Thai Scene Text Recognition

Mr. Thananop Kobchaisawat

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Computer Engineering
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2019

การรู้จำข้อความภาษาไทยในภาพถ่าย

นายธนานพ กอบชัยสวัสดิ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Thesis Title	Thai Scene Text Recognition
By	Mr. Thananop Kobchaisawat
Field of Study	Computer Engineering
Thesis Advisor	Associate Professor THANARAT CHALIDABHONGSE, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Doctor of Philosophy

.... Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

DISSERTATION COMMITTEE

.... Chairman

(Professor BOONSERM KIJSIRIKUL, D.Eng.)

.... Thesis Advisor

(Associate Professor THANARAT CHALIDABHONGSE, Ph.D.)

.... Examiner

(Assistant Professor SUKREE SINTHUPINYO, Ph.D.)

.... Examiner

(Associate Professor CHOTIRAT RATANAMAHATANA, Ph.D.)

.... External Examiner

(Supakorn Siddhichai, Ph.D.)

ธนานพ กอบชัยสวัสดิ์ : การรู้จำข้อความภาษาไทยในภาพถ่าย. (Thai Scene Text Recognition) อ.ที่ปรึกษาหลัก : รศ. ดร.ธนารัตน์ ชลิดาพงศ์

การระบุตำแหน่งและรู้จำข้อความจากภาพถ่ายโดยอัตโนมัติ สามารถนำไปใช้ประโยชน์ได้หลากหลายในชีวิตประจำวัน เช่น การอ่านป้ายบอกทาง ฉลากสินค้า และการช่วยเหลือคนพิการทางการมองเห็น การอ่านข้อความจากภาพถ่ายนั้น มีความแตกต่างจากภาพเอกสารในหลายแง่มุม เช่น ความหลากหลายของรูปแบบอักษร การเรียงตัวของข้อความและสภาพแสงที่คาดเดาได้ยาก ปัญหานี้สามารถแบ่งได้เป็น 2 ปัญหาย่อยคือ การระบุตำแหน่งข้อความและการอ่านข้อความจากภาพถ่าย ขั้นตอนวิธีการระบุตำแหน่งข้อความที่เสนอ ใช้หลักการจำแนกประเภทระดับจุดภาพร่วมกับการบ่งบอกบริเวณของข้อความ และการเรียนรู้เชิงลึกแบบคอนโวลูชันทั้งหมด วิธีการที่นำเสนอนั้นสามารถตรวจจับข้อความได้ไม่จำกัดภาษา โดยไม่จำกัดรูปแบบ ผลการทดลองด้วยวิธีที่นำเสนอบนชุดข้อมูลทดสอบมาตรฐานแสดงให้เห็นถึงประสิทธิภาพที่ดีขึ้นทั้งในด้านความแม่นยำและความเร็วเมื่อเทียบกับวิธีอื่นๆ ส่วนภาพของข้อความจะถูกตัดแบ่งเพื่อเข้าสู่ขั้นตอนวิธีการรู้จำข้อความจากภาพถ่าย ประกอบไปด้วย 4 ขั้นตอนคือ การแปลงสภาพ การสกัดคุณลักษณะสำคัญ การสกัดคุณลักษณะของลำดับและการทำนาย ขั้นตอนที่เสนอถูกออกแบบเป็นโมเดลการเรียนรู้เชิงลึก แบบสามารถเรียนรู้ได้ทั้งหมดร่วมกับกลไกจุดสนใจแบบหลายระดับ ตามรูปแบบการเขียนในภาษาไทย โดยใช้ชุดข้อมูลสอนจากภาพข้อความที่สร้างขึ้นจากรูปแบบตัวอักษร ร่วมกับขั้นตอนวิธีที่ทำให้ภาพข้อความใกล้เคียงกับที่ปรากฏในภาพถ่าย ผลการทดลองบนชุดข้อมูลทดสอบ แสดงให้เห็นถึงประสิทธิภาพ ความแม่นยำ และผลกระทบของส่วนต่างๆในขั้นตอนวิธีที่นำเสนอ

สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต	....
ปีการศึกษา	2562	ลายมือชื่อ อ.ที่ปรึกษาหลัก	....

5871451121 : MAJOR COMPUTER ENGINEERING

KEYWORD: Thai Character Recognition, Thai Text Recognition, Scene Text Detection, Scene Text Recognition, Optical Character Recognition, Convolutional Neural Network

Thananop Kobchaisawat : Thai Scene Text Recognition. Advisor: Assoc. Prof. THANARAT CHALIDABHONGSE, Ph.D.

Automatic scene text detection and recognition can benefit a large number of daily life applications such as reading signs and labels, and helping visually impaired persons. Reading scene text images becomes more challenging than reading scanned documents in many aspects due to many factors such as variations of font styles and unpredictable lighting conditions. The problem can be decomposed into two sub-problems: text localization and text recognition. The proposed scene text localization works at the pixel level combined with a new text representation and a fully-convolutional neural network. This method is capable of detecting arbitrary shape texts without language limitations. The experimental results on the standard benchmarks show the performance in terms of accuracy and speed compared to the existing works. The cropped text instances are passed into the proposed text recognition algorithm, which consists of four stages: transformation, feature extraction, sequence modeling, and prediction. The proposed method is designed based on a fully-learnable deep learning-based model in combination with multi-level attention, which inspires from Thai writing system. The training data is purely synthesized from various fonts and novel techniques to make the generated images looked sensible. The experimental results on the test dataset show excellent accuracy and inference time.

Field of Study:	Computer Engineering	Student's Signature ....
Academic Year:	2019	Advisor's Signature ....

ACKNOWLEDGEMENTS

I would like to show my greatest appreciation to my advisor, Assoc. Prof .Dr. Thanarat Chalidabhongse. Without her enthusiasm, patient, encouragement, support, and endless optimism, this dissertation would hardly have been completed. She is not only the greatest advisor but also the best mentor I have ever had.

I would like to thank my dissertation committee: Prof. Dr. Boonserm Kitsirikul, Assoc. Prof. Dr. Chotirat Ratanamahatana, Asst. Prof. Dr. Sukree Sinthupinyo, and Dr. Supakorn Siddhichai for their valuable comments and suggestions.

To my lab-mates and all previous students, thank you for listening, offering me advice, and supporting me through this journey.

To both of my greatest teachers, Mr.Surachai Piyaprapaphan and Ms.Warunee Damrongwong, thank you for valuable knowledge, mentoring, and guidance. Without your guidance and never-ending encouragement to pursue in the computer field, this dissertation would hardly have been completed.

I appreciate and gratefully acknowledge the doctoral scholarship from the 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship.

Most importantly, I would like to thank my family for their moral support, never-ending warm encouragements, raising me to value education, and inspiring me to follow my dreams.

Thananop Kobchaisawat

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

1. Introduction

Communication is a process of sending and receiving information. The ways of communication among people are not only limited to face-to-face oral conversation but also textual communication through various sources, for example, books, internet contents, and signs. On a daily basis, several kinds of meaningful information can be retrieved from text in images, such as product labels, billboards, and traffic signs. If there is an algorithm that can automatically extracts and captures this useful information is designed, a large number of applications can be created.

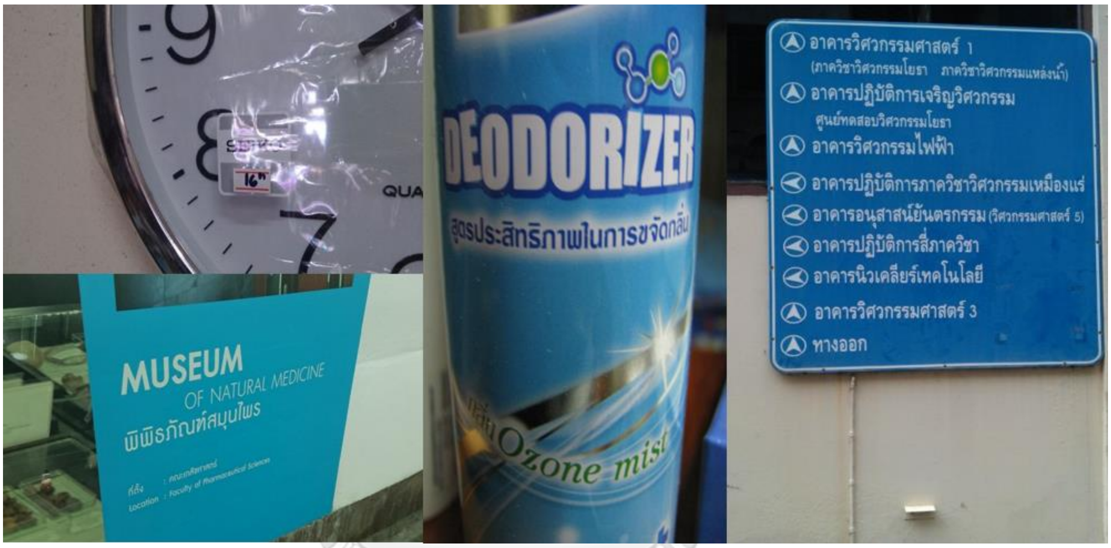

SEIKO
16"
QUA
MUSEUM
OF NATURAL MEDICINE
พิพิธภัณฑ์สมุนไพร
DEODORIZER
สูตรประสิทธิภาพในการขจัดกลิ่น
Ozone mist
อาคารวิศวกรรมศาสตร์ 1
(ภาควิชาวิศวกรรมโยธา ภาควิชาวิศวกรรมแหล่งน้ำ)
อาคารปฏิบัติการเจริญวิศวกรรม
ศูนย์ทดสอบวิศวกรรมโยธา
อาคารวิศวกรรมไฟฟ้า
อาคารปฏิบัติการภาควิชาวิศวกรรมเหมืองแร่
อาคารอนุสาสน์ยันตรกรรม (วิศวกรรมศาสตร์ 5)
อาคารปฏิบัติการสี่ภาควิชา
อาคารนิวเคลียร์เทคโนโลยี
อาคารวิศวกรรมศาสตร์ 3
ทางออก

Figure 1 Samples of daily life text images

Automatic language translation from images is one of real-wold scene text recognition applications. In this decade, smartphones have become extremely popular among all age groups. Automatic sign translation apps can help tourists understand the local languages and make them more comfortable communicating with locals.

Visually impaired and elderly people also gain benefits from automated reading applications. The mobile application named “Be My Eye” can help blind, elderly, and low vision people see things in their daily life routine, such as objects identification, reading product signs, or identify products’ expiry dates. Even so, this app still needs sighted volunteers to assist them through live video chat. If an autonomous scene text reading algorithm can be built, it could help these people have a better quality of life.

Many repeated routines can earn benefits from an automated scene text reading, such as license plate recognition, traffic sign surveillance, and video captioning. License plate recognition is one of the real-world applications used in public places. However, many designed systems still rely on specific working conditions. For example, working reliably only under good indoor lighting scenes with a fixed position camera. Improving scene text localization and recognition algorithms can help these systems to be able to function under more arbitrary scene conditions.

Robotics is another area that requires ability to sense and understand surrounding environment including scene text. An autonomous driving car is one of the most potential robotic applications. It needs an ability to locate and perceive the surrounding text from traffic and road signs.

Optical Character Recognition (OCR) is a long-standing problem in the document analysis field. Documents in the form of books, newspapers, government, and historical documents are the sources of printed characters. The main aim of OCR is to digitize these documents into an electronic form. The standard OCR algorithms focus on recognizing texts in scanned documents, which have many specific characteristics like simple black text on white background, horizontal text alignment, and fixed page layout in the printed text. Nonetheless, scene text images are usually taken by smartphone cameras, with arbitrary viewpoint under uncontrollable lighting and probably shaking. These lead to many challenges such as a non-uniform motion blur problem. These extensive properties make the existing scanned document-based OCR may not work well on scene text situations. Figure 2 shows comparison between scanned document and scene text images.

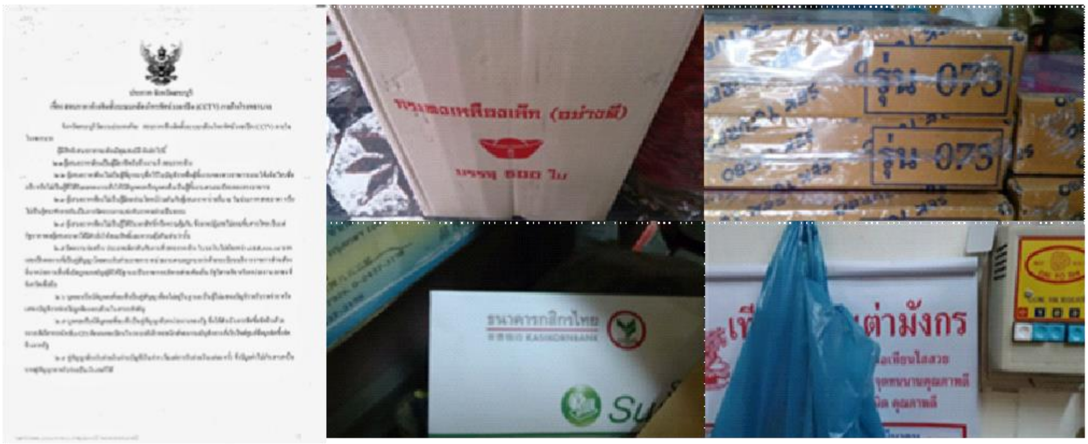


Figure 2 Image comparison between scanned document and scene text images

According to the literature survey on scene text recognition, this problem consists of two subproblems: scene text localization and recognition. The main aim for scene text localization is to locate texts in images into precise word or line level bounding boxes. There are many existing English scene text localization methods. Nevertheless, these methods may not work well on Thai scene text due to some language-specific properties. In Thai, writing system a text lines consists of alphabets, vowels and tone marks can be written in 4-levels: top, upper, main, and below base line. Since tone marks are usually much smaller than other components in text lines, the typical English based scene text localization algorithms often suppress these components, leading to incorrect recognition output. The comparison between English and Thai writing systems is shown in Figure 3.

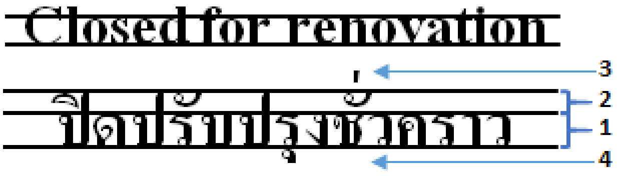


Figure 3 Text layout and writing systems comparison between Thai and English

To the best of our knowledge, there is no Thai scene text recognition algorithm that is accurate. Since the existing Thai character recognition algorithms mostly designed for scanned documents problem; they may not be suitable reading scene texts. A new algorithm for Thai language needs to be designed to achieves a good performance on Thai scene text recognition problem.

1.1. Aims and Objectives

The main objective of this dissertation is to build an automatic Thai text recognition algorithm which will achieve better results than traditional ThaiOCR on Thai scene text problem.

1.2. Scope of Study

- Thai text recognition limits to Thai characters consisting of consonants, vowels, and tone marks

- The proposed algorithm will be tested on BEST 2015 and our Thai scene text dataset, which have the following constraints:

- Text in testing images must be taken under good lighting conditions.
- Text in testing images must be written in a horizontal, not vertical, direction.
- Text in testing images must not be affected by motion blur.
- Text in images must have height not less than 32 pixels.

2. Literature Survey

This chapter describes the survey of literature related to this dissertation. It begins with works in scene text localization and recognition, then, Thai OCR, and Thai scene text localization.

2.1. Scene Text Localization and Recognition

Scene text recognition is a computer vision problem that can be dated back to decades. At high-level perception, this problem consists of two sub-problems: text localization and text recognition. The main aim of text localization is to locate individual words or lines of text. Once the text regions in the image have been spotted, those regions can be identified as the actual words and lines, which can be fed to text recognition algorithm.

2.1.1. Scene Text Localization

The process of locating text lines in natural scene images is different from the one to locate text lines in scanned documents since scene images consist of complex components and various kinds of challenges. Text regions can be different in styles, sizes, and colors. In addition, a large variety of font styles, sizes, a wide range of complex backgrounds, and occlusions makes the problem more challenging.

According to the existing scene text localization and recognition, text localization algorithms can be categorized into two groups: connected component analysis (CCA) and object or pixel classification based approaches.

2.1.1.1. Connected Component Analysis Based Approaches

The scene text localization based on connected component analysis (CCA) typically can be treated as a graph-based algorithm. This kind of algorithms typically use colors, textures, edge intensity, and stroke-width as features. Each set of connected components is grouped into text regions based on feature correspondence heuristics, such as color similarity and text line location. There are many interesting CCA based text localization methods, which are mentioned below.

(Subramanian et al., 2007) proposed a CCA based scene text recognition algorithm. A horizontal line scan was applied to find the regions which have text-like properties from intensity image. Then, seed-growing, color, and spatial location features were fed to apparent rules to classify each text region into text bounding boxes.

In (Zongyi & Sarkar, 2008) work, the method began with initial segmentation by using the Niblack binarization technique and some rules to discard noise and non-text regions. Then, intensity and shape filtering was applied to acquire text areas.

In (Minhua & Chunheng, 2008) work, a background complexity analysis was applied together with CCA based text detection method. First, the gray density was calculated from a grayscale input image to classify background into three classes: simple, middle, and complex background. Then, the edge maps were extracted and thresholded by using a defined algorithm based on the background classification result. Finally, the rule-based text refinement was employed on each candidate CCs to discard non-text components.

(Shivakumara et al., 2009) employed a gradient-based feature for text detection. In this work, the assumption that the gradient information in text areas was different from non-text regions in terms of higher contrasts was proposed. The gradient difference feature in the x-direction was extracted from the grayscale image, obtaining the minimum and maximum gradient values. The global threshold was then determined based on the average value of gradient difference to suppress non-text regions. The output text bounding boxes can be acquired by applying a conventional projection profile on the threshold image.

(Epshtein et al., 2010) introduced the famous stroke width transform (SWT) for scene text localization. This method used the Canny edge detector (Canny, 1986) to extract the edge map from the input image. Then, for each location, the stroke width feature was computed from the gradient direction of the edge map. The modified CCA and rule-based techniques were used in combination with stroke width feature and CCs (Connection Components) spatial location to generate candidate text bounding boxes.

(Karaoglu et al., 2010) employed the statistical binarization technique to suppress most background from the input image. Then, geometric, shape regularity and corner based interpolated features were extracted from each CCs. To classify between text and non-text CCs, the extracted features were fed into a random forest classifier trained on ICDAR 2003 and 2005 dataset. Finally, the output text bounding boxes were built by using spatial location feature combining with statistical rules.

(X. Huang & Ma, 2010) stated an assumption that the edge feature of scene text regions is richer and denser than the background area. The novel edge features were retrieved by using six directions of the 2D Log-Gabor filter to build a text stroke map. Then, coarseness feature and appearance-based rules were employed to filter out non-text CCs. Finally, Harris corner detection and CCA were applied to produce candidate text regions.

(Lee et al., 2010) proposed a color-based text localization method. K-Mean clustering was used to find the most dominant colors from the input image. In this work, the hypothesis that the text region in the image is generally relatively small, only K-means clustering based on the color distribution often yields insufficient segmentation, was stated. The modified version of K-Mean clustering, which utilizes the HCL (hue, chroma, and luminance) distance was applied to find segmented regions. Each pixel was assigned to one of K labels, and the eight-connected pixels were merged into the same region. Markov Random Field (MRF) uses locations, sizes, and shapes features to determine the textness score in the text region verification stage. Nevertheless, in this work, the text bounding box generation algorithm was not mentioned.

(Lukas Neumann & Matas, 2010) proposed the novel Maximally Stable Extremal Region (MSER) based text localization method. MSER is a famous feature extractor which applies multiple binarization thresholding on the input image and keeps the satisfying regions. For each extremal region (ER) in the MSER set, the following scale-invariant features were extracted: aspect ratio, relative segment height, compactness number of holes, convex hull area to surface ratio, character color consistency, background color consistency, and skeleton length to perimeter ratio. A standard Radial Basis Function (RBF) Support Vector Machine (SVM) was used as text and non-text classifier for each ER. A horizontal text line was treated as a linear sequence of characters with a straight or slightly curved bottom line in this work. From this assumption, the candidate text lines were built using spatial character location, size, and color feature in combination with Least Median Square (LMS) fitting to generate output text lines.

The updated version of MSER based text localization algorithm was proposed by (Lukas Neumann & Matas, 2012). Multiple MSER components were extracted from seven channels of image: R,G,B, H,S,I, and intensity gradient channels. The descriptor for each ER was built by the following incrementally computed descriptors: area (pixel), bounding box spatial location, perimeter, and horizontal line crossing. These features were used in Real AdaBoost text classifier to filter out weak text CCs. In the second stage, the ERs that pass the first stage were arranged into character and non-character classes using SVM and more computationally expensive scale-invariant features: hole area ratio and convex hull ratio. Finally, the output text lines were built in the same way proposed in (Lukas Neumann & Matas, 2010).

(Yin et al., 2012) employed MSER to extract the candidate letter regions. For each pair candidate letter region, the text region was built by using the following letter adjacency features: letter candidate's widths, heights, centroids, colors, stroke widths, top, and bottom alignments. These candidate regions were classified by AdaBoost using horizontal and vertical variances feature to acquire the text bounding boxes.

(Mosleh et al., 2012) used bandlet transforms in combination with SWT. Compared with the original SWT, instead of using Canny edge detection, they used bandlet edge transform. The output text edge map was used in SWT and geometric rule-based CCA to generate the output text instances.

(Koo & Kim, 2013) also utilized MSER as a candidate text CCs detector. Then, geometric, spatial location, and color features were extracted from each CC and fed into the first stage text/non-text AdaBoost classifier. Since MSER generates multiple overlapped regions at the same characters, the CC clustering was applied to group these regions into candidate text areas. For each CC group, skew normalization and geometric feature extraction were employed to generate the normalize corresponding regions for the reliable text/non-text classification. In the last stage, a sliding window based Multilayer Perceptron (MLP) was used in combination with gradient features to classify each region into text or non-text bounding boxes.

(Gomez & Karatzas, 2013) proposed a MSER based method for multi-script text detection. MSER components were extracted from the input image. Then, a set of possible grouping hypotheses was created from predefined features and criteria. Each group was merged from the grouping hypotheses by extracting only the most meaning features and compared with the designed metrics. Finally, the output CCs were formed into text lines by using defined criteria.

(Le Kang et al., 2013) proposed a higher-order correlation clustering for multi-oriented text line detection in natural images. A text line detection was treated as a graph partitioning problem. A graph of color MSER components was constructed from the input image. Then, MSER components were coarsely grouped base on their consistency with neighbor components to create weak hypothesis by using the following prior knowledge of text: text line must be elongated, and the projection profile of a text line should have higher variance. Finally, correlation clustering was employed on given a MSER graph, to assign a binary label to each edge, indicating whether the two vertices were connected.

(Gómez & Karatzas, 2014) proposed a real-time MSER based text detection and tracking in video sequences. This work was an extension of their previous work (Gomez & Karatzas, 2013). Instead of using only the hierarchical grouping results, the Real AdaBoost was applied for non-text regions pruning to reduce the misclassification rate. The text regions were tracked through each frame by MSER features, MSER-tracking, and RANSAC-based algorithms.

(Iqbal et al., 2014) used MSER and Bayesian networks to detect text from natural scene images. The proposed system can be divided into four stages. First, MSER component extraction and rule-based CC filtering were performed on the grayscale image to extract the candidate letter regions. Second, text regions were constructed by pruning repeated components and locating text directly in images using geometric features such as adjacent character size and color similarity. Furthermore, the candidate text binary mask was used cooperating with the learned Bayesian network to extract each character region again. Finally, each character was grouped into text bounding boxes based on height similarity and alignment score constraints.

In 2015, an object proposal idea spread around many famous object detection community. An aim of object proposal is to generate high-quality object location by using low computational complexity features. The main interest is its ability to speed up the detection pipelines that use complex and expensive classifiers by considering only a few thousands of bounding boxes instead of searching on entire images. (Gómez & Karatzas, 2015) proposed a MSER based text proposal method. This method started from MSER component extraction from multiple input image channels and merged iteratively using the single linkage criterion. The clustering/proposal ranking was calculated based on the selected three ranking methods: pseudo-random, cluster meaningfulness, and text classifier confidence.

(Su & Xu, 2015) utilized Stroke Width Transform (SWT) from previous work (Epshtein et al., 2010) Instead of using only SWT and CCA constraints to create the output text bounding boxes, seed stroke segmentation detection was proposed. After acquiring the edge map and stroke map from SWT, seed and non-seed stroke segments were calculated based on defined criteria. For each stroke seed, the edge-based region growing with the edge boundary and merging conditions were used to create candidate character regions. In the end, CCA and rule-based filtering were applied to each region to generate the output bounding boxes.

(Feng et al., 2015) employed grayscale MSER to generate ERs (Extremal Regions) tree as a text detector combined with minimal and maximal component size criteria. For each ERs, the corner feature and HOG descriptor were extracted and fed into AdaBoost text and non-text classifier to remove non-text components. Finally, the remaining candidate ERs were merged into text lines using color, spatial location, and scale-invariant features.

(Bušta et al., 2015) proposed FASText : Efficient Unconstrained Scene Text Detector, which focused on localization speed while preserving an acceptable detection rate. This system was built based on the famous FAST corner feature extractor. FAST feature keypoints were extracted and categorized into Stroke Ending Keypoint (SEK) and Stroke Bend Keypoint (SBK) using pixel intensity constraints. These keypoints were used to segment the candidate letter regions. At the last step, the character stroke area and geometric set were used to group the regions into text bounding boxes.

(L. Neumann & Matas, 2015) proposed end-to-end MSER based text localization and recognition. In the text localization part, MSER components were classified into three distinct classes: characters, multi-characters, and background by using geometric and stroke features. The characters and multi-character regions were merged using standard agglomerative clustering and iterative Grabcut to generate the text output.

(H. Cho et al., 2016) also utilized MSER as a weak text detector. In the first stage, they employed weak constraints MSER detector on YCbCr and inverted channel image to construct the ERs tree. The ERs components which have the highest stability criterion for each group were kept. The surviving character candidates were classified into three classes: strong text, weak text, and non-text. The strong text candidates were iteratively grouped into the minimum-area encasing rectangle bounding boxes.

(Qin & Manduchi, 2016) proposed MSER and CNN (Convolutional Neural Network) based text localization method. MSER feature was computed on seven channels: R,G,B, H,S,V, and grayscale, which produce a large amount of possibly overlapped regions. These regions were merged and discarded by using overlapping criterion and Jaccard similarity index. The remaining MSER patches were resized and fed into CNN to classify between text and non-text. In the end, the text lines were formed using predefined criteria, starting from the highest probability regions.

(Ray et al., 2016) proposed MSER and CNN based text localization method. The process started with extracting the region proposal using weak MSER constraints. Each component was fed to a pre-trained ImageNet CNN model, which was fine-tuned with synthetic text images and MSER outputs. At the fully connected layer, six features: stroke width, Histogram of Oriented Gradients (HOG), entropy, intensity, distance variation, and color divergence were computed and concatenated into features vector for text and non-text SVM classifier. The set of detected text regions was further merged to form text lines using non-parametric mean shift clustering. A bottom-up grouping was performed within each cluster so that characters belonging to the same text line were grouped.

(Chen et al., 2016) proposed a method for text localization in digital-born images. Since text strokes in born-digital images mostly have complete contour, they generated the candidate text CCs and categorized them into two groups: smooth and non-smooth regions. They proposed an assumption that the text contour pixels should have higher contrast with the adjacent pixels than other stroke interior region pixels. For each pixel in the smooth regions, the gradient magnitude was calculated from RGB channel separately, and keep the largest value as the final magnitude. Then, Otsu local binarization with window size 5x5 pixel was performed on the non-smooth regions. In order to filter out the remaining non-text regions, the surroundedness feature was calculated and thresholded at the defined value to discard non-text regions. Finally, CCA was performed to generate the final text regions.

From the presented connected component analysis-based scene text localization method, the overall pipeline can be concluded into the diagram shown in Figure 4.

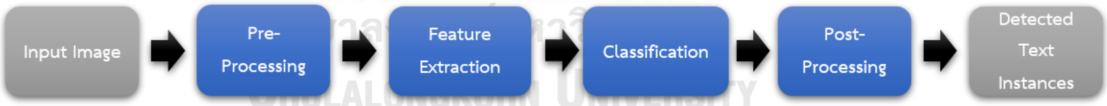


Figure 4 Overall pipeline for connected component analysis-based scene text localization.

Pre-processing stage – In this stage, the input image is pre-processed to discard foreseeable clutters or improve the text instances visibility by using well-known filters such as gaussian or median filter. The potential contours are gathered by using the proposed methods.

Feature Extraction stage – The handcrafted features, for example, geometric features, colors, and stroke width, are gathered from each extracted contour.

Classification stage – The contours are classified into text and non-text contours by using the extracted features. The classifier can be based on either predefined rules or machine learning techniques

Post-processing stage – The potential text contours are grouped into text regions/instances based on defined human observable criteria.

From the above connected component analysis-based scene text localization pipeline, the presented methods can be concluded into Table 1.

Table 1 Connected component analysis-based scene text localization.

Proposed By	Pre-Processing	Feature Extraction	Classifier	Post-Processing (Text Grouping)
(Subramanian et al., 2007)	Gaussian Filter	Simple geometric features	Predefined rules	Seed-Growing
(Zongyi & Sarkar, 2008)	-	Binary image	Predefined rules	CCA
(Minhua & Chunheng, 2008)	Gaussian Filter	Background analysis, edge feature	Predefined rules	CCA
(Shivakumara et al., 2009)	-	Gradient different feature	Predefined rules	Projection Profile
(Epshtein et al., 2010)	-	Stroke width feature	Predefined rules	CCA
(Karaoglu et al., 2010)	Gaussian Filter	Geometric, shape regularity and corner-based features	Random Forest	CCA
(X. Huang & Ma, 2010)	-	2D Log Gabor and stroke width features	Predefined rules	CCA
(Lee et al., 2010)	-	Color and spatial feature	K-mean clustering	Markov Random Field
(Lukas Neumann & Matas, 2010)	-	MSER and geometric features	Support Vector Machine	Least Median Square
(Lukas Neumann &	-	MSER geometric and spatial feature	Real AdaBoost and SVM	Least Median Square

Proposed By	Pre-Processing	Feature Extraction	Classifier	Post-Processing (Text Grouping)
Matas, 2011)				
(Yin et al., 2012)	-	MSER geometric and spatial feature	AdaBoost	CCA
(Mosleh et al., 2012)	-	Stroke width feature	Predefined rules	CCA
(Koo & Kim, 2013)	-	MSER and geometric features	Neural Network	Clustering
(Gomez & Karatzas, 2013)	-	MSER	Predefined rules	Predefined rules
(Gómez & Karatzas, 2014)	-	MSER	Real AdaBoost	CCA and Real AdaBoost
(L. Kang et al., 2014)	-	MSER	Predefined rules	Graph based
(Iqbal et al., 2014)	-	MSER and geometric features	Bayesian Network	Predefined rules
(Su & Xu, 2015)	Gaussian Filter	Stroke width and edge features	Predefined rules	CCA
(Feng et al., 2015)		MSER, HOG and geometric features	AdaBoost	CCA
(Buta et al., 2015)	-	FAST corner feature and geometric features	Predefined rules	CCA
(L. Neumann & Matas, 2015)	-	MSER geometric and spatial feature	Predefined rules	Agglomerative Clustering
(H. Cho et al., 2016)	-	MSER	CNN	Predefined rules

Proposed By	Pre-Processing	Feature Extraction	Classifier	Post-Processing (Text Grouping)
(Ray et al., 2016)	-	MSER, HOG, geometric and spatial features	CNN (as feature extractor) and SVM	Mean Shift Clustering
(Chen et al., 2016)	-	Gradient based and surroundedness feature	Predefined rules	CCA

Unfortunately, the presented methods use handcrafted features such as contours, edges, strokes, and geometric features or the novel feature extractor, for instance, MSER, HOG, and FAST, along with human assumptions about text instances. These handcrafted features may not be robust enough in complex scenarios caused by the unexpected scene conditions, for example, orientation, occlusion, reflection, and noise. Moreover, the exsiting text grouping algorithms are still based on axis-aligned bounding boxes assumption, which is not capable of covering the entire text instances in some complex situations, e.g., curved and tilt texts.

2.1.1.2. Object or Pixel Classification Based Approaches

Object or pixel-based scene text localization approaches treat text regions as objects in object detection problem. At present, the methods in this group can be divided into two sub-categories: object-based in bounding box and pixel-based.

2.1.1.2.1. Object Detection Based Approaches

Object detection has been one of the most important applications in computer vision field which can dates back over decade. This is the task of simultaneous localization and classification of the objects presenting in an image. Many state-of-the-art object detection algorithms are modified from the previous works on text localization to be more suitable for text instance characteristics. The existing object detection-based scene text localization algorithms can be divided into two subcategories: sliding window and proposal/regression-based approaches.

2.1.1.2.1.1. Sliding Window Based Approaches

Sliding window-based scene text localization methods treat text regions as object in object detection and localization. Typically, each patch/window will be fed to a selected classifier, such as Neural Network (NN), SVM, or CNN to be classified as text and non-text. The post-processing techniques are used to combine the results from text classifier into text region bounding boxes. The existing sliding window-based scene text localization methods are described below.

(Xiangrong & Yuille, 2004) created the text detector by using edge and statistical-based features. In this work, sliding window was applied on the input images at a range of 14 scales. The window size ranges from 20 by 10 to 212 by 106, with a scaling factor of 1.2. For each window, the selected features were extracted and fed to AdaBoost text and non-text classifier. However, the text grouping method was not mentioned in this paper.

(Gllavata et al., 2004) utilized a high-frequency wavelet transform to detect text from scene images. The three channels of high-frequency wavelet: HH, HL, and LH were extracted from input images. Standard derivation of histogram feature was extracted from the sliding window size of 32x8 pixel and k-mean clustering was also applied to compute the foreground (treat as text) and background pixel. For each region, rule-based CCA was employed to generate the result.

(Hanif et al., 2008) also proposed an AdaBoost based text detector. Instead of using statistical features like (Xiangrong & Yuille, 2004), mean and standard deviation difference and HOG features from sliding window size of 32x32 pixel, were used as text features. Each text region was merged based on edge density. Finally, the output text lines were formed by CCA base on the predefined rules.

(Y.-F. Pan et al., 2008) also utilized AdaBoost as a text and non-text classifier. In this work, multi-scale local binary pattern (msLBP) and HOG were used as features for text regions classification. The multi-scale overlapped regions were merged and MRF-based CCA was applied to create the text bounding boxes.

(Hanif & Prevost, 2009) work was an updated version of (Hanif et al., 2008). The complexity AdaBoost (CAdaBoost) was used instead of a typical version. The sliding window was applied on input images to extract a set of features like (Hanif et al., 2008), which was then fed into CAdaBoost. Each detected text window was verified by using geometric features and MLP. Finally, all verified connected components for a text word were clustered to construct a single text rectangle.

(Y. F. Pan et al., 2011; Y.-F. Pan et al., 2009) presented a WaldBoost and HOG based text detector. The multi-scale text confidence maps were constructed from the sliding window based WaldBoost text detector. Then, CCs were extracted from each region and treated as graph vertices labeled as text and non-text. The minimum spanning tree was used to infer the output text lines from each text CCs.

(Y.-F. Pan et al., 2010) proposed two stages coarse and fine text detector. In the coarse stage text detector, simple edge and gradient features were used in combination with WaldBoost text classifier. Then, the coarse text lines were built by using vertical and horizontal projection profile analysis. In the fine text detection stage, each candidate text line was verified using HOG, LBP, and Discrete Cosine Transform (DCT) features, which were formed into features vector for the polynomial classifier. Finally, the remaining text lines were analyzed by rule-based CCA to discard non-text components.

(Bouman et al., 2011) presented a block-based text localization in sign images. The grayscale input image was divided into K x K non-overlapping windows. For each block, the homogenous features, calculated from luminance value, were used to determine block homogeneity. Then, a seed growing algorithm was applied to homogeneity blocks to extract the sign regions. Finally, CCA was used to separate between background and foreground.

(Coates et al., 2011) presented an unsupervised text detection and character recognition in scene images. In this work, 8x8 pixel text and non-text patches were collected from multiple sources and used as training data. For the training stage, the statistical preprocessing was applied to transform these patches into a new dataset. The unsupervised algorithm was employed to create a mapping from input patches to text and non-text features vectors, yielding a set of patches dictionary for each class, which were used for linear SVM training. At the test stage, a 32x32 pixel sliding window was applied and divided into 8x8 sub patch features vectors and fed them into a trained text/non-text classifier.

(Meng & Song, 2012) proposed a salient region-based text localization method. This work proposed assumptions that the texts are distinguishable from other regions in terms of saliency. A set of text saliency features: color distribution, contrast, center-surround histogram, and stroke width similarity, were proposed. These features were used in combination with Conditional Random Field (CRF) to generate a saliency map. Then, Niblack's binarization was applied to segment CCs from each salient region. Six geometric features were extracted and fed into text/non-text SVM classifier to filter out non-text CCs. However, this work did not include the text grouping method.

(A Mishra et al., 2012) presented the sliding window-based text localization and recognition algorithm. The 64-way character SVM model was applied to recognize each window individually at multiple scales into 64 English character and digit classes. The word graph was then built based on CRF energy minimization and lexicon prior to producing the recognition results.

In 2012, the Convolutional Neural Network based scene text localization and recognition was presented by (T. Wang et al., 2012). Instead of using handcrafted features created by humans, the well-trained CNN was used as a feature extractor. The well-designed text detector based on CNN was connected to the classification layer, which classifies the 32x32 pixel input patches into text or non-text class. Then, text confidence maps at multiple scales were constructed from the sliding window classifier. The resulted text bounding boxes were generated from merged text confidence maps and Non-Maximum Suppression (NMS) algorithm.

(Jaderberg et al., 2014) proposed a CNN based text localization and recognition. The CNN text classifier was employed on 16 different input image scales to generate text saliency maps. These maps were used to generate text bounding at different scales, which were merged by run-length smoothing algorithm and linkage CCA. Finally, the text bounding boxes were divided into word level bounding boxes by Otsu thresholding and predefined CCA rules.

A MSER and CNN based text localization was presented by (W. Huang et al., 2014). This work utilized MSER as a first stage detector. These regions were resized into 32x32 pixel patches and fed into text and non-text CNN-SVM classifier to generate text confidence map. The components analysis was employed on both CNN and overlapped MSER components to split each character individually and merged them into text bounding boxes.

(Jaderberg et al., 2016) proposed an algorithm for reading text in the wild using Convolutional Neural Network. Since 2015, the object proposal algorithms have gained much attention from many researchers in object recognition field. Instead of searching the entire image using an exhaustive search strategy, the object proposal would propose high-quality object locations based on defined criteria. This work proposed a text proposal algorithm by using EdgeBoxes (Zitnick & Dollár, 2014) in combination with the aggregate channel features. Each proposed region was filtered out and refined by using CNN text detector.

From the presented sliding window-based scene text localization method, the overall pipeline can be drawn into the diagram shown in Figure 5.

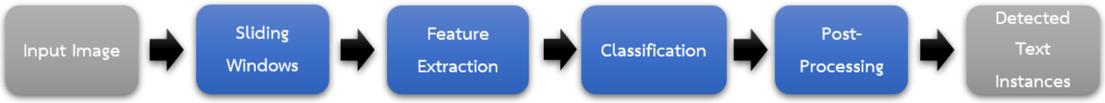


Figure 5 Overall pipeline for sliding window-based scene text localization.

Sliding window – The patches or windows are extracted from the input image. The extracted patch can be single or multiple scaled based on the proposed algorithm.

Feature Extraction stage – The proposed features, for example, geometric features, colors, stroke width, and notable features extractors, HOG, LBP, are extracted from each patch.

Classification stage – The patches are classified into two classes: text and non-text contours by using the extracted handcrafted or learnable features. The classifier can be either predefined rules or machine learning techniques, however, most sliding window-based methods usually use famous machine learning algorithms to distinguish between text and non-text regions.

Post-processing stage – The potential text patches are grouped into text regions/instances based on defined human observable criteria.

From the above sliding window-based scene text localization pipeline, the presented methods can be concluded in Table 2.

Table 2 Sliding window-based scene text localization methods

Proposed By	Feature Extractor	Classifier	Post-Processing (Text Grouping Method)
(Xiangrong & Yuille, 2004)	Geometric and gradient features	AdaBoost	-
(Gllavata et al., 2004)	Mean and SD from High frequency wavelet coefficient	K-Mean Clustering	CCA
(Hanif et al., 2008)	Statistical features, HOG	AdaBoost	CCA
(Y.-F. Pan et al., 2008)	msLBP, HOG	AdaBoost	MRF and CCA
(Hanif & Prevost, 2009)	Statistical features, HOG	cAdaBoost and MLP	Clustering
(Y.-F. Pan et al., 2009)	HOG	WaldBoost	Graph Labeling
(Y.-F. Pan et al., 2010)	HOG, LBP, and DCT	WaldBoost and Polynomial Classifier	CCA
(Y. F. Pan et al., 2011)	HOG	WaldBoost	Graph Labeling (CRF) and CCA
(Bouman et al., 2011)	Homogenous features from luminance value	Predefined rules	CCA
(Coates et al., 2011)	Unsupervised features learning	SVM	-
(Meng & Song, 2012)	Saliency features	SVM	-

Proposed By	Feature Extractor	Classifier	Post-Processing (Text Grouping Method)
(A Mishra et al., 2012)	HOG	SVM	CRF
(Jaderberg et al., 2014)	Learned features from CNN	CNN and SVM	NMS
(W. Huang et al., 2014)	MSER and Learned features from CNN	CNN and SVM	CCA
(Jaderberg et al., 2016)	Learned features from CNN and aggregate channel features	CNN	CCA

Sliding window-based approaches are tremendous computational complexity because they need to extract and classify every possible even those overlapped regions, which may not be practical in real-world applications. Moreover, it has already demonstrated in many recent text detectors that global and wide receptive field context are beneficial features to increase the overall accuracy while the sliding window-based methods only focus on local features.

