รายงานการปฏิบัติงานสหกิจศึกษา

เรื่อง ระบบตรวจหาข้อความบนภาพมังงะด้วยเทคนิค Stroke Width Transform

ปฏิบัติงาน ณ มหาวิทยาลัยฮอกไกโด

โดย

นาย บุญฤทธิ์ พิริย์โยธินกุล
รหัสประจำตัว 58070077

รายงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา สหกิจศึกษา
สาขาวิชาวิศวกรรมซอร์ฟแวร์คณะเทคโนโลยีสารสเนทศ
ภาคเรียนที่ 1 ปีการศึกษา 2561
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

รายงานการปฏิบัติงานสหกิจศึกษา

เรื่อง ระบบตรวจหาข้อความบนภาพมังงะด้วยเทคนิค Stroke Width Transform

ปฏิบัติงาน ณ มหาวิทยาลัยฮอกไกโด

โดย

นาย บุญฤทธิ์ พิริย์โยธินกุล
รหัสประจำตัว 58070077

ปฏิบัติงาน ณ มหาลัยวิทยาลัยฮอกไกโด
Hokkaido University Kita 8, Nishi 5, Kita-ku,
Sapporo, Hokkaido, 060-0808 Japan
Web site : www.global.hokudai.ac.jp

วันที่ 10 พฤศจิกายน พ.ศ. 2561

เรื่อง ขอส่งรายงานการปฏิบัติงานสหกิจศึกษา
เรียน อาจารย์ กิติ์สุชาต พสุภา
ที่ปรึกษาสหกิจศึกษาในสาขา วิศวกรรมซอร์ฟแวร์

ตามที่ข้าพเจ้า นาย บุญฤทธิ์ พิริย์โยธินกุล นักศึกษาสาขาวิชา วิศวกรรมซอร์ฟแวร์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ได้ปฏิบัติงานสหกิจศึกษาระหว่างวันที่ 23 กรกฎาคม พ.ศ. 2561 ถึงวันที่ 30 พฤศจิกายน พ.ศ. 2561 ในตำแหน่ง ผู้ช่วยนักวิจัย ณ สถานประกอบการชื่อ มหาวิทยาลัยฮอกไกโด และได้รับมอบหมายจากพนักงานที่ปรึกษาให้ศึกษาและจัดทำรายงาน เรื่อง ระบบตรวจหาข้อความบนภาพมังงะด้วยเทคนิค Stroke Width Transform

บัดนี้ การปฏิบัติงานสหกิจศึกษาได้สิ้นสุดลงแล้ว จึงใคร่ขอส่งรายงานการปฏิบัติงาน สหกิจศึกษาดังกล่าวมาพร้อมนี้ จำนวน 1 เล่ม เพื่อขอรับคำปรึกษาต่อไป

จึงเรียนมาเพื่อโปรดพิจารณา

ขอแสดงความนับถือ

.....
(นาย บุญฤทธิ์ พิริย์โยธินกุล)

กิตติกรรมประกาศ

ตามที่ข้าพเจ้า นาย บุญฤทธิ์ พิริย์โยธินกุล ได้มาปฏิบัติงานสหกิจศึกษา ณ มหาวิทยาลัยฮอกไกโด ตั้งแต่วันที่ 23 กรกฎาคม พ.ศ. 2561 ถึงวันที่ 30 พฤศจิกายน พ.ศ. 2561 ทำให้ข้าพเจ้าได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีคุณค่ามากมาย สำหรับรายงานสหกิจศึกษาฉบับนี้สำเร็จลงได้ด้วยดี จากความช่วยเหลือและความร่วมมือสนับสนุนของหลายฝ่าย ดังนี้

1. คุณ Masanori Sugimoto ตำแหน่ง ศาสตราจารย์
2. คุณ Jiang Ye ตำแหน่ง นักศึกษาปริญญาเอก ปี 2

นอกจากนี้ยังมีบุคคลท่านอื่น ๆ อีกที่ไม่ได้กล่าวไว้ ณ ที่นี้ ซึ่งให้ความกรุณาแนะนำในจัดทำรายงานสหกิจศึกษาฉบับนี้ ข้าพเจ้าจึงใคร่ขอขอบพระคุณทุกท่านที่ได้มีส่วนร่วมในการให้ข้อมูลและให้ความเข้าใจเกี่ยวกับชีวิตของการปฏิบัติงาน รวมถึงเป็นที่ปรึกษาในการจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์

นาย บุญฤทธิ์ พิริย์โยธินกุล
ผู้จัดทำรายงาน
วันที่ 10 พฤศจิกายน พ.ศ. 2561

ชื่อรายงานการปฏิบัติงานสหกิจศึกษา	ระบบตรวจหาข้อความบนภาพมังงะด้วยเทคนิค Stroke Width Transform
ผู้รายงาน	นาย บุญฤทธิ์ พิริย์โยธินกุล
คณะ	เทคโนโลยีสารสนเทศ
สาขาวิชา	วิศวกรรมซอร์ฟแวร์

.....

(อาจารย์ กิติ์สุชาต พสุภา)

อาจารย์ที่ปรึกษาสหกิจศึกษา

.....

(Masanori Sugimoto)

พนักงานที่ปรึกษา

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับรายงานการปฏิบัติงานสหกิจศึกษาฉบับนี้ เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิศวกรรมซอร์ฟแวร์

ชื่อนักศึกษา ระบบตรวจหาข้อความบนภาพมังงะด้วยเทคนิค Stroke Width Transform
รหัสนักศึกษา นาย บุญฤทธิ์ พิริย์โยธินกุล
สาขาวิชา วิศวกรรมซอร์ฟแวร์
อาจารย์ที่ปรึกษา อาจารย์ กิติ์สุชาต พสุภา
ปีการศึกษา 2561

บทคัดย่อ

การ์ตูนญี่ปุ่น หรือที่รู้จักกันอย่างแพร่หลายว่า **มังงะ (Manga)** กลายเป็นหนึ่งในหัวข้อที่ถูกหยิบมาวิจัย งานวิจัยชิ้นนี้มุ่งเน้นที่ปัญหาการตรวจหาข้อความบนภาพวาดมังงะ เนื่องจากปัญหาการสร้างชุดข้อมูลภาพมังงะและข้อมูลประกอบ (Annotation) อย่างเช่นการระบุขอบเขตข้อความ ซึ่งต้องใช้แรงงานคนและกินเวลาอย่างมาก ดังนั้นระบบอัตโนมัติที่สามารถเข้ามาช่วยในงานส่วนนี้จึงเป็นที่ต้องการอย่างมาก โดยเราได้นำเสนอวิธีการตรวจหาข้อความบนภาพมังงะแบบใหม่ด้วยการใช้ Stroke Width Transform (SWT) ร่วมกับการใช้ Support Vector Machine (SVM) อย่างไรก็ดีวิธีการเดิมที่ใช้ SWT เพื่อการตรวจหาข้อความบนภาพถ่ายนั้นไม่สามารถประยุกต์ใช้กับภาพมังงะได้ เพราะความแตกต่างระหว่างวัตถุและลักษณะอักษรของข้อความในภาพวาดและภาพถ่าย ไม่ว่าจะในเชิงความคล้ายคลึง, ขนาด, รูปร่าง, และลักษณะของเส้น ดังนั้นเพื่อให้สามารถใช้งานกับมังงะได้ เราจึงนำวิธีการตรวจหาข้อความด้วย SWT ต้นฉบับมาปรับปรุงและพัฒนาขึ้นเป็นวิธีการใหม่ของเรา เราได้ปรับปรุงกฎเกณฑ์ในการค้นหาวัตถุที่คล้ายคลึงอักษร (Letter candidates) ซึ่งช่วยเพิ่มประสิทธิภาพในการตรวจจับอักษรได้ครบถ้วนมากขึ้น และใช้ SVM เพื่อคัดแยกวัตถุอื่น ๆ ออกจากอักษร ช่วยในการลด False positive ของผลลัพธ์ ในท้ายที่สุดเรานำประสิทธิภาพวิธีการของเรามาเปรียบเทียบร่วมกับวิธีการต้นฉบับและวิธีอื่น ๆ รวมถึงวิธีที่ใช้ Deep learning เป็นส่วนประกอบ ในท้ายที่สุดประสิทธิภาพของวิธีการใหม่ของเรานั้นสามารถทำคะแนน F-measure ได้สูงสุดเทียบกับวิธีการอื่น ๆ ที่ 0.506

สารบัญ

หน้า

สารบัญตาราง

หน้า

สารบัญภาพ

หน้า

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

การ์ตูนญี่ปุ่นเป็นที่รู้จักกันอย่างแพร่หลายทั่วโลกในฐานะสื่อบันเทิง หรืออีกชื่อหนึ่งคือ “มังงะ (Manga)” ในปัจจุบันมีงานวิจัยในหัวข้อมังงะอย่างหลากหลาย ในหลาย ๆ งานวิจัย [2–8] มีการใช้ชุดข้อมูลสำหรับการทดสองอย่างเช่น Manga109 [9] โดยชุดข้อมูลนี้เป็นข้อมูลที่ถูกสร้างขึ้นจากภาพมังงะจำนวน 20,260 หน้า จากมังงะ 109 เรื่อง ถูกวาดโดยนักวาดมืออาชีพชาวญี่ปุ่น นอกจากภาพสแกนของมังงะแล้ว ชุดข้อมูลนี้ยังประกอบไปด้วยข้อมูลอธิบายประกอบ หรือ Annotation ต่าง ๆ เช่น ขอบเขตและตำแหน่งของใบหน้า ร่ายกาย กรอบภาพ เป็นต้น นอกจากนี้ยังมีข้อมูลขอบเขตและตำแหน่งของข้อความที่ปรากฏในภาพมังงะซึ่งถูกป้อนข้อมูลด้วยแรงงานคนโดยไม่พึ่งพาระบบอัตโนมัติใด ๆ ในการป้อนข้อมูลดังกล่าวนั้นใช้เวลานานและต้องพึ่งพาแรงงานมนุษย์ ด้วยเหตุนี้ระบบอัตโนมัติที่จะเข้ามาช่วยในการระบุข้อมูล Annotation นั้นจึงมีประโยชน์และลดภาระงานลงได้อย่างมาก

