

Indonesian Plate Number Identification Using YOLACT and Mobilenetv2 in the Parking Management System

I Kadek Gunawan¹, I Putu Agung Bayupati², Kadek Suar Wibawa³, I Made Sukarsa⁴, Laurensius Adi Kurniawan⁵

^{1,2,3,4,5}Information Technology, Universitas Udayana, Indonesia

¹gunawankadek@student.unud.ac.id, ²bayupati@unud.ac.id, ³suar_wibawa@unud.ac.id, ⁴sukarsa@unud.ac.id, ⁵adikurniawan@student.unud.ac.id

Abstract - A vehicle registration plate is used for vehicle identity. In recent years, technology to identify plate numbers automatically or known as Automatic License Plate Recognition (ALPR) has grown over time. Convolutional Neural Network and YOLACT are used to do plate number recognition from a video. The number plate recognition process consists of 3 stages. The first stage determines the coordinates of the number plate area on a video frame using YOLACT. The second stage is to separate each character inside the plate number using morphological operations, horizontal projection, and topological structural. The third stage is recognizing each character candidate using CNN MobileNetV2. To reduce computation time by only take several frames in the video, frame sampling is performed. This experiment study uses frame sampling, YOLACT epoch, MobileNet V2 epoch, and the ratio of validation data as parameters. The best results are with 250ms frame sampling succeed to reduce computational times up to 78%, whereas the accuracy is affected by the MobileNetV2 model with 100 epoch and ratio of split data validation 0,1 which results in 83,33% in average accuracy. Frame sampling can reduce computational time however higher frame sampling value causes the system fails to obtain plate region area.

Keywords: ALPR, convolutional neural network, frame sampling, horizontal projection, YOLACT

I. INTRODUCTION

The number of vehicles in the world, especially in Indonesia, is increasing annually, which has an impact on the need for parking spaces is also increasing in particular urban areas. Indonesia's statistical agency recorded the number of vehicle ownership in Indonesia in 2018 as many as 146,858,759 vehicles, an increase of nearly 8 million vehicles compared to the previous year [1]. This rapid growth should be followed by the development of sophisticated technology in detecting vehicle number plates automatically. ALPR technology

can be used to facilitate vehicle surveillance in the parking area can be with ALPR technology. ALPR (Automatic License Plate Recognition) is the realm of computer vision capable of recognizing vehicle registration plate number from an image or video. ALPR technology is generally divided into 3 processes, license plate localization, character segmentation, and character recognition. Each country in the world has a different shape and character in its vehicle registration plate number, based on government rule. Indonesia possesses several types of the vehicle registration plate, the black color with white lettering is intended for a personal vehicle, yellow color with black lettering for the public transport vehicle, red color intended for government vehicle, and particular plate number for military and police departments which has different letters and bearing the logo of the relevant agencies.

Many factors affect ALPR recognition accuracies such as light ambient, image resolution, camera angles, etc. Previously in ALPR researches has been done, there were 2 methods Generally used such as morphological approach [2–5] and Deep Learning approach [6–9]. The deep learning approach is often used recently because has better accuracy than the edge detection approach, Despite the fact deep learning approach has poor performance speed. The previous research about ALPR consisted of several different research focuses such as plate number detection, character segmentation, character recognition, or includes all domains that have been mentioned.

The research about ALPR which tried to segments characters in Argentina plate numbers based on color. Several preprocessing phases are used such as resize the input image, erosion with a 3x3 kernel, and the weighted sum method is performed to enhances the character's white strokes. Image binarization and image projection are used to segments each character. the result obtained 96,49% success rates; the average

computation is 6,6 ms with resolutions 120 x 30 pixels [10]. The ALPR research [9] attempted to segments Bhutan plate numbers and vehicle with Yolo. localization of plate number considered if it situates inside localization of vehicle. The best result obtained with 5000 epochs was 98,6% mAP. Ref. [11] performed Myanmar vehicle license plate detection with edge detection and morphological operations. The data used are in the form of images of the front and backside of vehicles. from 40 image tests, the system performs only 1 failed detection.

