Knowledge Engineering and Data Science (KEDS)
Vol 5, No 1, December 2022, pp. 67–77

pISSN 2597-4602
eISSN 2597-4637

Fish Image Classification using Transfer Learning Method with Adaptive Learning Rate

Rizka Suhana [1, *], Wayan Firdaus Mahmudy [2], Agung Setia Budi [3]

Faculty of Computer Science, Brawijaya University
Jl. Veteran no. 8, Malang 65145, Indonesia
[1] *rizka28294@student.ub.ac.id* *; [2] *wayanfm@ub.ac.id;* [3] *agungsetiabudi@ub.ac.id*
* *corresponding author*

ARTICLE INFO

Article history:
Received 23 June 2021
Revised 14 July 2022
Accepted 14 August 2022
Published online 7 November 2022

Keywords:
Fish Images
Image Classification
CNN
Transfer Learning
Mobilenet v2

ABSTRACT

The diversity of fish species in coral reef ecosystems is one of the indications in determining health in coral reef ecosystems. Many Indonesian Fisheries and Marine Research and Development Agency experts carefully classify fish images. A reliable technique for performing image classification is Convolutional Neural Network (CNN). Transfer learning appears and adopts part of CNN, namely the modified convolution layer. The paper aims to solve the fish classification problem using the pre-trained model of Mobilenet V2. The model has a low computational process and does not use too many memory resources when training image data. The research image data used is 49,281 data of various sizes and 18 types of fish. The image is entered into the transformation process (random rotation, random resize crop, random horizontal flip) on the training and test data to produce varied data. After the transformation process, the image data is entered into the training process using the Mobilenet V2 architecture. Testing the Mobilenet V2 architectural model obtained an accuracy score of 99.54%, which is reliable in classifying fish images.

I. Introduction

Indonesia is an archipelagic country with a coral reef area of more than 85,700 km^2 [1], directly there is the potential for abundant natural resources and very high biodiversity. Fishery production in Indonesia accounts for more than 50% of which comes from coastal areas, especially from seagrass ecosystems, mangroves, and coral reefs. Indonesia is included in the coral triangle center as the center area of the coral triangle [2]. More than 412 species, including 44 families and 146 genera of fish, have been identified in the Karimun Jawa National Park area, Jepara Regency, Central Java Province [3]. The diversity of reef fish or other organisms living on coral reefs indicates that the ecosystem is healthy [4]. Conservation activities are critical to monitor the coral reef environment regularly.

Conservation data in video, then processed to produce fish image data. The fish image will be analyzed by experts, including what type of fish image is. Experts use the level of diversity of fish species as an indicator of a healthy coral reef ecosystem [5]. The study of Villon et al. [6] obtained an accuracy value of 89.3% in the manual classification of fish images, namely direct observation using the naked eye by researchers, and there may still be errors in classifying what types of fish are in the image.

Image classification is included in the primary research area in image processing, which has broad prospects in various scientific fields such as image segmentation, image recognition, and many more. In k-Nearest Neighbors (KNN) [7], Random Forest [8], and XGBoost [9][10][11][12] are all machine learning methods that can be applied to image classification. In essence, the image classification process depends on feature extraction and feature classification composition. The first is feature extraction, which extracts all features from the image and is stored in tabular form. The second is

https://doi.org/10.17977/um018v5i12022p67-77

feature extraction. Classification, namely deviating the label from the classification image. After going through the feature extraction and classification processes, the data from the image can be processed using each of the above methods.

The application of deep learning methods can be one solution to the problem of fish image classification. Convolutional Neural Networks (CNN) can solve problems related to fish classification, according to the research of Alshdaifat et al. [13] and Cui et al. [14]. In the case of fish image classification, using learning methods that utilize pre-trained models, or transfer learning, is also more efficient than building deep-learning architectural models from scratch [15]. Classification methods on fish images are beneficial for researchers in terms of speed to identify fish [6][16][17]. The pre-trained architecture of the Mobilenet model is reliable for image recognition.

Mobilenet V2 is efficient because it can be inserted into mobile or other vision devices [18]. In the Mobilenet V1 architecture model, using a convolution layer type called depthwise separable convolution makes the computing process on the mobilenet V1 architecture faster than the traditional CNN architecture. The Mobilenet V2 [19] got an update on the following architecture, using Inverted Residual and Linear Bottleneck on the convolution layer in the Mobilenet V2 architecture model.

Models with good performance will undoubtedly depend on optimal hyper-parameters, which will directly affect the performance/performance of the model, so the selection of hyper-parameters becomes very important [11]. One of the hyper-parameters used is learning rate and batch size. The learning rate is a hyper-parameter that controls how fast and slow the learning of the neural network model is to solve problems [19][20]. So there is an update on optimizing an adaptive learning rate that can gradually change to obtain a global minimum [21]. Batch size is a hyper-parameter that controls the accuracy of the estimated gradient error when learning the neural network and controls the speed and stability of the neural network's learning process [22].

Experts need to maintain the diversity of fish species and want to make it easier to classify fish species in the field of conservation. Researchers decided to solve the problems of the experts. Researchers who studied the method from previous research in this study will use the architecture of Mobilenet V2 by combining optimization techniques, namely adaptive learning rate. It is hoped that using the Mobilenet V2 architecture with an adaptive learning rate carried out by researchers can relieve and help experts at the Fisheries and Marine Research and Development Agency.