2.1.1.2.1.2. Proposal and Regression Based Approaches

The methods in this group inspired from the famous proposal and regression based object detection work, for example, Faster-RCNN (Ren et al., 2017), Single Shot MultiBox Detector (SSD) (W. Liu, Anguelov, et al., 2016), YOLO (Redmon & Farhadi, 2018), and Mask-RCNN (K. He et al., 2017). Various text representations such as axis-aligned bounding boxes, quadrangles, and pixel mask are used methods in this group. The existing proposal and regression-based scene text localization methods are described below.

(Tian et al., 2016) presented a method called Connectionist Text Proposal Network (CTPN). Reading text is a fine-grained recognition task that requires an accurate detection that covers a full region of a text line or word. Therefore, a vertical anchor mechanism jointly with LSTM text proposal connection was proposed in this work. The method began by feeding the entire input image into VGG-16 network. Then, at each location on conv5 feature map, a 3x3 sliding window was applied to extracted visual features, which were later concatenated and fed into the connectionist layer based on Long-Short Term Memory (LSTM) and fully connected layer. Finally, a set of possible text anchors was calculated from the regression output from the fully connected layer. However, this work only focused on regular shape text instances, represented by a typical axis-aligned bounding boxes, which may not be sufficient to express complex text instances.

(X. Zhou et al., 2017) proposed a method call an Efficient and Accurate Scene Text Detector (EAST). Instead of using the standard object detection anchor paradigm, this work directly regressed the text instance locations from convolution feature map. The input image was directly fed into VGG-16 network, which was extended with specially design feature merging branch. At the last layer of feature merging branch, the convolution layer with N output channels was connected, where N equals to the number of points used by text representation, for example, $N = 8$ for quadrangle or $N = 5$ for rotated bounding box. As the object detection algorithm produces the overlapped bounding boxes or regions, non-maxima suppression (NMS) was used to suppress those detections. However, standard NMS usually requires $O(n^2)$, which is not practical in this dense proposal prediction problem. Hence, locality-aware non-maxima suppression, which uses only $O(n)$ complexity, was proposed under the assumption that the geometries from nearby tend to be highly correlated and merged without comparing with the entire detections.

(Liao et al., 2017) proposed a single-stage regression-based scene text detector called TextBoxes. This work was mainly based on Single Shot MultiBox Detector (SSD) with a modified anchor mechanism. As text instances in natural scene images usually appear in horizontal and dense alignment which are different from ordinary objects in terms of aspect ratio and density, making the existing anchor may not capture long and complex text instances. In this work, long and dense text anchors were proposed to capture those cases. To produce the final predictions, standard NMS was used to discard overlapped and kept the best detections from the overall outputs. Nevertheless, the proposed anchor mechanism still based on axis-aligned bounding boxes, which may not be capable of covering the complex text instances.

(P. He et al., 2017) presented a single-stage regression-based scene text detector. Instead of using traditional SSD implementation, this work incorporated the text attention mechanism and a hierarchical inception module, which aggregates multi-scaled inception features. The text attention module was built on the Aggregated Inception Feature (AIF), which generated the text probability heatmaps at each pixel location. These heatmaps were fed into the hierarchical inception module, which were designed to capture text instances in difference scales. Finally, all features from hierarchical inception modules were aggregated to produce possible text anchors, which all overlapped regions were later discarded by standard NMS. Nonetheless, the axis-aligned bounding box was still used as text representation.

ArbiText: oriented text detection method was presented by (Xing et al., 2017). This method was based on SSD implementation but with different anchor representation. The pyramid pooling module was used in combination with the standard VGG-16 backbone to generate feature maps. These feature maps were directly connected to the text proposal layer, which was modified to use a novel circle anchor. The proposed circle anchor expresses each text proposal using five parameters: area, radius, and rotated angle of a circle anchor. Finally, the overlapped text proposals were merged using locality- aware NMS proposed in (X. Zhou et al., 2017)

(Jiang et al., 2018) proposed a proposed based scene text detector called R^2CNN. This work is mainly based on Faster-RCNN with rotational anchors. Rather than using a standard axis-aligned bounding box (XYWH), this work used rotational bounding box representation (XYHA) to express text instances. The proposal convolution layer output was changed from 4 to 5 channels, representing the center of rotational box (X,Y), box height, angle, and longer side size. The inclined NMS was used to merged overlapped detection.

(Minghui Liao & Bai, 2018) proposed an upgraded version of TextBoxes called TextBoxes++. This version was designed to support oriented text detection. Based on their previous work, a quadrilateral representation (X_1Y_1, X_2Y_2, X_3Y_3, X_4Y_4) was used in combination with long and dense text anchors. The same style VGG-16 with multi-scaled feature was used to produce multi-scaled text proposals, improving overall detection performance. To produce the final predictions, standard NMS was used to discard overlapped and kept the best detections.

(Liao et al., 2018) proposed a single-stage regression-based scene text detector called RRD. This work was mainly based on Single Shot MultiBox Detector (SSD). The main contribution of this work is a different CNN convolution filter called active rotating filters (ARF), which convolves a feature map with a canonical filter and its rotated clones. A wide receptive field inception convolution block was connected to ARF layer to handle the long text lines, yielding the regression and classification branches. The same anchor styles and post-processing NMS proposed in (Minghui Liao & Bai, 2018) were also used in this work to produce final predictions.

(Ma et al., 2018) proposed a proposal-based scene text detector. This work was mainly based on Faster-RCNN with a new rotation region of interested pooling (RROI). The work began with the typical Faster-RCNN pipeline, at the region proposal stage, instead of using the standard XYWH bounding box, the proposal representation was changed to rotational bounding box (XYWA). Then, a novel rotation region of interested pooling (RROI), which is capable of extracting feature in non-axis-aligned manners, was used to pool the rotation sensitive features. These rotation sensitive features were used to classify the corresponded proposal into text and non-text classes. At the final stage, the overlapped "text" proposals were discarded by skew-based non-maxima suppression, which not only used intersection-over-union criterion but also the proposal skew angle, were used in place of traditional NMS.

(Zhu et al., 2018) proposed a fusion feature extractor for scene text detection problem. The main contribution of this work was the feature from multi-scaled were combined using the same idea, which was presented in feature pyramid network (FPN) (Lin, Dollár, et al., 2017). This work was mainly based on the Faster-RCNN framework with a quadrilateral text representation.

In (S. Zhang et al., 2018) work, the feature enhancement network and region proposal combined with hyper feature generation were presented. This method were mainly based on the Faster-RCNN framework. Multi-scaled feature maps were pool from ResNet-101 backbone and divided into two branches: region proposal and classification. For the region proposal branch, typically, the data imbalance between text and non-text anchors extensively affects the detection results. Hence, the positive matching strategy, which is a method to pick only potential anchors, was proposed. The ordinary axis-aligned bounding box was used to represent text instances in this stage. For the classification branch, an adaptive weighted position-sensitive region of interest pooling, which capable of extracted the oriented feature, was used to extract the text or non-text feature at each anchor corresponding location on classification feature map. Finally, standard NMS was employed, obtaining the final detection results. This work was focused only on an ordinary text which may not be able to capture text in some complex cases.

SPCNET was proposed by (E. Xie et al., 2019). This work was mainly based on Mask-RCNN object detection framework, which produces both bounding box and pixel mask for each detected object. ResNet-50 combined with feature pyramid network (FPN) was used as a main feature extractor backbone. At each stage of lateral connection, the output feature map was connected to the text context module consisting of two sub-branches called Pyramid Attention Module (PAM) and Pyramid Fusion Module (PFM). The feature maps from both branches were fed into Mask-RCNN branch, which produces text instance classification, text axis-aligned bounding box, and pixel mask.

(J. Liu et al., 2019) presented a Mask-RCNN based text detector. The main contribution of this work was the plane clustering algorithm and soft pyramid text mask. Rather than directly learn a text pixel masks in the mask detection branch of Mask-RCNN, a soft pyramid labeling was used to represent text mask. The soft pyramid text mask was calculated from a linear combination of two lines across the center point of considered text mask. This soft mask was then clustered by the proposed plane clustering algorithm to reconstruct the output text mask for each text instance.

(C. Zhang et al., 2019) proposed a text detector call Look More Than Once (LOMO). This work contained three major modules: direct regression, iterative refinement, and shape expansion modules. The feature map from ResNet50-FPN backbone was fed into the direct regression to predict the text instances coordinate directly. However, the direct regressor output coordinate may not accurate enough to capture text instance. Thus, the iterative refinement was used to refine that coordinate to be more precise in the quadrangle format.

From the presented proposal and regression-based scene text localization methods, the overall pipeline can be concluded into the diagram shown in Figure 6.

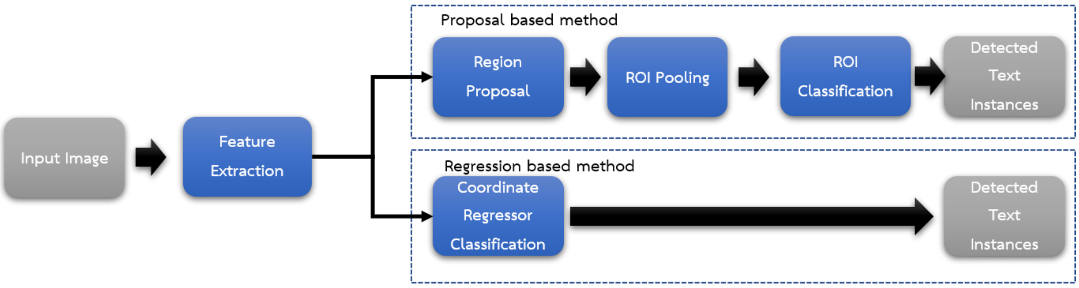


Figure 6 Proposal and regression-based scene text localization diagram.

Feature Extraction stage – The entire input image is fed into selected CNN feature extractor structure, for example, VGG16, ResNet50, or ResNet101. Various layers, configurations, and connections can be added to improve the feature quality.

Proposal-based method stage – The feature maps from the feature extraction stage are fed into the region proposal layer, which produces the potential "text" regions in selected formats such as axis-aligned, rotated bounding boxes, and quadrangles. The features are then extracted from those corresponding regions on feature maps to produce accurate text or non-text prediction.

Regression-based method stage – Instead of using two stages like the proposal-based method, regression-based method just directly learns the potential text region coordinate and its class simultaneously without additional layers. This makes the methods in this group usually faster but achieve lower accuracy compared to two stage methods.

From the above proposal and regression scene text localization pipeline, the presented methods can be concluded into Table 3.

Unfortunately, the methods mentioned above were not designed or trained for multi-language scene text, particularly Thai language, which contains many specific characteristics that do not appear in other languages. Moreover, the methods that are capable of detecting irregular shape text instances require a tremendous amount of computational power, which may not be practical in real-world applications.

Table 3 Proposal and regression-based scene text localization methods.

Proposed By	Based on	Feature Extractor	Post-Processing	Output
(Tian et al., 2016)	SSD	VGG16	NMS	Axis-aligned bounding box
(X. Zhou et al., 2017)	Direct Regression	VGG16	Locality-Aware NMS	Quadrangles
(Liao et al., 2017)	SSD	VGG16	NMS	Axis-aligned bounding box
(Xing et al., 2017)	SSD	VGG16	Locality-Aware NMS	Quadrangles
(Jiang et al., 2018)	Faster-RCNN	VGG16	Inclined NMS	Rotated bounding box
(Minghui Liao & Bai, 2018)	SSD	VGG16	NMS	Quadrangles
(Liao et al., 2018)	SSD	VGG16	Skew NMS	Quadrangles
(Zhu et al., 2018)	Faster-RCNN	ResNet101 with Feature Fusion	Locality-Aware NMS	Quadrangles
(S. Zhang et al., 2018)	Faster-RCNN	ResNet101 with Feature Enchantment Module	NMS	Axis-Aligned bounding box
(E. Xie et al., 2019)	Mask-RCNN	ResNet-50	NMS	Axis-Aligned bounding box and Text mask
(J. Liu et al., 2019)	Mask-RCNN	ResNet-50FPN	NMS	Axis-Aligned bounding box and Text mask

Proposed By	Based on	Feature Extractor	Post-Processing	Output
(C. Zhang et al., 2019)	Direct Regression	ResNet-50FPN	NMS + Shape Expansion	Rotated bounding box and Text mask

2.1.1.2.2. Pixel Based Approaches

The methods in this group were inspired by the long-standing topic in computer vision field called semantic segmentation. Rather than directly detecting text in bounding box levels, the methods in this group label text and non-text regions in a pixel level joined with other pixel-level features to help separating nearby text instances. The existing pixel-based scene text localization methods are described below.

(Z. Zhang et al., 2016) presented a multi-oriented text localization method utilizing the Fully Convolutional Neural Network (FCN), which was widely used in semantic segmentation work. This work tried to create a mapping between the input image and text regions mask by FCN. The VGG-16 pre-trained weight was used as a feature extractor. The outputs from each convolution layer were accumulated and deconvoluted to generated text saliency maps at the input image original scale. To further extract accurate text bounding boxes, the characters were extracted within the detected text block by MSER. Finally, the skew correction and text line estimation based on predefined criteria, were used to create the output minimal text bounding boxes.

(Deng et al., 2018) presented a scene text detection method based on FCN called PixelLink. This work used a standard FCN pipeline in combination with feature upsampling to generate text map representation. Rather than directly uses the text pixel mask to represent text instances, this work was inspired by a connected component analysis algorithm. The eight way connection maps for each pixel were used with ordinary text masks to express each text instance boundary. Since the ratio between text and non-text pixels is highly imbalance, the weighted cross-entropy loss was used to correct this problem.

TextSnake was presented by (S. Long et al., 2018) in 2018. The flexible text representation, which is capable of arbitrary express irregular text instances, was proposed. This work used a standard FCN pipeline in combination with VGG16 network to generate text map representation. Each text instance was represented by text pixel mask, text center line pixel, radius from center line to text border, and angle from center line pixel to text border. These text instance features were learned in the form of pixel-based maps.

(Y. Baek et al., 2019) proposed a work called Character-Region Awareness for Text detection (CRAFT). The main contribution of this work is to generate a precise pixel mask at the character level. Nevertheless, to create training data with character level annotation is very time consuming and costly, the weakly-supervised learning method was proposed. Starting from the synthetic scene text dataset, which all characters were annotated at bounding boxes level, at the first stage, word-level pixel mask (region score) was learned until the model converges. The character pixel level mask was then further learned by this model using the ground truth generated from watershed algorithm. At the inference stage, a simple connected component analysis was directly used on the character pixel map to create final detection.

From the presented pixel-based scene text localization methods, the overall pipeline can be drawn into the diagram shown in Figure 7.

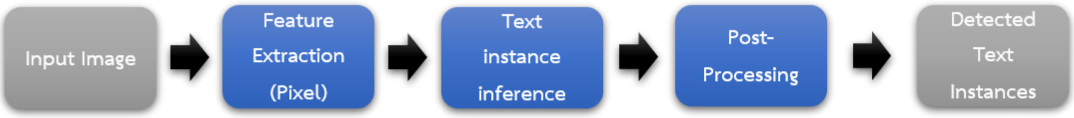


Figure 7 Pixel based scene text localization method pipeline.

Feature extraction stage - extracts text instance representation feature from the input image at a pixel level.

Text instance inference stage – combines the pixel level text instance feature to form the potential text instances.

Post-processing – discards the false-positive text instance based on defined criteria or learnable rules.

From the above pixel-based scene text localization pipeline, the presented methods can be concluded into Table 4.

Table 4 Presented pixel-based scene text localization methods.

Proposed By	Feature Extractor	Text Representation	Post-Processing
(Z. Zhang et al., 2016)	VGG-16	Text Pixel and MSER	Predefined rules
PixelLink (Deng et al., 2018)	VGG-16	Text pixel and direction maps	Pixel aggregation
TextSnake (S. Long et al., 2018)	VGG-16	Text pixel, text center line, radius from center line to text border, and angle from center line pixel to text border masks	Scanline pixel algorithm
CRAFT (Y. Baek et al., 2019)	VGG-16	Precise text mask at character level	Connected component analysis

The methods in this group have proved their performance on standard scene text localization benchmark datasets. The main contribution of the methods in this group is the capability of detecting text in arbitrary shapes, not only limit to the axis-aligned bounding box representation. This characteristic is very useful in real-world scene text localization applications and further increase the scene text recognition algorithm due to precise text instance capturing.

2.1.2. English Scene Text Recognition

Scene text recognition aims at taking cropped word images and transcribing them into text. Text recognition in natural scene images differs from the one in scanned document images and handwritten recognition since generic scene images consist of highly variable foreground and background textures that do not appear in digital-born documents.

2.1.2.1. Character Based Scene Text Recognition Approaches

The methods in this group transform the text instance image into text transcription at a character level. Character based methods can be divided into two sub-groups: segmentation and segmentation-less methods.

2.1.2.1.1. Segmentation Based Approaches

Segmentation based approaches perform segmentation algorithm to extract character individually from cropped word or sentence images. Each character is classified individually by feeding the cropped character into the proposed classification methods.

(Campos et al., 2009) presented a cropped English character dataset called Char64k dataset. In this work, a series of experiments on different feature extractors on character recognition problem was conducted. These features were extracted and classified by a nearest-neighbor classifier, support vector machine (SVM), and multiple kernel learning (MKL).

(Kai et al., 2011) proposed an end-to-end system for scene text recognition. The method began with character class detection by applying multi-scale sliding window search in combination with HOG features extractor and 64-way random forest classifier to acquire each location character class probabilities. To combine detected character into words, they employed pictorial structure scoring to find the optimal configuration based on the provided lexicon.

Nevertheless, segmenting individual characters from cropped scene text images is very difficult due to complex background and various lighting conditions. Moreover, many modern scene text recognition methods have shown that the global feature and context from the entire cropped text image are essential features that can dramatically improve recognition performance.

2.1.2.1.2. Segmentation-less Based Approaches

Rather than segmenting the cropped word images into characters, the methods in this group classify the entire cropped images into text transcriptions. The existing segmentation-less scene text recognition methods are described below.

(T. Wang et al., 2012) utilized CNN as the end-to-end system for text recognition. For the cropped word recognition part, 62-way CNN and SVM are employed on word image in sliding window fashion to acquire the 62 classes probability matrix for each window. The standard Non-Maximum Suppression (NMS) was applied to locate the locations where a character is most likely to be presented.

(Alsharif & Pineau, 2013) proposed a CNN-HMM network for cropped word recognition. The proposed CNN model for character text recognition task consists of three convolution layers followed by maxout and softmax layers. For additional information, maxout layer is a layer that applied max function to several inputs and set others to zero. They then applied this CNN in sliding window fashion on input cropped word image to acquire a character sequence. In this work, a recognized character sequence was treated as a sequencing problem. The final transcribe result was built by feeding a character sequence into HMM cooperating with dictionary-based word correction.

(Bissacco et al., 2013) from Google proposed PhotoOCR, a method for uncontrolled text recognition. In this work, the main character recognizer was built based on five layers of deep neural network. MRF-based text detection and segmentation was applied to acquire segmented characters. These characters were normalized, extracted HOG feature, and fed into five layers deep neural network. The proposed network output consists of 100 classes, including a noise class. In the end, the recognized character output was corrected based on a standard n-grams algorithm.

(Jaderberg et al., 2014) also proposed a sliding window based CNN for end-to-end word recognition. The proposed model shared the same weight for the first and second convolution layers. The output features maps from second convolution layer were connected to other networks: text/non-text classifier, case non-sensitive character classifier. Instead of applying a single character output for each sliding window, they employed 604-way softmax output to produce a bigram class probability matrix, yielding an optimal breakpoint between each character. The recognized output was then corrected based on the provided lexicon.

(Jaderberg et al., 2016) utilized their old work (Jaderberg et al., 2014) and proposed a joint model between individual and bigram character classification. The output sequence from both individual character and bigram classifier was fed into a beam search and the path-selected layer. While a beam search layer produced the best recognition result for each sliding window location, the path-selected layer also calculated the best combination bigram class traversal path based on the current window location. The results from each location on both layers was used to produce recognition output

(Pan He et al., 2016) applied maxout CNN and Recurrent Neural Network (RNN) to scene text recognition problem. Like other CNN scene text recognition approaches, they employed 32x32 sliding window on input cropped word and fed into maxout CNN proposed by (Alsharif & Pineau, 2013), to produce a classification result for each pixel column, yielding a recognized character sequence. Since RNN has shown a strong capability for learning meaningful information from an ordered sequence, in this work, to generate cropped word text transcription, the recognized sequence was treated as a sequence labeling problem and used in RNN to find optimal word labeling output.

(X. Liu et al., 2016) proposed a CNN based scene text recognition approach. They utilized a 62-way case-sensitive CNN model as sliding window-based character recognizer on the cropped word image in this work. Then, the recognition result for each window was evaluated, obtaining the output character class matrix. Since the result class matrix from CNN often includes classification error, Weight Finite State Transducer (WFST) was applied by treating the recognition output as a character to word state transition problem based on the defined lexicon.

(Ray et al., 2016) proposed a scene text recognition method which focused on curved word images. Since scene text may appear irregular shape, for example, perspective text, which is caused by side-view camera angles while some have curved shapes, meaning that its characters are placed along curves. In this work, Spatial Transform Network (STN) and Sequential Recognition Network (SRN) were used for curved text recognition. In the STN, a cropped word image was transformed into a rectified image (regular aligned word), which was more appropriate input for text recognition. Each window from the rectified image was then fed into SRN network, which is a combination between typical CNN and RNN to produce sequence character classification into lexicon word.

(W. Liu, Chen, et al., 2016) presented a scene text recognition seq2seq pipeline called Spatial Attention Residue Network (STAR). At the first stage, STN was used to normalize the cropped word into a more suitable shape for recognition. The normalized image was fed into ResNet backbone to extract visual feature and Bidirectional Long-Short Term Memory (BiLSTM) to extract the discriminative sequence feature. This sequencing feature was then decoded into text transcription using Connectionist Temporal Classification (CTC) and standard best path decoder.

(Shi et al., 2016) presented a Robust Scene Text Recognition with Automatic Rectification (RARE), which is capable of automatically transform irregular shape text into a suitable. This work pipeline was mostly like STAR (W. Liu, Chen, et al., 2016) but with some modification in STN and transcription stages. In the STN stage, not only standard STN transformation (rigid transformation) but also non-rigid transformation was used and called Thin Plate Spline Transform (TPS). For the transcription stage, Gated Recurrent Unit (GRU) was used to generate the output transcription text sequence.

(Shi, Bai, & Yao, 2017) proposed a famous pipeline for scene text recognition called Convolutional Recurrent Neural Network (CRNN). This pipeline consists of three stages: feature extraction, sequence feature extraction (BiLSTM), and transcription. The cropped word image was applied into a handcrafted structure CNN based feature extractor then passed into the sequence feature extraction layer. The sequence feature was decoded into transcription by using CTC loss and greedy decoder.

In 2018, Facebook (Borisyuk et al., 2018) proposed a large-scaled scene text recognition pipeline called Rosetta. This pipeline was designed to process very large-scaled images under specific time and memory constraints, so the proposed method must be efficiently designed to meet the system criterion. To minimize the recognition runtime, STAR structure was used but without time-consuming layers, which were normalization layer (STN) and sequence feature extraction (BiLSTM). The best path decoded was used to transcript sequence into text transcription.

A Multi-Object Rectified Attention Network for Scene Text Recognition (MORAN) was proposed by (Luo et al., 2019). This scene text recognition pipeline consists of four stages: rectification, feature extraction, sequence modeling, and decoder. Like the previous work, MORN still used STN to normalize the input image to rectify the irregular shape text into a horizontal line. Then, the handcrafted, designed CNN feature extractor was used to extract visual feature from rectified text image. The visual feature was then transformed into a sequence feature by using standard BiLSTM. At the last decoding layer, attention-based sequence recognition was used to decode sequence feature into text transcription.

From the described segmentation-less scene text recognition methods, the overall pipeline can be concluded into the diagram shown in Figure 8.

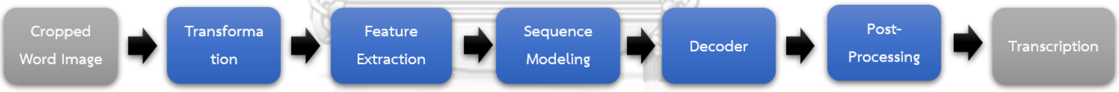


Figure 8 Segmentation-less scene text recognition method pipeline.

Transformation stage - transforms the cropped word images into more suitable shapes for recognition.

Feature extraction stage - extracts visual features from transformed images.

Sequence modeling stage - provides the contextual information from characters sequence, rather than doing it independently.

Decoder stage - determines the output sequence from extracted image features.

Post-processing – corrects the recognized output based on some specific context.

From the above segmentation-less scene text recognition methods, the presented works can be concluded into Table 5.

Table 5 Segmentation-less scene text recognition methods

Proposed By	Transformation	Feature Extractor	Sequence Modeling	Decoder	Post-Processing
(T. Wang et al., 2012)	-	Handcrafted CNN	-	SVM + Sliding Windows	NMS
(Alsharif & Pineau, 2013)	-	Handcrafted CNN	HMM	Maxout	Dictionary based correction
(Bissacco et al., 2013)	-	Handcrafted CNN	-	MRF	-
(Jaderberg et al., 2014)	-	Handcrafted CNN	-	Softmax + Sliding Windows	Dictionary based correction
(Jaderberg et al., 2016)	-	Handcrafted CNN	-	Softmax + Beam Search	Dictionary based correction
(Pan He et al., 2016)	-	Handcrafted CNN + Maxout	RNN	Best Path Decoder	-
(Ray et al., 2016)	STN	CNN	RNN	Best Path Decoder	-
STAR (W. Liu, Chen, et al., 2016)	STN	Handcrafted CNN	RNN	CTC + Best Path	-

Proposed By	Transformation	Feature Extractor	Sequence Modeling	Decoder	Post-Processing
				Decoder	
RARE (Shi et al., 2016)	TPS	Handcrafted CNN	RNN	GRU + Best Path Decoder	-
CRNN (Shi, Bai, & Yao, 2017)	-	Handcrafted CNN	BiLSTM	CTC	-
Rosetta (Borisyuk et al., 2018)	-	Handcrafted CNN	-	CTC	-
MORAN (Luo et al., 2019)	STN	Handcrafted CNN	BiLSTM	Attention (GRU)	-

The methods in this group have proved their effectiveness on famous scene text recognition benchmark datasets. However, the previously proposed pipelines are designed for English scene text recognition which is different from Thai scene text recognition in many aspects, for example, multi-level writing style and many difficult distrainable characters, make the existing method may not work well on Thai text. Thus, the modification and further studies for the method in this group are required to make an algorithm suitable for Thai scene text recognition.

2.1.2.2. Word Based Approach

(Jaderberg et al., 2016) proposed a CNN word-based recognition method. Instead of applying a sliding window on input cropped word to classify each window individually, the proposed CNN model took whole cropped word images as input. Each word was individually mapped to one class. The input image was normalized to CNN input size (32x100 pixel) without aspect ratio preserving. The proposed CNN consists of five convolution layers and three fully connected layers connecting to fixed 90k words SoftMax layer. Nevertheless, this method could only recognize the word in the predefined lexicon, making this method not suitable for Thai scene text recognition.

2.2. Hybrid Scene Text Localization and Recognition

In work (Ray et al., 2016), a scene text localization algorithm based on recognition result was proposed. Instead of using typical localization-recognition linear pipeline, the character recognition on coarse text localization results to refine the fine-scale output text bounding boxes was used. The proposed method began with grayscale MSER components extraction on the input image. Since MSER usually generates a huge number of false-positive, a set of extracted features from CNN concatenating with six handcrafted features were used to improve the result. These features were fed into SVM to classify between text and non-text. Each text patch was classified by both SVM and AdaBoost 64-way character recognizer based on Aggregate Channel Feature (ACF). The set of detected text regions was further merged to form text lines using non-parametric mean shift clustering. A bottom-up grouping was performed within each cluster so that characters belonging to the same text line were grouped together. Finally, the expanded text bounding boxes were pushed into this loop again until the output text regions were stable.

2.3. Thai Optical Character Recognition

After the text locations are retrieved, the selected recognition model recognizes these detected text regions. To the best of our knowledge, there is no existing Thai scene text recognition. In this section, the existing Thai optical character recognition will be presented, including their applications in different scenarios. After investigating the existing works, Thai Optical Character Recognition can be categorized into two groups; handcrafted and learned feature based methods.

2.3.1. Handcrafted Feature Based Approaches

The existing handcrafted feature based Thai Optical Character Recognitions which are usually multi-stages algorithms can be divided into three main stages: image preprocessing, character segmentation, and character recognition. The algorithms in this group typically use human observable characters characteristics such as pixels, shapes, or gradient direction and similarity as features. There are many interesting methods in this group which are described below.

(Vel et al., 1995) presented a pixel-based Thai character recognition method. Given a 32x32 pixel binary character patch, they employed simulated light sensitive (SLS) as a features extractor, yielding 32x1 feature vector. This feature vector was fed into three layers MLP to produce the classification result.

(Tanprasert & Koanantakool, 1996) proposed a method for OCR in Thai scanned documents. The input scanned image was aligned and preprocessed to filter out unwanted noise pixels. Later, the line and character segmentation were applied to separate each pixel blob into individual character. Finally, each segmented character was transformed into 8x8 handcrafted feature vector to classify by using 78 way MLP.

(Kijsirikul et al., 1998) proposed a novel feature extractor for Thai characters. In this work, the proposed feature vector consisted of primitive vectors, representing the direction of the line or the type of circle, and feature from multiple regions in character images. These features were used in Progal, which is an Induction Logic Programming (ILP) algorithm, to generate 77 rules for each character. Since none of the defined rules might match the input character, especially with noisy or unseen input, MLP was employed to approximate the nearest match.

(Mitatha et al., 2001) proposed Thai character recognition based on rough sets theory by using pixel as a feature. Each individual segmented 16x32 pixel character image was divided into 4x4 pixel patches. Then, the number of black pixels in each patch is generated 32x1 feature vector. Finally, the generated feature vector was classified by a rule-based voting algorithm.

(Watjanapong & Chom, 2001) also utilized rough set theory like proposed by (Mitatha et al., 2001). In this work, the two stages Thai character recognition algorithm, which were coarse and fine classifications, was proposed. Instead of using simple pixel binary features extractor, the x and y axes projection for 1st level were used in combination with a region of interest projection on the defined area for each sub-class. These features were used to classify into 76 classes of Thai characters.

(Thongkamwitoon et al., 2002) presented novel distinctive features for Thai and English handwritten character recognition. The proposed features were divided into two groups: primary and secondary features, which can be defined as common features and rare features for some characters, respectively. The primary features consisted of several loops, islands, loop height, and loop connections. These primary features were used to categorize characters into defined groups. The secondary features based on the active region, geometric loop ratio, and character ripple were used to classify each categorized character. Since this work includes both Thai and English characters, six extended distinctive features: level of character, loop to character area, width ratio, and loop generated point, were used for language classification. These features were fed into the handcrafted decision tree to produce a result for each character.

In 2003, a Thai handwritten recognition system based on Hidden Markov Model (HMM) and fuzzy logic was proposed by (Budsayaplakorn et al., 2003). In this work, the stroke features sub-stroke coordinate, stroke angles, and length were used as features for each character. These features were fed into HMM, which was used as a first stage classifier. Then, the three distinctive features consisting of character head, starting-ending points, and curl of character in combination with the posterior probabilities from HMM were used in the fuzzy logic classification to improve the final recognition result.

(Roongroj & Povey, 2003) also presented the HMM based for Thai handwritten recognition. The composite image was constructed from input character by concatenating the input image with a 90-degree rotation and polar coordinate transformation of itself. Then, the block-based Principle Component Analysis (PCA) was used to generate to reduce the output feature vector dimension. This output vector was fed into HMM character classifier to produce the classification results.

(Thammano & Duangphasuk, 2005) proposed a novel ARTMAP, a kind of supervised neural network, designed for Thai character recognition. First, the input character image was divided into five depth levels in both horizontal and vertical. Each depth level was fed into the proposed DEPTH feature extractor to determine the representative level feature vector. These multi-level DEPTH features were concatenated and used to classify each character into individual classes using the proposed ARTMAP structure.

(I Methasate & Sae-tang, 2004) proposed a clustering technique for Thai handwritten recognition. In this work, each character was represented by a set of global features. The vertical stroke feature was represented by applying the Triangle method on vertical pixel projection. From hills position information, the image was divided into 7x10 blocks based on predefined criteria and the average value from each block was used to form the input vector to be fed into neural network. In the training phase, since some characters have similar structures, these characters were grouped into the same class, yielding 23 groups.

(Thammano & Duangphasuk, 2005) proposed an updated version of ARTMAP, the previous work on handwritten recognition. The feature used in this work acquires from the direction codes, which represented the stoke direction from 9x9 blocks input image. Instead of using a typical neural network, they proposed hierarchical cross-correlation ARTMAP, which consisted of multiple neural networks voting for the best features. The winning features vector was fed into the clustering and output layer, producing the character classification result.

(I Methasate et al., 2005) proposed the multiple features for Thai character recognition. The proposed features can be divided into two groups: the global and local features. The segmented input character was divided into 7x10 nonlinear blocks and the global pixel distribution feature was extracted from each block. The local feature was represented the Thai character symbolic structure, consisting of loops, endpoint, junction point, and curl. In the classification stage, the two stages classifier was proposed. It began with the coarse classification to distinguish the input image into 20 groups by using the global feature. Second, the fine classification consists of 20 MLPs, equal to the coarse classification output, to product the recognition result.

(Sa-ngamuang et al., 2007, p.) proposed a Thai license plate recognition (LPR) system. The cropped input license plate image was segmented into individual character by using thresholding and pixel projection in both vertical and horizontal directions. For each character, the essential handcrafted feature was extracted and performed the inclusion tests on predefined character templates to produce classification results.

(Tangwongsan & Jungthanawong, 2008) presented a refinement stoke features for printed Thai character recognition. In this work, a set of stoke features, which consists of a number of loops, character components, connected stoke, and location in both horizontal and vertical for each component, was proposed. The characters were clustered into 12 groups of consonants, digits, special symbols, and 8 groups of vowels and tone marks based on the predefined physical properties rules in the classification stage. As characters in the same group may have similar physical features, however, the stoke attributes are different, causing each character to be classified into distinguished classes.

(Chaivatna & Supachai, 2010) presented a method for recognizing Thai historical documents. Since the traditional ThaiOCRs usually perform well on clean, sharp, and handprinted Thai documents. However, these characteristics did not appear on historical documents, which suffer from a lot of degradation problems such as, fading intensity of ink coupled with the yellowish nature of the historical codex background. The model λ, which was specific towards particular Thai font characteristics and language, was proposed. First, to generate a sequence of broken character regions, the proposed set-partitions method was applied. For each segmented region, the proposed algorithm generates top-N possible potential partition sets based on the defined sizing and well-formedness score. These potential sets were recognized by a heuristic search to find the most readable character sequences from the predefined lexicon.

The bilingual OCR for Thai and English were proposed by (Tangwongsan & Suvacharakulton, 2012). This work can be separated into three main stages: language identification, character recognition, and error correction. Language can be identified for the first stage through the unique geometric and characteristic properties between Thai - English and a set of predefined decision rules. Second, four groups of character feature consisting of geometric, termination, and intersection point, vertical and horizontal lines, and character loops were extracted from the thinned character image. These features were used in two levels handcrafted decision trees to identify the input character into 18 groups at the coarse stage and then classify into individual character class at the fine classification stage. Finally, the recognized characters were grouped into word level and corrected by the predefined dictionary.

(Mitrpanont & Imprasert, 2011) presented a novel zig-zag feature for Thai handwritten recognition. First, the zig-zag feature was extracted from a 64x64 pixel segmented character image in a vertical direction, acquiring 64x1 vector fed into zig-zag and non-zigzag MLP classifier. The zig-zag classification output was used in MLP in combination with geometric and additional features for special characters.

(Wiwatcharakoses & Patanukhom, 2013) proposed two-stage classifiers for Thai and English characters recognition. The four groups of scale-invariant feature, ratio, edge projection, boundary, and Pyramid Histogram of Oriented Gradients (PHOG) features were proposed as each character feature. These features were fed into the coarse classifier, which was fuzzy c mean clustering (FCM) to find the K nearest classes of given input features vector. In the refinement stage, the fine classifier was selected depends on the result from the coarse classifier. SVM was chosen in case that the result satisfies the performance on validation data to reduce computational complexity while the simple kNN was used in other cases.

(Siriteerakul, 2013) utilized HOG as the main features extractor for Thai character recognition task. In this work, HOG feature was extracted from a 24x24 segmented character image to form a character feature vector for SVM classifier. This research also showed the recognition accuracy comparison between using simple intensity value and HOG as a feature extractor.

(Iamsa-at & Horata, 2013) presented a Deep Learning Feedforward Backpropagation Neural Network (DFBNN) for Thai handwritten recognition. The HOG feature vector was formed by extracting from the normalized 32x32 input image. The output feature vector was fed into 70-way DFBNN classifier. This paper also performed tests on multiple configurations of DFBNN hidden nodes number and compares the output accuracy with Extreme Learning Machine (ELM) classifier.

Nation Electronics and Computer Technology Center (NECTEC) held a national competition on Thai printed character recognition and released the competition report in (Ithipan Methasate & Marukatat, 2013). There were four algorithms proposed in this competition as follows:

- HOG and SVM based Thai character recognition.
- Three stages Thai character recognition method. In the first stage, k-mean cluster was used to separate the input global feature vector into defined character groups. Second, the SVM classifier in each group was used to produce the output into individual class. Furthermore, the convexity features were used in combination with the result from the second stage in a case that the SVM cannot accurately classify the output class.
- Template Matching based method. The geometric features were extracted and Singular Vector Decomposition (SVD) was used to build a covariance matrix as a template for each class. In the recognition phase, the input image was measured with the stored template using cosine similarity.
- Wavelet feature and two stages classifier method.

In (Fukue et al., 2017) work, multi-level feature points definition (MFFD) was used as the character feature extractor. Each handwritten blob character was segmented into individual character and fed into Individual change control processing (ICCP) to normalize the input. Then, the normalized character was fed into MFFD to produce a multi-level feature vector. To classify each feature vector, template matching was used in combination with simple Euclidean distance.

The novel circular scanned histogram was proposed by (Kaothanthong et al., 2017) as Thai character feature extractor. The proposed circular scanned histogram was a scale-invariant distance-based feature calculated by measuring the scan line distance between thinned character image and origin coordinate. The extracted feature was then classified using $L1$ distance matching with the predefined template.

From the presented handcrafted feature based Thai optical character recognition works, the overall pipeline can be concluded into four stages diagram as shown in Figure 9.

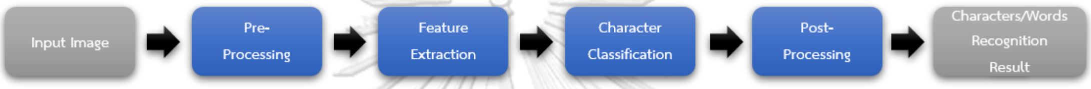


Figure 9 Standard pipeline for handcrafted feature based ThaiOCR.

Pre-processing stage – In this stage, the input image is segmented into an individual character based on the proposed algorithm in each method, for example, horizontal and vertical line scan, connected component analysis, or simple hand-segmentation.

Feature Extraction stage – The segmented character from the previous stage is applied into the proposed feature extraction methods such as pixel-based feature, shapes and statistic features, or well-known HOG feature.

Character Classification stage – The extracted feature is then fed into proposed classifiers such as predefined rules, neural network, decision tree, k-mean clustering, or template matching.

Post-processing stage – The recognized character may concatenate or merge into word level and perform error correction by using the predefined dictionary or lexicon.

From the above handcrafted feature based ThaiOCRs pipeline, the presented methods can be concluded into Table 6.

Table 6 Handcrafted feature based ThaiOCRs.