ถึงแม้สำหรับภาพการ์ตูนญี่ปุ่นจะมีทั้งแบบภาพวาดทั่วไปและมังงะ แต่ในงานวิจัยนี้เรามุ่งเน้นไปที่มังงะเป็นหลักเนื่องจากข้อความมักมีบนหนังสือการ์ตูนมากกว่าภาพวาดทั่วไปอย่างที่ทราบกัน สำหรับวิธีการตรวจหาข้อความในภาพมังงะนั้นมีการพัฒนามาหลากหลายก่อนหน้านี้ [10, 11] แต่วิธีเหล่านี้ถูกจำกัดให้ทำงานภายในโครงสร้างของภาพมังงะต่าง ๆ เช่น กรอบช่องภาพวาด, ลักษณะของกล่องคำพูด เป็นต้น นอกจากนี้บางวิธียังคงต้องพึ่งพาการป้อนข้อมูลเข้าจากภายนอกทำให้ไม่สามารถทำงานได้อัตโนมัติอย่างสมบูรณ์ อย่างไรก็ดีไม่นานมานี้มีการพัฒนาวิธีการใหม่โดยใช้วิธีการ Deep learning อย่างเช่นเทคนิค Convolutional neural network เพื่อช่วยในการสกัดลักษณะเด่น (Feature) ออกจากภาพมังงะเพื่อช่วยในการตรวจหาข้อความในภาพมังงะ [12] ซึ่งวิธีนี้การลดข้อผิดพลาด False positive ไปได้โดยปราศจากการใช้โครงสร้างในภาพมังงะต่าง ๆ แต่อย่างไรก็ดี Deep learning ยังเป็นการวิธีการที่มีข้อเสียคือใช้ทรัพยากรในการคำนวนของระบบมากกว่าระบบอื่น ๆ [12]

ในงานวิจัยนี้เราต้องการพัฒนาระบบตรวจหาข้อความที่ทำงานได้กับมังงะอย่างหลากหลายและไม่ถูกจำกัดด้วยโครงสร้างหรือลักษณะบางประการของภาพมังงะ เราจึงเลือกใช้ Stroke Width Transform (SWT) ในการสกัดลักษณะเด่นของเส้นต่าง ๆ ที่อยู่ในภาพออกมา โดยวิธีการนี้ถูกใช้เป็นขั้นตอนแรกเริ่มในการตรวจหาข้อความบนภาพถ่ายมาก่อน ทำงานด้วยการพึ่งพาสมมติฐานที่ว่าขอบของเส้นอักษรในข้อความนั้นมีขอบที่ชัดเจนและหนาแน่นบนพื้นหลังที่ราบเรียบ [1] อย่างไรก็ดีการใช้วิธีการนี้กับการตรวจหาข้อความบนภาพมังงะส่งผลให้เกิด False positive จำนวนมากในการตรวจหาซึ่งส่งผลทางลบต่อประสิทธิภาพการทำงานของระบบ ปัญหานี้เกิดจากความแตกต่างของภาพถ่ายและภาพวาดมังงะ มังงะนั้นเป็นภาพขาวดำ และลักษณะของวัตถุภายในมังงะ เช่น ขนาด, เส้น, พื้นหลัง นั้นมีความคล้ายคลึงกับตัวอักษรของข้อความมาก ดังนั้นเราจึงตั้งเป้าหมายในการปรับปรุงและพัฒนา SWT ที่ถูกใช้ในภาพถ่าย [1] และนำมันมาปรับปรุงพัฒนาเพิ่มเพื่อให้สามารถทำงานกับภาพมังงะได้

วิธีการใหม่ของเราที่จะนำเสนอในงานนี้สามารถแบ่งออกเป็น 4 ส่วนดังนี้ (i) the Stroke Width Transform (ii) ค้นหาวัตถุที่เข้าข่ายลักษณะของตัวอักษร (iii) คัดแยกอักษร โดยใช้ Support Vector

Machine (SVM) ร่วมกับ Histogram of Oriented Gradients Feature (iv) จัดกลุ่มอักษรที่ผ่านการคัดแยกแล้วเป็นบรรทัดเดียวกันหรือกลุ่มข้อความเดียวกัน

1.2 วัตถุประสงค์

พัฒนาระบบค้นหาตำแหน่งข้อความสำหรับมังงะ โดยนำ Stroke Width Transform ที่ถูกใช้เป็นกระบวนการแรกเริ่มในเทคนิคตรวจหาข้อความบนภาพถ่ายมาพัฒนาและปรับปรุงเพื่อให้ใช้กับภาพมังงะได้ดีมากขึ้น

1.3 ขอบเขตของงานวิจัย

1. พัฒนาระบบตรวจหาตำแหน่งข้อความซึ่งใช้สำหรับภาพมังงะขาวดำ
2. ภาษาของเนื้อหาในมังงะที่นำมาใช้งาน คือ ภาษาญี่ปุ่น
3. ข้อมูลที่ใช้ในการวิจัยนำมาจากฐานข้อมูล Manga109

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ Stroke Width Transform ที่ผ่านการปรับปรุงสำหรับภาพมังงะโดยเฉพาะมาเพื่อใช้ในการใช้งานด้านอื่น ๆ ต่อไป
2. ทำให้ทราบถึงลักษณะที่เป็นเอกลักษณ์ของมังงะซึ่งแตกต่างจากภาพถ่ายทั่วไป

บทที่ 2
แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การตรวจหาข้อความในภาพถ่ายด้วยเทคนิค Stroke Width Transform

วิธีการการตรวจหาข้อความในภาพถ่าย [1] นี้เป็นวิธีการที่เรานำมาใช้ศึกษาและเป็นต้นแบบในการพัฒนาเพื่อทำงานร่วมกับภาพมังงะ โดยมีขั้นตอนทั้งหมดแบ่งได้เป็น 3 ขั้นตอน ขั้นแรกคือการใช้ Stroke Width Transform ในการปรับเปลี่ยนข้อมูลให้แสดงลักษณะของความกว้างในแต่ละเส้นภายในภาพ ขั้นที่สอง ค้นหาวัตถุที่คล้ายคลึงกับตัวอักษรในภาพโดยใช้กฎเกณฑ์ที่กำหนดไว้ ขั้นสุดท้าย คือการจัดกลุ่มตัวอักษรเข้าด้วยกันเป็นบรรทัดของข้อความ

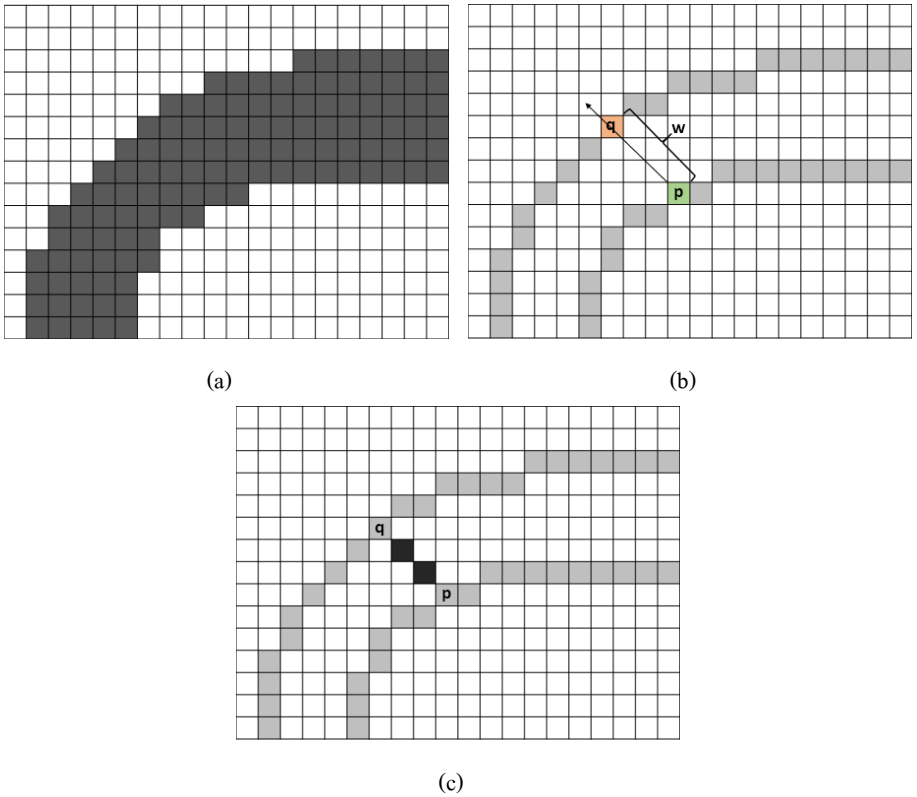
2.1.1 Stroke Width Transform

Stroke Width Transform หรือ SWT เป็นเทคนิคที่ใช้ช่วยในการทำงานของระบบตรวจหาข้อความในภาพถ่าย โดยสกัดลักษณะ (Feature) ของเส้นต่าง ๆ ในภาพ เช่น เส้นของตัวอักษร เป็นต้น [1] ด้วยลักษณะดังกล่าว ทำให้เราสามารถใช้ในการคัดแยกวัตถุที่เป็นตัวอักษรออกจากวัตถุอื่น ๆ ได้โดยพึ่งพาลักษณะเด่นเหล่านั้น

