence 1

$$\tilde{h} = tanh\left(\begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}\right.$$

$$\left. + \begin{bmatrix}1.1251 & 1.0871\\ 1.1137 & 1.1995\end{bmatrix}\begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + [0 \quad 0]\right)$$

$$= \begin{bmatrix}0.4709 & -0.1511\\ 0.4323 & 0.3335\end{bmatrix}$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

- Batch 1 Sequence 2

$$\tilde{h} = tanh\left(\begin{bmatrix}0.5112 & -0.1523\\ 0.4627 & 0.3468\\ 0.3452 & 0.0196\end{bmatrix}\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} + \begin{bmatrix}1.1251 & 1.0871\\ 1.1137 & 1.1995\end{bmatrix}\begin{bmatrix}1.2455 & 0.5413\\ 0.7224 & -0.0235\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + [0 \quad 0]\right)$$

$$= \begin{bmatrix}0 & 0.4709\\ 0 & 0.4323\end{bmatrix}$$

จะได้ $\tilde{h} = \left(\begin{bmatrix}0.4709 & -0.1511\\ 0.4323 & 0.3335\end{bmatrix}\begin{bmatrix}0 & 0.4709\\ 0 & 0.4323\end{bmatrix}\right) = \begin{bmatrix}0.4709 & 0.3198\\ 0.4323 & 0.7658\end{bmatrix}$

จะได้ Output(h)

$$h = z * h_{(t-1)} + (1-z)\tilde{h} \tag{3.14}$$

$$h = \begin{bmatrix}1.1251 & 1.0871\\ 1.1137 & 1.1995\end{bmatrix}\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix} + \left(1 - \begin{bmatrix}1.1251 & 1.0871\\ 1.1137 & 1.1995\end{bmatrix}\right)\begin{bmatrix}0.4709 & 0.3198\\ 0.4323 & 0.7658\end{bmatrix}$$

$$= \begin{bmatrix}-0.0953 & -0.1048\\ -0.1412 & -0.1904\end{bmatrix}$$

Backward

ในการทำ Backward หรือการทำ Back-propagation เป็นการคำนวณย้อนกลับเพื่อทำการปรับค่าน้ำหนัก ส่งผลให้ Model มีการเรียนรู้ข้อมูลได้ถูกต้องแม่นยำมากยิ่งขึ้น โดยก่อนการทำ Backward ต้องผ่านการทำ Forward ก่อน เพื่อคำนวณหาผลลัพธ์และจะคำนวณย้อนกลับเพื่อหาค่าน้ำหนักใหม่ที่เหมาะสมที่สุด

(4) ชั้นฟูลลีคอนเนก (Fully connected layer)

เป็นชั้นสุดท้ายของ CNN ที่ได้ทำการตัดออกแล้วต่อเข้ากับ Bidirectional GRU เพื่อทำนายผลว่าในขั้นตอนก่อนหน้าควรเป็นอักษรตัวใด โดยมีกระบวนการ 2 กระบวนการ ดังนี้

- **Soft Max Function** เป็นการหาระยะห่างระหว่างคำตอบกับผลเฉลย โดยตัวเลขที่ได้มากจาก Function Soft Max จะผ่านการทำ Normalization ให้ค่าอยู่ระหว่าง 0 – 1 ดังสมการ

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_{i=1}^{k} x^{x_j}} f_{Or}\ j = 1, \dots, k \tag{3.15}$$

โดยที่ x_j คือ ค่าที่ Input

e^{x_j} คือ ค่าที่ได้จากการนำ e มายกกำลังด้วย x_j

ตัวอย่างการคำนวณ x_j = (5.0,3.0,8.0)