Proposed By	Input Image	Pre-processing	Feature Extraction	Classifier	Post-Processing
(Vel et al., 1995)	Cropped Character	-	Simulated light feature (intensity value)	Neural Network	-
(Tanprasert & Koanantakool, 1996)	Scanned Document	Line and Character Segmentation	Intensity value	Neural Network	-
(Kijsirikul et al., 1998)	Cropped Character	-	Direction and regions features	Predefined rules (from Progal) and Neural Network	-
(Mitatha et al., 2001)	Cropped Character	-	Number of black pixels from 32 4x4 patches	Predefined rules	-
(Watjanapong & Chom, 2001)	Cropped Character	-	Horizonral and vertical pixel projection	Predefined rules	-
(Thongkamwitoon et al., 2002)	Cropped Character	-	Primary (number of loops, island, and geometric features) and Secondary (ratio-based features)	Decision Tree	-

Proposed By	Input Image	Pre-processing	Feature Extraction	Classifier	Post-Processing
(Budsayaplakorn et al., 2003)	Cropped Character	-	Stroke based features	Hidden Markov Model (HMM)	-
(Roongroj & Povey, 2003)	Cropped Character	-	Intensity value and polar transform of input image	Hidden Markov Model (HMM)	Handcrafted feature in combination with results from HMM
(Pornchaikajornsak & Thammano, 2003)	Cropped Character	-	Intensity value from defined regions	ARTMAP (Neural Network)	-
(I Methasate & Sae-tang, 2004)	Cropped Character	-	Stroke based features	Neural Network	-
(Thammano & Duangphasuk, 2005)	Cropped Character	-	Stroke based feature (direction code)	ARTMAP (Neural Network)	-
(I Methasate et al., 2005)	Cropped Character	-	Statistic and symbolic features	2 stage Neural Network	-
(Sa-ngamuang et al., 2007, p.)	Cropped Character	-	Essential features	Predefined rules	-
(Tangwongsan & Jungthanawong, 2008)	Cropped Character	-	Stroke based features	K-mean clustering and Rule-Based	-

Proposed By	Input Image	Pre-processing	Feature Extraction	Classifier	Post-Processing
(Chaivatna & Supachai, 2010)	Cropped Character	-	Intensity value (pixel set partition)	Rule-based heuristic search	-
(Tangwongsan & Suvacharakulton, 2012)	Cropped Character	Character language identification	Geometric and stroke-based features	Decision Tree	-
(Mitrpanont & Imprasert, 2011)	Cropped Character	-	Zigzag and geometric features	Neural Network	Geometric features in combination with results from Neural Network
(Wiwatcharakoses & Patanukhom, 2013)	Cropped Character	-	Edge projection, geometric and PHOG features	k-NN and SVM	-
(Siriteerakul, 2013)	Cropped Character	-	HOG	SVM	-
(Iamsa-at & Horata, 2013)	Cropped Character	--	HOG	DFBNN and ELM	-
(Fukue et al., 2017)	Handwritten blob	Character Segmentation	Multi-level feature points definition (MFFD)	Template matching using Euclidean	-

Proposed By	Input Image	Pre-processing	Feature Extraction	Classifier	Post-Processing
		and Individual change control processing normalization		distance	
(Kaothanthong et al., 2017	Cropped Character	-	Circular scanned histogram	Template matching using L1 distance	-

From the previous works on handcrafted feature extraction Thai character recognition methods mentioned above, none of them focuses on Thai scene text recognition. Furthermore, the used feature extractors are built based on human perception on the printed character in the scanned documents, which may not work well under complex scene text conditions such as occlusion, blur, alignment, and poor lighting conditions.

2.3.2. Learned Feature Based Approaches

In contrast to the handcrafted feature-based approaches, the methods in this group utilize the learnable CNN feature extractor. There are many interesting methods in this group which are described below.

In 2016, a non-segmentation LSTM based Thai character recognition method was presented by (Emsawas & Kijsirikul, 2016). Since segmentation is a crucial component in character recognition, the unsegmented method for Thai character recognition in scanned documents was proposed in this work. As Thai character alignment can be separated into four levels, a large number of vertically occurring character combinations can occur. The novel vertical component shifting using defined CCA rules, which aim to separate the vertically occurring characters (tone marks) into individual components, was proposed to overcome this problem. The output image from vertical component shifting was fed into bidirectional-LSTM to produce the recognition result for each input image sentence. However, in this work, they did not mention about the used features.

(Chomphuwiset, 2017) proposed the CNN based Thai character recognition method. It began with character segmentation from a document by using Otsu's thresholding. Each segmented character was fed into five layers CNN which used standard logistic softmax function to derive a loss-function. In this work, the proposed CNN network achieved the best performance among the other handcrafted features, which were chained code, Hu moments, and HOG.

(Chamchong et al., 2019) proposed CNN-RNN based Thai handwritten recognition from Thai archive manuscript. In this work, the text recognition problem was treated as a sequence-to-sequence problem. First of all, the input document was segmented into line-level by using human annotation. Then, each line was fed into nine layers CNN-RNN structure and the output character sequence was learned by using standard CTC loss.

Similar to the handcrafted feature based Thai character recognition, none of the presented methods focus on Thai scene text recognition. Even if the features are learned from training data, the used training data are created based on the printed character in the scanned documents assumption.

2.4. Thai Scene Text Localization

As Thai characters contain some specific features that do not appear in English, an existing English-based scene text localization may not work well. Similar to the English scene text localization, the Thai text localization algorithms can be divided into two groups: the connected component analysis and the object detection-based approaches.

2.4.1. Connected Component Analysis Based Approaches

According to the details in each previous works on CCA base scene text localization, most of the existing works are English text localization. Since Thai contains specific characteristics that do not appear in English such as, multi-level vowels and tone marks, punctuation between sentences, these characteristics make the existing English based scene text localization may not work well. The existing Thai scene text localization methods are listed below.

(Jirattitichareon & Chalidabhongse, 2006) proposed an automatic text detection and segmentation from low quality sign images. In this work, they stated four text assumptions as follows.

- Text is designed with high contrast to its background in both of color and intensity.
- Each character is composed of one or several connected regions.
- The characters in the same context have almost the same in both of size and intensity but may be different in color.
- The characters in the same context have almost the same background patterns.

Image pre-processing and Laplacian of Gaussian (LoG) were performed on the input image to generate the edge map. Then, candidate text regions were created using rule-based CCA. The non-text regions were filtered out by region ratio and Thai text alignment criteria. Finally, Gaussian Mixture Model (GMM) was used to extract foreground and background pixels to create the output text areas. Since this work focused on text localization and segmentation from sign images, the proposed method might not work well with text in natural scene images, which have much larger variations.

(Woraratpanya et al., 2013) proposed a method for Thai scene text localization. The scene text characteristic assumptions in this work were close to (Jirattitichareon & Chalidabhongse, 2006). This work used Canny edge detector to generate an edge map from the input image. Then, Fast Boundary Clustering (FBC) was performed by extracting the color features from each object edges pixel and fed to k-mean clustering algorithm. Finally, CCA based on defined Thai text characteristics was performed to generate the text output.

(Woraratpanya et al., 2014) proposed a text-background decomposition for Thai text. In this work, the upgraded version of FBC from (Woraratpanya et al., 2013) was proposed. Instead of using a fixed K value, which is equal to five in their old work, they quantized the color of edge pixel into eight predefined colors and transformed the output into a histogram. As a result, the K value for k-mean clustering would depend on the number of histogram bins, which were greater than the mean pixel value. Then, the n-Point boundary clustering was performed to extract each character individually by using the color feature. This work did not include the method for text bounding box groping.

(Wiwatcharakoses & Patanukhom, 2015) proposed a MSER based text localization algorithm. The MSER components were extracted from the input image and a set of geometric and color features were gathered from each component. The MSER components and their connectivity were modeled as an undirected graph. The edge weights were calculated based on spatial distance, color difference, size difference, and stroke width difference between each node. Finally, the components in each node were grouped by using defined constraints.

From the presented connected component analysis based Thai scene text recognition method, the overall pipeline can be concluded into the diagram shown in Figure 10.

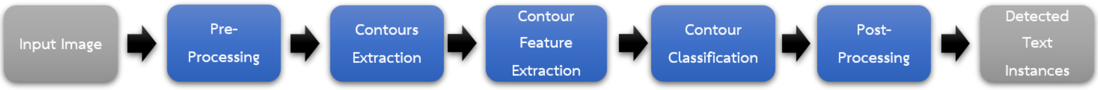


Figure 10 Overall pipeline for connected component analysis based Thai scene text localization.

Pre-processing stage – In this stage, the input image is being pre-processed to discard foreseeable clutters or improve the text instances visibility by using well-known filters such as gaussian or median filter.

Contour Extraction stage – The potential contours are extracted from the preprocessed image by using well-known techniques, for example, sobel, canny edge detection, and MSER.

Contour Feature Extraction stage – The handcrafted features, for example, geometric features, colors, and stroke width, are gathered from each extracted contour.

Contour Classification stage – The contours are classified into two classes: text and non-text contours by using the extracted features. The classifier can be either predefined rules or machine learning techniques

Post-processing stage – The potential text contours are grouped into text regions/instances based on defined human observable criteria.

From the above connected component analysis based Thai scene text localization pipeline, the presented methods can be concluded in Table 7.

Table 7 Connected component analysis based Thai scene text localization.

Proposed By	Pre-Processing	Contour Extraction	Contour Feature Extraction	Contour Classifier	Post-Processing
(Jirattitichareon & Chalidabhongse, 2006)	Laplacian	LoG & CCA	Geometrics	Ruled-Based	Text Grouping
(Woraratpanya et al., 2013)	Color Quantization	CCA	Colors	Ruled-Based	Text Grouping
(Woraratpanya et al., 2014)	Color Quantization	CCA	Colors	Ruled-Based	Text Grouping
(Wiwatcharakoses & Patanukhom, 2015)	-	MSERs	Colors and geometrics	Ruled & Graph Based	Text Grouping

Unfortunately, the presented methods use text analyzing techniques based on some specific characteristics of sign and text on non-complex background images that may not work well on complex scene images, which contain lots of higher variations of text images. Furthermore, the proposed textness features, for example, contours, edges, and geometric features, may not be capable of representing complex scene text instances.

2.4.2. Object Detection Based Approaches

Sliding window-based scene text localization methods handle text regions as objects in object detection and localization works. Typically, each patch/window is fed to a chosen classifier, such as Neural Network (NN), SVM, or CNN to classify between text and non-text. The post-processing techniques are used to combine the results from text classifier into text region bounding boxes. The existing sliding window based Thai scene text localization methods are described below.

Our previous work in 2014, (Kobchaisawat & Chalidabhongse, 2014), a CNN based Thai scene text localization algorithm was presented. We employed a CNN text detector and with NMS to generate the candidate text bounding boxes in this work. Since we need to create accurate bounding boxes for Thai text, we proposed a novel Thai text characteristic analysis to refine the text bounding box by using CCA based on the Thai language writing styles. As a result, our method can localize Thai text more accurately compared to other English based methods.

In 2015, a multi-oriented text localization version of our previous work was proposed in (Kobchaisawat & Chalidabhongse, 2015). A sliding window-based CNN text detector at multiple scales input image to generate text confidence map at the original scale is also applied in this version. The proposed text line estimation based on CCA and linear regression were used to form the output accurate text lines. Finally, Thai text characteristic analysis was applied to construct the appropriate text bounding boxes.

Even though the presented methods were specifically designed for Thai scene text, they may not be able to handle text instances in some complex cases such as irregular shapes and vertical text. Furthermore, the presented Thai scene text localization method also a huge amount of computational power because of not only all regions in the input image need to be extracted and classified but also all possible scales.

3. Related Theories

3.1. Convolutional Neural Network (CNN)

Before describing the convolutional neural network, the typical neural network in the feed-forward pass is presented below. The network is illustrated in Figure 11.

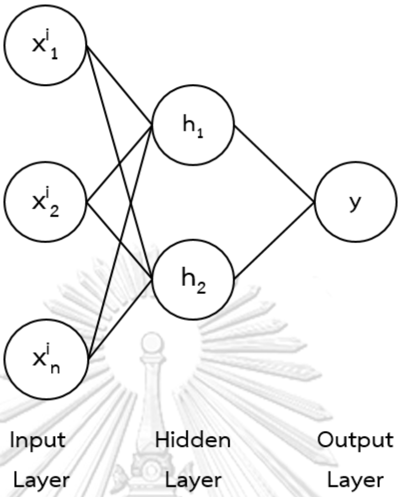


Figure 11 Typical neural network structure

Considering a set of training data $\{x^i, y^i\}$, where $\mathrm{x^i}$ and y^i denote feature vector and label for i^{th} element. Given such a nonlinear function $f_w(x)$, at the high-level intuition, neural network is learned to represent the x by adjusting $f_w(x)$ to fit the training data. The function $f_w(x)$ is tuned to minimize the chosen loss function by parameterizing the weight matrix $\mathbf{w}$. Figure 11 denotes the three-layer neural network, where x and y represent input and output vector, which can be defined as $y^i = f_w\left(x^i_1, x^i_2, \dots, x^i_n\right)$

In neural network, the neuron units in each layer take input vector from previous layers. Considering h_1 neuron in the hidden layer, its input vector is $[x^i_1, x^i_2, x^i_3]$. The output of this neuron can be computed as the linear combination of the input vector and weight by $a_1 = \sum_{j=1}^{3} \mathrm{x^i} w_{1j} + b_1$. This equation can be further extended to n elements feature vector and output a from k^{th} neuron as shown below

$$a_k = \sum_{j=1}^{n} \mathrm{x^i} w_{kj} + b_k$$

where w and b represent the connection weight between a considering pair of neuron and bias term. Since real-world data is usually nonlinear, activation function or nonlinear function is applied to the output of neuron a_k. Many common activate functions such as hyperbolic tangent (tanh), rectified linear unit (ReLU), Swish, Mish, and sigmoid can be chosen. Hence, the output from an activated neuron k can be represented by

$$z_k = f_w(a_k) = f_w\left(\sum_{j=1}^{n} \mathrm{x}^{\mathrm{i}} w_{kj} + b_k\right)$$

where f_w is chosen activation function. From this calculation thought out each layer, the output $y^i = f_w\left(x_1^i, x_2^i, x_3^i\right)$ for the i^{th} element data can be obtained. The weight matrix $\mathbf{w}$ and bias $\mathbf{b}$ are learnable parameters which are adjusted to minimize the given loss or objective function.

Convolutional Neural Network (CNN) is a special kind of neural network with different architectures, connections, and layer types. Considering 32x32 pixels RGB image, this image can be represented in 32x32x3 matrix. In typical three-layer fully connected neural network, the total weights parameter between an input and a hidden node would be 3,072 parameters. Nevertheless, when it comes to visual recognition problems, the adjacent or local area pixels tend to be strongly correlated while the others appear to be uncorrelated, this characteristic appears in CNN but not in the traditional neural network. This locality property can be seen from many famous computer vision feature extractors such as HOG, SURF, BRIFE, and SIFT.

The second property of CNN is weight sharing. Given a top left (0,0) pixel in a grayscale image in the traditional neural network, this pixel has its weight matrix connected to each neuron in the hidden layer and other pixels. On the other hand, for CNN, the same weight matrix is used to evaluate the layer’s output over the input value. This characteristic can reduce many free parameters to adjust during the training process, make it easier to train CNN and less overfit phenomenon occurs.

The third distinguish between CNN and the typical neural network is pooling. In CNN, there is a layer called “Pooling Layer”. The main objective of this layer is to reduce the dimension and condense the information of the convolution layer output.

In a visual recognition CNN, the output from each layer is called feature maps. The typical layer types which can be found are listed below

- Convolution Layer
- Pooling Layer
- Nonlinearity Layer
- Fully Connected Layer
- Dropout Layer
- Batch Normalization Layer
- Softmax Layer

Convolution Layer – The convolution layer applies a correlation operator between input and weight or filter matrix. Mostly, the convolve operator is employed in valid border mode as shown in Figure 12.

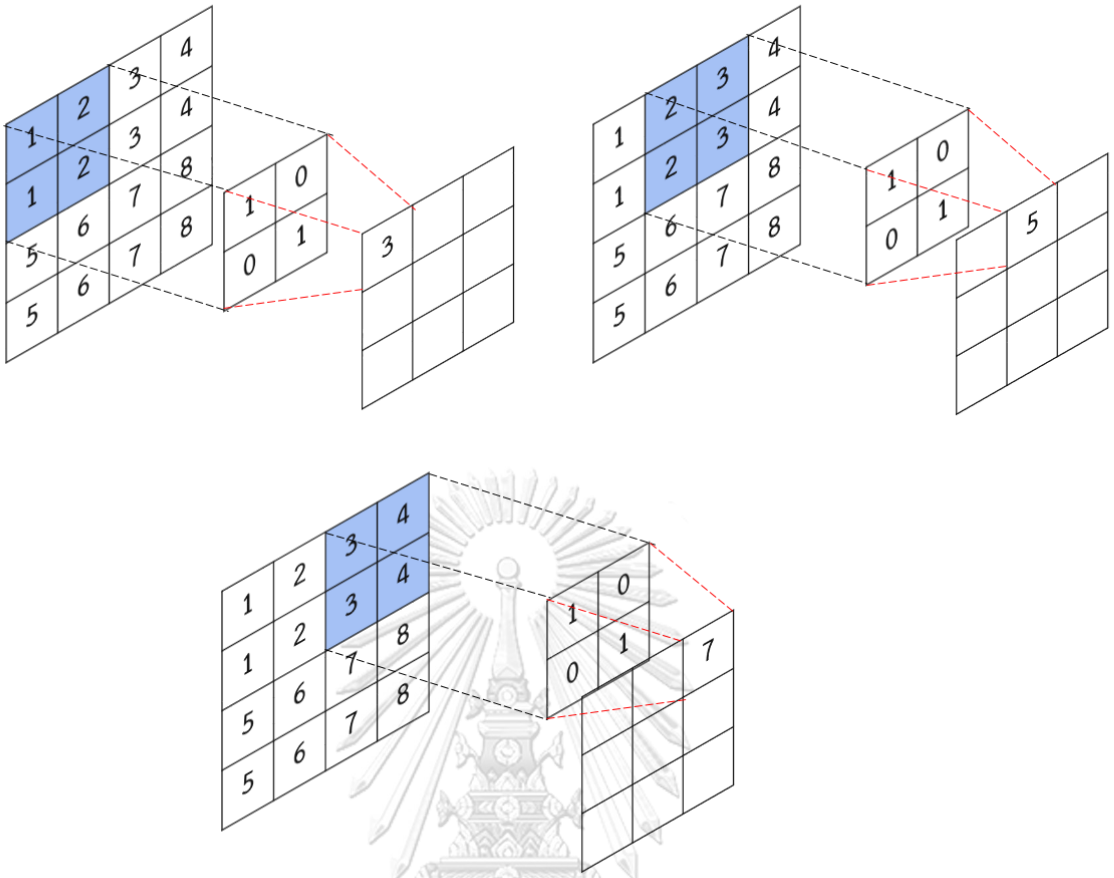


Figure 12 An example of valid convolution operation on 4x4 input image and 2x2 kernel.

Typically, convolution layer has four configurable parameters:

- P amount of input border padding in pixel
- F filter size
- S convolution strides
- W input size

The output feature map spatial size can be calculated from $\frac{(W-F+2P)}{S}+1$. For example, given 32x32x3 input images to 3x3x3x16 convolution layer with stride = 1 and no border padding. The output feature size would be $\frac{(32-3+2(0))}{(1)}+1=30.$

Pooling Layer - This layer plays an important role in reducing and condensing the input spatial dimensions. It can reduce learning parameters and computation complexity. There are many types of pooling functions, the most common of which is max or average function. Given an input size 4x4, the output from 2x2 max pooling with 2 strides and no border padding is illustrated in Figure 13.

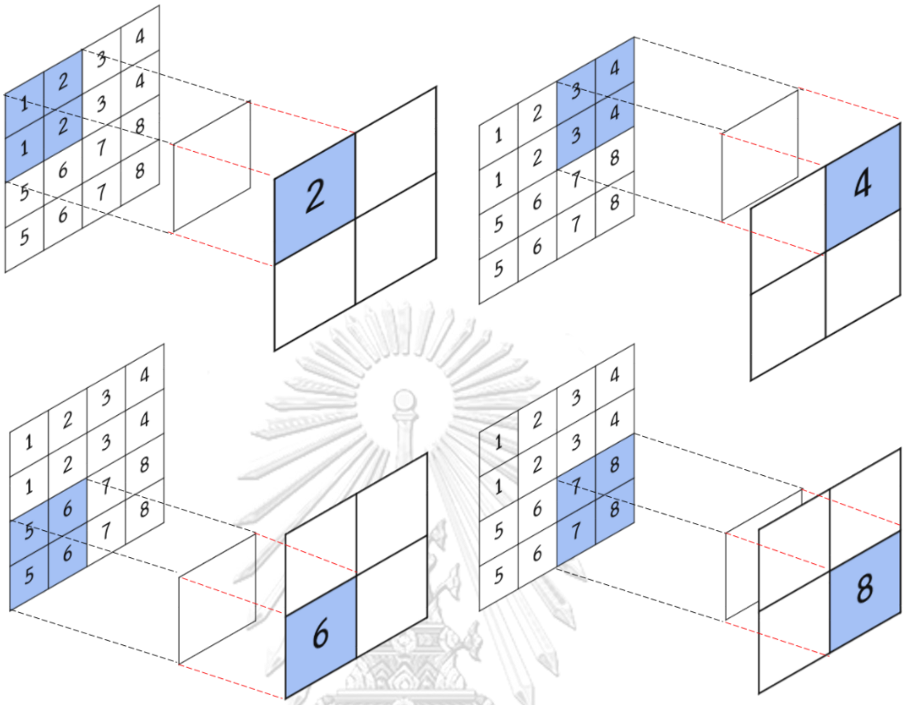


Figure 13 An example of max pooling on 4x4 input matrix and 2x2 kernel.

Nonlinearity Layer – As convolution is a linear operator, but generally, the real world is not. In CNN, many nonlinear or activate functions can be chosen, for example, hyperbolic tangent (tanh) or rectified linear unit (ReLU) defined in the following equation. Typically, the nonlinear layer usually follows the convolution layer.

$$f(x) = max(0, x)$$

Fully Connected Layer – Since the output from the convolution layer represents high-level data features, adding a fully connected layer is one way to learn the nonlinear combination of these features. This layer usually follows the behavior of the hidden layer in the traditional neural network. Given an input matrix size of $w \times h \times d$ from the previous convolution layer, the input is flattened into a column vector, yields a 2D input vector size of $(w \times h \times d) \times 1$.

Dropout Layer – According to (Srivastava et al., 2014), a dropout layer tends to reduce the overfit phenomenon in CNN by randomly set some locations on features map to zero. Overfitting occurs when the trained model is too complex or contains too much parameter relative to the observation parameters.

Batch Normalization Layer – As mentioned in (Ioffe & Szegedy, 2015), batch normalization can improve training stability, regularization, and model generalization by mitigating the internal covariance shift due to randomness in parameter initialization. Small changes in shallower layers can be amplified throughout deeper layers, resulting in serve gradient exploding, or vanishing problems when training a very deep network. Batch normalization can cure this problem by normalizing the output of a previous layer by subtracting the batch mean and dividing by the batch standard deviation. Both mean and standard deviation in each batch are learnable parameters.

Softmax Layer – In many classification problems, the output should be in the probability for each class. The output vector from another layer, for example, fully connected layer, is difficult to interpret since they are in floating point form without boundaries. Softmax function gives a slightly more intuitive output (normalized class probabilities). This function can be defined by the following equation.

$$f(x) = \frac{e^x}{\sum e^x}$$

where x represents the feature vector from the previous layer. $f(x)$ gives probability vector for each class in range $[0,1]$.

3.2. Long-Short Term Memory (LSTM)

In many tasks, for example, speech recognition and video captioning, the current context is based on previous data. In order to get the computer to know the current video context, the captioning algorithm may have to understand each frame context based on previous frames. The information from previous frames persists and is used to inform the current frame caption. Since a traditional neural network cannot overcome this issue, a kind of loop network called recurrent neural network (RNN) is introduced.

Considering data sequence $\boldsymbol{x}$ and network A, where $\boldsymbol{x}_t$ denotes feature vector at time t and output label y_t. The loop recurrent neural network and its unrolled version are shown in Figure 14. The existing loop in a recurrent neural network makes the previous information can be passed to its successor layer.

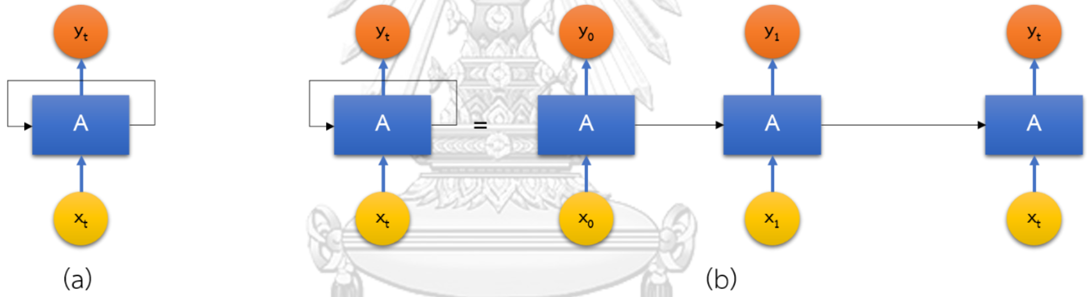


Figure 14 (a) Typical recurrent neural network

(b) Unrolled recurrent neural network

If a network hidden state is denoted as $\boldsymbol{h}$, the memory state at time t, $\boldsymbol{h}_t$, can be computed from the following equation.

$$\boldsymbol{h}_t(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) = f(\boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{W}\boldsymbol{h}_{t-1})$$

where $\boldsymbol{U}$ and $\boldsymbol{W}$ represent weight matrices of the desired hidden layer, f is a chosen nonlinear function such as hypobolic tangent (tanh) or rectified linear unit (ReLU).

However, it is difficult to train the standard RNN on long-term temporal dependencies because of the gradient vanishing problem. (Hochreiter & Schmidhuber, 1997) presented a special kind of RNN, which can avoid long-term dependencies problem, called Long-Short Term Memory Network (LSTM).

Instead of using a single hidden layer neural network, LSTM has three special hidden layers called gates. A typical LSTM structure is shown in Figure 15.

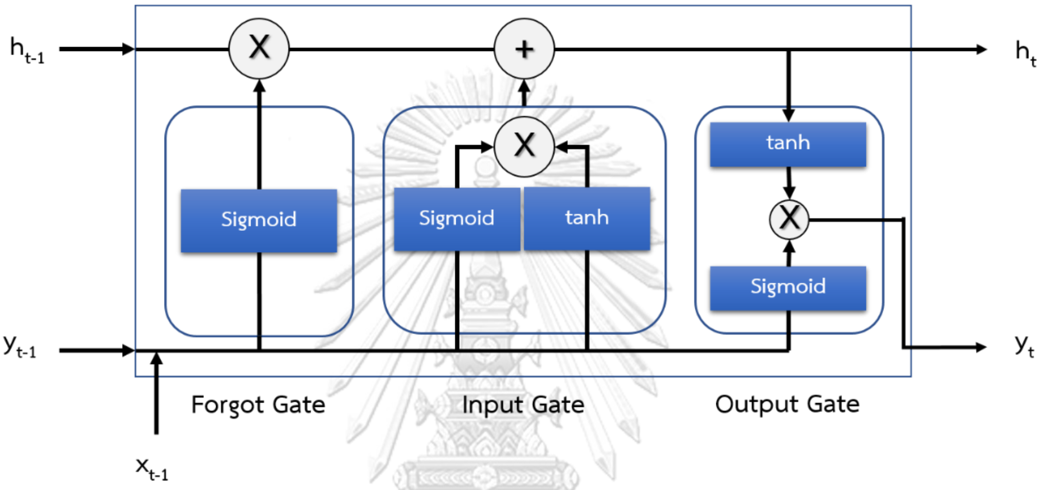


Figure 15 Typical LSTM block diagram.

Considering the previous state result y_{t-1}, the forgot gate controls a mechanism to delete or keep the old information from predecessor state. The result f_{forgot} from this gate can be represented as the following equation.

$$f_{forgot}(\boldsymbol{x}_t, y_{t-1})_t = \sigma(\boldsymbol{W}_{forgot} \times [y_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_{forgot})$$

where $\boldsymbol{W}_{forgot}$ and $\boldsymbol{b}_{forgot}$ represent forgot gate weight and bias matrices, respectively. Since the range of sigmoid function is $[0,1]$, the value 1 can represent the meaning of “keep all information” while 0 means “discard all information” from its predecessor.

The input gate duty is to decide which new feature needs to be stored in the current hidden state. The output vector $\boldsymbol{\sigma}_{input}$ from sigmoid function decides which feature needs to be updated while tanh layer is generated the replacement feature vector $\boldsymbol{T_{input}}$ for the current state. The output f_{input} from this gate can be expressed by the following equation.

$$\sigma_{input}(x_t, y_{t-1})_t = \sigma(\boldsymbol{W_{input\_\sigma}} \times [y_{t-1}, x_t] \ + \ \boldsymbol{b_{input\_\sigma}})$$

$$T_{input}(x_t, y_{t-1})_t = tanh(\boldsymbol{W_{input\_tanh}} \times [y_{t-1}, x_t] \ + \ \boldsymbol{b_{input\_tanh}})$$

$$f_{{input}_t} = \sigma_{input}(x_t, y_{t-1})_t \times T_{input}(x_t, y_{t-1})_t$$

where $\boldsymbol{W_{input}}$ and $\boldsymbol{b_{input}}$ represent the input gate weight and bias matrices for sigmoid and tanh layer, respectively. The current hidden memory state $\boldsymbol{h_t}$ can be expressed by the following equation.

$$h_t = \left(f_{forgot}(x_t, y_{t-1}) \times h_{t-1}\right) + f_{input}(x_t, y_{t-1})$$

The output gate produces the output result $\boldsymbol{y_t}$ based on current hidden memory state $\mathrm{h_t}$.

$$y_t = \sigma\left(\boldsymbol{W_{output}} * [y_{t-1}, x_t] \ + \ \boldsymbol{b_{output}}\right) \times tanh(h_t)$$

3.3. Spatial Transformer Network (STN)

Spatial transformer network (STN) is a specialized layer in CNN proposed by (Jaderberg et al., 2015). This kind of layer contains spatial transform modules that attempt to make the network actively and spatially invariant to the input images. As a typical convolution layer lacks an ability to be spatial invariance in a computationally and parameter efficient manner, STN may be able to make the challenging input becomes much easier to identify through its flexible transformation. Mainly three transformations that can be learned by STN are shown below.

1. Affine transformation is a linear transformation that preserve collinearity and the ratios of distances between points on a line. Consider a source point (x_1, y_1), destination point (x_2, y_2) and transformation matrix $\boldsymbol{A_T}$, affine transformation can be defined as follows.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \boldsymbol{A_T} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & 0 \\ T_{21} & T_{22} & 0 \\ T_{31} & T_{32} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

Translation, rotation, scaling and shear transformation can be expressed in transformation matrix A_T as shown in the following Table 8.

Table 8 Learnable affine transformations in Spatial Transformer Network

Transformation	Transformation Matrix $\boldsymbol{A_T}$	Parameters
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	t_x and t_y are displacement along x and y axis, respectively
Rotation	$\begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	θ is the angle of rotation
Scaling	$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	s_x and s_y are scaling factors along x and y axis, respectively
Shear	$\begin{bmatrix} 1 & sf_x & 0 \\ sf_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	sf_x and sf_y are shearing factors along x and y axis, respectively

2. Projective transformation (Homography) is a linear transformation which does not preserve parallelism, length, and angle but still preserves collinearity and incidence. The different between projective and affine transformation is illustrated in Figure 16.

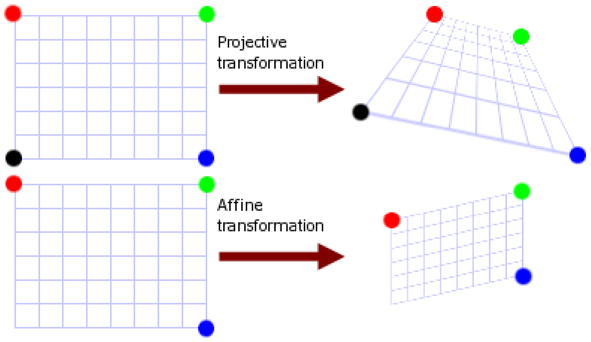


Figure 16 Comparison between projective and affine transformation[1]

Consider a source point (x_1, y_1), destination point (x_2, y_2), and projective transformation matrix $\boldsymbol{P_T}$, affine transformation can be defined as follows.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \boldsymbol{P_T} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

3. Thin plate spline transformation (TPS) is a non-rigid transformation model used in many tasks such as polygon matching and image alignment.

Considering i^{th} control points (x_c^i, y_c^i) and input point (x_s, y_s), the input point is transformed to target point (x_t, y_t) base on the following equations.

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^{n} F r_i^2 \ln r_i^2 \\ \sum_{i=1}^{n} G r_i^2 \ln r_i^2 \end{bmatrix}$$

[1] https://www.graphicsmill.com/docs/gm5/Transformations.htm

$$r_i^2 = (x_s - x_c^i)^2 + (y_s - y_c^i)^2$$

where $r_i^2 ln r_i^2$ is a thin plate spline radial basis function. The target point is calculated based on learnable variables a, b, and distance between source and al control points, which allows flexible transformation can be performed.

Spatial transformer mechanism composed of three parts, localization net, grid generator, and sampler as shown in Figure 17.

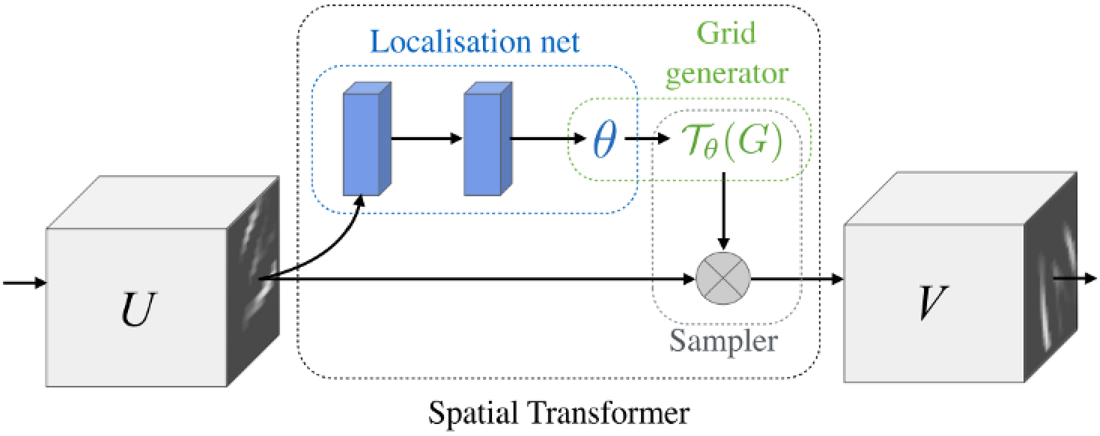


Figure 17 Spatial Transformer Network diagram (Jaderberg et al., 2015)

1. Localization net - Considering an input image or feature map U size of $H \times W \times C$, a source coordinate (x^s, y^s), destination coordinate (x^t, y^t), the localization network which is composed of multiple convolution layers and fully-connected layers, will generate the transformation matrix $\boldsymbol{T_\theta}$ or the parameter θ_{ij} as shown below.

$$\begin{bmatrix} x^s \\ y^s \\ 1 \end{bmatrix} = \boldsymbol{T_\theta}(G) = T_\theta \begin{bmatrix} x^t \\ y^t \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{bmatrix} \begin{bmatrix} x^t \\ y^t \\ 1 \end{bmatrix}$$

Since the source coordinate is known, but the destination is not, all original source coordinates are computed from the network transformation parameters.

2. Grid Generator - Considering a grid G, the output from $T_\theta(G)$ is a point which is a set of points with destination coordinate (x^t, y^t), where the parameter θ_{ij} are learn from the localization net. The output sample from grid generator is shown in Figure 18.

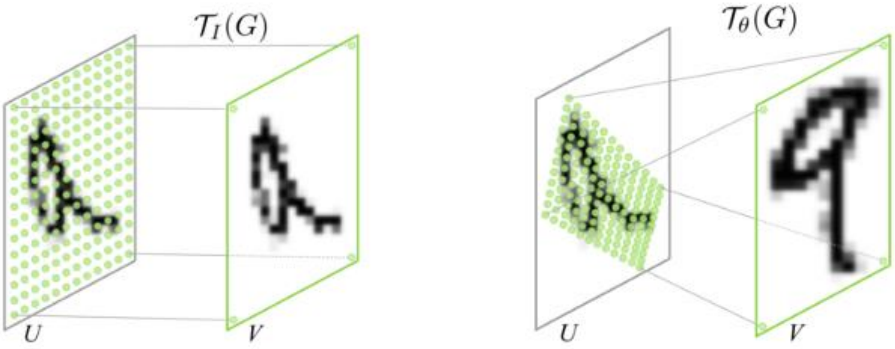


Figure 18 Sampling grid output

(a) Identity transformation when transformation matrix T_θ is an identity matrix

(b) Affine transformation

3. Sampler – As the set of destination coordinate can be aligned in arbitrary shape, the intensity for each point needs to be interpolated from the original value, which can be pixel or sub-pixel sampling. This work proposed two types of differentiable sampling kernels integer and bilinear sampling which can be defined as

Integer sampling kernel:

$$V_{x^t y^t} = \sum_{n}^{H}\sum_{m}^{W} U_{nm}\delta(\lfloor x^s + 0.5\rfloor - m)\delta(\lfloor y^s + 0.5\rfloor - n)$$

Bilinear sampling kernel:

$$V_{x^t y^t} = \sum_{n}^{H}\sum_{m}^{W} U_{nm} max(0,1 - |x^s - m|) max(0,1 - |y^s - n|)$$

where U_{nm} and $V_{x^t y^t}$ refer to the input value at coordinate (n, m) and the output value at coordinate (x^t, y^t), δ. is the Kronecker delta function, respectively.

3.4. Attention Mechanism

Attention is a mechanism that provides a clearer encoded source sequence from which to construct a context vector that can then be used by the decoder. This mechanism enables the model to learn what positions or features in the source sequence to pay attention and how much the weights during the prediction of each step in the target sequence. There are many types of attention models. This dissertation focuses on the attention mechanism proposed by (Bahdanau et al., 2015) (Additive Attention), which has been used in many neural machine translation works.

RNN Encoder-Decoder structure is used in many previous machine translation works. Given an input sequence X, the encoder is transformed X into intermediate vector C. The hidden state at time step t of RNN in encoder can be defined as

$$h_t = f(x_t, h_{t-1})$$

$$c = q(\{h_1, \dots, h_{Tx}\})$$

where f and q are a nonlinear function depends on RNN implementation.

For the decoder stage, by giving the intermediate vector c, decoder predicts the next token y_t depends on the previous predicted token $\{y_1, \dots, y_{t-1}\}$, which can be formulated as

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c)$$

where g is a nonlinear function which produces the probability of y_t. s_t is the hidden state of RNN at time step t.

Unfortunately, the RNN based methods still struggle on long term dependency sequence and the gradient vanishing problem. (Bahdanau et al., 2015) proposed a new aspect which can be visualized in Figure 19.

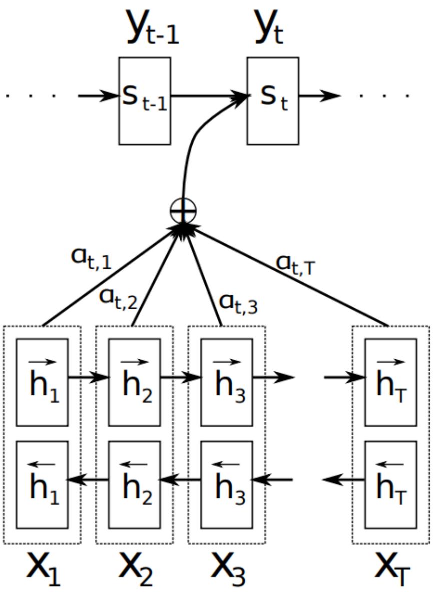


Figure 19 Encoder-Decoder with Attention (Bahdanau et al., 2015)

For the encoder stage, given input sequence X, to capture not only the preceding tokens but also the following tokens. Thus, this work utilizes bidirectional long-short term memory (BiLSTM) to capture the context from both sides. The hidden state h at time step t are concatenated from forward and backward hidden state and can be defined as follows.

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ represent the forward and backward hidden state of BiLSTM.

For the decoder stage, the conditional probability for the next token y_t depends on the previous predicted token $\{y_1, \ldots, y_{t-1}\}$ can be formulated as .

$$p(y_t | \{y_1, \ldots, y_{t-1}\}, x) = g(y_{t-1}, s_t, c_t)$$

where s_t represents hidden state at time step t, which can be defined as.

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

The intermediate vector c_t depends on the hidden state vector h where h_t compact the whole input sequence information with a strong focus on the t^{th} position. c_t can be computed as a weighted sum of h_t as shown below.

$$c_t = \sum_{t=1}^{T} \propto_{tu} h_t$$

The weight $\propto$ for each hidden state h_t can be computed as follows.

$$\propto_{tu} = \frac{exp(e_{tu})}{\sum_{v=1}^{T} exp(e_{tv})}$$

$$e_{tu} = a(s_{t-1}, h_u)$$

where e_{tu} is an alignment model that scores the inputs around position u and output at time step t. The alignment model is parameterized a as a feedforward neural network.

3.5. ICDAR Text Localization Evaluation Method

The method from ICDAR (International Conference on Document Analysis and Recognition) is usually selected as a standard text localization evaluator.

Given such a ground truth set of target text bounding boxes T and the bounding boxes returned by the tested system, which is called estimate, E. The number of correct estimates is called c. Precision and recall can be defined as the following equations.

$$precision\ (p) \quad = \quad \frac{c}{|E|}$$

Precision, p is defined as the number of correct estimates divided by the total number of estimates.

$$recall(r) \quad = \quad \frac{c}{|T|}$$

Recall, r is defined as the number of correct estimates divided by the total number of targets (ground truth).

As the output bounding boxes from the proposed system may not perfectly match the ground truth bounding boxes as illustrated in Figure 20.

