Comparative analysis of Tesseract and Google Cloud Vision for Thai vehicle registration certificate

Karanrat Thammarak[1], Prateep Kongkla[1], Yaowarat Sirisathitkul[1], Sarun Intakosum[2]
[1]School of Engineering and Technology, Walailak University, Thai Buri, Thailand
[2]Department of Computer Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

Article Info	**ABSTRACT**
	Optical character recognition (OCR) is a technology to digitize a paper-based document to digital form. This research studies the extraction of the characters from a Thai vehicle registration certificate via a Google Cloud Vision API and a Tesseract OCR. The recognition performance of both OCR APIs is also examined. The 84 color image files comprised three image sizes/resolutions and five image characteristics. For suitable image type comparison, the greyscale and binary image are converted from color images. Furthermore, the three pre-processing techniques, sharpening, contrast adjustment, and brightness adjustment, are also applied to enhance the quality of image before applying the two OCR APIs. The recognition performance was evaluated in terms of accuracy and readability. The results showed that the Google Cloud Vision API works well for the Thai vehicle registration certificate with an accuracy of 84.43%, whereas the Tesseract OCR showed an accuracy of 47.02%. The highest accuracy came from the color image with 1024×768 px, 300dpi, and using sharpening and brightness adjustment as pre-processing techniques. In terms of readability, the Google Cloud Vision API has more readability than the Tesseract. The proposed conditions facilitate the possibility of the implementation for Thai vehicle registration certificate recognition system.

Corresponding Author:

Karanrat Thammarak
School of Engineering and Technology, Walailak University
222 Thaiburi, Thasala District, Nakhon Si Thammarat 80160, Thailand
Email: kanchan.th@mail.wu.ac.th

1. INTRODUCTION

Computer vision is a branch of artificial intelligence (AI) that trains a computer to analyze data from images [1], [2]. It has been used in many domains such as medicine [3], [4], education [5], banking [6], [7] logistics and transportation [8], and agriculture [9]. Moreover, this technology is being increasingly applied in intelligence systems, internet of things (IoT), cloud computing, robotics [10]–[12], and IoT-based vehicle verification systems [13].

Computer vision is also used for document processing systems, for extracting data from handwritten or printed text. Optical character recognition (OCR) plays an important role in extracting characters from paper-based documents. Integrating OCR into a document processing application increases the quality of data, efficiency, and speed, and reduces human error in extracting data [14]. OCR can also be used by the government to prevent crime and enforce the law. For example, it can be used in a passport identification system for immigration admission or a vehicle license plate system for detecting and controlling traffic offences [10]. Nowadays, the development of OCR applications [15], [16] is much more convenient because multiple tools, both free of charge and commercial, are available for their development, and these tools

support various native languages. Furthermore, each tool in the market is competing to develop APIs and libraries that will further improve their computation quality, such as Google Cloud Vision, Tesseract, Huawei Cloud Vision, matix laboratory-OCR (MATLAB-OCR), and OpenCV's EAST detector.

This study aims to compare the performance of the Google Cloud Vision API [17] and Tesseract [18] to recognize document images of Thai vehicle registration certificates to describe car information. Currently, Thai vehicle registration certificates are still in a hard-copy format. Government officers, police, or people who want to buy and sell vehicles cannot be automatically connected to the database. The user must import data to the system with key-ins that may be error-prone. Moreover, this paper studies existing pre-processing techniques to increase the accuracy of the performance of Google Cloud Vision and Tesseract.

The rest of the paper is organized as follows: section 2 presents a literature review of the relevant issues. Section 3 presents the research method, including the study of tools and techniques for enhancing the process quality and the design and development of the proposed system. Section 4 presents the results of the comparison and discussion. The last section presents conclusions and suggestions for future work.

2.   LITERATURE REVIEW

Computer vision uses deep learning to create neural networks that guide image processing and analysis [19]. Computer vision models can recognize objects, detect, and recognize people, and even track movement when receiving full training. All available tools used in computer vision include mathematics, geometry, linear algebra, statistics and operational research, functional analysis, and intelligent models. They are used to create algorithms for the segmentation and grouping of images so that computers can understand the scenery of their features. OCR is a process for recognizing and translating text that appears in images, documents, symbols, or marked scenes, which the computer can access and process as text characters. OCR can convert images containing text that are handwritten or printed [14]. OCR technology increases the data entry speed and reduces human error. In addition, OCR can improve the recovery and file handling performance of a storage device. Printed text is typewritten text from any device, such as a computer or typewriter. The OCR subtype targets to proceed with one character at a time using pattern matching and feature analysis. In contrast, intracranial response (ICR) may use machine-learning techniques to proceed with the text. Handwritten text is a handwritten input from any source, such as paper documents or photos. The offline mode is the process of a static document, whereas the online version uses handwriting movement analysis. The significant phases of OCR consist of the following six phases [19]–[22]:

− Optical scanning captures images from various sources, such as scanners or cameras.
− Pre-processing improves image quality. Examples of pre-processing techniques include binarization, sharpening, and image adjustments.
− Character segmentation passes the character to the recognition engine. The most straightforward techniques used for this are connected component analysis and projection profiles.
− Feature extraction extracts and recognizes different features in the image.
− Character classification maps features to different categories using classification techniques.
− Post-processing: After classification, the results are not 100% correct, especially for complex languages. Post-processing improves the accuracy of OCR systems. These techniques utilise natural language processing and geometric and linguistic contexts to correct errors in OCR results.

OCR has been used in many different domains. For example, Patel and Patel applied the Tesseract OCR and Transym OCR tools to consider vehicle number plates [23]. In another case related to an autonomous vehicle, Sugadev et al. [11] applied the Google Cloud Vision API to process images, improve the obstacle detection accuracy, and provide the capability to identify obstacles for rough terrain autonomous robots. Moreover, many domains use this technology to increase the data entry speed and reduce human error in extracting data from document processing systems [24], [25]. Moreover, such technology has been increasingly applied in intelligence systems, IoT, cloud computing, and robotics, such as signboard text translation [26] and IoT-based vehicle verification systems [27]. Since 1969, when the US Army introduced the first application to use OCR technology in IBM 360 computers, the development of OCR technology has become much more convenient than in the past because many development tools have been introduced to support the OCR developer. Today, tools support both offline and online use. Furthermore, each tool is competing to develop new APIs and libraries that will further improve computation quality. Such tools include Google Cloud Vision [17], Tesseract [18], MATLAB [28], and OpenCV [29]. Google Cloud Vision API is a Google Cloud platform service that leverages Google technology capabilities to facilitate image analysis. This service can help to understand image content by using the machine learning tools developed by Google. The necessary capabilities of the Google Cloud Vision API include the following [17]:

− Inspecting objects within the image (entity detection)
− Reading text within images (OCR)
− Checking for inappropriate images, such as porn, using the Google Safe Search
− Finding faces within the photo
− Checking the location within the image (landmark detection) and
− Checking a logo inside the image (logo detection)

The first step to detect text from an image using the Google Cloud Vision API is uploading the original image to cloud storage. A cloud function is triggered, which uses the vision API to extract the text and detect the source language. The text in each language can then be translated by publishing a message to pub/sub. A cloud function uses the translation API to translate the text in the translation queue and sends the results to the result queue. Then, another cloud function saves the translated text from the result queue to cloud storage.

The Tesseract is an open-source OCR tool that began as a PhD research project in HP labs to develop HP flatbed scanners [18]. The Tesseract is available under the license of Apache 2.0. It can be used directly, or an API can be used to extract printed and handwritten text from images. Tesseract supports more than 100 languages and both printed and handwritten texts. The Tesseract has several available third parties, including more than 40 partners, such as TesseractStudio.Net, PDF OCR X, and TaxWorkFlow. In addition, Tesseract is compatible with many programming languages, such as C, C++, and Python. Currently, the latest stable version of the Tesseract is 4.0. When the user inputs the original image to the process, the binarization algorithm will produce it to the outline and gather it together, purely by nesting, into Blobs. Recognition then proceeds into a two-part process. The recognition model attempts to recognize each word, and then the adaptive classifier recognizes the text lower down the page more accurately. In the second part, Tesseract 4.0 uses a convolutional neural network (CNN) to recognize an image containing a single character and provide recurrent neural networks (RNNs) and long short-term memory (LSTM) with a sequence of characters. A comparison of the basic features and architecture of the Tesseract and Google Cloud Vision API is shown in Table 1.

Table 1. Comparison of the basic features between the Tesseract and the Google Cloud Vision API

Feature	Tesseract	Google Cloud Vision API
Latest Stable Vision	4	Unknown
Technology	CNN, LSTM, VGSL	Unknown
Cost	Free	Pay per use
Software Type	Opensource	Cloud Service
Architecture	On Web/On Device	On Web
SDK	Yes	Yes
License	Apache	Proprietary
Supporting OS	macOS, Window, Unix, Linux	Browser
Encoder	Base 64	Base 64
Programming Language	C, C++, Python	C#, Go, Java, Node.js, PHP, Ruby, Python
Font	Printed and Handwritten Script	Printed and Handwritten Script
Supported Language	Printed=100+, Script=35+	Printed=200+, Script=200+
Thai Supporting	Printed and Handwritten Script	Printed and Handwritten Script
Third-Party	40+	Unknown
Supported Mobile Platform	Android, iOS	Android, iOS

