Automatic Fish Identification Using Single Shot Detector

Arie Vatresia[1*], Ruvita Faurina[2], Vivin Purnamasari[3], and Indra Agustian[4]
[1−3]Informatika, Fakultas Teknik, Universitas Bengkulu
Bengkulu 38371, Indonesia
[4]Teknik Elektro, Fakultas Teknik, Universitas Bengkulu
Bengkulu 38371, Indonesia
Email: [1]arie.vatresia@unib.ac.id, [2]ruvita.faurina@unib.ac.id, [3]vivingionino@gmail.com,
[4]iagustian@unib.ac.id

***Abstract*—The vast sea conditions and the long coastline make Bengkulu one of the provinces with a high diversity of marine fish. Although it is predicted to have high diversity, data on the diversity of marine fish on the Bengkulu coast is still very limited, especially in the process of fish species detection. With the development and expansion of computer capabilities, the ability to classify fish can be done with the help of computer equipment. The research presents a new method of automating the detection of marine fish with a Single Shot Detector method. It is a relatively simple algorithm to detect an object with the help of a MobileNet architecture. In the research, the Single Shot Detector used is six extra convolution layers. Three of the extra layers can generate six predictions for each cell. The Single Shot Detector model, in total, can generate 8,732 predictions. The research succeeds in identifying seven from ten genera of marine fish with a total dataset of 1,000 images, with 90% training data and 10% validation data. Each fish genus has 100 images with different shooting angles and backgrounds. The results show that the Single Shot Detector model with MobileNet architecture gets an accuracy value of 52.48% for the identification of 10 genera of marine fish.**

***Index Terms*—Automatic Fish Identification, Single Shot Detector, Sorting Machine**

I. Introduction

FISH detection is a way of classifying fish based on special characteristics. It can be through a description of the shape, body pattern of fish, color, or other characteristics [1–3]. This detection typically can be identified by fishermen and some people with certain knowledge [4, 5]. Unfortunately, knowledge is not available to everybody because each area has a different name for each fish despite its scientific name. Thus, it needs help from a machine to do it automatically based on the knowledge embedded into the algorithm, including deep learning.

There is a significant increase in the study of classification systems in the field of biology based on morphology (appearance of shapes) and taxonomy (similarities and distinctions of properties) automatically [6–8]. It is due to the increase in processing capabilities and computer equipment [9, 10]. There is a need to model fish recognition to improve the performance of the sorting process since only fishermen can detect the name of fish in the system. Furthermore, the implementation of the latest technology for fish detection can help to sort machine development for smart fishery systems. It can be beneficial for the economic and tourism aspect of the area of Bengkulu, where the fishery normally takes place.

Deep learning is an analytical method used to analyze large amounts of data more deeply. It is one of the popular artificial intelligence methods in the last ten years, which has succeeded in achieving the best results in various problems. It requires hundreds of thousands or millions of images for the best results and high-performance computing. Moreover, it has also been widely used for fish classification. Based on the previous case study, the algorithm used are AlexNet, VGG16 & VGG19, and Google Net for the classification of exotic fish species in the water and Single Shot Detector, YOLO, VGG16, and Inception 3 for the classification of fish in different environments [11–13].

Single Shot Detector is a deep learning model used to detect an object with a single shot. It uses a relatively simple algorithm because it does not go through the stages of making a proposal and a feature in the resampling stage but through the stages of summarizing all calculations in a single shot [6, 12, 14]. It makes Single Shot Detector easy to train and can be directly integrated into the system. Single Shot Detector also implements the bounding boxes feature

Received: Jan. 25, 2022; received in revised form: May 13, 2022; accepted: May 17, 2022; available online: Sep. 14, 2022.
*Corresponding Author

