Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 157 (2019) 499–505

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12-13 September 2019

Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application

Abdul Robby G.[a], Antonia Tandra[a], Imelda Susanto[a], Jeklin Harefa[a], Andry Chowanda[a,*]

[a] *Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

Abstract

Recognising characters from text have been a popular topic in the computer vision area. The application can benefit to many problems in the world. For example: recognising text in documents, classifying the text or scripts of documents, plate recognition, etc. Many researchers have been developed the methods for recognising characters in by using Optical Character Recognition methods. Although text recognition problem using Optical Character Recognition has been more or less solved, most of the Optical Character Recognition problem explored is belong to Latin alphabet texts. Meanwhile, there are several languages have non-Latin scripts as the written text. Recognising a non-Latin script is quite challenging as the contour and shape of the text are relatively different with a Latin script text. This research aims to collect datasets for OCR in Javanese characters. A total of 5880 characters were collected and trained with several methods with Tesseract OCR tools. The models then be implemented to a mobile phone (Android based). The highest accuracy (97,50%) achieved by the model was achieved by combining single boundary box for the whole parts of the character and the separate boundary boxes in main body and *sandangan* parts.

Keywords: OCR; Non-Latin Script; Javanese Script; Tesseract; Android

1. Introduction

Computer vision field has been popular to solve several problems such as: recognising faces[1,2], recognising emotion[2,3], recognising places[?], recognising objects[4], etc. One of the popular implementation of computer vision in the real world is Optical Character Recognition as known as OCR. Optical Character Recognition (OCR) is one of the techniques that converts the scanned or printed image document into an editable text document[5,6]. This technique has become a major field since it already been implemented in various application, such as automatic license plate recognition[7,8], the classification of script recognition[9,10], recognising text from traffic sign[11,12], and many more. As a

* Corresponding author. Tel.: +62-21-534-5830
E-mail address: achowanda@binus.edu

1877-0509
10.1016/j.procs.2019.09.006

country that consists of many islands, Indonesia has a very wide variety of cultures, religions, languages, and so on. Furthermore, Java is one of the islands in Indonesia that has a large population in Indonesia[13]. Javanese script is a specific set of characters used by Java island and these characters are still often found in the street signboards, wall cravings or another historical relic. Javanese Scripts is also considered as the national heritage of Indonesia. However, the people nowadays are facing the problem where not all the Javanese people are able to read the Javanese scripts, particularly the young generation people[14].

Many researchers have been developed the methods in order to get the better accuracy in recognising characters by using Optical Character Recognition. For example, in 2017, Phangtriastu et al.[15] combined some extracted features such as zoning algorithm, projection profile and Histogram of Oriented Gradients (HOG) using the comparison of two most commonly classifier: Support Vector Machine (SVM) and Artificial Neural Network (ANN). The proposed method achieves the highest accuracy of 94,43% by using Support Vector Machine (SVM) Classifier with the feature extraction algorithms, which are: Projection Profile and the combination of Zoning + Projection Profile. Moreover, in 2013, Mithe, Indalkar and Divekar[16] presented the method for recognizing characters by using Tesseract, text recognition OCR engine in Android system. Their experiment has presented 3 important steps: segmentation, feature extraction and classification. Pawar et al.[17] also implemented the Tesseract Optical Character Recognition (OCR) Engine to extract the textual data from the scanned documents or images. This system prove that the tesseract OCR engine can be used for recognizing the scanned documents. Another research comes from Chanda et al.[18]. They proposed a system to address the problem in recognising Han based characteristics (Chinese, Japanese, Korean and Roman scripts) using directional chain-code histogram-based feature along with the Gaussian Kernel-based Support Vector Machine. Their experiment obtained 98,39% accuracy at character level and 99,85% at block level. Some of attempt to solve the OCR problem using deep learning. Dewa, Fadhilah and Afiahayati[19] implement CNN to recognise the javanese scipt and achieved the best of 82,00% of accuracy. Moreover, Khadijah, et.al.[20] also implement CNN and DNN to clasify the javanese script and achieved 70,22% and 64,65% respectively.