The research [8] performed plate number detection, segmentation, and character recognition using Mask Region Convolutional Neural Network (Mask R-CNN). This study used GoogLeNet [12] without inception module as feature extraction, and swish activation function. The First process is to separate the plate number from an image by a model that was trained previously with a positive and negative sample of the plate number. The second process is separating the character inside plate number using a trained model with character and non-character data. The last process is recognized each character with the model that was trained with 38-character classes (0-9, A-Z+ one Arabic word + negative class). 4 datasets from several countries used to do a benchmark. Benchmark results are divided into 2 types, plate number detection recall, and plate number recognition recall. Application-Oriented LP (AOLP) Dataset which are divided to three categories Road Patrol (RP), (LE), (AC) yields 99.4%, 99,2 % and 98,9 % Recall in plate detection and 97,8% 97,4% 96,3% recall in character recognition, CALTECH dataset yields 98.6% Recall in plate detection 97,2% in character recognition. Delpdar overall has a great result of detection and recognition performance however it has limitations when dealing with low resolution, terrible illumination, accidental occlusion, and have poor speed performance.

YOLACT [13] is one of the deep learning approach commonly used to locate an object in an image or video. Ref. [14] perform semantic segmentation for a drone in an image with YOLACT and Architecture HRNet. Our study use YOLACT framework with Resnet-50 [15] as feature backbone which is a deep learning approach to localized vehicle registration plate number from an image or video frame, morphological operations, horizontal projection, and topological structural

analysis [16] used to segment each character in the detected plate number area and lightweight CNN architecture to classify each character. Two datasets are used in this study, those are labeled plate number dataset and binarized Indonesian plate number character. Binarized Indonesian plate number character is acquired from Tel-U License Plate Data-Set V1.0 [17].

II. METHOD

The proposed system in this study is a real-time Indonesian plate number recognition using YOLACT and MobileNetV2 in the parking management system. The proposed system consists of 3 main phases, object localization, character segmentation, and character recognition. We applied YOLACT for object localization to gain plat number area in a frame of video. The second phase is segmented candidate character areas in plate number with Horizontal Projection and topological Structural Analysis. The candidate character areas are recognized with CNN architecture MobileNetV2. The proposed system focuses on recognition accuracy and computational time.

A. Framework YOLACT

Instance segmentation is a domain of computer vision, that is capable of determining several objects from an image and seek localization of the objects instance segmentation framework is growing rapidly in the harmony of expeditious development of Convolutional Neural Network. Almost all instance segmentation frameworks recently use Convolutional Neural Network no exception with YOLACT framework. YOLACT framework consists of 4 processes, feature backbone, feature pyramid network[18], prediction head and Non-maximum suppression. YOLACT is also capable of generating masking for the localized object, but our proposed research does not use masking to reduce computational time.

B. Tracking

Object detection process in a video is usually done in each frame. The tracking algorithm is used in the proposed system, so it is capable of tracking movement from the detected objects in the sequence of frames. The tracking procedure is formalized as shown in Fig. 1.

```

Input: obj contain coordinate  $x_2, y_2$ , was width.  $h$  as height, and  $p$  as predicted class object.
ObjectsTracked = {obj1, ..., objn}
nw = objnew
Output: true if nw is a new object
1: procedure trackingVehicle(ObjectTracked, nw)
2: for each tr ∈ ObjectsTracked do
3:   if |nw.x2 - tr.x2| ≤ tr.w & |nw.y2 - tr.y2| ≤ tr.h & tr.p = nw.p then
4:     return false
5:   end if
6: end for
7: ObjectsTracked.append(nw)
7: return true
8: end procedure
    
```

Fig. 1 Tracking algorithm

C. Convolutional Neural Network

CNN is a feed-forward network in which the information data flow from an input image to several different network layers and an output layer that predicts the input image. CNN currently is spearheading image classification and object localization task. CNN generally composed of a convolutional layer capable of learning the feature representation of the input image, a pooling layer that useful for reducing spatial resolution of feature maps, an activation function that deciding a neuron would activate or not, and an output layer that gives a recognition or classification result [19]. The proposed system uses 2 different CNN architecture, Resnet-50, and MobileNetV2. The proposed system uses Resnet-50 as a feature backbone in YOLACT Framework, the reason we use Resnet-50 among several architectures that have tested in YOLACT framework research, it's because Resnet-50 has balance performance between time computation and recognition accuracy is used for character classification. We also apply architecture MobileNetV2 as character recognition, initially, this architecture developed for image classification or object detection in mobile devices, thus has excellent time computation amid other CNN architecture [20].

D. Image Projection

Image projection is a method that uses the sum of pixels based on the x-axis or y-axis of a binary image. Image projection is often used in character segmentation.