Both the Tesseract and Google Cloud Vision API can recognize printed and handwritten text and support many programming languages and other third parties. Base 64 is the standard format of a data format that the Tesseract and Google Cloud Vision API use to encode data from image to text. Tesseract is open-source software; therefore, it cost little compared with the Google Cloud Vision API, a cloud service where the cost depends on usage. However, Tesseract can be more challenging to set up and configure on the server than the Google Cloud Vision API, which comes with ready-to-use environments. Moreover, the Google Cloud Vision API provides the interoperability of cloud storage, cloud function, and Google translation through pub/sub. Simultaneously, Tesseract works together through each internal module, which may significantly affect the processing time.

The Thai language elements will consist of 44 letters, 21 vowels and 32 sounds, and four tone marks. Moreover, Thai letters can be divided according to the writing style into five groups. The similarity of the letters makes it easy to recognize the Thai script shown in Table 2. In Thailand, computer vision has been used for more than ten years for scenarios such as citizen ID cards [30], translating Thai menus into Malay [31], and license plates [32]. Also, Thai character recognition is studied in various techniques, as shown in Table 3.

Table 2. The Thai letter, vowel, and tone mark

	Thai letter
Head-in Type	ง ฌ ญ ฒ ณ ด ต ถ ผ ฝ ย ล ว ส อ ฮ
Head-out type	ค ศ จ ฉ ฎ ฏ ฐ ท น บ ป พ ฟ ภ ม ร ศ ษ ห ฬ
Double head type	ข ช
Broken head	ข ช ฆ ฑ
Headless type	ก ธ
Thai vowel and sound	อะ / แอ / เออะ / อา / เอียะ / เออ / อิ / เอีย / อำ / อี / เอือ / ไอ / อึ / เอือ / ไอ / อื / อัวะ / เอา / อุ / อัว / ฤ / อู / โอะ / ฤๅ / เอะ / โอ / ฦ / เอ / เอาะ / ฦๅ / แอ / ออ
Thai tone marks	อ่ อ้ อ๊ อ๋

Table 3. The articles that study techniques of Thai language OCR and results

Article	Method	Result: Accuracy
Kraisin and Kaothanthong [33]	Method: Extreme learning machine and b=4 histogram Data: 45 Thai license plates	90%
Kitvimonrat and Watcharabutsarakham [34]	Method: Combining the EAST model and text recognition and using Bi-LSTM Data: Data set with 128x64 pixels Grayscale image of Thai license plates Error Character: ข-ช/ค-ด/ฌ-ณ and ก-ภ	28% to 90%
Kobchaisawat and Chalidabhongse [35]	Method: Convolutional neural networks (CNN). Data: Standard data sets with 640x480 pixels resolution.	Precision: 0.70, Recall: 0.73 F-Measure: 0.71
Sumetphong and Tangwongsan [36]	Method: MATLAB, an adaptive switch median filter and 2-D Otsu thresholding method. Data: Training-30 pages Testing-10 pages	93%
Thammano and Duangphasuk [37]	Method: the fuzzy ARTMAP neural network Data: 12 no-head fonts and eight experiments	82.42%
Somboonsak [38]	Method: Thai character clusters and longest matching Data: 15,000 characters	86.51%
Chomphuwiset [39]	Method: A convolution neural network (CNN). Data: 124,080 of training images 20,424 of testing images	98%
Pornpanomchai and Daveloh [40]	Method: A genetic algorithm. Data: 1,015 printed Thai characters.	97.14%
Tangwannawit and Saetang [41]	Method: The Tesseract OCR engine in a mobile application. Data: 50 tickets.	98.05% and 100%
Jirattitichareon and Chalidabhongse [42]	Method: Laplacian of Gaussian, and Gaussian mixture model. Data: 192 Thai sign images	90.22%
Toolpeng et al. [43]	Method: Service routine procedure. Data: 156 available samples.	MLP: 100% SVM: 85%
Duangphasuk and Thammano [44]	Method: the hierarchical cross-correlation ARTMAP. Data: 32 experiments and two databases.	88.71%

The survey found that computer vision technology has not yet been applied to the Thai vehicle registration certificate. Figure 1 shows the Thai vehicle registration certificate issued by the Department of Land Transport [45]. This document shows the vehicle information, including the body number, color, and weight. The staff records the identification information of transfers, cancellations, modifications, and the vehicle owner's tax history to provide up-to-date information. The document helps readers (e.g., police officers, registrars, or car buyers) confirm that the car in question is a registered one and has not been stolen or tampered with to conceal an offence. It also helps car owners maintain, inspect, and replace spare parts that match the vehicle's model and design. Furthermore, a car owner can use a copy of the Thai vehicle registration certificate to represent the original document. However, Thai vehicle registration certificates are still in a hard-copy format, which cannot be automatically connected to the database system. Users must import the data into a system with key-ins that may be error-prone.