II. Methods

A. Dataset Searching

The dataset is obtained on the Fish4Knowledge website and a European foundation formed for water conservation [23]. From Table 1, we can know the distribution of fish images. The data is in the form of video recordings with a complete recording of 87.000 hours with a total of 524.000 recordings.

Table 1. Quantity distribution of image

ID	Species	Data	Training (80%)	Testing (20%)
01.	Abudefduf vaigiensis	403	322	81
02.	Acanthurus nigrofuscus	2729	2183	546
03.	Amphiprion clarkii	7034	5627	1407
04.	Chaetodon lunulatus	5028	4022	1006
05.	Chaetodon trifascialis	565	452	113
06.	Chromis chrysura	7186	5748	1438
07.	Dascyllus aruanus	738	590	148
08.	Dascyllus reticulatus	15.308	12246	3062
09.	Hemigymnus fasciatus	238	190	48
10.	Hemigymnus melapterus	189	151	38
11.	Lutjanus fulvus	206	164	42
12.	Myripristis kuntee	3454	2763	691
13.	Neoglyphidodon nigroris	145	116	29
14.	Neoniphon sammara	299	239	60
15.	Pempheris Vanicolensis	78	62	16
16.	Plectroglyphidodon dickii	5139	4111	1028
17.	Pomacentrus moluccensis	181	14	37
18.	Zebrasoma scopas	361	288	73

In this study, the experts tried to do the image by cropping the screen-captured image on the videotape. So get an image dataset of various sizes.

Dataset transformation, the initial process before the data is entered to train the architectural model used, is through several transformation stages. The images in this data have different dimensions. As in Figure 1, the image dimensions are 36x36 pixels. This study uses the pre-trained model to resize the image to 224 × 224 pixels.

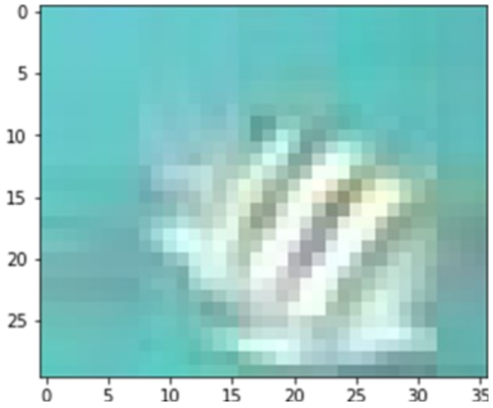


Fig. 1. Species of fish type abudefduf vaigiensis

The data train transformation transforms the fish image on the training data, including random rotation, random resize crop, random horizontal flip, tensor (converting to tensor data), and data normalization. Random Rotation 10°, sets random rotation between left or right with a predetermined degree of inclination. The second setting, Random Resize Crop (1-0.8 scale), randomly changes size with cutting with a predetermined scale between 1 and 0.8. Random Horizontal Flip rotates the fish images horizontally randomly. The next step converts previous data into tensor data (PyTorch). The last step, data normalization, used for transforming training data, is normalizing tensor data according to data normalization in the Mobilenet V2 architectural model. Mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225] on each image channel which has three channels (RGB).

The data test transformation for the fish image on the test data includes random resize crop, random center crop, tensor (convert to tensor data), and data normalization. In the image, transformation test data is not done flipping because as much as possible to approach the image according to the original image. Resize process changes the image size to 230x230 pixels because the fish image test data have different sizes. The Second is Center Crop to change the resized image to 230x230 pixels and then crop it in the center to 224x224 pixels. The next is Convert previous data into tensor data (PyTorch). The last step for transforming training data is normalizing tensor data according to data normalization in the Mobilenet V2 architectural model. Mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225] on each image channel which has three channels (RGB).

The Image Structuring phase transforms the image data into a tabular or tabular dataset by extracting / flattening and featuring each image pixel, as shown in Figure 2. The Figure extracted each pixel of the image to get the feature data of the image data. Label the fish image using the folder name of the extracted image as a record. Figure 3 shows Feature images and labels from images, 12289 features of a 3-channel image with a size of 64 × 64 pixels.

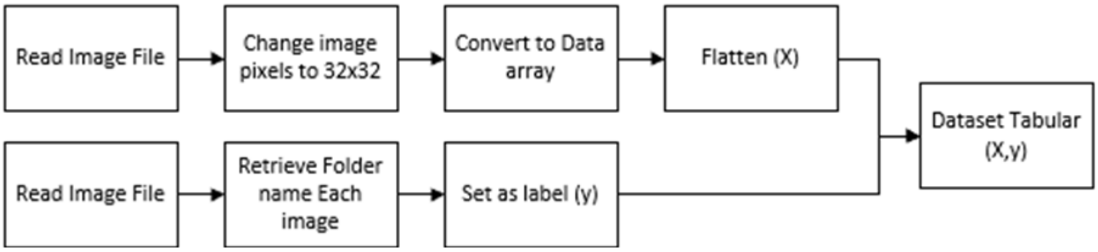
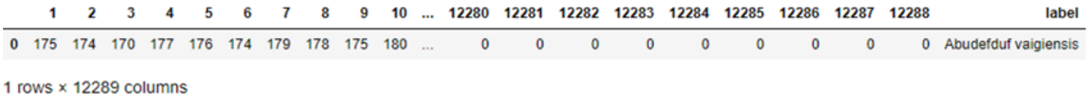


