IJASST International Journal of Applied Sciences and Smart Technologies

Volume 06 Issue 01

p-ISSN 2655-8564, e-ISSN 2685-943, pp. 197-208, 2024

Image Detection Analysis for Javanese Character Using YOLOv9 Models

Hari Suparwito[1*]

[1]*Faculty of Science and Technology, University of Sanata Dharma, Yogyakarta 55281, Indonesia*
[*]*Corresponding author: shirsj@jesuits.net*

(Received 05-06-2024; Revised 19-06-2024; Accepted 20-06-2024)

Abstract

The Javanese script needs to be digitized to improve access and usage, especially among younger generations. Digitizing Javanese characters is crucial for preserving Javanese culture and traditions in the long term. This study aims to detect and recognize Javanese characters using the YOLOv9 algorithm model, known for its ability to detect various object types, including Latin and non-Latin scripts. The dataset used consists of 85 images of complete Javanese script arranged in a 4x5 grid of different characters. The dataset was divided into a training dataset (75 images) and a validation dataset (10 images). All data pre-processing was done using Roboflow tools. Two experiments were conducted, varying the weights of the YOLOv9 algorithm model: YOLOv9-c and YOLOv9-c-converted. The research results showed that the YOLOv9-c model outperformed YOLOv9-c-converted, achieving a confidence level of over 80% and an mAP value of 0.95 in recognizing Javanese script images. In other words, the YOLOv9 algorithm model succeeded in detecting and recognizing Javanese scripts well.

Keywords: Deep learning, Image detection, Javanese character, Roboflow, YOLOv9

1 Introduction

The Javanese script, an ancient writing system utilized in Indonesia, particularly on the island of Java, holds immense cultural significance. It manifests itself in an array of written forms, including literature, documents, and inscriptions. Consequently, the digitization of the Javanese script becomes imperative, as it enables broader accessibility and utilization by the wider community, notably the younger generation [1]. The importance of digitizing the Javanese script is twofold: firstly, it aids in the preservation and transmission of Javanese cultural identity to future generations. Secondly, it has the potential to enhance public awareness and appreciation for Javanese culture, thereby stimulating efforts to safeguard Indonesia's cultural heritage. Ultimately, the digitalization of Javanese characters plays a critical role in ensuring the continuation and long-term viability of Javanese culture and traditions. Consequently, it becomes evident that the digitalization of the Javanese script contributes significantly to the preservation of Javanese culture and traditions [2].

With the increasing availability of digital images, there is a growing demand for efficient and accurate methods to detect Javanese script in these images. Traditional approaches to detect Javanese script involve the use of feature extraction and classification techniques such as convolutional neural network (CNN) and support vector machine (SVM) [3]. Previous studies have employed various methods and algorithms to detect Javanese script images. Katili et al. [4] proposed a novel character recognition workflow that combines Local Binary Pattern (LBP) and Information Gain. The LBP method is used to determine the texture or shape characteristics of an image. The feature selection algorithm utilizes Information Gain, while the classification method employs SVM, k-NN, and Naïve Bayes. This study demonstrates that the information gain method can improve accuracy by 2%. Additionally, the study compares SVM classification to k-NN and Naive Bayes classification methods, revealing that SVM classification yields the highest accuracy results (87.86%) with a ten-fold cross-validation and a 64x64 cell size. Another study utilizes the YOLOv5 model in conjunction with the U-Net model to assist in identifying the location of Javanese characters in an input document image [5]. Experimental results show that YOLOv5 model performs well in character location detection, with a loss rate of approximately 0.05. Similarly, the U-Net model achieves an accuracy range of 75% to 90% in predicting character regions. Although YOLOv5 model may not be able to perfectly detect all Javanese characters, using the U-Net model can enhance the detection rates by 1.2% [6]. The Linear Binary Pattern (LBP) feature extraction technique is also employed to process the dataset, aiding in the local characterization and description of image texture for the analysis and detection of Javanese scripts [2]. In this study, a pre-processing step is conducted using $r = 4$ and a thresholding method with $d = 0.3$. Subsequently, the K-Nearest Neighbour algorithm is used for further research. The results obtained from ten Javanese script word datasets indicate an average accuracy rate of 90.5%. The highest accuracy achieved was 100%, while the lowest accuracy was 50%.

