khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Object Detection to Identify Shapes of Swallow Nests Using a Deep Learning Algorithm

Denny Indrajaya [1*], Adi Setiawan[1], Djoko Hartanto[2], Hariyanto[2]
[1]Department of Mathematics and Data Science, Faculty of Science and Mathematics
[1]Universitas Kristen Satya Wacana
Salatiga
[2]PT Waleta Asia Jaya
Salatiga
*662018003@student.uksw.edu

Abstract-Object detection is basic research in the field of computer vision to detect objects in an image or video. the TensorFlow framework is a widely adopted framework to create object detection programs and models. In this study, an object detection program and model are designed to detect the shape of a swallow's nest which consists of three classes, namely oval, angular, and bowl. The purpose model creation is to find out the likeliness of the swallow's nest to the three classes for the swallow's nest sorting machine. The adopted architecture in the modeling is the MobileNet V2 FPNLite SSD since the model obtained from this architecture results in a good speed in detecting objects. Based on the evaluation results that has been carried out, the model can detect the shape of the swallow's nest which is divided into 3 classes, but in some cases swallow's nest are detected into two classes. This issues can still be handled by adjustmenting several parameterss to the object detection program. Results shows that the obtained mAP value of 61.91%, indicating the model can detect the shape of a swallow's nest moderately.

Keywords: object detection, swallow's nest, SSD MobileNet V2 FPNLite, classification, deep learning

Article info: *submitted December 1, 2021, revised May 10, 2022, accepted June 11, 2022*

1. Introduction

Swallow's nest is a nest formed from swallow's saliva which can be consumed by humans. The nest has many benefial properties and it tastes delicious [1]. Before the swallow's nest is consumed, it needs to be processed, in which the condition of the swallow's nest before being processed varies according to various things, such as the color, shape, and intensity of the feathers. With these various conditions, of course, the treatment during processing given for each nest condition is different. Therefore, the first step that needs to be done in processing swiftlet nests is the nest sorting process. Currently, a company i.e., PT Waleta Asia Jaya processes swallow nests by sorting the swallow nests based on color, shape, and feather intensity manually by humans power, thus it slows the production lane. In the development of technology, various tasks can be done with the help of machines to improve company performance in various aspects, two of which are speed and automation. In connection with the process of sorting swallow's nests, a sorting machine can be designed that has the human-like ability to sort swallow's nests. To make machines able to do the task, it is necessary to adopt the theory of computer vision and embed it to the machines.

Computer vision is a branch of computer science that involves image processing and pattern recognition to understand an image or object in images and videos [2], [3]. In computer vision, object detection science uses deep learning algorithm to do complex things such as tracking an object, detecting events, and analyzing behavior [4]. One of the deep learning architectures for creating object detection models with a learning process using data in the form of digital images is SSD MobileNet.

SSD MobileNet is an SSD architecture (Single Shot Multibox Detector) with the MobileNet extractor feature, which architecture can work with little computation, so it is fit to run in real-time [5]. SSD itself is an object detection architecture that has high accuracy and is fast [6], while MobileNet is a lightweight feature extractor [7]. In the research of M. F. Supriadi, E. Rachmawati, and A. Arifianto, the mean Average Precision (mAP) value of the SSD MobileNet V2 architecture is better than the SSD MobileNet V1 in detecting 20 types of objects in the house [8]. In addition, some studies develop the MobileNet SSD architecture using the Feature Pyramid Network (FPN) module to improve detection accuracy [9]. Research on bird nest detection has been carried out several times, but in these studies, swallow nests were not used and bird nests were not divided into classes [10]–[12].

Based on the description, in this study, an object detection model will be created that can be used for the development of a swallow's nest sorting machine in a swallow's nest company. The model made focuses on detecting swallow nests which have 3 classes based on shape, namely swallow nests with oval, angular, and bowl shapes. The model for shape detection is made because the shape of the swallow's nest affects the nest processing process after sorting and the shape of the finished goods obtained, which will also affect the selling price. The model in this study was made using the python programming language, TensorFlow framework, and the SSD MobileNet V2 FPNLite architecture because this architecture it can make the detection process fast and light, which is suitable for use on sorting machines that demand speed in the sorting process.

2. Methods

In carrying out this research, a training process was carried out using data in the form of digital images. The research process consists of several steps, including:

a. Research Implementation Planning

Planning is the first stage carried out in this research, including identifying the problems faced, literature study, and analysis to solve these problems.

b. Data retrieval

Making object detection models requires data from digital images. The object detection model is designed to detect swallow nests and distinguish it shapes to bowl, oval, or angular. The shapes of the swallow's nest can be seen from the part of the nest attached to the wall before the nest is harvested. The bowl shape is a nest that has a flat surface on the part that is attached to the wall while the oval shape has a surface that is not flat, but there is a hollow that forms an angle of less than 90° and not close to 90°. For the angular shape, the surface attached to the wall when viewed in 2 dimensions forms an angle of about 90°. Illustrations of shapes for each class can be seen in Figure 1.