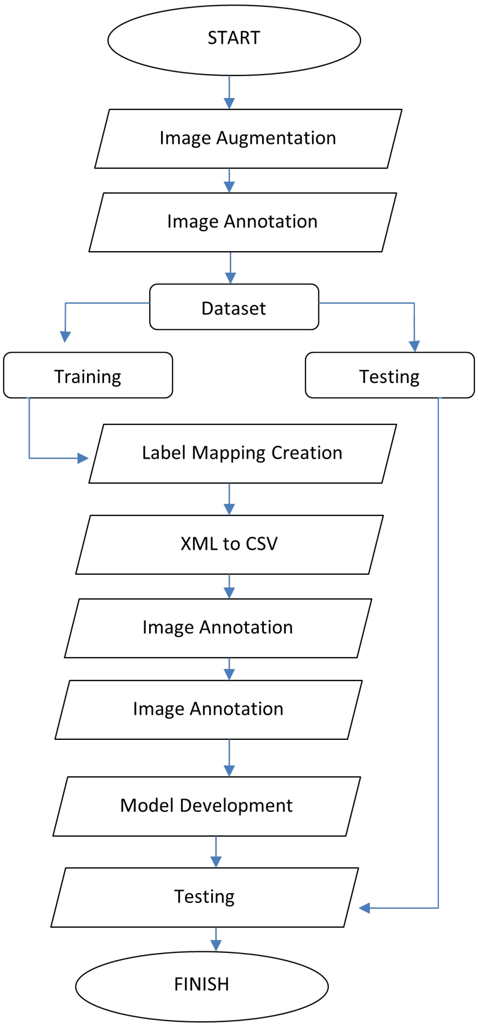


Fig. 1. Preprocessing method.

to estimate the location of the detected object, which has a higher computation and detection speed value than other models [3, 15–17].

Based on the results of the previous research, the researchers are interested in conducting research on the identification of captured marine fish genera using a Single Shot Detector. The identification process involves localization to estimate the location of the detected object. The research aims to apply the Single Shot Detector model to identify fish genera in Bengkulu. Furthermore, the research also calculates the accuracy value of the architectural model used.

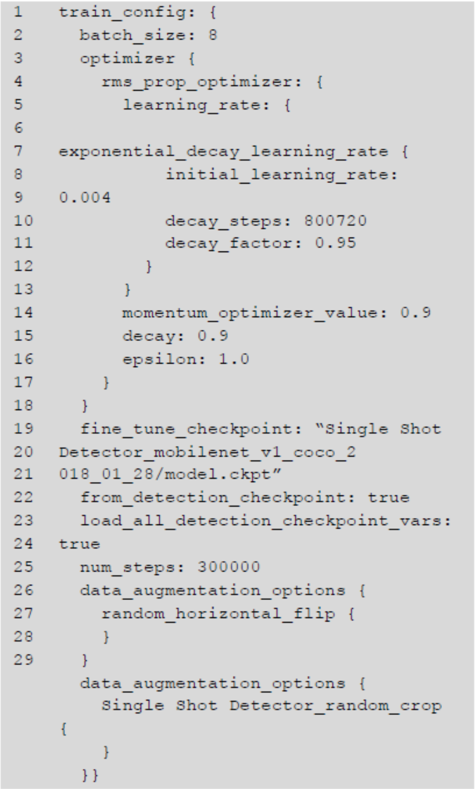
II. Research Method

The type of research is applied research. The researchers search, study, and collect data from various relevant international and national journals, scientific articles, literature reviews containing the concepts studied, and books on fish species obtained from the Ministry of Marine Affairs and Fisheries Website [18]. The researchers collect information directly from the research site. The preprocessing method can be seen in Fig. 1. The researchers have taken pictures of the fish directly caught by fishermen in several areas, namely the Coastal Market, Bengkulu Market, and Pulai Baai, Bengkulu City. Then, the researchers consult with one marine lecturer at the University of Bengkulu. It is the discussion to validate the fish data that has been obtained and to determine the genus of the fish images as a support for the research.

Next, the testing process carried out on the model is evaluation metrics [19, 20]. Before making a model, a model analysis has been carried out first. The analysis is very important for the needs of whether the model should be made up-to-date or updating an existing model. System analysis includes analysis of the workflow model. The design of the model aims to describe in detail the workings of this model. The design is carried out in making the identification model of captured marine fish using Single Shot Detector.

