การจำตัวอักษรแบบตัวพิมพ์โดยใช้นิวรอลเน็ท

โดย

รศ.ดร. กิตติ ไพฑูรย์วัฒนกิจ

นายสาธิต อินทจักร์

นายยุทธนา สวนสุข

นายสุลักษณ์ สุ่มมาตย์

น.ส.จิราภร หมายพึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการวิจัย มรดก. ปีงบประมาณ 2538

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การจำตัวอักษรแบบตัวพิมพ์โดยใช้นิวรอลเน็ท

งานวิจัยนี้นำนิวรอลเน็ทเวิร์คมาประยุกต์ใช้งานเพื่อจำอักษรไทยแบบตัวพิมพ์ นิวรอลเน็ทที่ใช้มีโทโปรโลยี่เป็นแบบ multi-layer perceptron โดยอัลกอริทึมที่ใช้ในการฝึกสอนคือ Back Propagation อินพุทของนิวรอลเน็ทจะเป็นแบบ feature-based แทนที่จะใช้แบบ pixel-based โดยคุณลักษณะที่ใช้นั้นคือ moment ส่วน Input-layer จะมีทั้งหมด 7 โหนด ส่วน Hidden-layer มี 2 ชั้น โดยใช้ 22 โหนดในแต่ละชั้น และส่วน Output-layer มี 76 โหนด ปัญหาระหว่างการสอนก็คือจะติด local minimum ดังนั้น จึงอาศัยเทคนิคในการปรับ learing rate, momentum และค่า bias ระหว่างการฝึกสอนแทนที่จะกำหนดค่าคงที่ตลอดเหมือนแบบวิธีทั่ว ๆ ไป จากการทดสอบพบว่า ผลลัพธ์ในการจำตัวอักษรเป็นที่น่าพอใจ แต่ส่วนสำคัญที่สุดที่ต้องดำเนินการต่อไปคือ ขั้นตอน Preprocessing เพื่อใช้แยกตัวอักษรแต่ละตัวจากเครื่อง สแกนเนอร์ มาป้อนให้ นิวรอลเน็ทซึ่งส่วนนี้จะมีผลต่อผลลัพธ์โดยรวมต่อการจำและการนำไปใช้งาน

Characters Recognition by Neural Net

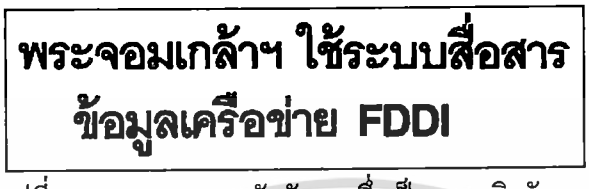
This research applies the neural net for printed Thai characters recognition. The topology of the net is multi-layer perceptron by using back propagation algorithm for the training. Instead of traditional pixel-based method, we propose the moment as feature-based. The configuration of the network has 7 nodes for input layer, two-hidden layer with 22 nodes each and 76 nodes for the output layer. To avoid the trapping with local minimum values, the learing rate, the momentum and the bias value are adapted during the learning process. The satisfied results have been attained. The further crucial preprocssing stage, however, has to be achieved for extract each character from the scanner and feeded its to the neural network.

RCH
TK
7882
.P3
ก7446
เลขหมู่.....
เลขทะเบียน.....32254
วัน, เดือน, ปี.....11 ธ.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. บทนำ

การรู้จักตัวอักษร (Character Recognition) ที่เป็นข้อมูลภาพเชิงตัวเลข[1] (Digital Image) ซึ่งได้มาจากการสแกนเนอร์ (ดังแสดงในรูปที่ 1.) ที่ใช้สแกนตัวอักษรเข้ามา แล้วใช้คอมพิวเตอร์ทำการแปลงอักษรจากรูปภาพแต่ละรูปให้เป็นตัวอักษรในรหัส ASCII ซึ่งกระบวนการนี้เราเรียกว่า OCR (Optical Character Recognition)



รูปที่ 1. แสดงภาพของตัวอักษร ซึ่งเป็นภาพเชิงตัวเลข

ปัจจุบัน OCR ได้มีการนำไปใช้ประโยชน์กันอย่างกว้างขวาง ที่เห็นได้ชัดเจนที่สุดก็ได้แก่ ระบบการจัดเก็บเอกสาร เนื่องจาก OCR จะลดขนาดของข้อมูลลงได้อย่างมาก เพราะว่าการจัดเก็บเอกสารในลักษณะของรูปภาพ จุดภาพ (Pixel) หนึ่งจุดจะเก็บเป็นหนึ่งไบต์ ไม่ว่าจะเป็นภาพขาวดำ(Binary Images) หรือภาพระดับสีเทา(Grey-Scale Images) ถ้าภาพที่เก็บต้องการความคมชัด ดังนั้นภาพจะต้องมีความละเอียดสูง เช่นตัวอักษรหนึ่งตัวมีขนาด 20x8 จุด จะต้องใช้เนื้อที่ในการเก็บเท่ากับ 160 ไบต์ แต่เมื่อนำรูปภาพมาผ่านกระบวนการ OCR ตัวอักษรหนึ่งตัว จะใช้เนื้อที่เก็บเพียงหนึ่งไบต์ สำหรับประโยชน์ในด้านอื่นๆ คือ มีความสะดวกในการจัดเก็บ, การค้นหา เนื่องจากข้อมูลที่เป็นไฟล์ตัวอักษร (Text File) จะมีความสะดวกและเหมาะสมกับระบบฐานข้อมูลมากกว่าไฟล์รูปภาพ