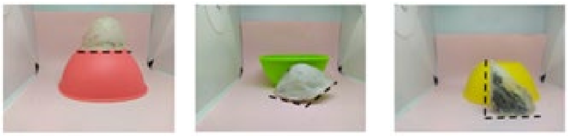


Figure 1. Swallow's nest shape illustration

Detection of objects is done on 2-dimensional images. For this reason, in taking digital pictures using a smartphone camera, it is necessary to pay attention to the shape of a swallow's nest when viewed from the camera's point of view. In addition, giving the background color to the image is also considered in this study. The sample data used in the study is shown in Figure 2.

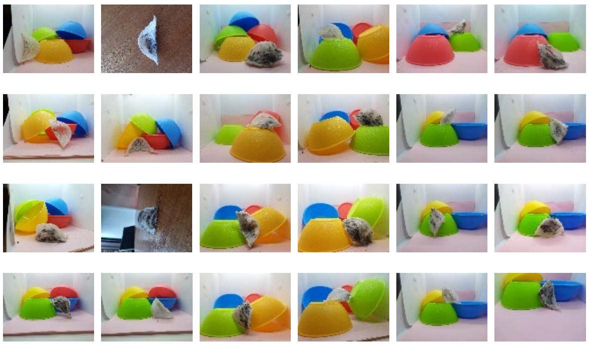
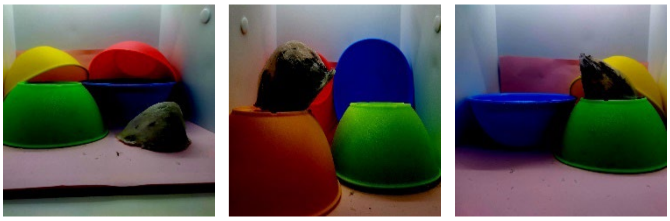


Figure 2. Sample images used in research

c. Data processing

Prior to data processing for model creation, image sorting based on its clarity is conducted at first. From this step, we obtain images with various sizes, where the length and width of the images obtained are 4608×3456, 4000×3000, and 3264×2448 (pixels) with a horizontal resolution of 72 dpi, and a vertical resolution of 72 dpi and the proportion of each swallow's nest in the figure is 5% to 20%. The next process is to divide the image into 2 parts, namely 80% as a training dataset and 20% as a testing dataset. The number of images used is 360 images with 1 nest object in each image, which means there are 288 images in the training dataset and 72 images in the testing dataset. In this study, swallow nests were grouped into 3 classes, where the data used for each class were 96 images on the training dataset and 24 images on the testing dataset.

In making the model, the class labeling process is carried out using the LabelImg software with the output in the form of a file with *.xml format for each image. The results of the class labeling process are then combined into a *.csv file format. There are 2 files with *.csv format created, namely files for dataset training and dataset testing. Then the two files are converted back into files with the *.record format. In addition, a file with the *.pbtxt format is also created which contains a list of the classes used. In the training process, the image data is also reprocessed into a size of 640×640, which is shown in Figure 3.



Bowl Oval Angular

Figure 3. Sample training pictures

d. Object Detection Modeling

Modeling is done using the python programming language and utilizing TensorFlow which is one of the deep learning frameworks for making object detection [13].

In this study, the SSD MobileNet V2 FPNLite architecture is used, where the SSD (Single Shot Multibox Detector) plays a role in detecting by creating bounding boxes to create image localization and determine object positions [14]. Based on [15], the determination of regional candidate boxes on the SSD architecture is used by formula (1).

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(k-1), \quad k \in [1, m] \tag{1}$$

Note:

m = many layers,

s_{min} = lowest feature map scale,

s_{max}= highest feature map scale.

The regional width of the candidate box is calculated using the formula (2).

$$w_k^a = s_k\sqrt{a_r} \tag{2}$$

The regional height of the candidate box is calculated using formula (3).

$$\left(\frac{(i+0{,}5)}{w_{fk}}, \frac{(j+0{,}5)}{h_{fk}}\right), \ j \in [0, h_{fk}), i \in [0, w_{fk}), \tag{3}$$

Special a_r=1 additional scale required $s_k' = \sqrt{s_k s_{k+1}}$.

The coordinate center for each regional candidate box is

$$\left(\frac{(i+0{,}5)}{w_{fk}}, \frac{(j+0{,}5)}{h_{fk}}\right), \ j \in [0, h_{fk}), i \in [0, w_{fk}),$$

Note:

W_{fk} = width of feature map k,

H_{fk} = height of feature map k.

