Comparative Transfer Learning Techniques for Plate Number Recognition

by Rizki Rafiif Amaanullah

Comparative Transfer Learning Techniques for Plate Number Recognition

Rizki Rafiif Amaanullah[1], Rifqi Akmal Saputra[2], Faisal Dharma Adhinata[3], Nur Ghaniaviyanto Ramadhan[4]
[1,2,3,4]Department of Software Engineering, Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Indonesia
email: 19104010@ittelkom-pwt.ac.id, 19104022@ittelkom-pwt.ac.id, faisal@ittelkom-pwt.ac.id, ghani@ittelkom-pwt.ac.id

***Abstract*— Monitoring vehicle activity both on the highway and in certain places such as parking lots needs to be done if there is a specific incident. Unexpected events such as accidents or vehicle theft may occur anytime. Therefore, tracking through number plate recognition has become something important and has become a hot topic with the various methods used. Previous research used machine learning techniques to recognize characters on number plates. The use of this technique has not produced optimal accuracy. Therefore, we propose using transfer learning techniques to achieve better accuracy results. This research evaluated three transfer learning models, namely DenseNet121, MobileNetV2, and NASNetMobile models. The experiment in this research was carried out using the data on number plates in the parking lot. The accuracy calculation counted the number of correctly recognized characters divided by the total characters on the number plate. The experimental results show that the DenseNet121 model produced the best accuracy, 96.42%. Differences in number plate writing style also affected the accuracy results. This research could provide insight into the use of transfer learning techniques in the case of number plate recognition.**

Keywords* — *DenseNet121, MobileNetV2, NASNetMobile, Number plate recognition

I. INTRODUCTION

The vehicle number plate is one of the identities of motorized vehicles. The number plate material is metal or plastic with the identity number mounted on the front and back of the vehicle. The vehicle's identity is the main clue when an incident occurs on the road, such as an accident or traffic violation. The authorities will identify the vehicle number plate to see the driver's identity data when something happens on the road [1]. The authorities' identification process carried out manually requires the authorities to be at the case scene. Currently, many Closed-Circuit Television (CCTV) cameras are installed on the road that monitor activity on the road [2]. This CCTV camera can be used for automatic number plate recognition that utilizes Artificial Intelligence technology.

Research on vehicle number plate recognition is beneficial for developing smart cities, such as smart transportation systems, missing vehicle searches, traffic monitoring, city management, and toll entry payments. Several previous studies did the number plate recognition using machine learning methods [3][4][5] and deep learning [6][7]. Machine learning techniques have not yet achieved maximum accuracy for plate number recognition. Research by Gunawan et al. [8] used the K-NN method to identify Indonesian vehicle number plates. The results showed that not all characters were detected, with an average accuracy of 92.86%. Then, the use of the Random Forest method also has not produced optimal accuracy, which is 90.9% [9]. Other researchers use transfer learning methods to detect number plates [10]. This research resulted in 99% accuracy in the training validation process. However, an evaluation has not been carried out using the vehicle plate character matching one by one. Therefore, we propose a deep learning approach to improve the accuracy of the results. The proposed research aims to obtain a more accurate vehicle number plate recognition.

The feature extraction and classification stages are the vehicle number plate recognition that affects the accuracy value. For character recognition, both numbers and letters, there are 36 classes with ten numeric characters and 26 letter characters. To distinguish these classes, a method that can classify them is needed. One of the powerful techniques for classifying many classes is the transfer learning technique [11]. Previous studies compared DenseNet121 and NASNetMobile transfer learning models for COVID-19 detection using Chest X-ray images [12]. The results showed that the accuracy values were not much different. Similarly, research by Agarwal et al. [13] also detected COVID-19 using the DenseNet121, MobileNetV2, and NASNetMobile methods. The research results are also not too much different in the value of accuracy. Therefore, the proposed research used DenseNet121, MobileNetV2, and NASNetMobile model transfer learning techniques. This model is also used because the three models are not too large in the number of parameters. We hope that the results of this research can provide insight into the transfer learning method for vehicle plate character recognition.

