JURNAL INFOTEL
Informatics - Telecommunication - Electronics

Website Jurnal : http://ejournal.st3telkom.ac.id/index.php/infotel
ISSN : 2085-3688; e-ISSN : 2460-0997

Real-Time Object Detection For Wayang Punakawan Identification Using Deep Learning

Afandi Nur Aziz Thohari[1*], Rifki Adhitama[2]
[1,2] Program Studi S1 Rekayasa Perangkat Lunak, Fakultas Teknologi Industri dan Informatika, Institut Teknologi Telkom Purwokerto
[1,2]Jl. D.I Panjaitan No.128, Purwokerto, 53147, Indonesia
*Corresponding email: afandi@ittelkom-pwt.ac.id

Received 12 December 2019, Revised 16 December 2019, Accepted 19 December 2019

Abstract — Indonesia is a country that has a variety of cultures, one of which is wayang kulit. This typical Javanese performance art must continue to be preserved so that future generations can know it. There are many wayang figures in Indonesia, and the most famous one is punakawan. Wayang punakawan consists of four character namely semar, gareng petruk, and bagong. In order to preserve wayang punakawan to be known by the next generation, this study created a system that identifies punakawan objects in real-time using deep learning technology. The method that is used is Single Shot Multiple Detector (SSD) as one of the models of deep learning that has an excellent ability to classify data with three-dimensional structures such as real-time video. SSD model with MobileNet layer can work in slight computation, so that it can be run in a real-time system. Two steps that must be done to classify objects, such as the training process and the testing process. The training process takes 28 hours with 100.000 steps of iteration.The result of the training process is a model which is used to identify the object. Based on the test result, the accuracy in detecting the object was 98,86%. It proves that the system has been able to optimize object in real-time accurately.

Keywords – accuracy, object detection, SSD, wayang punakawan

I. INTRODUCTION

Indonesia has a variety of cultural heritages from cultural practitioners in the past as country that is rich in diversity of ethnicity, race, and cultures. One culture that still exists and was preserved today was wayang. As a culture that has been recognized by UNESCO, wayang has become a masterpiece because of its high value for human culture [1]. A form of cultural art that plays using the screen and shadow is often found in several cultural activities in the area of Java. However, only a handful of groups who know puppet's characters can perform the attraction. Young people or the next generation are less familiar with the character of the wayang being played.

Therefore, to conserve national culture, the researcher built a system that recognizes types of wayang through real-time video. As research material, the wayang characters which are used were restricted for punakawan character. So that, the system can only identify characters such as semar, gareng, petruk, and bagong. The technology that can be used for identifying punakawan character is deep learning. This branch of artificial intelligence technology implements machine learning method in its algorithm.

Relying machine learning for object recognition on the image is not enough. Because the data needed for recognizing image are very complicated and extensive. The image has different shapes and patterns so it's difficult to recognize the object in the image. Besides that, the similar of patterns and shapes also may cause error in predictions because the model can't make the prediction properly. Therefore, in this study, deep learning is used it has many layers of networks that can be trained repeatedly.

One of the deep learning methods used to classify images is Single Shot MultiBox Detector (SSD). SSD model with MobileNet layer can work in slight computation, so that it can be run in real-time on mobile devices [2]. As a comparison, the method like Faster-RCNN will have a more massive calculation, but produces a much more accurate detection [3]. In the research, the SSD method was used to detection real-time objects. SSD method was chosen because it is lightweight, fast, and has good accuracy.

Research that applies real-time object recognition techniques is rare because it caused high hardware requirements for data processing.But there are still some researches that are relevant to object recognition, such as research conducted by [4] to make a busway lane violation detection system using Single Shot Multibox Detector (SSD) method. Based on the test result, obtained accuracy is 98,2% for detecting vehicle objects such as bus, car, and motorcycle. The other research which implemented deep learning to identify object was research about advertisement billboard detection [5]. This research uses AlexNet Deep Convolutional Neural Network (DCNN) method. Experimental results show that the technique archives 92.7% training accuracy for advertisement billboard detection, while for overall testing results, it gives 71.86% testing accuracy. The last research about the detection object is implementing the convolutional neural network (CNN) to identify Table and Chair Jepara carving motifs [6]. The result's accuracy of the CNN method to identify Jepara carving table and chair can reach up to 98%. The identification of table and chair also can be applied to videos.