ขั้นตอนที่ 1 หาค่า e^{x_j} แต่ละค่า $e^{5.0} = 2.718^{5.0}$ = 148.3362

$e^{3.0} = 2.718^{3.0}$ = 20.0793

$e^{8.0} = 2.718^{8.0}$ = 2,978.4864

ขั้นตอนที่ 2 หาค่า $\sum_{i=1}^{k} e^{x_j}$ = 148.3362+20.0793+2,978.4864

Computer Science Department
Faculty of Informatics, Mahasarakham University

= 3,146.9019

ขั้นตอนที่ 3 หาค่า Soft Max โดยแทนค่าในสมการ จะได้

$$\sigma(5.0) = \frac{148.3362}{3{,}146.9019} = 0.0471$$

$$\sigma(3.0) = \frac{20.0793}{3{,}146.9019} = 0.0064$$

$$\sigma(8.0) = \frac{2{,}978.4864}{3{,}146.9019} = 0.9465$$

ซึ่งเมื่อนำค่า Soft Max ทุกค่ามารวมกันจะได้ผลลัพธ์เท่ากับ 1

- **Cross Entropy** เป็นฟังก์ชันของการตัดสินใจว่าภาพนี้ควรจะเป็นตัวอักษรใด โดยหากค่าที่ผ่านการ Soft Max และค่าจริงที่แสดงว่าอักษรนั้นอยู่ Class Lable ไหน มีค่าเป็น 1 ตรงกัน คอมพิวเตอร์จะทำนายว่าข้อมูลนั้นถูกต้อง และจะทำนายผลออกมาซึ่งสามารถหาได้ดังสมาการ

$$\text{Cross Entropy} = -\sum Y_i' * log(Y_i) \quad (3.16)$$

โดยที่ Y_i' คือ ค่าจริงที่แสดงว่าอักษรนั้นอยู่ Class Label ไหน

Y_i คือ ค่าที่ผ่าน Function Soft Max

(5) Connectionist Temporal Classification (CTC)

เป็นขั้นตอนในการจัดเรียงตัวอักษรให้เป็นคำที่มีลักษณะสวยงาม โดยตัดช่องว่างและการตัดอักขระที่มีลักษณะติดกัน จะทำการตัดตัวที่ซ้ำออก โดยคำนวณได้จากสมการ

$$-\sum_{(x,z)\in S} log_{eP}\,(z|x) \quad (3.17)$$

โดยที่ x คือ Training Simple

S คือ Training Data

z คือ Generated Sequence

	Time - Step					
	t1	t2	t3	t4	t5	t6
ก	0	0	0	0	0	0
...	...	...	...	...	...	...
ณ	0	0	0	0.3	0.6	0.7
...	...	...	...	...	...	...
ม	0	0	0	0	0	0
...	...	...	...	...	...	...
ร	0	0.6	0.7	0.8	0.7	0
ล	0	0	0	0	0	0
ว	0.2	0.8	0.3	0	0	0.9
...	...	...	...	...	...	...
ฮ	0	0	0	0	0	0
สระ	0	0	0	0	0	0
...	...	...	...	...	...	...
-(Black)	0.1	0.2	0.4	0.3	0.1	0.2

ภาพประกอบที่ 3.50 กราฟแสดงความน่าจะเป็นของอักขระทีละ Time-Step

Computer Science Department
Faculty of Informatics Mahasarakham University

โอกาสที่จะเป็นคำว่า “วรรณ” จากข้อมูลที่มี 6 Time – Step ที่ต่างกัน จะได้

กราฟเส้นสีน้ำเงิน ว-รรรณ = 0.2 x 0.2 x 0.7 x 0.8 x 0.7 x 0.7

= 0.010976

กราฟเส้นสีแดง วรร-รณ = 0.2 x 0.6 x 0.7 x 0.3 x 0.7 x 0.7

= 0.012348

นำค่าที่คำนวณความเป็นไปได้ข้างต้นมารวมกัน จะได้ 0.010976+0.012348 = 0.023324 นำไปแทนค่าในสูตร เพื่อหาค่า loss

จากสูตร $loss = -log_e(\text{ผลรวมของความน่าจะเป็น})$

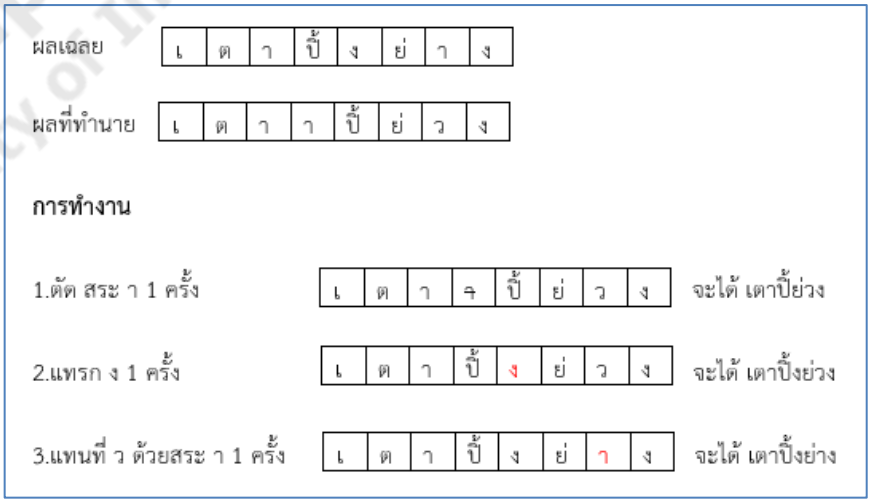
ดังนั้น $loss = -log_e(0.023324) = 3.7583$