In this study, the YOLOv9 algorithm model is selected due to its real-time object detection capabilities, which do not require explicit feature extraction or classification. The YOLOv9 algorithm model has been successfully used for various object detection tasks, including Latin and non-Latin scripts [7]. Its ability to handle variations in font style and image quality makes it well-suited for detecting Javanese script. Therefore, the YOLOv9 algorithm model will be employed in this study to detect Javanese script. To assess image detection, the shape of the Javanese script image will be altered. Typically, the Javanese script image is organized in a 4x5 matrix format, which will be utilized in the training and validation datasets. For testing purposes, different image formats such as 4x5 and 2x10 character image matrices, as well as single images, will be used. Detecting Javanese script is expected to contribute to the advancement of image detection applications in various fields, particularly in language learning and the transliteration of regional languages with special characters into Latin characters.

2 Material and Methods

2.1 Data acquisition

The data utilized in this study was acquired from the publicly available Kaggle dataset [8]. This dataset comprises 85 images, each encompassing 20 Javanese character images. It is divided into two subsets: the training dataset, consisting of 75 images, and the validation dataset, comprising 10 images. The 85 data images are formed in 4x5 arrangement of Javanese word images as shown in Fig. 1. All pre-processing tasks were undertaken by Roboflow tools.

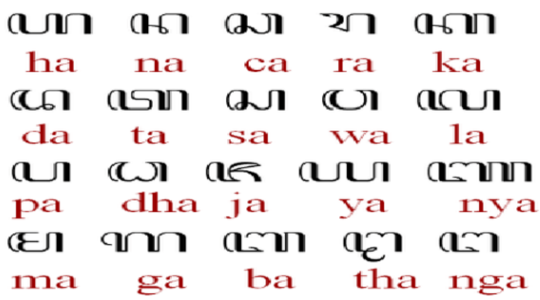

Figure 1. 4x5 arrangements of Javanese character images [9]

Conventionally, these 20 characters are systematically arranged in a 4x5 matrix. This research encompasses a collection of 85 images, wherein each image features a 4x5 matrix showcasing 20 Javanese characters. Subsequently, a data labelling procedure is implemented to assign a unique character label to each of the aforementioned characters, resulting in 20 distinctive character labels present within a solitary image. To put it differently, a single image comprises 20 script image labels.

2.2 YOLO model

The You Only Look Once (YOLO) algorithm model is widely recognized in the field of computer vision for its real-time processing capabilities and significant impact on object detection. Initially released in 2016 as YOLOv1, this algorithm introduced a single neural network that efficiently predicted bounding boxes and class probabilities from complete images in a single pass. This approach greatly accelerated the object detection process, distinguishing itself from traditional methods that employed multiple stages for detection [10].

Fig. 2 describes the CNN architecture used in YOLO (You Only Look Once). It consists of three key components: Convolutional Layers, Max Pooling Layers, and Fully Connected Layers. The Convolutional Layers extract features from the input image, reducing spatial dimensions and increasing the number of channels. ReLU activation functions introduce non-linearity. Max Pooling Layers are used to down-sample the

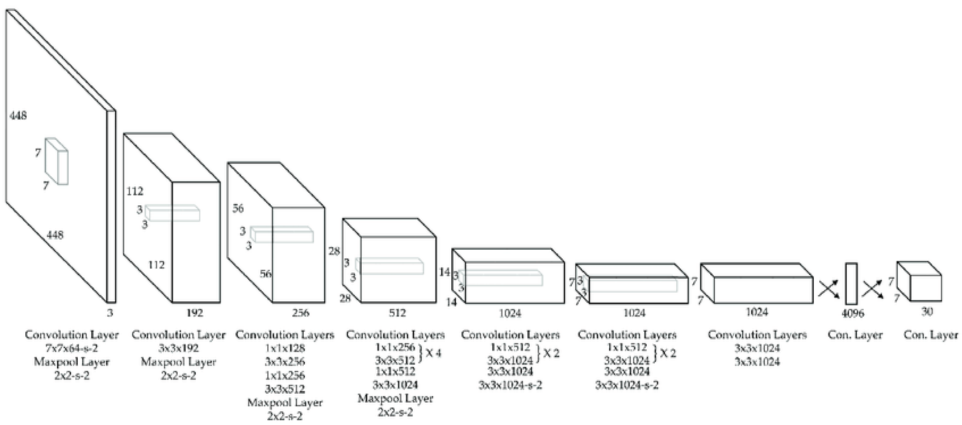

Figure 2. The basic CNN algorithm that is in YOLO model architecture [11]