The Thai vehicle registration certificate consists of four parts, as follows: Figure 1(a) the cover page, showing the vehicle license number and the registered province. Figure 1(b) Car information, showing the registration date, registration number, province, car type, car model, brand, model, year, color, engine number, type of fuel, number of gas tanks, cylinders, cc, horsepower, weight, payload weight, and seat. Figure 1(c) Owner information, showing the vehicle possession, owner ID, date of birth, nationality, address, occupant information, address, phone number, hire purchase contract and ownership, and officer and registrar

signature. Figure 1(d) Officer recording, showing any change in the history of the car and owner information, such as cancellation, color changes, and additional parts.

Figure 1. Four parts of the Thai vehicle registration certificate (a) cover page, (b) car information, (c) owner information, and (d) officer recording

3. RESEARCH METHOD

This study used 84 sample files of Thai vehicle registration certificates for testing, which were categorized according to the image size/resolution and image characteristics as shown in Table 4. This paper used Intel Core i7-10750H CPU, which has 16 GB (8 GB X2) DDR4 2666 MHz RAM and NVIDIA GEFORCE RTX 2060 6 GB GDDR 6 for testing. The computer installed the Tesseract OCR v4.1.0.20190314 and PHP: 7.4.15 to develop a web application for sending images to Google Cloud Vision and Tesseract recognition via APIs shown in Figures 2 and 3. Moreover, this paper also applied the following three image enhancement techniques for the pre-processing process: i) sharpening applies a 3×3 kernel size, ii) contrast adjustment changes the contrast of the image, and iii) brightness adjustment changes the brightness of the image from -255 to 255.

Table 4. Sample files characteristics of Thai vehicle registration certificates

Categories	Files
Image size/resolution	
- Large size/high resolution (>=1024×768px/300dpi)	19
- Standard size/medium resolution (>=640×480px/120dpi)	45
- Small size/low resolution (<640×480px/72dpi)	20
Image characteristics	
- Non-Damage image	53
- Low brightness image	13
- Blurred or glared image	4
- Torn or wrinkled image	2
- Improperly positioned image	12

```
use thiagoalessio\TesseractOCR\TesseractOCR;

$text = (new TesseractOCR($filepath))->lang('tha', 'eng')->run();
```

Figure 2. Sending images to be processed with the Tesseract

```
$arrHeader = array();
$arrHeader[] = "Content-Type: application/json";


$objImgData = file_get_contents($filepath);
$objImgBase64 =  base64_encode($objImgData);


$arrPostData = array();
$arrPostData['requests'][0]['image']['content'] = $objImgBase64;


$arrPostData['requests'][0]['features'][0]['type'] = "TEXT_DETECTION";
$arrPostData['requests'][0]['features'][0]['maxResults'] = "5";
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL,$strUrl);
curl_setopt($ch, CURLOPT_HTTPHEADER, $arrHeader);
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($arrPostData));
curl_setopt($ch, CURLOPT_RETURNTRANSFER,true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
$result = curl_exec($ch);
curl_close ($ch);


$data = json_decode($result);
$text = $data->responses[0]->textAnnotations[0]->description;
```

Figure 3. Sending images to be processed with the Google Cloud Vision API

The results were evaluated in terms of accuracy and readability. This study counted the words and specific terms used in the Thai vehicle registration certificate to compare the accuracy. On the other hand, the readability was counted from the number of unreadable files. The experiment consisted of 5 steps:

- Test the performance of the Tesseract and the Google Cloud Vision API with each image size.
- Test the performance of the Tesseract and the Google Cloud Vision API according to the given image characteristic.
- Improve image quality with three enhancing techniques before processing and repeat the first and second steps.
- Count the number of words and unreadable files in each step to identify the accuracy and readability of the tools.
- Record and evaluate the results.

4. RESULTS AND DISCUSSION
4.1. Experiment results
The results with 84 samples of vehicle registration files showed that Google Cloud Vision was better at reading and interpreting vehicle registration certificate information when compared with Tesseract, with an average accuracy of 84.43% as shown in Table 5. Tesseract had an average accuracy of 47.02%. Considering the file size and resolution, the Google Cloud Vision API had a percentage accuracy that was significantly higher than that of Tesseract; the Google Cloud Vision API had an accuracy of 94.72% for the large size/high resolution (1024×768 px/300 dpi). Moreover, the small size or low-resolution images had the lowest accuracy.