Although text recognition problem using Optical Character Recognition has been more or less solved, most of the Optical Character Recognition problem explored is belong to Latin alphabet texts. Meanwhile, there are several languages have non-Latin scripts as the written text. Only few research have been done to explore non-Latin text recognition problem using Optical Character Recognition[21,22]. Recognising a non-Latin script is quite challenging as the contour and shape of the text are relatively different with a Latin script text. In addition, they also quite distinguishable with the other non-Latin script text[21,22]. This research aims to collect a numerous number of training dataset in Javanese scripts. As currently there is no existing publicly available Javanese scripts for OCR training. Moreover, this research contributes in the exploration in training models for OCR in Javanese scripts using Tesseract OCR modules and APIs[6]. The trained models then were deployed to an Android application as a mobile OCR application in Javanese scripts. The results show a good promise in recognising Javanese characters with a mobile phone in real time. Several experiments have been done in this research and showed that in the first experiment, the system is able to read the Javanese characters with left or right hand boundary box belong to sandangan (e.g. sets of ”o” and sets of ”e” characters). While it was relatively harder for the first testing method to identify the Javanese scripts with the sandangan that has the position in above or below the main body (e.g. sets of ”i” and sets of ”u” characters). The problem were fixed in the second experiment, where it is able to read the Javanese characters with above or below hand box (sandangan) position (e.g. Sets of ”i” and Sets of ”u” characters). In almost all models in all experiments, it was quite hard to distinguish between Sets of ”e” and Sets of ”e” Javanese characters so that the Sets of ”e” character is often detected as Sets of ”i” character and it causes lower accuracy of Sets of ”e” characters. The highest accuracy belongs to Sets of ”a” characters trained and tested in the digital font dataset (97,50%) in the third experiment.

2. Proposed Method

Fig. 1 illustrates the proposed research methodology. The research method is divided into five phases. First step is collecting the dataset. The collected data is Javanese script that obtained from digital or handwritten texts or photographs. A total 46 sets of handwriting Javanese scripts were collected resulting in 5520 characters (120 characters per-set), in addition to three sets of digital Javanese text (normal text and italic text) also were collected from the Internet (Hanacaraka.ttf), resulting in 360 characters. The handwriting Javanese scripts data collection was carried out in several schools from around 10 person with various type of ages in Java island area.

Fig. 1: Proposed Method)

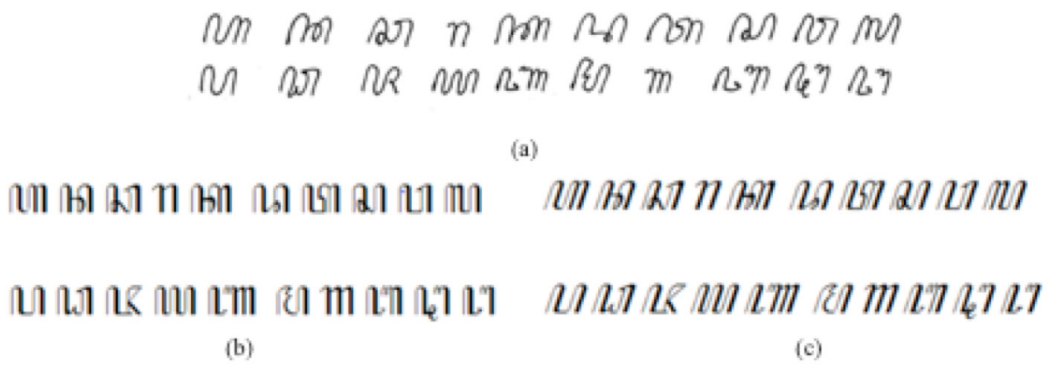

Fig. 2: (a) Handwritten Javanese Scripts; (b) Digital Javanese Scripts; (c) Italic Digital Javanese Scripts