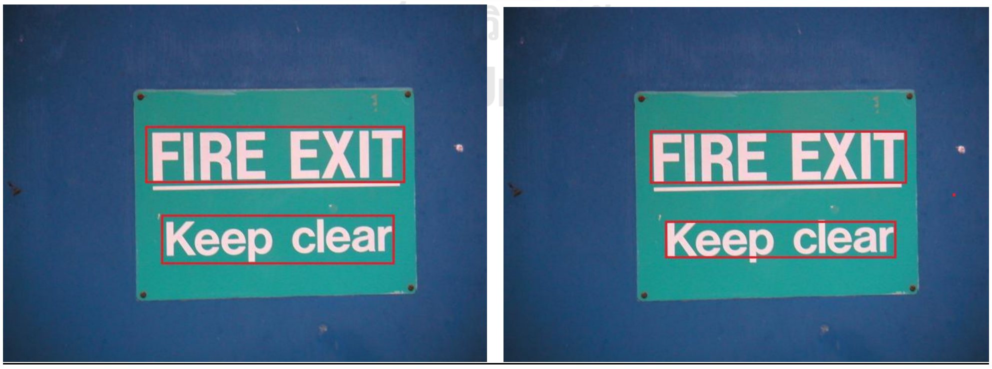


Figure 20 An example of not precise match between ground truth (left) and system output bounding boxes (right)

To correct this problem, ICDAR defines a flexible notion of a match m_p between two rectangles as intersection area divided by the area of a minimal rectangle which containing both rectangles as shown in Figure 21.

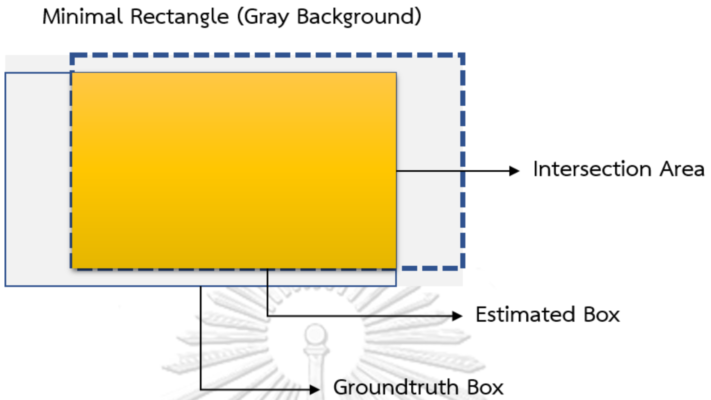


Figure 21 ICDAR definition for text localization evaluation.

Hence, the best match $m(r,R)$ for the rectangle r in a set of rectangles R is defined as.

$$m(r,R) = max\, m_p(r,r')|\, r' \in R$$

Refer to the above notation, precision, recall, and f-measure can be redefined as following equations.

$$precision\ (p') = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|}$$

$$recall\ (r') = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|}$$

$$f - measure\ (f) = \frac{1}{\frac{0.5}{p'} - \frac{0.5}{r'}}$$

4. Proposed Methods

4.1. System Overview

In this work, the problem is divided into two main sub-problems, scene text localization and text recognition. The overall pipeline is shown in Figure 22.

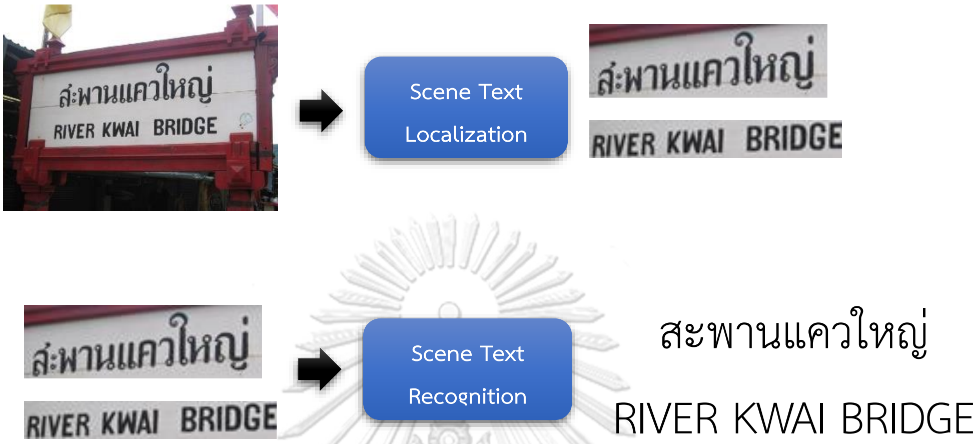
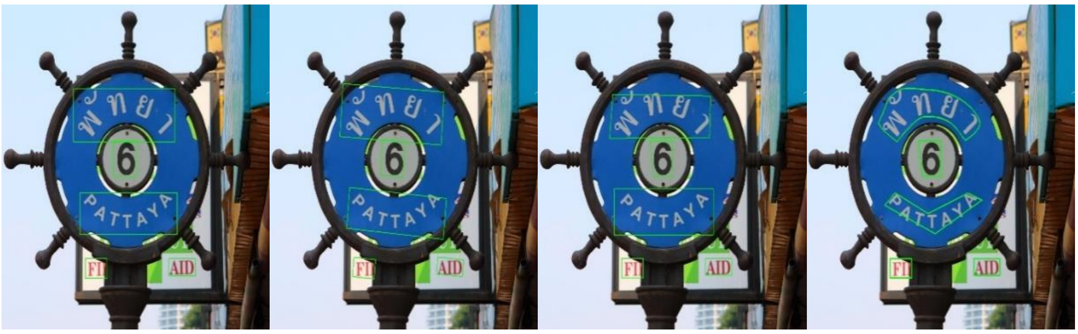


Figure 22 Designed pipeline for Thai scene text localization and text recognition.

4.2. Scene Text Localization

Since text in natural scene images is different from scanned documents in many aspects such as styles, sizes, and colors. Instead of using traditional handcrafted features, this problem is formulated into a semantic segmentation problem. Convolutional Neural Network (CNN) has demonstrated strong capabilities of learning great representations from the image in many computer vision problems such as object detection, instance segmentation, and semantic segmentation problems.

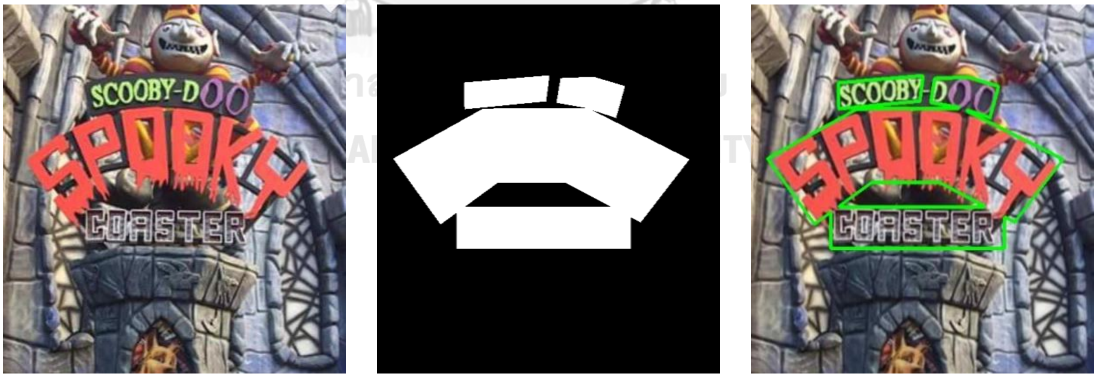
Most existing scene texts localization methods are typically based on state-of-the-art object detection algorithms, represent text instances in the form of 2-D rectangles containing text. Unfortunately, some existing methods failed to locate text accuracy in complex cases, such as arbitrarily shaped and curved texts, which is difficult to represent with a single rectangle or quadrangle used in generic object detectors, as shown in Figure 23.



(a) Axis-aligned boxes (b) Rotated rectangles (c) Quadrangles (d) Polygons

Figure 23 Text instance representation.
Text polygon representation can accurately capture the location, scale, and bending of the curved text, while the others cannot provide accurate text instance locations.

To tackle this problem, the text localization is treated as a semantic segmentation problem. However, only semantic segmentation might not distinguish very close text instances, resulting in a single merged text instance as shown in Figure 24. In order to deal with this problem, in addition to representing the text instances using only text pixels, the proposed method also learns the text border pixels and offset masks, which can greatly help to separate the nearby text instances.



(a) Input Image (b) Text segmentation map (c) Inference text instances from segmentation map

Figure 24 Merged inferenced text instances due to the connected segmentation map.

4.2.1. System Overview

In this section, the system overview of proposed text localization will be elaborated.

The proposed method can be separated into three parts. In the pre-processing part, the input image longer side is resized to the defined size. Then, the resized image to fed into the text localization network. The network outputs consist of three components, text masks, border masks, and offset masks. These outputs are combined and used to retrieve the text instances in the post-processing, which are polygon scoring, restoration, and suppression. The text localization algorithm diagram is shown in Figure 25.

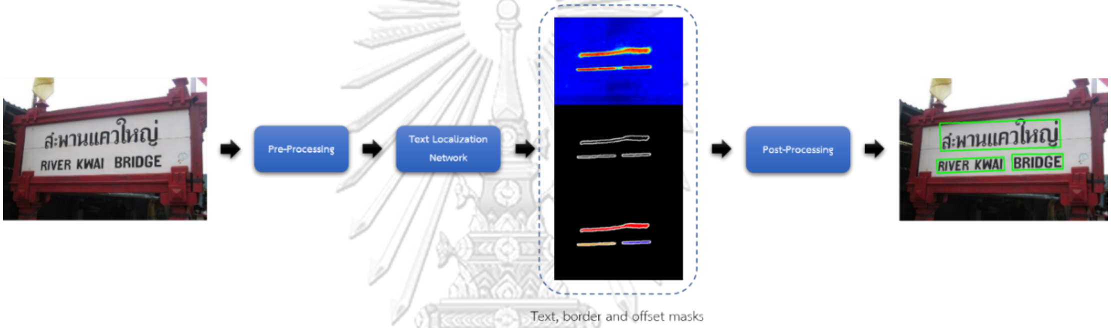


Figure 25 Text localization overall diagram.

4.2.2. Text Instance Representation

In many previous works, scene texts are typically represented by axis-aligned bounding boxes, which are aligned with the axes of the coordinate system. Some works use quadrangles to make the bounding box fit more precisely to the text regions. Nevertheless, in some challenging cases, quadrangles are still not capable of covering the entire text instances.

Some existing methods used text pixel masks to represent arbitrarily shaped text instances. Although only text pixel masks might not be able to distinguish the very close text instances. Hence, instead of just using text masks, the shrunk text masks and offset masks are used to represent text instances. Offset masks are offsetting polygons that can be either inward or outward. Furthermore, to make the network capable of capturing different text sizes, each original text instance polygon is offset into multiple scales based on its area and perimeter. If the text instance t_i is considered with polygon scaling factor α, the polygon offsetting ratio d_i can be defined from the following equation.

$$d_i = \frac{\alpha * \ Area(t_i)}{Perimeter(t_i)}$$

For each image, the ground truth can be defined as text masks g_{tm}, which are filled offsetting polygons; offset masks g_{om}, where each polygon area is filled with the d_i; and outer border masks g_{bm}, representing each text instance border. The proposed text representation is visualized in Figure 26.

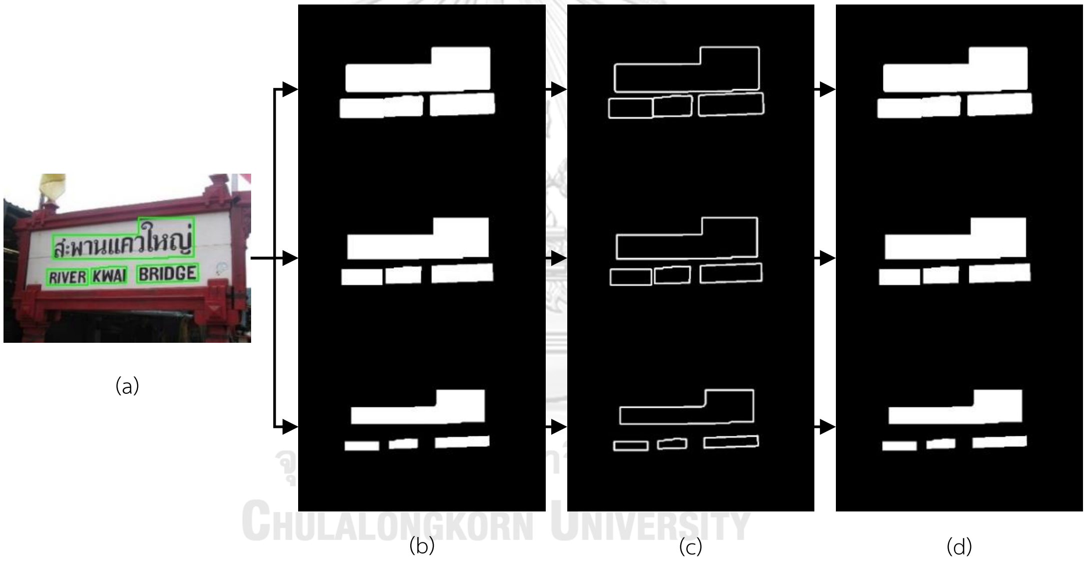


Figure 26 Proposed multi-scaled text instance representation.
(a) Original text polygon t_i (b) text masks (c) border masks and (d) offset masks.

4.2.3. Network Structure

A fully convolutional neural network (J. Long et al., 2015) combined with an encoder-decoder style network based on residual neural network (K. He et al., 2016) is used as a core network, joining with feature pyramid network (FPN) (Lin, Dollár, et al., 2017) to avoid spatial information loss in encoder-decoder blocks. FPN has shown a strong capability to handle multi-scaled semantic information in recent many semantic segmentation works. The lateral connections between deep and shallow feature maps can help generated the fine-grained feature maps from low and high semantic features. The feature maps can be downsampling and upsampling in the encoder-decoder style network in two major ways, fractional stride convolution and interpolation. A fractional stride convolution operator usually causes the checkerboard pattern, resulting in the inaccurate output text masks; thus, standard bilinear interpolation is used for upsampling feature maps to the desired size.

The text localization network's output composes three branches, text masks, offset masks, and border masks, respectively. The text localization network structure is schematically shown in Figure 27.

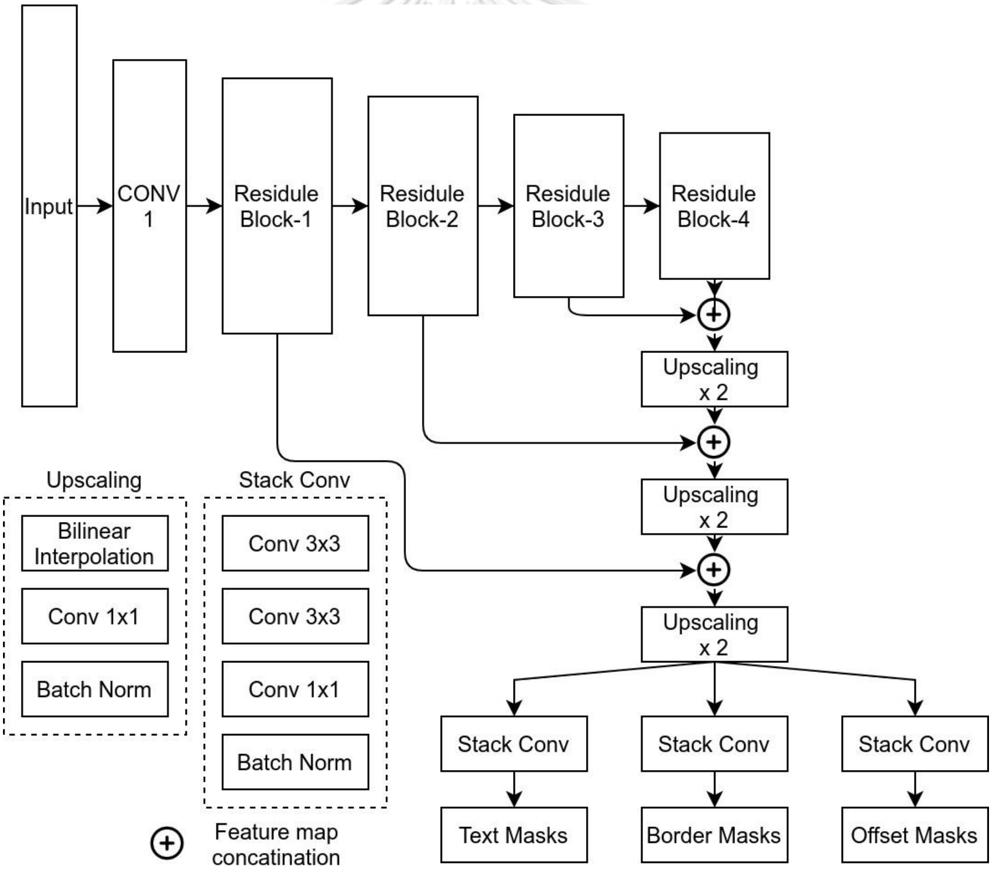


Figure 27 Text localization network structure.

4.2.4. Loss Function

As mentioned in the above section, the network outputs consist of three components: text, offset, and border masks. For the text and border masks, since the ratio between text, non-text pixels, and especially border pixels in scene text images are significantly imbalanced, making the network tends to put more emphasis on non-text pixels, resulting in false detections on both text and non-text pixels when using the following standard binary cross-entropy loss.

$$L_{CE} = -\frac{1}{N}\sum_{j} y_j * log\ (\widehat{y_j})$$

where N represents the number of pixels in image. y_j and $\widehat{y_j}$ represent the actual and predicted class, respectively.

The main aim of the text pixel labeling problem is to maximize the overlapping regions between ground truth and the predicted segmentation mask. There are many region-based losses that can be applied with this problem, such as weighted cross-entropy, tversky loss (Abraham & Khan, 2019), sensitivity-specificity loss (Hashemi et al., 2019), and focal loss (Lin, Goyal, et al., 2017). Nevertheless, to maximize classification accuracy on these losses, the parameters need to be tuned. To address this problem, the non-parametric dice-loss (Milletari et al., 2016) is used for both the text masks L_{tm} and border masks L_{bm}. The multi-task loss can be shown as follows.

$$L_{tm}(o_{tm}, g_{tm}) = \frac{2 * \sum o_{tm}(x,y) * g_{tm}(x,y)}{\sum o_{tm}(x,y)\ \sum g_{tm}(x,y)}$$

$$L_{bm}(o_{bm}, g_{bm}) = \frac{2 * \sum o_{bm}(x,y) * g_{bm}(x,y)}{\sum o_{bm}(x,y) \sum g_{bm}(x,y)}$$

Smooth L_1 loss is used to ensure good training loss stability on offset masks. The loss function for the offset mask L_{om} can be formulated as follows.

$$L_{om}(o_{om}, g_{om}) = \sum SmoothL_1\big(o_{om}(x,y) - g_{om}(x,y)\big)$$

$$SmoothL_1(x) \ = \begin{cases} 0.5x^2 \\ |x| \ - \ 0.5 \end{cases}$$

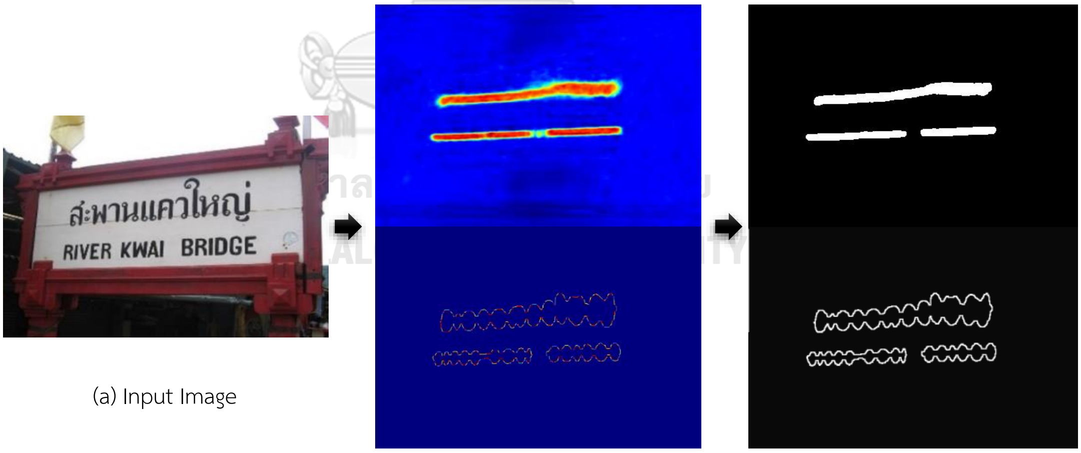
All losses above can be combined into multi-task loss L which can be defined as

$$L = \lambda_1 L_{tm} + \lambda_2 L_{bm} + \lambda_3 L_{om}$$

where λ_1, λ_2, and λ_3 are the weights balance between text, border, and offset masks, respectively.

4.2.5. Text Instance Inference and Border Augmentation

The network forward pass outputs are multi-scaled text masks, border, and offset masks. In order to infer text instances from these outputs, simple thresholding is employed on both text and border masks. The input image, text masks, border masks, and their thresholding results are shown in Figure 28.



(a) Input Image

(b) Raw text and border

(c) Thresholded text and

Figure 28 Input image and its single scale raw and thresholded text and border masks.

In this work, the border augmentation is introduced consolidated with connected component analysis (CCA) to detect and separate adjacent components.

Border augmentation is a fast and straightforward operation between corresponding scaled text masks o_{tm} and border masks o_{bm} can be defined as:

$$o_{ba}(x,y) = \begin{cases} 1 & if\ o_{tm} = 1\ and\ o_{bm} = 0 \\ 0 & otherwise, \end{cases}$$

where o_{ba} represents the output text border augmented masks.

Each text instance score can be computed from the polygon instances scoring algorithm, which can be formulated as follows.

$$P(t_i) = \frac{1}{N} \sum_{(x,y) \in t_i} t_m(x,y)$$

where P and N represent text instance score and the number of pixels in text instance t_i, respectively.

The original text instances can be altered to their original shapes by using the offset value $V(t_i)$, which can be calculated from output offset masks o_{om}, as follows:

$$V(t_i) = \underset{(x,y) \in t_i}{median}\{o_{om}(x,y)\}$$

Given such text polygon candidates with their associated scores from all scales, polygon non-maximum suppression (PNMS) is performed to eliminate the overlapping detections and obtain the final text instances.

4.3. Thai Scene Text Recognition

As described in the Scene Text Localization section, text in natural scene images differs from scanned documents due to many challenges, such as styles, sizes, and colors. Many of the existing Thai character recognition works still rely on the process of segmenting the cropped words or sentences into a single character. Unfortunately, in the particular situation of scene text problem, it is difficult to obtain a single cropped character due to the complex backgrounds, various lighting conditions, and text styles, as shown in Figure 29.

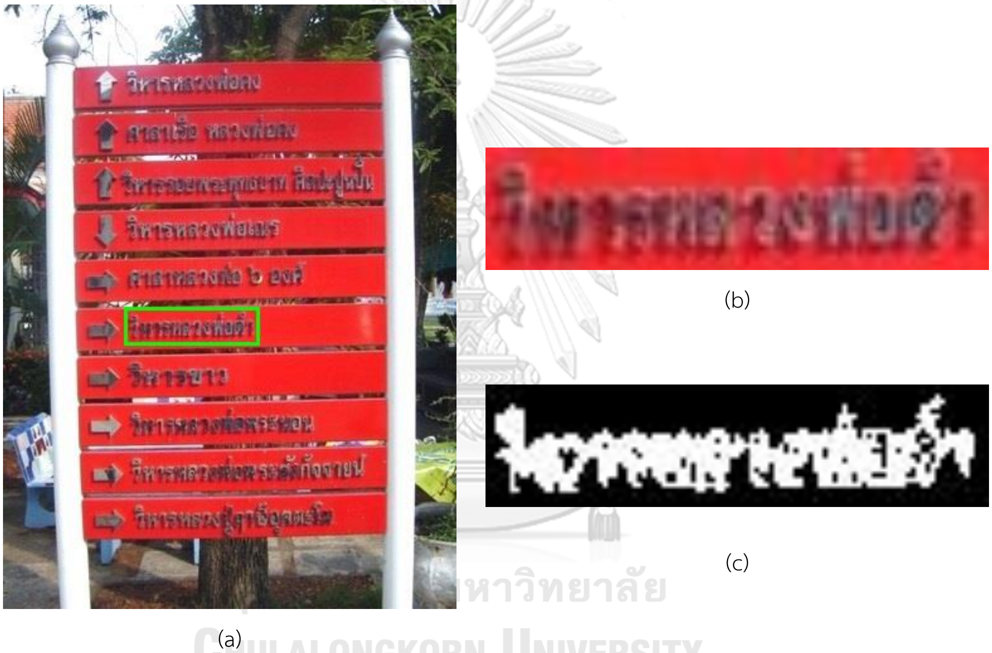


Figure 29 (a) Sample of detected text instance. (b) Cropped detected text instance. (c) Simple thresholded result. As show in the figure, simple thresholding cannot distinguish cropped word into single individual character.

Recent great success on English and Multi-Language (MLT) scene text recognition has been dominated by the CNN-based method. Many scene text recognition sub-problems such as curved text recognition and multi-languages recognition in a single model, have also enjoyed numerous successes. However, to the best of our knowledge, there is no practical Thai scene text algorithm.

4.3.1. System Overview

In this section, the system overview of proposed Thai scene text recognition will be presented.

According to (J. Baek et al., 2019), the standard scene text recognition pipeline can be separated into four stages:

1. Transformation: transforms cropped word images into more suitable shapes for recognition.

2. Feature extractor: extracts visual features from transformed images.

3. Sequence modeling: provides the contextual information from characters sequence, rather than doing it independently.

4. Prediction: determines the output sequence from extracted image features.

The overall components of the proposed text recognition method are shown in Figure 30.

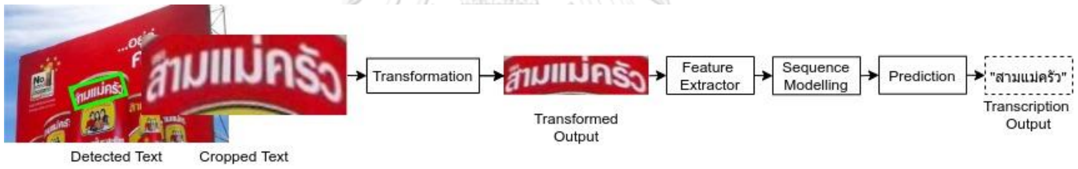


Figure 30 Overall pipeline of proposed Thai scene text recognition method.

4.3.2. Transformation

Scene text images usually appear in irregular shapes, as shown by curved and perspective texts, due to many factors such as camera angles and text orientations. The existing normalization techniques are based on morphological analysis or image registration techniques, which may not be able to generalize on scene text images.

In this work, Thin Plate Spline Transform (TPS), which is a variation of Spatial Transform Network (STN) used in (Shi et al., 2019), is applied to transform irregular and regular input text images into appropriate shapes for recognition. TPS does learn not only the affine transformation matrix but also the projective mapping from the input image. This characteristic makes TPS can warp images in arbitrary ways.

Given an input image X and number of fiducial points F, an aligned image X_A can be determined as the TPS output. The input and output image from this stage are shown in Figure 31.

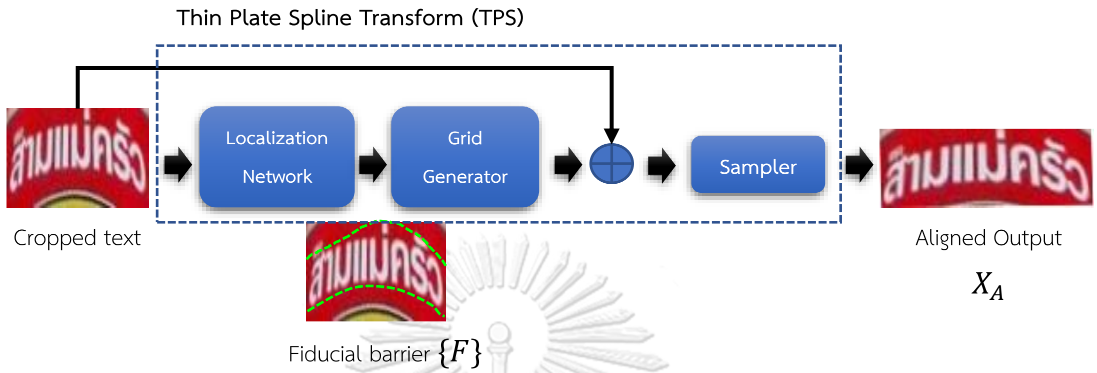


Figure 31 Cropped text input and aligned output from thin plate spline transform. The set of fiducial points $\{F\}$ are visualized as green dashed line.

4.3.3. Feature Extraction

In this part, a cropped text image X_A size of $H \times W \times C$ will be fed into the feature extractor network, yielding a visual feature V size of $H_v \times W_v \times C_v$. Each column W_{vi} represents a corresponding distinguishable text location along the horizontal axis. The graphical representation is shown in Figure 32, where J_v represents the association between aligned input and output visual features.

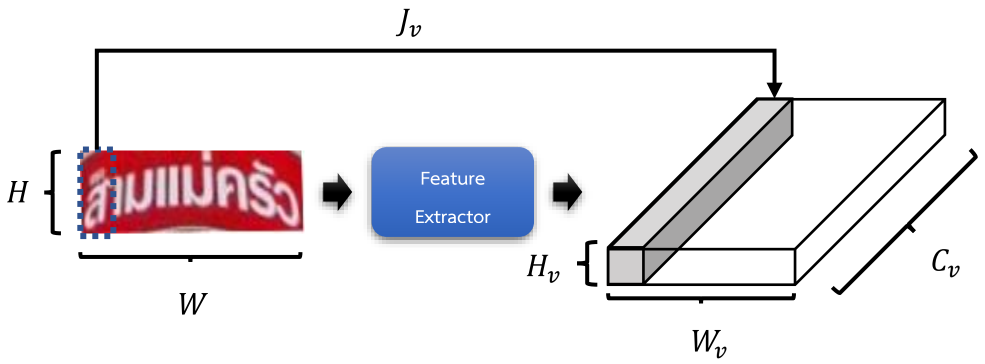


Figure 32 Graphical representation between aligned text image and output feature.

In this work, the residual neural network (ResNet) incorporate with feature pyramid network (FPN) is used as a main feature extractor for scene text recognition. Since Thai language consists of many small elements (vowels and tone marks), many existing works on small and fine-grained image classification had proved that using the feature from both shallow and deep can significantly boost the model accuracy. Hence, the feature map aggregation network, which is illustrated in Figure 33, is designed.

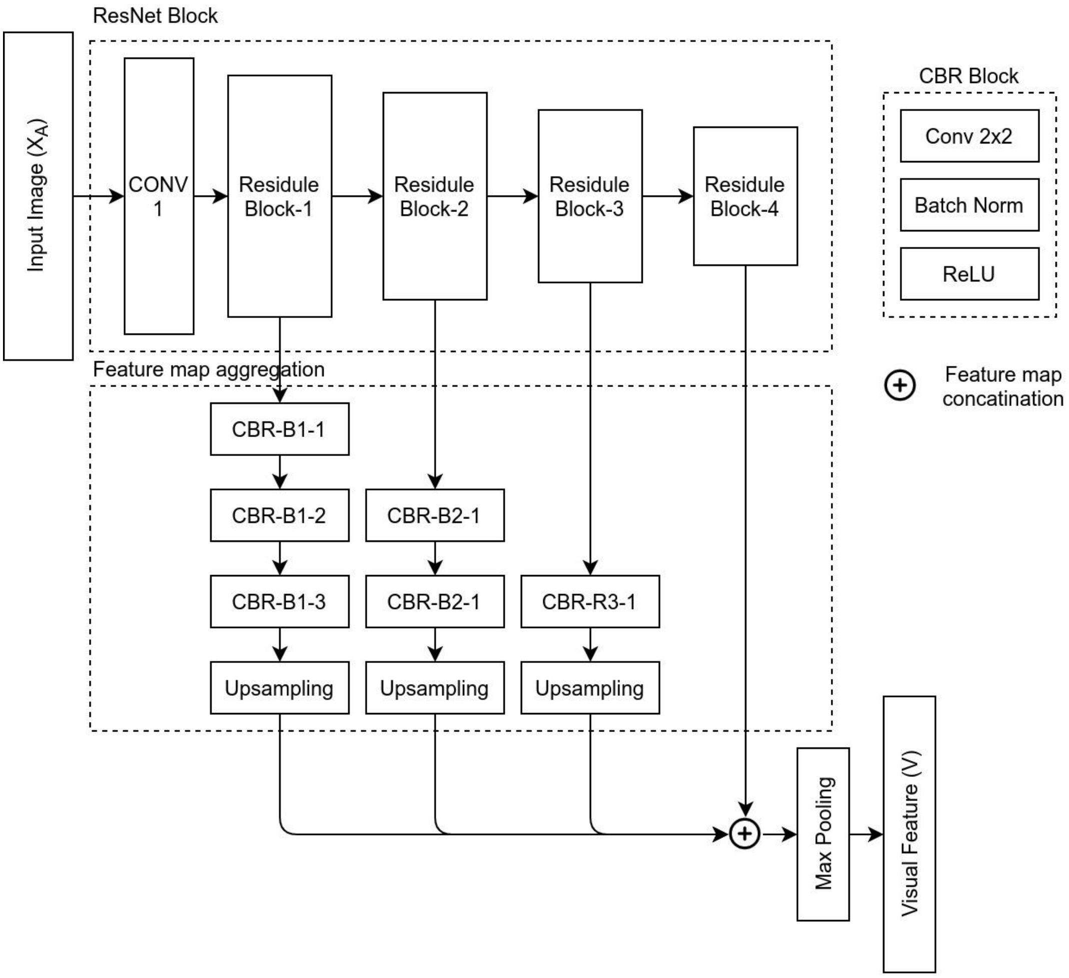


Figure 33 Scene text recognition feature extractor network structure.

To prove the proposed method effectiveness, a lightweight version of feature extractor is also proposed. This lightweight network is designed based on MobileNetV2 (Sandler et al., 2018). The proposed structure is shown in Figure 34, where DWConv refers to depthwise separable convolution.

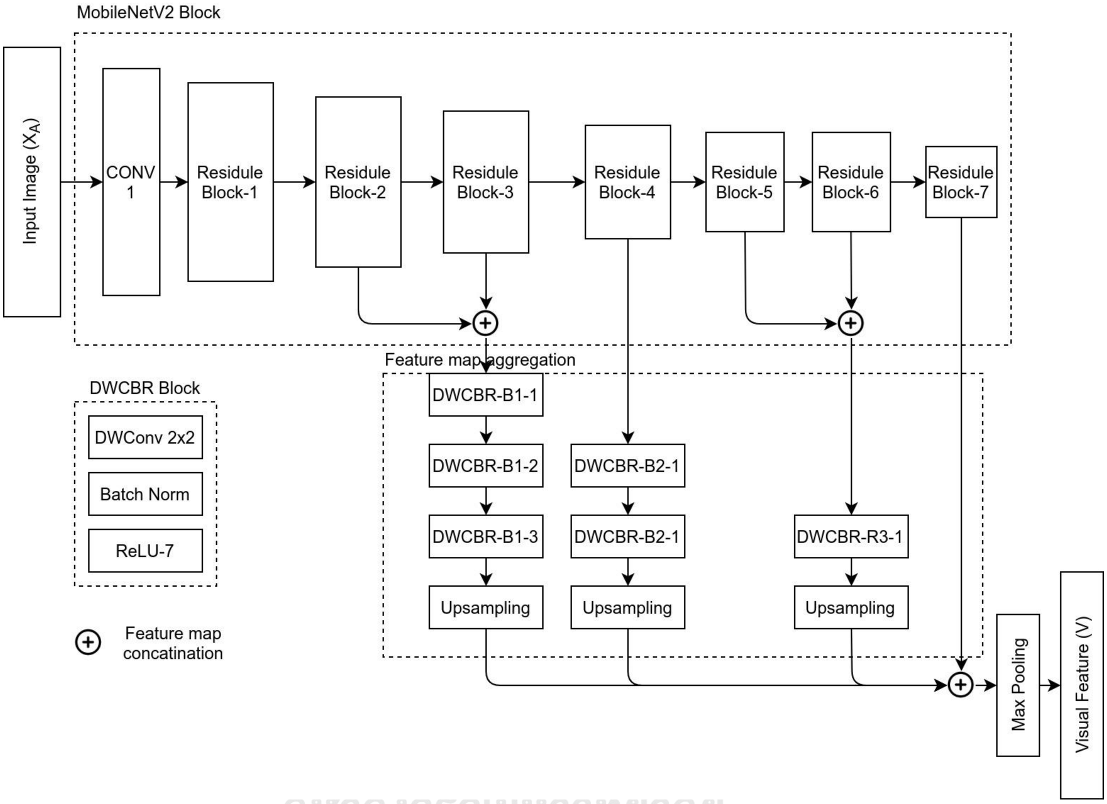


Figure 34 A lightweight version of proposed feature extractor network structure.

4.3.4. Sequence Modeling

The interdependencies between each location on visual feature maps include meaningful information that would help identify or guess some ambitious characters. However, the visual feature V from the feature extraction stage still lacks contextual information.

Many previous works on contextual sequence learning problem relay on Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), and Gated Recurrent Unit (GRU) (K. Cho et al., 2014). Since the vanishing gradient problem tends to occur in RNN and theoretically, LSTM can remember longer sequences than GRU and outperform in tasks that require long-distance relations. In this work, because the characters in a cropped word are aligned and represented by visual feature sequence, which can be either arranged from left to right or right to left, LSTM is adopted to learn this sequence information. Each column in visual feature V is fed into two stack LSTM layers with 512 hidden units to produce visual sequence feature V_s , shown in Figure 35.

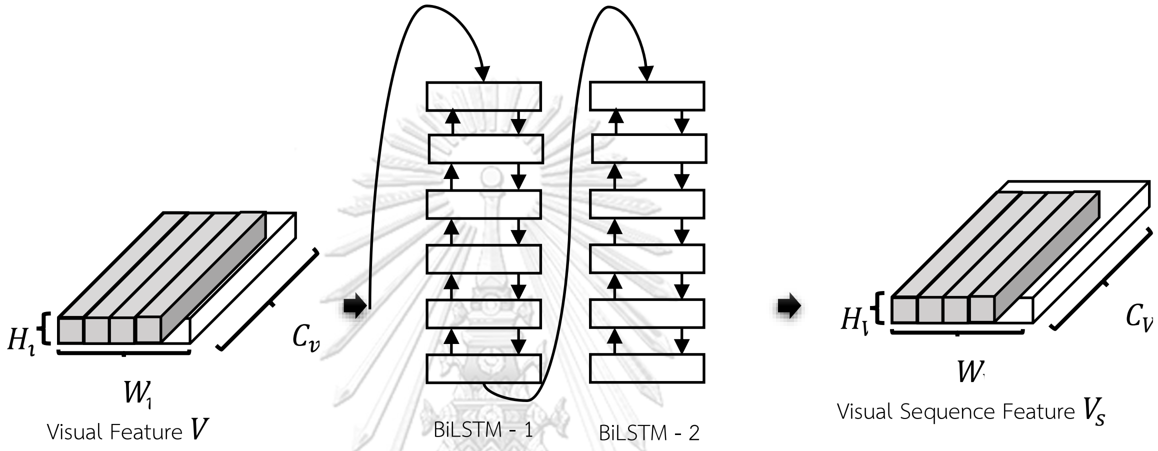


Figure 35 Sequence modelling diagram.

where H_{V_s} and W_{V_S} equal to H_V, and W_V and C_{V_S} equal to 1024, respectively.

4.3.5. Prediction

Sequence-to-sequence learning (seq2seq) is a new paradigm to convert sequences from one domain sequences in another domain. This model has been widely used in many tasks and achieves great successes, such as machine translation, speech recognition, molecular synthesis, and text recognition. In this stage, the 2D attention-based encoder-decoder network is adopted to create the seq2seq model between sequence features and output character sequence.

Given the visual sequence feature V_s, this stage translates the feature sequence into a character sequence O. In this work, the attention mechanism implemented by (Shi et al., 2019) is used. This implementation is a unidirectional recurrent network that works iteratively for W_{V_S} steps, yielding an output sequence of length T, where

$T \leq W_{V_S}$. At each time step t, the output from this layer can be either a character token or end-of-sequence token, based on the input visual sequence feature V_s, attention internal state and previous token at time step $t - 1$. The number of hidden units in the attention layer is set to 1,024.

4.3.5.1. String Encoding Scheme

Sequence-to-sequence learning requires the input and output sequences to be encoded in categorical class labels. Unlike English text, which all alphabets and vowels are written on a single line, in Thai, one line can be divided into four levels, as shown in Figure 36. Alphabets and some vowels are written in the main level while the others might be written above or below. These unique characteristics create many possible ways to encode the input sequences. In this work, four possible encoding schemes are studied.

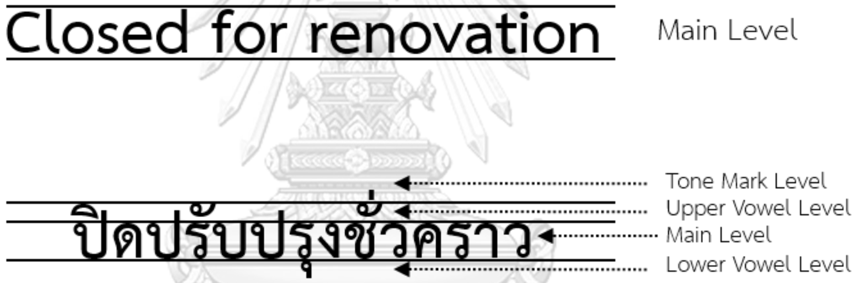


Figure 36 Writing style comparison between English and Thai.