OCR สำหรับตัวอักษรภาษาไทย ได้มีผู้พัฒนาเป็นซอฟต์แวร์สำเร็จรูปกันบ้างแล้ว แต่ซอฟท์แวร์เหล่านั้น ระดับความถูกต้องในการจำแนกยังต่ำอยู่ เนื่องจากภาษาไทย ตัวอักษรที่ประกอบขึ้นเป็นคำ มีถึง 4 ระดับคือ ตัวพยัญชนะ, สระบน, สระล่าง และวรรณยุกต์ที่ประกอบอยู่กับสระบน ดังนั้นขั้นตอนการแยกอักษรแต่ละตัวออกมาจากภาพจึ่งเป็นเรื่องที่ยุ่งยากมากทีเดียว ซึ่งในขั้นตอนนี้เราได้ใช้การโปรแกรมแบบ Object-Oriented หรือ OOP ช่วยสำหรับการแยกตัวอักษรแต่ละตัวออกจากภาพ เมื่อเราแยกตัวอักษรได้แล้ว ขั้นตอนต่อไปเป็นการดึงลักษณะที่สำคัญ (Feature Extraction) ของตัวอักษรออกมาเพื่อเป็นอินพุทของ นิวรอลเน็ตเวิร์คซึ่งจะเป็นตัวจำแนกว่าเป็นอักษรตัวใด

สำหรับตัวอักษรภาษาไทยที่ใช้ในกระบวนการ OCR ในบทความชุดนี้ได้แสดงไว้ในตารางที่ 1. ซึ่งมีรหัส สมอ. ที่ใช้แทนรหัสของอักษรแต่ละตัวหลังจากผ่านการประมวลผลด้วย OCR แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1. แสดงตัวอักษรที่จะใช้เป็นอินพุทของ OCR

ตัวอักษร	รหัส สมอ.
ก	161
ข	162
ฃ	163
ค	164
ฅ	165
ฆ	166
ง	167
จ	168
ฉ	169
ช	170
ซ	171
ฌ	172
ญ	173
ฎ	174
ฏ	175
ฐ	176
ฑ	177
ฒ	178
ณ	179
ด	180
ต	181
ถ	182
ท	183
ธ	184
น	185
บ	186

ตัวอักษร	รหัส สมอ.
ป	187
ผ	188
ฝ	189
พ	190
ฟ	191
ภ	192
ม	193
ย	194
ร	195
ฤ	196
ล	197
ฦ	198
ว	199
ศ	200
ษ	201
ส	202
ห	203
ฬ	204
อ	205
ฮ	206
ฯ	207
ะ	208
ั	209
า	210
ิ	212
ี	213

