INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS

ISSN(print): 2643-9840, ISSN(online): 2643-9875

Volume 07 Issue 04 April 2024

DOI: 10.47191/ijmra/v7-i04-03, Impact Factor: 8.22

Page No. 1458-1463

Real-Time Smart Fish Feeder Model Using Ssd Mobile Net V2

Jesus M. Lindo, Jr.[1], Arnel C. Fajardo[2]

[1,2] College of Computing Studies, Information and Communications Technology,Isabela State University Cauayan Campus Philippines

ABSTRACT: The success of fish farming will depend on improved feed management and lower operating costs, which are essential factors in facilitating an efficient food allocation to the fish. Various automatic fish feeders were used to feed the fish at set intervals, it consists of a mechanical and electrical system to form a device and execute a programmed method, instead of manually feeding the fish by hand. Many methods are effective at evaluating and quantifying fish feeding intensity but are mostly done on the movement and behavior of the fish. However, recognition accuracy is affected due to water quality and the overlapping of fish. To solve this problem, in this study, the author will be focusing on the model in counting the fish pellets, it will capture the image, and count the pellets and the results will be a novel method as a basis for releasing pellets in laying the logical foundation in creating a modern real-time smart fish feeder. The model produced a significant result that detected small objects like fish pellet and count that has gathered a minimal loss in terms of classification, localization, regularization, and normalization.

KEYWORDS: Machine Learning, IoT, CNN, Deep Learning, Fish Feeder, Smart Aquaculture

1 INTRODUCTION

Aquaculture will be more effective and sustainable through the use of smart fish farming. Smart aquaculture can be the solution in solving the problems of fishery operations and can increase aquaculture productivity. It also monitors fish at various stages, reduces the risk of failure, and increases profitability and productivity. In intensive aquaculture, the fish's feeding stage has an impact on the breeding costs and production efficiency. For some fish species, more than 60% of all production costs go toward feeding costs. [1] [2].

Overfeeding lowers production efficiency because of food waste that can also impact the water quality of the pond while underfeeding has a detrimental effect on fish growth. Traditional feeding decisions for assessing the hunger desire of fish are often impeded by high fish density and water turbidity [3]. Rapid aquaculture growth has also led to several problems, including water contamination, fish malnutrition, cannibalism and changing for normal water parameters can result to fish kills. Therefore, a reliable and highly trustworthy fish feeding monitoring system is required to observe the fish behavior in consuming fish pellets.

Modern fish farming in aquaculture is the integration of cutting-edge technologies that can promote sustainability in the industry while reducing the need for traditional methods, such technologies are the Internet of Things (IoT), big data, cloud computing, and artificial intelligence (AI) [4] [5].

In 2019, a study proposed a deep convolutional neural network (DCNN) to categorize fish behavior into four classification levels, namely, "none" means they don't respond to food, "weak" means they are passive eater "medium" which means they take the food but return to original position and "strong" which means they will consume all available foods with 90% classification accuracy [6]. Furthermore, several other studies have also used neural network models to assess fish behavior using a convolutional neural network (CNN) and long short-term memory (LSTM) network to predict feeding and non-feeding behavior of salmon species with an accuracy of 80% [7]. From a detailed analysis of the existing literature related to smart aquaculture, the application of deep learning can be divided into four categories: live fish identification, species classification, behavioral analysis, and biomass estimation. It is also revealed from a literature review that fish identification and species classification are the most popular areas of research. In contrast, behavior identification has been less targeted than others.

Up to now, considerable works on fish appetite assessment have been done. Studies have shown that the behaviors fish with the current condition of their growing habitat exhibited during feeding can directly affect the fish appetite [8] [9]. Therefore, assessing fish appetite based on fish feeding behavioral characteristics has become a research priority. Computer vision, as a noncontact and cost-effective technology, has become the main technological means for fish appetite assessment. For example, a computer vision-based feeding activity index for Atlantic salmon has been proposed to assess fish appetite. They assessed the intensity of fish feeding activity by analyzing the aggregation and dispersion of fish school. Another study developed a method to assess the feeding intensity of tilapia in recirculating aquaculture system (RAS) based on the Lucas-Kanade optical flow algorithm (OFA) and entropy [10].