4.3.5.1.1. Single Level – Single Character Encoding

In this encoding scheme, all characters, including vowels and tone marks, are treated as individual characters. Given an input string S “ปิดปรับปรุงชั่วคราว”, a character C at location i (C_i) is mapped to its corresponded class number, resulting in an encoded array D. This encoding procedure can be visualized in Figure 37.

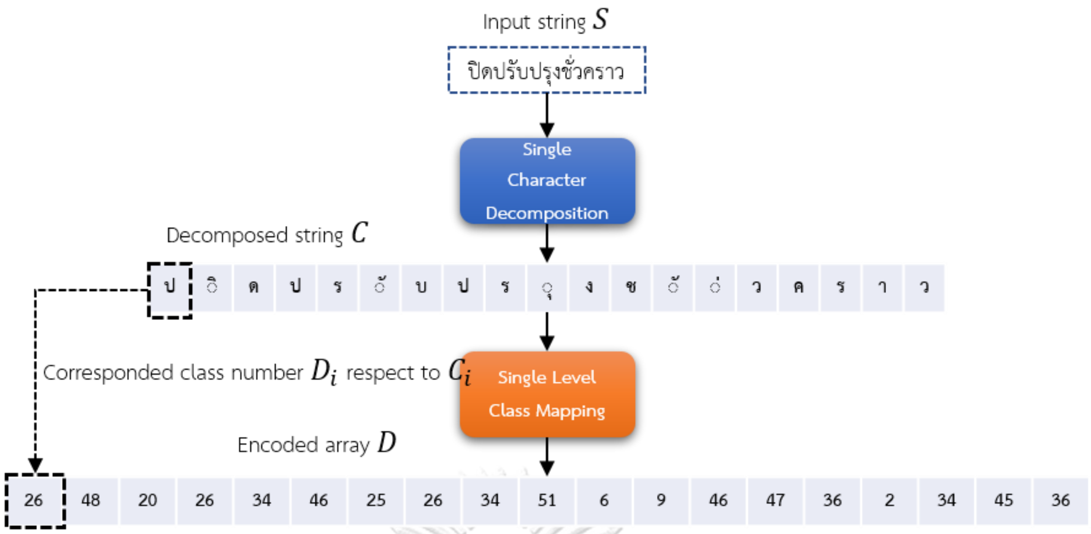

Input string S
ปิดปรับปรุงชั่วคราว
Single Character Decomposition
Decomposed string C
ป ิ ด ป ร ั บ ป ร ุ ง ช ั ่ ว ค ร า ว
Corresponded class number D_i respect to C_i
Single Level Class Mapping
Encoded array D
26 48 20 26 34 46 25 26 34 51 6 9 46 47 36 2 34 45 36

Figure 37 Single level single character encoding scheme.

4.3.5.1.2. Single Level – Group Character Encoding

In this encoding scheme, a main line character with different tone marks and vowels is treated as an individual character, for example, ซ, ซั and, ซั่ are mapped to separate classes. Given the input string S “ปิดปรับปรุงชั่วคราว”, group of characters C at the same location i (C_i) are combined to one class and mapped to its corresponded class number, resulting in an encoded array D. This encoding scheme is shown in Figure 38.

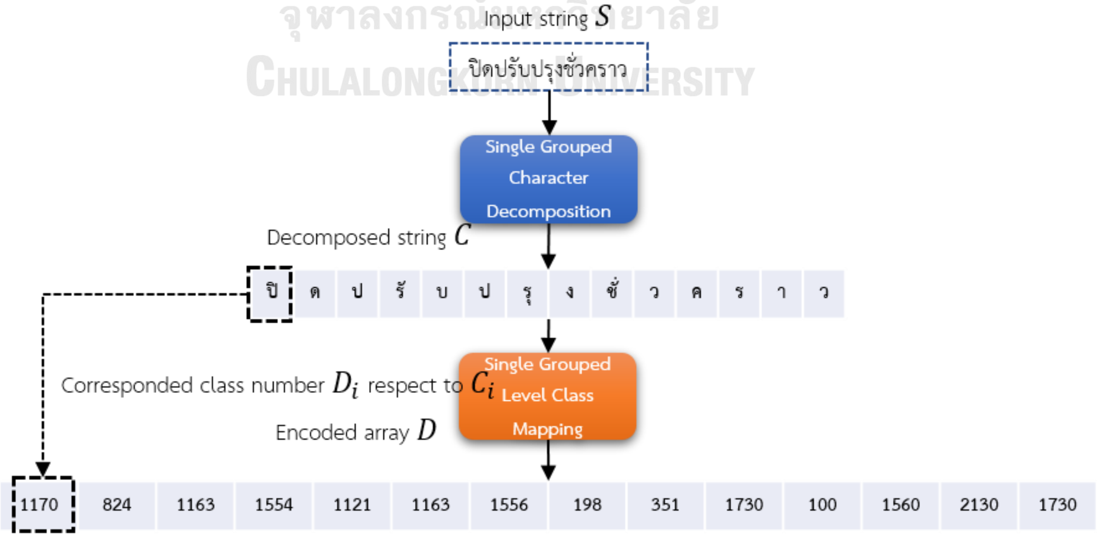

Input string S
ปิดปรับปรุงชั่วคราว
Single Grouped Character Decomposition
Decomposed string C
ปิ ด ป รั บ ป รุ ง ชั่ ว ค ร า ว
Corresponded class number D_i respect to C_i
Single Grouped Level Class Mapping
Encoded array D
1170 824 1163 1554 1121 1163 1556 198 351 1730 100 1560 2130 1730

Figure 38 Single line grouped character encoding scheme.

4.3.5.1.3. Three-Level Encoding

This encoding scheme will handle the input string like single level encoding, but each character is categorized into three classes. Given the input string S “ปิดปรับปรุงชั่วคราว”, a character C at location i (C_i) , each character is categorized into upper vowel and tone mark U, main line character M, and lower vowel L. The output encoded arrays, upper vowels, and tone marks D^U, main line characters D^M, and lower vowels D^L, are built based on character classes. This encoding scheme is shown in Figure 39, where “-“ indicates a blank.

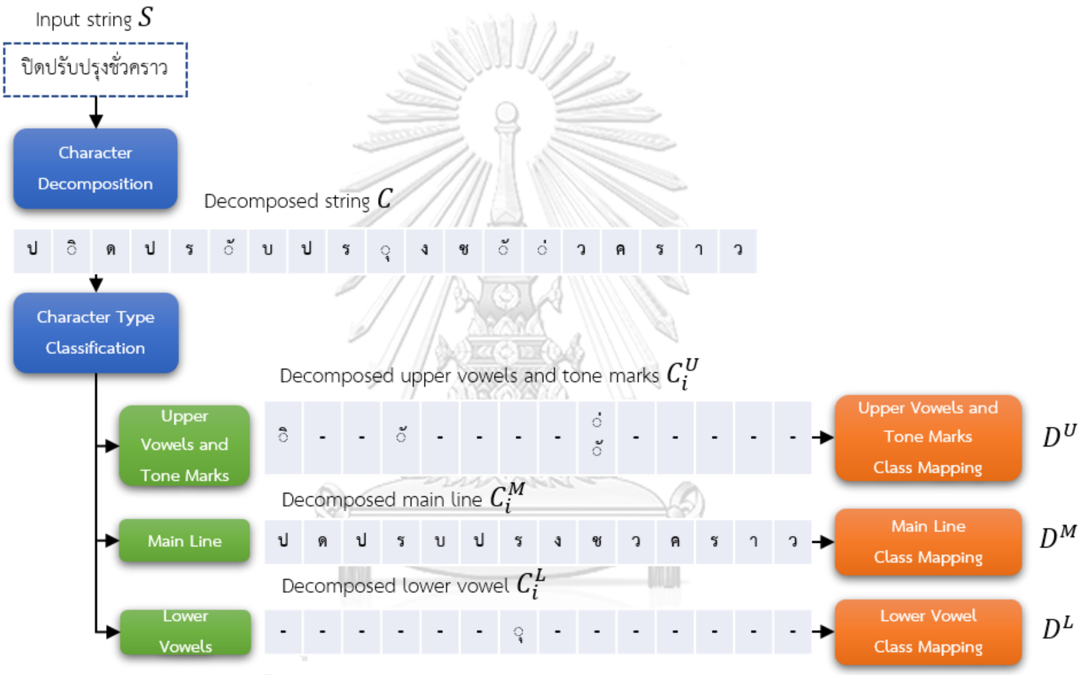

Input string S
ปิดปรับปรุงชั่วคราว
Character Decomposition
Decomposed string C
ป ิ ด ป ร ั บ ป ร ุ ง ช ั ่ ว ค ร า ว
Character Type Classification
Decomposed upper vowels and tone marks C_i^U
Upper Vowels and Tone Marks
ิ - - ั - - - - ่ั - - - - -
Upper Vowels and Tone Marks Class Mapping
D^U
Decomposed main line C_i^M
Main Line
ป ด ป ร บ ป ร ง ช ว ค ร า ว
Main Line Class Mapping
D^M
Decomposed lower vowel C_i^L
Lower Vowels
- - - - - - ุ - - - - - - -
Lower Vowel Class Mapping
D^L

Figure 39 Three-level encoding scheme. “-“ indicate a blank (no character).

4.3.5.1.4. Four-Level Encoding

This encoding scheme behaves like three-level encoding, but each character is categorized into four classes Given the input string S “ปิดปรับปรุงชั่วคราว” and a character C at location i (C_i) , each character is categorized into tone mark T, upper vowel U, main line character M, and lower vowel L. The output encoded arrays D, tone marks D^T, upper vowels, and D^U, main line characters D^M, and lower vowels D^L, are built based on character classes. This encoding scheme is shown in Figure 40, where “-“ indicates a blank.

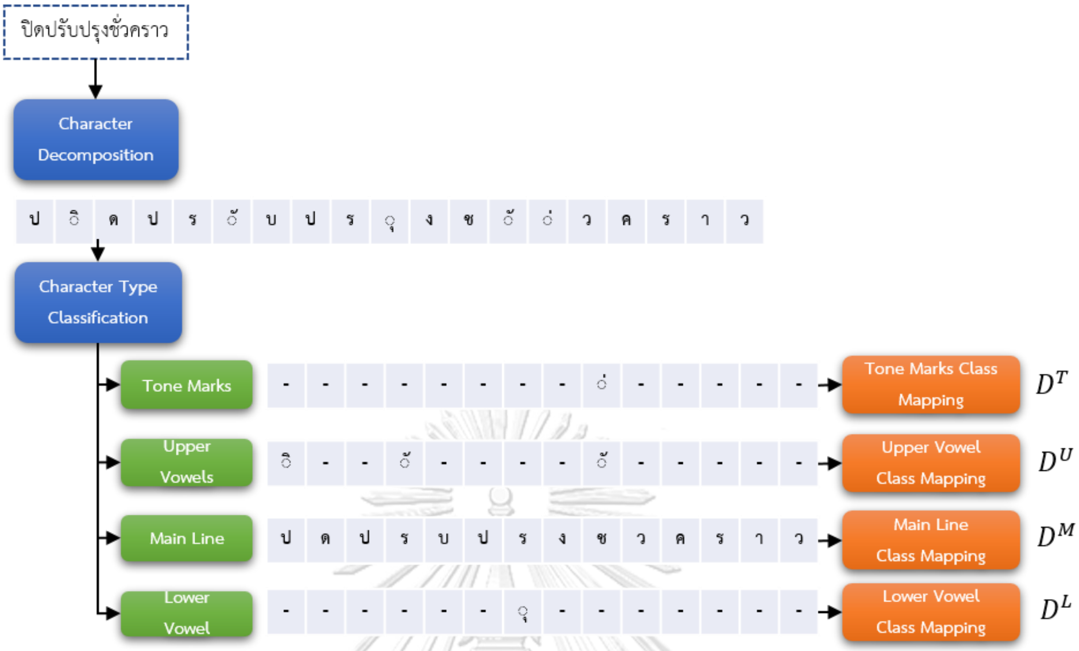


Figure 40 Four-level encoding scheme. “-“ indicate a blank (no character).

4.3.5.2. Multi-Level Attention Mechanism

As mentioned in the previous section, Thai text writing system can be divided into multiple levels. In this work, instead of using only one attention layer to capture the entire multiple levels character sequence, the multi-level attention mechanism is proposed. As proposed in section 4.3.5.1, Thai text can be encoded into various variations. The multi-level attention is designed to capture each writing line character sequence by the sharing input visual sequence feature, the. Given the encoded sequence D, where $D = \{D^T, D^U, D^M, D^L\}$ depends on the encoding method and n equal to $|D|$, the multi-level attention can be shown in Figure 41.

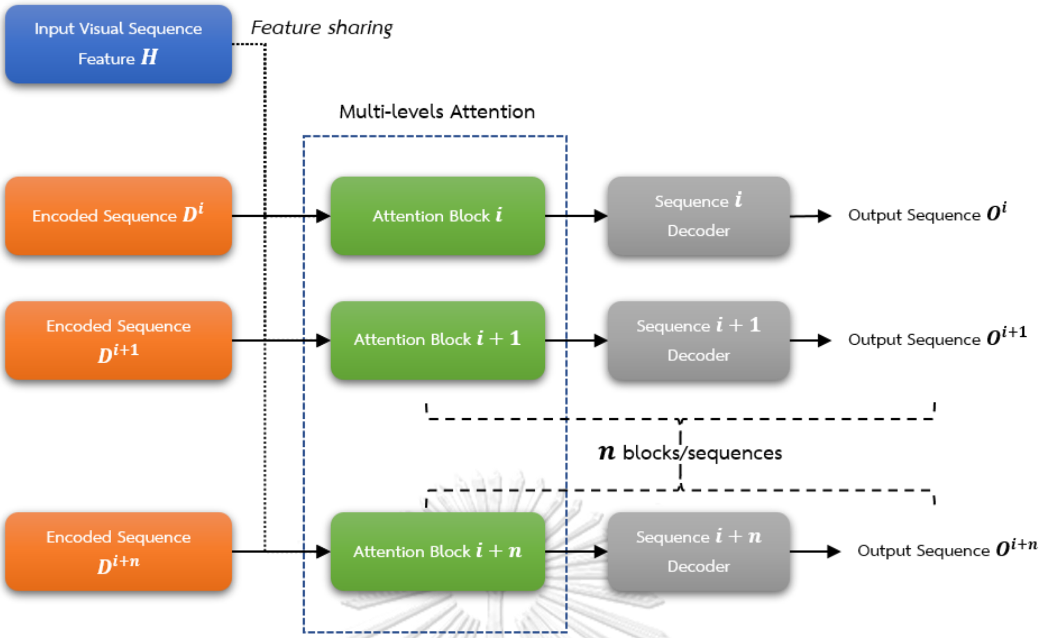


Figure 41 Multi-level attention diagram.

4.3.6. Loss Function

Generally, a single-level attention mechanism utilized standard cross-entropy which can be shown as follows:

$$L_{Attn}^{i} = -\frac{1}{N}\sum_{j} o_j^i * log\,(\widehat{o_j^i})$$

where N represents the time step number, i represents the i^{th} sequence in multi-level attention mechanism, o and $\hat{o}$ represent the one hot encoding vector of actual and predicted main line characters in the input image, respectively.

The overall multi-level attention loss L_{Attn} can be defined as follows:

$$L_{Attn} = \sum_{i}^{n} \alpha^i L_{Attn}^{i}$$

where n represents the number of multi-level sequences, i represents the i^{th} sequence in multi-level attention mechanism, and α represents the weight balance for each sequence, respectively. In this work, all sequence weight balance parameters α are set to 1.

As the attention mechanism usually focuses on a chuck of each time step feature, which may be comparable to each character, bigram, or trigram level in an English word. On the other hand, Thai is different from English in terms of writing style. All words in a single sentence are consecutively written without space. If the global context can be captured from sequence features, it may improve overall recognition accuracy. In this work, the global character context loss L_{GC} is introduced. The visual sequence feature V_s is connected to the adaptive max-pooling follows by the fully connected layer as shown in Figure 42.

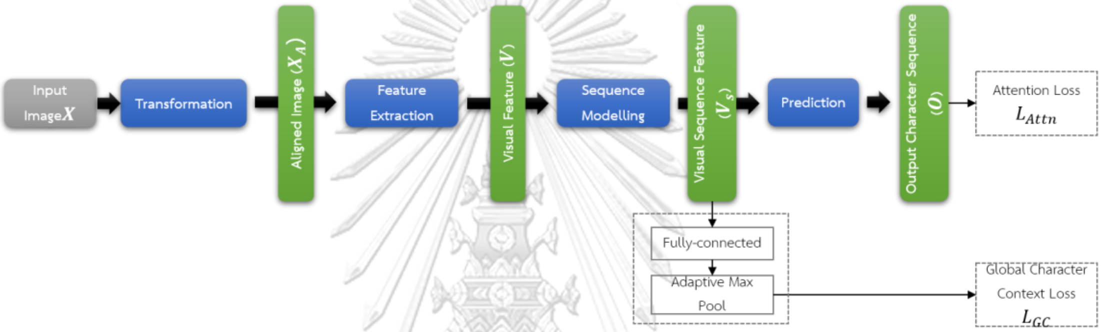


Figure 42 Multi-task loss for the proposed scene text recognition method.

The output from the fully connected layer is then optimized by binary cross entropy loss, which is defined as global context loss, can be shown as follows:

$$L_{GC} = -\frac{1}{N}\sum_{j} y_j log\ (\hat{y}_j) + (1 - y_j) log\ (\hat{y}_j)$$

where N represents the number of main line classes, y and $\hat{y}$ represent the one hot encoding vector of actual and predicted main line characters in the input image, respectively.

All losses above are combined into multi-task recognition loss L_R for scene text recognition problem, which can be defined as:

$$L_R = \lambda_1 L_{Attn} + \lambda_2 L_{GC}$$

where λ_1 and λ_2 are the weights balance between multi-level attention and global context losses, respectively.

4.3.7. Training Data Acquisition

In order to obtain an accurate and precise Thai text recognition model, a very significant amount of labeled training data is required. To the best of our knowledge, there are no standard Thai scene text datasets. Thus, a fast, adaptive, close to real-world, and multi-language supported synthetic scene text generation engine needs to be created.

Generally, the synthetic scene text image generation starts by collecting text and a background image from multiple sources. The Thai text corpus is gathered from the following sources:

- Thai people names and surnames from Thailand Open Government data (*Open Government Data of Thailand*, 2020).
- Provincial administration information includes provinces, districts, sub-districts, roads, alley, and villages names from Thailand Open Government data.
- Government office, hospital, school, stadium, police station, post office names from Thailand Open Government data.
- Thai articles from Wikipedia.
- TNC Top-5000 Words.
- BEST (NECTEC) Thai entity recognition corpus (ระบบคลังสื่อประสมและข้อความกำกับ, 2020).
- NECTEC question and answering corpus from Thai Wikipedia.

From the above sources, 687,895 non-overlapping words and sentences were collected and used as Thai text corpus. Samples of text corpus are shown in Table 9.

Table 9 Thai text corpus samples

น้ำแข็งใสจั๊มบ๊ะ	นวนผ่อง	บุตรสร้อย	ศาสนิกชน	มีนมรกต
สามสุวรรณ	มวลมงคลมณี	งามเลื่อน	บุญฤกษ์	ไชยสองแก้ว
ชุมชนหนองแวง	ชลาลัยศิริกุล	วรุณกาญจน์	วงศ์ธนบูรณ์	เตียวล่ำซำ
ชีพบริสุทธิกุล	เมตร	คลอสตริเดียม	ทารา	ยกเถ
คออินทร์	การอุปถัมภ์	กลุ่มงานสอบสวน	ชู้ต	ทันตกิจกุล
จรรยาวรรธ์	มักกะ	นิมลมูล	นานาชาติเวลล์	เชยโมภักดี
จุ้มใจ๋	ที่โล่งแจ้ง	อีศพงศ์	ลิลิตวรรณ	947
โรจนกิตติ	แสงสุริศรี	กัลย์จรัศม์	ปองเกรียงไกร	พัญญู

Text font styles are also a crucial factor that affects the recognition accuracy of the designed model. In this work, 3,005 Thai fonts are collected from the internet.

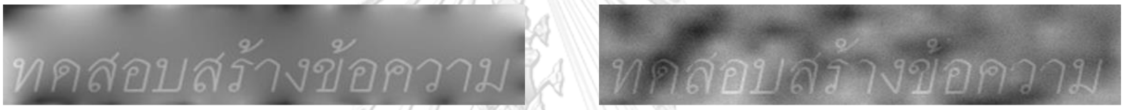
Photo sharing community and search engines such as Flickr and Google Image Search contain a wide range of scene images. To generate realistic text images, 8,500 background images are gathered through multiple queries related to different scenes, objects, locations, and natural/artificial territories. However, there gathered images must not contain any text of their own. Thus, a baseline text detection algorithm based on the EAST (X. Zhou et al., 2017) is used in combination with a manual human inspection to reject those images.

Scene text images tend to be appeared in continuous regions, for instance, sign, billboard, and continuous smooth color regions. To gather the regions respect to those constraints, a strong edge detection Holistically-Nested Edge Detection (HED) (S. Xie & Tu, 2015) in combination with gPb-UCM (Arbeláez et al., 2011) contour hierarchies are used.

To find an appropriate text color for extracted background regions, the color palette for text and background is learned from cropped word images in the IIIT5K word dataset (Anand Mishra et al., 2009). In each cropped word, pixels are divided into two sets using K-mean clustering algorithm. The color pair result from clustering is the approximated colors of text (foreground) and background. In some cases, however, the approximated text color is still not suitable for the chosen background.

Therefore, Web Content Accessibility Guidelines (WCAG) (Initiative (WAI), 2020) contrast ratio, which is a standard measure to provide enough contrast between text and background so that it can be readable by most people, is also used.

Poisson image editing (Pérez et al., 2003) is utilized to realistically blend a synthetic text with a background image to simulate the text – background image natural composition effects. The main highlight of this work is that working with input image gradients instead of image intensities will produce close to real-world output results. An OpenCV seamless cloning implementation of Poisson image editing in mixed clone mode is used. The comparison between typical text-background pasting and seamless clone results are shown in Figure 43.



(a) Simple text paint to background image

(b) Text painting with poisson image editing

Figure 43 Comparison between (a) simple text painting and (b) Poisson image editing result. Poisson image editing makes the generated text blended with the background looks more natural, and close to real-world scene text.

Not only the image blending is applied to simulate the scene text real-world situations, but also standard image augmentation. The following techniques are employed to make the synthetic text look sensible.

- Standard, gaussian, motion, and median blur.
- Additive, multiplicative, and gaussian noise.
- Uncropped image rotating.
- Grid-based elastic transformation.
- Perspective Transformation
- Emboss and sharpen effect.
- Randomly adjust brightness and contrast.

- Simulate JPEG and PNG compression artifact.

The sample of synthetic Thai text images are shown in Figure 44.

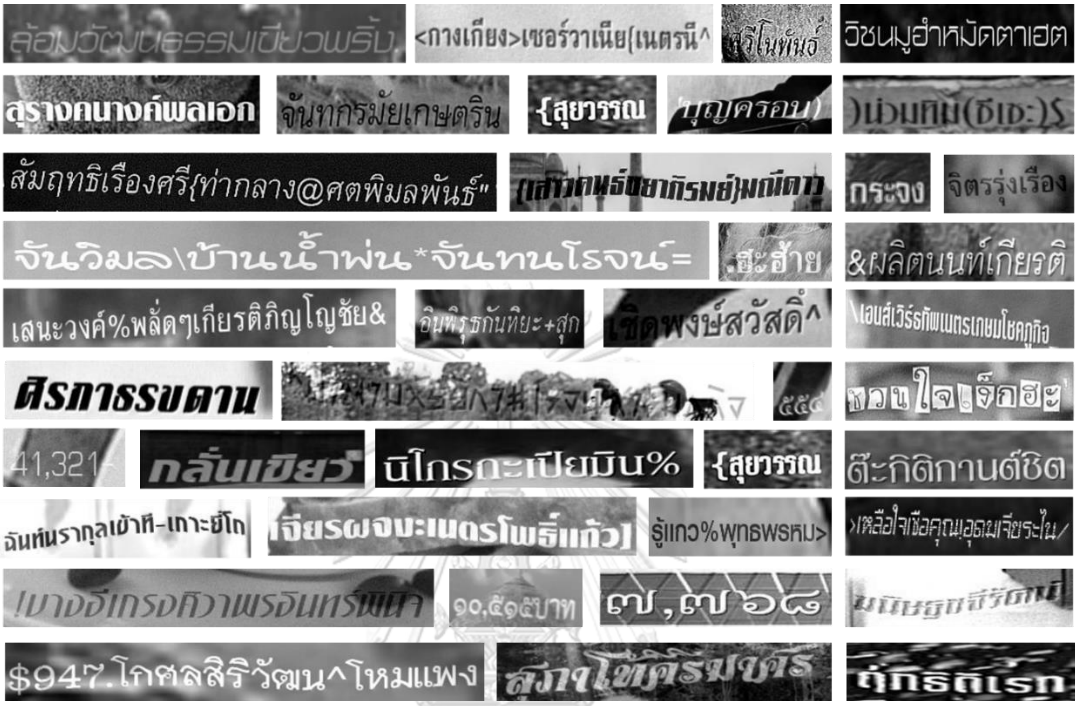


Figure 44 Samples of Thai scene text recognition training dataset.

5. Experimental Results

As the problem can be decomposed into two main sub-problems: scene text localization and text recognition, the experiments were conducted on both algorithms to show the performance in many aspects.

5.1. Scene Text Localization

To evaluate the proposed scene text localization performance, a quantitative test on standard scene text detection benchmarks and our self-collected dataset are employed and compared with the existing methods.

5.1.1. Training and Test Dataset Acquisition

For this work, the training dataset for scene text localization can be divided into two languages: English and Thai. The following synthesized and real-world scene text datasets are used as training and test datasets.

ICDAR2003 first appeared in the 2003 scene text detection robust reading competition (Lucas et al., 2003). This dataset was the first scene text localization and text recognition benchmark, containing a total of 509 images, which can be divided into 258 training and 251 testing images. The text instances from this dataset were labeled in a word-level axis-aligned bounding box. The samples from ICDAR2003 dataset are shown in Figure 45.

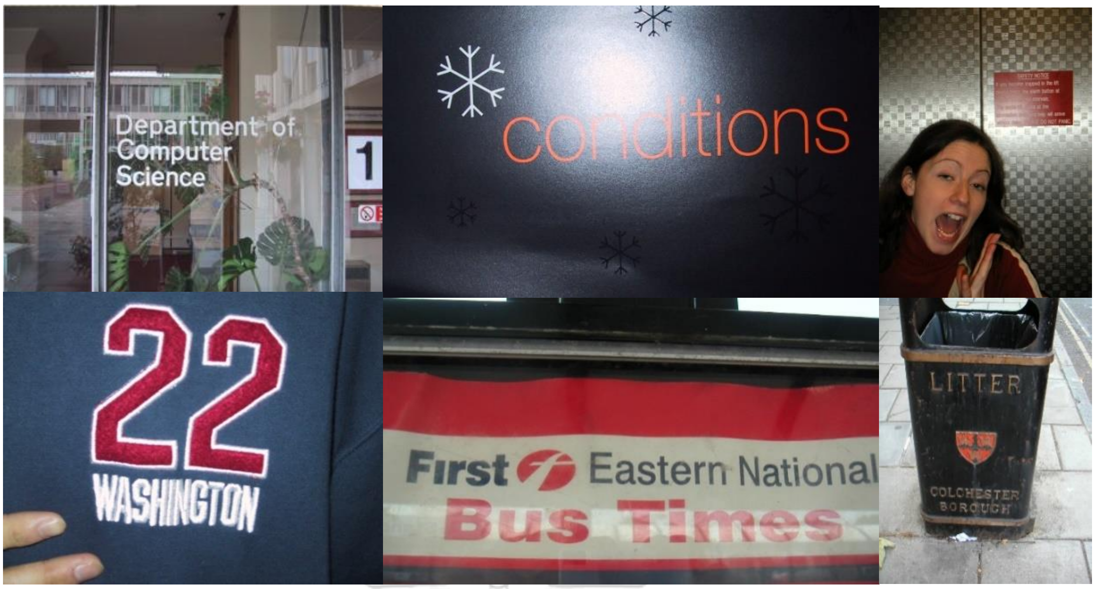


Figure 45 ICDAR2003 sample images.

ICDAR2015 appeared in the 2015 incidental scene text detection robust reading competition (D. Karatzas et al., 2015). According to the ICDAR definition, incidental scene text refers to text that appears in the scene without the user having taken any specific action to cause its appearance or improve its positioning or quality in the frame. This dataset was gathered by Google Glasses without taking image quality and viewpoint into consideration. There were 1,500 images in total, which can be separated into 1,000 training and 500 testing images. The text instances from this dataset were labeled in word-level quadrangles. The samples from ICDAR2015 dataset are shown in Figure 46.

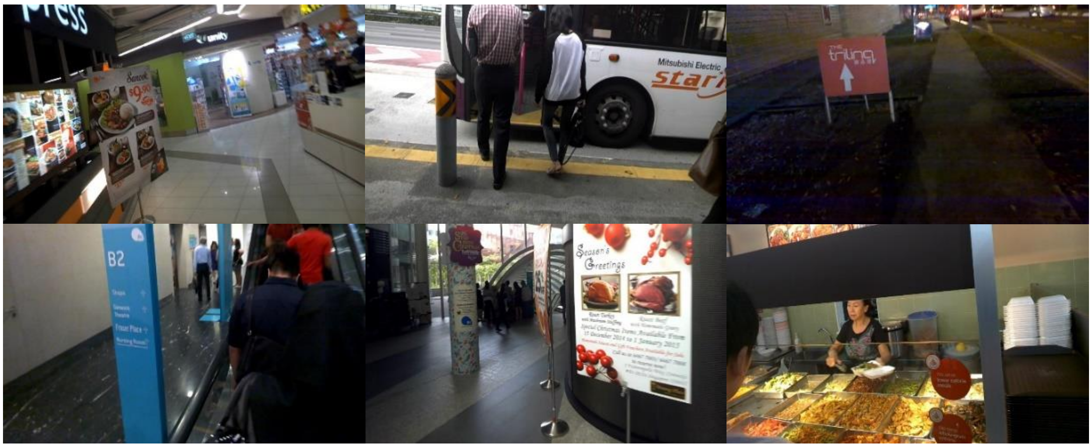


Figure 46 ICDAR2015 sample images.

In many text localization algorithms, researchers enhance the training dataset with synthetic examples in order to improve model performance. SynthText (Gupta et al., 2016) is a grand-scale, computer-generated dataset. This dataset contains approximately 800,000 images. The text images were created by blending natural background images with rendered text from various font styles. Artificial transformations, such as colors, text alignments, background blending, and orientations, have been implemented to make the text appear more natural. Text instances in this dataset were annotated in both word and character levels. The samples from SynthText dataset are shown in Figure 47.

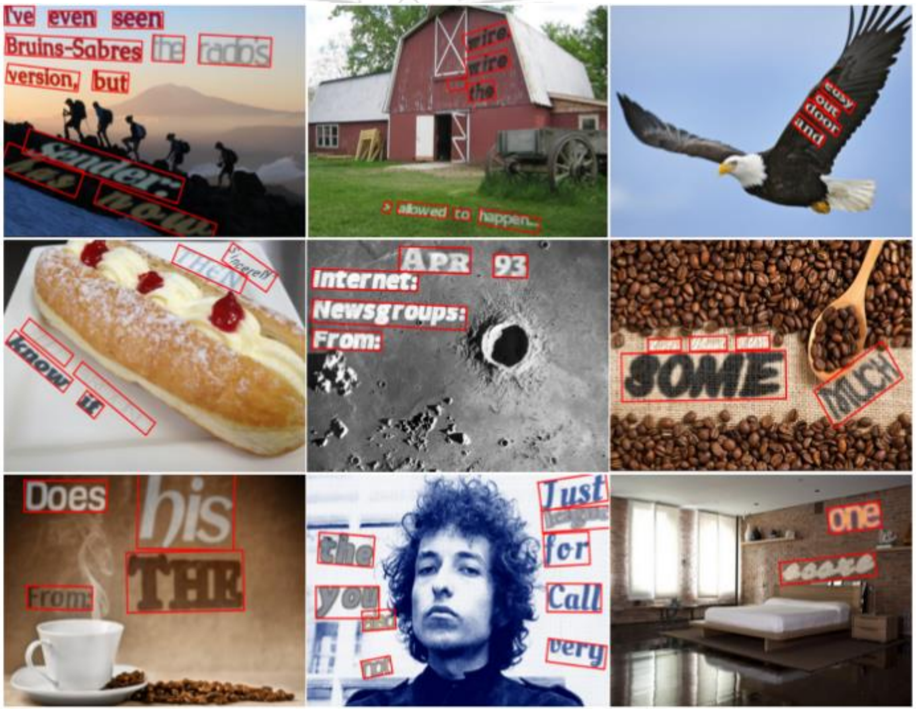


Figure 47 Samples from SynthText dataset.

ICDAR2017-MLT (N. Nayef et al., 2017) is a first competition multi-lingual scene text dataset. This dataset included 7,200 training, 1,800 validation, and 9,000 testing images, containing text from nine languages representing six different scripts. The considered languages are Chinese, Japanese, Korean, English, French, Arabic, Italian, German, and Indian. The text instances from this dataset were annotated at word level by using four vertices quadrangles. The sample from this dataset is shown in Figure 48.

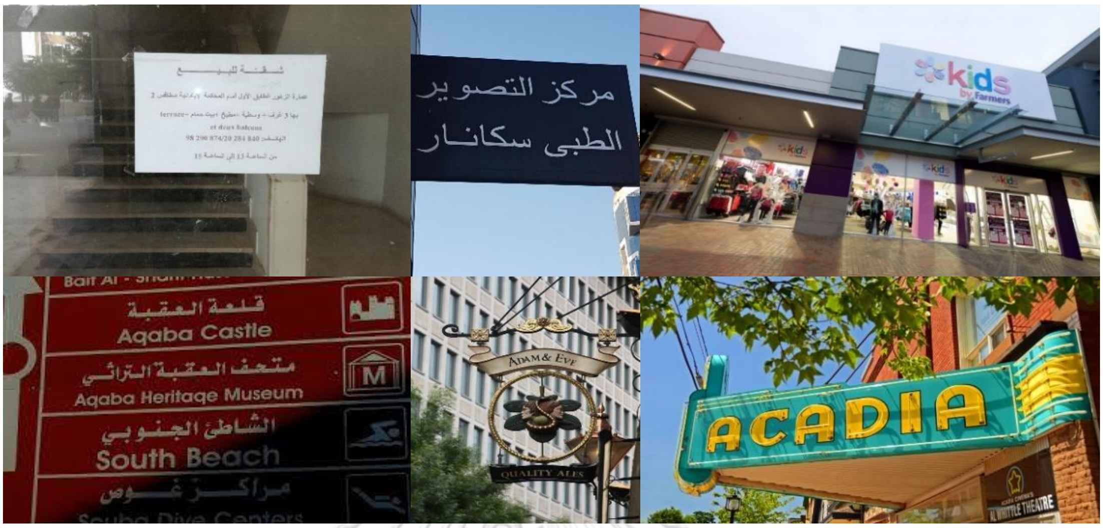


Figure 48 Samples from ICDAR2017-MLT dataset.

Total-Text dataset (Ch'ng & Chan, 2017) includes both horizontal and multi-oriented text instances. The main highlight of this dataset is curved text, which is rarely shown in other benchmark datasets. The dataset is split into training and testing sets with 1,255 and 300 images, respectively. Text instances from this dataset have been annotated in both character level pixel masks and polygons. The samples from this dataset are shown in Figure 49.

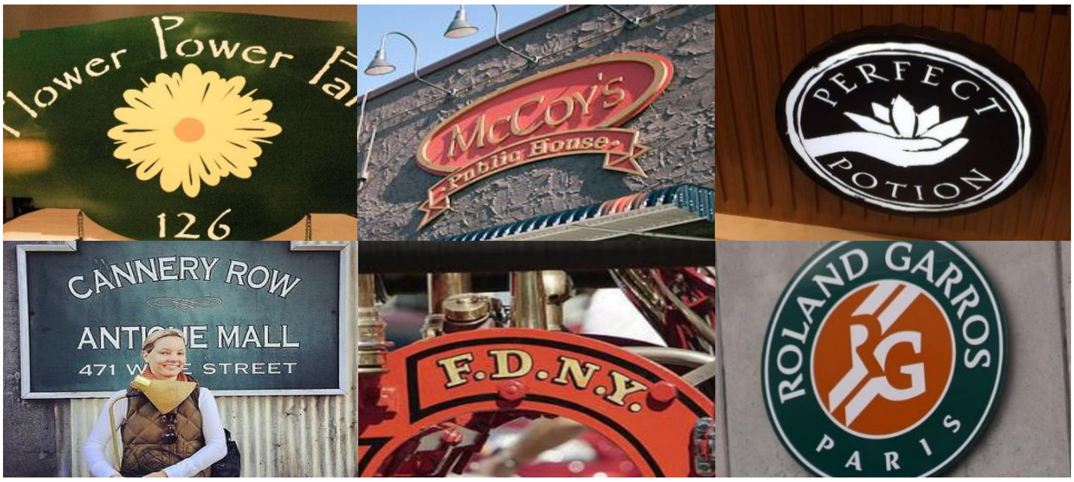


Figure 49 Samples from Total-Text dataset.

ICDAR2019-MLT (Nibal Nayef et al., 2019) is the latest multi-lingual scene text dataset. This real-world dataset consisted of 10,000 training and 10,000 testing images containing text from ten languages representing seven different scripts. The languages in this dataset are Chinese, Japanese, Korean, English, French, Arabic, Italian, German, Bangla, and Hindi (Devanagari). The text instances from this dataset were annotated at word level by using four vertices quadrangles like ICDAR2017-MLT. The samples from this dataset are shown in Figure 50.

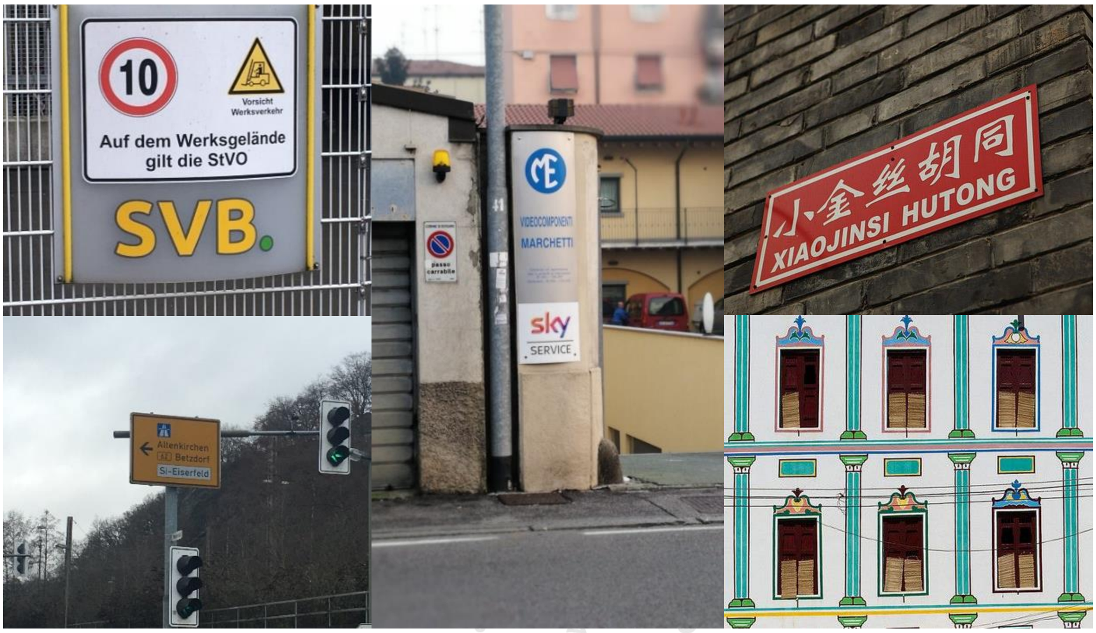


Figure 50 Samples from ICDAR2019-MLT dataset.

For Thai language, to the best of our knowledge, there is no standard Thai scene text dataset. Thus, the new dataset was collected by using smartphone cameras and crawling from multiple websites. The text images from this dataset appear in multiple conditions, such as various font styles, color, low-lighting, low resolution, and calligraphy texts, containing 3,568 images and 6,594 text instances. Text instances from this dataset have been annotated in four vertices quadrangles format. In this work, this entire dataset is used as test data. The samples from this dataset are shown in Figure 51.

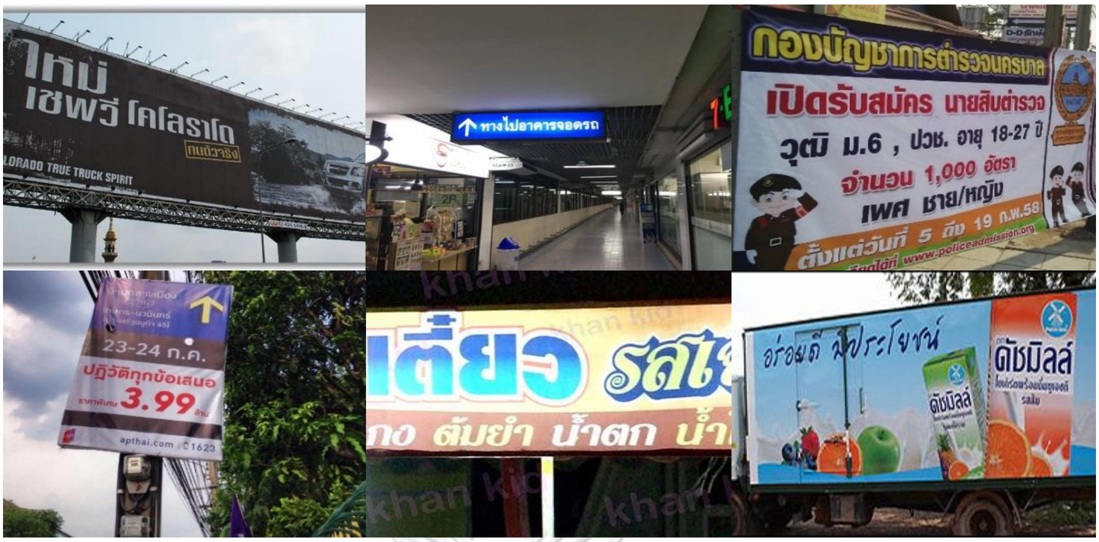


Figure 51 Samples from self-collected Thai text dataset.