Fig. 2 Flowchart to create metadata for machine learning



```
     1    2    3    4    5    6    7    8    9   10  ...  12280  12281  12282  12283  12284  12285  12286  12287  12288                label
0  175  174  170  177  176  174  179  178  175  180  ...      0      0      0      0      0      0      0      0      0  Abudefduf vaigiensis

1 rows × 12289 columns
```

Fig. 3 Feature images and labels from images

B. Architecture Configuration

The model architecture used in this research is Mobilenet V2 (Sandler et al., 2018), with modifications to the previous classification layer to classify 1000 types of images. The architectural model of Mobilenet V2 consists of a complete convolution layer with 32 filters and 19 residual bottleneck layers. Modifying the classification layer is changing to a fully connected layer with an input layer of 1280 and an output layer of 18. The architecture of Mobilenet V2 follows Table 2. input is the initial image size before entering into the convolution process, the operator is a simple name of the convolution layer, t is the expansion factor, c is output, n is repeated times of convolutions layer, s are strides.

Table 2 Architecture Mobilenet V2

input	Operator	t	c	n	s
224^2 x 3	conv2d	-	32	1	2
112^2 x 32	bottleneck	1	16	1	1
112^2 x 16	bottleneck	6	24	2	2
56^2 x 24	bottleneck	6	32	3	2
28^2 x 32	bottleneck	6	64	4	2
14^2 x 64	bottleneck	6	96	3	1
14^2 x 94	bottleneck	6	160	3	2
7^2 x 160	bottleneck	6	320	1	1
7^2 x 320	conv2d 1x1	-	1280	1	1
7^2 x 1280	avrgpool 7x7	-	-	1	-
1^2 x 1280	conv2d 1x1	-	k	-	

From the Mobilenet V2 architecture table above, it can be simplified further in the Figure below by describing all parts of the Inverted Residual Block as one bottleneck, which will result in feature extraction, and the final layer, there is a classification layer. This research uses transfer learning architecture with an adaptive learning rate, and the architectural model of transfer learning that will be used is the pre-trained model from Mobilenet V2. Figure 4 is a simple description of the architecture above.

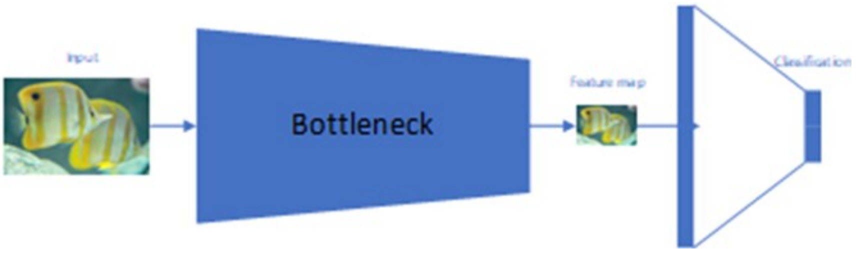


Fig. 4 Simple architectural model in this research

The steps in modeling the Mobilenet V2 architecture are the Input layer, the first layer in the architectural model as an input layer, and a fish image that has gone through image data pre-processing. A bottleneck is a simple arrangement described in the architectural model in which various layers comprise the V2 Mobilenet architectural model. 19 layers comprise the bottleneck. One is the depthwise convolution layer and pointwise convolution layer using skip connection in each layer. Flatten layer changes the results of the feature map in the previous layer into features that can later be processed on the neural network. The last layer is used for the classification process to determine the class the processed image belongs to.

C. Optimizer (AdamW)

The AdamW optimizer is an Adam optimization combined with L2 regularization and weight decay [21], while the Adam Optimizer is an optimization algorithm that replaces Stochastic gradient descent in the deep learning model training stage. Adam's optimization represents the best properties of other optimization algorithms, such as AdaGrad and RMSProp, which have the advantage of an adaptive learning rate. AdamW algorithm, using hyperparameter α=0.001, β_1=0.9, β_2=0.999, ε= 10^-^{8}, λ ∈ R. Hyperparameters are pre-set, and parameters t ←0, the first-moment vector is initialized to the value of 0 (m_t ←0), The second-moment vector is also initialized to the value of 0 (v_t ←0) and the schedule multiplier parameter is set to zero (η_t ←0 ∈ R). n the AdamW algorithm, the parameter t will increase as the number of iterations increases, as in (1).

$$t \leftarrow t + 1 \tag{1}$$

Then add the derivative formula of gradient loss to weight in (2).

$$g_t \leftarrow \frac{\partial L_t}{\partial W_{i,t}} + \lambda \frac{\partial L_t}{\partial W_{i,t}}\ (ignore\ the\ value\ of\ i) \tag{2}$$

Next, the first step is a formula similar to the momentum m_t in (3), and the second step is the same as RMSProp v_t in (4).

$$m_{i,t}\ =\ \beta_1 m_{i-1,t}\ + (1-\beta_1) g_t \tag{3}$$

$$v_{i,t}\ =\ \beta_2 m_{i-1,t}\ + (1-\beta_2)(g_t)^2 \tag{4}$$

Of course, the AdamW algorithm still has a technique to do *bias correction* by adding a formula to escape the value of $m_t\ and\ v_t$, being unbiased to 0 or close to 0. The following explains if, in the first iteration $(t = 1)$, the momentum and RMSProp values are given a 0. So there is an additional formula in the next step to avoid bias in the initial iteration, that is $\widehat{m}_{i,t}$ (m hat, m hat, as momentum in (5)) dan $\hat{v}_{i,t}$ (v hat, as RMSProp in (6)).