Fig. 2 shows the sample of the dataset of the Javanese scripts. Fig. 2 (a) illustrates the sample for the script collected from the handwriting scripts. Fig. 2 (b) demonstrates the sample of normal text set gathered from Hanacaraka.ttf, while Fig. 2 (c) illustrates the sample of the italic text set gathered from Hanacaraka.ttf. After data has been collected, the next step is pre-processing in order to obtain the higher quality of inputted image. Some of image enhancement methods were applied in this phase. This is due to the data collected (some of) from the handwriting process are quite blurry, not aligned, missing some of the components of the font (i.e. *sandangan* the phonetic symbol in Javanese scripts), etc. Some of the image pre-processing operation that used in this experiment are: rotation, filling the missing of Javanese characters, noise removal and clarify the stroke of handwritten Javanese characters. The dataset then was trained with Neural-Network API from the Tesseract OCR tool[23,6,10]. This research proposes three training methods, they are: using separate bounding box to annotate the main body of the scripts and the phonetic symbols (i.e. sandangan) (see Fig. 4), using the same bounding box to annotate both the main body and the phonetic symbols (i.e. *sandangan*) (see Fig. 4), the combination (a hybrid) of first and second methods. The models trained with those three methods were then evaluated, and finally the best model was deployed to an Android application. A simple Android application was developed as the working OCR tools to recognise Javanese scripts in real-time with a mobile phone. All the details of the results from the data collection to the system development phases are described thoroughly in the next section.

3. Experimental Results

A total of 5880 Javanese characters were collected from both digital (3 sets x 120 characters) and hand-writing (46 sets x 120 characters) sources. Some of the characters collected from the hand-writing process are quite blurry or noisy (of the inks), not aligned, missing some of the components of the font (i.e, sandangan the phonetic symbol in Javanese scripts), etc (see Fig. 3). Hence, some of image enhancement methods were applied in the pre-processing phase. Fig. 3 shows the results of the characters before (original, Fig. 3 upper side) and the characters after the image enhancement methods applied (Fig. 3 lower side). All the image enhancement methods were applied to all the original images, hence, the hand written dataset collected is clean and robust to be used as the training dataset for Javanese scripts OCR recognition.

One final step before the dataset was thrown into the training phase was the annotation. The parts of Javanese script were annotated using three different methods. There are two main parts in the Javanese script text: the main body, and the phonetics symbols (i.e. *sandangan*). There is always only one main body with no (0) to several (1-3) phonetics

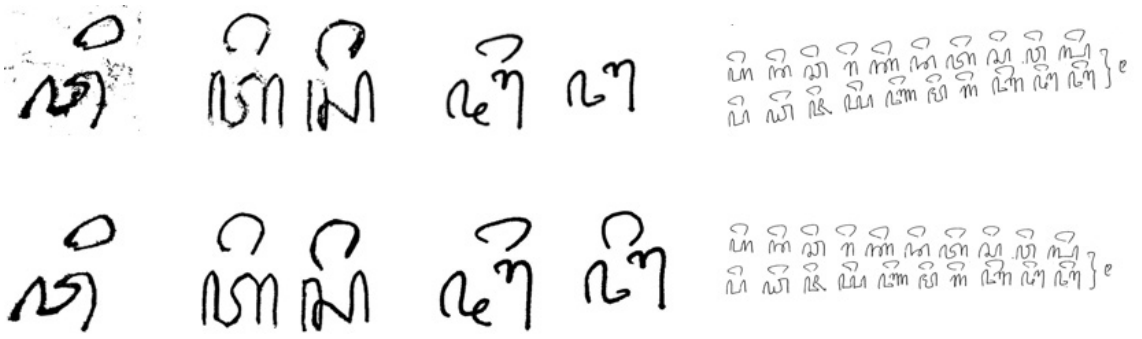

Fig. 3: Original Hand Written Scripts (upper side), and Pre-Processed Scripts (lower side).

symbols (i.e. *sandangan*) in a Javanese character. Fig. 4 shows the annotation methods in the training phases for both digital and hand writing datasets. There are three annotation methods in this research. The first method is using separate bounding box for each parts of the Javanese character. A main body of the text has one bounding box, and the phonetics symbols (*sandangan*, if any) also have their own bounding boxes. Fig. 4 (a) shows the annotation method 1 in the hand writing dataset, while Fig. 4 (b) illustrates the annotation method 1 in the digital dataset. The second annotation method for training is using the same bounding box to annotate the whole parts of the Javanese character (i.e. the main body of the text and the phonetics symbols (*sandangan*, if any) shared the same bounding box). Fig. 4 (c) demonstrates the annotation method 2 in the hand writing dataset, while Fig. 4 (d) shows the annotation method 2 in the digital dataset. The last method is the combination (hybrid) between method 1 and method 2. So, there are multiple bounding boxes indicating the main body and the phonetics symbols (*sandangan*, if any) (annotation method 1) as well as a whole bounding box indicating both main body and the phonetics symbols (annotation method 2).