In 2015, National Electronics and Computer Technology Center (NECTEC) held the Thai scene text localization competition named Benchmark for Enhancing the Standard of Thai language processing (BEST). However, this dataset contains only 100 images and 250 usable text instances, which is too small for training dataset. Thus, in this work, this dataset is only used as a test dataset. Text instances from this dataset are annotated in axis-aligned bounding boxes format. The sample from this dataset is shown in Figure 52.

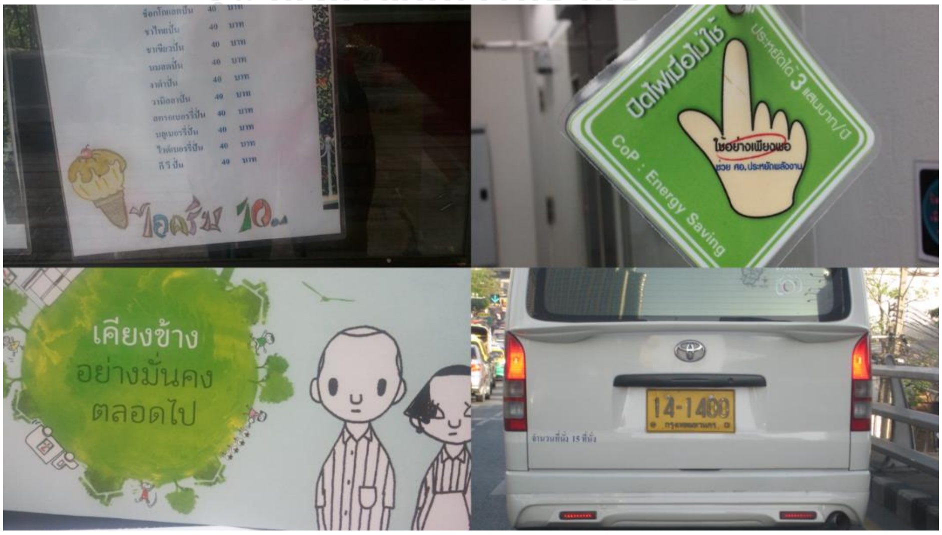


Figure 52 Samples from BEST2015 dataset.

5.1.2. Results

In this section, a series of experiments are conducted on standard scene text localization datasets from International Conference on Document Analysis and Recognition (ICDAR), Benchmark for Enhancing the Standard for Thai language Processing (BEST), and our self-collected datasets. The experiments are launched in the same environment with the source codes and results from papers. All numerical experimental results are tested by using the only single-scale test. Table 10 shows that the proposed method gives a good balance result in terms of the f-measure while still preserve acceptable precision and recall compared to the other state-of-the-art methods.

Table 10 Experimental results on standard English benchmark datasets.
P, R and F denote precision, recall, and f-measure, respectively.

Method	Datasets											
	ICDAR2015			ICDAR2017			ICDAR2019			Total-Text		
	P	R	F	P	R	F	P	R	F	P	R	F
CTPN (Tian et al., 2016)	51.6	74.2	60.9	-	-	-	-	-	-	-	-	-
EAST (X. Zhou et al., 2017)	80.5	72.8	76.4	-	-	-	-	-	-	-	-	-
SegLink (Shi, Bai, & Belongie, 2017)	73.1	76.8	75.0	-	-	-	-	-	-	-	-	-
TextBoxes++ (Minghui Liao & Bai, 2018)	87.2	76.7	81.7	-	-	-	-	-	-	-	-	-
R2CNN (Jiang et al., 2017)	85.6	79.7	82.5	-	-	-	-	-	-	-	-	-
PixelLink (Deng et al., 2018)	85.5	82.5	83.7	-	-	-	-	-	-	-	-	-
TextSnake (S. Long et al., 2018)	84.9	80.4	82.6	-	-	-	-	-	-	82.7	74.5	78.4
PSENet (W. Wang et al., 2019)	88.7	85.5	87.1	75.4	69.2	72.1	-	-	-	84.0	78.0	80.9
SPCNet (Q. Wang et al., 2018)	88.7	85.8	87.2	73.4	66.9	70.0	-	-	-	83.0	82.8	82.9
Pixel-Anchor (Li et al., 2018)	88.3	87.1	87.7	79.5	59.5	68.1	-	-	-	-	-	-

Method	Datasets											
	ICDAR2015			ICDAR2017			ICDAR2019			Total-Text		
	P	R	F	P	R	F	P	R	F	P	R	F
PMTD (J. Liu et al., 2019)	91.3	87.4	89.3	85.2	72.7	78.5	87.5	78.1	82.5	-	-	-
CRAFT (Y. Baek et al., 2019)	89.8	84.3	86.9	80.6	68.2	73.9	81.4	62.7	70.9	87.6	79.9	83.6
LOMO (C. Zhang et al., 2019)	91.2	83.5	87.2	78.8	60.6	68.5	87.7	79.8	83.6	87.6	79.3	83.3
Proposed Method (without Border Augmentation)	87.2	84.9	86.0	77.2	67.4	72.3	83.3	72.4	77.9	85.2	78.2	81.5
Proposed Method (with Border Augmentation)	89.8	86.8	88.1	78.7	69.8	73.4	86.1	75.7	80.9	88.2	79.8	83.5

5.1.2.1. Multi-Oriented English Text

In this section, an experiment on ICDAR2015 dataset is conducted and compared with existing. The pre-trained weight from SynthText is further fine-tuned on this dataset for 200 epochs. The longer size of the input image is resized to 1,280 pixels in the testing stage while still preserving the image aspect ratio and using only single-scale testing. The numerical single-scale results list in Table 10 show that the proposed method gives a competitive result in terms of the f-measure. The samples of several detection results are shown in Figure 53.

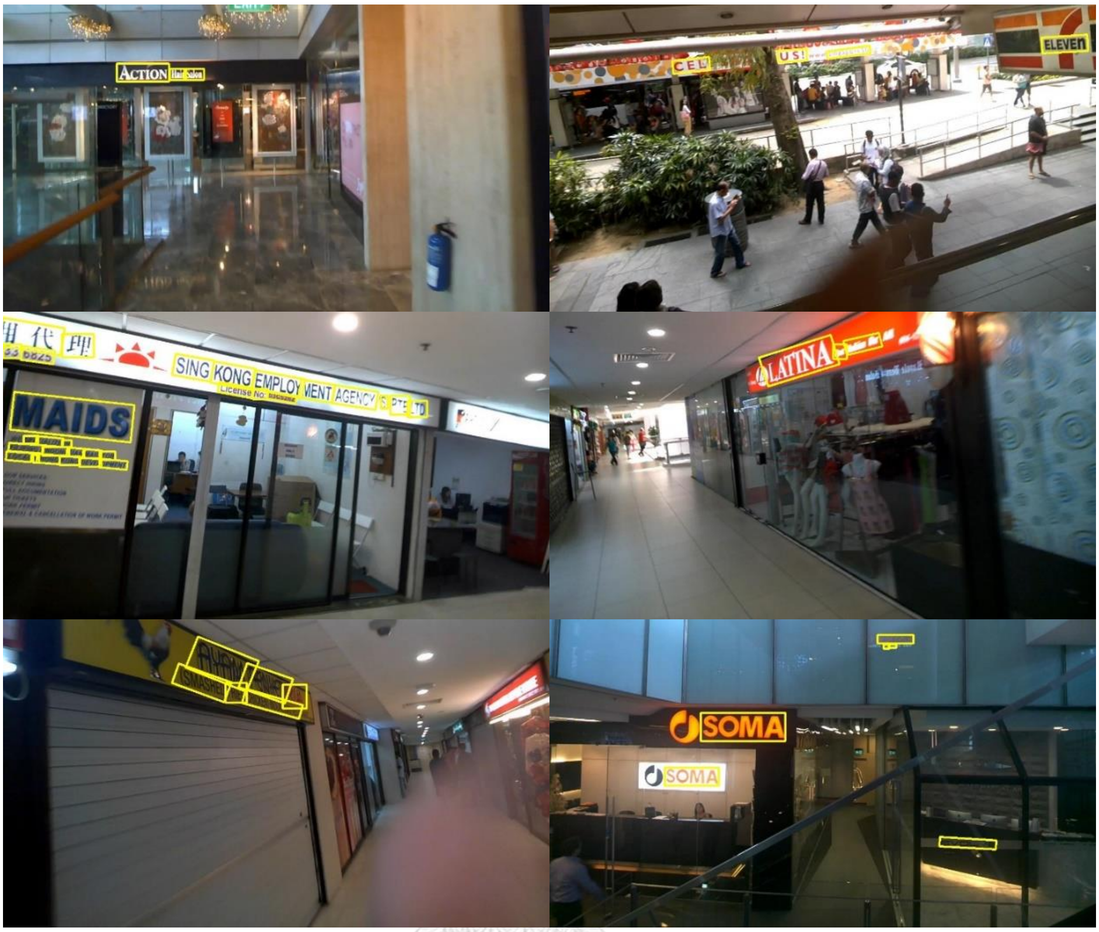


Figure 53 Example results on ICDAR2015 dataset.

5.1.2.2. Multi-Oriented and Multi-Language Text

The experiments on ICDAR2017-MLT and ICDAR2019-MLT datasets are conducted to test the robustness of the proposed text localization on multi-language scene text images. The SynthText pre-trained weight is further fine-tuned for 300 epochs on the ICDAR2017-MLT training dataset, and 450 epochs on ICDAR2019-MLT. Since the image sizes in both datasets were not equal like ICDAR2015, the input image longer size is resized to 1,280 pixels, while still maintaining the aspect ratio. The experimental results show in Table 10 prove that the proposed method gets reasonable performance compared to other state-of-the-art methods. The sample text detection results on both ICDAR2017-MLT and ICDAR2019-MLT datasets are shown in Figure 54 and Figure 55, respectively.

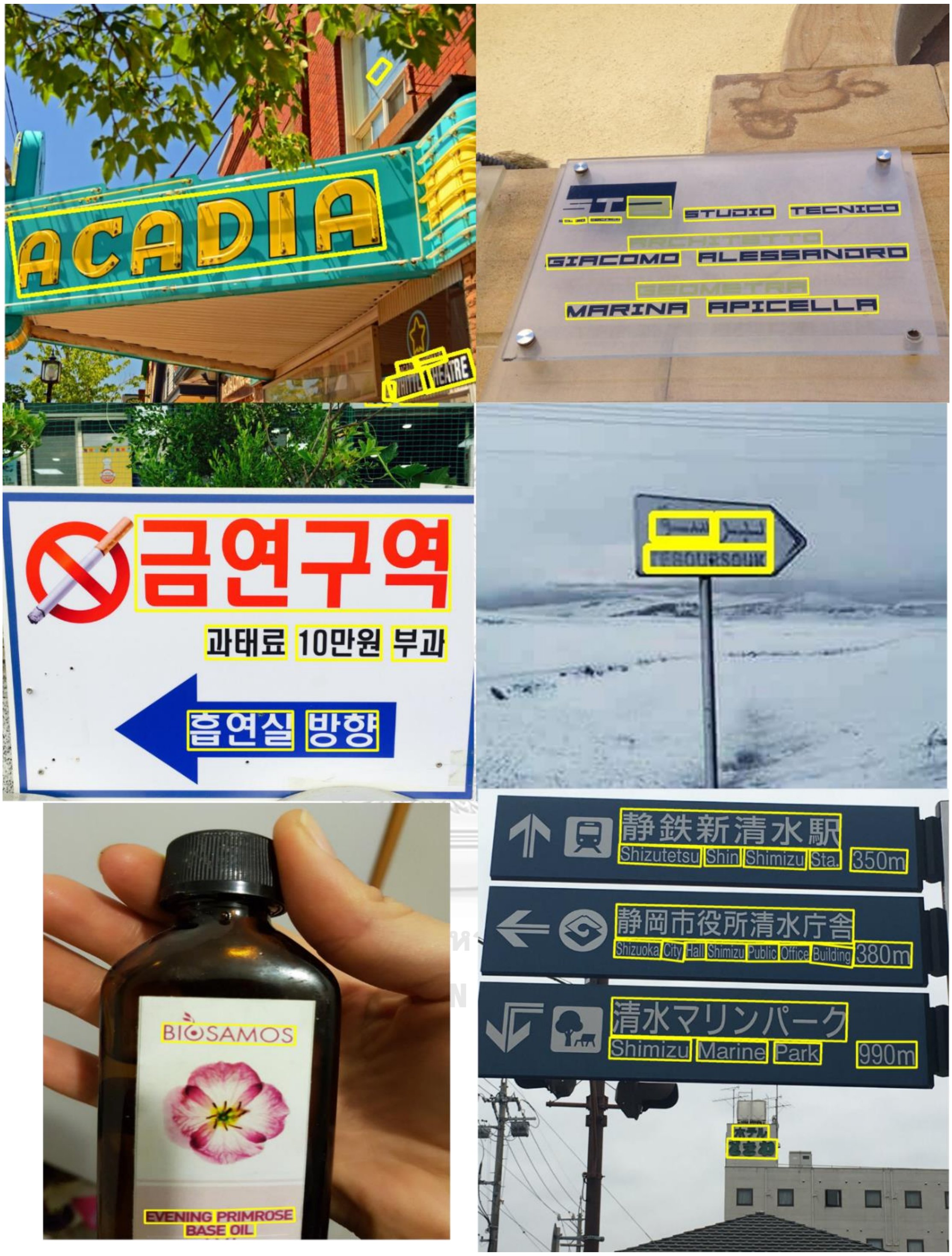


Figure 54 Example results on ICDAR2017 dataset.

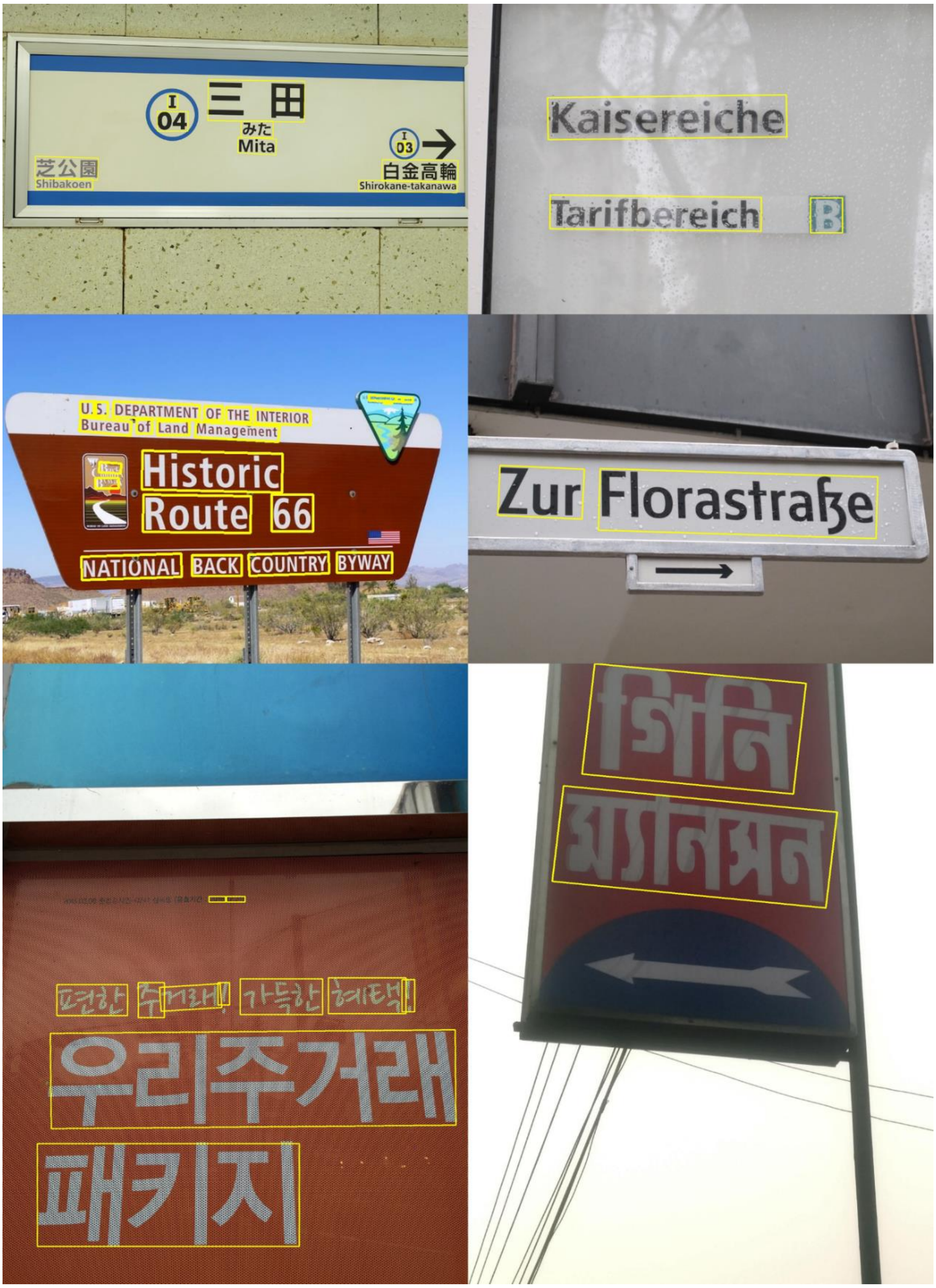


Figure 55 Example results on ICDAR2019 dataset.

5.1.2.3. Multi-Oriented and Curved English Text

Similarly to the experiments on ICDAR2017-MLT and ICDAR2019-MLT, to prove the robustness of the curved text detection problem, the qualitative experiment is also conducted on Total-Text dataset. The pre-trained weight from SynthText is used as based and further fine-tuned on Total-Text for 150 epochs. The experimental results in Table 10 show that the proposed method surpassed other methods in precision respectable recall and f-measure. Figure 56 shows that our method can detect curved text in various styles, shapes, and orientations.

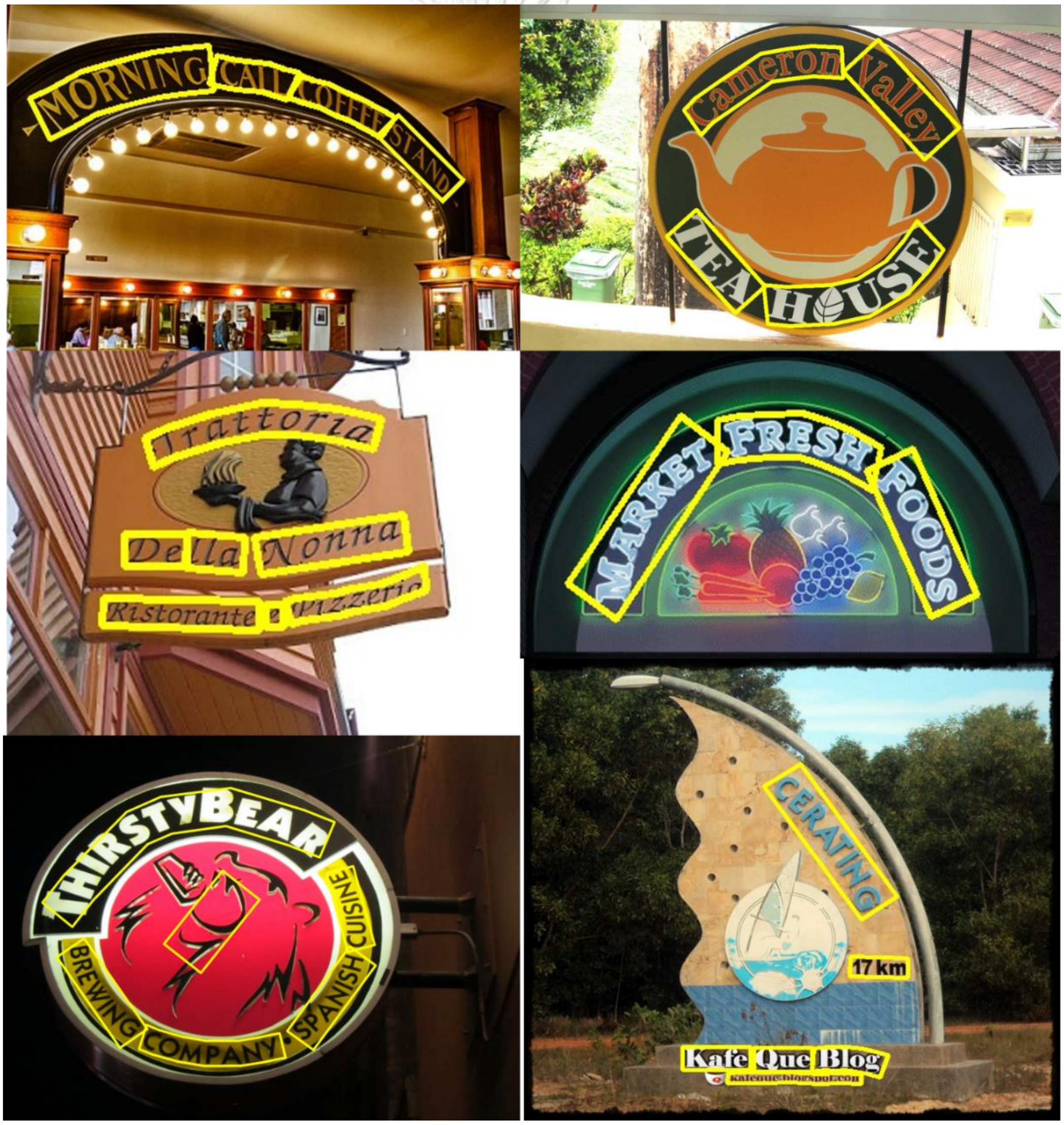


Figure 56 Example results on TotalText dataset.

5.1.2.4. Multi-Orient Thai Text

As the main focus of this dissertation is to design an algorithm for Thai scene text localization and text recognition, the qualitative experiments show in Table 11 are conducted on both BEST2015 and self-collected Thai scene text datasets. The sample results from both datasets are shown in Figure 57 and Figure 58.

Table 11 Experimental results on Thai benchmark datasets. P, R and F denote precision, recall, and f-measure, respectively.

Method	Datasets					
	BEST2015 Dataset			Self-Collected Dataset		
	P	R	F	P	R	F
CTPN (Tian et al., 2016)	71.6	77.2	74.4	70.6	74.2	72.4
EAST (X. Zhou et al., 2017)	74.5	79.8	77.2	72.5	75.8	74.2
SegLink (Shi, Bai, & Belongie, 2017)	74.2	78.8	76.5	74.2	74.8	74.5
TextBoxes++ (Minghui Liao & Bai, 2018)	84.2	80.7	82.5	80.2	78.7	79.5
R2CNN (Jiang et al., 2017)	85.6	80.8	83.2	79.6	77.2	78.4
PixelLink (Deng et al., 2018)	85.7	82.5	84.1	81.7	80.5	81.1
TextSnake (S. Long et al., 2018)	86.9	82.4	84.7	82.2	81.3	81.8
PSENet (W. Wang et al., 2019)	88.7	83.5	86.1	83.7	82.4	83.1
SPCNet (Q. Wang	89.7	85.2	87.5	88.7	84.3	86.5

Method	Datasets					
	BEST2015 Dataset			Self-Collected Dataset		
	P	R	F	P	R	F
et al., 2018)						
Pixel-Anchor (Li et al., 2018)	90.3	86.1	88.2	89.3	87.9	88.6
PMTD (J. Liu et al., 2019)	92.3	90.4	91.4	89.7	89.4	89.6
CRAFT (Y. Baek et al., 2019)	91.8	89.8	90.8	90.1	87.8	89.0
LOMO (C. Zhang et al., 2019)	91.2	92.3	91.8	90.3	89.2	89.8
Proposed Method (without Border Augmentation)	90.2	87.9	88.6	88.2	84.4	85.4
Proposed Method (with Border Augmentation)	92.8	89.8	90.1	89.8	88.8	89.1

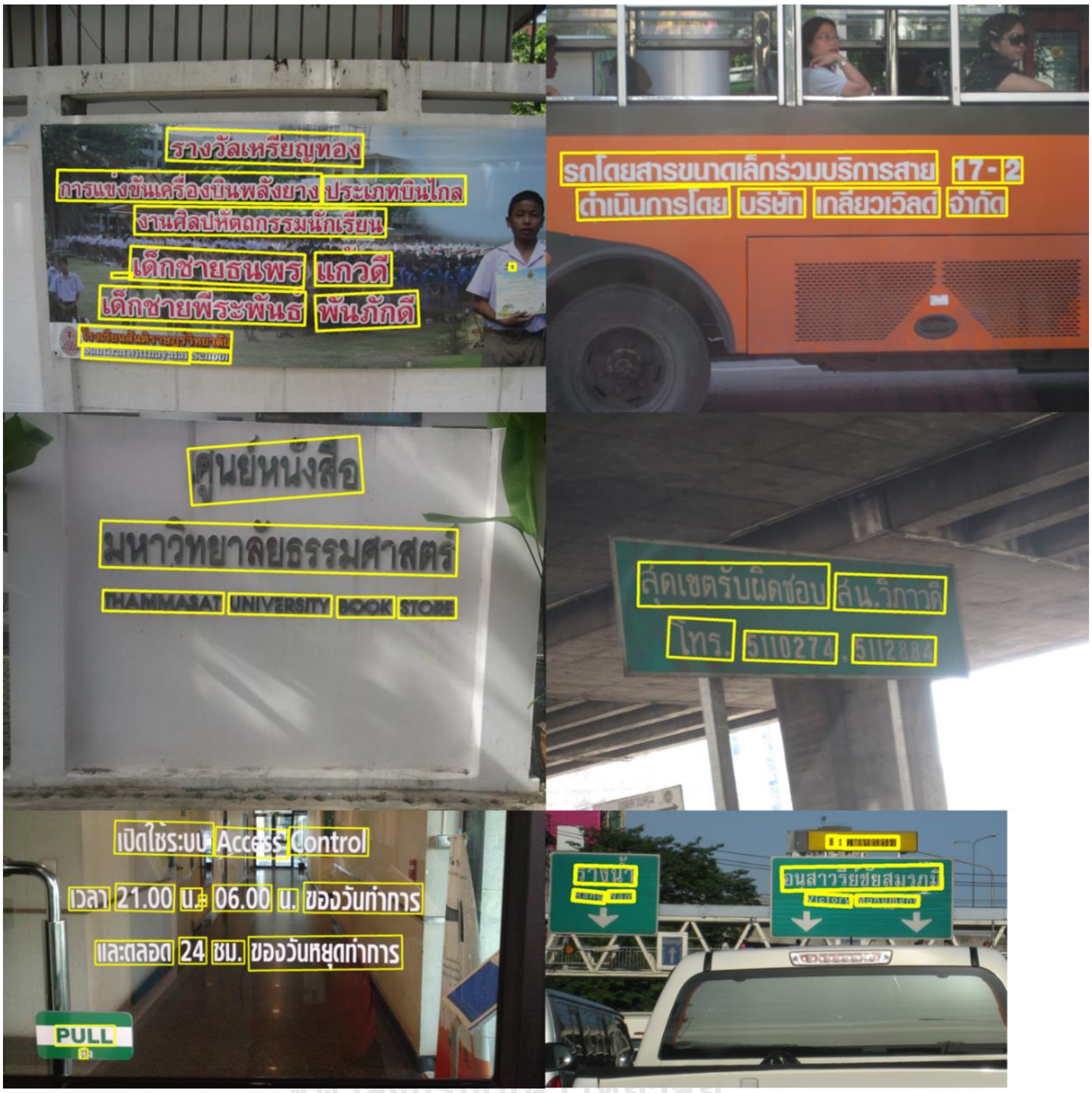

รางวัลเหรียญทอง
การแข่งขันเครื่องบินพลังยาง ประเภทบินไกล
งานศิลปหัตถกรรมนักเรียน
เด็กชายธนพร แก้วดี
เด็กชายพีระพันธ์ พันภักดี
รถโดยสารขนาดเล็กร่วมบริการสาย 17-2
ดำเนินการโดย บริษัท เกลียวเวิลด์ จำกัด
ศูนย์หนังสือ
มหาวิทยาลัยธรรมศาสตร์
THAMMASAT UNIVERSITY BOOK STORE
สุดเขตรับผิดชอบ สน.วิภาวดี
โทร. 5110274 5112884
เปิดใช้ระบบ Access Control
เวลา 21.00 น. - 06.00 น. ของวันทำการ
และตลอด 24 ชม. ของวันหยุดทำการ
PULL
อนุสาวรีย์ชัยสมรภูมิ
Victory Monument

Figure 57 Example results from BEST2015 dataset.

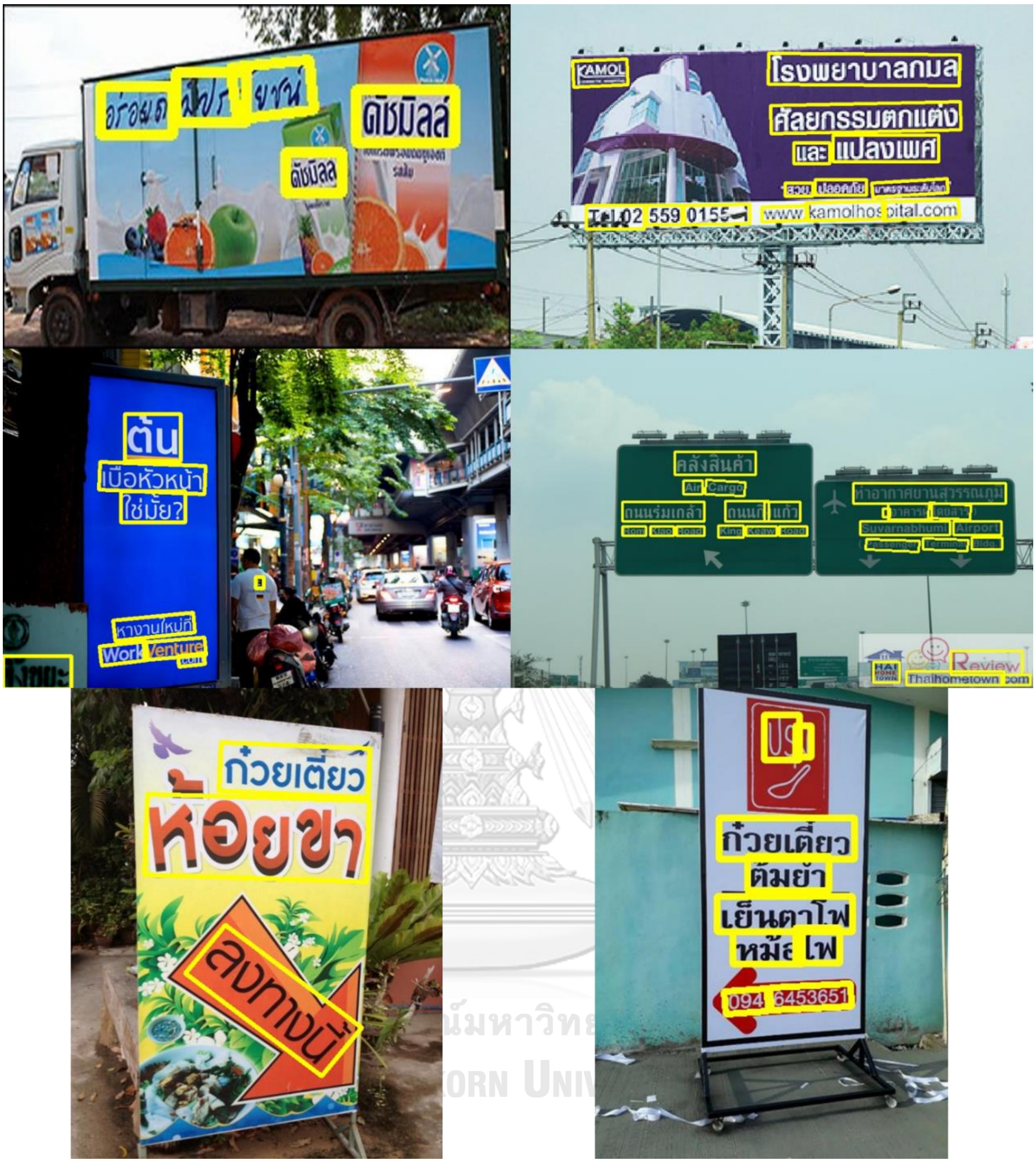
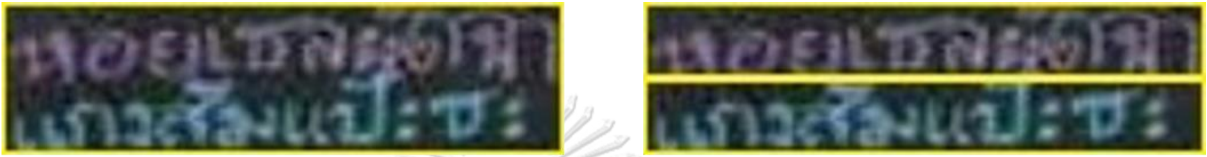


Figure 58 Examples from self-collected dataset.

5.1.3. Border Augmentation Analysis

To analyze the adjacent text instance separation capability of the proposed method, the ablation study on this matter was performed by removing the entire border augmentation module. As shown in Table 10 and Table 11, the border augmentation can improve the result on both English and Thai datasets. The sample effect of border augmentation is shown in Figure 59.



(a) Without border augmentation

(b) With border augmentation

Figure 59 The proposed border augmentation effect.
As shown in (b), the proposed border augmentation algorithm is able to provide
a clear cut between adjacent text instances.

5.1.4. Speed and Accuracy Trade-off Analysis

In this section, the speed assessment experiment is employed. All evaluations were tested on NVIDIA GTX 1080 Ti and Intel i7-4770K. The FPS is calculated by measuring the average per-image execution time.

The qualitative results stated in Table 12 show that the proposed method offers a good balance between execution time and detection accuracy. In this experiment, three feature extraction backbones, ResNet50, ResNet34, and MobileNetV2, are chosen to compare the detection performance in term of accuracy, execution speed and FLOPs counts as shown in Table 13.

Table 12 Text detection speed on various benchmark datasets.

Method	Datasets F-Measure						FPS
	ICDAR			Total-Text	BEST2015	Self-Collected	
	2015	2017	2019				
CTPN (Tian et al., 2016)	60.9	-	-	-	74.4	72.4	7.5
EAST (X. Zhou et al., 2017)	76.4	-	-	-	77.2	74.2	17.1
SegLink (Shi, Bai, & Belongie, 2017)	75.0	-	-	-	76.5	74.5	12.2
TextBoxes++ (Minghui Liao & Bai, 2018)	81.7	-	-	-	82.5	79.5	13.2
R2CNN (Jiang et al., 2017)	82.5	-	-	-	83.2	78.4	-
PixelLink (Deng et al., 2018)	83.7	-	-	-	84.1	81.1	-
TextSnake (S. Long et al., 2018)	82.6	-	-	78.4	84.7	81.8	12.7
PSENet (W. Wang et al., 2019)	87.1	72.1	-	80.9	86.1	83.1	9.6
SPCNet (Q. Wang et al., 2018)	87.2	70.0	-	82.9	87.5	86.5	-
Pixel-Anchor (Li et al., 2018)	87.7	68.1	-	-	88.2	88.6	-
PMTD (J. Liu et al., 2019)	89.3	78.5	82.5	-	91.4	89.6	-
CRAFT (Y. Baek et al., 2019)	86.9	73.9	70.9	83.6	90.8	89.0	11.2
LOMO (C. Zhang et al., 2019)	87.2	68.5	83.6	83.3	91.8	89.8	-
Proposed Method (ResNet50) without Border Augmentation	86.0	72.3	77.9	81.5	88.6	85.4	18.7

Method	Datasets F-Measure						FPS
	ICDAR			Total-Text	BEST2015	Self-Collected	
	2015	2017	2019				
Proposed Method (ResNet50) with Border Augmentation	88.1	73.4	80.9	83.5	90.1	89.1	17.5
Proposed Method (ResNet34) without Border Augmentation	83.2	67.6	72.5	78.9	84.3	79.9	26.2
Proposed Method (ResNet34) with Border Augmentation	84.5	68.9	75.4	80.1	86.6	81.3	25.1
Proposed Method (MobileNetV2) without Border Augmentation	74.6	59.8	68.4	72.1	80.8	72.6	36.7
Proposed Method (MobileNetV2) with Border Augmentation	76.6	60.9	69.4	73.5	81.3	74.4	35.9

Table 13 FLOPs counter of the proposed method on various feature extractor backbones.

Method	Datasets						FLOPs
	ICDAR			Total-Text	BEST2015	Self-Collected	
	2015	2017	2019				
Proposed Method (ResNet50) with Border Augmentation	88.1	73.4	80.9	83.5	90.1	89.1	6.2 G
Proposed Method (ResNet34) with Border Augmentation	84.5	68.9	75.4	80.1	86.6	81.3	4.9 G
Proposed Method (MobileNetV2) with Border Augmentation	76.6	60.9	69.4	73.5	81.3	74.4	0.9 G

From the experimental results, changing the backbone size can significantly reduce the inference time and computational complexity. By choosing the proper model respects to used applications or computational constraints, for example, translation applications requiring real-time detection speed, users can choose ResNet34 or MobileNetV2 depending on their usage cases.

5.1.5. Implementation Details

The model is first trained on the generated SynthText dataset for 1 epoch and fine-tuned on each standard text localization benchmark datasets except Thai datasets until converged. All weights in trainable layers are initialized by using Kaiming's initialization (K. He et al., 2016). Adaptive Momentum Estimation (Adam) (Kingma & Ba, 2015) is used as an optimizer for this model. During the training on SynthText, the learning rate was initially set to 10^{-3} and decayed to 10^{-4} by using a cosine annealing scheduler. The batch size was set to 4 and then increased to 16 by using gradient accumulation. According to this work (Smith et al., 2018), increasing training batch sizes may slightly boost the model generalization and accuracy.

After training on SynthText was finished, the model was then fine-tuned on standard benchmarks, ICDAR2015, ICDAR2017-MLT, ICDAR2019-MLT, Total-Text, and Thai Text dataset. The learning rate was initially set to 10^{-4} decayed to 10^{-5} by using cosine annealing scheduler. The batch size was set to 4. Since the ICDAR datasets contained some non-readable text regions, which were labeled as "###'", these regions were not used during the training. Online Hard Negative Mining (OHEM) (Shrivastava et al., 2016) is adopted with a ratio of 1:3 to correct the imbalance ratio between the number of text and non-text pixels.

As the data augmentation is a crucial part of increasing the algorithm robustness, the following online augmentation techniques are used to raise the number of training samples:

- Standard photometric distortion (W. Liu, Anguelov, et al., 2016).
- Image rotation, horizontal and vertical.
- Image size re-scale in range [0.5, 3].
- Randomly cropping image regions.
- Mean and standard deviation normalization.

In this work, a new mosaic data augmentation is also introduced. Mosaic augmentation is a new technique that randomly combines various interesting patches into a new training image. This technique can increase the amount of training data and algorithm robustness. The sample input and output images are shown in Figure 60.

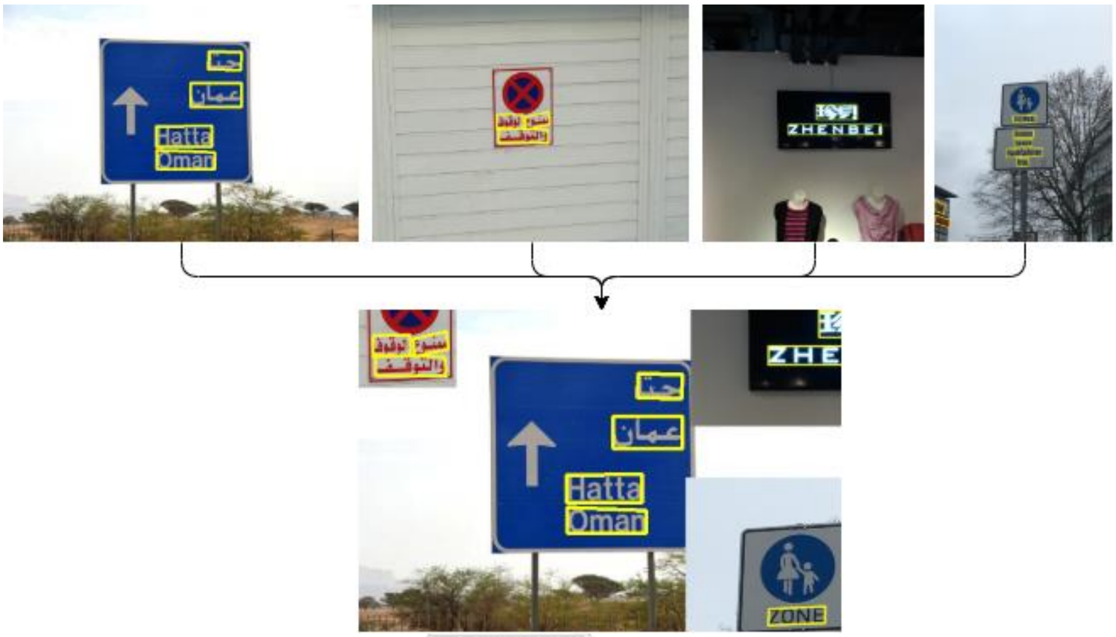


Figure 60 Mosaic image augmentation.