feature maps, further reducing spatial dimensions and decreasing the number of parameters in the network. The final stage of the network is made up of Fully Connected Layers, which predict class probabilities and bounding box coordinates. These layers make predictions based on the features extracted by the convolutional and pooling layers.

The network uses a combination of techniques such as batch normalization and dropout to regularize the model and prevent overfitting. The CNN architecture is designed to be efficient and fast, allowing it to process images in real-time. The network is trained using a combination of techniques such as data augmentation and transfer learning to improve its performance on object detection tasks.

As of the most recent version, YOLOv9 model, the YOLO series continues to be a leading force in the field of object detection research, consistently pushing the boundaries of real-time detection tasks in terms of speed, accuracy, and efficiency. The progression of YOLO from its origin to the present day exemplifies the ongoing advancements in deep learning and computer vision, fundamentally shaping the manner in which we comprehend and engage with visual data across diverse applications.

2.3 Methods

The following diagram (Fig. 3) illustrates the process carried out in the study of detection and recognition of Javanese script using the YOLOv9 algorithm model. This study commenced by acquiring data in the form of images containing Javanese scripts. To facilitate data collection, we utilized a publicly available Kaggle dataset. For data pre-processing, the Roboflow tools were employed, specifically for labelling and segregating the data into training and validation datasets. Subsequently, the YOLOv9

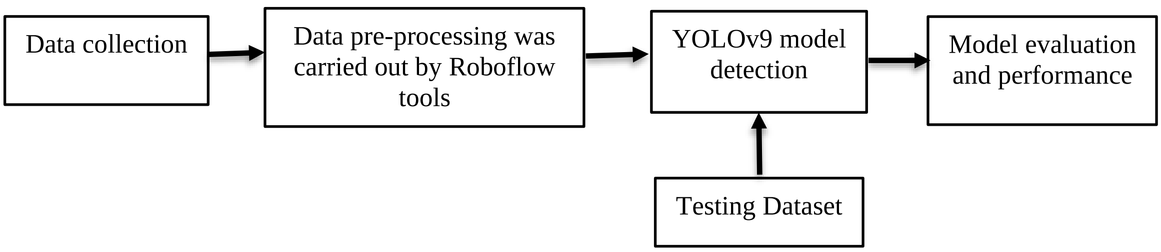

Figure 3. The steps in Javanese script images detection analysis using YOLOv9

algorithm was utilized for modelling. The resulting model was then assessed using the validation dataset. The final step involved employing a distinct set of testing dataset, which differed from both the training and validation dataset, to evaluate the model's capability to accurately detect and identify Javanese script objects. Notably, the testing dataset incorporated diverse variations of Javanese script that were not present in the training and validation datasets.

2.4 Experiments

Two experiments were conducted to compare the utilization of learning rate and weight parameters in YOLOv9 model, aiming to investigate their impact on the model's performance. The learning rate function, a crucial hyperparameter in deep learning, governs the extent to which the model adjusts its weights during the training process. It determines the step size for each iteration of the optimization algorithm, enabling the model to converge towards the minimum loss function. Given that the learning rate function can influence the weights, this study examines its effect on the weight parameter, evaluating the outcome using mean Average Precision (mAP) on images tested with the training model. Therefore, this study would compare the performance of 2 weight parameters of YOLOv9, namely YOLOv9-c and YOLOv9-c-converted. The remaining parameters, including activation function, dropout, and epochs, adhere to the default settings of the YOLOv9 algorithm. Table 1. showed the YOLOv9 parameters used in the experiments.