Numerous applications of machine learning using computer vision are being used in aquaculture, including classification, fish counting, size measurement, and behavior analysis. [11]. Deep learning (DL) is also applied in aquaculture because of its effective ability to express features. DL is a multi-layer learning network that can extract semantic information from the pixel level, which is suitable for fish behavior recognition through images [12]. DL-based feeding decision-making research has made outstanding progress in recent years. As a result, accurate recognition of fish behavior can achieve optimal feed control, lower feeding costs, and increase economic efficiency [13]. Various researchers have conducted a significant amount of research on fish behavior. They evaluated fish feeding behavior using near-infrared imaging technology and measured the feeding behavior index using a support vector machine (SVM) and gray gradient symbiosis matrix. The results through experiment and the expert manual assessment provided a correlation value of 0.945 [14]. Zhou continued his research in 2018 and improved the results by using an adaptive neuro fuzzy inference system (ANFIS) to assess and analyze fish-eating behavior. The results showed that the ANFIS accuracy was 98% [15].

In this study the researchers proposed a model that analyzes the consumption of the fish pellets by capturing an image within the time intervals, the results will be compared to the starvation and normal condition of the fish through their behavior and will be used as a reference to support the decision system in whether to feed more or stop the feeding process.

2 MATERIALS AND METHODS

2.1 Image acquisition

An artificial pond setting was setup using a large tub to imitate the appearance of a fish pond. The pond was sprinkled with fish pellets and the 720p camera was setup directly above the pond using a tripod. The videos were taken both at bright and dark environment so that during training the model would still generalize on both dark and bright environment. After the videos were taken, different frames were extracted from the video and 77 images were randomly taken to build the image dataset. All of the images were resized to 500x500 pixels for consistency during model training and validation.

The image dataset was then divided into training set and validation set. 57 images fall to the training set and 20 images for the validation set. The training set is used to train the model and the validation set is used to test the performance of the model.

2.2 Image data annotation

Image annotation is the process of marking the features in the image that the object classification model needs to recognize. This process adds metadata to the dataset for specifying the ground-truth. The images on the dataset were annotated using an open-source tool called “labelImg” using Pascal VOC (Visual Object Classes) format. The Pascal VOC is a common annotation format saved as XML file that is human-readable.

The dataset was manually annotated by drawing boxes over the pellets in the image using the “Create RectBox” option of the LabelImg tool. In total, there are 1,162 bounding boxes on the training set and 334 bounding boxes on the validation set.

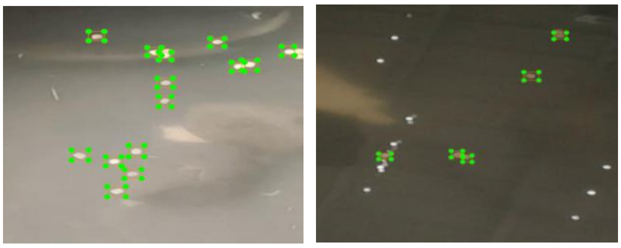


Figure 1. Image annotation.

2.3 Training the object detection model

This study deals with a single-class object detection model since the model only needs to predict one class which is the pellet. However, the task focuses on identifying the positions of the pellets in the image as it will then be used to learn the number of pellets. The Google colab was used along with the Python programming language to write and run all the codes required to train the model. The Tensorflow Object Detection API was used to import necessary libraries and files that are needed to build, train, and evaluate the model.

Before training the model, the hyperparameters to be used were identified. Following are the hyperparameters configuration:

Table 1. *Hyperparameters*

Hyperparameter	**Value**
Architecture	SSD MobileNetV2
Optimization algorithm	Gradient Descent with Momentum
Activation function	Relu6
Batch size	16
Training steps	20,000

The baseline model architecture used is the SSD MobilenetV2 using Relu6 as the activation function. SSD MobileNetV2 is a one-stage object detection model which is being developed by Google. This architecture was chosen for this study because it is suitable to be deployed on low-compute devices with high accuracy performance.

The optimization algorithm used is the Gradient Descent with Momentum. The optimization algorithm is responsible on minimizing the error (loss) during model training. Minimizing the loss results in better model generalization. The optimization algorithm minimizes the cost function (loss function) by finding an optimized value for a learnable parameter. For this, the Gradient Descent was chosen which is currently the most popular optimization algorithm used in deep learning and machine learning. There are different types of Gradient Descent, however, gradient descent with Momentum is used for the reason that it shows faster updates in gradient descent.

Data augmentation is performed before training the model. Random horizontal flip and random cropping were used to artificially expand the training data during training. This is done to give the model more samples to learn from. The model training was done using a batch size of 16 and with 20,000 steps. The model training finished after 7.3 hours.