E. Frame sampling

The video formed by a sequence of images, the information between adjacent frames is not much different, on this basis we use a method called frame sampling to ignore several frames with certain conditions, to save computation time. In this research,

we use object localization as a condition frame sampling should start as in Fig. 2.

```

Input: vids ← {v1, vi+1, vi+2, ... vn} where vids is array of frames in video.
sValue, how many frames would be skip in milliseconds,
fps ← input video fps
1: procedure frameSampling(vids, sValue, fps)
2: skipStatus ← false
3: frameIndex ← 0
4: nextIndex ← 0
5: skipInterval ← ⌊  $\frac{sValue}{1000} \cdot fps$  ⌋
6: while frameIndex ≤ vids.length() do
7:   if ObjectFound(vids[frameIndex]) then
8:     frameIndex ← frameIndex + 1
9:   else
10:    frameIndex ← frameIndex + skipInterval
11:   end if
12: end while
13: end procedure
    
```

Fig. 2 Frame sampling algorithm

F. System Overview

Our Proposed system divided into plate localization, character segmentation & character recognition as shown in Fig. 3.

The first process of the proposed system is detecting the vehicle localization and plate number localization with YOLACT while YOLACT is detecting no object in the processed frame, then the frame sampling process will be enforced, on the other hand, process object tracking will take effect and the system will read the frames normally. The reason we use object tracking in our proposed system, we only perform character segmentation and recognition once, so that we need the best plate number image quality by its resolution.

Fig. 4 shows a visualization of line tracking. Three different lines tracking, the upper line with light blue color is used for start tracking of the plate number, the red line applied for capturing plat number, and the bottom line used for end tracking. When the system successfully captures the plate number area, the next

step we apply the character segmentation process as shown in Fig. 5.

The process of character segmentation process begins with resizing the image five times from its size. Next, we apply histogram equalization to the resized image so the character's area will protrude among the background. we do erosion, closing, and opening simultaneously to remove noise in the thresholded image. In the next process, we perform topological structural analysis to find each character from a binary image, and horizontal projection to get the character line position. Horizontal projection uses the sum of pixels based on the x-axis, then we find the 2 local minima based on the conditions where the distance between the local minima is at least 100 pixels. Finally, we cropped each character found in the resized image with the result of topological structural analysis and horizontal projection. the area that is considered a character if it satisfies algorithm shown in Fig. 6.



Fig. 4 Line tracking visualization

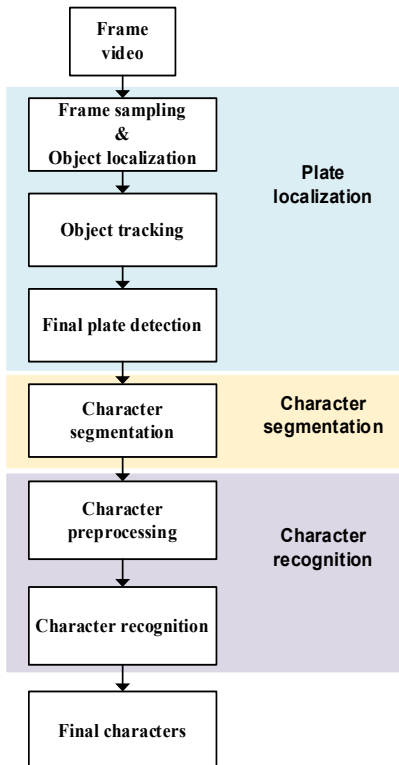


Fig. 3 The architecture of the proposed system

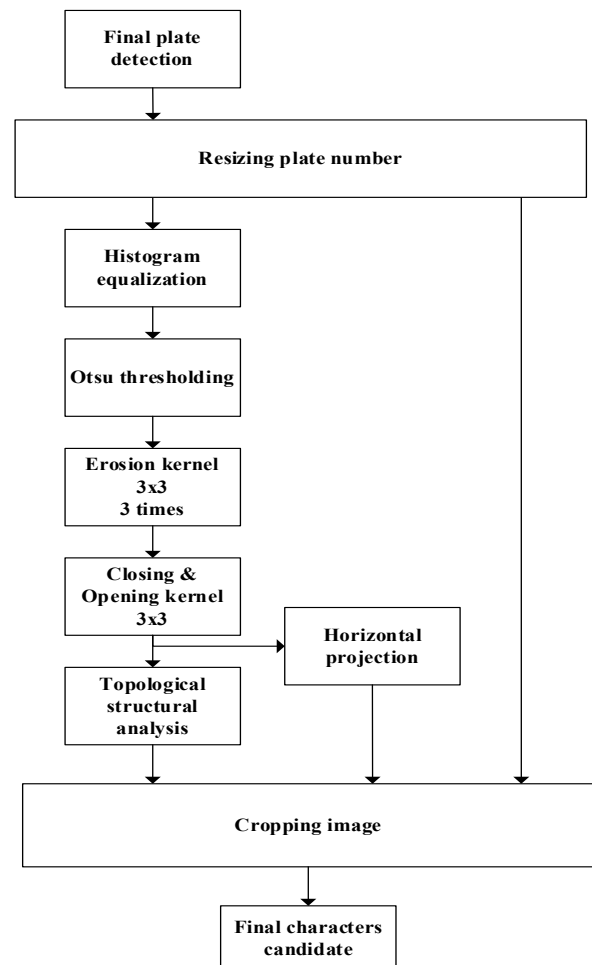
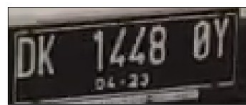