Table 1. YOLOv9 parameters used in experiments

Parameters	Experiment 1	Experiment 2
Weight	YOLOv9 – c	YOLOv9 – c – converted
Epoch	100	
Dropout	0.5	
Optimizer	Adam	
Batch size	8	
Learning rate	0.001	

The YOLOv9 model is trained using 75 Javanese script images, followed by validation with ten additional images. In order to optimize the results, the model is trained for 100 epochs, utilizing a batch size of 8 and learning rates of 0.001. By analyzing the changes in learning rate and corresponding mAP results, it becomes possible to determine which parameters yield the highest accuracy.

3 Results and Discussions

After training the model, we proceeded to evaluate its performance using the validation set. The model's accuracy in detecting Javanese characters was assessed using metrics such as F1-score and mAP (mean Average Precision). Two distinct weight variations provided by YOLOv9, namely YOLOv9-c and YOLOv9-c-converted, were employed in this study. Table 2 presents the mAP values and F1-scores obtained from these two weight configurations. However, it is important to note that the observed differences in both metrics are not statistically significant. The results of the model evaluation are presented in the following table.

Table 2. The mAP and F1-confidence results

Weight parameters	mAP50	mAP50-95	F1 score
YOLOv9-c	0.95	0.72	0.943
YOLOv9-c-converted	0.92	0.71	0.932

The graph presented in Fig.4 displays the F1-score and Confusion Matrix results for all Javanese characters at varying confidence levels. Only the most optimal results are presented in this analysis. The F1-confidence score in YOLO (You Only Look Once) model is a measure of a model's accuracy that considers both precision and recall. It is the harmonic mean of precision and recall, providing a balance between the two. The F1-confidence score is plotted against different confidence thresholds in the F1 Confidence Curve, which helps in determining the optimal confidence threshold for making predictions. A higher F1 score indicates better performance, and the confidence threshold at which the F1 score is maximized is often considered the optimal threshold. Fig. 4 showed that for all classes (20 classes) the higher value is 0.89 at 0.552. This means that with a confidence value of 0.55, a good average precision value will be obtained for all classes (for all characters). However, it can also be seen that for each class, the F1 value and the confidence value are different for each script. If we focus on the confidence value of 0.55, it can be seen that the characters "ba", "sa", and "la" give the three lowest F1 values compared to other characters. Next, we also computed the confusion matrix for each character in the Javanese script.

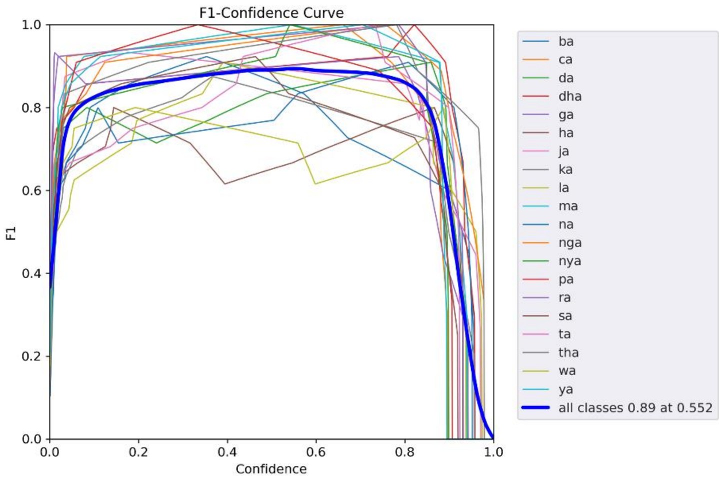

Figure 4. F1-confidence graph for each Javanese character and all classes using the best training model