II. PROPOSED METHOD

The character recognition system on motorized vehicle license plates starts from the character training stages of letters and numbers. The character dataset was pre-processed first to adjust to the size of the training process using transfer learning techniques [14]. Then, to increase the diversity of the data, we also used data augmentation. The dataset that had been pre-processed was divided into training data and testing data. The next stage was feature extraction using DenseNet121, MobileNetV2, or NASNetMobile. The result of feature extraction was a model used to match the features with the vehicle's character. Then, the vehicle number plate data was obtained from the image at the testing stage. The image data obtained was searched for the number plate area. The detected number plate was segmented using a bounding box. The segmentation results were used for matching with the feature extraction model. The final result was the character on the vehicle plate. Figure 1 shows the stages of the vehicle plate number recognition.

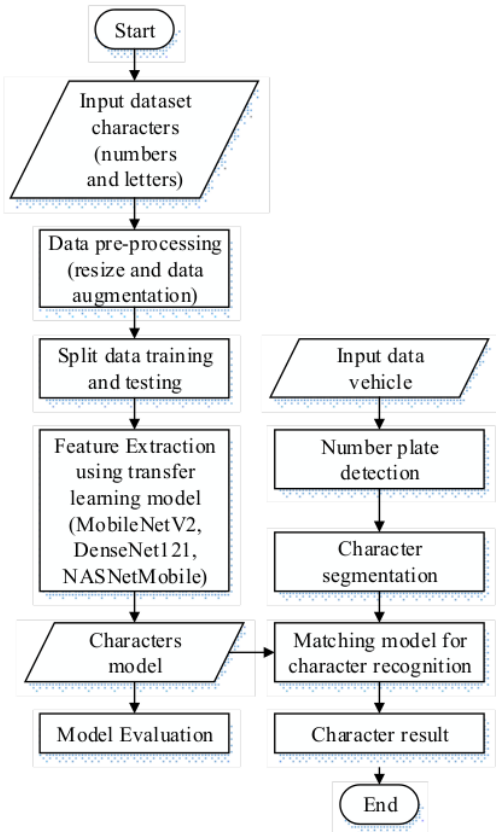

Fig. 1. Stages of vehicle plate number recognition

A. Dataset

The character dataset used in this research was obtained from Kaggle [15]. There were 36 characters with ten numeric characters and 26 letter characters. The data used was in black and white, with the character color being white and the background black. The total data used in this research were 37,623 data. Figure 2 shows an example of the data used in this research. Then, for the testing data, we used motor vehicle image data. The image was shot perpendicular to the vehicle object. We used the image of Indonesian vehicle plates in the parking lot.

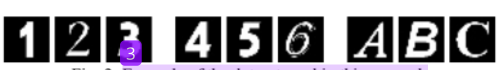

Fig. 2. Example of the dataset used in this research

B. Data pre-processing

Before being used for the feature extraction stage, the dataset was carried out in a pre-processing stage to adjust to the transfer learning model. The pre-processing stage was done by resizing to 224 x 224 to fit the pre-trained transfer learning model. Then, we also used data augmentation to add diversity to the data used in the feature extraction stage. We used the ImageDataGenerator library [16], provided by the Python programming language for data augmentation.

C. Feature extraction

Traditional feature extraction cannot automatically determine what features are present in an object. When the number of available features is large, the feature extraction method used must also have many parameters to extract these features. Therefore, deep learning provides a breakthrough by automatically extracting features on image objects. The popularity of deep learning continues to grow as large amounts of data become available, namely ImageNet [17]. However, the use of large amounts of data is challenging to obtain. Therefore, there is an alternative called transfer learning [18]. Transfer learning is a training method in which the model has been previously trained, and the results can be used for other case datasets. Many transfer learning models, including MobileNetV2, NASNetMobile, and DenseNet121. MobileNetV2

1) MobileNetV2: is a development model of MobileNet. In the image classification experiment using ImageNet, MobileNetV2 shows better accuracy results than MobileNetV1 with fewer parameters [19]. In MobileNetV2, there are two types of blocks, namely, the residual block with stride one and stride two. The two blocks are arranged to form the MobileNetV2 architecture. Figure 3 shows the architecture of MobileNetV2.