$$\widehat{m}_{i,t} = \frac{m_{i,t}}{1-\beta_1^t} \tag{5}$$

$$\hat{v}_{i,t} = \frac{v_{i,t}}{1-\beta_2^t} \tag{6}$$

Then (7), as an update of the weight on AdamW, the new weight equals the old weight subtracted from the multiplication of the coefficient η with $\alpha\widehat{m}_{i,t}$ divided by $\sqrt{\hat{v}_{i,t}} +\ \varepsilon$ then added $\lambda g_{\boldsymbol{t}}$.

$$W_{i,t}\ =\ W_{i-1,t}\ + \eta_t \left(\frac{\alpha\widehat{m}_{i,t}}{\sqrt{\hat{v}_{i,t}} + \varepsilon}\right) \lambda gt \tag{7}$$

III. Results and Discussion

A. Learning Rate Testing

The recommended learning rate from the testing process is between 0.1 to $1e^{-6}$, and $1.74e^{-3}$ is obtained. The learning rate is an essential component that must be considered if the learning rate is too large, then we will not reach a minimal global loss, but on the contrary, if the learning rate is too low, it will take too long to reach the global minimum and even get stuck in the local minimum.

Figure 5 is the recommended learning rate for use in the architectural model of learning transfer modification because, according to Smith's research [24], the ideal learning rate is neither too large nor too small. The value of the learning rate of $1.74e^{-3}$ is the better choice in this study.

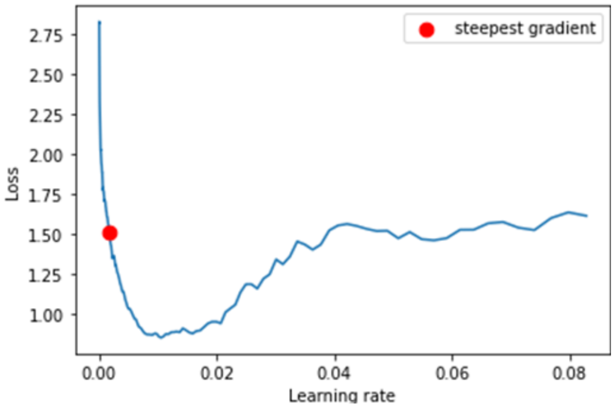


Fig. 5. Suggested learning rate

B. Batch Size Testing

In the Batch size test results, researchers get different test results on train costs, test fees, train scores, test scores, and the number of epochs. The result is shown in Table 3 and Table 4.

Table 2. Phase 1 (adaptation) batch size test results

Batch Size	Train_cost	Test_cost	Train_score	Test_score	Epoch
64	0.3885	0.4139	0.8735	0.8708	7
128	0.1759	0.1830	0.9402	0.9407	17
256	0.2251	0.2341	0.9263	0.9270	8

The phase 1 (adaptation) batch size test results in Table 3 show that the small batch size value affects the test cost and test score because, with at least training data in 1 iteration, it will affect the results. Meanwhile, for large batch sizes (256), more training data will be obtained in 1 iteration, the excellent test cost and test scores evidence this. The model's number of epochs on a batch size 64 is stuck at the local minimum.

Table 3. Phase 2 batch size test results

Batch Size	Train_cost	Test_cost	Train_score	Test_score	Epoch
64	0.0052	0.0287	0.9985	0.9921	48
128	0.0018	0.0167	0.9994	0.9947	46
256	0.0038	0.0204	0.9994	0.9938	36

The results of the batch size test in phase 2 in Table 4, the value of train cost, test cost, train score, and test score, get good results. The most striking change is in the epoch section, which decreases in large batch sizes.

From testing phase 1 of adaptation and phase 2, the value of the size of the learning rate and early stopping is efficient with the accuracy value obtained. As in phase 1 adaptation with learning rate = 0.001, indeed, with a small batch size value will get a small accuracy value as well because it is affected by early stopping, which will stop the training process when the accuracy value is not increased [20].

C. Performance of the model modified transfer learning

Performance testing using the architectural model of the transfer learning modification in phase 1 and phase 2, the result will be shown in Table 5 and Table 7. Table 5 shows the highest accuracy value of 93.85% with two early stops. In this phase, the values obtained in training and validation are not much different, neither overfit nor underfit. Table 5 can be visualized as a graph, as shown in Figure 6.

Table 4. Accuracy and loss values in phase 1 (adaptation)

Epoch	**Training**		**Validasi**	
	Avg accuracy	***Avg loss***	***Avg accuracy***	***Avg loss***
1	89.28	0.36	89.16	0.38
2	90.97	0.29	91.05	0.29
3	92.31	0.25	91.60	0.27
4	92.87	0.23	92.05	0.25
5	92.88	0.22	92.33	0.24
6	93.52	0.20	92.69	0.23
7	93.87	0.19	93.32	0.21
8	93.94	0.19	93.16	0.22
9	93.98	0.18	93.49	0.20
10	93.81	0.19	92.94	0.21
11	94.42	0.17	93.64	0.20
12	94.38	0.17	93.85	0.19
13	94.39	0.17	93.24	0.20
14	94.64	0.16	93.62	0.20

In Figure 6, the accuracy value of training and validation, when viewed from Table 5 there is no significant difference, but if seen in Figure 6, it is evident because of the effect of several parameters that have been prepared above, such as early stopping and learning rate.