Different from the previous research, the object which will be identified in this research is wayang punakawan. The purpose of identifying wayang object is to preserve national culture. This research aims to increase the number of visitors in the Naladipa Museum, Dermaji Village, Banyumas District. The object which will be identified is limited to the punakawan character in the museum. In the next research, the other objects will be included, so that it becomes an attraction for visitors.

II. RESEARCH METHOD

There are four steps in this research, collecting the data, processing the data, analyzing the data, and the evaluation. The explanation from each step is shown in the following points.

A. Collecting Data

The data collecting is carried out by capturing each photo of wayang punakawan. So, the data is grouped into four categories : semar, gareng, petruk, and bagong. The number of samples in each category are 110 images. So, the total amount of pictures is 440 images. Next, those images will be separated into two types, the training data and the testing data. The ratio between training data and testing data is 80% for training and 20% for testing.

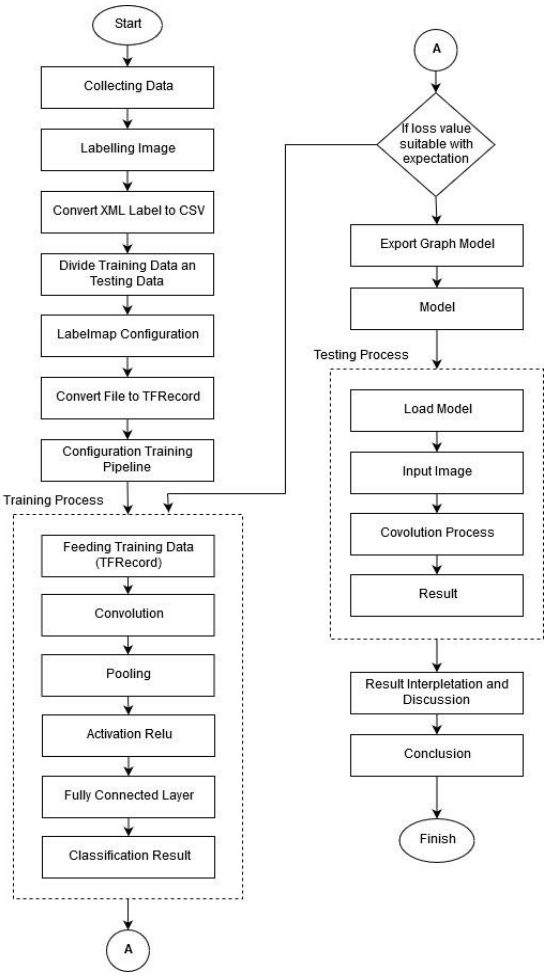

Fig.1. Step of Processing Data

B. Collecting Data

The data collecting is carried out by capturing each photo of wayang punakawan. So, the data is grouped into four categories : semar, gareng, petruk, and bagong. The number of samples in each category are 110 images. So, the total amount of pictures is 440 images. Next, those images will be separated into two types, the training data and the testing data. The ratio between training data and testing data is 80% for training and 20% for testing.

C. Processing Data

After wayang punakawan images obtained, the next step is processing the data. The result of data processing is the training model. The stages of data processing are shown in Fig.1.

Based on Fig.1, the processing step is carried out after we get punakawan images. Then the next step is labeling the pictures. The software that is used to label the pictures is LabelImg. The result of this image

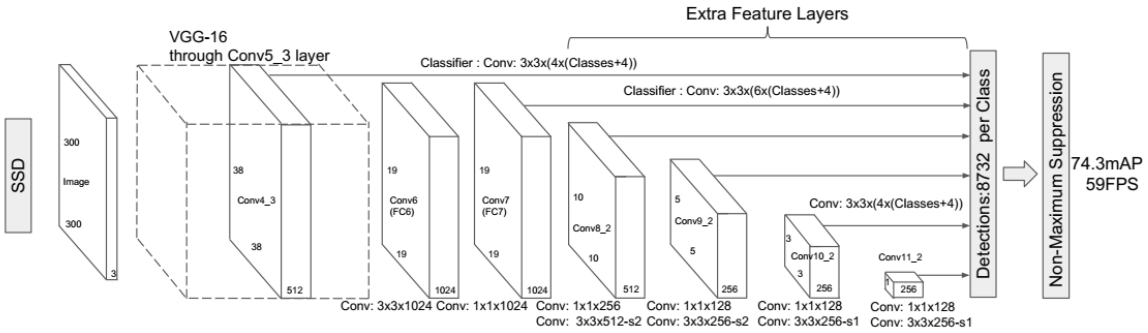

Fig.3. Training Process With Single Shot MultiBox Detector [2]

labeling process is XML files. The image Labeling process is shown in Fig.2.