The SSD architecture itself uses VGG as a feature extractor. The SSD architecture with the VGG-16 feature extractor is shown in Figure 4. However, on the MobileNet V2 FPNLite SSD, the VGG-16 feature extractor is changed to MobileNet V2 FPNLite.

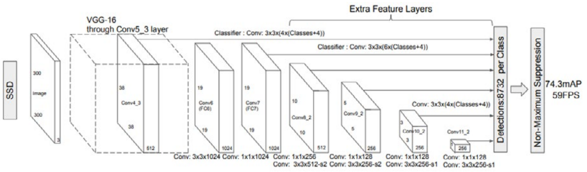


Figure 4. SSD architecture with VGG-16 feature extractor [16]

As explained in the previous paragraph, in the MobileNet V2 FPNLite SSD architecture, MobileNet V2 FPNLite is a feature extractor in extracting features on the image which will later be used on the SSD architecture for detecting objects in the image and their classification [9], [17], [18]. FPN itself is an architecture to produce pyramidal features in object detection, whereas FPNLite is a development of FPN which can produce models with lighter detection capabilities when run [19], [20].

The architecture of MobileNet is shown in Figure 5. The MobileNet structure uses Batch Normalization and the activation function of Rectified Liner Unit (ReLU) for depthwise convolution and pointwise convolution. Formula (4) is the ReLU6 activation function used in the MobileNet V2 structure [21], namely

$$y=\min(\max(z,0),6) \tag{4}$$

where z is the value for each pixel in the feature map. The MobileNet V2 architecture begins with a convolution layer of 32 filters, followed by 19 residual bottleneck layers as shown in Table 1.

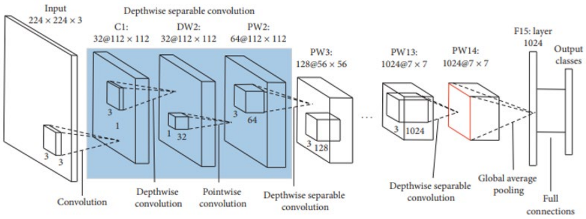


Figure 5. The architecture of MobileNet [22]

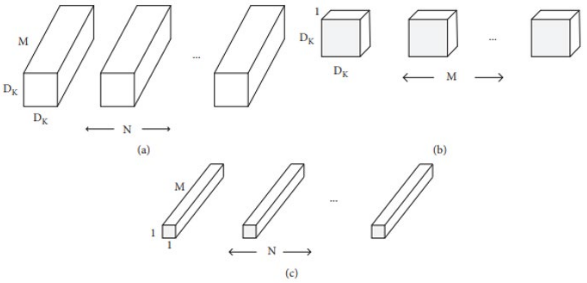


Figure 6. Standard convolutional filters and depthwise separable filters. (a) Standard convolutional filters, (b) depthwise convolutional filters, and (c) point convolutional filters [22]

Table 1. MobileNet V2 Architecture [21]

Input	Operator	*t*	*c*	*n*	*s*
224^2×3	conv2d	-	32	1	2
112^2×32	bottleneck	1	16	1	1
112^2×16	bottleneck	6	24	2	2
56^2×24	bottleneck	6	32	3	2
28^2×32	bottleneck	6	64	4	2
14^2×64	bottleneck	6	96	3	1
14^2×96	bottleneck	6	160	3	2
7^2×160	bottleneck	6	320	1	1
7^2×320	conv2d 1x1	-	1280	1	1
7^2×1280	avgpool 7x7	-	-	1	-
1×1×1280	conv2d 1x1	-		-	

In Table 1 each row describes a sequence of 1 or more identical layers (modulo stride) which are repeated *n* times. All layers in the same order have the same number of output channels c. The first layer of each sequence has stride s and the others have stride 1. All spatial convolutions use a kernel of size 3 × 3. The expansion factor t is always applied to the input size as shown in Table 2 [21].

Table 2. Bottleneck residue block transformation from channel k to k' with stride s and expansion factor t [21]

Input	Operator	Output
$h \times w \times k$	1x1 conv2d, ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise , ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

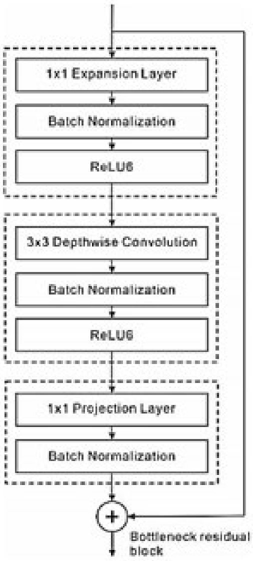


Figure 7. MobileNet V2 convolutional blocks [23]

e. Evaluation

The evaluation of the model is conducted using the object detection program. In the evaluation, the mean Average Precision (mAP) value is used by utilizing the confusion matrix. To obtain the mAP value, data from the test results and data from the class labeling process are used. The data from the test results are composed from the results of the classification, confidence score, and corner points in the predicted bounding box. The data used from the class labeling process is the result of the classification and the corner points in the ground-truth bounding box.