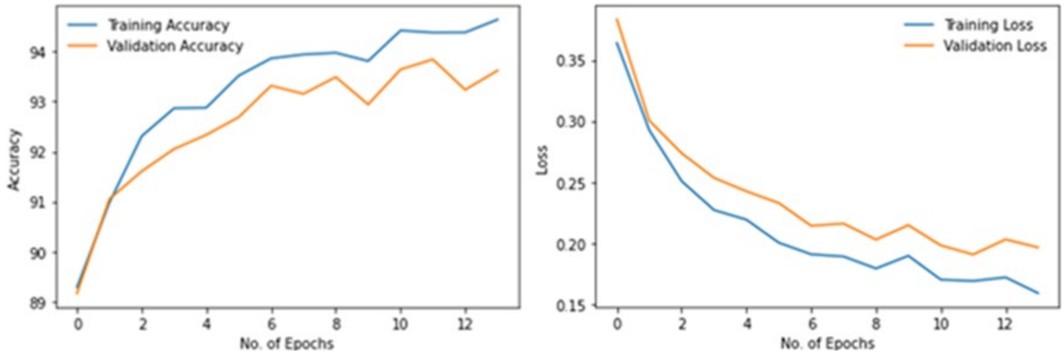
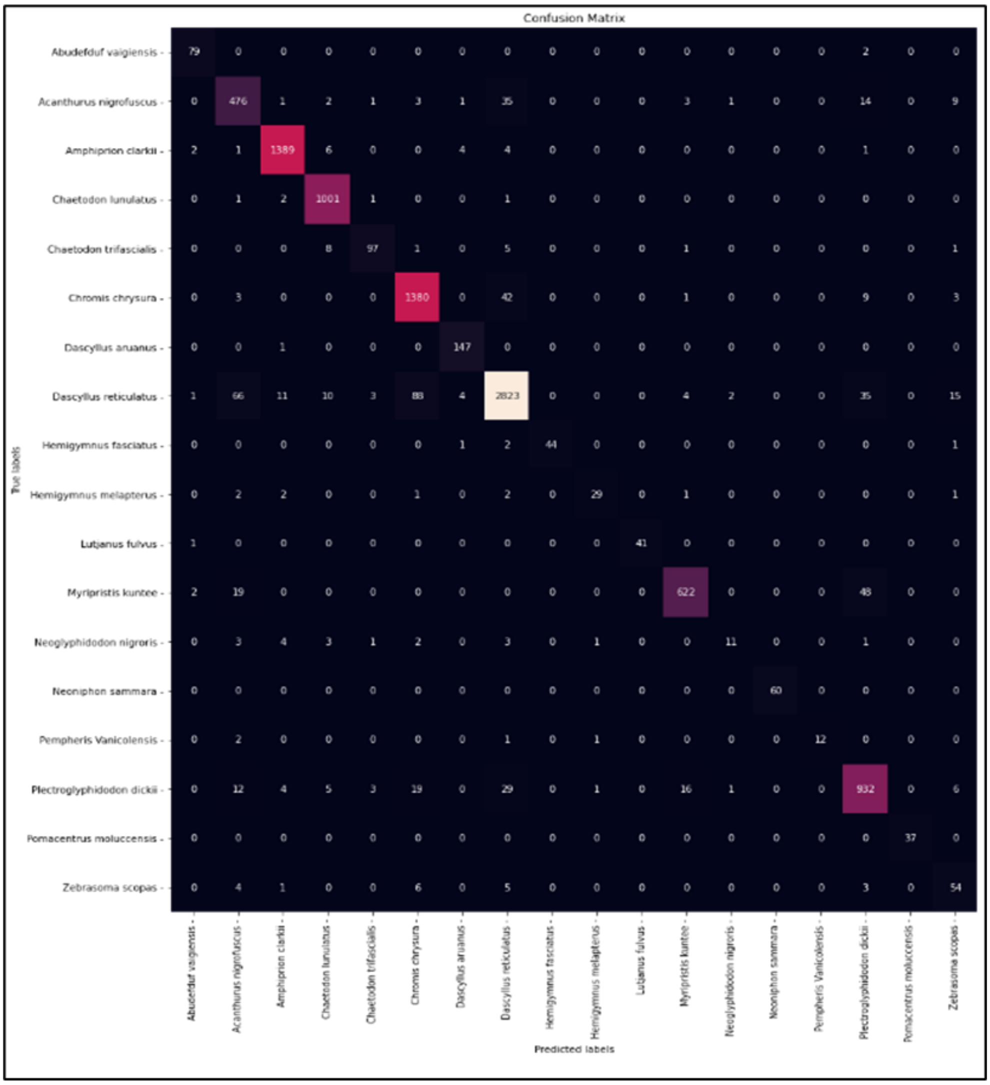


Fig. 6. Phase 1 accuracy and loss graph

This research section will discuss the performance of the prediction model that the researcher uses. A confusion matrix, one of which is used in the prediction model in supervised learning. The function of the confusion matrix is one of the benchmarks for evaluating the supervision learning model, namely by calculating accuracy, precision, sensitivity/recall. From the results of the prediction, values obtained a confusion matrix in phase 1 (adaptation) as shown in Figure 7 and the calculations for Accuracy, precision, sensitivity/recall in the confusion matrix phase 1 (adaptation). The above confusion matrix calculation is described in tabular form according to Table 6.