2.4 Model Validation

Counting detected pellets

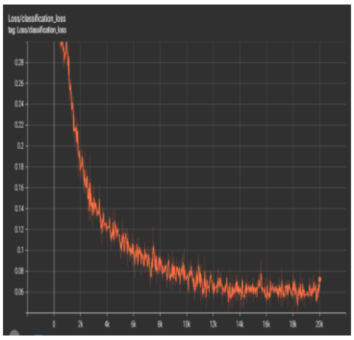
The task of the trained object detection model is to detect pellet on a given image and record the location of each pellet. Each detection of pellet has its own classification confidence to define how sure the model that the certain object is indeed a pellet. This is helpful in such cases that there are debris that could be slightly mistaken by the model. In order to provide a better accuracy for counting, a confidence threshold of 25% has been set. The python code utilized for this task is shown in figure 2.

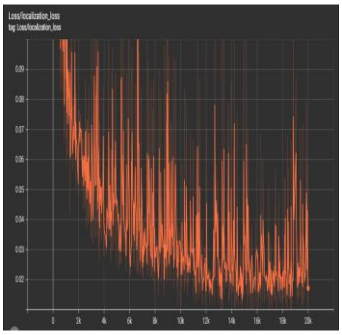
```python
def count_pellet(image_path):
    image_np=load_image_into_numpy_array(image_path)
    input_tensor=tf.convert_to_tensor(image_np)
    input_tensor=input_tensor[tf.newaxis, ...]

    detections=detect_fn(input_tensor)

    num_detections=int(detections.pop('num_detections'))
    detections={key:value[0,:num_detections].numpy()
                   for key,value in detections.items()}

    detections['num_detections']=num_detections
    detections['detection_classes']=detections['detection_classes'].astype(np.int64)
    image_np_with_detections=image_np.copy()

    if 'detection_masks' in detections:
      detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
              detections['detection_masks'], detections['detection_boxes'],
              image_np.shape[0], image_np.shape[1])
      detection_masks_reframed = tf.cast(detection_masks_reframed > 0.2, tf.uint8)
      detections['detection_masks_reframed'] = detection_masks_reframed.numpy()


    threshold = 0.25
    labels = "pellet"
    detection_count = 0

    for i, (y_min, x_min, y_max, x_max) in enumerate(detections['detection_boxes']):
         if detections['detection_scores'][i] > threshold and \
          (labels == None or category_index[detections['detection_classes'][i]]['name'] in labels):
            detection_count += 1

    print('Pellet count: ' + str(detection_count))
```

Figure 2. Python funtion for counting detected fish pellets

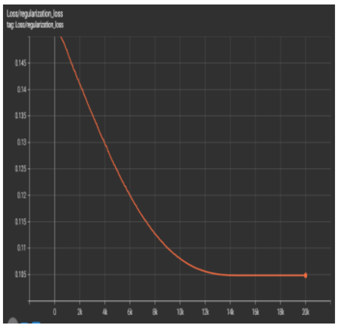
3 RESULTS AND DISCUSSION

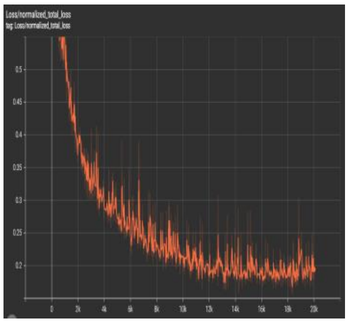
During the training of the model the result of this study shows a significant that is shown in figure 3 below. The Training result from step 18, 100 – step 20,000 produce a total loss of 0.198 in detecting the fish pellets.





a. Classification Loss **b. Localization Loss**





c. Regularization Loss **d. Normalization Loss**

Figure 3. Loss result of a) Classification, b) Localization, c) Regularization and d) Normalization.

In this section, we used three factors to evaluate the performance of the model by dropping pellets in an artificial pond that we setup. The first drop of pellet is composed of enormous number of pellets see figure 4. Once the pellet reaches the target threshold it will drop the second batch of pellets which is composed of average amount of pellets shown in figure 5 and lastly in figure 6 the final drop once it also reaches the desire threshold and drop a lesser number of pellets.