เริ่มแรกเราสร้างภาพ Output ที่มีขนาดเท่ากับภาพที่ต้องการตรวจหาข้อความ โดยแต่ละ Pixel ในภาพถูกกำหนดให้มีค่าอนันต์ (∞) จากนั้นใช้ Canny edge detection [13] ซึ่งเป็นเทคนิคตรวจหาขอบของเส้นหรือวัตถุ ตรวจหาตำแหน่งของขอบของวัตถุในภาพ ต่อมาคือการคำนวนความกว้างของเส้นโดยใช้ขอบที่ได้มา ความกว้างคำนวนได้จากระยะห่างระหว่างขอบของเส้นโดยพิจารณาทุก Pixel p ของขอบที่ได้จาก Canny edge detection เพื่อหา Pixel q ที่เข้าคู่กัน อย่างที่แสดงให้เห็นในภาพ 2.1(b) การหา q จาก p ทำได้โดยใช้ทิศทางของเฉดสี หรือ Gradient direction ของ p ซึ่งคือ d_p โดย d_p จะชี้ไปหา q และหาก d_p และ d_q มีทิศทางตรงกันข้ามโดยประมาณ $d_q = -d_p \pm \pi/6$ ให้กำหนดค่าให้กับแต่ละ Pixel ที่อยู่ภายใต้เส้นทางระหว่าง p และ q ให้เท่ากับ $\|\overrightarrow{p-q}\|$ เว้นแต่ Pixel ที่จะระบุค่าให้นั้นมีค่าน้อยกว่าค่าใหม่ที่จะระบุให้ หากค่าใหม่น้อยกว่าค่าเดิมใน Pixel ให้ทำการระบุค่าใหม่แทนทีค่าเดิมให้กับ Pixel นั้น อย่างที่ปรากฏในภาพ 2.1(c) ในที่สุดเราจะได้เมทริกซ์ Output ขนาดเท่าภาพ Input ที่มีความยาวของขอบเป็นค่าที่ถูกระบุในแต่ละพื้นที่

2.1.2 ค้นหาวัตถุที่มีลักษณะใกล้เคียงตัวอักษร

ในขั้นตอนนี้เรานำผลลัพธ์จากขั้นตอนก่อนหน้ามากำจัดวัตถุที่ไม่เกี่ยวข้องกับอักษรออก เริ่มจากจับกลุ่มแต่ละ Pixel ในผลลัพธ์ของขั้นตอน SWT การจับกลุ่มทำได้โดยเปรียบเทียบแต่ละ Pixel กับ Pixel เพื่อนบ้านรอบข้าง หาก Pixel สองตัวที่เปรียบเทียบกันมีค่าของความกว้างเส้นไม่ต่างกันเกิน 3.0 Pixel ทั้งสองจะถูกจัดกลุ่มเข้าด้วยกัน อย่างไรก็ดีหากวัตถุที่เกิดจากการรวมกลุ่มของ Pixel นั้นใหญ่หรือเล็กเกินไปก็จะถูกคัดออก การคัดวัตถุลักษณะดังกล่าวออกไปทำโดยการใช้กฎสองข้อ (i) อัตราส่วนระหว่างเส้นผ่าศูนย์กลางต่อมัธยฐานของความกว้างของเส้นตัวอักษรนั้นต้องน้อยกว่า 10 (ii) ความสูงต้องมากกว่า 10 และน้อยกว่า 300 ตามที่แสดงในสมการ 2.1



(a) (b)

(c)

รูปที่ 2.1: ขั้นตอนการทำงานของ Stroke Width Transform

$$f(d,h,\tilde{s}) = \begin{cases} 1, & \text{if } \frac{d}{\tilde{s}} < 10 \text{ and } 10 < h < 300 \\ 0, & otherwise \end{cases}, \tag{2.1}$$

โดย d คือ เส้นผ่าศูนย์กลางของวัตถุ, h คือ ความสูงของวัตถุ, และ $\tilde{s}$ คือ มัธยฐานของของความกว้างเส้นตัวอักษร โดยค่าความกว้างได้มาจากค่าของ Pixel ในพื้นที่ของเส้นที่ถูกระบุไปโดยขั้นตอน SWT

2.1.3 จัดกลุ่มตัวอักษร

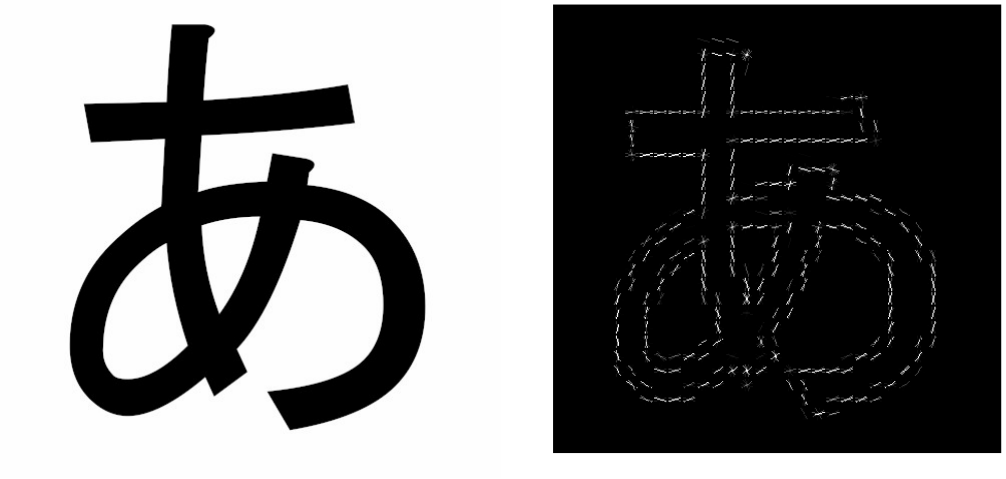
วัตถุแต่ละชิ้นที่ทีลักษณะคล้ายคลึงกับอักษรซึ่งผ่านการคัดกรองด้วยกฎเกณฑ์จากขั้นตอนก่อนหน้าจะถูกนำมาจับกลุ่มในขั้นตอนนี้ ด้วยการเปรียบเทียบความคล้ายคลึงระหว่างลักษณะอักษร ประกอบไปด้วย ระยะห่างระหว่างวัตถุ, อัตราส่วนความกว้างของเส้นอักษร, และความสูงของอักษร โดยสองวัตถุจะถูกจัดกลุ่มกันต่อเมื่อ (i) ทั้งสองมีค่ามัธยฐานความกว้างเส้นน้อยกว่า 2 เท่า (ii) ความสูงของอักษรทั้งสองต่างกันไม่เกิน 2 เท่า (iii) ระยะห่างระหว่างสองวัตถุนั้นมีค่าไม่เกิน 3 เท่าของวัตถุที่กว้างที่สุดในสองตัวที่ใช้เปรียบเทียบ หลังจากการจัดกลุ่มนี้เราจะได้โซ่ของอักษรที่ถูกจัดกลุ่มเข้าด้วยกัน แต่ละโซ่ประกอบไปด้วยอักษรสองตัวที่ถูกจัดกลุ่ม ถัดจากนั้นแต่ละโซ่จะถูกรวมเข้าด้วยกันได้ด้วยเช่นกันหากโซ่อักษรมีอักษรในโซ่ของตนร่วมกันโซ่อื่น ๆ และทิศทางของโซ่มีความใกล้เคียงกัน

สุดท้ายขั้นตอนนี้จะจบลงเมื่อไม่มีโซ่อักษรใด ๆ ถูกเชื่อมต่อเพิ่มเติม ในที่สุดเราจะได้กลุ่มหรือโซ่ของอักษรที่เกิดจากการจัดกลุ่มด้วยความคล้ายคลึงของอักษรและทิศทางของข้อความ อีกนัยนึงคือเรา

ได้กลุ่มบรรทัดของแต่ละประโยคออกมาจากภาพถ่ายเรียบร้อยในขั้นตอนนี้

2.2 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) เป็นการสกัด Feature ของภาพโดยอาศัยรูปแบบ Histogram ของทิศทางเฉดสีในภาพ หรือ Gradient direction เพื่อพิจารณาลักษณะของวัตถุต่าง ๆ อย่างที่แสดงในภาพ 2.2 ด้วยความสามารถนี้ จึงมีการนำ HOG มาใช้สำหรับสกัด Feature เพื่อใช้ในงานจำพวก Object detection อย่างหลากหลาย ทั้ง การตรวจจับท่าทางของมือ [14] , การตรวจจับรถบนท้องถนน [15], การตรวจจับมนุษย์ในภาพ [16] และไม่เพียงแค่สามารถใช้กับงานตรวจจับวัตถุ แต่ยังสามารถใช้กับงานด้านตรวจหาข้อความในภาพได้เช่นกัน [17–19]

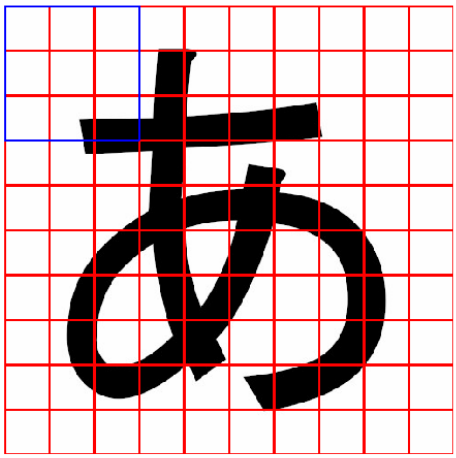


(a) (b)

รูปที่ 2.2: ตัวอย่าง Histogram of Oriented Gradients ก่อนแปลงสภาพกลายเป็นเวกเตอร์ของตัวอักษรภาษาญี่ปุ่น