Table 5 Classification report phase 1

Class	**precision**	**recall**	**f1-score**
Abudefduf vaigiensis	0,929	1	0,963
Acanthurus nigrofuscus	0,808	0,886	0,845
Amphiprion clarkii	0,981	0,987	0,983
Chaetodon lunulatus	0,967	0,995	0,98
Chaetodon trifascialis	0,915	0,788	0,846
Chromis chrysura	0,92	0,959	0,939
Dascyllus aruanus	0,936	0,993	0,963
Dascyllus reticulatus	0,956	0,921	0,938
Hemigymnus fasciatus	1	0,916	0,956
Hemigymnus melapterus	0,906	0,763	0,829
Lutjanus fulvus	1	0,976	0,987
Myripristis kuntee	0,959	0,9	0,928
Neoglyphidodon nigroris	0,733	0,379	0,499
Neoniphon sammara	1	1	1
Pempheris Vanicolensis	1	0,75	0,857
Plectroglyphidodon dickii	0,891	0,906	0,898
Pomacentrus moluccensis	1	1	1
Zebrasoma scopas	0,6	0,739	0,662
Average	0,916	0,881	0,892



	Abudefduf vaigiensis	Acanthurus nigrofuscus	Amphiprion clarkii	Chaetodon lunulatus	Chaetodon trifascialis	Chromis chrysura	Dascyllus aruanus	Dascyllus reticulatus	Hemigymnus fasciatus	Hemigymnus melapterus	Lutjanus fulvus	Myripristis kuntee	Neoglyphidodon nigroris	Neoniphon sammara	Pempheris Vanicolensis	Plectroglyphidodon dickii	Pomacentrus moluccensis	Zebrasoma scopas
Abudefduf vaigiensis	79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Acanthurus nigrofuscus	0	476	1	2	1	3	1	35	0	0	0	3	1	0	0	14	0	9
Amphiprion clarkii	2	1	1389	6	0	0	4	4	0	0	0	0	0	0	0	1	0	0
Chaetodon lunulatus	0	1	2	1001	1	0	0	1	0	0	0	0	0	0	0	0	0	0
Chaetodon trifascialis	0	0	0	8	97	1	0	5	0	0	0	1	0	0	0	0	0	1
Chromis chrysura	0	3	0	0	0	1380	0	42	0	0	0	1	0	0	0	9	0	3
Dascyllus aruanus	0	0	1	0	0	0	147	0	0	0	0	0	0	0	0	0	0	0
Dascyllus reticulatus	1	66	11	10	3	88	4	2823	0	0	0	4	2	0	0	35	0	15
Hemigymnus fasciatus	0	0	0	0	0	0	1	2	44	0	0	0	0	0	0	0	0	1
Hemigymnus melapterus	0	2	2	0	0	1	0	2	0	29	0	1	0	0	0	0	0	1
Lutjanus fulvus	1	0	0	0	0	0	0	0	0	0	41	0	0	0	0	0	0	0
Myripristis kuntee	2	19	0	0	0	0	0	0	0	0	0	622	0	0	0	48	0	0
Neoglyphidodon nigroris	0	3	4	3	1	2	0	3	0	1	0	0	11	0	0	1	0	0
Neoniphon sammara	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0
Pempheris Vanicolensis	0	2	0	0	0	0	0	1	0	1	0	0	0	0	12	0	0	0
Plectroglyphidodon dickii	0	12	4	5	3	19	0	29	0	1	0	16	1	0	0	932	0	6
Pomacentrus moluccensis	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0
Zebrasoma scopas	0	4	1	0	0	6	0	5	0	0	0	0	0	0	0	3	0	54

Fig. 7. Confusion matrix in phase 1 (adaptation)

In Table 7, the highest accuracy value is 99.54375%, and the training accuracy value touches the value of 99.92389% in two early stops, so the values obtained in training and validation are overfitting at this stage. However, Overfitting does not make a big difference from Table 7, you can visualize it in the form of a graph as shown in Figure 8.

Table 6. Accuracy and loss values in phase 2

Epoch	Training		Validasi	
	Avg accuracy	*Avg loss*	*Avg accuracy*	*Avg loss*
1	90.27348	0.42452	89.87	0.44
2	90.21006	0.42567	89.87124	0.44435
3	94.33254	0.23217	93.69	0.25
4	94.22853	0.23249	93.69360	0.25335
5	96.58278	0.14528	96.04	0.17
…	…	…	….	…
27	99.91374	0.00455	99.54	0.02
28	99.92389	0.00455	99.54375	0.01831
29	99.93911	0.00417	99.49	0.02
30	99.89852	0.00456	99.49305	0.01938
31	99.93911	0.00370	99.52	0.02
32	99.94672	0.00359	99.52347	0.01674
33	99.94926	0.00339	99.52	0.02