3.6 การวัดประสิทธิภาพ

การประเมินความถูกต้องของการรู้จำตัวอักษร โดยใช้ Levenshtein Edit Distance [11] ซึ่งเป็นขั้นตอนการวัดเพื่อหาค่าความแตกต่างของข้อความทั้งสองชุดระหว่างชุดแรกที่เป็นผลเฉลย และชุดที่สองที่เป็นชุดผลการทำนาย โดยจะนับจำนวนครั้งของการแทรก การตัด และการแทนที่ เช่น ผลเฉลยดังภาพประกอบที่ 3.51 และ ภาพประกอบที่ 3.52

ตัวอย่าง “เตาปิ้งย่าง” ผลการทำนายได้ “เตาาปี้ย่วง” จะวัดค่า Character Error Rate (CER) ได้ดังสมการ

$$CER = \frac{(\text{ผลรวมของจำนวนที่ ตัดออก แทรก และแทนที่})}{(\text{จำนวนของอักขระที่ถูกทั้งหมด})} \times 100 \qquad (3.18)$$



ภาพประกอบที่ 3.51 ตัวอย่างการทำงานของ Levenshtein Edit Distance ของคำว่า เตาปิ้งย่าง

Computer Science Department
Faculty of Informatics Mahasarakham University

ตัวอย่างการหา Character Error Rate (CER) ของคำว่า “เตาปิ้งย่าง”

จากสูตร $CER = \frac{(\text{ผลรวมของจำนวนที่ ตัดออก แทรก และแทนที่})}{(\text{จำนวนของอักขระที่ถูกทั้งหมด})} \times 100$

แทนค่า $CER = \frac{(1+1+1)}{(6)} \times 100$

จะได้ $CER = 50\%$

ดังนั้น Error rate ของคำว่า “เตาปิ้งย่าง” มีค่าผิดพลาดอยู่ที่ 50%

ผลเฉลย	จ	า	น	แ	ร	ก	ห	มู
ผลที่ทำนาย	จ	า	า	แ	ร	ก	ห	มู

การทำงาน

1.ตัด สระ า 1 ครั้ง	จ	า	ๅ	แ	ร	ก	ห	มู	จะได้ จาแรกหมู
2.แทรก น 1 ครั้ง	จ	า	น	แ	ร	ก	ห	มู	จะได้ จานแรกหมู

ภาพประกอบที่ 3.52 ตัวอย่างการทำงานของ Levenshtein Edit Distance ของคำว่า จานแรกหมู

ตัวอย่างการหา Character Error Rate (CER) ของคำว่า “จานแรกหมู”

จากสูตร $CER = \frac{(\text{ผลรวมของจำนวนที่ ตัดออก แทรก และแทนที่})}{(\text{จำนวนของอักขระที่ถูกทั้งหมด})} \times 100$

แทนค่า $CER = \frac{(1+1)}{(7)} \times 100$

จะได้ $CER = 28.57\%$

ดังนั้น Error rate ของคำว่า “เตาปิ้งย่าง” มีค่าผิดพลาดอยู่ที่ 28.57%

3.7 ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลที่ใช้ในการสร้างโมเดลรู้จำ 500 ภาพ จะแบ่งออกเป็นชุดสอน 80% ชุดวาลิเดท 10% และชุดทดสอบ 10% ซึ่งเป็นใบเสร็จของร้านชาบู อู๊ด เป็นต่อ (ภาพประกอบที่ 3.53) และนำมาตัดเป็นข้อมูล 1 บรรทัด (ภาพประกอบที่ 3.54) จากนั้นนำมาตัดเป็นคำ (ภาพประกอบที่ 3.55)

Computer Science Department
Faculty of Informatics Mahasarakham University

Table no.