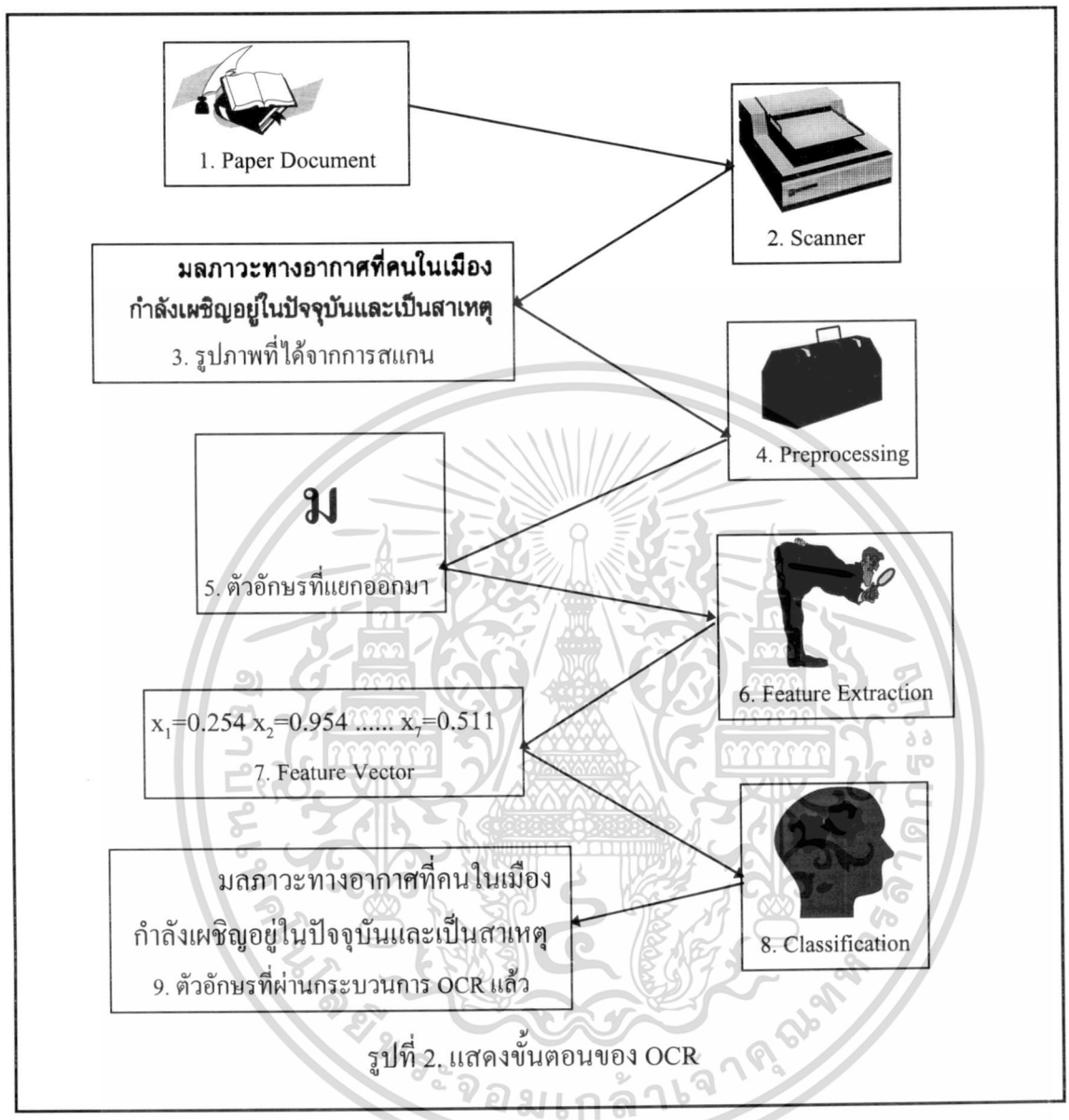
ตัวอักษร	รหัส สมอ.
ึ	214
ื	215
ุ	216
ู	217
เ	224
โ	226
ใ	227
ไ	228
ๆ	230
็	231
่	232
้	233
๊	234
๋	235
์	236
ำ	237
๐	240
๑	241
๒	242
๓	243
๔	244
๕	245
๖	246
๗	247
๘	248
๙	249

*หมายเหตุ ตัวอักษร “อ” ที่ใช้ประกอบกับสระบนและสระล่างเพื่อประโยชน์ให้สามารถพิมพ์สระเหล่านี้ได้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขั้นตอนของ OCR

ขั้นตอนการประมวลผลต่างๆ ของ OCR ได้แสดงไว้ในรูปที่ 2. โดยแต่ละขั้นตอนมีการ



รูปที่ 2. แสดงขั้นตอนของ OCR

ดำเนินการดังนี้

2.1 ข้อความต่างๆ ที่เราต้องการแปลงให้เป็นไฟล์ข้อมูล ซึ่งอาจจะอยู่ในรูปของแผ่นกระดาษหรือหนังสือ

2.2 สแกนข้อความที่เราต้องการด้วยสแกนเนอร์ โดยการสแกนควรใช้โหมดไบนารี

2.3 ข้อมูลที่ได้จากการสแกนจะเป็นภาพไบนารี หรืออาจจะเป็นภาพระดับสีเทาขึ้นอยู่กับการเลือกโหมดในการสแกน ข้อมูลภาพนี้อาจจะส่งไปประมวลผลในขั้นตอนต่อไปหรืออาจจะเก็บไว้ในลักษณะของไฟล์รูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 นำข้อมูลภาพมาเข้ากระบวนการประมวลผลก่อน (Preprocessing) ซึ่งในหัวข้อนี้เป็นขั้นตอนที่สำคัญมากขั้นตอนหนึ่งใน OCR และมีรายละเอียดต่างๆ ดังนี้

2.4.1 เริ่มจากการวิเคราะห์ของมูลภาพที่ป้อนเข้าสู่ระบบ คือถ้าไม่ใช่ภาพไบนารีก็ต้องแปลงเป็นภาพไบนารี โดยใช้วิธีการ Threshold ข้อมูลภาพ [1]

2.4.2 กรองสัญญาณรบกวน โดยใช้ตัวกรองสัญญาณความถี่ต่ำ (Low Pass Filter) [1] การกำจัดสัญญาณรบกวนก็เพื่อให้การแยกอักษรแตกละตัวง่าย และมีความถูกต้องสูงขึ้น [2]