การสกัด Feature ของ HOG นั้นทำได้โดยเริ่มจากแบ่งภาพเป็นส่วนเล็ก เรียกว่า Cell หรือช่องสีแดงตามที่แสดงในภาพ 2.3 จากนั้นสร้าง Histogram สำหรับ Cell นั้น ๆ ด้วยค่า Gradient direction และ Magnitude โดย Histogram นี้จะเป็นตัวแทนของลักษณะขอบและรูปร่างที่อยู่ภายใน Cell นั้น ๆ จากนั้นจะทำการ Normalization กับ Histogram ของแต่ละ cell ด้วยกลุ่มของ Cell หรือเรียกมันว่า Block อย่างที่เห็นเป็นช่องสีน้ำเงินในภาพ 2.3 สุดท้ายเราจะได้ Histogram ของ Gradient direction จากทุก ๆ Cell ของภาพซึ่งเป็นตัวแทนของรูปร่างวัตถุแต่ละส่วน ด้วย Histogram ที่ได้มาจะถูกนำไปเข้ากระบวนการ Vectorization เพื่อให้สามารถใช้ในงานอื่น ๆ ได้ต่อไป

อย่างไรก็ดีการที่จะได้ HOG ของวัตถุที่เราต้องการตรวจสอบจำเป็นต้องใช้ภาพของวัตถุนั้น ๆ ในการคำนวณ แต่ในสถานการณ์จริงภาพของวัตถุอาจอยู่ในภาพถ่ายขนาดใหญ่ที่ประกอบไปด้วยหลายวัตถุ เราจึงต้องตัดภาพของวัตถุเพื่อใช้คำนวณกับ HOG เราเรียกภาพส่วนย่อยที่ถูกตัดออกมานั้นว่า Patch ดังนั้นเราจำเป็นต้องใช้ Patch ที่มีขนาดใกล้เคียงกับวัตถุนั้น ๆ เป็นเหตุให้หากภาพมีขนาดใหญ่แต่วัตถุที่ต้องการตรวจพบมีขนาดเล็ก จำนวน Patch ก็จะมากขึ้น



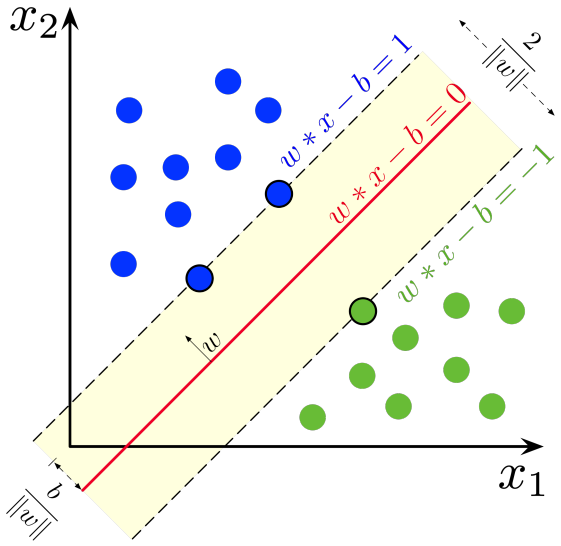
รูปที่ 2.3: Cell และ Block ในการทำงานของ Histogram of Oriented Gradients

ในกรณีของมังงะ ตัวอักษรมักมีขนาดเล็ก (20px – 40px โดยส่วนใหญ่อ้างอิงจาก Dataset ของเรา) เมื่อเทียบกับขนาดภาพมังงะ (1170px อ้างอิงจาก dataset ของเรา) ดังนั้นจำนวนของ Patch ที่ต้องสร้างและคำนวนด้วย HOG จึงมีมหาศาลและสร้างภาระแก่การคำนวน Feature โดย SVM อย่างมาก ซึ่งปัญหาส่วนนี้ส่งผลกระทบต่อความเร็วในการทำงานของเรา เราจึงใช้ SWT สำหรับสร้าง Patch แทนการ Sliced window ทีละส่วนบนภาพที่ต้องการใช้งาน

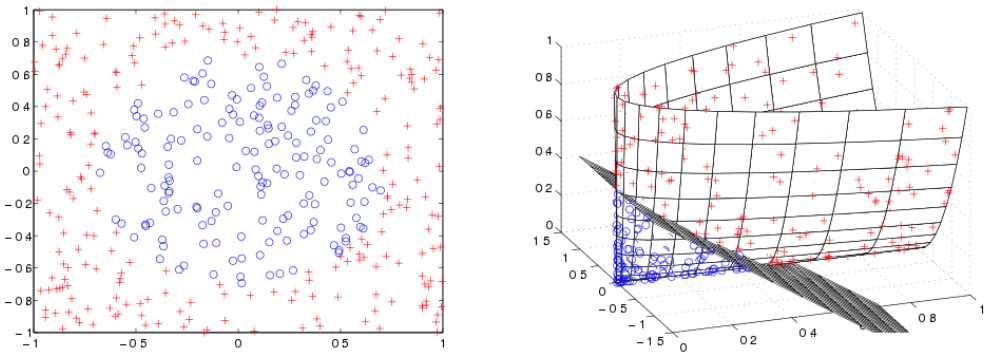
2.3 Support Vector Machine

Support Vector Machine (SVM) [20] นั้นเป็นเทคนิค Pattern recognition แบบ Supervised learning ซึ่งถูกใช้ทั้งในงานเพื่อ Classification และ Regression ซึ่งภายในงานนี้ได้ใช้งานเพื่อ Classification โดยทำงานด้วยการสร้าง Hyper-plane ที่เหมาะสมที่สุด (Optimal) เพื่อจำแนกแยกข้อมูลสองกลุ่มอย่างที่แสดงในภาพ 2.4 ซึ่งแตกต่างจาก Logistic regression ที่ไม่สามารถสร้างเส้นแบ่งข้อมูลระหว่างกลุ่มได้เหมาะสมที่สุดเหมือน SVM

เพื่อที่จะแยกข้อมูลทั้งสองกลุ่มด้วย Optimal hyper-plane นั้น $w \times x - b = 0$ จะทำหน้าที่แบ่งข้อมูลสองกลุ่มออกจากกันโดยมี Support vector ทำหน้าที่เป็นกันชนระหว่างข้อมูลที่ใกล้กันที่สุดระหว่างกลุ่มข้อมูลทั้งสอง ซึ่ง SVM นั้นจะสร้างพื้นที่การตัดสินใจขึ้นมา หรือก็คือพื้นระหว่าง $w \times x - b = 1$ และ $w \times x - b = -1$ โดยจะปรับให้ระยะห่างระหว่างทั้งสองนั้นกว้างที่สุดเท่าที่ทำได้ โดยระยะห่างนั้นมีค่าเท่ากับ $\frac{2}{\|\vec{w}\|}$ โดยใช้การ Minimize $\|\vec{w}\|$ อย่างไรก็ดี ในการแบ่งข้อมูลแบบ Non-linear นั้นสามารถใช้ Kernel เข้ามาช่วยในการเปลี่ยนมิติของข้อมูลเพื่อให้สามารถแบ่งแยกข้อมูลทั้งสองกลุ่มได้ด้วย Linear hyper-plan ตามที่แสดงในภาพ 2.5



รูปที่ 2.4: การแบ่งแยกกลุ่มข้อมูลด้วย Hyper-plane ของ SVM



รูปที่ 2.5: ตัวอย่างแสดงคุณสมบัติการเปลี่ยนมิติของข้อมูลด้วย Kernel

บทที่ 3
วิธีการทดลอง

ในบทนี้เรากล่าวถึงวิธีการใหม่ของเราที่ได้ปรับปรุงและพัฒนาขึ้นมาโดยใช้ SWT เป็นต้น เพื่อทำให้มันสามารถใช้งานร่วมกับภาพมังงะได้อย่างมีประสิทธิภาพ และดำเนินการทดลองเพื่อวัดประสิทธิภาพของวิธีการใหม่ของเราว่าสามารถทำงานได้ดีขึ้นหรือไม่อย่างไรเมื่อเปรียบเทียบกับวิธีการต้นฉบับ [1] นอกจากนี้เรายังนำวิธีการอื่น ๆ มาร่วมเปรียบเทียบเพิ่มเติม

3.1 วิธีการใหม่ที่ถูกปรับปรุงและพัฒนาเพิ่มเติม

สำหรับวิธีการอย่างที่ได้กล่าวไปในบทที่ 1 วิธีการใหม่ของเราได้ใช้ประโยชน์จาก SWT ร่วมกับความสามารถของ SVM โดยใช้ HOG เป็น Feature อย่างไรก็ดีตามที่ได้กล่าวไปในบทที่ 1 จุดประสงค์หลักของ SWT นั้นที่ถูกนำเสนอในงานวิจัยก่อนหน้าถูกออกแบบเพื่อการตรวจหาข้อความบนภาพถ่าย ด้วยเหตุผลนี้ทำให้การทำงานร่วมกับภาพมังงะไม่สามารถทำงานได้ดีอย่างที่ควร ก่อให้เกิด False positive จำนวนมาก ตามที่แสดงให้เห็นในภาพ 3.1 สาเหตุหลักคือความแตกต่างเชิงเอกลักษณ์ของวัถตุในภาพจริงและภาพวาดมังงะ นอกจากนี้องค์ประกอบในภาพวาดของมังงะนั้นยังมีความคล้ายคลึงกับลักษณะของตัวอักษรในภาพมากเกินไป เช่น เส้นของต้นหญ้า, เส้นผมของตัวละคน, และรายละเอียดบนพื้นหลัง อย่างที่แสดงในภาพ 3.1(a) และภาพ 3.1(b) ด้วยเหตุนี้วิธีการของเราจึงปรับปรุงขั้นตอนบางส่วนของการค้นหาวัตถุที่คล้ายคลึงอักษร, การจับกลุ่มอักษร, และเพิ่มขั้นตอนใหม่บางส่วนเพื่อให้สามารถใช้งานกับมังงะได้อย่างมีประสิทธิภาพมากขึ้น