With regard to image characteristics as shown in Table 6, the group of images taken from the torn or wrinkled documents was the group in which both the Tesseract and Google Cloud Vision API had the lowest accuracy at 3.55% and 17.73%, respectively. For the group of images taken from blurred or glared documents, the accuracy was 35.05% and 55.19%, respectively. For improperly positioned images, such as images that were taken from a long distance, skewed images, or partially captured images, the Google Cloud Vision API provided a high recognition rate. For the preprocessing techniques, the performance evaluation is based on a combination of three image enhancement techniques: Sharpening, contrast adjustment, and brightness adjustment. Table 7 shows that the sharpening and brightness adjustment techniques have the highest accuracy rates for both the Tesseract and Google Cloud Vision API. Table 8 shows a comparison of the unreadable file of the Tesseract and Google Cloud Vision API. All image enhancement techniques could reduce the number of unreadable files, except for the contrast and sharpening technique by Tesseract. The Google Cloud Vision API also works well for both the original and enhanced images.

The misinterpreted Thai characters are presented in Table 9. As can be seen from the tale, all unrecognized characters ('ส' and 'ล', 'น' and 'น') have a similar appearance. When considered in conjunction with the image file size and enhancement techniques, the interpretation error rate could be reduced in larger

files or higher resolutions and with the sharpening techniques because of the increased clarity that allowed the tool to distinguish the characters better.

Table 5. Accuracy of the Tesseract and Google Cloud Vision API for image file size/resolution

Image Size	Accuracy (%)	
	Tesseract	G-Vision
Large size/high resolution	60.21	94.72
Standard size/medium resolution	41.17	79.85
Small size/low resolution	41.21	71.17
Average	47.02	84.43

Table 6. Accuracy of the Tesseract and Google Cloud Vision API for damage image characteristics

Image	Accuracy (%)	
	Tesseract	G-Vision
Low brightness image	34.11	72.44
Blurred or glared image	35.05	55.19
Torn or wrinkled image	3.55	17.73
Improperly positioned image	19.51	83.15

Table 7. Accuracy of the Tesseract and Google Cloud Vision API for enhancement techniques

Image	Accuracy (%)	
	Tesseract	G-Vision
Sharpening	49.36	87.67
Contrast	44.20	71.49
Sharpening and Brightness	52.88	88.90
Contrast and Sharpening	45.47	76.51

Table 8. Number of an unreadable file by the Tesseract and Google Cloud Vision API

Image	Unreadable (Number)	
	Tesseract	G-Vision
Original	12	9
Sharpening	9	6
Contrast	11	8
Sharpening and Brightness	8	6
Contrast and Sharpening	12	7

Table 9. List of Thai characters misinterpreted by the Tesseract and Google Cloud Vision API

Actual Character	Recognized Character	
	Tesseract	G-Vision
จ	ร, ธ, จ	ร
บ	ย, บ, ช, ห	ป, ย, ข, น
ท	บ, ช, ก	ก, ช, ร, ส
พ	ผ, ม, ห	
ส	ล	ศ, ล
ซ	ช, ธ, ข	ข
น	ย, ม, แ, ผ	ย, ช, ผ, ร
ค	ต	ท, อ
ก	ค, ศ, ต	ท, ธ
ต	ล, ค	อ, ค
ล	ฎ, จ, ต	ค, ท
ญ	ย, ล	
ม		ง
จ	ฉ, ธ	น
ย	ผ	ฮ
อ	ภ	

4.2. Discussion and suggestion

This section discusses some significant results and fascinating issues related to other usage areas or research and suggests some possible ways to improve the tools to be more efficient in recognizing and interpreting results. The Google Cloud Vision API has a higher accuracy than Tesseract in terms of both size and image characteristics. In terms of other aspects, we found the following:
− English letters and Arabic numbers could be recognized equally well by both tools.
− Both tools were able to recognize high-definition images with a high accuracy, whereas images that were very small, far, or blurred could not be recognized by the tools.

− Recognizing Thai vowels and tone marks led to the most frequent mistakes, followed by consonants that were similar or looked close to each other.
− The original Thai vehicle registration certificate uses grey color characters and a watermark background that effectively reduces the recognition performance.
− Some Thai vehicle registration certificates have the original strikethrough text and are written with a pattern, which makes the models used for reading printed text unable to read handwritten text.
− Some Thai vehicle registration certificates have errors in printing specific vehicle information on different label lines, which leads to the wrong sequencing problem.