The Confusion Matrix is a table used to measure the performance of algorithms or classification models [24]. The values in the confusion matrix used to measure the performance of an algorithm are True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Here is the form of the confusion matrix.

Table 3. Confusion matrix

	Positive Prediction	**Negative Prediction**
Positive Actual	*True Positive* (TP)	*False Negative* (FN)
Negative Actual	*False Positive* (FP)	*True Negative* (TN)

In using the confusion matrix in object detection, the Intersection over Union (IoU) threshold value will be determined first. IoU is the ratio between the intersection and the union of the ground-truth bounding box (Bgt) and the predicted bounding box (Bp). Formula (5) shows the calculation of the IoU value, namely

$$\text{IoU} = \frac{area\ (B_p \cap B_{gt})}{area\ (B_p \cup B_{gt})} \qquad (5)$$

The IoU threshold value is used to determine whether the detected object is true or false. For example, if the IoU threshold of 0.5 is selected, then:

- True Positive (TP): the model detects the object correctly and the IoU ≥ 0.5.
- False Positive (FP): the model detects the object correctly but the IoU value < 0.5, or the model detects the background as an object when it shouldn't be an object.
- False Negative (FN): the model failed to detect the object.
- True Negative (TN): the model does not detect the background or other objects.

The number of regional candidate boxes contained in the background or other objects that are not detected is very large. Therefore, TN cannot be used. Based on the values obtained from the confusion matrix, the precision and recall values can be searched with formulas (6) and (7), namely

$$Precision = \frac{\text{TP}}{\text{TP+FP}} (6)$$

$$Recall = \frac{\text{TP}}{\text{TP+FN}} (7)$$

Precision shows the model's ability to detect objects correctly, and recall shows the model's ability to detect objects in an image [25].

In measuring the performance of the object detection model, mAP can be used which is indicated by the formula (8) [26].

$$\text{mAP} = \frac{\sum_{i=1}^{k} \text{AP}_i}{k} \qquad (8)$$

note, k = number of classes used.

Based on the mAP formula shown in formula (8), it is necessary to find the value of the Average Precision (AP) for each class first with formula (9).

$$\text{AP} = \int_0^1 p(r)dr \qquad (9)$$

AP can be defined as the area on the Interpolated Precision Recall curve, which is to find the AP value by approximating the formula (10).

$$\text{AP} = \sum_{i=0}^{n-1} (r_{i+1} - r_i) P_{interp}(r_{i+1}) \qquad (10)$$

with

$$P_{interp}(r_{i+1}) = \max_{r' \geq r_{i+1}} P(r') (11)$$

To use formula (10) a table can be designed as shown in Table 4, where the table is the result of model testing to detect 5 objects. The following are the table cration steps:

1. Write down the detection results in the table, including the ID of the detection result which shows the predicted bounding box, then the IoU value and confidence score.
2. Sort the detection results in the table based on the confidence score from the largest to the smallest.
3. Fill in the data in the TP and FP columns based on the detection results and the IoU value.
4. Fill in the data in the Cumulative TP and Cumulative FP columns. In filling in the data in these columns, the data in the TP and FP columns are used in the row to be filled in and the previous rows.
5. Fill in the TP+FP column to calculate the value in the Precision column.
6. Fill in the TP+FN column to calculate the value in the Recall column. In this case, the object used in the test is 5 objects so the value of TP+FN is always 5.
7. Fill in the Precision and Recall fields. Using data in the Cumulative TP, Cumulative FP, TA+FP, and TP+FN columns.

Table 4. Table to calculate AP

Detection result ID	Confidence score	IoU	TP	FP	Cumulative TP	Cumulative FP	TA+FP	TP+FN	Precision	Recall
A	92	92	1	0	1	0	1	5	1	0.2
B	83	73	1	0	2	0	2	5	1	0.4
F	74	21	1	0	3	0	3	5	1	0.6
E	72	52	0	1	3	1	4	5	0.75	0.6
C	71	23	1	0	4	1	5	5	0.8	0.8
D	66	88	0	1	4	2	6	5	0.67	0.8

By using Table 4, especially the values in the Precision and Recall columns, a graph is made as shown in Figure 8. In this case, the recall values (r〗_i) and precision (p(r〗_i)) are implemented in the graph and then connected by orange lines. By using formula (11) the points P_interp (r_(i+1)) are also connected by a green line. Furthermore, the AP value is obtained by calculating the area under the green curve.