All the dataset was trained three times for each method, resulting in nine times trainings combination. For each method, three datasets were trained using the same learning algorithm. The datasets are a digital dataset (2 sets @120 characters), a hand writing dataset (46 sets @120 character, and a dataset of hand writing and digital characters (1 set @120 characters). Due to the different characteristic in the testing phase, the number of characters in the dataset used in the testing depending on the type of the dataset (e.g. the hand written dataset has more characters compare to the digital dataset). The training algorithm used in this research is based on the APIs and tools of Tesseract OCR[6]. JTessBoxEditor[6] was used as a tool for training in this research. Moreover, Neural network algorithm was chosen as the method used in this research as the default method in the JTessBoxEditor. The pipeline for training using JTessBoxEditor is annotating the training dataset using bounding box, set the variables for the cluster of the characters. To enhance the accuracy for the training dataset with non-Latin characters (e.g. Arabic, Mandarin, etc.), a shapeclustering method[6,10] used in all the training combinations. The Neural Network consists of 5 layers of neural, two are the convolutional layers, another two are the pooling layers, and the last layer consists of fully connected layer and classification layer. Please refer to the original papers for the details of methods and architecture[6,10,23].

In the testing, there are three methods for identify the Javanese script. The first method is by using the first training method with the combination of handwriting Javanese scripts, digital Javanese scripts and italic digital Javanese scripts. Hence, there were a total of twelve combination of training and testing of the dataset. The first evaluation was the evaluation towards the first training method, where the separate bounding boxes were used to annotate each parts of Javanese character (the main body and the sandangan parts). The total data used in this testing were 780 characters which consists of 300 characters of digital and italic digital Javanese scripts, and 480 characters of handwriting Javanese scripts. Table 1 shows the result of first testing method. Training Dataset Type column indicates the dataset type used in the training, they are digital font, handwriting, and the combination of digital font and handwriting dataset. Column Testing Dataset indicates the dataset type used in the testing, they are digital font and handwriting. Sets of ”a” column means all the sets of ”a” characters in the Javanese scripts (e.g. ha na ca ra ka da ta sa wa la), sets of ”i” column means all the sets of ”i” characters in the scripts (e.g. hi ni ci ri ki di ti si wi li) and so on. The highest accuracy achieved by Sets of ”a” and Sets of ”é” characters (92,62%) in digital font training dataset and digital font testing dataset. The best dataset combination in training and testing in this method is digital font training dataset and digital font testing dataset. Handwriting dataset in the training and testing, in this method did not performed quite