5.2. Thai Scene Text Recognition

To evaluate the performance of the proposed scene text localization, a quantitative test on BEST2015 and self-collected datasets are without any dictionary correction.

5.2.1. Test Dataset Acquisition

For Thai scene text recognition performance evaluation, the same dataset from Thai scene text recognition is used. Text instances from BEST2015 and self-collected are combined into new text instances pool, yielding 6,844 text instances. From the combined pool, only Thai text instances are selected, resulting in the Thai scene text test dataset, which contains 5,296 text instances. The text instance samples are shown in Figure 61.

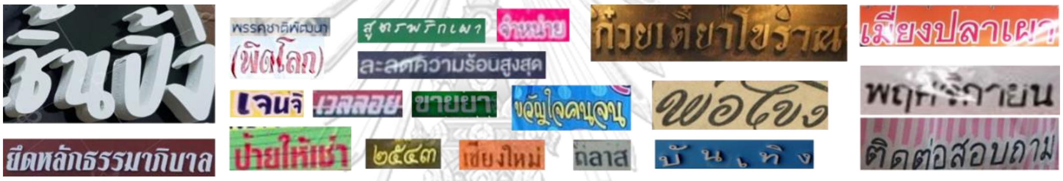


Figure 61 Thai scene text instances samples.

5.2.2. Results

In this section, a series of extensive experiments were conducted on all possible proposed combinations as follows:

Transformation: This module normalizes the input image into a more appropriable format for recognition. It can be select or deselect base on the configuration.

Feature Extraction: This module extracts the visual information feature from the normalized image. In this experiment, ResNet50-FPN is selected as a baseline. Further studies on backbone configurations are shown in the ablation studies section.

Sequence Modeling: The visual sequence feature is formed from this module. This experiment utilizes Long-Short Term Memory (LSTM) as a baseline. This module can be either select or deselect base on the configuration.

Prediction: In seq2seq problem group, two main concepts used in this kind of problem are Connectionist Temporal Classification (CTC) and Attention Mechanism

Encoding Scheme: This dissertation proposes the four encoding schemes for Thai text recognition. The experiment is conducted on all proposed encoding schemes.

All the evaluated models are trained from scratch by using the same set of training data and hyperparameters. The qualitative experimental results are shown in Table 14. In this work, full matched between actual and predicted string is a required criterion to be counted as a corrected output. No lexicon and dictionaries are used during the evaluation.

Table 14 Thai scene text qualitative results.

No	Transformation	Feature Extraction Backbone	Sequence Modeling	Prediction		Encoding Scheme	Accuracy (%)	Average Edit Distance
	TPS		LSTM	CTC	Attention			
1A	✗	ResNet-50-FPN	✗	✓		Single Level - Single Character	67.2	0.36
2A	✗		✗		✓		68.3	0.33
3A	✗		✓	✓			67.4	0.36
4A	✗		✓		✓		69.1	0.32
5A	✓		✗	✓			68.9	0.35
6A	✓		✗		✓		71.2	0.31
7A	✓		✓	✓			73.8	0.29
8A	✓		✓		✓		74.0	0.29
1B	✗		✗	✓		Single Level - Grouped Character	72.2	0.30
2B	✗		✗		✓		75.1	0.26
3B	✗		✓	✓			74.6	0.28
4B	✗		✓		✓		77.8	0.25
5B	✓		✗	✓			74.9	0.27
6B	✓		✗		✓		76.4	0.25
7B	✓		✓	✓			76.8	0.25
8B	✓		✓		✓		78.1	0.22
1C	✗		✗	✓		Three-Level	80.3	0.19
2C	✗		✗		✓		80.9	0.18
3C	✗		✓	✓			79.4	0.19
4C	✗		✓		✓		82.6	0.16
5C	✓		✗	✓			83.1	0.16
6C	✓		✗		✓		82.9	0.16
7C	✓		✓	✓			81.9	0.18
8C	✓		✓		✓		83.0	0.16
1D	✗		✗	✓		Four-Level	80.6	0.17
2D	✗		✗		✓		81.2	0.16
3D	✗		✓	✓			80.6	0.17
4D	✗		✓		✓		83.4	0.15
5D	✓		✗	✓			83.2	0.15
6D	✓		✗		✓		83.4	0.15
7D	✓		✓	✓			82.2	0.17
8D	✓		✓		✓		**84.5**	**0.14**

According to the all possible combinations experiment shown in Table 14, the proposed multi-level attention methods (Three and Four levels) achieve the first and second rank accuracy and lowest editing distance compared to the other proposed encoding method by significant margins. On the contrary, the traditional encoding scheme (Single Level – Single Character) achieves the lowest accuracy among the others. These qualitative experimental results confirm the effectiveness of the proposed multi-level attention. More analysis of each encoding scheme behaviors is shown in the ablation studies section.

For good clarity on experimental result comparison, all possible configuration accuracies are plotted in Figure 62.

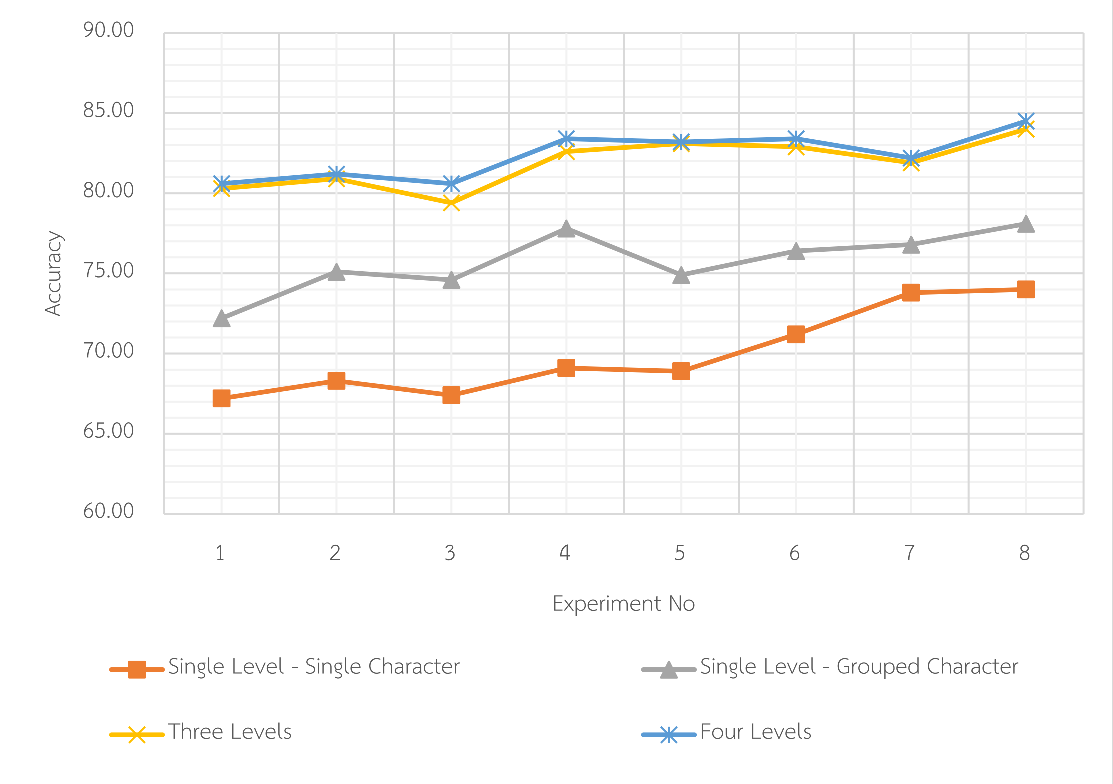
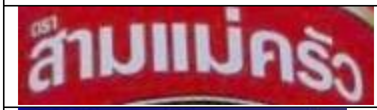
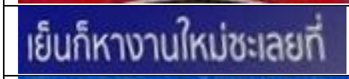
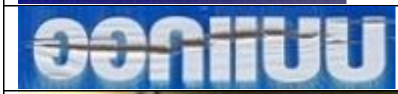
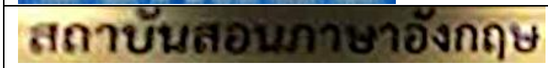
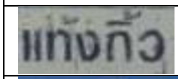
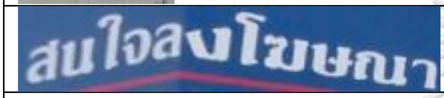
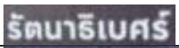
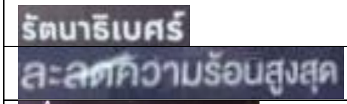
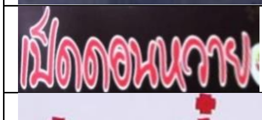
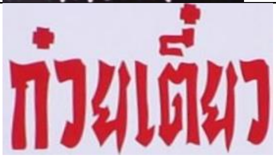
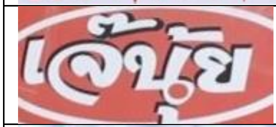
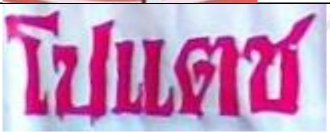
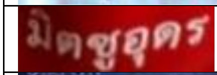
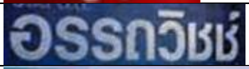
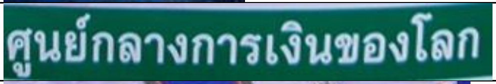
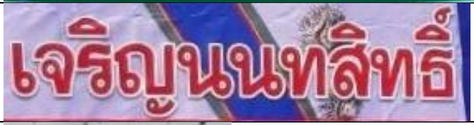
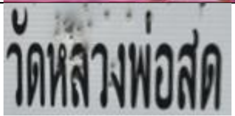
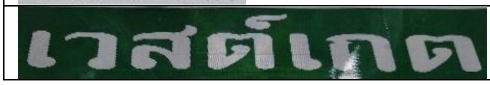


Figure 62 Experimental results on all possible configurations.

The example of output recognition results and input images from the best combination (TPS/ResNet50-FPN/LSTM/Attention) are shown in Table 15.

Table 15 Examples of recognition result from the best configuration.

Text Instance Image	Ground truth	Predicted Output
	สามแม่ครัว	สามแม่ครัว
	เย็นก็หางานใหม่ซะเลยที่	เย็นก็หางานใหม่ซะเลยที่
	ออกแบบ	ออกแบบ
	สถานบันสอนภาษาอังกฤษ	สถานบันสอนภาษาอังกฤษ
	แท้งกิ้ว	แท้งกิ้ว
	สนใจลงโฆษณา	สนใจลงโฆษณา
	รัตนาธิเบศร์	รัตนาธิเบศร์
	ละลดความร้อนสูงสุด	ละลดความร้อนสูงสุด
	เป็ดดอนหวาย	เป็ดดอนหวาย
	ก๋วยเตี๋ยว	ก๋วยเตี๋ยว
	เจ๊นุ้ย	เจ๊นุ้ย
	โปแตช	โปแตช
	มิตซูอุดร	มิตซูอุดร
	อรรถวิชช์	อรรถวิชช์
	ศูนย์กลางการเงินของโลก	ศูนย์กลางการเงินของโลก
	เจริญนนท์สิทธิ์	เจริญนนท์สิทธิ์
	วัดหลวงพ่อสด	วัดหลวงพ่อสด
	เวสต์เกต	เวสต์เกต

5.2.3. Ablation Studies

To analyze the impact of each component on the recognition performance and behaviors, a series of ablation studies are conducted in the below section. All the evaluated models are trained from scratch by using the same training data and hyperparameters.

5.2.3.1. Encoding Scheme Analysis

In this section, each proposed encoding scheme's impact is further studied based on the best combination, which is TPS + ResNet50-FPN + LSTM + Attention. The result from each encoding scheme is shown in Table 16.

Table 16 Best result from each proposed encoding scheme.

No	Transformation	Feature Extraction Backbone	Sequence Modelling	Prediction	Encoding Scheme	Accuracy (%)	Average Edit Distance
	TPS		LSTM				
1	✓	R ResNet-50-FPN	✓	Attention	Single Level - Single Character	74.0	0.29
2					Single Level – Grouped Character	78.1	0.22
3					Three-Level	83.0	0.15
4					Four-Level	84.5	0.14

5.2.3.1.1. Single Level – Single Character Encoding

This encoding scheme got the lowest accuracy among the other encoding methods. After analyzing the output wrong predicted results, the following reasons can be concluded.

1. Feature alignment inconsistency between prediction stage and input image character

Many previous works on English scene text recognition algorithm also treat this problem as seq2seq problem. In the ordinary English writing system, only one character is appearing at each location in text instance. This feature makes the input image or visually features aligned with sequence features as shown in Figure 63.

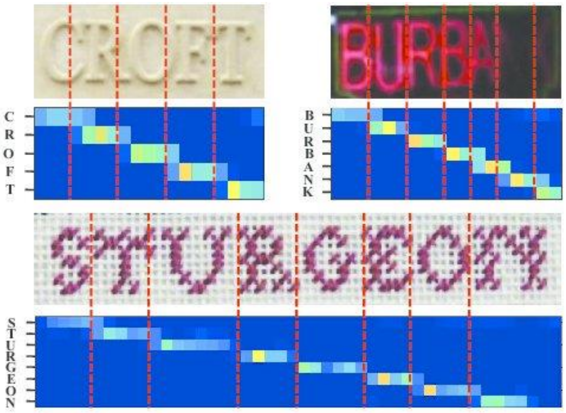


Figure 63 Visual feature aligned with sequence feature (Shi et al., 2019).

However, in Thai writing system, multiple characters can be writing over and above at the same location, for instance, Thai string "ชื่อ", tone mark ่ and vowelื write over main character ช. In this encoding scheme, after inspecting the attention layer weights respect to the corresponded location visually features, the attention layer output seems to produce an inaccurate prediction when taking into deeper time steps on long text instance as shown in Figure 64. It may be possible to conclude that using this encoding scheme makes the visual features not aligned with sequence features.

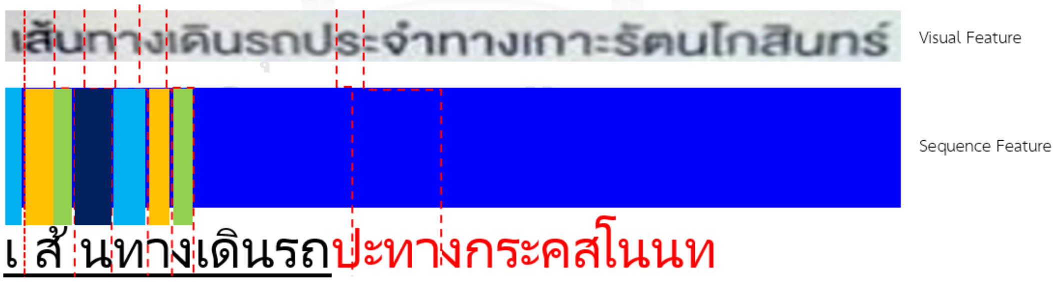
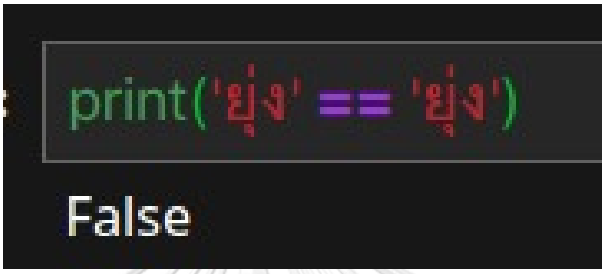

เส้นทางเดินรถประจำทางเกาะรัตนโกสินทร์
Visual Feature
Sequence Feature
เ ส้ นทางเดินรถปะทางกระคสโนนท

Figure 64 Example of unaligned visual and sequence feature. Each color strip represents the focused sequence feature respected to character. By using this encoding scheme, at character ส้, upper vowel ้ visual feature is not aligned to its sequence feature. This unaligned behavior will be increased in a long text instance, making the network tends to produce wrong recognition output.

2. Position swapping among tone marks and vowels

As mentioned in the previous section, Thai text writing system can be divided into multiple levels. This characteristic also appears in the OS font rendering system and programming paradigm, which may be confusing in some cases, as shown in Figure 65. Both displayed outputs by the font rendering system are similar, but the writing systems are not.

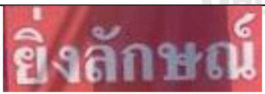
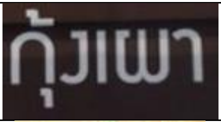
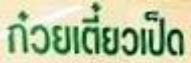
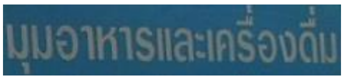


```
print('ยุ่ง' == 'ยุ่ง')
False
```

Figure 65 Thai writing system inconsistency in programming paradigm: the right string “ยุ่ง” consists of “ย ุ ่ ง” while the left string “ยุ่ง” consists of “ย ่ ุ ง”.

After investigating the output shown in Table 17, many incorrect results caused by tone marks and vowels appear in inconsistent locations.

Table 17 The effect of tone marks and vowels location inconsistency in single level – single character encoding

Text Instance Image	Ground truth	Predicted Output
	ยิ่งลักษณ์	่ ย ิ ง ล ั ก ษ ณ ์
	กุ้งเผา	้ ก ุ ง เ ผ า
	ก๋วยเตี๋ยวเป็ด	ก ๋ ว ย เ ๋ ตี ย ว เ ป ็ ด
	มุมอาหารและเครื่องดื่ม	ม ุ ม อ า ห า ร แ ล ะ เ ค ่ รื อ ง ด ่ ื ม

5.2.3.1.2. Single Level – Grouped Character Encoding

Single Level – Grouped Character encoding scheme acquires the third rank among the other encoding methods. This encoding method determines all possible character combinations at each location and maps them into individual classes. According to the Thai writing system character types, as shown in Table 18, all possible combinations can be calculated from the writing system criteria shown in Table 19.

Table 18 Thai character classification based on Thai writing system.

Characters Type	Characters	Number of Character
Tone Marks	่้๊๋ +	4
Upper Vowels	ัิีึื ํ /็์	4 + (2)
Main Line	ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ / ะ า ๅ เ แ โ ไ ใ ๆ ฯ	44 + (10)
Lower Vowels	ุู	2

Table 19 Calculation of possible class combination based on Thai writing system.

Thai writing system criteria		Total Combinations
Main line character without any vowel and tone mark	44 + 10	54
Main line character (exclude main line vowel) + upper tone mark	44 x 4	176
Main line character (exclude main line vowel) + upper vowel	44 x 8	352
Main line character (exclude main line vowel) + lower vowel	44 x 2	88
Main line character (exclude main line vowel) +	44 x 4 x 8	1,408

Thai writing system criteria		Total Combinations
upper tone mark + upper vowel		
Main line character (exclude main line vowel) + upper tone mark + lower vowel	44 x 4 x 2	352
Possible combinations		**2,430**

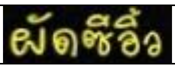
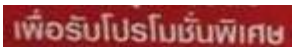
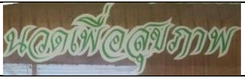
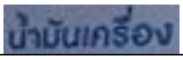
Table 19 shows that there is a large number of possible combinations that can be formed from Thai writing system. Typically, in the image classification problem, the expected accuracy will be lower when the output class number gets higher. As stated by the qualitative experimental result, this assumption also valid on this problem.

5.2.3.1.3. Three-Level Encoding

The three-level encoding scheme is ranked second among the other encoding. This encoding scheme combined a group of tone marks and upper vowels into an individual class, yielding 33 upper-level classes (including blank character), 54 main-line classes, and three lower-level classes (including blank character), respectively.

After investigating the prediction results, this encoding scheme achieved competitive accuracy compared to four-level encoding and surpassed single-level encoding methods. This confirmed the effectiveness of the proposed multi-level encoding could improve overall recognition accuracy. However, three-levels encoding give an unexpected upper line recognition result in some cases. The small upper vowels such as Mai Eak (่) and Mai Jai Ta Wa (๋) sometimes disappear. The example of incorrect results is shown in Table 20.

Table 20 Samples of incorrect result using three-level encoding scheme.

Text Instance	Ground truth	Predicted Output
	ผัดซีอิ้ว	ผัดซี**อิ**ว
	เพื่อรับโปรโมชั่นพิเศษ	เพื่อรับโปรโม**ชั**นพิเศษ
	นวดเพื่อสุขภาพ	นวดเ**พื**อสุขภาพ
	น้ำมันเครื่อง	**น้**ำมันเค**รื**อง

5.2.3.1.4. Four-Level Encoding

Four-level encoding delivers the best accuracy among the proposed encoding methods and exceeded single-level encoding methods by a large margin. This encoding scheme separate Thai text into four levels, as indicated in Table 16, where each level is treated as seq2seq problem using shared visual feature. By using this strategy, each level attention module can focus and distinguish character sequence, leading to an accurate text recognizer.

5.2.3.2. Speed Analysis

An experiment on overall speed measurement is conducted on the best combination (TPS+ResNet50-FPN+LSTM+Attention) to analyze each stage's impact in the proposed method. The experiment is conducted under the same hardware specification, which is NVIDIA GTX 1080 Ti and Intel i7-4770K with 16GB memory. To ensure fair speed testing, the experiments are launched 100 times and measure the average run time with a batch size of 1.

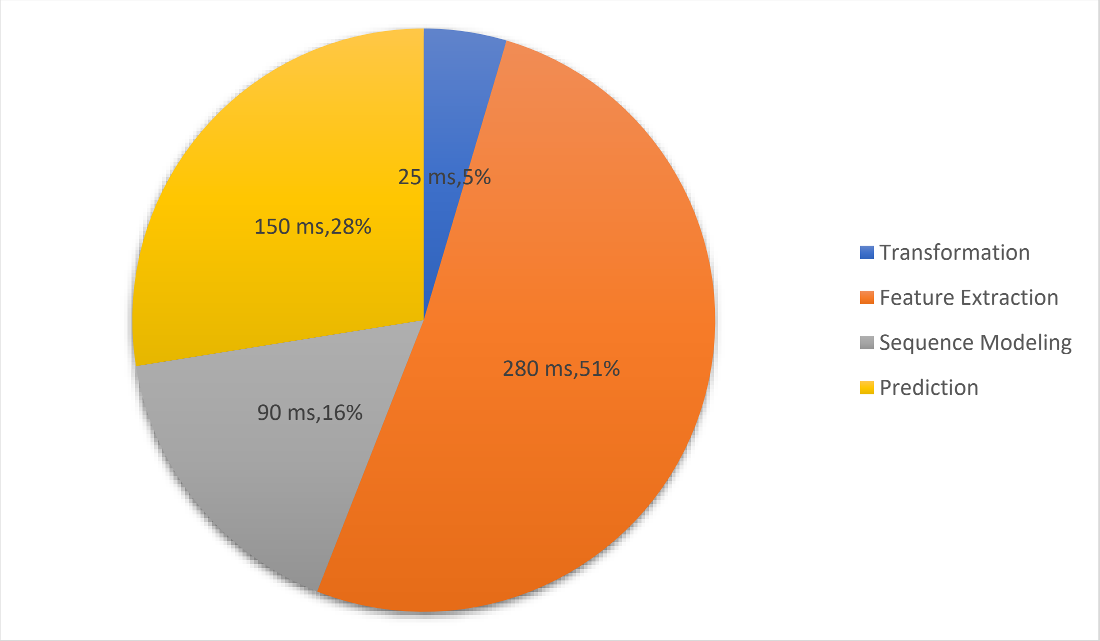


Figure 66 Average runtime of each stage with sequence length equals to 32.

As shown in Figure 66, the proposed method runtime (545ms) mostly spends on the feature extraction stage (280ms) follows by the prediction stage (150ms), while the transformation stage slightly impacts the overall runtime. This experiment is conducted with ResNet50-FPN backbone and limits the sequence length to 32 main line characters. This reflects that the heaviest part of the proposed method is feature extraction. If the user wants to decrease the overall runtime or has a computational time criterion, reducing the feature extractor backbone to the proper size or change the other efficient backbones can greatly decrease overall runtime.

The addition experiments on different backbones are shown in the backbone analysis section.

On the other hand, the prediction stage still spends the second rank runtime, among other stages. If the sequence length is further extended to 64. The overall runtime is increased and shown in Figure 67.

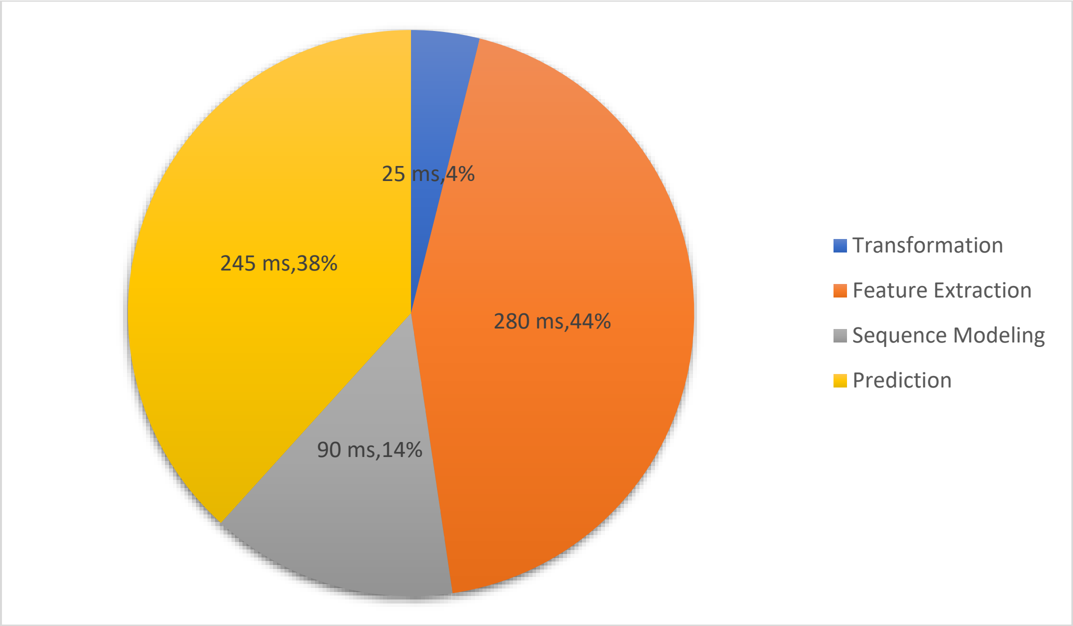


Figure 67 Average runtime of each stage with sequence length equals to 64.

As shown in Figure 67, the overall runtime increases from 545ms to 640ms (+17.4%). This shows that the overall runtime is directly related to the sequence length. Thus, choosing a proper sequence length limit is one variable to reduce overall inference time.

5.2.3.3. Backbone Feature Pyramid Effect

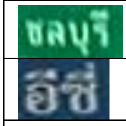
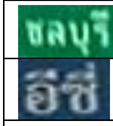
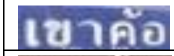
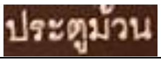
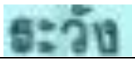
In this section, the experiment is conducted to prove the effectiveness of the proposed feature pyramid feature extractor. Many previous works on small object image classification and detection have proved that combining multi-scaled features from shallow and deep layers can dramatically improve recognition performance. From this assumption, in this work, the scene text recognition network based on ResNet-50 incorporated with Feature Pyramid is designed.

Table 21 Accuracy comparison of proposed network structure between with and without feature pyramid.

No	Transformation	Feature Extraction Backbone	Sequence Modelling	Prediction	Encoding Scheme	Accuracy (%)	% Gain	FLOPs
1	TPS	ResNet50	LSTM	Attention	Four Level	77.4	-	50.7G
2		ResNet50-FPN				84.5	+7.1%	59.7G

As shown in Table 21, the recognition performance significantly improves (74.4% -> 84.5%) without any change in other components except feature extractor structure. The increasing accuracy gap comes from the reasons that many Thai characters are looking very similar, requiring multiple scaled fine-grained features to be distinguishable, and small components such as tone marks and vowels also benefit from shallow layer visual features. The sample images and their recognition outputs, which profit from FPN module, are shown in Table 22.

Table 22 Sample images and recognition outputs from without and with FPN network structure.

Text Instances	Predicted Output (without FPN)	Predicted Output (with FPN)
	ชลบุ**ริ**	ชลบุรี
	อี**ซิ่**	อีซี่
	เขา**คั**อ	เขาค้อ
	ประตู**มั**วน	ประตูม้วน
	ระ**วิ**ง	ระวัง

5.2.3.4. Backbone Size Analysis

To verify the impact of network size/parameters and recognition accuracy, the experiments on different ResNet structures, which are ResNet-18, ResNet-34, ResNet-50, are conducted. In this work, not only the accuracy-wise models are proposed but also the low complexity model like MobilNetV2. As shown in Table 23, in the low and middle model regime, ResNet-34 and MobileNetV2 get the second (78.1%) and third (71.8%) ranks. The biggest network (ResNet-50) achieves the best accuracy (84.5%) among the others, which is correlated to the number of network parameters.

In the aspect of model complexity, MobileNetV2 utilizes the lowest computational complexity while still giving competitive accuracy to ResNet34 compared to the number of FLOPs. The depthwise separable convolutional used in MobileNetV2 can reduce a lot of floating-point operations, suitable for low computation power situations such as edge computing deployment without GPU, and smartphone. To increase this model accuracy, a further study on knowledge distillation may require.

Table 23 Comparison between network size and recognition accuracy

No	Transformation	Feature Extraction Backbone	Sequence Modelling	Prediction	Encoding Scheme	Accuracy (%)	Accuracy Gain (%)	FLOPs	% FLOPs Increase
1	TPS	ResNet18-FPN	LSTM	Attention	Four Level	68.9	-	40.6G	-
2		MobileNetV2-FPN				71.8	+2.9	27.9G	(-31.3%)
3		ResNet34-FPN				78.1	+9.2	49.3G	21.4%
4		ResNet50-FPN				84.5	+15.6	59.7G	47.1%

5.2.3.5. Number of Training Data Analysis

Four experiments with different amounts of training data are conducted on the best combination (TPS+ResNet50-FPN+LSTM+Attention) to analyze the amount of training data effect. The bigger datasets are generated by inheriting the smaller datasets and extending with new images. In this section, all experiments are trained from scratch with different amounts of training data. The experimental results are shown in Figure 68.

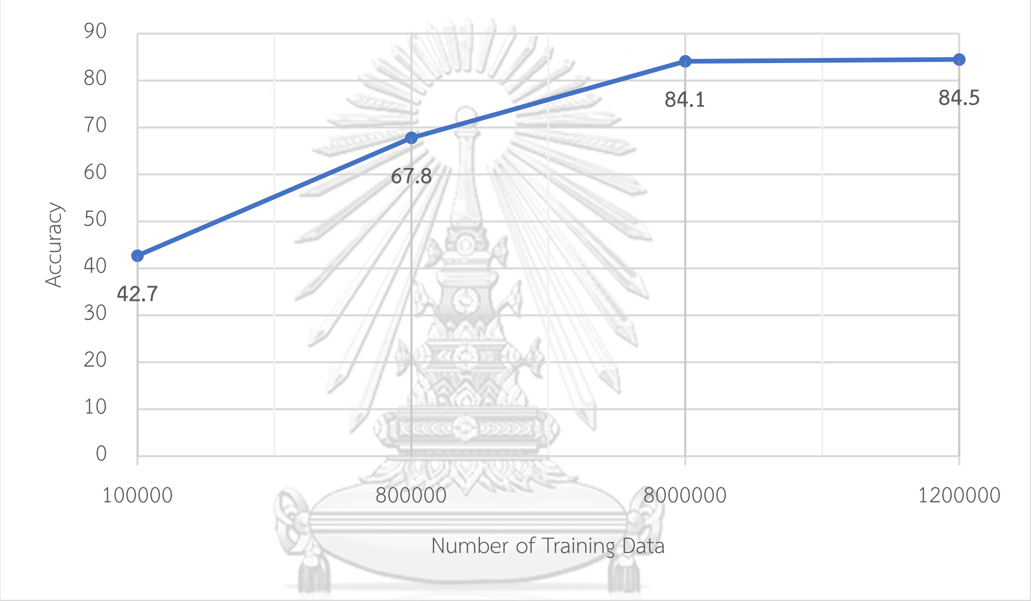


Figure 68 Experimental result on the different number of training data.

Starting with 100k training cropped word images, the best combination achieves 42.7% accuracy on test data. After increasing the amount of training data by 800% (100k->800k), recognition performance is improved from 42.7% to 67.8% (+25.1%). This reflects that there is still room for accuracy improvement if the training data is further increased.

The experiment is further conducted by increasing the number of training data by 1000% (800k->8M). The recognition accuracy is still improved from 67.8% to 84.1% (+16.3%). For the last experiment, the training data is risen by 150% (8M->12M). However, the accuracy improvement is minimal, gaining from 84.1% to 84.5%

(+0.4%) while training time rises from 3.5 days to 5 days. It may conclude that the new recognition method and an improved text synthesis method are needed.

5.2.3.6. RGB and Grayscale Input Analysis

Two experiments are launched under the same network configuration to determine the effect of color and computational complexity in text recognition problem, but the input images are in grayscale and RGB colorspaces.

Table 24 Accuracy and computational complexity on different input image colorspaces.

No	Input Image Colorspace	Transformation	Feature Extraction Backbone	Sequence Modelling	Prediction	Encoding Scheme	Accuracy (%)	FLOPs	% FLOPs Increase
1	Grayscale	TPS	ResNet50-FPN	LSTM	Attention	Four Level	84.54	59.7G	-
2	RGB						84.58	64.1G	7.37%

As shown in Table 24, the network trained on RGB input image receives marginal gain accuracy (84.58%) compared to grayscale input (84.54%), while the computational complexity (FLOPs) increased by 7.37 percent. The gained accuracy in RGB model may come from the number of parameters in first layer are more than the grayscale the first layer or the randomness in parameter initialization. In conclusion, the gained accuracy is negligible and may not worth the increased computation complexity.

5.2.3.7. Input size Analysis

To analyze the effect of input image resolution, the best model combination (TPS+ResNet50-FPN+LSTM+Attention) are trained with different input image resolution.

Table 25 Experimental results on different input image sizes.

No	Input size	Transformation	Feature Extraction Backbone	Sequence Modelling	Prediction	Encoding Scheme	Accuracy (%)	FLOPs	% FLOPs Increase
1	32x300	TPS	ResNet50-FPN	LSTM	Attention	Four Level	66.3	**50.2G**	-
2	48x350						74.2	**54.1G**	**7.7%**
3	64x350						84.5	**59.7G**	**18.9%**
4	77x400						84.9	**62.8G**	**25.7%**

According to the experiment results shown in Table 25, input image size significantly impacts the final recognition accuracy. The bigger input image size gives better recognition accuracy at the cost of computational efficiency.

Starting with the smallest input image size 32x300, this size gives the lowest accuracy (66.3%). After increasing the input image size to 48x350 pixels, the recognition accuracy is raised from 66.8% to 74.2% (+7.4%). This shows that the recognition accuracy is directly related to input image size.

The input image size is further increased from 48x350 to 64x350 pixels. The recognition accuracy is still improved from 74.2% to 84.5% (+10.3%). In the last experiment, the input image size is increased to 77x400 pixels. However, there is a limited improvement in the recognition accuracy from 84.5% to 84.9% (+0.4%), which may not worth the gained computation complexity (+25.0%).

Thai language consists of many small components, i.e., vowels and tone marks. These small components require fine-grained input image resolution to distinguish between similar Thai characters. As shown in the experimental results, increasing the input image size can improve the overall accuracy. Nevertheless, the increased image size also increases computational complexity. Hence, in this work, the final input image size is set to 64x350 pixels, which is a good balance input image size between accuracy and computation complexity.

5.2.3.8. Global Character Context Loss Effect Analysis

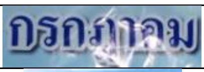
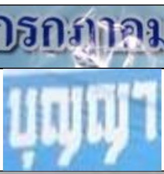
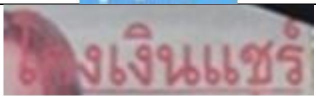
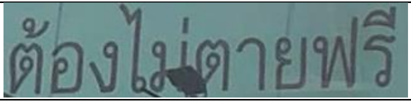
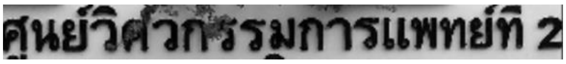
In order to demonstrate the effectiveness of the proposed global character loss, the network is trained with and without global character context loss configurations. The experimental results are shown in Table 26.

Table 26 Accuracy comparison between with and without global character context loss.

No	Transformation	Feature Extraction Backbone	Sequence Modelling	Prediction	Encoding Scheme	Global Character Loss	Accuracy (%)
1	TPS	ResNet50-FPN	LSTM	Attention	Four Level	✗	82.3
2						✓	84.5

The experimental results in Table 26 prove that the proposed global character context loss can improve the overall accuracy in the foreseeable margin (+2.3%), especially on some occluded long text instances and ambiguous characters. The sample of recovered results from global character loss are shown in Table 27.

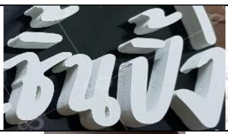
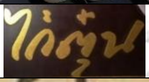
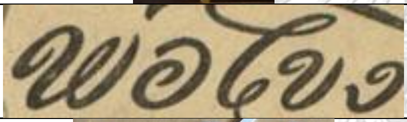
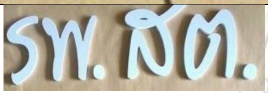
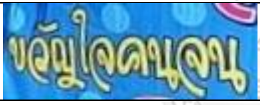
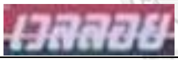
Table 27 Predicted outputs comparison between without and with global character context loss structure.

Text Instance	Predicted Output (without global character context loss)	Predicted Output (with global character context loss)
	กรก**ภ**าคม	กรก**ฎ**าคม
	บุ**ณุณ**า	บุ**ญญ**า
	โ**ถ**งเงินแชร์	โ**ก**งเงินแชร์
	ต้องไ**เ**ตายฟรี	ต้องไ**ม่**ตายฟรี
	ศูนย์วิ**ค**วกรรม การแพทย์ที่ 2	ศูนย์วิ**ศ**วกรรม การแพทย์ที่2

5.2.4. Failed Cases

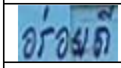
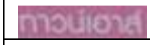
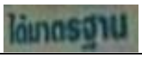
Calligraphic Fonts - In this case, text instances generally appear in difficult or unique font styles, for example, brands or cursive writing. The characters in this case require distinct visual features to recognize rather than generalize visual features. The sample images in this case are shown in Table 28.

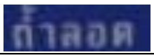
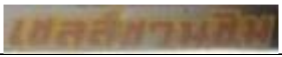
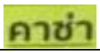
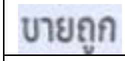
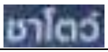
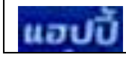
Table 28 Calligraphic text instances and recognition outputs.

Text Instance	Ground truth	Predicted Output
	ชิ้นปิ้ง	**ชินบิ้ง์**
	ไก่ตุ๋น	ไก่ตุ๋**บ**
	พ่อโขง	**ม**่อ**เบ**ง
	รพ.สต.	ธพ.สต.
	ขวัญใจคนจน	ขวัญ**ใล**คน**ล**น
	เวลลอย	เ**า**ลล**0**ย

Low Resolution – The proposed models may not distinguish low-resolution text instances, especially small characters like vowels and tone marks. Using multi-scaled image pyramids or super-resolution may improve recognition performance in this case. The sample images and recognition results are shown in Table 29.

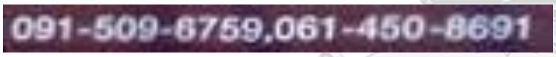
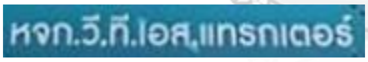
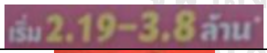
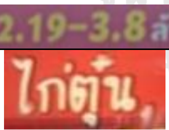
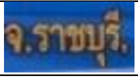
Table 29 Low resolution text instance and recognition outputs.

Resized Text Instance	Original Image Size (HxW)	Ground truth	Predicted Output
	14x25	อร่อยดี	อ**เอ**ดี
	16x64	ทาวน์เฮาส์	**ก**าว**เอา**ส์
	20x43	ได้มาตรฐาน	ไ**ก้**มา**ด**รฐาน

Resized Text Instance	Original Image Size (HxW)	Ground truth	Predicted Output
	14x52	ถ้ำลอด	**ถำ**ลอด
	20x137	เชลล์ชวนชิม	เ**ซ**ล**สข**วนชิม
	22x46	คาซ่า	คา**ช่**า
	20x52	ขายถูก	**บ**ายถูก
	15x37	ชาโตว์	**ซ**าโตว์
	15x41	แฮปปี้	แ**อ**บปี้

Special Characters – Sometimes model cannot distinguish ambiguous special characters, for example, , (comma) and . (dot) or ‘ (apostrophes) and “ (quotation marks). The samples from this case are shown in Table 30.