Preparation of the dataset is the first step in the training process. The dataset is one of the most important parts of the identification process. A collection of datasets is used to train and test object identification. In the research, the identification process uses 10 fish genera: Caranx, Carangoides, Alepes, Elagatis, Katsuwonus, Parastromateus, Thunnus, Scomberomorus, Psettodes, and Rastrelliger [9]. Dataset samples are taken from various sources. Those files have various sizes and different shooting angles. Then, the samples also use synthesized data. Each dataset is 100, with a total sample of 1,000. It is to avoid dataset imbalances. When the training process is carried out, not having a balanced number of samples can interfere with the performance of the final detection and the accuracy of the detection [3, 6, 16]. The augmentation of the dataset is to convert the original image into synthetic data form and obtain the diversity of the dataset. The process for augmenting the data is by cropping the original fish image and moving it into a blank background [15].

After all the images have been annotated, they are saved in the Pascal VOC format because labeling is only in two formats, namely Pascal VOC and YOLO. The file type of Pascal VOC is XML. This XML type will be converted into CSV type so that it can be generated into TensorFlow records format. After annotating the dataset, the dataset division divides the sample data into the train and test folders. The train folder is used for training the model, while the test folder is for evaluating the model. Usually, the dataset



```
train_config: {
  batch_size: 8
  optimizer {
    rms_prop_optimizer: {
      learning_rate: {

exponential_decay_learning_rate {
          initial_learning_rate:
0.004
          decay_steps: 800720
          decay_factor: 0.95
        }
      }
      momentum_optimizer_value: 0.9
      decay: 0.9
      epsilon: 1.0
    }
  }
  fine_tune_checkpoint: "Single Shot
Detector_mobilenet_v1_coco_2
018_01_28/model.ckpt”
  from_detection_checkpoint: true
  load_all_detection_checkpoint_vars:
true
  num_steps: 300000
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  data_augmentation_options {
    Single Shot Detector_random_crop
{
    }
  }}
```

Fig. 2. Configuration training script.

distribution ratio is 9:1, which is 90% of the training image and 10% of the testing image. The configuration of the training script can be seen in Fig. 2.

Next, TensorFlow requires a label map, which maps each label to an integer value or a list of classes/objects. The label map is used in the training and detection process. Figure 2 shows a map of labels named “object-detection” with the file extension of .pbtxt, assuming that the data contain ten labels needed (Alepes.sp, Carangoides.sp, Caranx.sp, Elagatis.sp, Katsuwonus.sp, Parastromateus.sp, Psettodes.sp, Rastrelliger.sp, Scomberomorus.sp, Thunnus.sp) during pipeline configuration. Pipeline configuration is needed as a training and evaluation process because TensorFlow uses Protobuf to store and exchange information/data in a structured manner. The configuration file is divided into five parts [21–23], as follows.

1) Model is the process of initializing the type of model to be trained (i.e., meta-architecture and feature extractor);
2) Training_config determines the parameters that must be used to train the model (e.g., Stochastic Gradient Descent (SGD), input preprocessing, and initialization value of feature extraction);
3) Eval_config determines the measurement matrix that will be used for the evaluation process;
4) Train_input is the process of initializing the dataset file to be trained;
5) Eval_input initializes the dataset file that the model will evaluate. Usually, it must be different from the input dataset in training.