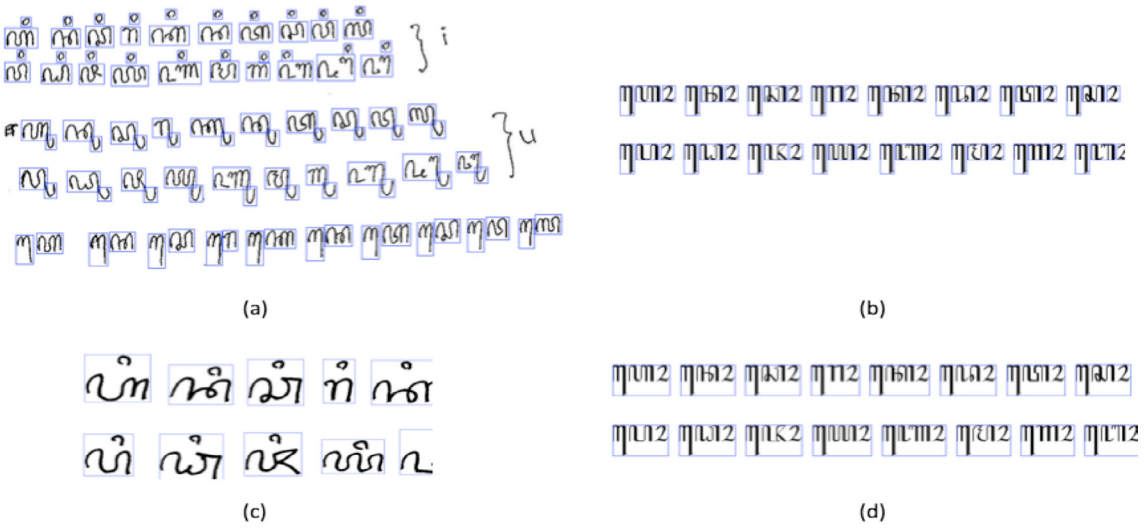

Fig. 4: The Annotation Methods for Training; Method 1 Main body of the text and phonetic symbol (*sandangan*) were annotated using separate bounding boxes (a) in hand writing text, (b) in digital text; Method 2 the same bounding box was used to annotated both the main body and phonetic symbol (*sandangan*) (c) in hand writing text, (d) in digital text.

Table 1: The result of first testing method

Training Dataset	Testing Dataset	Sets of a	Sets of i	Sets of u	Sets of e	Sets of o	Sets of é
Digital font	Digital font	92,62%	0,00%	16,50%	0,00%	92,50%	92,62%
	Handwriting	60,00%	0,00%	0,00%	0,00%	45,00%	30,00%
Handwriting	Digital font	27,50%	0,00%	0,00%	0,00%	0,00%	0.00%
	Handwriting	40,00%	0,00%	0,00%	0,00%	17,50%	16,25%
Digital font + Handwriting	Digital font	85,00%	2,50%	7,50%	0,00%	85,00%	85,00%
	Handwriting	57,50%	0,00%	0,00%	0,00%	65,00%	40,00%

well, due to the variation of the handwriting data. This experiment also shows that the first testing method is able to read the Javanese characters with left or right hand boundary box belong to *sandangan* (e.g. sets of ”o” and sets of ”” characters). While it was relatively harder for the first testing method to identify the Javanese scripts with the sandangan that has the position in above or below the main body (e.g. sets of ”i” and sets of ”u” characters).

The second evaluation was the evaluation towards the second training method, where the main body and the *sandangan* parts share the same bounding box (see Fig. 4 (c) and (d)). Similar with the first evaluation, the data used in this testing were 780 characters which consists of 300 characters of digital and italic digital Javanese scripts, and 480 characters of handwriting Javanese scripts. Table 2 shows the result of second testing method. Identital with the first evaluation, all the characters were evaluated six times with six combination of digital font, handwriting and combination of both datasets in the training and testing of the models. The highest accuracy score achieved by the Sets of ”a” characters (92,62%) in digital font training dataset and digital font testing dataset. The best combination for testing and training dataset type still the digital font datasets, however, in this method, there were some improvement to the handwriting models. In this evaluation, it can be concluded that the Sets of ”a”, Sets of ”i” and Sets of ”u” characters achieved high accuracy (92,62%, 87,02% and 92,07% respectively) using digital font dataset and data testing. This experiment also shows that the second testing method is able to read the Javanese characters with above or below hand box (*sandangan*) position (e.g. Sets of ”i” and Sets of ”u” characters). For this second testing method, it is harder to distinguish between Sets of ”e” and Sets of ”i” Javanese characters so that the Sets of ”e” character is often detected as Sets of ”i” character and it causes lower accuracy of Sets of ”e” characters. This is due to the *sandangan* (phonetics symbols) of Sets of ”e” and Sets of ”i” are similar, with the one is smaller than the other. Seeing the advantages in the both training methods, we decided to combine both methods and evaluated in the third method.