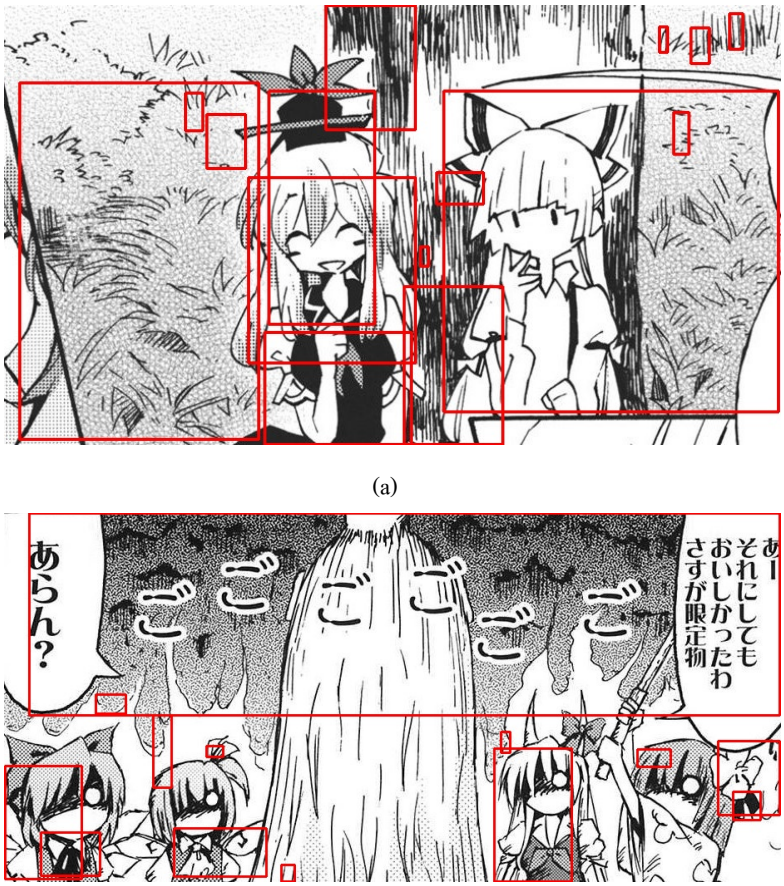
ความแตกต่างของวิธีต้นฉบับและวิธีการใหม่ของเรานั้นถูกแสดงให้เห็นในภาพ 3.2 อย่างที่เห็นในภาพ 3.2(b) เราได้เพิ่มขั้นตอนการคัดแยกตัวอักษรใหม่ขึ้นมาก่อนการจับกลุ่มตัวอักษรเข้าเป็นประโยค ขั้นตอนการคัดแยกนี้ใช้ความสามารถของ SVM classification เพื่อช่วยลด False positive ของผลลัพธ์การค้นหาวัตถุที่คล้ายตัวอักษรจากขั้นตอนก่อนหน้า

3.1.1 The Stroke Width Transform

ในขั้นตอนนี้ใช้วิธีการเดียวกับงานวิจัย [1] ที่กล่าวไปในบทที่ 2 โดยเรานำ SWT มาดำเนินการบนภาพมังงะเพื่อให้อยู่รูปแบบตัวดำเนินการ SWT ซึ่งข้อมูล Output ของ SWT นั้นเป็นเมทริกซ์มีขนาดเท่ากับภาพ Input ซึ่งจะใช้ในขั้นตอนต่อไป

3.1.2 ค้นหาวัตถุที่ใกล้เคียงอักษร

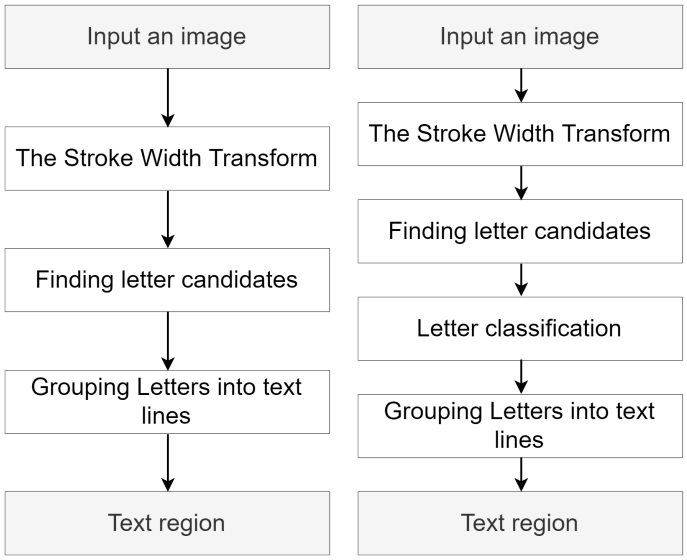
ในมังงะนั้น ข้อความหรืออักษรทั้งหลายมีขนาดที่หลากหลายและแตกต่างไปจากภาพถ่าย เราจึงต้องนำกฎเกณฑ์ที่ใช้ในการคัดกรองวัตถุกับตัวอักษรบนภาพถ่ายมาดัดแปลงให้เหมาะกับสภาพลักษณะของอักษรในมังงะ โดยกฎดังกล่าวถูกดังแปลงให้อยู่ในรูปแบบสมการ 3.1



(a)

(b)

รูปที่ 3.1: ตัวอย่างผลลัพธ์จากการตรวจหาข้อความบนภาพมังงะด้วยวิธีต้นฉบับ [1] ซึ่งแสดงให้เห็น False positive จำนวนมาก (a) นักวาด: Shinoasa (b) นักวาด: Kousei (Public Planet)



(a) (b)

รูปที่ 3.2: แผนผังการทำงานของ (a) วิธีการดั้งเดิม [1] และ (b) วิธีการใหม่ของเรา

$$f(d,h,w,\tilde{s}) = \begin{cases} 1, & \text{if } 1 < \frac{d}{\tilde{s}} < 15 \text{ and } \tilde{s} \leq 80 \text{ and} \\ & 5 < h, w < 50 \\ 0, & otherwise, \end{cases} \tag{3.1}$$

โดยที่ตัวแปรที่เพิ่มเข้ามาคือ w คือ ความกว้างของวัตถุนั้น ๆ

เมื่อเรากำจัดวัตถุที่ไม่มีลักษระแตกต่างจากอักษรอย่างสิ้นเชิงออกไปแล้ว เราจะได้กลุ่มของวัตถุที่มีลักษณะใกล้เคียงอักษร อย่างที่แสดงในภาพด้านล่าง ขั้นตอนนี้ของวิธีการต้นฉบับไม่สามารถรวบรวมวัตถุที่คล้ายคลึงอักษรได้ครบถ้วนเพียงพอตามที่แสดงให้เห็นในภาพ 3.3 แต่กฎที่ถูกปรับปรุงนั้นสามารถรวบรวมวัตถุที่คล้ายอักษรได้ครอบคลุมมากขึ้น อย่างไรก็ดีกฎใหม่นั้นสร้าง False positive ที่มากขึ้นตาม ซึ่งปัญหาดังกล่าวจะถูกแก้ไขด้วยขั้นตอนคัดแยกอักษรด้วย SVM ต่อไป

3.1.3 คัดแยกอักษรด้วย SVM

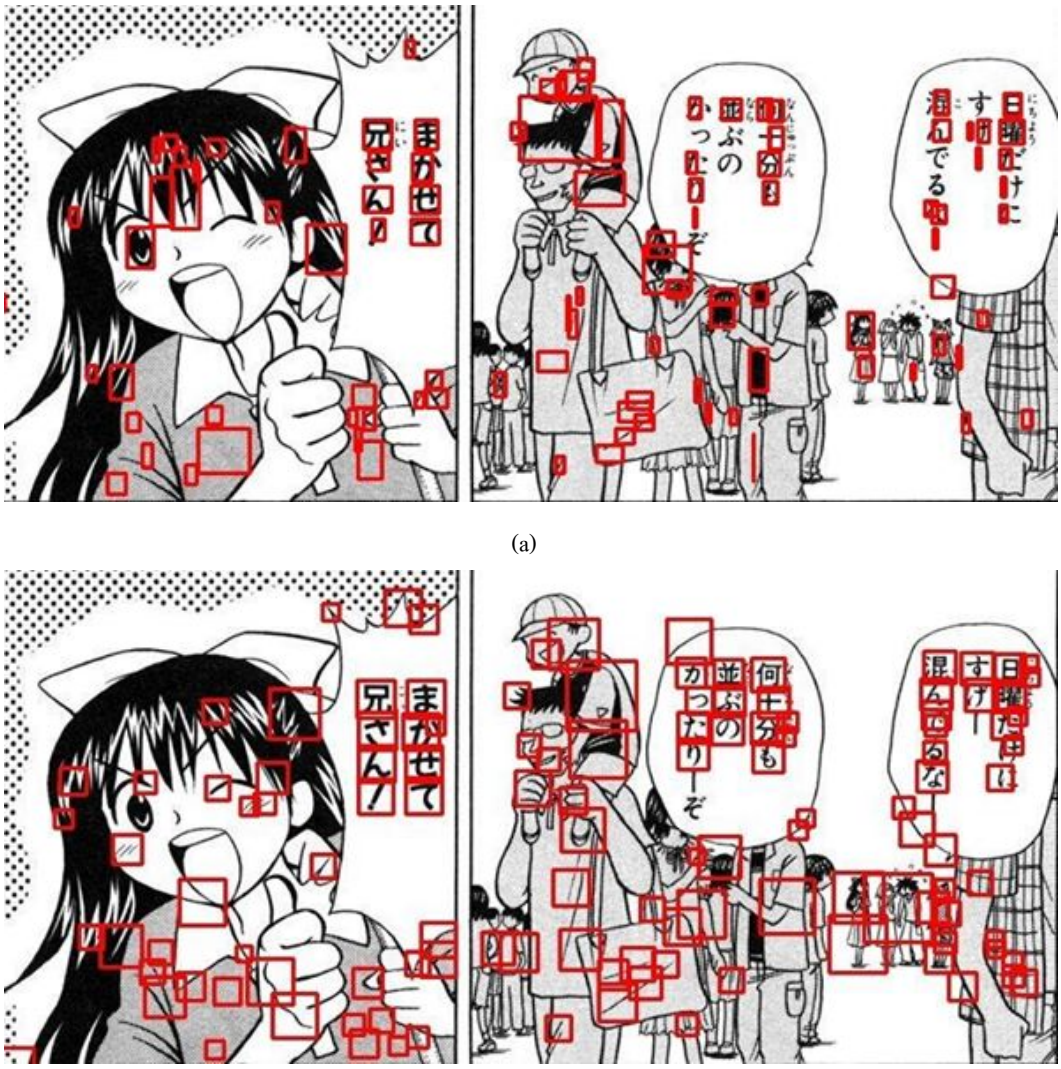
ในขั้นตอนนี้เราสร้างภาพขนาดเล็ก (Patch) ซึ่งใช้ตำแหน่งและขนาดของวัตถุที่คล้ายคลึงอักษรจากขั้นตอนก่อนหน้า โดยภาพขนาดเล็กเหล่านี้จะถูกคัดแยกเป็นกลุ่มที่เป็นอักษรและกลุ่มที่ไม่ใช่ซึ่งจะช่วยลด False positive ให้ต่ำลง