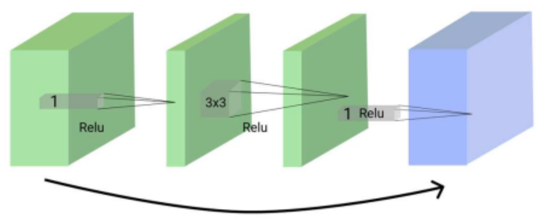

Fig. 3. MobileNetV2 architecture that skips connection

2) NASNet: is a model based on research by finding the best block architecture in a small dataset, then copying the best-found architecture for use in a larger dataset, namely ImageNet. NASNetMobile is a smaller NASNet architecture with the same number of parameters as MobileNet but with better accuracy performance [20]. The NASNet architecture consists of 2 main blocks: normal cells and reduction cells. The correct layer arrangement for the two cells is searched using a recurrent neural network. Figure 4 shows the NASNet search space.

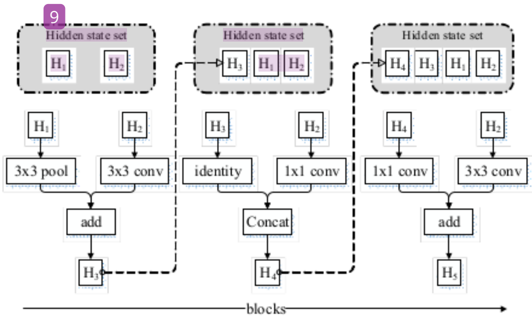

Fig. 4. NASNet search space

3) DenseNet121: is an architectural model with unique characteristics called dense blocks where in this block, each layer is directly connected to all layers. A layer takes input from the output of all previous layers and provides output for all layers, allowing the network to be leaner. It is different from the traditional convolution layer, where a layer takes input from the previous layer and provides output for the next layer [21]. Figure 5 shows the layers in the DenseNet201 model.

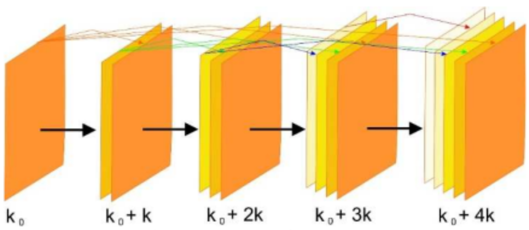

Fig. 5 DenseNet201 layer structure

D. System Evaluation

The evaluation of the system in this research was done using the image of a vehicle with a number plate. The vehicle image detects the position of the number plate. Then, the detected number plate was executed, not a segmentation process to divide each character on the number plate. The character on the license plate was matched with the feature extraction results of the MobileNetV2, NASNetMobile, or DenseNet121 models. Evaluation was done by counting the correct number of characters divided by the number on the license plate. We used 15 number plate data, so the final accuracy was obtained from the average accuracy of each number plate.

III. Result and Discussion

This section discusses the results of making a number plate recognition system using transfer learning. In this research, the program was created using the Python programming language version 3. The discussion was divided into two stages: the training and testing stages.

A. Training stage

The training stage is the stage of model formation for vehicle number plate recognition. The configuration of the transfer learning method in this research used transfer learning in the feature extraction section or the base model. The classification layer uses Dense and Dropout layers. This research used a Dropout value of 0.5. Then in the loss function, we used categorical crossentropy with the Adam optimizer. We used the EarlyStopping function to stop the training process so that the resulting graph did not look overfitting. Figure 6 shows a graph of the training results. In both models' graphs generated from the training process using MobileNetV2, DenseNet121, and NASNetMobile overfitting did not occur. The training and validation accuracy graphs from epoch five seem to be parallel. Similarly, the training and validation loss graphs were also parallel.