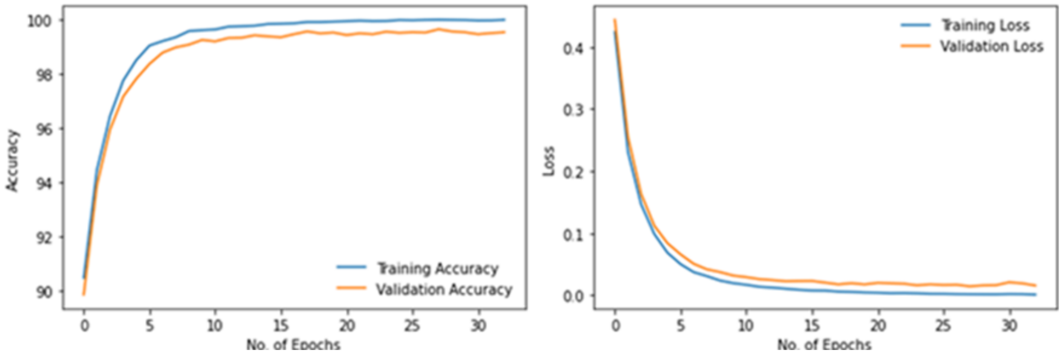
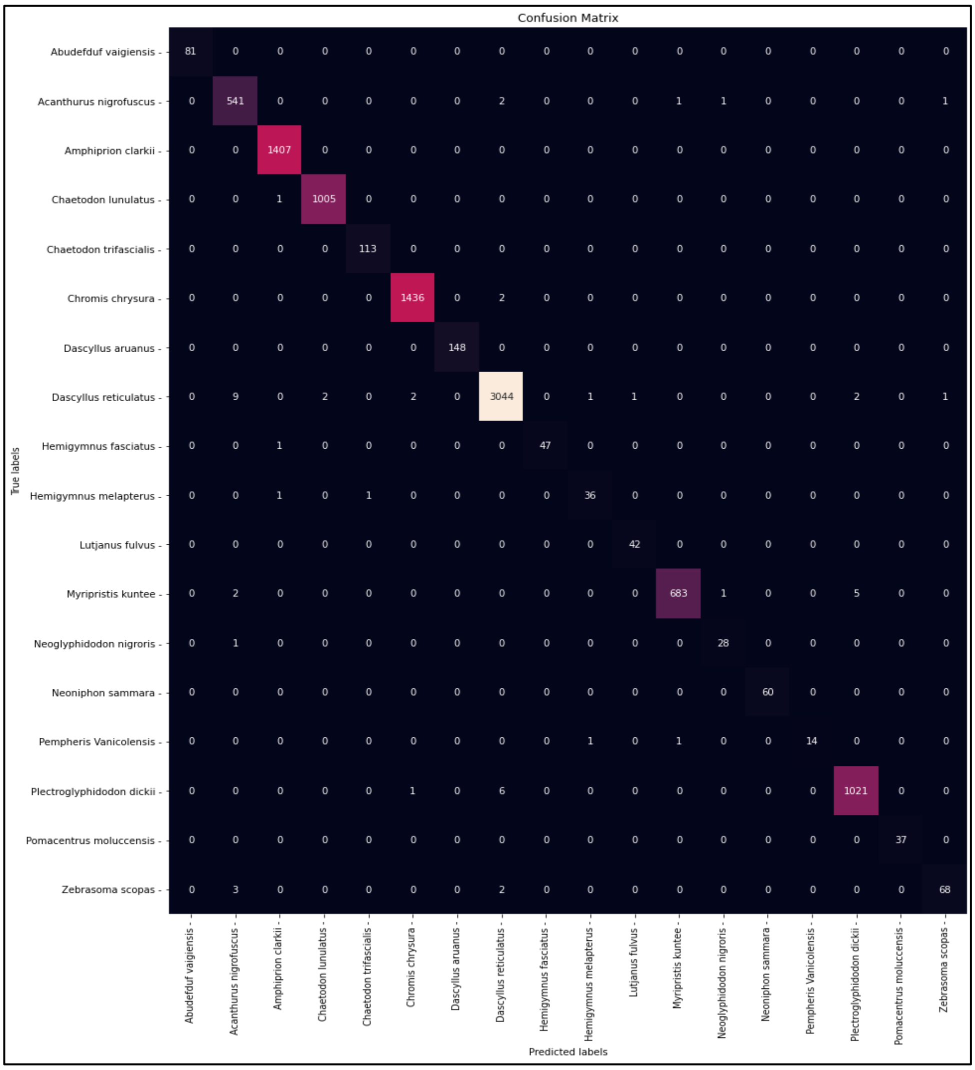


Fig. 8. Phase 2 accuracy and loss graph

In Figure 8, the training and validation accuracy values are in Table 7 show a smooth graph until the difference is less significant, as small learning rates can reach the global minimum. Figure 9 shows Phase 2, where the performance of the fish image classification system in the transfer learning process is analyzed using the Mobilenet V2 architecture. The modified transfer learning architecture model has improved performance, decreased FN and FP values, and increased TP values. Following are the calculations for Accuracy, precision, sensitivity/recall in the phase 2 confusion matrix. The above confusion matrix at Figure 9 calculation is described in tabular form according to Table 8.