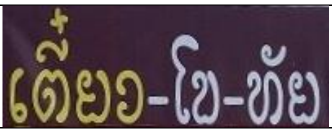
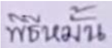
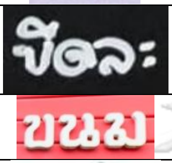
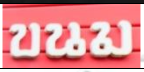
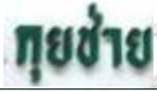
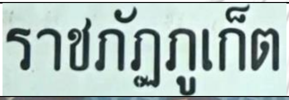
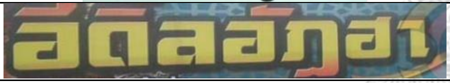
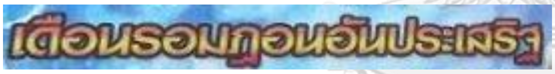
Table 30 Sample images from ambiguous special characters case.

Text Instance	Ground truth	Predicted Output
	091-509-6759.061-450-8691	091-509-6759**,**061-450-8691
	หจก.วี.ที.เอส.แทรกเตอร์	หจก.วี.ที.เอส**,**แทรกเตอร์
	เริ่ม 2.19-3.8ล้าน’	เริ่ม 2.19-3.8ล้าน**”**
	ไก่ตุ๋น**,**	ไก่ตุ๋น**.**
	จ.ราชบุรี**,**	จ.ราชบุรี**.**

Ambiguous Characters – Many Thai characters look very similar, such as “ฎ and ฏ”, “ข” and “บ”, “ด and ต”, and “ท and ฑ”. Even though the proposed global character loss can improve the recognition results in this case, some text instances, shown in

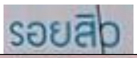
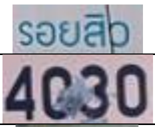
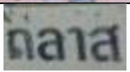
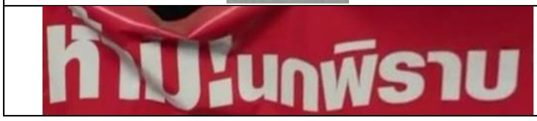
Table 31, are still difficult to distinguish by the proposed model.

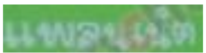
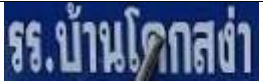
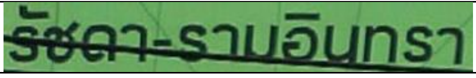
Table 31 Sample images from ambiguous Thai character case.

Text Instance	Ground truth	Predicted Output
	เตี๋ยว-โข-ทัย	เตี๋ยว-โ**บ**-ทัย
	พิธีหมั้น	**ฟิ**ธีหมั้น
	ขีดละ	**บี**ดละ
	ขนม	**บ**นม
	กุยช่าย	**ทุ**ยช่าย
	ราชภัฏภูเก็ต	ราชภั**ฎ**ภูเก็ต
	อีดิลอัฏฮา	อีดิลอั**ฎ**ฮา
	เดือนรอมฏอนอัน ประเสริฐ	เดือนรอม**ฎ**อนอัน ประเสริฐ

Partially occlusion and noise artifacts – The proposed method cannot handle text in some occlusion and disturbance cases, as shown in Table 32. Further study on sequence modeling may be able to improve the recognition output in this case.

Table 32 Sample images from partially occlusion case.

Text Instance	Ground truth	Predicted Output
	รอยสิว	รอยสิ**0**
	4030	40**8**0
	กลาส	**ถ**ลาส
	ห้ามนกพิราบ	ห้า**บ!**นกพิราบ

Text Instance	Ground truth	Predicted Output
	แพลนเน็ต	แพล**บ1**น็ต
	รร.บ้านโคกสง่า	รร.บ้านโ**ก**กสง่า
	รัชดา-รามอินทรา	**5**ช**ถ**า-**5**า**บ**อินทรา

Very long text instance – The proposed model does not explicitly handle very long text images. Further study is required on the sequence modeling and prediction methods. The sample of this case is shown in Figure 69.

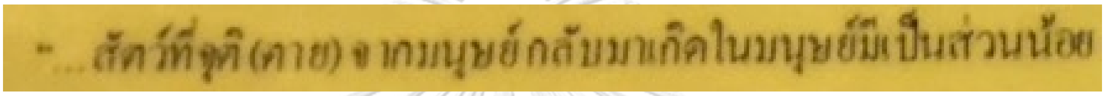


Figure 69 Very long Thai text instance sample.

5.2.5. Implementation Details

To the best of our knowledge, there is no standard Thai scene text dataset. Since deep learning generally requires a massive amount of labeled training data to be well trained, a method that generated close to real-world Thai scene text is proposed in this dissertation. The training data used by the proposed model is generated from the proposed Thai scene text generation method, as described in section 4.3.7. There were 12 million Thai text instances generated in Lighting Memory-Mapped Database (LMDB) format from the proposed algorithm. It took 20 hours to generate the entire dataset.

All trainable layers parameters are initialized by using Kaiming's initialization method (K. He et al., 2016). Ranger, which is an adaptive optimizer combing RAdam (Rectified Adam) (L. Liu et al., 2020) and LookAhead (M. R. Zhang et al., 2019), is used as an optimizer for this model. In the original seq2seq work (Bahdanau et al., 2015), warmup scheduling can improve model accuracy and generalization in the adaptive optimizer case. In this work, a cosine aneling proposed in (Goyal et al., 2018) is used as a warmup scheduler. The warmup stage learning rate was initially set to 10^{-7} and increased to 10^{-3} during the first 20k iterations. Then, the learning rate 10^{-3} is decayed by a factor of 10 at every 400k iterations until the deceased to 10^{-5}. The batch size was set to 32 and then increased to 64, 128, and 256 at 100k, 1M, 3M, and 8M iterations by using gradient accumulation. According to this work (Smith et al., 2018), increasing training batch sizes may slightly boost the model generalization and accuracy. The model is then trained until the overall loss is stable. For each experiment combination, the average training time until overall loss converged is about 5-6 days, which mostly depends on used feature extractor size and the number of multi-level attention layers.

All proposed models were implemented using PyTorch 1.3 and CUDA 10.1, trained and tested under the same hardware specification, which is NVIDIA GTX 1080 Ti and Intel i7-4770K with 16GB memory. Each configuration was run for 100 times and measure the average run time to ensure fair speed comparisons.

6. Conclusion

In this work, a novel Thai scene text localization and text recognition method is presented. This work can be divided into two main sub-problems: scene text localization and text recognition. The method begins with text detection by using the proposed multi-languages supported scene text localization algorithm. Each detected text instance is extracted and fed into Thai scene text recognition to recognize the input region into text transcription.

For the scene text localization section, a pixel-based scene text localization algorithm is proposed. The proposed text localization pipeline was inspired from semantic segmentation, which is a long-standing problem in the computer vision field. An extended text representation based on text and border masks is also proposed to support arbitrary text shapes. ResNet-50 combined with feature pyramid network is used as a backbone to extract multi-scaled features from the input image. Each scaled feature is combined and resized into the original size by using consecutive upscaling modules, enhancing the output segmentation results. To inference the text instances in each scale, connected component analysis is used along with the text border map to ensure a distinct area between each text instance. Finally, polygon-based non-maxima suppression is used on all scaled text instances, obtaining the output text instances. The contributions for this section can be shown as follows:

- A new text representation that can express arbitrary shape text instance is proposed. The proposed text representation uses the text pixel masks to represent the text instances and incorporates the offset masks and text instance border, which helps the distinguishing of adjacent text instances.
- The backbone networks for scene text localization are optimized for both accuracy and speed, depending on applications.
- A post-processing method, which helps the predicted output yield higher accuracy while marginally impact the overall inference time, was proposed.

- The qualitative experiments on standard benchmark and Thai scene text datasets show competitive results in terms of inference speed while still preserving acceptable accuracy.

To further improve the proposed scene text localization method, further study on the cause of failed cases, for example, very long text instances and small text, may provide essential clues for better text localization algorithm. Besides, further study and experiment on a single-stage method for end-to-end scene text localization and text recognition can lead to better practical, real-world applications.

After the text instances from scene text localization are retrieved, each cropped text instance is fed into the proposed scene text recognition method, which can be divided into four stages: transformation, feature extraction, sequence modeling, and prediction. The transformation stage normalizes the cropped text instance images into more appropriate shapes for recognition. Then, the visual feature is extracted from normalized text image by using ResNet incorporate with Feature Pyramid Network (FPN), which dramatically represents text in complex cases such as heavy background clutter and small character components. Since the extracted visual feature still lacks useful contextual information, the visually contextual feature is extracted in the sequence modeling stage. This contextual feature is then aligned and passed into the final prediction stage. In the prediction stage, a multi-level attention mechanism designed explicitly for multi-level writing style languages is used to predict the character sequence.

As Thai writing system, the global character context loss, which can slightly improve the overall accuracy in case of occlusion and ambiguous characters, is also incorporated to enhance global context into the predicted transcription. From the presented Thai scene text recognition, the contribution can be concluded as follows

- The specialty design multi-scaled network structure for Thai scene text recognition, capable of handling and representing small Thai character components.

- The multi-level attention mechanism can effectively transcript multi-level language word image like Thai into text transcription
- The novel global character context loss can capture the global context to improve overall recognition accuracy in challenging cases such as occlusion and ambiguous characters.
- The qualitative experiments on Thai scene text datasets show excellent performance in terms of both recognition accuracy and inference speed without any post-processing technique.
- The series of ablation studies show the effectiveness of the proposed pipeline in various configurations.

To further improve the accuracy of proposed Thai scene text recognition method, further study on the new seq2seq paradigms such as Transformer (Vaswani et al., 2017), Reformer (Kitaev et al., 2020), and Linformer (S. Wang et al., 2020) may be able to help archive better performance in term of both accuracy and computation efficiency. Furthermore, Natural Language Processing (NLP) can be used in the post-processing stage to improve recognition accuracy.

REFERENCES

Abraham, N., & Khan, N. M. (2019). A Novel Focal Tversky Loss Function With Improved Attention U-Net for Lesion Segmentation. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 683–687. https://doi.org/10.1109/ISBI.2019.8759329

Alsharif, O., & Pineau, J. (2013). End-to-end text recognition with hybrid HMM maxout models. *ArXiv Preprint ArXiv:1310.1811*.

Arbeláez, P., Maire, M., Fowlkes, C., & Malik, J. (2011). Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(5), 898–916. https://doi.org/10.1109/TPAMI.2010.161

Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., & Lee, H. (2019). What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4714–4722. https://doi.org/10.1109/ICCV.2019.00481

Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character Region Awareness for Text Detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9357–9366. https://doi.org/10.1109/CVPR.2019.00959

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *ICLR 2015 : International Conference on Learning Representations 2015*.

Bissacco, A., Cummins, M., Netzer, Y., & Neven, H. (2013). *PhotoOCR: Reading Text in Uncontrolled Conditions*. 785–792. https://doi.org/10.1109/ICCV.2013.102

Borisyuk, F., Gordo, A., & Sivakumar, V. (2018). Rosetta: Large Scale System for Text Detection and Recognition in Images. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 71–79.

Bouman, K. L., Abdollahian, G., Boutin, M., & Delp, E. J. (2011). A Low Complexity Sign Detection and Text Localization Method for Mobile Applications. *IEEE Transactions on Multimedia*, *13*(5), 922–934. https://doi.org/10.1109/TMM.2011.2154317

Budsayaplakorn, R., Asdornwised, W., & Jitapunkul, S. (2003). *On-line Thai handwritten character recognition using hidden Markov model and fuzzy logic*. 537–546. https://doi.org/10.1109/NNSP.2003.1318053

Bušta, M., Neumann, L. aš, & Matas, J. i. (2015). FASText: Efficient Unconstrained Scene Text Detector. *2015 IEEE International Conference on Computer Vision (ICCV 2015)*, 1206–1214. https://doi.org/10.1109/ICCV.2015.143

Buta, M., Neumann, L., & Matas, J. (2015). *FASText: Efficient Unconstrained Scene Text Detector*. 1206–1214. https://doi.org/10.1109/ICCV.2015.143

Campos, T. E. de, Babu, B. R., & Varma, M. (2009). CHARACTER RECOGNITION IN NATURAL IMAGES. *International Conference on Computer Vision Theory and Applications*, 273–280.

Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-8*(6), 679–698. https://doi.org/10.1109/TPAMI.1986.4767851

Chaivatna, S., & Supachai, T. (2010). *Recognizing broken characters in Thai Historical documents*. *1*, V1-99-V1-103. https://doi.org/10.1109/ICACTE.2010.5579053

Chamchong, R., Gao, W., & McDonnell, M. D. (2019). Thai Handwritten Recognition on Text Block-Based from Thai Archive Manuscripts. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1346–1351. https://doi.org/10.1109/ICDAR.2019.00217

Chen, K., Yin, F., & Liu, C. L. (2016). *Effective Candidate Component Extraction for Text Localization in Born-Digital Images by Combining Text Contours and Stroke Interior Regions*. 352–357. https://doi.org/10.1109/DAS.2016.30

Ch’ng, C. K., & Chan, C. S. (2017). Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition. *14th IAPR International Conference on Document Analysis and Recognition ICDAR*, 935–942. https://doi.org/10.1109/ICDAR.2017.157

Cho, H., Sung, M., & Jun, B. (2016). *Canny Text Detector: Fast and Robust Scene Text Localization Algorithm*. 3566–3573. https://doi.org/10.1109/CVPR.2016.388

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv:1406.1078 [Cs, Stat]*. http://arxiv.org/abs/1406.1078

Chomphuwiset, P. (2017). Printed thai character segmentation and recognition. *2017 IEEE 4th International Conference on Soft Computing Machine Intelligence (ISCMI)*, 123–127. https://doi.org/10.1109/ISCMI.2017.8279611

Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Tao, W., & Ng, A. Y. (2011). Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. *11th International Conference on Document Analysis and Recognition, 2011. (ICDAR2011)*, 440–445. https://doi.org/10.1109/ICDAR.2011.95

D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, & E. Valveny. (2015). ICDAR 2015 competition on Robust Reading. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 1156–1160. https://doi.org/10.1109/ICDAR.2015.7333942

Deng, D., Liu, H., Li, X., & Cai, D. (2018). PixelLink: Detecting scene text via instance segmentation. *Thirty-Second AAAI Conference on Artificial Intelligence*.

Emsawas, T., & Kijsirikul, B. (2016). Thai Printed Character Recognition Using Long Short-Term Memory and Vertical Component Shifting. In R. Booth & M.-L. Zhang (Eds.), *PRICAI 2016: Trends in Artificial Intelligence: 14th Pacific Rim International Conference on Artificial Intelligence, Phuket, Thailand, August 22-26, 2016, Proceedings* (pp. 106–115). Springer International Publishing. https://doi.org/10.1007/978-3-319-42911-3_9

Epshtein, B., Ofek, E., & Wexler, Y. (2010). Detecting text in natural scenes with stroke width transform. *IEEE Conference on Computer Vision and Pattern Recognition, 2012. (CVPR2012)*, 2963–2970. https://doi.org/10.1109/CVPR.2010.5540041

Feng, Y., Song, Y., & Zhang, Y. (2015). *Scene text localization using extremal regions and Corner-HOG feature*. 881–886. https://doi.org/10.1109/ROBIO.2015.7418882

Fukue, K., Tomokiyo, D., & Tangtisanon, P. (2017). Offline handwriting identification system for Thai characters using individual change control processing. *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 123–126.

https://doi.org/10.1109/ECTICon.2017.8096188

Gllavata, J., Ewerth, R., & Freisleben, B. (2004). Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. *17th International Conference on Pattern Recognition, 2004. (ICPR2004)*, *1*, 425–428 Vol.1. https://doi.org/10.1109/ICPR.2004.1334146

Gómez, L., & Karatzas, D. (2014). *MSER-Based Real-Time Text Detection and Tracking*. 3110–3115. https://doi.org/10.1109/ICPR.2014.536

Gomez, L., & Karatzas, D. (2013). Multi-script Text Extraction from Natural Scenes. *12th International Conference on Document Analysis and Recognition , 2013. (ICDAR2013)*, 467–471. https://doi.org/10.1109/ICDAR.2013.100

Gómez, L., & Karatzas, D. (2015). *Object proposals for text extraction in the wild*. 206–210. https://doi.org/10.1109/ICDAR.2015.7333753

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., & He, K. (2018). Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *ArXiv:1706.02677 [Cs]*. http://arxiv.org/abs/1706.02677

Gupta, A., Vedaldi, A., & Zisserman, A. (2016). Synthetic Data for Text Localisation in Natural Images. *IEEE Conference on Computer Vision and Pattern Recognition*.

Hanif, S. M., & Prevost, L. (2009). Text Detection and Localization in Complex Scene Images using Constrained AdaBoost Algorithm. *10th International Conference on Document Analysis and Recognition, 2009. (ICDAR2009)*, 1–5. https://doi.org/10.1109/ICDAR.2009.172

Hanif, S. M., Prevost, L., & Negri, P. A. (2008). A cascade detector for text detection in natural scene images. *19th International Conference on Pattern Recognition,2008 (ICPR2008)*, 1–4. https://doi.org/10.1109/ICPR.2008.4761536

Hashemi, S. R., Salehi, S. S. M., Erdogmus, D., Prabhu, S. P., Warfield, S. K., & Gholipour, A. (2019). Asymmetric Loss Functions and Deep Densely Connected Networks for Highly Imbalanced Medical Image Segmentation: Application to Multiple Sclerosis Lesion Detection. *IEEE Access*, *7*, 1721–1735. https://doi.org/10.1109/ACCESS.2018.2886371

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

https://doi.org/10.1109/ICCV.2017.322

He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. 770–778. https://doi.org/10.1109/CVPR.2016.90

He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., & Li, X. (2017). Single Shot Text Detector with Regional Attention. *2017 IEEE International Conference on Computer Vision (ICCV)*, 3066–3074. https://doi.org/10.1109/ICCV.2017.331

He, Pan, Huang, W., Qiao, Y., Loy, C. C., & Tang, X. (2016). *Reading scene text in deep convolutional sequences*. 3501–3508.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, W., Qiao, Y., & Tang, X. (2014). Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV* (pp. 497–511). Springer International Publishing. https://doi.org/10.1007/978-3-319-10593-2_33

Huang, X., & Ma, H. (2010). Automatic Detection and Localization of Natural Scene Text in Video. *20th International Conference on Pattern Recognition,2010 (ICPR2010)*, 3216–3219. https://doi.org/10.1109/ICPR.2010.786

Iamsa-at, S., & Horata, P. (2013). Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Network. *International Conference on IT Convergence and Security, 2013. (ICITCS2013)*, 1–5. https://doi.org/10.1109/ICITCS.2013.6717840

Initiative (WAI), W. W. A. (2020, July 5). *Web Content Accessibility Guidelines (WCAG) Overview*. Web Accessibility Initiative (WAI). https://www.w3.org/WAI/standards-guidelines/wcag/

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of The 32nd International Conference on Machine Learning*, 448–456.

Iqbal, K., Yin, X. C., Hao, H. W., Asghar, S., & Ali, H. (2014). *Bayesian network scores based text localization in scene images*. 2218–2225. https://doi.org/10.1109/IJCNN.2014.6889731

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep structured output learning for unconstrained text recognition. *ArXiv Preprint ArXiv:1412.5903*.

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading Text in the Wild with Convolutional Neural Networks. *International Journal of Computer Vision*, *116*(1), 1–20. https://doi.org/10.1007/s11263-015-0823-z

Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). Spatial transformer networks. *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2017–2025.

Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., & Luo, Z. (2017). *R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection*.

Jiang, Y., Zhu, X., Wang, X., Yang, S., Li, W., Wang, H., Fu, P., & Luo, Z. (2018). R2 CNN: Rotational Region CNN for Arbitrarily-Oriented Scene Text Detection. *2018 24th International Conference on Pattern Recognition (ICPR)*, 3610–3615. https://doi.org/10.1109/ICPR.2018.8545598

Jirattitichareon, W., & Chalidabhongse, T. H. (2006). Automatic Detection and Segmentation of Text in Low Quality Thai Sign Images. *IEEE Asia-Pacific Conference on Circuits and Systems, 2006. (APCCAS2006)*, 1000–1003. https://doi.org/10.1109/APCCAS.2006.342256

Kai, W., Babenko, B., & Belongie, S. (2011). *End-to-end scene text recognition*. 1457–1464. https://doi.org/10.1109/ICCV.2011.6126402

Kang, L., Li, Y., & Doermann, D. (2014). *Orientation Robust Text Line Detection in Natural Images*. 4034–4041. https://doi.org/10.1109/CVPR.2014.514

Kang, Le, Li, Y., & Doermann, D. (2013). Orientation Robust Text Line Detection in Natural Images. *\ldots of the IEEE Conference on Computer \ldots*. http://www.cv-foundation.org/openaccess/content_cvpr_2014/html/Kang_Orientation_Robust_Text_2014_CVPR_paper.html

Kaothanthong, N., Theeramunkong, T., & Chun, J. (2017). Improving Thai Optical Character Recognition Using Circular-Scan Histogram. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, *01*, 567–572. https://doi.org/10.1109/ICDAR.2017.98

Karaoglu, S., Fernando, B., & Trémeau, A. (2010). A Novel Algorithm for Text Detection and Localization in Natural Scene Images. *International Conference on Digital Image Computing: Techniques and Applications, 2010. (DICTA2010)*, 635–642. https://doi.org/10.1109/DICTA.2010.115

Kijsirikul, B., Sinthupinyo, S., & Supanwansa, A. (1998). *Thai printed character recognition by combining inductive logic programming with backpropagation neural network*. 539–542. https://doi.org/10.1109/APCCAS.1998.743876

Kingma, D. P., & Ba, J. L. (2015). Adam: A Method for Stochastic Optimization. *ICLR 2015 : International Conference on Learning Representations 2015*.

Kitaev, N., Kaiser, L., & Levskaya, A. (2020). Reformer: The Efficient Transformer. *ICLR 2020 : Eighth International Conference on Learning Representations*.

Kobchaisawat, T., & Chalidabhongse, T. H. (2015, December). A Method for Multi-Oriented Thai Text Localization in Natural Scene Images using Convolutional Neural Network. *4th IEEE International Conference on Signal and Image Processing Applications, 2015. (ICSIPA 2015)*.

Kobchaisawat, T., & Chalidabhongse, T. H. (2014). Thai text localization in natural scene images using Convolutional Neural Network. *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2014. (APSIPA2014)*, 1–7. https://doi.org/10.1109/APSIPA.2014.7041775

Koo, H. I., & Kim, D. H. (2013). Scene Text Detection via Connected Component Clustering and Nontext Filtering. *IEEE Transactions on Image Processing*, *22*(6), 2296–2305. https://doi.org/10.1109/TIP.2013.2249082

Lee, S., Cho, M. S., Jung, K., & Kim, J. H. (2010). *Scene Text Extraction with Edge Constraint and Text Collinearity*. 3983–3986. https://doi.org/10.1109/ICPR.2010.969

Li, Y., Yu, Y., Li, Z., Lin, Y., Xu, M., Li, J., & Zhou, X. (2018). Pixel-Anchor: A Fast Oriented Scene Text Detector with Combined Networks. *ArXiv:1811.07432 [Cs]*. http://arxiv.org/abs/1811.07432

Liao, M., Shi, B., Bai, X., Wang, X., & Liu, W. (2017). TextBoxes: A Fast Text Detector with a Single Deep Neural Network. *AAAI*.

Liao, M., Zhu, Z., Shi, B., Xia, G., & Bai, X. (2018). Rotation-Sensitive Regression for Oriented Scene Text Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5909–5918. https://doi.org/10.1109/CVPR.2018.00619

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007. https://doi.org/10.1109/ICCV.2017.324

Liu, J., Liu, X., Sheng, J., Liang, D., Li, X., & Liu, Q. (2019). Pyramid Mask Text Detector. *ArXiv:1903.11800 [Cs]*. http://arxiv.org/abs/1903.11800

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., & Han, J. (2020). On the Variance of the Adaptive Learning Rate and Beyond. *ArXiv:1908.03265 [Cs, Stat]*. http://arxiv.org/abs/1908.03265

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 21–37). Springer International Publishing.

Liu, W., Chen, C., Wong, K.-Y. K., Su, Z., & Han, J. (2016). STAR-Net: A SpaTial attention residue network for scene text recognition. *British Machine Vision Conference 2016*.

Liu, X., Kawanishi, T., Wu, X., & Kashino, K. (2016). *Scene text recognition with CNN classifier and WFST-based word labeling*. 3999–4004. https://doi.org/10.1109/ICPR.2016.7900259

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

Long, S., Ruan, J., Zhang, W., He, X., Wu, W., & Yao, C. (2018, September). TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. *The European Conference on Computer Vision (ECCV)*.

Lucas, S. M., Panaretos, A., Sosa, L., Tang, A., Wong, S., & Young, R. (2003). ICDAR 2003 robust reading competitions. *7th International Conference on Document Analysis and Recognition, 2003. (ICDAR2003)*, *1*, 682–687. https://doi.org/10.1109/ICDAR.2003.1227749

Luo, C., Jin, L., & Sun, Z. (2019). MORAN: A Multi-Object Rectified Attention Network for scene text recognition. *Pattern Recognition*, *90*, 109–118.

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., & Xue, X. (2018). Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE Transactions on Multimedia*, *20*(11), 3111–3122.

Meng, Q., & Song, Y. (2012). Text Detection in Natural Scenes with Salient Region. *10th IAPR International Workshop on Document Analysis Systems, 2012. (DAS2012)*, 384–388. https://doi.org/10.1109/DAS.2012.85

Methasate, I, Marukatat, S., Sae-tang, S., & Theeramunkong, T. (2005). The feature combination technique for off-line Thai character recognition system. *8th International Conference on Document Analysis and Recognition, 2005. (ICDAR2005)*, 1006–1009 Vol. 2. https://doi.org/10.1109/ICDAR.2005.236

Methasate, I, & Sae-tang, S. (2004). The clustering technique for Thai handwritten recognition. *9th International Workshop on Frontiers in Handwriting Recognition, 2004. (IWFHR2004)*, 450–454. https://doi.org/10.1109/IWFHR.2004.101

Methasate, Ithipan, & Marukatat, S. (2013). BEST 2013: Thai Printed Character Recognition Competition. *10th Symposium on Natural Language Processing, 2013. (SNLP-2013)*.

Milletari, F., Navab, N., & Ahmadi, S. (2016). V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, 565–571. https://doi.org/10.1109/3DV.2016.79

Minghui Liao, B. S., & Bai, X. (2018). TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Transactions on Image Processing*, *27*(8), 3676–3690. https://doi.org/10.1109/TIP.2018.2825107

Minhua, L., & Chunheng, W. (2008). *An adaptive text detection approach in images and video frames*. 72–77. https://doi.org/10.1109/IJCNN.2008.4633769

Mishra, A, Alahari, K., & Jawahar, C. V. (2012). Top-down and bottom-up cues for scene text recognition. *IEEE Conference on Computer Vision and Pattern Recognition,* *2012. (CVPR2012)*, 2687–2694. https://doi.org/10.1109/CVPR.2012.6247990

Mishra, Anand, Alahari, K., & Jawahar, C. V. (2009). Scene Text Recognition using Higher Order Language Priors. *British Machine Vision Conference 2009*, 1–11.

Mitatha, S., Dejharn, K., Chevasuvit, F., Chankuang, B., & Kasemsiri, W. (2001). *Experimental results of using rough sets for printed Thai characters recognition*. *1*, 331–334 vol.1. https://doi.org/10.1109/TENCON.2001.949608

Mitrpanont, J. L., & Imprasert, Y. (2011). Thai handwritten character recognition using heuristic rules hybrid with neural network. *8th International Joint Conference on Computer Science and Software Engineering, 2008. (JCSSE2008)*, 160–165. https://doi.org/10.1109/JCSSE.2011.5930113

Mosleh, A., Bouguila, N., & Hamza, A. B. (2012). *Image Text Detection Using a Bandlet-Based Edge Detector and Stroke Width Transform*. 2012 British Machine Vision Conference. https://doi.org/10.5244/C.26.63

Nayef, N., Yin, F., Bizid, I., Choi, H., Feng, Y., Karatzas, D., Luo, Z., Pal, U., Rigaud, C., Chazalon, J., Khlif, W., Luqman, M. M., Burie, J., Liu, C., & Ogier, J. (2017). ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification—RRC-MLT. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, *01*, 1454–1459. https://doi.org/10.1109/ICDAR.2017.237

Nayef, Nibal, Patel, Y., Busta, M., Chowdhury, P. N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.-C., Liu, C., & Ogier, J.-M. (2019). ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition—RRC-MLT-2019. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1582–1587. https://doi.org/10.1109/ICDAR.2019.00254

Neumann, L., & Matas, J. (2015). Efficient Scene text localization and recognition with local character refinement. *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 746–750. https://doi.org/10.1109/ICDAR.2015.7333861

Neumann, Lukas, & Matas, J. (2011). A Method for Text Localization and Recognition in Real-World Images. In R. Kimmel, R. Klette, & A. Sugimoto (Eds.), *Computer Vision – ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown,* *New Zealand, November 8-12, 2010, Revised Selected Papers, Part III* (pp. 770–783). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-19318-7_60

Neumann, Lukas, & Matas, J. (2010). A Method for Text Localization and Recognition in Real-world Images. *Computer Vision – ACCV 2010*, 770–783. https://doi.org/10.1007/978-3-642-19318-7_60

Neumann, Lukas, & Matas, J. (2012). Real-time scene text localization and recognition. *IEEE Conference on Computer Vision and Pattern Recognition, 2012. (CVPR2012)*, 3538–3545. https://doi.org/10.1109/CVPR.2012.6248097

Open Government Data of Thailand. (2020, July 5). https://data.go.th/

Pan, Y. F., Hou, X., & Liu, C. L. (2011). A Hybrid Approach to Detect and Localize Texts in Natural Scene Images. *IEEE Transactions on Image Processing*, *20*(3), 800–813. https://doi.org/10.1109/TIP.2010.2070803

Pan, Y.-F., Hou, X., & Liu, C.-L. (2008). A Robust System to Detect and Localize Texts in Natural Scene Images. *8th IAPR International Workshop on Document Analysis Systems, 2008. (DAS2008)*, 35–42. https://doi.org/10.1109/DAS.2008.42

Pan, Y.-F., Hou, X., & Liu, C.-L. (2009). Text Localization in Natural Scene Images Based on Conditional Random Field. *10th International Conference on Document Analysis and Recognition, 2009. (ICDAR2009)*, 6–10. https://doi.org/10.1109/ICDAR.2009.97

Pan, Y.-F., Liu, C.-L., & Hou, X. (2010). Fast scene text localization by learning-based filtering and verification. *17th IEEE International Conference on Image Processing, 2010. (ICIP2010)*, 2269–2272. https://doi.org/10.1109/ICIP.2010.5651862

Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson Image Editing. *ACM Trans. Graph.*, *22*(3), 313–318. https://doi.org/10.1145/882262.882269

Pornchaikajornsak, A., & Thammano, A. (2003). *Handwritten Thai character recognition using fuzzy membership function and fuzzy ARTMAP*. *1*, 40–44 vol.1. https://doi.org/10.1109/CIRA.2003.1222060

Qin, S., & Manduchi, R. (2016). *A fast and robust text spotter*. 1–8. https://doi.org/10.1109/WACV.2016.7477663

Ray, A., Shah, A., & Chaudhury, S. (2016). *Recognition based text localization from* *natural scene images*. 1177–1182. https://doi.org/10.1109/ICPR.2016.7899796

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv Preprint ArXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Roongroj, N., & Povey, D. (2003). *Discriminative training for HMM-based offline handwritten character recognition*. 114–118 vol.1. https://doi.org/10.1109/ICDAR.2003.1227643

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Sa-ngamuang, P., Thamnittasana, C., & Kondo, T. (2007). *Thai car license plate recognition using essential-elements-based method*. 41–44. https://doi.org/10.1109/APCC.2007.4433496

Shi, B., Bai, X., & Belongie, S. (2017). Detecting Oriented Text in Natural Images by Linking Segments. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3482–3490. https://doi.org/10.1109/CVPR.2017.371

Shi, B., Bai, X., & Yao, C. (2017). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(11), 2298–2304.

Shi, B., Wang, X., Lyu, P., Yao, C., & Bai, X. (2016). Robust Scene Text Recognition with Automatic Rectification. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4168–4176.

Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., & Bai, X. (2019). ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(9), 2035–2048. https://doi.org/10.1109/TPAMI.2018.2848939

Shivakumara, P., Phan, T. Q., & Tan, C. L. (2009). *A Gradient Difference Based Technique* *for Video Text Detection*. 156–160. https://doi.org/10.1109/ICDAR.2009.85

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training Region-Based Object Detectors with Online Hard Example Mining. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 761–769. https://doi.org/10.1109/CVPR.2016.89

Siriteerakul, T. (2013). Mixed Thai-English Character Classification Based on Histogram of Oriented Gradient Feature. *12th International Conference on Document Analysis and Recognition, 2013. (ICDAR2013)*, 847–851. https://doi.org/10.1109/ICDAR.2013.173

Smith, S. L., Kindermans, P.-J., & Le, Q. V. (2018). Don't Decay the Learning Rate, Increase the Batch Size. *ICLR 2018 : International Conference on Learning Representations 2018*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Su, F., & Xu, H. (2015). *Robust seed-based stroke width transform for text detection in natural images*. 916–920. https://doi.org/10.1109/ICDAR.2015.7333895

Subramanian, K., Natarajan, P., Decerbo, M., & Castanon, D. (2007). Character-Stroke Detection for Text-Localization and Extraction. *9th International Conference on Document Analysis and Recognition, 2007. (ICDAR2007)*, *1*, 33–37. https://doi.org/10.1109/ICDAR.2007.4378671

Tangwongsan, S., & Jungthanawong, O. (2008). A refinement of stroke structure for printed Thai character recognition. *9th International Conference on Signal Processing, 2008. (ICSP2008)*, 1504–1507. https://doi.org/10.1109/ICOSP.2008.4697418

Tangwongsan, S., & Suvacharakulton, B. (2012). OCR with Word Prediction Technique for Bilingual Documents. *11th International Conference on Computer and Information Science, 2012. (ICIS2012)*, 343–347. https://doi.org/10.1109/ICIS.2012.77

Tanprasert, C., & Koanantakool, T. (1996). Thai OCR: a neural network application. *IEEE TENCON. Digital Signal Processing Applications, 1996. (TENCON1996)*, *1*, 90–95 vol.1. https://doi.org/10.1109/TENCON.1996.608717

Thammano, A., & Duangphasuk, P. (2005). Printed Thai character recognition using the hierarchical cross-correlation ARTMAP. *17th IEEE International Conference on Tools with Artificial Intelligence, 2005. (ICTAI2005)*, 4 pp.–698. https://doi.org/10.1109/ICTAI.2005.100

Thongkamwitoon, T., Asdornwised, W., Aramvith, S., & Jitapunkul, S. (2002). *On-line Thai-English handwritten character recognition using distinctive features*. *2*, 259–264 vol.2. https://doi.org/10.1109/APCCAS.2002.1115220

Tian, Z., Huang, W., He, T., He, P., & Qiao, Y. (2016). Detecting Text in Natural Image with Connectionist Text Proposal Network. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016* (pp. 56–72). Springer International Publishing.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5998–6008.

Vel, O. de, Wangsuya, S., & Coomans, D. (1995). *On Thai character recognition*. *4*, 2095–2098 vol.4. https://doi.org/10.1109/ICNN.1995.488999

Wang, Q., Gao, J., Zhang, M., Xing, J., & Hut, W. (2018). SPCNet: Scale Position Correlation Network for End-to-End Visual Tracking. *2018 24th International Conference on Pattern Recognition (ICPR)*, 1803–1808. https://doi.org/10.1109/ICPR.2018.8545053

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. *ArXiv Preprint ArXiv:2006.04768*.

Wang, T., Wu, D. J., Coates, A., & Ng, A. Y. (2012). End-to-end text recognition with convolutional neural networks. *21st International Conference on Pattern Recognition,2012 (ICPR2012)*, 3304–3308.

Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., & Shao, S. (2019, June). Shape Robust Text Detection With Progressive Scale Expansion Network. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Watjanapong, K., & Chom, K. (2001). *Printed Thai character recognition using fuzzy-rough sets*. *1*, 326–330 vol.1. https://doi.org/10.1109/TENCON.2001.949607

Wiwatcharakoses, C., & Patanukhom, K. (2015). *MSER based text localization for multi-language using double-threshold scheme*. 62–71. https://doi.org/10.4108/icst.iniscom.2015.258413

Wiwatcharakoses, C., & Patanukhom, K. (2013). Two-Stage Recognition for Printed Thai and English Characters Using Nearest Neighbor and Support Vector Machine. *International Conference on Signal-Image Technology Internet-Based Systems, 2013. (SITIS2013)*, 71–78. https://doi.org/10.1109/SITIS.2013.23

Wararatpanya, K., Boonchukusol, P., Kuroki, Y., & Kato, Y. (2013). Improved Thai text detection from natural scenes. *International Conference on Information Technology and Electrical Engineering, 2013. (ICITEE2013)*, 137–142. https://doi.org/10.1109/ICITEED.2013.6676227

Wararatpanya, K., Pasupa, K., Suttapakti, U., Boonchukusol, P., Titijaroonroj, T., Hokking, R., Kuroki, Y., & Kato, Y. (2014). Text-background decomposition for thai text localization and recognition in natural scenes. *Information Technology and Electrical Engineering, 2014. (ICITEE 2014)*, 1–6. https://doi.org/10.1109/ICITEED.2014.7007914

X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, & J. Liang. (2017). EAST: An Efficient and Accurate Scene Text Detector. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2642–2651. https://doi.org/10.1109/CVPR.2017.283

Xiangrong, C., & Yuille, A. L. (2004). *Detecting and reading text in natural scenes*. *2*, II-366-II-373 Vol.2. https://doi.org/10.1109/CVPR.2004.1315187

Xie, E., Zang, Y., Shao, S., Yu, G., Yao, C., & Li, G. (2019). Scene Text Detection with Supervised Pyramid Context Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, *33*, 9038–9045. https://doi.org/10.1609/aaai.v33i01.33019038

Xie, S., & Tu, Z. (2015). Holistically-Nested Edge Detection. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1395–1403. https://doi.org/10.1109/ICCV.2015.164

Xing, D., Li, Z., Chen, X., & Fang, Y. (2017). ArbiText: Arbitrary-Oriented Text Detection in Unconstrained Scene. *ArXiv:1711.11249 [Cs]*. http://arxiv.org/abs/1711.11249

Yin, X., Yin, X. C., Hao, H. W., & Iqbal, K. (2012). *Effective text localization in natural scene images with MSER, geometry-based grouping and AdaBoost*. 725–728.

Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., & Ding, X. (2019). Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10544–10553.

Zhang, M. R., Lucas, J., Hinton, G., & Ba, J. (2019). Lookahead Optimizer: K steps forward, 1 step back. *ArXiv:1907.08610 [Cs, Stat]*. http://arxiv.org/abs/1907.08610

Zhang, S., Liu, Y., Jin, L., & luo, canjie. (2018). Feature Enhancement Network: A Refined Scene Text Detector. *AAAI-18 AAAI Conference on Artificial Intelligence*, 2612–2619.

Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., & Bai, X. (2016). *Multi-oriented Text Detection with Fully Convolutional Networks*. 4159–4167. https://doi.org/10.1109/CVPR.2016.451

Zhu, Z., Liao, M., Shi, B., & Bai, X. (2018). Feature Fusion for Scene Text Detection. *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 193–198. https://doi.org/10.1109/DAS.2018.60

Zitnick, C. L., & Dollár, P. (2014). Edge Boxes: Locating Object Proposals from Edges. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *13th European Conference, 2014. (ECCV 2014)* (pp. 391–405). Springer International Publishing. https://doi.org/10.1007/978-3-319-10602-1_26

Zongyi, L., & Sarkar, S. (2008). *Robust outdoor text detection using text intensity and shape features*. 1–4. https://doi.org/10.1109/ICPR.2008.4761432

ระบบคลังสื่อประสมและข้อความกำกับ. (2020, July 5). https://www.nectec.or.th/corpus/

VITA

NAME	Thananop Kobchaisawat
DATE OF BIRTH	15 October 1988
PLACE OF BIRTH	Bangkok
INSTITUTIONS ATTENDED	Chulalongkorn University
HOME ADDRESS	203 Surpa Rd. Prombprabsattupai, Bangkok
PUBLICATION	- Kobchaisawat Thananop, and Thanarat H. Chalidabhongse. “Scene Text Detection with Polygon Offsetting and Border Augmentation.” Electronics, vol. 9, no. 1, 2020, p. 117. - Wongta Pitchakorn, Kobchaisawat Thananop, and Thanarat H. Chalidabhongse. “An Automatic Bus Route Number Recognition.” 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016, pp. 1–6. - Metsiritrakul Kawin, Kobchaisawat Thananop, and Thanarat H. Chalidabhongse. “UP2U: Program for Raising Awareness of Phubbing Problem with Stimulating Social Interaction in Public Using Augmented Reality and Computer Vision.” 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016, pp. 1–6. - Kobchaisawat Thananop, and Thanarat H. Chalidabhongse. “A Method for Multi-Oriented Thai Text Localization in Natural Scene Images Using Convolutional

Neural Network.” 2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2015, pp. 220–225.

- Kobchaisawat Thananop, and Thanarat H. Chalidabhongse. “Thai Text Localization in Natural Scene Images Using Convolutional Neural Network.” Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, 2014, pp. 1–7.

AWARD RECEIVED

- 1st place in Benchmark for Enhancing the Standard for Thai language Processing (BEST) competitions (2015-2018) organized by NECTEC.
- 1st place in IT Princess Award in student research innovation category, organized by Foundation for Research in Information
Technology (FRIT)
- First runner up in Thailand ICT Award 2016 (TICTA) in research and innovation category.