2.4.3 สแกนหาความสูงของแต่ละบรรทัดของหน้ากระดาษ เพื่อกำหนดความสูงของบรรทัด ซึ่งความสูงของบรรทัดในหนึ่งหน้ากระดาษ สามารถแบ่งได้เป็นสองระดับ (หรือบางครั้งอาจจะมากกว่านี้) คือความสูงของบรรทัดปกติ ซึ่งเป็นความสูงของบรรทัดในพารากราฟ และความสูงของหัวข้อเรื่อง ซึ่งจะมีความสูงกว่าบรรทัดปกติ

2.4.4 ดึงข้อมูลภาพตัวอักษรออกมาที่ละบรรทัด เพื่อส่งไปให้ขั้นตอนการแยกตัวอักษร (Character Segmentation) ซึ่งเราจะใช้วิธีการตัดตัวอักษรออกมาที่ละบล็อก (Dissection Technique) โดยการวิเคราะห์การฉายภาพ (Projection Analysis) หรือเรียกอีกชื่อหนึ่งว่าการวิเคราะห์ฮีสโตแกรมในแนวตั้ง (Vertical Histogram Analysis) [3] ดังแสดงในรูปที่ 3.

รูปที่ 3. แสดงฮีสโตรแกรมในแนวตั้งของรูปภาพตัวอักษร

และในกรณีที่ฮีสโตแกรมในแนวตั้งไม่สามารถตัดอักษรแต่ละตัวออกมาได้ อย่างในกรณีของคุณภาพของข้อมูลภาพที่สแกนมาไม่ดีพอ, หรือในกรณีของตัวอักษรเอียง อักษรแต่ละตัวจะมีบางส่วนที่ทับกันอยู่ ดังนั้นในการดึงตัวอักษรในลักษณะเช่นนี้ออกมาเราจะใช้วิธีการโปรแกรมแบบพลวัต (Dynamic Programming) [4] เพื่อหาระยะทางที่สั้นที่สุด และการโปรแกรมแบบ OOP เข้ามาช่วยเหลือในการรวมและการแยก วัตถุในภาพแต่ละชิ้น หลังจากเราตัดตัวอักษรในแนวดิ่งออกมาได้แล้ว เราก็จะประมวลผลต่อในกระบวนการตัดตัวสระ และวรรณยุกต์ในแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตัดส่วนที่เป็นสระและวรรณยุกต์ เราแบ่งออกเป็น 4 ระดับด้วยกันคือ

- พยัญชนะ
- สระล่าง ซึ่งมีอยู่สองตัวคือ สระอุ และ สระอู
- สระบน วรรณยุกต์ (เช่นในคำว่า "พ่อ", "ช่วย", "ต้อง" เป็นต้น) สำหรับสระประกอบด้วย 6 ตัวได้แก่ อั, อิ, อี, อึ, อื และ อำ และรวมไปถึง อ์ และ อ็ ด้วย
- วรรณยุกต์ที่ประกอบกับตัวอักขระบน เช่น "ที่", "ซึ่ง", "ต่ำ" เป็นต้น

สำหรับเงื่อนไขในการตัดอักขระในแนวนอน เราใช้หลักการณ์อย่างเดียวกันกับการตัดอักษรในแนวตั้ง จะต่างกันก็ตรงเงื่อนไขในทางการเขียนโปรแกรม

2.4.5 เมื่อตัดตัวอักษรแต่ละตัวออกมาได้แล้ว ขนาดของตัวอักษรจะไม่เท่ากัน (สำหรับในงานวิจัยนี้ เราจะกำหนดขนาดของภาพตัวอักษรเท่ากับ 20x8 จุด) ดังนั้นเราจะทำการแปลงข้อมูลภาพทางเรขาคณิต (Geometric Transformation) [1] ในสองลักษณะด้วยกันคือ

- การแปลงขนาดของภาพตัวอักษรให้เท่ากับ 20x8 โดยใช้สมการที่ (1)

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (1)$$

โดยที่

x เป็นดัชนีของความสูงของตัวอักษร ซึ่งมีค่าตั้งแต่ 0 ถึงค่าความสูงของตัวอักษร

y เป็นค่าดัชนีความกว้างของตัวอักษร ซึ่งมีค่าตั้งแต่ 0 ถึงค่าความกว้างของตัวอักษร

a และ b เป็นสเกลของความสูงและความกว้างของตัวอักษร ตามลำดับ ซึ่งสามารถกำหนดได้ดังนี้

$$a = \frac{20}{\text{ความสูงของตัวอักษร}}$$

$$b = \frac{8}{\text{ความกว้างของตัวอักษร}}$$

เมื่อ 20 และ 8 เป็นความสูงและความกว้างที่เรากำหนดไว้

$x' = 0, 1, 2, ..., 19$

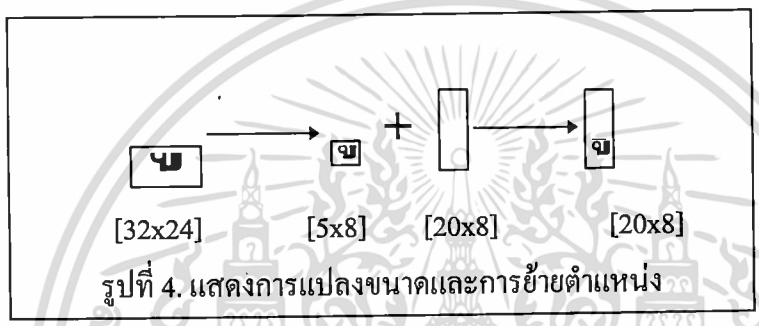
$y' = 0, 1, 2, ..., 7$

- การย้ายตำแหน่งของภาพ โดยใช้สมการที่ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