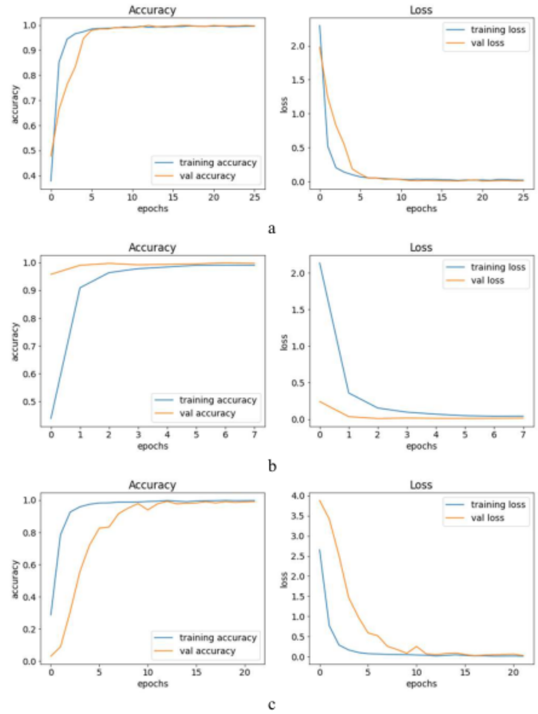

Fig. 6. Training result on accuracy and loss values, a) MobileNetV2, b) DenseNet121, c) NASNetMobile

The EarlyStopping function that we used monitored the validation loss value with a value of 5 so that if there was no better change of 5 epochs, the training process would be stopped. Table 1 shows the results of training using the three models. The DenseNet121 model had the fastest training process because it only had up to 10 epochs. Meanwhile, the NASNetMobile model reached the 22nd epoch. Then the MobileNetV2 model reached the 26th epoch. When compared to the validation loss values of the three models, the results of the MobileNetV2 model were the best. We saved these three models for testing using license plate testing data.

TABLE I. Result of Validation Loss Training Process

Epoch	Validation loss		
	MobileNetV2	**DenseNet121**	**NASNetMobile**
2	1.2397	0.0359	3.4199
4	0.5494	0.0175	1.4670
6	0.1066	**0.0116**	0.5874
8	0.0499	0.0162	0.2547
10	0.0339	0.0144	0.0813
12	0.0115	-	0.0674
14	0.0127	-	0.0757
16	0.0079	-	0.0510
18	0.0060	-	**0.0373**
20	0.0230	-	0.0513
22	**0.0050**	-	0.0449
24	0.0109	-	-
26	0.0069	-	-

B. Testing stage

In the testing phase, we used data taken in the parking lot. The testing data consisted of 15 number plates, both car and motorcycle license plates in Indonesia. The detected number plates were matched using the three training models. Table 2 shows the results of the number plate recognition accuracy. The percentage was obtained by counting the correct characters divided by the number of characters on the license plate. Then, the average accuracy is obtained from the average accuracy in each model.

TABLE II. RESULT OF PLATE NUMBER RECOGNITION

No	Image	Model					
		MobileNetV2	**%**	**DenseNet121**	**%**	**NASNetMobile**	**%**
1		R1374JR	100	R1374JR	100	R1374JR	100
2		AE1922PO	87.5	AE1922P0	87.5	AE1922PQ	100
3		H1143WF	100	H1143NF	85.7	H1143WF	100
4		D1297ACW	100	D1297ACN	87.5	D1297ACH	87.5
5		K1536VO	85.7	K1536VD	100	K7536VD	85.7
6		81991J8	71.4	B1991J6	85.7	81991J0	85.7
7		BP1O87FZ	87.5	BP1087FZ	100	BP1087FZ	100
8		R2699RH	100	R2699RH	100	R2699RH	100
9		81539KFH	87.5	B1539KFH	100	B1539KFH	100
10		A84252IZ	87.5	AB4252IZ	100	AB4252IZ	100
11		B457UTY	100	B457UTY	100	6457UTY	85.7
12		B9320VUA	100	B9320VUA	100	B9320VUA	100
13		B1690BAC	100	B1690BAC	100	B1690BAC	100
14		B1820UZN	100	B1820UZN	100	61820UZN	87.5
15		816T7MB	87.5	B1611MB	100	B1611MB	100
Average Accuracy (%)			**92.97**		**96.42**		**95.47**

Table 2 shows that the DenseNet121 model produced the best average accuracy than the MobileNetV2 and NASNetMobile models. Several characters were detected incorrectly, for example, the letter D with O, and B with the number 8. When viewed in general, the characters have almost the same shape. The dataset used was also in black and white. No color stood out from each character, resulting in errors when matching. However, overall, the accuracy produced by the DenseNet121 model was better than the use of machine learning techniques [8][9]. This accuracy result is also due to the influence of the number of parameters on the DenseNet121 model than on the MobileNetV2 and NASNetMobile models. Future research can add datasets with various writing styles to increase the accuracy of results.