เราได้นำ SVM มาดำเนินการในขั้นตอนนี้ SVM คือ เทคนิค Supervised learning แบบหนึ่งซึ่งมักถูกใช้ในงานด้านคัดแยก (Classification) และ สมการถดถอยต่อเนื่อง (Regression) [20] สำหรับชุดข้อมูลสำหรับเทรนโมเดลของ SVM ในขั้นตอนนี้นั้นสร้างจากภาพขนาดเล็ก หรือ Patch โดยแบ่งออกเป็น ภาพที่เป็นอักษร (Positive) และ ภาพที่ไม่ใช่อักษร (Negative) อย่างทีแสดงในภาพ 3.4 ภาพขนาดเล็กเหล่านี้สร้างจาก Manga109 ซึ่งจะกล่าวหลังจากนี้ สำหรับภาพ Positive และ Negative ที่สร้างขึ้นมาจะถูกสกัดลักษณะเด่นหรือ Feature ด้วย HOG [14] ซึ่งเป็นเทคนิคการสกัดข้อมูลเชิงรูปร่างของวัตถุในภาพด้วยการพึ่งพาการกระจายตัวของทิศทางโทนสี โดย Feature ที่ถูกสกัดมานั้นมีเป็นข้อมูลรูปแบบเวกเตอร์ 2,916-dimension

เช่นเดียวกับข้อมูลสำหรับเทรน ภาพขนาดเล็กของวัตถุที่คล้ายคลึงอักษรจากขั้นตอนก่อนหน้าจะถูก HOG สกัด Feature ออกมาแล้วจึงนำไปให้ SVM ดำเนินการคัดแยกภาพที่เป็นอักษรและไม่ใช่ออกจากกัน หลังจากเสร็จสิ้นกระบวนการ ส่วนภาพที่เป็นอักษรจะถูกนำไปใช้ในการจัดกลุ่มอักษรในขั้นตอนต่อไป

3.1.4 จัดกลุ่มอักษรเป็นข้อความ

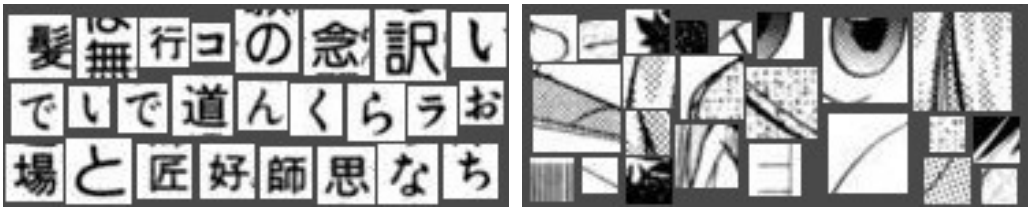
ในวิธีการต้นฉบับ [1] ขั้นตอนจัดกลุ่มวัตถุที่คล้ายคลึงอักษรเข้าด้วยกันได้ใช้หลักการเปรียบเทียบความคล้ายคลึงของตัวอักษร ประกอบไปด้วย ของความสูง ขนาดเส้น ทิศทาง และ ระยะห่าง เพื่อกำจัดข้อมูลรบกวนอื่น ๆ เช่น วัตถุที่ไม่ใช่อักษรที่กระจายอยู่ในภาพ โดยใช้สมมติฐานว่าประโยคหรือข้อความมักเกิดจากการรวมตัวกันของอักษรมากกว่าหนึ่งตัวและจัดเรียงอยู่ในทิศทางเดียวกันกับตัวอักษรอื่น ๆ ที่ขนาดใกล้เคียงกันตามที่ได้กล่าวไปใน 2 อย่างไรก็ดี วิธีการของเรานั้นได้กำจัดข้อมูลรบกวนเหล่านี้ออกไปแล้วในขั้นตอนการคัดแยกอักษรด้วย SVM ดังนั้นเราจึงใช้เพียงระยะห่างระหว่างอักษรเป็นปัจจัยในการจับกลุ่มอักษร



(a)

(b)

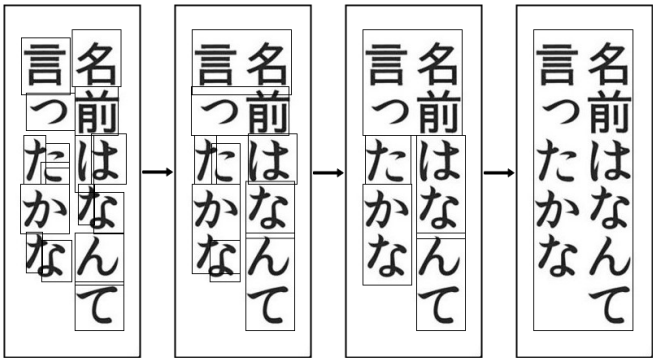
รูปที่ 3.3: ตัวอย่างแสดงการเปรียบเทียบผลลัพธ์ของขอบเขตข้อความที่ตรวจพบระหว่างวิธีการใหม่ (a) และวิธีการต้นฉบับ (b) ข้อมูลภาพถูกนำมาจากเรื่อง Arisa ©Yagami Ken



(a) (b)

รูปที่ 3.4: ตัวอย่างของ Patch: (a) ภาพ positive patches and (b) ภาพ negative patches

วิธีการจัดกลุ่มอักษรของเรานั้นจะใช้อักษรที่ถูกคัดแยกแล้วมาจัดกลุ่มเป็นประโยคโดยการจัดกลุ่มแต่ละอักษรที่อยู่ห่างกันไม่เกิน 1.5 เท่าของตัวอักษรที่แคบที่สุดของคู่อักษรที่ใช้เปรียบเทียบ หากตัวอักษรใดที่ห่างจากกันเกินกว่าค่าที่กำหนดจะถือว่าเป็นอักษรของคนละประโยคซึ่งจะไม่ถูกจับกลุ่มเข้ามา โดยตัวอย่างในภาพ 3.5 แสดงถึงตัวอย่างขั้นตอนการจับกลุ่มด้วยระยะห่าง นอกจากนี้แต่ละ



รูปที่ 3.5: ตัวอย่างแสดงการจับกลุ่มของตัวอักษร

ประโยคที่ถูกจับกลุ่มด้วยระยะห่างแล้วจะถูกพิจารณาขนาดด้วยเช่นกัน โดยแต่ละข้อความต้องมีพื้นที่ (ความกว้าง $\times$ ความสูง) เกินกว่า 2,550px อ้างอิงจากการทดลองกับชุดข้อมูลของเรา สุดท้ายเราจะได้กลุ่มของอักษรหรือประโยคข้อความจากภาพในมังงะออกมา

3.2 ชุดข้อมูลสำหรับการเทรนโมเดล SVM

เราได้นำภาพจากชุดข้อมูลภาพมังงะขนาดใหญ่ Manga109 [9] ประกอบไปด้วยภาพมังงะพร้อมข้อมูลประกอบ (Annotation) ของมังงะ 109 เรื่อง จัดทำโดยห้องทดลอง Aizawa Yamasaki แห่งมหาวิทยาลัยโตเกียว มังงะทั้งหมดในชุดข้อมูลนี้ถูกวาดโดยนักวาดมังงะมืออาชีพชาวญี่ปุ่นและถูกจัดจำหน่ายในช่วงปี 1970 ถึงปี 2010 แต่ละหน้ามังงะถูกระบุตำแหน่งของข้อความในภาพซึ่งเหมาะสำหรับการใช้เทรนโมเดลและทดสอบวิธีการของเรา

3.3 การทดลอง

เราดำเนินการทดลองในรูปแบบที่คล้ายคลึงกับงานวิจัยของAramaki et al. [12] ซึ่งจะช่วยให้เราสามารถเปรียบเทียบผลการทดลองประสิทธิภาพวิธีการของเรากับงานวิจัยอื่น ๆ ก่อนหน้าได้ เราได้เลือกภาพมังงะด้วยวิธีการสุ่มเลือก 100 หน้าสำหรับเทรน และอีก 100 หน้าสำหรับทดสอบประสิทธิภาพ โดยภาพมังงะทั้งหมดนี้ถูกสุ่มเลือกจากมังงะ 6 เรื่อง ได้แก่ *Aosugiru Haru, Arisa 2, Bakuretsu Kung Fu Girl, Dollgun, Love Hina, and Uchiha Akatsuki EvaLady*