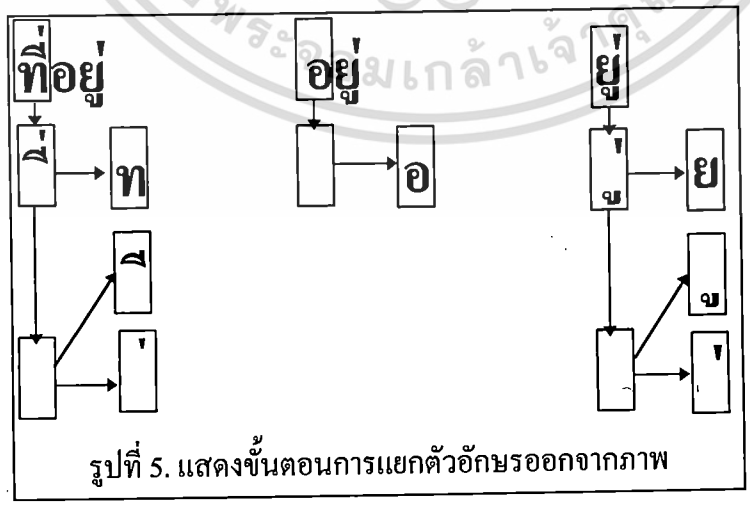
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \qquad (2)$$

โดยที่ x และ y เป็นตำแหน่งของจุดภาพที่ตัดมา เช่น สระล่างที่ตัดมาจากกรอบขนาด 32x16 ตำแหน่งของ (x,y) อาจจะอยู่ที่ (24,0) ดังนั้นเมื่อเราทำการแปลงขนาดเป็น 20x8 จุดแล้ว (x,y) จะย้ายไปอยู่ที่ (15,0) และ (x,y) ไปสิ้นสุดที่ตำแหน่ง (19,7) และนำสระล่างที่ผ่านกระบวนการแปลงขนาดแล้วไปเติมลงในเมตริกศูนย์ (x_0, y_0) เราก็จะได้ข้อมูลตัวอักษรใหม่ (x', y') ดังแสดงในรูปที่ 4.



รูปที่ 4. แสดงการแปลงขนาดและการย้ายตำแหน่ง

การแยกอักษรแต่ละตัวออกมาจากภาพ สำหรับภาษาไทยแล้วถือได้ว่าเป็นขั้นตอนที่สำคัญที่สุด เพราะถ้าไม่สามารถแยกตัวอักษรแต่ละตัวได้ถูกต้องตามตารางที่ 1. แล้ว ขั้นตอนที่ตามมาจะผิดทั้งหมด สำหรับขั้นตอนการแยกอักษรแต่ละตัวได้แสดงไว้ในรูปที่ 5. ซึ่งเป็นการแยกอักษรจากคำว่า "ที่อยู่"



รูปที่ 5. แสดงขั้นตอนการแยกตัวอักษรออกจากภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.5 นำตัวอักษรที่แยกออกมาได้มาจัดลำดับเพื่อรอส่งไปขั้นตอน Feature Extraction เนื่องจากในการแยกตัวอักษรออกมาแต่ละตัว บางครั้งอาจจะมี สระ และวรรณยุกต์ประกอบอยู่ด้วย ดังนั้นเราจะต้องจัดลำดับการส่งไปประมวลผลเหมือนกับขั้นตอนการพิมพ์ในตัวประมวลผลคำ (Word Processor)

2.6 Feature Extraction เป็นขั้นตอนที่มีความสำคัญมากพอๆ กับขั้นตอนการประมวลผลก่อน เนื่องจากขั้นตอนนี้จะเป็นการดึงเอาคุณลักษณะที่จะบ่งบอกว่า รูปตัวอักษรที่กำลังพิจารณาอยู่คือตัวอะไร สำหรับวิธีการของ Feature Extraction ที่ใช้กับ OCR มีอยู่หลายวิธีด้วยกัน ดังแสดงในตารางที่ 2. [5] ซึ่งเราจะเห็นว่าแต่ละวิธีมีลักษณะการใช้ที่แตกต่างกันไป ตามลักษณะของข้อมูลภาพที่เรากำลังประมวลผลอยู่ สำหรับวิธีของ Feature Extraction ที่ดีจะต้องเป็นตัวแทนข้อมูลภาพของตัวอักษรได้เป็นอย่างดี ถึงแม้ว่าข้อมูลภาพจะมีสัญญาณรบกวนอยู่ก็ตาม