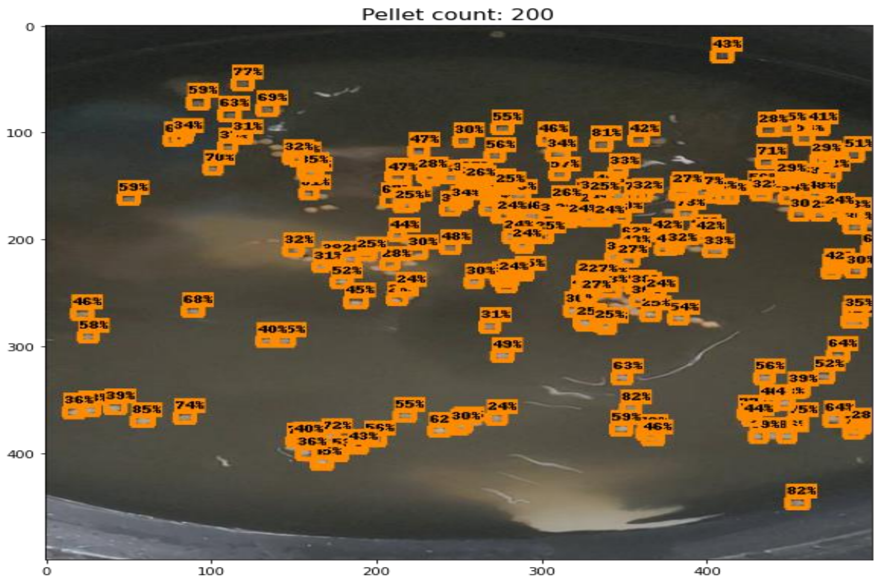


Figure 4. First dropping of pellets.

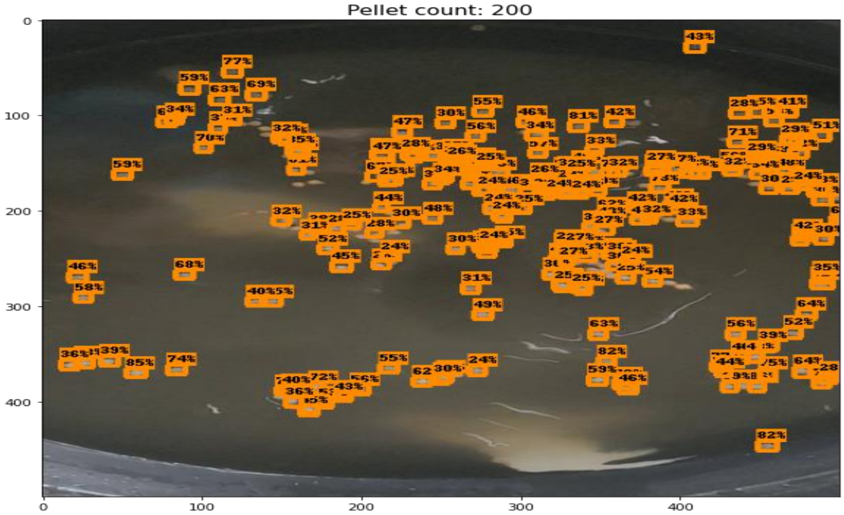


Figure 5. Second drop of pellets.

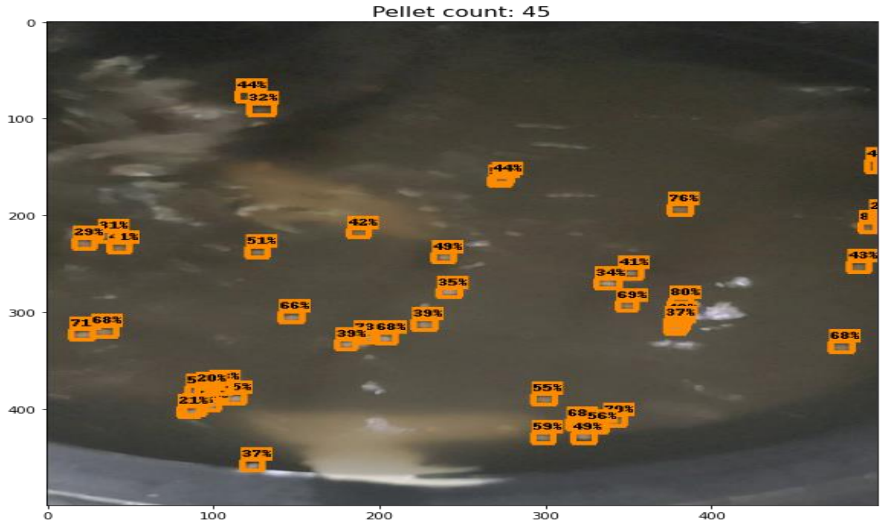


Figure 6. Last dropping of pellets.