31

ชาบู อู๊ด เป็นต่อ

มหาสารคาม

6/5 ซอย47 ถนนริมคลองสมถวิล ตำบลตลาด
อำเภอเมือง จังหวัดมหาสารคาม 44000

Tel: 0812559572

Invoice

Table: 31
Staff: Pimpavee Pimpavee
Guests: 7
Date: 03/12/20 Time: 21:19

2	จานแรกหมู	0.00
7	Buffet หมู 199	1,393.00
7	น้ำริฟิล	273.00
1	น้ำซุปชาบู	0.00
2	น้ำ ต้มยำน้ำใส	0.00
1	เตาปิ้งย่าง	0.00

Items: 20

Subtotal: 1,666.00

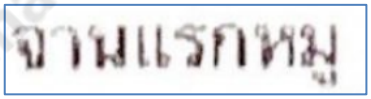
Total: 1,666.00

Powered by FoodStory

ภาพประกอบที่ 3.53 ตัวอย่างข้อมูลใบเสร็จของร้านชา บู อู๊ดเป็นต่อ (ชุดข้อมูล 1 ภาพ)

2 จานแรกหมู 0.00

ภาพประกอบที่ 3.54 ตัวอย่างชุดข้อมูลใบเสร็จของร้านชาบู อู๊ด เป็นต่อ (ชุดข้อมูล 1 บรรทัด)



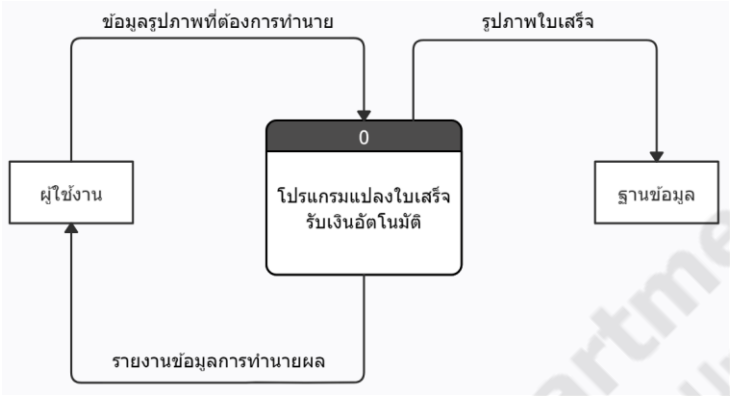
จานแรกหมู

ภาพประกอบที่ 3.55 ตัวอย่างชุดข้อมูลใบเสร็จของร้านชาบู อู๊ด เป็นต่อ (ชุดข้อมูล 1 คำ)

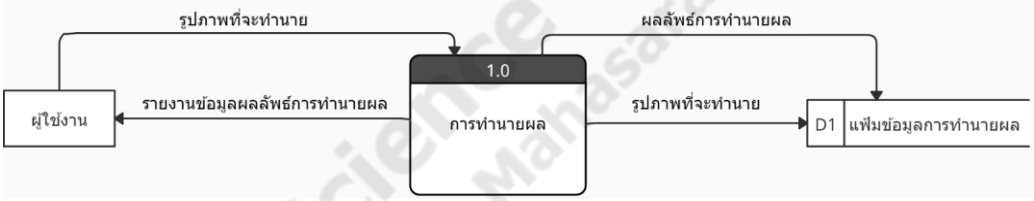
Computer Science Department
Faculty of Informatics, Mahasarakham University

3.8 ผลการออกแบบและพัฒนาระบบ

3.8.1 Context Diagram



ภาพประกอบที่ 3.56 Context Diagram



ภาพประกอบที่ 3.57 ภาพแผนภาพกระแสข้อมูล ระดับที่1 (Data Flow Diagram Level)

3.8.2 External Entity Description

ตารางที่ 3.2 ตารางแสดง External Entity Description

Name	Description	Input Data Flow	Output Data Flow
ผู้ใช้งาน	ผู้ใช้บริการของระบบสามารถส่งรูปภาพเพื่อเข้ามาประมวลผลได้	- รูปภาพที่จะทำนาย	- รายงานผลลัพธ์การทำนายผล

3.8.3 Data Store Description

ตารางที่ 3.3 ตารางแสดง Data Store Description

ID	Name	Description	Data Structure
D1	แฟ้มข้อมูลการทำนายผล	ฐานข้อมูลสำหรับเก็บข้อมูลในการแปลงใบเสร็จรับเงินอัตโนมัติ	รหัสการทำนายผล + ชื่อไฟล์รูปภาพที่ทำนาย + วันที่ทำนาย + เวลาทำนาย + ผลของการทำนาย