ตารางที่ 2. แสดงวิธีการของ Feature Extraction ที่ใช้กับรูปแบบของข้อมูลภาพที่ต่างๆ กัน

ภาพระดับสีเทา	ภาพไบนารี่		เวคเต่อร์ของโครงร่างตัวอักษร
	Solid Symbol	Outer Contour	
Template Matching	Template matching		Template matching
Unitary transforms	Unitary transforms		
Zoning	Zoning	Zoning	Zoning
Geometric moments	Geometric moments	Spline curve	Discrete Features
Zernike moments	Zernike moments	Fourier descriptors	Graph description
Deformable templates	Projection histograms		Deformable templates

สำหรับวิธี Feature Extraction ที่เรานำมาใช้ในการดึงคุณลักษณะที่สำคัญของอักษรแต่ละตัวในที่นี้คือโมเมนต์เชิงเรขาคณิต Geometric Moment ซึ่งเป็นตัวเดียวกันกับโมเมนต์ที่ใช้ในทางสถิติ [6] ซึ่งโมเมนต์ที่หนึ่งก็คือค่าเฉลี่ย, โมเมนต์ที่สองคือค่าความแปรปรวน เป็นต้น และสำหรับโมเมนต์ที่นำมาใช้ในทางด้านการประมวลผลภาพ จะใช้ในการวิเคราะห์รูปร่างของวัตถุ (Shape Analysis) [7] โดยเฉพาะโมเมนต์ที่ 0-3 จะถูกนำมาใช้ในกระบวนการของการทำให้คอมพิวเตอร์สามารถมองเห็นวัตถุ (Computer Vision) และวิธีนี้ยังสามารถให้ผลลัพธ์ได้ดีแม้ว่าขนาด, ตำแหน่ง และทิศทางของวัตถุเปลี่ยนไป ในสมการที่ (3) เป็นโมเมนต์ที่มีศูนย์กลางอยู่ที่ค่าเฉลี่ย ($\bar{x}, \bar{y}$)

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y) \qquad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32254

โดยที่ f(x,y) เป็นจุดภาพที่ตำแหน่ง (x,y) เมื่อ $f(x,y) \geq 0$

p, q เป็นอันดับของโมเมนต์

ในสมการที่ (4) เป็นสมการของโมเมนต์อันดับที่ 1-7

$$\phi_1 = \mu_{20} + \mu_{02}$$

$$\phi_2 = (\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2$$

$$\phi_3 = (\mu_{30} + 3\mu_{12})^2 + (3\mu_{21} + \mu_{03})^2$$

$$\phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$\phi_5 = (\mu_{30} + 3\mu_{12})(\mu_{21} + \mu_{03})\left[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right] + (\mu_{03} - 3\mu_{21})(\mu_{03} + \mu_{21})\left[(\mu_{03} + \mu_{21})^2 - 3(\mu_{12} + \mu_{30})^2\right] \quad (4)$$

$$\phi_6 = (\mu_{20} + \mu_{02})\left[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right] + 4\mu_{11}(\mu_{03} + \mu_{21})(\mu_{30} + \mu_{12})$$

$$\phi_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})\left[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right] - (3\mu_{12} - \mu_{30})(\mu_{03} + \mu_{21})\left[(\mu_{03} + \mu_{21})^2 - 3(\mu_{12} + \mu_{30})^2\right]$$

จากสมการที่ (4) สามารถนอร์มอลไลซ์ (Normalize) ได้ดังนี้

$$\psi_1 = \frac{\phi_1}{m_{00}^2}$$

$$\psi_2 = \frac{\phi_2}{m_{00}^4}$$

$$\psi_3 = \frac{\phi_3}{m_{00}^5}$$

$$\psi_4 = \frac{\phi_4}{m_{00}^5} \quad (5)$$

$$\psi_5 = \frac{\phi_5}{m_{00}^{10}}$$

$$\psi_6 = \frac{\phi_6}{m_{00}^7}$$

$$\psi_7 = \frac{|\phi_7|}{m_{00}^{10}}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $m_{pq} = \sum_{x}\sum_{y} x^p y^q f(x,y)$

จากสมการที่ (5) เราคำนวณค่าของโมเมนต์ที่ 1-7 และได้แสดงในตารางที่ 3.