III. Results and Discussion

The research shows a different architecture than the previous research. It describes how the performance of Single Shot Detector defines the fish type over manually input data. The data are collected from the fishery collection and market for the training and testing data. However, the data are limited to camera performance that can be improved in future research. The Single Shot Detector model in detecting objects uses a single layer. The predicted bounding box area on the Single Shot Detector is recognized by the default bounding box through various scales and ratios for each feature map location. Each default box with IoU > 0.5 is categorized as matched [24, 25]. Then, the Single Shot Detector model with MobileNet architecture is used as a feature extractor. In the research, the Single Shot Detector used is six extra convolution layers. Three of the extra layers can generate six predictions for each cell. The Single Shot Detector model, in total, can generate 8,732 predictions by utilizing these six layers. The Single Shot Detector model uses MobileNet v1 as its architecture. After the image is inserted, it is extracted first by the MobileNet architecture with depthwise convolution with a 3×3 matrix. Then, it combines the filtered values into a 1×1 matrix using pointwise convolution to create new features. After that, it proceeds with the four extra convolution layers of the Single Shot Detector. Then, the value is obtained in the form of a vector which proceeds to get output in the form of object detection with a bounding box.

Batch size is the size or number of samples that are used during training. In the script, the batch size is “8”, dividing the sample or dataset into eight batches. Batch size also affects the training/learning process because the larger the batch size is, the longer the computational time is required. The batch size must also be adjusted to the Graphic Processing Unit (GPU) used. Then, the learning rate is for optimization in determining the number of steps in each iteration. The learning rate value also affects the performance level of accuracy. If it uses a large enough learning rate value, the loss value will increase when running several iterations during training. However, if the learning rate value is small, the results are also not good. The learning rate value used in this configuration is 0.0004.

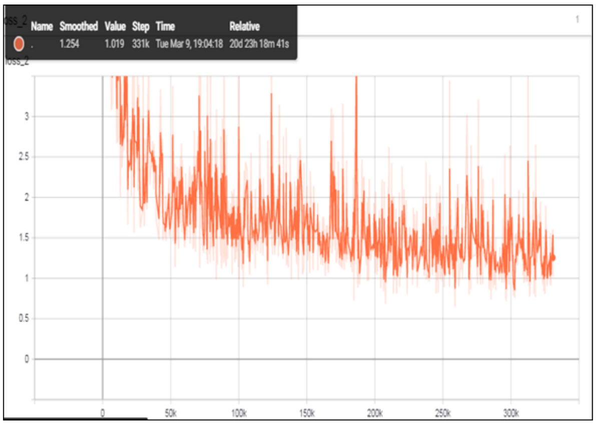
Moreover, steps are the number of steps used for the training process. This configuration has 300,000 steps. The measurement matrix used by the Single Shot Detector model is Coco Detection metrics [24–26].

To do this neural network training, the researchers also use Google Colaboratory as an execution process to run commands in the Python programming language. Using Google Collaboratory can simplify the training process because it provides free access to the GPU. Since only GPU limits usage to 12 hours and Google Colaboratory is run online, it requires a stable Internet network. Moreover, the training process takes a long time. In the research, it takes 1 to 5 days to complete the training because there are often disturbances in the Internet network, which causes the training process to be disrupted, limits in Google Colaboratory, and GPU restrictions. So, the researchers must wait for the network to be stable and GPU to return to normal to continue the training process. When the training process is running, observing or monitoring the progress of the process can be seen with the TensorBoard. It is one of the very nice features provided by TensorFlow, so the researchers can continuously observe, monitor, and visualize a number of different training or evaluation matrices when the model is trained.

The graphs on the TensorBoard visualizes all the work during the training process. With the graph on the TensorBoard, the researchers can see several graphs, such as loss, learning rate, and global step. In Fig. 3, it can be seen that with the number of iterations of 300,000 steps, the loss value at the end of the iteration is 1.3. The lower the graph is, the better the loss value will be. It shows the smaller possibility of errors during the training process. If the graph does not change while the process is still running, it means that it has shown convergence in the training process. In the research, the graph shows that it has not converged and requires more steps.

The new model, as a result of the training, is named “My-Model-Single Shot Detector”, which contains the frozen_inference_graph file. The file is used for the object detection testing process. The process of evaluating the model matrix is carried out to test the model that has been generated. In the research, the research uses the Intersection over Union (IoU), Precision, Recall, and mean Average Precision (mAP) as important matrices for model evaluation. It uses an IoU of 0.50 (green): 0.05 (red): 0.95 (blue). The results of using the threshold for each class/object can be seen in Fig. 4, where the green box is prediction, and the blue box is ground truth.