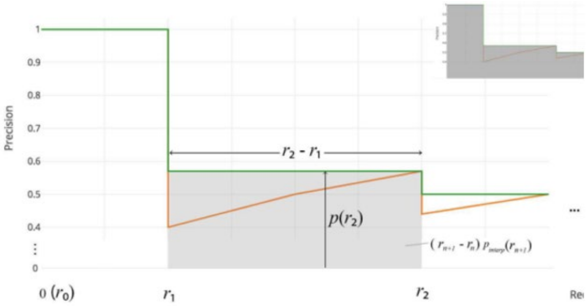


Figure 8. Precision-Recall Curve [27]

3. Result

The object detection model that has been created is then evaluated by testing using a program designed with the python programming language. The sample of the test results is shown in Figure 9.

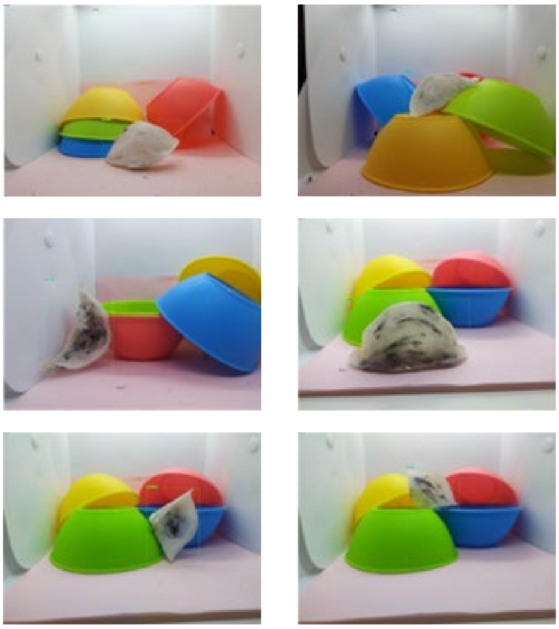


Figure 9. Sample object detection model test results

The test was carried out using 72 images which were divided into 3 parts, namely 24 images for each class. Where in each image there is only 1 bird's nest object, the results of the tests on the 72 images are shown in Figure 10.

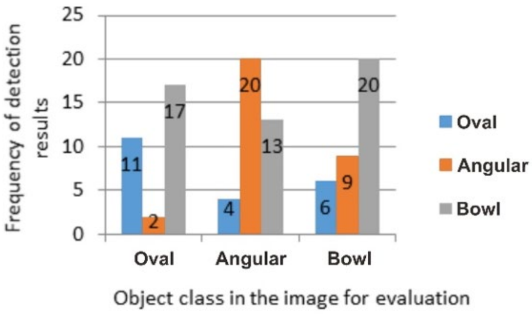


Figure 10. Graph of object detection model test results

Using the test results shown in Figure 10, an evaluation was carried out by finding the mean Average Precision (mAP). To get the mAP value, the Average Precision (AP) value for each class is required. In this case, the designed object detection has 3 classes. Therefore, the AP value of the three classes is sought first by using the Precision Recall curve.

In making the Precision Recall curve, an IoU threshold value is required, and the IoU threshold value used in this evaluation is 0.5. The following is the Precision Recall curve for each class obtained from the detection results shown in Figure 10.

Oval class

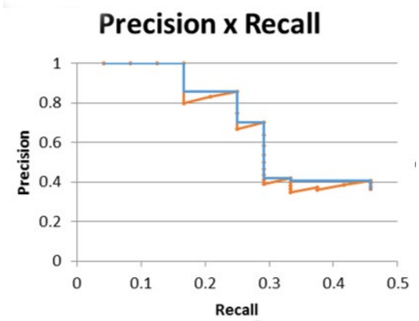


Figure 11. Precision Recall curve for oval class

Angular class

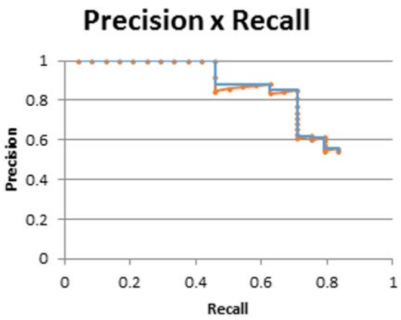


Figure 12. Precision Recall curve for angular class

Bowl class

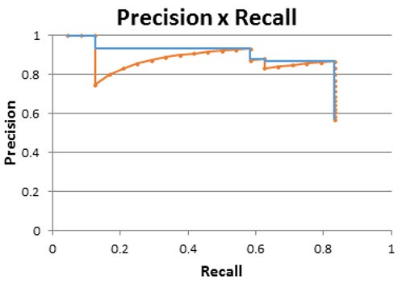


Figure 13. The precision-Recall curve for bowl class

Based on the Precision-Recall curve shown in Figures 11, 12, and 13, the AP value of swallow nest detection with oval class is 0.3357 or 33.57%, angular class is 0.7508 or 75.08%, and bowl class is 0.7707 or 77.07%. The AP values for each class are then averaged so that the mAP value is 0.6191 or 61.91%.