ตารางที่ 3. แสดงค่าของโมเมนต์อันดับที่ 1-7 ของตัวอักษร ที่เป็นตัวปกติและตัวเอียง

ตัวอักษร	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5	ψ_6	ψ_7
ก	0.599451	0.221665	0.105518	0.222261	0.033915	0.104494	0.002731
ก	0.563347	0.183021	0.078399	0.200802	0.025192	0.085890	0.005374
ณ	0.359481	0.071712	0.017760	0.050717	0.001521	0.013581	0.000138
ณ	0.361854	0.076215	0.020509	0.053482	0.001771	0.014765	0.000218
ญ	0.406422	0.088247	0.028387	0.079482	0.003708	0.023557	0.001548
ญ	0.416802	0.083169	0.020714	0.082120	0.003318	0.023626	0.001567
ฎ	0.480511	0.141867	0.059111	0.123851	0.010512	0.046593	0.005464
ฎ	0.335386	0.054932	0.014436	0.041464	0.000960	0.009717	0.000363
ฏ	0.550099	0.195491	0.102395	0.191063	0.026610	0.084374	0.013740
ฏ	0.418148	0.098841	0.033854	0.083278	0.004376	0.026159	0.002345
ฌ	0.348190	0.065924	0.016723	0.047990	0.001359	0.012322	0.000141
ฌ	0.346259	0.065441	0.014701	0.046285	0.001207	0.011840	0.000196
ถ	0.508875	0.147775	0.048384	0.129773	0.010246	0.049823	0.002018
ถ	0.504644	0.156898	0.055965	0.138039	0.012130	0.054676	0.000971
ภ	0.504977	0.158361	0.059950	0.136960	0.012405	0.054486	0.000698
ภ	0.447275	0.127780	0.049083	0.107341	0.007789	0.038370	0.001698

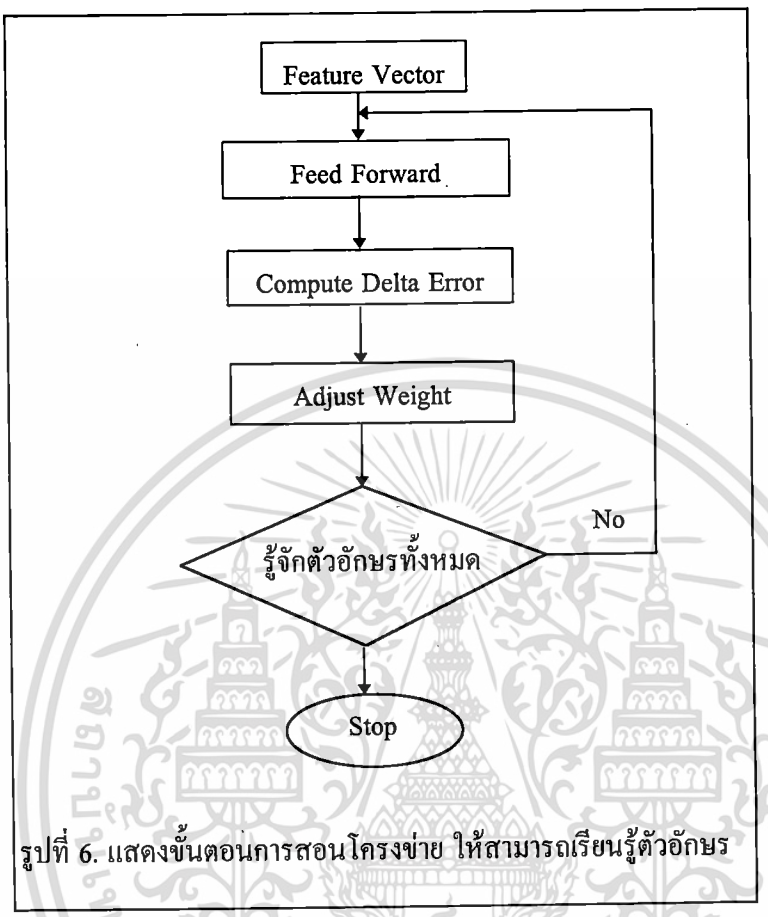
2.7 ในตารางที่ 3. เป็นเวคเตอร์ของตัวแปรที่บอกถึงลักษณะ (Feature Vector) ของอักษรแต่ละตัว และจากเวคเตอร์ของค่าโมเมนต์เหล่านี้ ก็จะถูกส่งไปประมวลผลในขั้นตอนการจำแนก (Classification) ตัวอักษรแต่ละตัว ด้วยนิวรอลเน็ตเวิร์ค (Neural Network)

2.8 การจำแนกตัวอักษรโดยใช้นิวรอลเน็ตเวิร์คในที่นี้ เราได้ใช้โครงข่ายแบบแบ็กค์โปรพาเกชัน (Back Propagation) ซึ่งเป็นโครงข่ายที่มี 4 ลำดับชั้น โดยในชั้นแรก (Input Layer) ประกอบด้วย 7 โหนด, ชั้นที่สองและสามเป็นชั้นฮิดเดน (Hidden Layer) ซึ่งแต่ละชั้นจะมีอยู่ 76 โหนด และชั้นสุดท้าย (Output Layer) มีอยู่ 76 โหนดเช่นกัน ซึ่งแต่ละโหนดในชั้นนี้ก็จะแทนตัวอักษรแต่ละตัวนั่นเอง