After getting all the XML files , the next step is to convert the XML label to CSV format. This research uses the python program to convert XML into CSV. The converting process produces two CSV files. One CSV file is used for saving the training data, which is comprised of 80% data. The other CSV file is used for the testing data, which is comprised of 20% data.

The third step, is to convert CSV files to TFRecord. This should be done because TensorFlow is only can able to read data from TFRecord. Before converting to TFRecord, we must set up the configuration in the python program. Because there are four labels, we added four labels using the following code.

```
1.  def class_text_to_int(row_label):
2.       if row_label =='Semar':
3.             return 1
4.        elif row_label =='Gareng':
5.             return 2
6.        elif row_label =='Petruk':
7.             return 3
8.        elif row_label =='Bagong':
9.             return 4
10.        else:
11.            return None
```

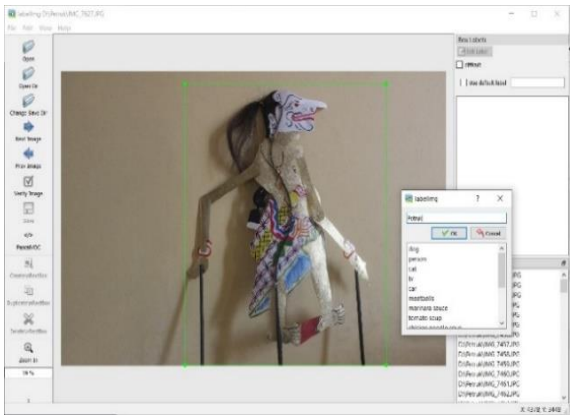

Fig.2. Labeling Image Process

The code above is the function to give a label when the object appears. After converting to the TFRecord file, the next step is the training process. The training process steps are the convolution, the pooling, the activation function, the fully connection layer, and the classification. More detail for training process, is shown the Fig.3.

Based on Fig.3, the training process has several steps. The first is to resize the image to 300x300 pixels. Then, the second step is the convolutional process to get the feature of image. There are some steps in the convolution process or convolutional layer. The convolutional process should be done repeatedly until reach the smallest part of image. The SSD method that implemented in this training is derived from the MultiBox objective, but it is extended to handle multiple object categories[7][8].

After the extraction process in the convolutional layer, the third step is using the artificial neural network method to classify the image. The fully connected layer with a backpropagation method was used to classify the feature of the image. The activation function that is applied to the classification is ReLu. The training process take long time to get identification model because there are so many images that should be processed. So, needs another alternative to process the training data. In this research, we used Intel Core i7 processor to process the data, and the researcher needs 12 hours to get precision model. However, if the Graphical Processing Unit (GPU) was used as an alternative, the processing time can be shortened until eleven times [9]. The hardware device and software which are used for this research were shown in Table 1.

D. Analyzing Data

After the training process, the next step is to analyze the data. The result that has been obtained from the training process is the model pattern. Those models will be tested to measure accuration. If the testing result shows the high loss value, The training will be executed again. But in that case, the parameters that will be used for the training are different.

Tabel 1. Hardware and Software Specification

Software	Python 3.6.5
	OpenCV 3.4.0
	Tensorflow-cpu 1.7.0
	Windows 10 Professional
Hardware	Intel Core i7 8569U 4,70 GHz
	8 Gb RAM
	GTX 1050Ti 4Gb VGA

The parameters that will be changed can be configured in the protobuf file. The protobuf file is a TensorFlow object detection API to configure thr training process. At a high level, the configuration file in protobuf can be divided into five paths.