4. Discussion

According to the obtained results, the mAP value is not so large, which is 61.91%. This is due to the relatively small AP value of the oval class when compared to the AP value of the angular and bowl class. In addition, the amount of data used and other parameters for the training process such as the value of the training step and also affect the obtained mAP value.

If we take a look deper, the detection results from the experiments that have been carried out, the obtained results of the detection of a few oval class and many oval class nests detected as bowls. Likewise, many angular class nests are detected as bowls, but many angular classes are detected correctly. While the detection results for the correct bowl class are the same as the angular class, the number of detection results in the overall image with the angular and bowl class nest object is different. In addition, the correct detection results for the corner and bowl class objects are the same, but the AP values obtained are different. The difference is caused by the number of detection results in the entire image and the IoU threshold value that has been determined. These things make the AP value for the small oval class and the AP value for the angular class almost the same as the bowling class. Of the three classes, the highest AP value was obtained by the bowling class. Referring to the Table 5 which presents a summary of the results of the detection of the shape of the swallow's nest.

Table 5. Summary of swallow nest detection results

Class	Evaluation Data		Number of Detection Results			The number of total object detection results	AP (%)
	Number of pictures	Number of nest objects	Oval	Angular	Bowl		
Oval	24	24	11	2	17	30	33.57
Angular	24	24	4	20	13	37	75.08
Bowl	24	24	6	9	20	35	77.07

In Table 5 it can be seen that the detection results obtained for each class are greater than the number of swallow nest objects that should be. In testing the ability to detect an oval-shaped swallow's nest, 24 images were used with 24 objects of an oval-shaped swallow's nest in the entire image, but the results of detection of a swallow's nest obtained in these 24 images were bounding boxes of 30 consisting of 11 detection results of oval class, 2 corner classes, and 17 bowl classes. This is caused by the detection results that appear more than 1 on an object.

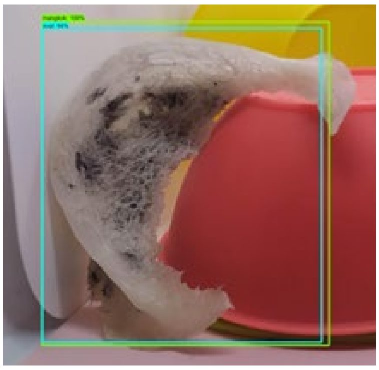


Figure 14. Two detection results on an object

The appearance of 2 detection results in 1 swallow's nest object shown in Figure 14 is caused by the shape of the swallow's nest which is almost similar when viewed in 2 dimensions. However, this can be handled by adjustments to the program made, such as limiting the number of objects that can be detected in 1 frame or 1 image and determining the minimum value for the confidence score. has been determined does not appear in the image. As is the case in Figure 14, if the minimum value of the 95% confidence score is determined, the results of the "oval" detection with a confidence score of 94% will not appear. That way, if the model is applied to a swallow's nest sorting machine, there will be no detection results of more than one result on 1 swallow's nest object.

Research related to bird nest detection has also been carried out by J. Li, D. Yan, K. Luan, Z. Li, and H. Liang [12]. In this study, several architectures were used to detect bird nests, namely Faster RCNN, Faster RCNN with Focal Loss, Cascade RCNN, and ROI Mining Faster RCNN with mAP values of 78.29%, 78.99%, 79.19%, and 82.51%, respectively. Based on this research, it can be seen that the object detection model can detect bird's nests with a fairly good performance. In this study, the development of detected nest objects, namely swiftlet nests, was divided into 3 classes based on their shape. From the results of the research that has been carried out, it is found that the object detection model that is made can detect the shape of a swallow's nest which is divided into 3 classes with an mAP value of 61.91%. The mAP value obtained is not so high when compared to the mAP value obtained from the bird's nest detection model in the study of J. Li, D. Yan, K. Luan, Z. Li, and H. Liang.