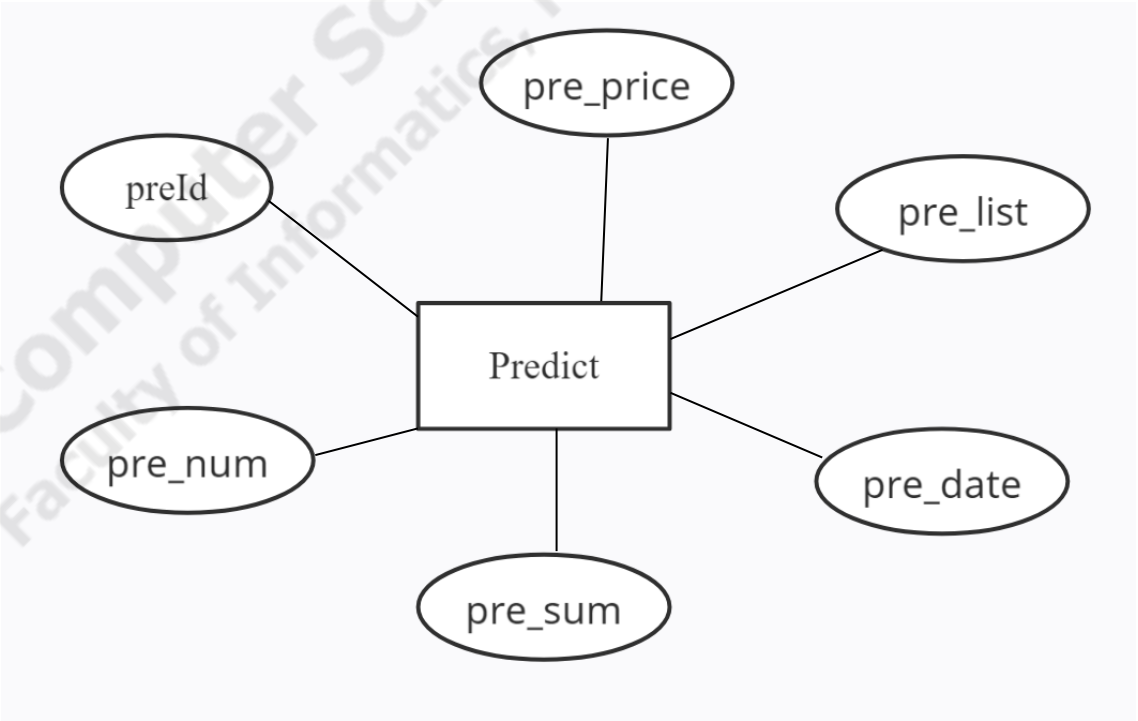
Computer Science Department
Faculty of Informatics, Mahasarakham University

3.8.4 Data Structure Description

ตารางที่ 3.4 ตารางแสดง Data Structure Description

Name	Description	Source	Destination	Data Structure
ข้อมูลรูปภาพที่ต้องการทำนาย	ไฟล์รูปภาพที่ผู้ใช้ส่งเข้ามาประมวลผล	ผู้ใช้งาน	Process 1.0 การทำนายผล	รหัสรูปภาพ + ชื่อไฟล์ + วันที่ทำนาย + เวลาทำนาย
		Process 1.0 การทำนายผล	D1 แฟ้มข้อมูลการทำนายผล	
รายงานข้อมูลการทำนายผล	ผลลัพธ์ข้อความที่ผ่านการประมวลผล	ผู้ใช้งาน	Process 1.0 การทำนายผล	รหัสการทำนายผล + ข้อความผลลัพธ์ + รหัสรูปภาพ + วันที่ทำนาย + เวลาทำนาย
		Process 1.0 การทำนายผล	D1 แฟ้มข้อมูลการทำนายผล	

3.8.5 ER Diagram



ภาพประกอบที่ 3.58 แผนผัง ER Diagram

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตารางที่ 3.5 ตารางแสดงข้อมูลการทำนายผล (Predict)

Attirbute Name	Type	Size	Description	Key type	References
preId	Int	11	รหัสการทำนายผล	primary key	10
pre_price	float	200	ชื่อไฟล์รูปภาพ		Imagshabu.jpg
pre_list	varchar	500	รายการที่ซื้อ		2 Buffet เนื้อ ทะเล 299
pre_date	Time stamp		วัน เวลาของการซื้อบนใบเสร็จ		27/06/2020 17:55
pre_sum	float	10	จำนวนเงินทั้งหมด		598.00
pre_num	float	10	จำนวนเงินต่อรายการ		598.00

Computer Science Department
Faculty of Informatics, Mahasarakham University