Fig. 5 Character segmentation process

```

Input: coordinate character candidate  $x_1, y_1, x_2, y_2$  and local minima value  $l_1$  and  $l_2$  and threshold as  $t$ .
Output: Character candidate status True or False
1: procedure checkCharacter ( $x_1, y_1, x_2, y_2, l_1, l_2$ )
2:  $width \leftarrow x_2 - x_1$ 
3:  $height \leftarrow y_2 - y_1$ 
4:  $TotalArea \leftarrow height * width$ 
5:  $TotalIntercetion \leftarrow 0$ 
6: if  $y_1 < l_1$  Then
7:    $TotalIntercetion \leftarrow TotalIntercetion + ((l_1 - y_1) * width)$ 
8: end if
9: if  $y_2 < l_2$  Then
10:   $TotalIntercetion \leftarrow TotalIntercetion + ((y_2 - l_2) * width)$ 
11: end if
12: if  $(TotalArea - TotalIntersection) \geq (1 - t) * TotalArea$  Then
13:  return True
14: else
15:  return False
16: end if
17: end procedure
    
```

Fig. 6 Filtering character candidate



(a)



(b)

Fig. 7 (a) Captured plate number (b) Plate number after closing and opening phase

Fig. 7a shows an image of the plate number from the capturing process. Fig. 7a is then processed according to the sequence in Fig. 5. Fig.7b shows an image after the opening and closing process, this result is used to perform two different processing, the first one is finding the local minima value of the image, and second, performing a topological structural analysis process.

Fig. 8 shows the result of the process of finding local minima, the y-axis is the vertical position of the captured plate number, the x-axis represents sums of the pixel values based on the row pixel and the red dot denotes the area of local minima. Besides, Fig. 9 shows the final character segmentation. Red lines represent the local minima which be used to determine characters area and green boxes are the final character candidate those obtained from topological structural analysis and satisfies the conditions of the check character equation.

Fig. 10a shows segmented characters before the pre-processing process, each character has a different size with RGB color. Fig. 10b shows the pre-processed image, we turn each character to binary with Otsu thresholding and positioning every character in the middle.

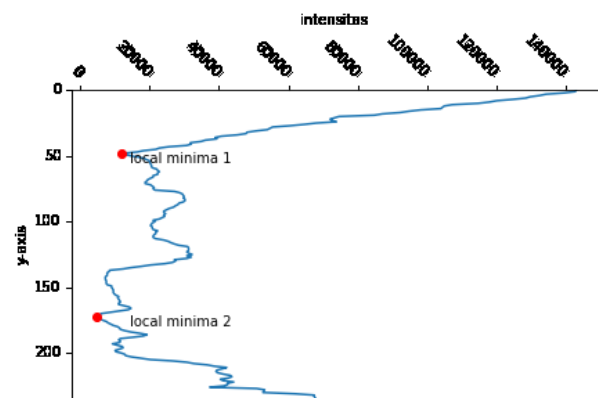


Fig. 8 Histogram of local minima

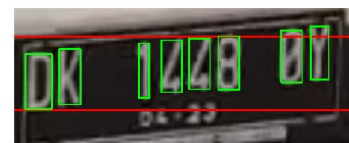


Fig. 9 Visualization character segmentation



(a)



(b)

Fig. 10 (a) characters candidate (b) pre-processed characters candidate

III. RESULTS AND DISCUSSION

Our proposed system is conducted on intel core i7 2,8ghz GTX 1050M and 16 GB of RAM to test the performance and detection accuracy. The evaluation process using 4 different videos with resolution 1280 x 720 pixels and bitrate 5000kbps. detection is considered correct when all characters in plate numbers are recognized correctly. The evaluation dataset that we use is shown in Table I.