เนื่องจากวิธีการของเราใช้ SVM ซึ่งต้องสร้างโมเดลสำหรับใช้งานคัดแยกภาพระหว่างภาพขนาดเล็ก (Patch) ระหว่างกลุ่มที่เป็นอักษรและไม่ใช่อักษรตามที่แสดงไปในภาพ 3.4 เราจึงได้สร้างชุดข้อมูลประกอบด้วยภาพอักษร 5,201 ภาพ และ ภาพขนาดเล็กอื่น ๆ ที่ไม่ใช่อักษรอีก 5,201 ภาพ กล่าวคือแบ่งเป็นข้อมูล Positive และ Negative ส่วนละ 50% เท่า ๆ กัน โดยภาพเหล่านี้ได้รับจากขั้นตอนการหาวัตถุที่คล้ายคลึงกับอักษรโดยใช้ภาพ 100 ภาพสำหรับเทรนเป็นข้อมูลนำเข้า

สำหรับ SVM เราใช้ Radial Basis Function Kernel โดย Hyperparameter ที่ร่วมใช้งานประกอบไปด้วย C และ γ เราใช้ Grid Search บนเครื่องคอมพิวเตอร์ *Google Cloud Compute Engine n1-highcpu-8* ในการค้นหาค่า Hyperparameter ที่ดีที่สุด ในช่วง 2^{-10} ถึง 2^{10} โดยได้ค่า C และ γ ที่ดีที่สุดที่ 2^5 และ

$2^{-6.75}$ ตามลำดับ โมเดลที่ถูกปรับปรุงให้เหมาะสม (Optimized) แล้วนี้จะถูกนำไปใช้ในขั้นตอนคัดแยกอักษร

การประเมินวิธีการตรวจหาข้อความที่เราปรับปรุงขึ้นใหม่นั้น เราได้ใช้รูปแบบการประเมินเดียวกับที่ใช้ใน ICDAR 2013 Robust Reading Competition [21] โดยถ้าอัตราส่วนระหว่างพื้นที่ Overlapped ต่อ พื้นที่ Ground-truth นั้นมากกว่าค่า t_p และอัตราส่วนระหว่างพื้นที่ Overlapped ต่อ พื้นที่ Detected region มากกว่า t_r ให้ถือว่า พื้นที่ระบุขอบเขตอักษรที่ถูกทำนายไว้นั้นถูกต้อง โดย t_p และ t_r นั้นมีค่าเท่ากับ 0.5 โดยอ้างอิงค่าตามงานวิจัยของ Aramaki et al. [12] สำหรับ Precision และ Recall เราได้คำนวณตามสมการดังต่อไปนี้ 3.2, 3.3 ตามลำดับ

$$P = \frac{\text{\#Correctly Detected Rectangles}}{\text{\#Detected Rectangles}} \tag{3.2}$$

$$R = \frac{\text{\#Correctly Detected Rectangles}}{\text{\#Rectangles of the Ground-truth}} \tag{3.3}$$

สำหรับ F-Measure เราคำนวนด้วยสมการดังนี้ 3.4

$$F = 2 \cdot \frac{P \cdot R}{P + R} \tag{3.4}$$

เราได้เปรียบเทียบผลการทดลองของวิธีการใหม่ของเราร่วมกับวิธีการต้นฉบับ [20] ซึ่งถูกใช้กับภาพถ่าย นอกจากนี้ยังเปรียบเทียบกับงานวิจัยก่อนหน้าอื่น ๆ ด้วย ดังนี้ Basic Grouping+ImageNet Classification model (BG+ImN) [12], Basic Grouping+Illustration2Vec model (BG+I2V) [12], Scene Text Detection (STD) [22], Speech Balloon Detection (SBD) [10], and Text Line Detection (TLD) [23] วิธีข้างต้นที่เรากล่าวถึงมีเทคนิคในการตรวจหาข้อความที่ต่างกัน เช่น การยึดหลักสมมติฐานพื้นฐาน (ทิศทางของข้อความ, รูปแบบการจัดวาง, ลักษณะของกล่องคำพูด) และ Convolutional nueral network

สำหรับ BG+ImN, BG+I2V, STD, SBD, และ TLD นั้นเราได้นำผลลัพทธ์การทดลองจากงานวิจัยของ Aramaki et al. [12] มาใช้ในการเปรียบของเราโดยตรง ซึ่งสามารถทำได้เนื่องจากเราได้ดำเนินการทดลองในสภาพแบบเดียวกับงานวิจัยดังกล่าว โดยผลลัพธ์การเปรียบเทียบและตัวอย่างขอบเขตข้อความที่วิธีการของเรานั้นสามารถตรวจพบถูกแสดงให้บทที่ 4

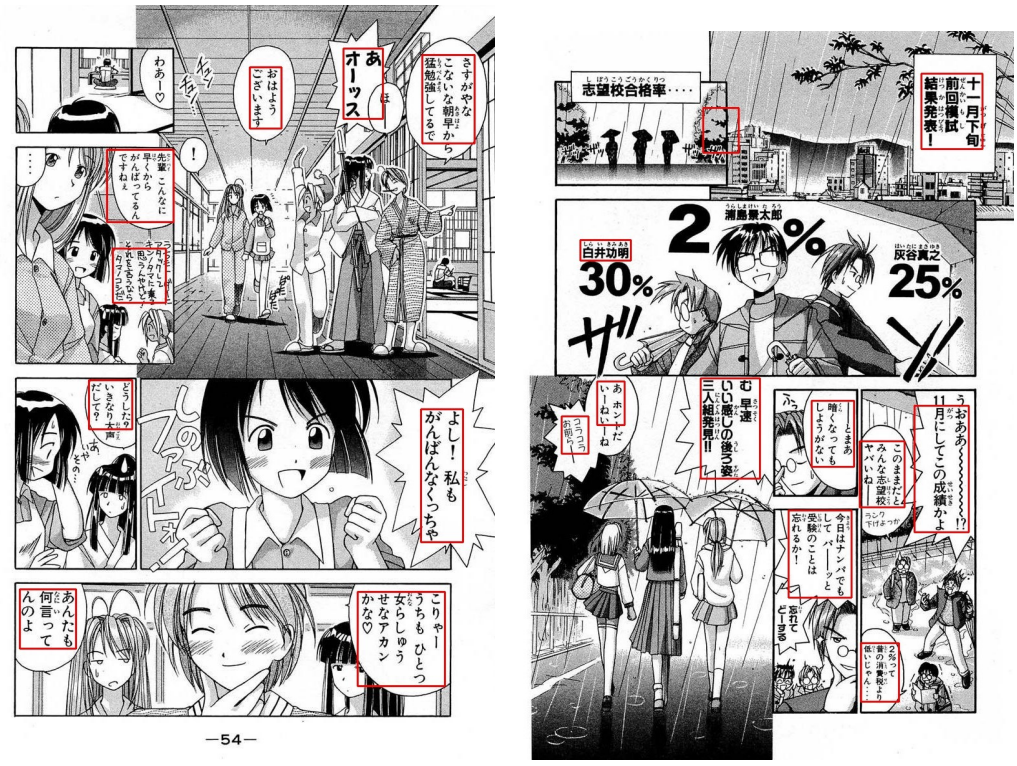
บทที่ 4
ผลการทดลอง

ผลลัพธ์การทดสอบประสิทธิภาพในวิธีการใหม่ของเราและการเปรียบเทียบกับงานวิจัยอื่น ๆ แสดงในตารางที่ 4.1 จากตารางดังกล่าว วิธีการของเราได้รับ F-measure สูงที่สุด ที่ 0.506 ซึ่งแสดงให้เห็นชัดเจนว่าวิธีการของเรามีประสิทธิภาพดีกว่าวิธีการต้นฉบับ [1] ยิ่งไปกว่านั้นวิธีการของเรายังได้รับ F-meaure ที่กว่าวิธีการ BG+ImN [12] และ BG+I2V [12] ซึ่งทั้งสองวิธีนี้ล้วนใช้เทคนิค Deep learning เป็นส่วนหนึ่งในการตรวจหาข้อความในภาพ อย่างไรก็ดี ค่า Precision และ Recall สูงสุดของการทดลองนี้อยู่ที่ 0.715 และ 0.481 เป็นของ BG+I2V และ BG+ImN ตามลำดับ สำหรับตัวอย่างขอบเขตของข้อความที่วิธีการใหม่ของเราตรวจพบถูกแสดงในภาพ 4.1

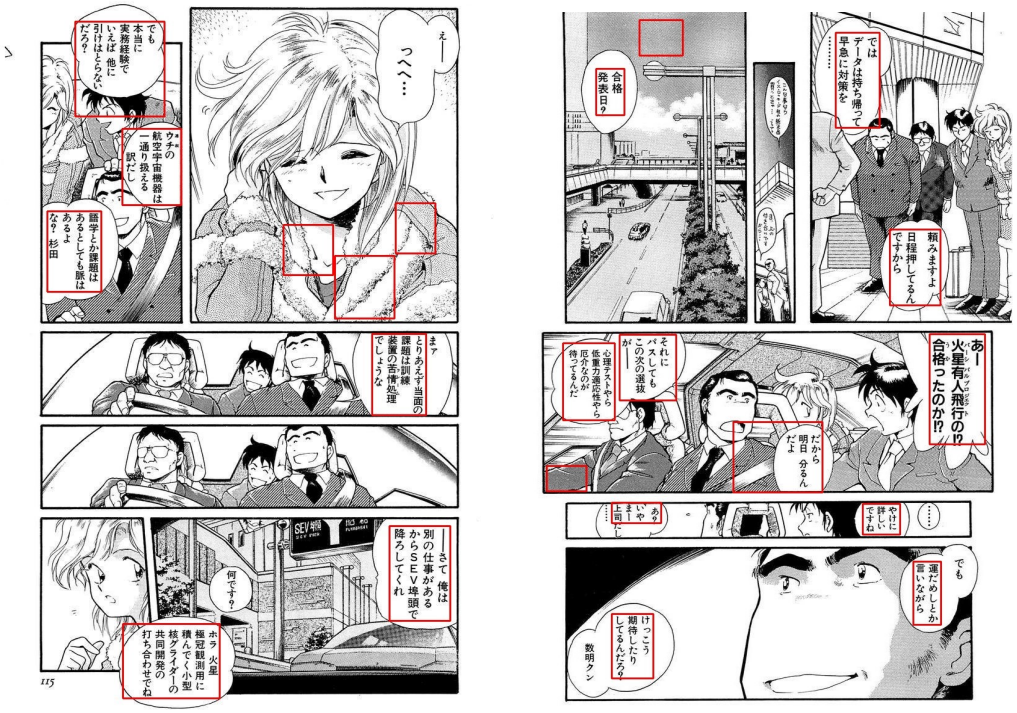
Method	Precision	Recall	F-measure
STD [22]	0.165	0.051	0.078
SBD [10]	0.180	0.102	0.130
TLD [23]	0.095	0.095	0.095
BG + ImN [12]	0.451	**0.481**	0.466
BG + I2V [12]	**0.715**	0.191	0.301
Baseline [1]	0.068	0.336	0.113
Our method	0.564	0.458	**0.506**