Tesseract is a flexible tool for a developer because it is an open-source software and thus developers can develop, customize, and manage Tesseract as per their individual needs. However, Tesseract can be a little tricky in terms of the installation process and configuration. The Google Cloud Vision API gives better performance than Tesseract because of the availability of a range of services. Furthermore, it is easy to connect and configure and run services on it. However, the Google Cloud Vision API may be less customizable owing to the availability of in-built services. When comparing the accuracy of the technique with other articles on Thai language recognition as detailed in Table 3, this paper found that the articles [33], [41]–[43] used the actual cases as a sample and there has higher accuracy than the proposed method. However, when studying the details of each article, article [41] recognized only six-digit numbers of the Thai lottery ticket. Articles [33], [42], [43] recognized the Thai sign and the car license plate. It contains characters and lines less complexity than the Thai vehicle registration certificate. Moreover, Kitvimonrat and Watcharabutsarakham [34] found that there are some characters that the tool often miscalculates such as "ข-ช", "ด-ต ", "ฌ-ฌ" and "ก-ภ" that consistency with Table 9.

Next, we explore the possible ways to improve the above tools to be more efficient in recognizing and interpreting results:
− The developer should define the boundary, frame, or pattern matching to reduce unnecessary reading information and avoid improperly positioned images.
− The developer should adjust the character's color or contrast before the recognizing process, including the removal of the additional watermark background.
− The developer should develop a program that can link models supporting handwritten and printed text recognition, helping solve the reading issues with the Thai vehicle registration certificate.
− Information retrieval methods or natural language processing methods can improve the accuracy of the post-processing results.

5. CONCLUSION

This paper compares the Thai character recognition performance between the Tesseract and Google Cloud Vision API to develop an automatic document recognition system for the Thai vehicle registration certificate in the future. The study and testing results showed that Google Cloud Vision was more accurate than Tesseract, with accuracies of 84.43% and 47.02%, respectively. This study recommends an image size of 1024×768 px or a resolution of 300 dpi or more, and preferably sharpening and brightness adjustment before processing, which can significantly improve the accuracy of the tools.

REFERENCES

[1]   V. Suma, "Computer vision for human-machine interaction-review," *Journal of Trends in Computer Science and Smart Technology*, vol. 2019, no. 02, pp. 131–139, Dec. 2019, doi: 10.36548/jtcsst.2019.2.006.
[2]   X. Li and Y. Shi, "Computer vision imaging based on artificial intelligence," in *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, Aug. 2018, pp. 22–25, doi: 10.1109/ICVRIS.2018.00014.
[3]   J. Park *et al.*, "Recent development of computer vision technology to improve capsule endoscopy," *Clinical Endoscopy*, vol. 52, no. 4, pp. 328–333, Jul. 2019, doi: 10.5946/ce.2018.172.
[4]   C.-Z. Dong and F. N. Catbas, "A review of computer vision–based structural health monitoring at local and global levels," *Structural Health Monitoring*, vol. 20, no. 2, pp. 692–743, Mar. 2021, doi: 10.1177/1475921720935585.
[5]   D. van der Haar, "Student emotion recognition using computer vision as an assistive technology for education," in *Information Science and Applications*, 2020, pp. 183–192.
[6]   K. D. Saputra, D. A. Rahmaastri, K. Setiawan, D. Suryani, and Y. Purnama, "Mobile financial management application using Google Cloud Vision API," *Procedia Computer Science*, vol. 157, pp. 596–604, 2019, doi: 10.1016/j.procs.2019.09.019.
[7]   P. Raka *et al.*, "OCR to read credit/debit card details to autofill forms on payment," *International Journal of Research in Engineering, Science and Management*, vol. 2, no. 4, pp. 478–481, 2019.

[8]   T. S. Gunawan, A. Mutholib, and M. Kartiwi, "Design of automatic number plate recognition on android smartphone platform," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 5, no. 1, pp. 99–108, Jan. 2017, doi: 10.11591/ijeecs.v5.i1.pp99-108.

[9]   H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation-A review," *Information Processing in Agriculture*, vol. 7, no. 1, pp. 1–19, Mar. 2020, doi: 10.1016/j.inpa.2019.09.006.

[10]   A. Puranic, Deepak K. T., and Umadevi V., "Vehicle number plate recognition system: a literature review and implementation using template matching," *International Journal of Computer Applications*, vol. 134, no. 1, pp. 12–16, Jan. 2016, doi: 10.5120/ijca2016907652.

[11]   M. Sugadev, Yogesh, P. K. Sanghamreddy, and S. K. Samineni, "Rough terrain autonomous vehicle control using Google Cloud Vision API," in *2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC)*, Aug. 2019, pp. 244–248, doi: 10.1109/ICPEDC47771.2019.9036621.