As previously shown, the mAP value obtained in this study is influenced by several things, such as the number of images used in the training process and the emergence of more detection results than the number of swallow nests in all images. In addition, the obtained mAP is also influenced by the image of a swallow's nest whose shape is difficult to classify as shown in Figure 15 (oval-shaped nest). The swallow's nest in the picture is difficult to classify because the part of the nest attached to the wall is not visible. Therefore, in classifying it, it is necessary to pay attention to certain parts of the nest, the parts in question are indicated by circles in Figure 15. After paying attention to these parts and a sufficient understanding of the shapes of the swallow's nest, the nest can be said to tend to enter the oval class. An example of the analysis in classifying the nest cannot be carried out by the object detection model that has been made because the model classifies the nest based on the color in the image that is converted to numeric and the computational process is based on the training results, so in this case, the probability of an error occurring in the model for classifying the nest will become large and has an impact on the mAP value.

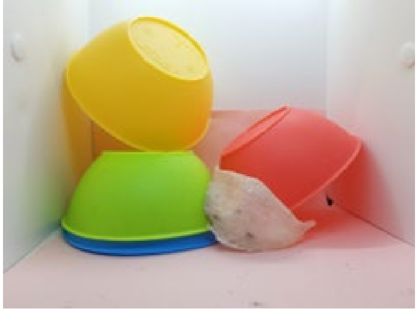


Figure 15. Swallow's nest is difficult to classify when viewed in 2 dimensions

Although in this study the mAP value obtained was not high, some of the detection results obtained showed good results. This can be seen from the size of the bounding box which corresponds to the size of the nest in the picture. In addition, it is also supported by the correct classification results and a high confidence score as shown in Figure 16. The results of incorrect detection can also be seen in Figure 17.

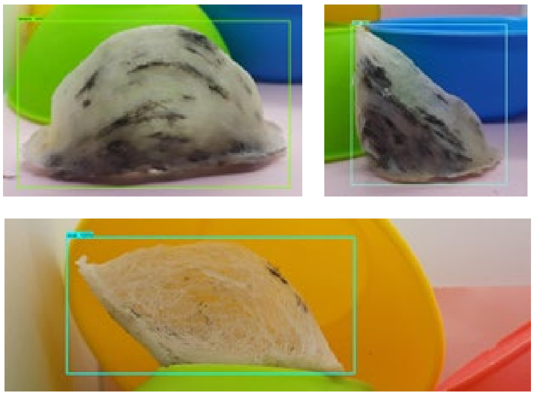


Figure 16. Good detection results

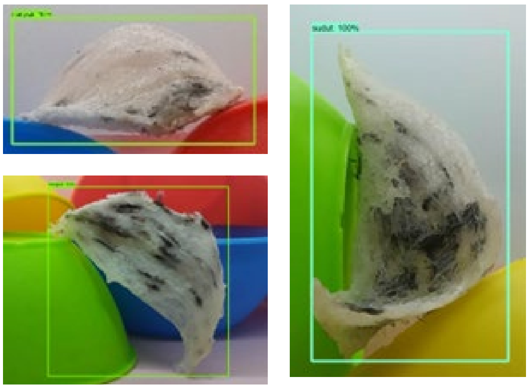


Figure 17. Incorrect detection result

In making the model, a training process is carried out, which if retraining is carried out with the same data, different results can be obtained. For this reason, the mAP value obtained in this study is still possible to be increased using the same data, by modifying the MobileNet V2 FPNLite SSD architecture and the values of the parameters used in the training process. Alternatives to using other architectures while taking into account the purpose of modeling can also be considered. In addition, it is also necessary to design an object detection model that is useful as validation when detecting swallow's nests. In this case the model will work by detecting the position of the swallow's nest from the camera's point of view. If the visible position of the swallow's nest is detected as a valid position, the results of the detection of the shape of the swallow's nest can be accepted. Valid in this case means that the shape of the swallow's nest when viewed from the camera's point of view can be determined.

5. Cunclusion

From the results of the research that has been done, the object detection model created using the SSD MobileNet V2 FPNLite architecture can detect the shape of a swallow's nest which has 3 classes, namely bowl, oval, and angular. In addition, based on the evaluation that has been carried out using 72 swallow nest images, an mAP value of 61.91% is obtained which shows the model's performance in detecting the shape of a swallow's nest which is divided into 3 classes.

Further research that can be done is to increase the mAP value by making a swallow nest shape detection model using another architecture or modifying the SSD MobileNet V2 FPNLite architecture by paying attention to the values of the parameters used in the training process. In addition, an object detection model can be made to detect the position of the swallow's nest when viewed from the camera's point of view, to reduce the error rate in the detection results if the model is to be used on a sorting machine.

6. Acknowledgment

Thank you to PT Waleta Asia Jaya for allowing us to conduct research using the company's swiftlet nests, as well as providing facilities for conducting research, such as computers used to create programs and object detection, models.

7. Reference

[1] L. Elfita, "Analysis on Protein Profile and Amino acid of Edible Bird's Nest (Collocalia fuchiphaga) from Painan," *Jurnal Sains Farmasi & Klinis*, vol. 1, no. 1, pp. 27–37, 2014.