Furthermore, a test is carried out by looking at the performance using the precision-recall curve to measure the performance of the Region-based Convolutional Neural Networks (R-CNN) mask algorithm. Precision-recall is one of the important measures for evaluating network performance on test datasets. In ad-



Name Smoothed Value Step Time Relative
● . 1.254 1.019 331k Tue Mar 9, 19:04:18 20d 23h 18m 41s

Fig. 3. Training loss graph.

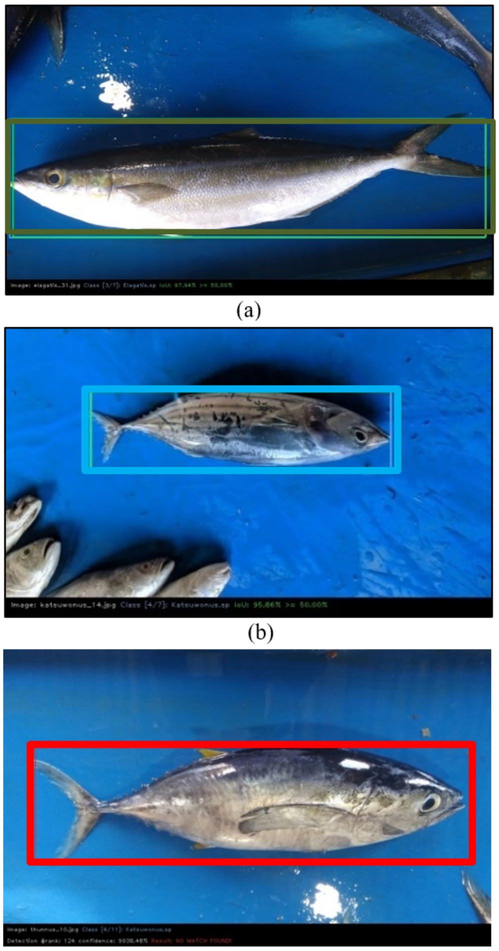


Image: elagatis_31.jpg Class [3/7]: Elagatis.sp IoU: 97.94% >= 50.00%

(a)

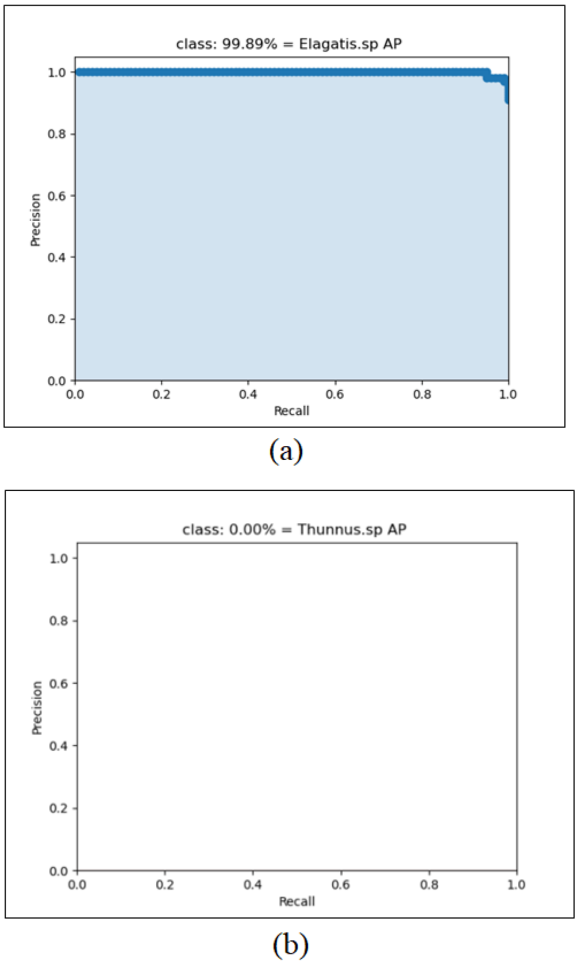
Image: katsuwonus_14.jpg Class [4/7]: Katsuwonus.sp IoU: 95.86% >= 50.00%

(b)

Image: thunnus_10.jpg Class [4/11]: Katsuwonus.sp
Detection @rank: 124 confidence: 9938.48% Result: NO MATCH FOUND!