[12]   W. Budiharto, E. Irwansyah, J. S. G. Suroso, and A. A. Santoso, "Design of object tracking for military robot using PID controller and computer vision," *ICIC Express Letters*, vol. 14, no. 3, pp. 289–294, 2020.

[13]   B. Arshad, R. Ogie, J. Barthelemy, B. Pradhan, N. Verstaevel, and P. Perez, "Computer vision and IoT-based sensors in flood monitoring and mapping: a systematic review," *Sensors*, vol. 19, no. 22, Nov. 2019, doi: 10.3390/s19225012.

[14]   K. Vasudeven and A. James, "A survey on malayalam optical character recognition system," in *Proceedings of the International Conference on Emerging Trends in Engineering & Technology (ICETET-2015)*, Oct. 2015, pp. 16–23, doi: 10.3850/978-981-09-5346-1_CSE-024.

[15]   P. S. Deokate and Y. V. Parkale, "A study of optical character recognition techniques," *International Journal of basic and applied research*, vol. 9, no. 5, pp. 228–234, 2019.

[16]   R. Salunke, D. Badhe, V. Doke, and Y. Raykar, "The state of the art in text recognition techniques," *IJARCCE*, pp. 442–444, Feb. 2015, doi: 10.17148/IJARCCE.2015.42100.

[17]   D. Vaithiyanathan and M. Muniraj, "Cloud based text extraction using google cloud vison for visually impaired applications," in *2019 11th International Conference on Advanced Computing (ICoAC)*, Dec. 2019, pp. 90–96, doi: 10.1109/ICoAC48765.2019.246822.

[18]   R. Smith, "An overview of the Tesseract OCR Engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, Sep. 2007, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.

[19]   A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, Jul. 2018, doi: 10.1016/j.neucom.2018.01.092.

[20]   S. Naz, S. Bin Ahmed, R. Ahmad, and M. I. Razzak, "Zoning Features and 2DLSTM for Urdu text-line recognition," *Procedia Computer Science*, vol. 96, pp. 16–22, 2016, doi: 10.1016/j.procs.2016.08.084.

[21]   A. Marial and J. Jos, "Feature extraction of optical character recognition: Survey," *International Journal of Applied Engineering Research*, vol. 12, no. 7, pp. 1129–1137, 2017.

[22]   J. M. Patil and A. P. Mane, "Multi font and size optical character recognition using template matching," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 1, pp. 504–506, 2013.

[23]   C. Patel, A. Patel, and D. Patel, "Optical character recognition by open source OCR tool tesseract: a case study," *International Journal of Computer Applications*, vol. 55, no. 10, pp. 50–56, Oct. 2012, doi: 10.5120/8794-2784.

[24]   J. Sharma, G. S. Sindhu, S. Sejwal, J. Solanki, and R. Majumdar, "Intelligent vehicle registration certificate," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Feb. 2019, pp. 418–423, doi: 10.1109/AICAI.2019.8701286.

[25]   F. Adamo, F. Attivissimo, A. Di Nisio, and M. Spadavecchia, "An automatic document processing system for medical data extraction," *Measurement*, vol. 61, pp. 88–99, Feb. 2015, doi: 10.1016/j.measurement.2014.10.032.

[26]   I. Kavati, G. K. Kumar, S. Kesagani, and K. S. Rao, "Signboard text translator: a guide to tourist," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2496–2501, Oct. 2017, doi: 10.11591/ijece.v7i5.pp2496-2501.

[27]   A. J. Samuel and S. Sebastian, "An algorithm for IoT based vehicle verification system using RFID," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 5, pp. 3751–3758, Oct. 2019, doi: 10.11591/ijece.v9i5.pp3751-3758.

[28]   R. Bhat and B. Mehandia, "Recognition of vehicle number plate using MATLAB," *International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering*, vol. 2, no. 8, pp. 1899–1903, 2015.

[29]   A. S. Agbemenu, J. Yankey, and E. O. Addo, "An automatic number plate recognition system using OpenCV and tesseract OCR engine," *International Journal of Computer Applications*, vol. 180, no. 43, pp. 1–5, May 2018, doi: 10.5120/ijca2018917150.

[30]   A. Chinapas, P. Polpinit, and K. Saikaew, "Personal verification system using thai ID Card and face photo for cross-age face," in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, Oct. 2019, pp. 325–330, doi: 10.1109/ICSEC47112.2019.8974845.

[31]   M. Pu, N. A. Abd Majid, and B. Idrus, "Framework based on mobile augmented reality for translating food menu in Thai language to Malay language," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 1, pp. 153–159, Feb. 2017, doi: 10.18517/ijaseit.7.1.1797.