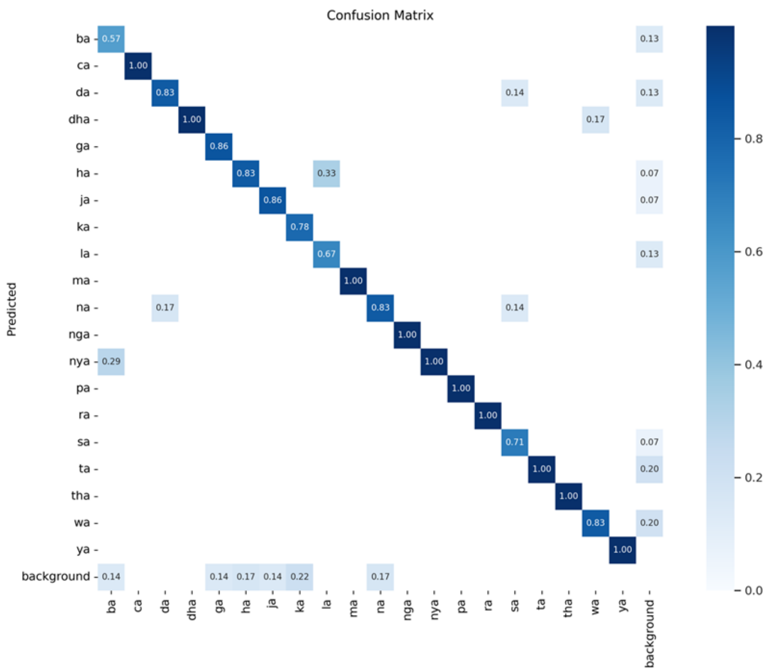

Figure 5. The Confusion Matrix for each Javanese character

A confusion matrix's diagonal values represent the number of instances in which the predicted label matches the true label. In other words, they display correctly classified instances. The higher the diagonal values, the better the model's performance at correctly identifying classes. Fig. 5 shows that the three lowest diagonal values are "ba," "la," and "sa," with values of 0.57, 0.67, and 0.71, respectively. This means that the model is less capable of predicting that three Javanese scripts. Otherwise, the model can accurately predict other Javanese scripts with a value greater than 0.8. The character "ba" received a low score of 0.57 in confusion matrix table because the shape of the character "ba" is similar to the shape of the character "nya". Similarly, the character "sa", in the confusion matrix table is equated with the characters "na" and "da" because of the similarity of the shape of each character

To successfully apply the trained YOLOv9 model for Javanese character recognition, we utilized the model to detect and recognize Javanese characters in various image formats, including a 4x5 matrix, a 2x10 matrix, and two individual images. The subsequent images display the obtained outcomes.

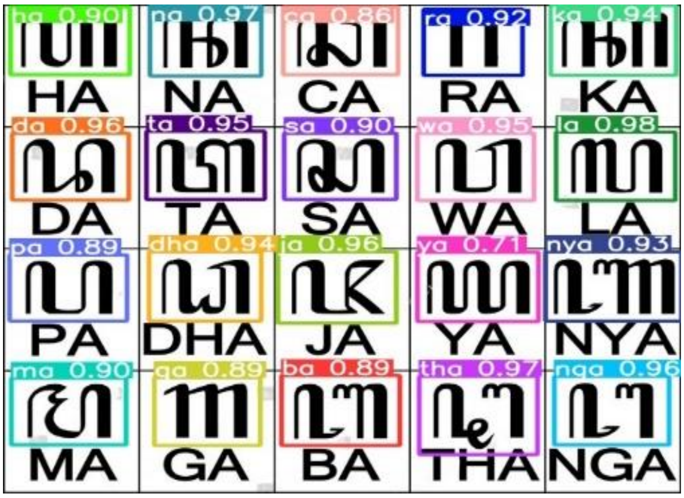

Figure 6. The detection model was implemented on the Javanese script arranged in 4x5 matrix

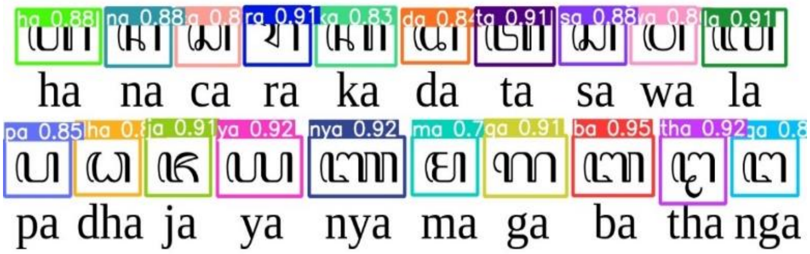

Figure 7. The detection model was implemented on the Javanese script arranged in 2x10 matrix

Figure 8. The detection model was implemented to the two Javanese scripts in the two single forms