(c)

Fig. 4. IoU for (a) Elagatis; (b) Katsuwonus; (c) Thunnus.



(a)

(b)

Fig. 5. (a) The best case model on the precision-recall graph of Elagatis, (b) The worst case model of Thunnus.

TABLE I
DETECTION RESULT.

Class/Genus	Confidence	$AP^{IoU\ 0.50:0.95}$
Alepes.sp	0.9	0.91
Carangoides.sp		0.95
Caranx.sp		0.98
Elagatis.sp		0.99
Katsuwonus.sp		0.97
Parastromateus.sp		0.91
Psettodes.sp		0.00
Rastrelliger.sp		2.50
Scomberomorus.sp		0.00
Thunnus.sp		0.00
mAP of all classes		0.5248 (52.48%)

dition, precision is measured based on the relevance of outcomes. Meanwhile, recall measures the total number of correct and relevant outcomes. The precision-recall curves are expressed in terms of the y-axis and x-axis, respectively. From the results, with higher precision, the recall rate also increases. It indicates that the model is efficient and well-integrated. Figure 5 shows the example.

The results of the evaluation of fish detection are carried out in ten separate classes to test the performance of the network. The evaluation parameter is calculated when the predictive confidence value is set to 0.9. The results can be seen in Table I. The mean AP (mAP) of the ten classes can be calculated with an average of ten mean precision (AP), and the result is 0.5248/52.48%.

The results show that the highest average precision for all classes is elagatis. Meanwhile, three classes (Thunnus, Scomberomorus, and Psettodes) are not detected at all. These three classes contribute false positives to various classes, reducing their AP values. Undetected results in object detection can occur due to several things, such as dataset imbalance. Likewise, Rastrelliger only gets an AP value of 2.50.