4 CONCLUSION

This study tested and benchmarked a model of convolutional neural network (CNN) - based object detection systems, SSD MobileNet v2 for the purpose of detecting the number of pellets in a pond and count it. The overall performance was benchmarked in terms of the losses tested using a common set of experimental trials. The model produced a significant result that detected small objects like fish pellet and count that has gather a minimal loss in terms of classification, localization, regularization and normalization. In the future, we plan on enhancing the model and deploy the smart fish feeder in big areas of ponds.

REFERENCES

1) De Verdal, H.; Komen, H.; Quillet, E.; Chatain, B.; Allal, F.; Benzie, J.A.H.; Vandeputte, M. Improving Feed Efficiency in Fish Using Selective Breeding: A Review. 2018, 10, 833–851.
2) Hu, W.-C.; Wu, H.-T.; Zhang, Y.-F.; Zhang, S.-H.; Lo, C.-H. Shrimp Recognition Using ShrimpNet Based on Convolutional Neural Network. J. Ambient Intelligence Humanized Computing 2020.
3) Liu, Z.; Li, X.; Fan, L.; Lu, H.; Liu, L.; Liu, Y. Measuring Feeding Activity of Fish in RAS Using Computer Vision. Aquacultural Engineering 2014, 60, 20–27
4) Wang, T.; Xu, X.; Wang, C.; Li, Z.; Li, D. From Smart Farming towards Unmanned Farms: A New Mode of Agricultural Production. Agriculture 2021, 11, 145.
5) Akbar, M.O.; Shahbaz Khan, M.S.; Ali, M.J.; Hussain, A.; Qaiser, G.; Pasha, M.; Pasha, U.; Missen, M.S.; Akhtar, N. IoT for Development of Smart Dairy Farming. J. Food Qual. 2020, 2020, 1212805

6) Zhou, C.; Xu, D.; Chen, L.; Zhang, S.; Sun, C.; Yang, X.; Wang, Y. Evaluation of Fish Feeding Intensity in Aquaculture Using a Convolutional Neural Network and Machine Vision. Aquaculture 2019, 507, 457–465.
7) Måløy, H.; Aamodt, A.; Misimi, E. A Spatio-Temporal Recurrent Network for Salmon Feeding Action Recognition from Underwater Videos in Aquaculture. Computer. Electronics. Agriculture. 2019, 167, 105087.
8) Li, Daoliang; Wang, Zhenhu; Wu, Suyuan; Miao, Zheng; Du, Ling Duan, Yanqing. Automatic recognition methods of fish feeding behavior in aquaculture: A review. Aquaculture. 2020, 528, 735508.
9) Zhou, C., Zhang, B., Lin, K., Xu, D., Chen, C., Yang, X., Sun, C., 2017. Near-infrared imaging to quantify the feeding behavior of fish in aquaculture. Computers and Electronic in Agriculture. 2017. 135, 233–241
10) Ye, Z.Y., Zhao, J., Han, Z.Y., Zhu, S.M., Li, J.P., Lu, H.D., Ruan, Y.J., 2016. Behavioral characteristics and statistics-based imaging techniques in the assessment and optimization of tilapia feeding in a recirculating aquaculture system. Trans. ASABE 59 (1), 345–355
11) Bradley, D.; Merrifield, M.; Miller, K.M.; Lomonico, S.; Wilson, J. R.; Gleason, M.G. Opportunities to Improve Fisheries Management through Innovative Technology and Advanced Data Systems. Fish and Fisheries. 2019, 20, 564–583.
12) Schneider, S.; Taylor, G.W.; Linquist, S.; Kremer, S.C. Past, Present and Future Approaches Using Computer Vision for Animal Re-Identification from Camera Trap Data. Methods Ecology and Evolution. 2019, 10, 461–470.
13) Rauf, H.T.; Lali, M.I.U.; Zahoor, S.; Shah, S.Z.H.; Rehman, A.U.; Bukhari, S.A.C. Visual Features Based Automated Identification of Fish Species Using Deep Convolutional Neural Networks. Computers and Electronic in Agriculture 2019, 167, 105075.
14) Zhou, C.; Zhang, B.; Lin, K.; Xu, D.; Chen, C.; Yang, X.; Sun, C. Near-Infrared Imaging to Quantify the Feeding Behavior of Fish in Aquaculture. Computers and Electronic in Agriculture. 2017, 135, 233–241
15) Zhou, C.; Lin, K.; Xu, D.; Chen, L.; Guo, Q.; Sun, C.; Yang, X. Near Infrared Computer Vision and Neuro-Fuzzy Model-Based Feeding Decision System for Fish in Aquaculture. Computers and Electronic in Agriculture. 2018, 146, 114–124.