1. *Model* : define what type of model will be trained (meta-architecture, feature extractor)
2. *Training-config* : specify what parameters shoud be used to train the model parameters (for example SGD parameter, processing input, and initialization value for feature extractor).
3. *Eval_config* : determine the measurement matrix which will rbe eported for evaluation.
4. *Training_input_config* : define dataset which must be trained.
5. *Eval_input_config* : define dataset which must be tested (must be different from dataset training)

The following code below is the parameters that will be used to change the configuration of the training method.

```
feature_extractor {
  type: "ssd_mobilenet_v1"
  depth_multiplier: 1.0
  min_depth: 16
  conv_hyperparams {
        regularizer {
            l2_regularizer {
                 weight: 0.00004
            }
        }
        initializer {
truncated_normal_initializer {
                mean: 0.0
                stddev: 0.03
            }
        }
        activation: RELU_6
        batch_norm {
            decay: 0.9997
            center: true
            scale: true
            epsilon: 0.001
            train: true
        }
    }
}
```

E. Evaluation

The last step in this research is the evaluation form the result. The evaluation result may show a high accuracy for identifying wayang punakawan. However, If the evaluation result shows, poor accuracy result still not good, then the model will be re-test again starting from the data processing step with the changes in parameter values.

III. RESULT

The number of the testing data is 22 images for each punakawan character. The total amount of the data is 110 images for each character. In this research, 80% of the data is used for the training process, and the remaining 20% is used for the testing. Table 2 shows the accuracy of the testing data, using our model that has been built.

Tabel 2. Testing Data Accuracy

No	Label	Valid	Fault	Total
1	Semar	22	0	22
2	Gareng	21	1	22
3	Petruk	22	0	22
4	Bagong	22	0	22
Total		87	1	88
Accuracy			98,86%	

Based on the data in Table 2, the accuracy value from a model that has been built was 98,86%. The system shows the bounding box with the name of punakawan character. Moreover, the system can shows the confidence value of each character besides the name. The result of system was shown in Fig.4.

The system sometimes misclassifies gareng and bagong. Because both of the characters have an almost similar pattern, but the system can still differentiate because the dataset provided is specific and numerous.

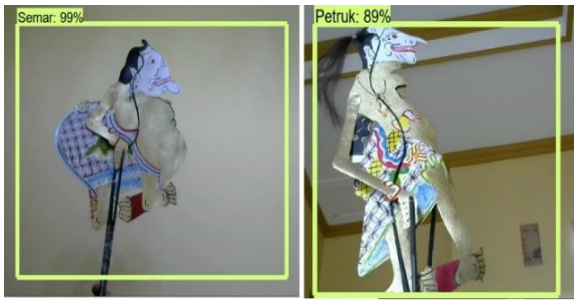

Fig.4. Classification Result of Testing Data

IV. DISCUSSION

After the training process has been done, the result is a model that will be used to recognize the object of the punakawan character. This chapter will discuss the processes that occur during the training process determined the loss value that appears to produce an accurate model.

The total loss that appears in the system must be considered. When doing the training, it is better to keep the loss value less than one. Because according to research [10] , the training process should have minimal total loss atau error value, which is less than one because it effects the detection accuracy. The total loss of the punakawan character detection during it's training process is shown in Fig.5.

The training process shown in Fig.5 lasts 100,000 steps and takes 28 hours to use the CPU as a place to process data. In the SSD network there are two components of the loss. The first one related to the goodness of classification and the second one related to the goodness of localization of the correctly classified object.

Classification loss is the loss value that appears when classifying an object. The smaller the value of classification loss, the more accurate the system in predicting an object. Classification loss during the training process is shown on the graph Fig.6. The graph shows that at the beginning step, the loss value is very high. However, the more steps used, the lesser loss value becomes, which in this case is decrease to the value of 0. This is because the system learns to recognize patterns in the Wayang Punawanawan objects.

On the other hand the localization loss is the mismatch between the ground truth box and the predicted boundary box. The SSD only penalizes predictions from positive matches. In the localization loss, the predictions from the positive matches get close to the ground truth. The negative matches can be ignored. The localization loss during the training process is shown in Fig.7.

Just like the classification loss, the value of loss decreases following the added steps. This is good because the smaller the loss value, the more accurate the bounding box that used to recognize the object.