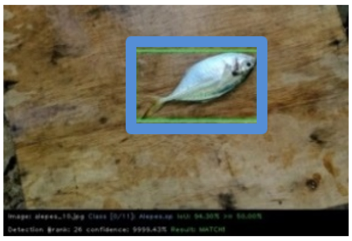
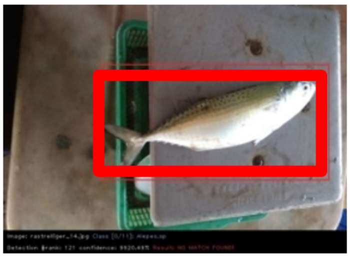
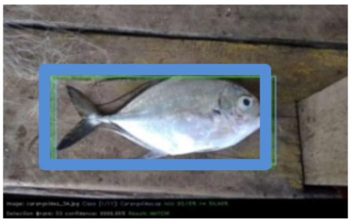
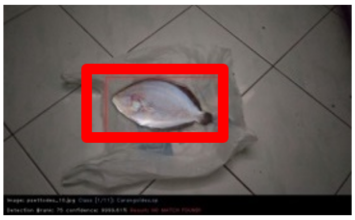
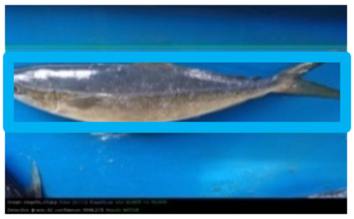
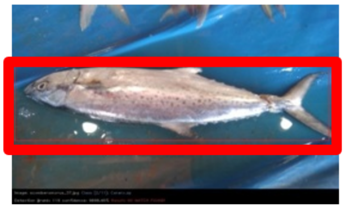
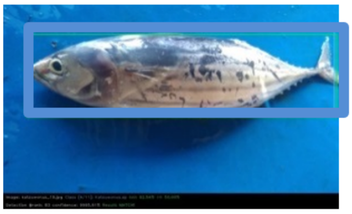
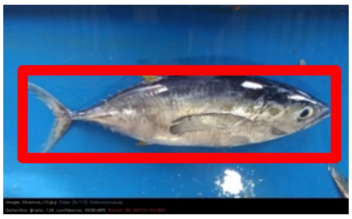
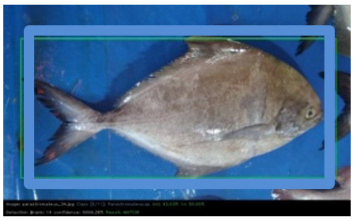
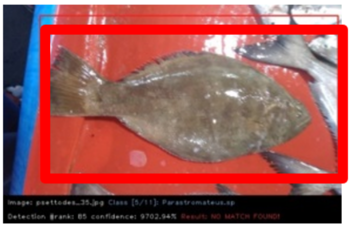
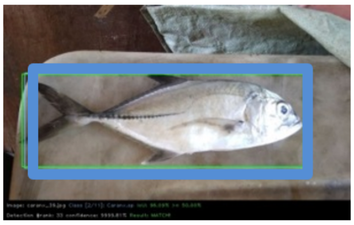
Table II shows how the algorithm works through the data. The blue and green colors mean that the algorithm successfully detects the fish genus. Meanwhile, the red color means that the algorithm experiences difficulty in learning the object. Undetected results in object detection can be influenced by several things, such as dataset imbalance, small object size in the image that makes the system unable to identify in detail, and lighting in the image that changes the color and physical characteristics of each fish. For example, Psettodes.sp is not detected because the fish has two parts with different body colors, so it is difficult to identify it. It is also because the dataset in each section is not the same. Moreover, the fish in the picture has a slight physical resemblance to Parastromateus.sp (black pomfret).

IV. Conclusion

The research presents a new method of automating the detection of marine fish with a Single Shot Detector method. The research proposes a new model from the process of training data and successful implementation of the algorithm. The performance of the algorithm shows fairly good detection for each object, with an mAP value of 52.48%. The research limitation is that Single Shot Detector model can only identify seven out of ten genera: Caranx, Carangoides, Alepes, Elagatis, Katsuwonus, Parastromateus, and Rastrelliger. Based on the analysis, model design, implementation, and model testing, the researchers suggest that the implementation model can be embedded into mobile applications and real-time. Thus, it will be linear to the usage of the MobileNet architecture, which is intended as a mobile vision.

Furthermore, future research can use the latest MobileNet architecture. It can also improve Single Shot Detector model. The suggested input for a model is

TABLE II
IMAGE DETECTION RESULT.

Match	No Match
 Alepes.sp	 Rastrelliger.sp
 Carangoides.sp	 Psettodes.sp
 Elagatis.sp	 Scomberomorus.sp
 Katsuwonus.sp	 Thunnus.sp
 Parastromateus.sp	 Psettodes.sp
 Caranx.sp	

images with the size of 512×512 with more numbers of training and testing data to improve the validation result.

References

[1] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, “Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system,” *ICES Journal of Marine Science*, vol. 77, no. 4, pp. 1295–1307, 2020.

[2] S. Cui, Y. Zhou, Y. Wang, and L. Zhai, “Fish detection using deep learning,” *Applied Computational Intelligence and Soft Computing*, vol. 2020, pp. 1–13, 2020.

[3] C. Shi, C. Jia, and Z. Chen, “FFDet: A fully convolutional network for coral reef fish detection by layer fusion,” in *2018 IEEE Visual Communications and Image Processing (VCIP)*. Taichung, Taiwan: IEEE, Dec. 9–12, 2018, pp. 1–4.

[4] J. Hu, G. S. Xia, F. Hu, and L. Zhang, “A comparative study of sampling analysis in the scene classification of optical high-spatial resolution remote sensing imagery,” *Remote Sensing*, vol. 7, no. 11, pp. 14 988–15 013, 2015.