In our evaluation process, there are 3 different parameters used for each video dataset, namely the YOLACT model, the Mobilenetv2 model, and the frame sampling value.

Table II show the training result from YOLACT framework. the training process consists of 3 different iteration, 5000, 10000, and 15000 iteration sets and 2 different batch sizes, 8 and 16 batch size. Box mAP and Mask mAP are assessed during the training process which states how well the training model can produce bounding boxes and masking based on the data validation set.

Table III show the training result from MobileNetV2. the training process consists of 2 different iterations, 50, 100, and split data validation 0,1 and 0,3.

Evaluation results are shown in Fig. 11, Fig. 12, and Fig. 13. Fig. 11 shows the effect of variable frame sampling on computation time, without frame sampling and sampling 250 milliseconds had equal detection accuracy, however the computation time between them is distinctive, frame sampling 250 had an average time of 16,06 seconds and without frame sampling was 73,23 seconds. Fig. 12 shows the detection accuracy and computation time based on the MobilenetV2 Model, it appears that the Mobilenetv2 model affects the detection accuracy, where the highest accuracy is obtained with the 100-epoch model and 0.1 validation split with value 83,33%, and in Fig. 13. reveal detection accuracy and computation time based on YOLACT Model, it shows all YOLACT model.

TABLE I
TEST DATA





Data	Plate number	Image	Duration (s)
1	DK1494BR		24
2	DK1448OY		25
3	DK1171FN		25
4	DK1194ME		8

TABLE II
YOLACT TRAINING RESULT

No	Iteration	Batch Size	Epoch	Box mAP	Mask mAP
1	5.000	8	84	80,87	94,30
2	5.000	16	172	89,85	98,28
3	10.000	8	169	90,85	98,47
4	10.000	16	344	95,09	98,79
5	15.000	8	254	93,76	98,56
6	15.000	16	517	97,00	99,06

TABLE III
MOBILENETV2 TRAINING RESULT

No	Epoch	Split train validation	Train-loss	Val loss
1	50	0,1	0.00395	0.03927
2	50	0,3	0.17106	0.22758
3	100	0,1	0.00020	0.00448
4	100	0,3	0.00135	0.05803