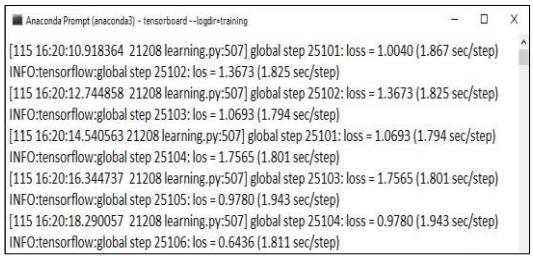

Fig.5. Dataset Training Process

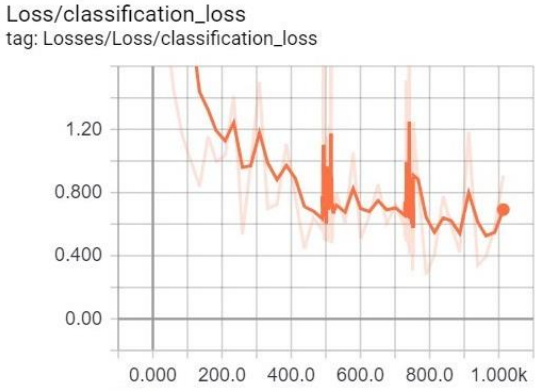

Fig.6. Classification Loss in Training Process

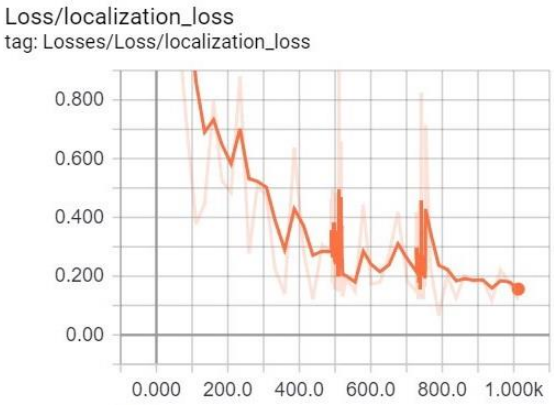

Fig.7. Localization Loss in Training Process

V. CONCLUSSION

The system that was built has succeeded in identifying the object of the wayang punakawan accurately. The model that produced through the training process has an accuracy rate of 98.86% for the identification of the punakawan characters. The utilization of SSD MobileNet with a depth of 6 layers is able to detect objects accurately. SSD configuration in the training process is 100,000 steps of iteration, using ReLU as an activation function, and the batch size is 3. In configuring the SSD method, it must adjust the device used. Number of iteration steps and number of images which used have affect to the accuracy of the object that tested. The more the number of iteration steps and the amount of image data, the higher the accuracy will become.

ACKNOWLEDGMENT

The acknowledgments are delivered to Naladipa museum, Dermaji village, Banyumas district that has been willing to lend wayang punakawan to be the object of this research. The Ministry of Research and Higher Education has funded this research through the Beginner Lecturer Research program in 2019.

REFERENCES

[1] B. Nurgiyantoro, "Wayang dan pengembangan karakter bangsa," *J. Pendidik. Karakter*, vol. 1, no. 1, pp. 1–17, 2011.

[2] W. Liu *et al.*, "SSD: Single shot multibox detector," in *European Conference on Computer Vision*, Amsterdam, 2016, pp. 21-37.

[3] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017, pp. 3296-3305.

[4] A. Susilo, "Implementasi Metode Ssd (Single Shot Multibox Detector) Untuk Mendeteksi Pelanggaran Jalur Busway Menggunakan Masukan Citra Digital," Universitas Teknologi Yogyakarta, 2019.

[5] R. F. Rahmat and O. S. Sitompul, "Advertisement billboard detection and geotagging system with inductive transfer learning in deep convolutional neural network," *Telkomnika*, vol. 17, no. 5, pp. 2659–2666, 2019.

[6] S. R. DEWI, "Deep Learning Object Detection Pada Video Menggunakan Tensorflow dan Convolutional Neural Network," Universitas Islam Indonesia, 2018.

[7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable Object Detection using Deep Neural Networks," in 2014 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colombus, 2014, pp. 2155-1262.

[8] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, High-Quality Object Detection," working paper, Google Inc. 1600 Amphitheatre Pkwy, CA, 2015. [Online]. Available: https://arxiv.org/pdf/1412.1441.pdf

[9] A. Thohari and G. B. Hertantyo, "Implementasi Convolutional Neural Network untuk Klasifikasi Pembalap MotoGP Berbasis GPU," in *Conference on Electrical Engineering, Telematics, Industrial Technology, and Creative Media*, 2018, pp. 50–55.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.