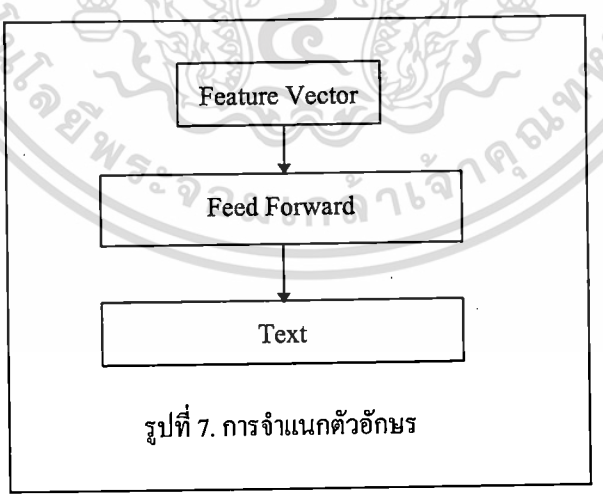
การจำแนกตัวอักษรด้วยนิวรอลเน็ตเวิร์ค มีอยู่สองขั้นตอนด้วยกันคือ ขั้นตอนการสอนโครงข่าย [1] (อัลกอริทึมของการเรียนรู้ได้แสดงในรูปที่ 6.) และเมื่อโครงข่ายได้เรียนรู้ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักษรทั้งหมดได้แล้ว ก็จะเป็นขั้นตอนของการใช้งาน ซึ่งอัลกอริทึมของขั้นตอนนี้ได้แสดงไว้ในรูปที่ 7.



รูปที่ 6. แสดงขั้นตอนการสอนโครงข่าย ให้สามารถเรียนรู้ตัวอักษร



รูปที่ 7. การจำแนกตัวอักษร

2.9 เมื่อผ่านกระบวนการจำแนกตัวอักษรแล้ว ผลลัพธ์ที่ได้จะเป็นตัวอักษรที่มีลักษณะเดียวกับ ข้อความในกระดาษที่สแกนเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. สรุป

การจำตัวอักษรแบบตัวพิมพ์โดยใช้นิวรอลเน็ตเวิร์คที่ได้เสนอไปแล้วนั้น ปัญหาของระบบจะอยู่ที่ การตัดตัวอักษรออกมาจากภาพ เนื่องจากลักษณะของภาษาไทยเป็นภาษาที่มีตัวอักษรหลายระดับในการเขียนเพื่อประกอบเป็นคำแต่ละคำ ดังนั้นนอกจากตัดตัวอักษรในแนวดิ่งแล้ว เรายังต้องแยกตัวอักขระในแนวระดับอีกด้วย นอกจากนี้ในขั้นตอน Feature Extraction เวคเตอร์ของตัวแปรยังมีจำนวนน้อย ทำให้การจดจำตัวอักษรที่คล้ายกันของโครงข่ายประสบปัญหา ซึ่งตัวอักษรเหล่านี้บางส่วนได้แสดงไว้ในตารางที่ 3.

สำหรับการแก้ไขปัญหาดังกล่าว ปัจจุบันเรากำลังพัฒนาการตัดตัวอักขระแต่ละตัวโดยใช้ การแยกตัวอักษรโดยใช้การรู้จัก (Recognition-Based Segmentation) [8] และในส่วนของ Feature Extraction เรากำลังทดลองนำตัวแปรที่ระบุถึงทางเดินของจุด ที่ประกอบขึ้นเป็นตัวอักษรมาใช้ร่วมกับตัวแปรเดิมที่เรามีอยู่ เพื่อให้โครงข่ายนิวรอลเน็ตเวิร์คสามารถจำจดและแยกแยะตัวอักษรได้ดีขึ้น

เอกสารอ้างอิง

[1] สาธิต อินทจักร์, **การหาขอบภาพโดยใช้แบบจำลองฟัซซี่และนิวรอลเน็ตเวิร์ค**, วิทยานิพนธ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2538

[2] S. Tsujimoto and H. Asada, "Major Components of a Complete Text Reading System," *Proc. IEEE*, vol. 80, no. 7, pp. 1133-1149, July 1992.

[3] K. Ohta, L. Kaneko, Y. Itamoto, and Nishijima, "Charecter Segmentation of Address Reading/Letter Sorting Machine for the Ministry of Posts and Telecommunications of Japan," *NEC Research and Development*, vol. 34, no. 2, pp. 248-256, Apr. 1993.

[4] J. Wang and J. Jean, "Segmentation of Merged Characters by Neural Networks and Shortest Path," *Pattern Recognition*, vol. 27, no. 5, pp. 649-658, May 1994.

[5] Oivind Due Trier, Anil K. Jain and Torfinn Taxt, "Feature Extraction Methods for Character Recognition - A Survey," *Pattern Recoginition*, Vol. 29, No. 4, pp. 641-662, 1996.

[6] C. W. Therrien, Decision Estimation and Classification, John Wiley & Sons, Inc. Canada, pp. 55-56, 1989.

[7] T. H. Resiss, Recognizing Planar Objects Using Invariant Image Features, Springer-Verlag, Berlin, 1993.

[8] A. Kundu, Y. He, and P. Bahl, "Recognition of Handwrite Word: First and Second Order Hidden Markov Model Based Approach," Pattern Recognition, vol. 22, no. 3, p. 283, 1989.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้