[32]   W. Yimyam and M. Ketcham, "The automated parking fee calculation using license plate recognition system," in *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, 2017, pp. 325–329, doi: 10.1109/ICDAMT.2017.7904985.

[33]   S. Kraisin and N. Kaothanthong, "Accuracy improvement of a province name recognition on Thai license plate," in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Nov. 2018, pp. 1–6, doi: 10.1109/iSAI-NLP.2018.8692983.

[34]   A. Kitvimonrat and S. Watcharabutsarakham, "A robust method for Thai license plate recognition," in *Proceedings of the 2020 3rd International Conference on Image and Graphics Processing*, Feb. 2020, pp. 17–21, doi: 10.1145/3383812.3383815.

[35]   T. Kobchaisawat and T. H. Chalidabhongse, "Thai text localization in natural scene images using convolutional neural network," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, Dec. 2014, pp. 1–7, doi: 10.1109/APSIPA.2014.7041775.

[36]   C. Sumetphong and S. Tangwongsan, "An optimal approach towards recognizing broken thai characters in OCR systems," in *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, Dec. 2012, pp. 1–5, doi: 10.1109/DICTA.2012.6411736.

[37]   A. Thammano and P. Duangphasuk, "Printed Thai character recognition using the hierarchical cross-correlation ARTMAP," in *17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)*, 2005, pp. 4 pp.--698, doi: 10.1109/ICTAI.2005.100.

[38]   P. Somboonsak, "Misspelling error detection in thai language application," in *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City - ICIT 2018*, 2018, pp. 19–24, doi: 10.1145/3301551.3301584.

[39]   P. Chomphuwiset, "Printed thai character segmentation and recognition," in *2017 IEEE 4th International Conference on Soft*

Computing & Machine Intelligence (ISCMI), Nov. 2017, pp. 123–127, doi: 10.1109/ISCMI.2017.8279611.

[40]   C. Pornpanomchai and M. Daveloh, "Printed Thai character recognition by genetic algorithm," in *2007 International Conference on Machine Learning and Cybernetics*, 2007, pp. 3354–3359, doi: 10.1109/ICMLC.2007.4370727.

[41]   S. Tangwannawit and W. Saetang, "Recognition of lottery digits using OCR technology," in *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, 2016, pp. 632–636, doi: 10.1109/SITIS.2016.105.

[42]   W. Jirattitichareon and T. H. Chalidabhongse, "Automatic detection and segmentation of text in low quality Thai sign images," in *APCCAS 2006 - 2006 IEEE Asia Pacific Conference on Circuits and Systems*, Dec. 2006, pp. 1000–1003, doi: 10.1109/APCCAS.2006.342256.

[43]   S. Toolpeng, K. Wannapoporn, and G. Phanomchoeng, "Thai license plate recognition algorithm with service routine procedure for automatic barrier gate," in *Proceedings of the 2019 3rd International Conference on Virtual and Augmented Reality Simulations*, Feb. 2019, pp. 77–81, doi: 10.1145/3332305.3332314.

[44]   P. Duangphasuk and A. Thammano, "Thai vehicle license plate recognition using the hierarchical cross-correlation ARTMAP," in *2006 3rd International IEEE Conference Intelligent Systems*, Sep. 2006, pp. 652–655, doi: 10.1109/IS.2006.348496.

[45]   Department of Land Transport, "Vehicle registration department," *Department of Land Transport*. Accessed: Nov. 21, 2020. [Online]. Available: https://www.dlt.go.th/site/skp2/m-news/1387/

BIOGRAPHIES OF AUTHORS

Karanrat Thammarak, is an instructor and researcher in computer engineering and electronics, school of engineering and technology at Walailak University, Thailand. She has received a Ph.D. in Computer Science from King Mongkut's Institute of Technology Ladkrabang, Thailand. Exciting research areas are computer vision and NLP. He can be contacted at email: kanchan.th@mail.wu.ac.th.

Prateep Kongkla is an academic computer engineering and intelligent systems in computer engineering and electronics, school of engineering and technology at Walailak University, Thailand. He has a bachelor's degree in software engineering from Walailak University. He is currently studied in system development and image analysis. He can be contacted at email: prateep.ko@mail.wu.ac.th.

Yaowarat Sirisathitkul is an instructor and researcher in computer engineering and electronics, school of engineering and technology at Walailak University, Thailand. Research areas are image processing and machine vision. She can be contacted at email: syaowara@wu.ac.th.

Sarun Intakosum received a doctoral degree in Computer Science from the Asian Institute of Technology. He is now Assistant Professor at the Department of Computer Science, School of Science, King Mongkut's Institute of Technology Ladkrabang. He can be contacted at email: intakosumsarun@gmail.com.