IV. Conclusion

Vehicle number plates need to be identified when unexpected events occur on the streets. Many CCTV cameras have been installed on the road and in the parking lot to monitor driving activities. Previous research used machine learning techniques to perform number plate recognition. The use of machine learning has not yet produced optimal accuracy. This research used three transfer learning techniques for number plate recognition. The results showed that the DenseNet121 model yielded an accuracy of 96.42%. This research still has shortcomings in recognizing the number plates with a variety of character models. Therefore, it is recommended that future research increase the diversity of data to increase accuracy.

References

[1] O. Shobayo, A. Olajube, N. Ohere, M. Odusami, and O. Okoyeigbo, "Development of Smart Plate Number Recognition System for Fast Cars with Web Application," *Applied Computational Intelligence and Soft Computing*, vol. 2020, 2020, doi: 10.1155/2020/8535861.

[2] U. Enggarsasi and N. K. Sa'Diyah, "Interaction of traffic police with motorized vehicle control by CCTV," *IOP Conference Series: Materials Science and Engineering*, vol. 434, no. 1, 2018, doi: 10.1088/1757-899X/434/1/012264.

[3] R. Kumari and S. Prakash, "A Machine Learning Algorithm for Automatic Number Plate Recognition," *International Journal of Computer Applications*, vol. 174, no. 1, pp. 6–9, 2017, doi: 10.5120/ijca2017915297.

[4] Sugeng and E. Y. Syamsuddin, "Designing Automatic Number Plate Recognition (ANPR) Systems Based on K-NN Machine Learning on the Raspberry Pi Embedded System," *Jurnal Teknik Elektro dan Vokasional*, vol. 5, no. 1, pp. 19–26, 2019.

[5] D. Habeeb *et al.*, "Deep-Learning-Based Approach for Iraqi and Malaysian Vehicle License Plate Recognition," *Computational Intelligence and Neuroscience*, vol. 2021, 2021, doi: 10.1155/2021/3971834.

[6] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning," *IOP Conference Series: Materials Science and Engineering*, vol. 1084, no. 1, p. 012027, 2021, doi: 10.1088/1757-899x/1084/1/012027.

[7] S. Alghyaline, "Real-time Jordanian license plate recognition using deep learning," *Journal of King Saud University - Computer and Information Sciences*, no. xxxx, 2020, doi: 10.1016/j.jksuci.2020.09.018.

[8] D. Gunawan, W. Rohimah, and R. F. Rahmat, "Automatic Number Plate Recognition for Indonesian License Plate by Using K-Nearest Neighbor Algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 648, no. 1, 2019, doi: 10.1088/1757-899X/648/1/012011.

[9] Z. Akhtar and R. Ali, "Automatic Number Plate Recognition Using Random Forest Classifier," *SN Computer Science*, vol. 1, no. 3, pp. 1–9, 2020, doi: 10.1007/s42979-020-00145-8.

[10] S. Bismantoko, M. Rosyidi, U. Chasanah, A. Haryono, and T. Widodo, "A Comparison Pre-Trained Models for Automatic Indonesian License Plate Recognition," *Majalah Ilmiah Pengkajian Industri*, vol. 15, no. 1, 2021, doi: 10.29122/mipi.v15i1.4738.

[11] A. Mustafid, M. M. Pamuji, and S. Helmiyah, "A Comparative Study of Transfer Learning and Fine-Tuning Method on Deep Learning Models for Wayang Dataset Classification," *IJID (International Journal on Informatics for Development)*, vol. 9, no. 2, pp. 100–110, 2020, doi: 10.14421/ijid.2020.09207.