We assessed the performance of the Javanese character reading model by comparing the recognized text with the ground truth text. In the first image (Fig. 6), the model achieved a good confidence level. However, only one character provided the lowest level i.e., "ya" obtained a confidence level at 0.71 while the highest confidence level was achieved for the character "la" at 0.98. The model seems to have difficulty predicting the image "ya" because it is similar to some other characters, such as "pa", "ba", and "ga" even though the format of the printed image is a 4x5 matrix similar to the printed image in training dataset. Additionally, in the second image (Fig.7), where the images were arranged in a 2x10 format, the confidence levels ranged between 0.8 and 0.95. For the third test, we examined two single form images (Fig.8). The character "ta" had a confidence value of 0.96, whereas the character "sa" had a value of 0.71. If we look at the confusion matrix result, it showed that the character "sa" got a fairly low result of 0.71 and this result seems affecting the prediction of the testing data in the single image. It can be seen here that the YOLOv9 model less capable to identified the Javanese characters that are in the similar shapes. The superior performance can be attributed to the fact that the training data images arranged in a 4x5 matrix yielded better average results compared to other arrangements.

4 Conclusions

The YOLOv9 algorithm model was used to detect Javanese script. Two studies were carried out to compare the effect of weight parameters on YOLOv9. The algorithm's ability to recognise Javanese characters was tested by comparing the recognised text to the ground truth text. Confidence levels ranged between 0.8 and 0.95 obtained when photos were grouped in a 2x10 matrix. Another trial with single form photos yielded a confidence value of 0.96 for "ta" and 0.71 for "sa". Image's data testing in a 4x5 matrix generated generally better results than alternative formats. In future studies, addressing the variability among Javanese scripts and the opportunity to use handwriting format remains an open and active research topic. Moreover, this study could be used as the development of the machine learning algorithm by implementing the model in different local character languages.

Acknowledgements

We express our gratitude to the Department of Informatics at University of Sanata Dharma for granting us permission to utilize their workstation facilities and the Artificial Intelligence Laboratory for the purpose of conducting this study.

References

[1] D. Iskandar, S. Hidayat, U. Jamaludin, and S. M. Leksono, "Javanese script digitalization and its utilization as learning media: an etnopedagogical approach," *International Journal of Mathematics and Sciences Education*, vol. 1, no. 1, 2023, pp. 21–30.

[2] A. Susanto, I. U. W. Mulyono, C. A. Sari, E. H. Rachmawanto, and R. R. Ali, "Javanese character recognition based on k-nearest neighbor and linear binary pattern features," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 2022, pp. 309–316.

[3] F. T. Anggraeny, Y. V. Via, and R. Mumpuni, "Image preprocessing analysis in handwritten Javanese character recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, 2023, pp. 860–867.

[4] I. F. Katili, M. A. Soeleman, and R. A. Pramunendar, “Character Recognition of Handwriting of Javanese Character Image using Information Gain Based on the Comparison of Classification Method,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, 2023, pp. 193–200.

[5] M. A. Rasyidi, T. Bariyah, Y. I. Riskajaya, and A. D. Septyani, “Classification of handwritten Javanese script using random forest algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, 2021, pp. 1308–1315.

[6] A. Prasetiadi, J. Saputra, I. Kresna, and I. Ramadhanti, “YOLOv5 and U-Net-based Character Detection for Nusantara Script,” *Jurnal Online Informatika*, vol. 8, no. 2, 2023, pp. 232–241.

[7] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[8] H. Sugiharto, “Aksara Jawa YOLO v5 Dataset.” [Online]. Available: https://www.kaggle.com/datasets/hermansugiharto/aksara-jawa-yolo-v5-dataset. [Accessed: May 04, 2024]

[9] A. Budi, “Hanacaraka dan Makna Bijak di Baliknya.” [Online]. Available: https://www.goodnewsfromindonesia.id/2017/01/23/hanacaraka-dan-makna-bijak-di-baliknya. [Accessed: May 15, 2024]

[10] M. Hussain, “YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection,” *Machines*, vol. 11, no. 7, 2023, p. 677.

[11] C. Koylu, C. Zhao, and W. Shao, “Deep neural networks and kernel density estimation for detecting human activity patterns from geo-tagged images: A case study of birdwatching on flickr,” *ISPRS Int J Geoinf*, vol. 8, no. 1, 2019, p. 45.