Table 2: The result of second testing method

Training Dataset	Testing Dataset	Sets of a	Sets of i	Sets of u	Sets of e	Sets of o	Sets of é
Digital font	Digital font	92,62%	87,02%	92,07%	0,00%	0,00%	4,29%
	Handwriting	60,00%	38,33%	46,25%	8,75%	1,25%	2,50%
Handwriting	Digital font	35,00%	17,50%	27,50%	0,00%	0,00%	0.00%
	Handwriting	43,75%	30,00%	5,00%	5,00%	0,00%	2,50%
Digital font + Handwriting	Digital font	77,50%	72,50%	82,50%	2,50%	0,00%	5,00%
	Handwriting	61,25%	48,75%	42,50%	8,75%	1,25%	5,00%

Table 3: The result of third testing method

Training Dataset	Testing Dataset	Sets of a	Sets of i	Sets of u	Sets of e	Sets of o	Sets of é
Digital font	Digital font	97,50%	83,58%	97,17%	37,50%	93,75%	76,00%
	Handwriting	58,75%	37,50%	47,50%	35,00%	20,00%	34,17%

The third testing method is by using the combination between first and second training methods with the combination of handwriting Javanese scripts, digital Javanese scripts and italic digital Javanese scripts. This last method is expected to improve the accuracy in recognising the Javanese characters. Unlike the previous methods, the third testing method only uses 300 Javanese characters which consists of the combination between digital and italic digital Javanese scripts. Table 3 shows the result of second testing method. This is due to the nature of the experiments, this testing method was aimed to test the hypothesis that looking at the both methods characteristics and results, if both methods were combined, it should result in a higher accuracy for the models. This method combine method 1 and method 2. So, there are multiple bounding boxes indicating the main body and the phonetics symbols (*sandangan*, if any) (annotation method 1) as well as a whole bounding box indicating both main body and the phonetics symbols (annotation method 2). Table 3 demonstrates the results of the third testing method. it can be concluded that the third testing method is able to achieve the better accuracy result than other previous methods. Compared to another character, only the Sets of ”e” characters get a fairly low accuracy result (37,50%) as well as Sets of ”o” characters (20,00%). Same as the second method, for this third testing method, it is harder to distinguish between Sets of ”e” and Sets of ”i” Javanese characters so that the Sets of ”e” character is often detected as Sets of ”i” character and it causes lower accuracy of Sets of ”e” characters. The highest accuracy still belongs to Sets of ”a” characters trained and tested in the digital font dataset (97.50%).

4. Conclusion and Future Work

This research aims to collect dataset for Optical Character Recognition in non-Latin characters (i.e. Javanese characters) and make it publicly available for research purposes. There is not many research have been done in recognizing character in Javanese scripts. Moreover, albeit Javanese scripts are considered as the national heritage of Indonesia. The usage of this language is getting faded away. To build the dataset, a total of 5880 Javanese characters were collected from both digital (3 sets x 120 characters) and hand-writing (46 sets x 120 characters) sources. Some of the characters collected from the hand-writing process are quite blurry or noisy (of the inks), not aligned, missing some of the components of the font (i.e, *sandangan* the phonetic symbol in Javanese scripts). Hence, some of image enhancement methods were applied in the pre-processing phase. The dataset then trained and tested with several methods. The highest accuracy (97,50%) achieved by the model was achieved by combining single boundary box for the whole parts of the character and the separate boundary boxes in main body and *sandangan* parts. In almost all models, it was harder to distinguish between Sets of ”e” and Sets of ”i” Javanese characters so that the Sets of ”e” character is often detected as Sets of ”i” character and it causes lower accuracy of Sets of ”e” characters. The next step for this research is to collect more handwriting datasets as well as working with the handwriting dataset. Moreover, some of deep learning (e.g. Convolutional Neural Network) methods can be explored to recognise characters in Javanese texts.