[12] M. S. Boudrioua, "COVID-19 Detection from Chest X-Ray Images Using CNNs Models: Further Evidence from Deep Transfer Learning," *SSRN Electronic Journal*, vol. 4, pp. 2–5, 2020, doi: 10.2139/ssrn.3630150.

[13] V. Agarwal, M. C. Lohani, A. S. Bist, E. P. Harahap, and A. Khoirunisa, "Analysis Of Deep Learning Techniques For Chest X-Ray Classification In Context Of Covid-19," *ADI Journal on Recent Innovation*, vol. 3, no. 2, pp. 208–216, 2022.

[14] F. D. Adhinata and A. Junaidi, "Gender Classification on Video Using FaceNet Algorithm and Supervised Machine Learning," *International Journal of Computing and Digital Systems*, vol. 11, no. 1, 2022.

[15] N. Gaur, "dataset_characters | Kaggle," *Kaggle.com*, 2021. https://www.kaggle.com/nainikagaur/dataset-characters (accessed Mar. 10, 2022).

[16] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[17] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Computer Science*, vol. 2, no. 6, pp. 1–20, 2021, doi: 10.1007/s42979-021-00815-1.

[18] F. D. Adhinata, G. F. Fitriana, A. Wijayanto, M. Pajar, and K. Putra, "Corn Disease Classification using Transfer Learning and Convolutional Neural Network," vol. 9, no. 2, pp. 1–7, 2021.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[20] K. Radhika, K. Devika, T. Aswathi, P. Sreevidya, V. Sowmya, and K. P. Soman, *Performance Analysis of NASNet on Unconstrained Ear Recognition*, vol. 871. Springer International Publishing, 2020.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

Comparative Transfer Learning Techniques for Plate Number Recognition

ORIGINALITY REPORT

10%	6%	8%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Maria Susan Anggreainy, Ajeng Wulandari, Abdullah M. Illyasu. "Pneumonia Detection using Dense Convolutional Network (DenseNet) Architecture", 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2021 Publication	2%
2	journal.uad.ac.id Internet Source	2%
3	Faisal Dharma Adhinata, Gita Fadila Fitriana, Aditya Wijayanto, Muhammad Pajar Kharisma Putra. "Corn Disease Classification Using Transfer Learning and Convolutional Neural Network", JUITA: Jurnal Informatika, 2021 Publication	1%
4	ejurnal.teknokrat.ac.id Internet Source	1%
5	Rafi Surya, Elliana Gautama. "Cassava Leaf Disease Detection Using Convolutional Neural	1%

Networks", 2020 6th International Conference on Science in Information Technology (ICSITech), 2020
Publication

6	Olamilekan Shobayo, Ayobami Olajube, Nathan Ohere, Modupe Odusami, Obinna Okoyeigbo. "Development of Smart Plate Number Recognition System for Fast Cars with Web Application", Applied Computational Intelligence and Soft Computing, 2020 Publication	1%
7	site.ieee.org Internet Source	<1%
8	Neelesh Karthikeyan. "Review of Deep Transfer Learning Models for Image Classification", International Journal of Recent Contributions from Engineering, Science & IT (iJES), 2022 Publication	<1%
9	S. V. Kogilavani, J. Prabhu, R. Sandhiya, M. Sandeep Kumar et al. "COVID-19 Detection Based on Lung Ct Scan Using Deep Learning Techniques", Computational and Mathematical Methods in Medicine, 2022 Publication	<1%
10	ijircce.com Internet Source	<1%

11	www.hindawi.com Internet Source	<1 %
12	cisak.perpika.kr Internet Source	<1 %
13	"Advances in Computer Vision", Springer Science and Business Media LLC, 2020 Publication	<1 %
14	beei.org Internet Source	<1 %
15	ejournal.ikado.ac.id Internet Source	<1 %
16	www.astesj.com Internet Source	<1 %
17	www.ijrte.org Internet Source	<1 %

Exclude quotes On
Exclude bibliography On
Exclude matches Off