ตารางที่ 4.1: ตารางแสดงการเปรียบเทียบประสิทธิภาพของวิธีการใหม่ของเราร่วมกับวิธีการอื่น ๆ

เป็นที่น่าสนใจอย่างมากที่วิธีการของเราสามารถทำงานได้ดีกว่าเทคนิค Deep learning ทั้งสองวิธี สมมติฐานแรกคือ BG+ImN นั้นใช้ ImageNet Classification Model [24] ซึ่งถูกเทรนบนภาพถ่ายของวัตถุจริง อย่างไรก็ดีภาพวาดมังงะของวัตถุต่าง ๆ นั้นมีความแตกต่างจากภาพวัตถุจริงอย่างชัดเจน ซึ่ง ณ จุดนี้ทำให้วิธีการนี้ไม่สามารถทำงานได้เต็มประสิทธิภาพ อีกวิธีหนึ่งที่ใช้ Deep learning คือ BG+I2V ถึงแม้ว่าวิธีการนี้จะได้รับ Precision สูงที่สุดในการทดลองของเรา แต่คะแนน Recal นั้นต่ำกว่าทั้ง BG+ImN และวิธีการของเรา วิธีการนี้ใช้โมเดล Illustration2Vec [25] เป็นโมเดลสำหรับคัดแยกข้อความจากวัตถุอื่น ๆ ส่วนของโมเดลนั้นถูกเทรบนภาพวาด Anime (ภาพการ์ตูนแบบญี่ปุ่น) และภาพมังงะจากหลากหลายแหล่ง ประกอบไปด้วย Danbooru และ Safebooru ซึ่งมีลักษณะงานคล้ายกับวิธีการของเราในเชิงข้อมูล แต่โมเดลนี้ถูกออกแบบมาเพื่อนการทำนายป้ายกำกับ (Tag Prediction) และค้นหาภาพที่คล้ายคลึงกัน ดังนั้นนี่จึงอาจเป็นเหตุผลว่าทำไมโมเดลนี้จึงไมมีประสิทธิภาพเท่าที่ควรในการทดลองนี้



(a) (b)



(c) (d)

รูปที่ 4.1: ตัวอย่างขอบเขตข้อความmujวิธีการของเราตรวจพบ (a–b) Love Hina ©Ken Akamatsu และ (c–d) Eva Lady ©Miyone Shi.

บทที่ 5
สรุปผล

ในการทดลองนี้ เราได้เสนอวิธีการตรวจหาข้อความบนภาพมังงะด้วยเทคนิค SWT ร่วมกับการใช้ SVM และ HOG ในการลด False positive ที่เกิดขึ้น การทดลองของเราดำเนินการบนข้อมูลจาก Manga109 ซึ่งเป็นชุดข้อมูลภาพมังงะที่ถูกระบุ Annotation มาเรียบร้อยแล้ว วิธีการของเรานั้นสามารถทำงานได้ผล F-measure ที่ดีที่สุดในการเปรียบเทียบกับวิธี Baseline และวิธีอื่น ๆ รวมถึงวิธีการที่ใช้ Deep learning ถึงแม้ว่างานของเรานั้นซึ่งได้ทดสอบบนการ์ตูนญี่ปุ่นสามารถทำงานได้เป็นผลดีเยี่ยม อย่างไรก็ดีวิธีการของเรานั้นยังต้องมีการพัฒนาและค้นคว้าเพิ่มเติมเพื่อปรับปรุงประสิทธิภาพและสามารถใช้งานร่วมกับภาษาอื่น ๆ ได้

บรรณานุกรม

[1] B. Epshtein, E. Ofek, and Y. Wexler, “Detecting text in natural scenes with stroke width transform,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA, USA, Jun 2010, pp. 2963–2970.

[2] H. Yanagisawa, T. Yamashita, and H. Watanabe, “A study on object detection method from manga images using CNN,” in *Proceedings of the International Workshop on Advanced Image Technology (IWAIT 2018)*, Chiang Mai, Thailand., Jan 2018, pp. 1–4.

[3] X. Liu, C. Li, H. Zhu, T.-T. Wong, and X. Xu, “Text-aware balloon extraction from manga,” *The Visual Computer*, vol. 32, no. 4, pp. 501–511, Apr 2016. [Online]. Available: https://doi.org/10.1007/s00371-015-1084-0

[4] X. Pang, Y. Cao, R. W. Lau, and A. B. Chan, “A robust panel extraction method for manga,” in *Proceedings of the 22nd ACM International Conference on Multimedia (MM 2014)*. New York, NY, USA: ACM, 2014, pp. 1125–1128. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654990

[5] Y. Aramaki, Y. Matsui, T. Yamasaki, and K. Aizawa, “Interactive segmentation for manga using lossless thinning and coarse labeling,” in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA 2015)*, Hung Hom, Kowloon, Hong Kong, Dec 2015, pp. 293–296.

[6] T. Ogawa, A. Otsubo, R. Narita, Y. Matsui, T. Yamasaki, and K. Aizawa, “Object detection for comics using manga109 annotations,” *CoRR*, vol. abs/1803.08670, 2018. [Online]. Available: http://arxiv.org/abs/1803.08670

[7] S. Kovanen and K. Aizawa, “A layered method for determining manga text bubble reading order,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2015)*, Quebec City, QC, Canada, Sept.

[8] Y. Matsui, T. Shiratori, and K. Aizawa, “Drawfromdrawings: 2D drawing assistance via stroke interpolation with a sketch database,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1852–1862, 2017.

[9] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, “Sketch-based manga retrieval using manga109 dataset,” *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21 811–21 838, Oct 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-4020-z

[10] H. Tolle and K. Arai, “Manga content extraction method for automatic mobile comic content creation,” in *Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS 2013)*, Bali, Indonesia, Sept 2013, pp. 321–328.

[11] C. Rigaud, T. Le, J. . Burie, J. Ogier, S. Ishimaru, M. Iwata, and K. Kise, “Semi-automatic text and graphics extraction of manga using eye tracking information,” in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, Santorini, Greece, April 2016, pp. 120–125.

[12] Y. Aramaki, Y. Matsui, T. Yamasaki, and K. Aizawa, “Text detection in manga by combining connected-component-based and region-based classifications,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2016)*, Phoenix, AZ, USA, Sept 2016.

[13] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[14] W. T. Freeman, W. T. Freeman, M. Roth, and M. Roth, “Orientation Histograms for Hand Gesture Recognition,” in *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 1994, pp. 296–301. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.618

[15] S. Bougharriou, F. Hamdaoui, and A. Mtibaa, “Linear SVM classifier based HOG car detection,” in *Proceedings of the 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA 2017)*, Monastir, Tunisia, Dec 2017, pp. 241–245.

[16] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA, USA, Jun 2005, pp. 886–893.

[17] D. Wang, H. Wang, D. Zhang, J. Li, and D. Zhang, “Robust scene text recognition using sparse coding based features,” *CoRR*, vol. abs/1512.08669, 2015. [Online]. Available: http://arxiv.org/abs/1512.08669

[18] S. Tian, S. Lu, B. Su, and C. L. Tan, “Scene text recognition using co-occurrence of histogram of oriented gradients,” in *2013 12th International Conference on Document Analysis and Recognition*, Washington, DC, USA, Aug 2013, pp. 912–916.

[19] A. K. Sah, S. Bhowmik, S. Malakar, R. Sarkar, E. Kavallieratou, and N. Vasilopoulos, “Text and non-text recognition using modified hog descriptor,” in *2017 IEEE Calcutta Conference (CALCON)*, Dec 2017, pp. 64–68.

[20] J. Suykens and J. Vandewalle, “Least squares support vector machine classifiers,” *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, Jun 1999. [Online]. Available: https://doi.org/10.1023/A:1018628609742

[21] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*, Kolkata, India, Aug 2013, pp. 1484–1493.

[22] L. Gómez and D. Karatzas, "Multi-script text extraction from natural scenes," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)*, Washington, DC, USA, Aug 2013, pp. 467–471.

[23] C. Rigaud, D. Karatzas, J. Van De Weijer, J.-C. Burie, and J.-M. Ogier, "Automatic text localisation in scanned comic books," in *Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, Barcelona, Spain, Feb 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00841492

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS 2012)*, vol. 1. Lake Tahoe, Nevada, USA: Curran Associates Inc., Dec 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[25] M. Saito and Y. Matsui, "Illustration2Vec: A semantic vector representation of illustrations," in *SIGGRAPH Asia 2015 Technical Briefs*. Kobe, Japan: ACM, Nov 2015, pp. 5:1–5:4. [Online]. Available: http://doi.acm.org/10.1145/2820903.2820907