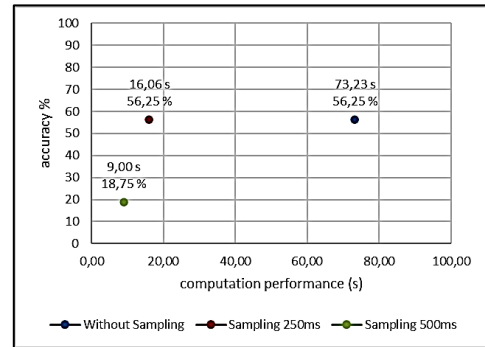


Fig. 11 Accuracy and computation time based on frame sampling

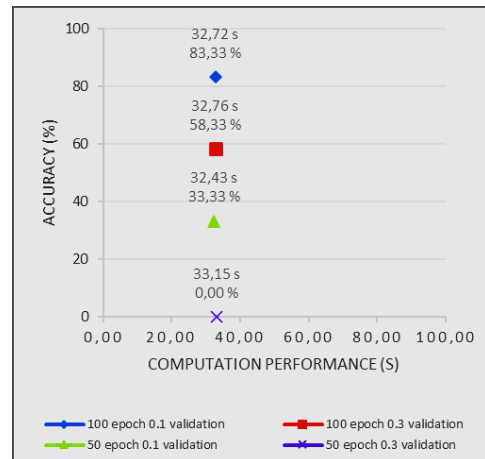


Fig. 12 Accuracy and computation time based on MobileNetv2 model

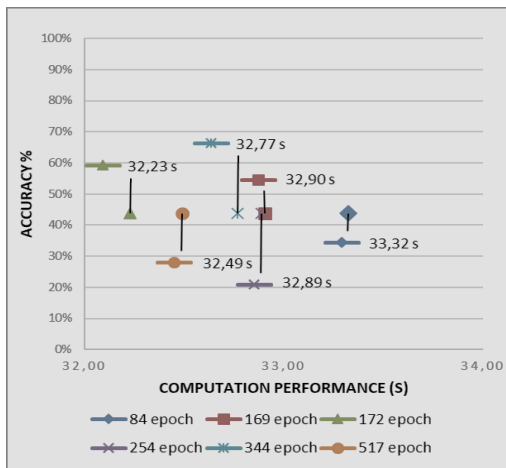


Fig. 13 Accuracy and computation time based on YOLACT model

IV. CONCLUSION

The proposed system uses YOLACT with architecture Resnet-50 as plate number localization, Topological structural analysis, and horizontal projection as methods to segment characters in localization plate number, for the characters recognition, we applied binarized each character found and classified with CNN architecture MobileNetV2. YOLACT model has been trained with labeled license plate and vehicle datasets, then for The MobileNetV2 trained with the binarized image characters of the Indonesian plate number. Our proposed system is implemented in the gate system to support monitoring and recording the number plates of vehicles entering the parking area. Frame sampling method applied to reduce computational time, wherein it becomes useful when there's no object detected by the YOLACT, the system will skip the detection process for the next frames. Experiments that have been carried out override the limitations of problems related to data acquisition problems such as lighting and shadows. The best results are with 250ms frame sampling succeed to reduce computational times up to 78%, whereas the accuracy is affected by MobileNetV2 model with 100 epoch and ratio of split data validation 0,1 which results in 83,33% in average accuracy. Frame sampling can reduce computational time however higher frame sampling value causes the system to fail to obtain plate region area.

REFERENCES

[1] BPS, "Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 1949-2017," 2017. <https://www.bps.go.id/linkTableDinamis/view/id/1133> (accessed Jan. 11, 2020).

[2] B. A. Fomani and A. Shahbahrami, "License plate detection using adaptive morphological closing and local adaptive thresholding," in *2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Apr. 2017, pp. 146–150, doi: 10.1109/IPRIA.2017.7983035.

[3] J. V. John, P. G. Raji, B. Radhakrishnan, and L. P. Suresh, "Automatic number plate localization using dynamic thresholding and morphological operations," in *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, Apr. 2017, pp. 1–5, doi: 10.1109/ICCPCT.2017.8074328.

[4] N. Saleem, H. Muazzam, H. M. Tahir, and U. Farooq, "Automatic license plate recognition using extracted features," in *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*, Sep. 2016, pp. 221–225, doi: 10.1109/ISCBI.2016.7743288.

[5] G. N. Balaji and D. Rajesh, "Smart Vehicle Number Plate Detection System for Different Countries Using an Improved Segmentation Method," vol. 3, no. 6, p. 7, 2017.

[6] R. Laroca *et al.*, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–10, doi: 10.1109/IJCNN.2018.8489629.

[7] R.-C. Chen, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning," *Image and Vision Computing*, vol. 87, pp. 47–56, 2019.

[8] Z. Selmi, M. B. Halima, U. Pal, and A. M. Alimi, "DELP-DAR System for License Plate Detection and Recognition," *ArXiv*, vol. abs/1910.01853, 2019.

[9] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time Bhutanese license plate localization using YOLO," *ICT Express*, 2019.

[10] O. De Gaetano Ariel, D. F. Martín, and A. Ariel, "ALPR character segmentation algorithm," Feb. 2018, pp. 1–4, doi: 10.1109/LASCAS.2018.8399954.

[11] K. P. P. Aung, K. H. Nwe, and A. Yoshitaka, "Automatic License Plate Detection System for Myanmar Vehicle License Plates," in *2019 International Conference on Advanced Information Technologies (ICAIT)*, Nov. 2019, pp. 132–136, doi: 10.1109/AITC.2019.8921286.

[12] C. Szegedy *et al.*, *Going Deeper with Convolutions*. 2014.

[13] D. Bolya, C. Zhou, F. Xiao, and Y. Lee, *YOLACT: Real-time Instance Segmentation*. 2019.

[14] Zihao Liu, Haiqin Xu, Yihong Zhang, Zhouyi Xu, Sen Wu, and Di Zhu, "A Real-Time Detection Drone Algorithm Based on Instance Semantic Segmentation," presented at the Proceedings of the 3rd International Conference on Video and Image Processing, Shanghai, China, 2019.

- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016, pp. 770–778.
- [16] S. Suzuki and others, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [17] Tjokorda Agung Budi W., ST., MT., "Tel-U Vehicle License Plate Data-set V1.0." Biometrics Research Center Laboratory, School of Computing, Telkom University Telekomunikasi No.1, Bandung, West Java, P.O.Box 40257, Indonesia, 2017, [Online]. Available: <https://cokagung.staff.telkomuniversity.ac.id/koleksi-dataset/tel-u-vehicle-license-plate-data-set-v1-0/>.
- [18] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [19] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.
- [20] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.