[5] F. Azadivar, T. Truong, and Y. Jiao, “A decision support system for fisheries management using operations research and systems science approach,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 2971–2978, 2009.

[6] M. H. Saleem, S. Khanchi, J. Potgieter, and K. M. Arif, “Image-based plant disease identification by deep learning meta-architectures,” *Plants*, vol. 9, no. 11, pp. 1–23, 2020.

[7] B. Qian, Y. Xiao, Z. Zheng, M. Zhou, W. Zhuang, S. Li, and Q. Ma, “Dynamic multi-scale convolutional neural network for time series classification,” *IEEE Access*, vol. 8, pp. 109 732–109 746, 2020.

[8] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 279–283, 2017.

[9] A. R. Singkam, A. P. Yani, and A. Fajri, “Keragaman ikan laut dangkal Provinsi Bengkulu,” *Jurnal Enggano*, vol. 5, no. 3, pp. 424–438, 2020.

[10] G. Cheng, J. Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[11] A. Jalal, A. Salman, A. Mian, M. Shortis, and F. Shafait, “Fish detection and species classification in underwater environments using deep learning with temporal information,” *Ecological Informatics*, vol. 57, pp. 1–13, 2020.

[12] G. Chandan, A. Jain, H. Jain, and Mohana, “Real time object detection and tracking using deep learning and OpenCV,” in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*. Coimbatore, India: IEEE, July 11–12, 2018, pp. 1305–1308.

[13] Y. Wageeh, H. E. D. Mohamed, A. Fadl, O. Anas, N. ElMasry, A. Nabil, and A. Atia, “YOLO fish detection with Euclidean tracking in fish farms,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 5–12, 2021.

[14] D. Diamanta and H. Toba, “Pendeteksian citra pengunjung menggunakan single shot detector untuk analisis dan prediksi seasonality,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, pp. 125–141, 2021.

[15] D. Jiang, B. Sun, S. Su, Z. Zuo, P. Wu, and X. Tan, “FASSD: A feature fusion and spatial attention-based single shot detector for small object detection,” *Electronics*, vol. 9, no. 9, pp. 1–20, 2020.

[16] W. Shi, S. Bao, and D. Tan, “FFESSD: An accurate and efficient single-shot detector for target detection,” *Applied Sciences*, vol. 9, no. 20, pp. 1–13, 2019.

[17] L. Jin and G. Liu, “An approach on image processing of deep learning based on improved ssd,” *Symmetry*, vol. 13, no. 3, pp. 1–15, 2021.

[18] Kementerian Kelautan dan Perikanan, “Kelautan dan perikanan.” [Online]. Available: https://statistik.kkp.go.id/

[19] D. Guan, H. Li, T. Inohae, W. Su, T. Nagaie, and K. Hokao, “Modeling urban land use change by the integration of cellular automaton and Markov model,” *Ecological Modelling*, vol. 222, no. 20-22, pp. 3761–3772, 2011.

[20] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, “Towards better analysis of deep convolutional neural networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2016.

[21] S. Qiu, G. Wen, J. Liu, Z. Deng, and Y. Fan, “Unified partial configuration model framework for fast partially occluded object detection in high-resolution remote sensing images,” *Remote Sensing*, vol. 10, no. 3, pp. 1–23, 2018.

[22] S. Kato, S. Amemiya, H. Takao, H. Yamashita, N. Sakamoto, and O. Abe, “Automated detection of brain metastases on non-enhanced CT using single-shot detectors,” *Neuroradiology*, vol. 63,

no. 12, pp. 1995–2004, 2021.

[23] A. Juneja, S. Juneja, A. Soneja, and S. Jain, "Real time object detection using CNN based single shot detector model," *Journal of Information Technology Management*, vol. 13, no. 1, pp. 62–80, 2021.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[26] ——, "YOLO V2.0," in *CVPR*, 2017.