References

1. Sutoyo, R., Harefa, J., Chowanda, A.. Unlock screen application design using face expression on android smartphone. In: *MATEC Web of Conferences*; vol. 54. EDP Sciences; 2016, p. 05001.
2. Suryani, D., Ekaputra, V., Chowanda, A.. Multi-modal asian conversation mobile video dataset for recognition task. *International Journal of Electrical and Computer Engineering (IJECE)* 2018;**8**(5).
3. Moniaga, J.V., Chowanda, A., Prima, A., Oscar, , Rizqi, M.D.T.. Facial expression recognition as dynamic game balancing system. *Procedia Computer Science* 2018;**135**:361 – 368. doi:\bibinfo{doi}{https://doi.org/10.1016/j.procs.2018.08.185}. The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life; URL http://www.sciencedirect.com/science/article/pii/S1877050918314741.
4. Budiharto, W., Gunawan, A.A.S., Suroso, J.S., Chowanda, A., Patrik, A., Utama, G.. Fast object detection for quadcopter drone using deep learning. In: *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. 2018, p. 192–195. doi:\bibinfo{doi}{10.1109/CCOMS.2018.8463284}.
5. Shinde, A.A., Chougule, D.. Text pre-processing and text segmentation for ocr. *International Journal of Computer Science Engineering and Technology* 2012;**2**(1):810–812.
6. Smith, R.. An overview of the tesseract ocr engine. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*; vol. 2. IEEE; 2007, p. 629–633.
7. Patel, C., Patel, A., Patel, D.. Optical character recognition by open source ocr tool tesseract: A case study. *International Journal of Computer Applications* 2012;**55**(10):50–56.
8. Patel, C., Shah, D., Patel, A.. Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications* 2013;**69**(9).
9. Unnikrishnan, R., Smith, R.. Combined script and page orientation estimation using the tesseract ocr engine. In: *Proceedings of the International Workshop on Multilingual OCR*. ACM; 2009, p. 6.
10. Smith, R., Antonova, D., Lee, D.S.. Adapting the tesseract open source ocr engine for multilingual ocr. In: *Proceedings of the International Workshop on Multilingual OCR*. ACM; 2009, p. 1.
11. Greenhalgh, J., Mirmehdi, M.. Recognizing text-based traffic signs. *IEEE Transactions on Intelligent Transportation Systems* 2014;**16**(3):1360–1369.
12. Reina, A.V., Sastre, R.L., Arroyo, S.L., Jiménez, P.G.. Adaptive traffic road sign panels text extraction. In: *Proceedings of the 5th WSEAS International Conference on Signal Processing, Robotics and Automation*. World Scientific and Engineering Academy and Society (WSEAS); 2006, p. 295–300.
13. Wilonoyudho, S., Rijanta, R., Keban, Y.T., Setiawan, B.. Urbanization and regional imbalances in indonesia. *Indonesian Journal of Geography* 2017;**49**(2):125–132.
14. Hambali, I., Sunarto, M.D., Sutanto, T.. Rancang bangun aplikasi pembelajaran aksara jawa berbasis android. *Jurnal JSIKA* 2013;**2**(2):106–112.
15. Phangtriastu, M.R., Harefa, J., Tanoto, D.F.. Comparison between neural network and support vector machine in optical character recognition. *Procedia computer science* 2017;**116**:351–357.
16. Mithe, R., Indalkar, S., Divekar, N.. Optical character recognition. *International journal of recent technology and engineering (IJRTE)* 2013;**2**(1):72–75.
17. Pawar, N., Shaikh, Z., Shinde, P., Warke, Y.. Image to text conversion using tesseract. *International Research Journal of Engineering and Technology* 2019;**6**(2):516519.
18. Chanda, S., Pal, U., Franke, K., Kimura, F.. Script identification–a han and roman script perspective. In: *2010 20th International Conference on Pattern Recognition*. IEEE; 2010, p. 2708–2711.
19. Dewa, C.K., Fadhilah, A.L., Afiahayati, A.. Convolutional neural networks for handwritten javanese character recognition. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 2018;**12**(1):83–94.
20. Khadijah, R., Nurhadiyatna, A.. Deep learning for handwritten javanese character recognition. In: *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE; 2017, p. 59–64.
21. Märgner, V., El Abed, H.. *Guide to OCR for Arabic scripts*. Springer; 2012.
22. Mozaffari, S., Faez, K., Faradji, F., Ziaratban, M., Golzan, S.M.. A comprehensive isolated farsi/arabic character database for handwritten ocr research. In: *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft; 2006, .
23. Smith, R., Gu, C., Lee, D.S., Hu, H., Unnikrishnan, R., Ibarz, J., et al. End-to-end interpretation of the french street name signs dataset. In: *European Conference on Computer Vision*. Springer; 2016, p. 411–426.