[2] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A Paper Review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 28–36, 2018.

[3] S. R. U. . Dompeipen, Tresya Anjali Sompie and M. E. I. Najoan, "Computer Vision Implementation for Detection and Counting the Number of Humans," *Jurnal Teknik Informatika*, vol. 16, no. 1, pp. 65–76, 2021.

[4] J. Deng, X. Xuan, W. Wang, Z. Li, H. Yao, and Z. Wang, "A Review of Research on Object Detection based on Deep Learning," *Journal of Physics: Conference Series*, vol. 1684, 2020.

[5] A. N. A. Thohari and R. Adhitama, "Real-Time Object Detection for Wayang Punakawan Identification Using Deep Learning," *Jurnal Infotel*, vol. 11, no. 4, pp. 127–132, 2019.

[6] J. Huang *et al.*, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3296–3305.

[7] M. N. Rizal, D. T. Nugrahadi, R. A. Nugroho, M. R. Faisal, and F. Abadi, "Implementasi SSD_ Resnet50_V1 untuk Penghitung Kendaraan," *Kumpulan Jurnal Ilmu Komputer (KLIK)*, vol. 8, no. 2, pp. 106–115, 2021.

[8] M. F. Supriadi, E. Rachmawati, and A. Arifianto, "Pembangunan Aplikasi Mobile Pengenalan Objek untuk Pendidikan Anak Usia Dini," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 2, pp. 357–364, 2021.

[9] Y. C. Chiu, C. Y. Tsai, M. Da Ruan, G. Y. Shen, and T. T. Lee, "Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems," in *2020 International Conference on System Science and Engineering (ICSSE)*, 2020, pp. 1–5.

[10] X. Wu, P. Yuan, Q. Peng, C. W. Ngo, and J. Y. He, "Detection of Bird Nests in Overhead Catenary System Images for High-Speed Rail," *Pattern Recognition*, vol. 51, pp. 242–254, 2016.

[11] F. Li *et al.*, "An Automatic Detection Method of Bird's Nest on Transmission Line Tower Based on Faster_RCNN," *IEEE Access*, vol. 8, pp. 164214–164221, 2020.

[12] J. Li, D. Yan, K. Luan, Z. Li, and H. Liang, "Deep Learning-Based Bird's Nest Detection on Transmission Lines Using UAV Imagery," *Applied Sciences*, vol. 10, no. 18, p. 6147, 2020.

[13] D. J. P. Manajang, S. R. U. A. Sompie, and A. Jacobus, "Implementasi Framework Tensorflow Object Detection dalam Mengklasifikasi Jenis Kendaraan Bermotor," *Jurnal Teknik Informatika*, vol. 15, no. 3, pp. 171–178, 2020.

[14] P. R. Aningtiyas, A. Sumin, and S. Wirawan, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih," *Jurnal Ilmiah Komputasi*, vol. 19, no. 3, pp. 421–430, 2020.

[15] K. Hu, F. Lu, M. Lu, Z. Deng, and Y. Liu, "A Marine Object Detection Algorithm Based on SSD and Feature Enhancement," *Complexity*, vol. 2020, 2020.

[16] W. Liu *et al.*, "SSD: Single Shot Multibox Detector," in *Computer Vision – ECCV 2016*, Springer International Publishing, 2016, pp. 21–37.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," 2017, pp. 2117–2125.

[18] I. B. Pakpahan and I. C. Dewi, "Pendeteksian Lubang Pada Jalanan Menggunakan Metode SSD-MobileNet," *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, vol. 11, no. 2, pp. 213–222, 2021.

[19] G. Ghiasi, T. Y. Lin, and Q. V. Le, "NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (CVPR), 2019, pp. 7029–7038.

[20] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles using Camera Data," *Remote Sensing*, vol. 13, no. 1, 2021.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[22] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A Novel Image Classification Approach via Dense-Mobilenet Models," *Mobile Information Systems*, vol. 2020, pp. 1–8, 2020.

[23] W. Rahmaniar and A. Hernawan, "Real-Time Human Detection using Deep Learning on Embedded Platforms: A Review," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 462–468, 2021.

[24] M. R. Faisal and D. T. Nugrahedi, *Belajar Data Science: Klasifikasi dengan Bahasa Pemrograman R.* Banjarbaru: Scripta Cendekia, 2019.

[25] R. Padilla, S. L. Netto, and E. A. B. Da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2020, pp. 237–242.

[26] G.-S. Peng, "Performance and Accuracy Analysis in Object Detection," California State University at San Marcos, 2019.

[27] J. Hui, "mAP (mean Average Precision) for Object Detection," *jonathan-hui.medium.com*, 2018. https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173.