Confusion Matrix

True labels

	Abudefduf vaigiensis	Acanthurus nigrofuscus	Amphiprion clarkii	Chaetodon lunulatus	Chaetodon trifascialis	Chromis chrysura	Dascyllus aruanus	Dascyllus reticulatus	Hemigymnus fasciatus	Hemigymnus melapterus	Lutjanus fulvus	Myripristis kuntee	Neoglyphidodon nigroris	Neoniphon sammara	Pempheris Vanicolensis	Plectroglyphidodon dickii	Pomacentrus moluccensis	Zebrasoma scopas
Abudefduf vaigiensis	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Acanthurus nigrofuscus	0	541	0	0	0	0	0	2	0	0	0	1	1	0	0	0	0	1
Amphiprion clarkii	0	0	1407	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chaetodon lunulatus	0	0	1	1005	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chaetodon trifascialis	0	0	0	0	113	0	0	0	0	0	0	0	0	0	0	0	0	0
Chromis chrysura	0	0	0	0	0	1436	0	2	0	0	0	0	0	0	0	0	0	0
Dascyllus aruanus	0	0	0	0	0	0	148	0	0	0	0	0	0	0	0	0	0	0
Dascyllus reticulatus	0	9	0	2	0	2	0	3044	0	1	1	0	0	0	0	2	0	1
Hemigymnus fasciatus	0	0	1	0	0	0	0	0	47	0	0	0	0	0	0	0	0	0
Hemigymnus melapterus	0	0	1	0	1	0	0	0	0	36	0	0	0	0	0	0	0	0
Lutjanus fulvus	0	0	0	0	0	0	0	0	0	0	42	0	0	0	0	0	0	0
Myripristis kuntee	0	2	0	0	0	0	0	0	0	0	0	683	1	0	0	5	0	0
Neoglyphidodon nigroris	0	1	0	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0
Neoniphon sammara	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0
Pempheris Vanicolensis	0	0	0	0	0	0	0	0	0	1	0	1	0	0	14	0	0	0
Plectroglyphidodon dickii	0	0	0	0	0	1	0	6	0	0	0	0	0	0	0	1021	0	0
Pomacentrus moluccensis	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0
Zebrasoma scopas	0	3	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	68

Predicted labels

Fig. 9. Confusion matrix in phase 2

Table 7 Classification report phase 2

Class	Precision	Recall	F1-score
Abudefduf vaigiensis	1	1	1
Acanthurus nigrofuscus	0,973	0,99	0,981
Amphiprion clarkii	0,997	1	0,998
Chaetodon lunulatus	0,998	0,9	0,946
Chaetodon trifascialis	0,9	1	0,947
Chromis chrysura	0,997	0,998	0,997
Dascyllus aruanus	1	1	1
Dascyllus reticulatus	0,996	0,994	0,994
Hemigymnus fasciatus	1	0,979	0,989
Hemigymnus melapterus	0,947	0,947	0,947
Lutjanus fulvus	0,976	1	0,987
Myripristis kuntee	0,997	0,988	0,922
Neoglyphidodon nigroris	0,933	0,965	0,948
Neoniphon sammara	1	1	1
Pempheris Vanicolensis	1	0,875	0,933
Plectroglyphidodon dickii	0,993	0,933	0,933
Pomacentrus moluccensis	1	1	1
Zebrasoma scopas	0,985	0,931	0,957
Average	**0,982**	**0,972**	**0,971**

D. Testing With Other AI Models

In this section, researchers compare machine learning and deep learning models. That is by using Traditional CNN, which has five convolution blocks and two hidden layer blocks with softmax function activation. The Convolution block has 3x3 layer filters, with stride = 1, padding = 1, function activation = ReLU, and type pooling = max pool.

From Table 9, the Modified transfer learning model gives the best results have some reason. Use a pre-trained architectural model (which has been trained previously). The data trained on the previous architecture and the data used by the researcher are not too different because the pre-trained model architecture used has been trained on 1000 different types of images. Traditional CNNs are computationally faster to train images than the transfer learning modifications that use the inverted residual layer, although they differ slightly from the transfer learning modifications used by researchers. Machine learning models from KNN, Random Forest, and XGBoost did not achieve accuracy values over 90%, but machine learning models were already suitable for classifying fish images. However, the data structuring process from image data / unstructured data to tabular/structured data still takes much time.

Table 8. Benchmarking table with machine learning model

No	Method	Accuracy
1	Modified Transfer Learning	99,64%
2	Traditional CNN	98,58%
3	KNN	85,5%
4	Random Forest	81,63%
5	XGBoost	86,55%

IV. Conclusion

This study aims to classify fish images and use transfer learning modifications to determine the best performance. Using a pre-trained model from Mobilenet, you can modify the classification layer to provide modified transfer learning results. Traditional CNNs can be used to classify fish images, but the design of hidden layers is time-consuming and requires much computation. Therefore, you can use modified transfer learning to solve the problem. The modified transfer learning performance and confusion matrix test results are excellent. When testing Phase 1, accuracy rating = 0.8751; accuracy value = 0.9355; recall / sensitivity value = 0.93055. In Phase 2 testing, accuracy value = 0.9895; accuracy value = 0.9947; recall / sensitivity value = 0.9947. Based on the study's results, we can conclude that modified transfer learning can be the best model.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information are available at http://journal2.um.ac.id/index.php/keds.

Publisher's Note: Department of Electrical Engineering - Universitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

[1] J. P. Schulze Rojas, "Reef front heterogeneity analysis and coral genera diversity pattern in the Bunaken National Park, Indonesia." 2010.

[2] I. Asaad, C. J. Lundquist, M. V. Erdmann, and M. J. Costello, "Delineating priority areas for marine biodiversity conservation in the Coral Triangle," Biol. Conserv., vol. 222, pp. 198–211, Jun. 2018.

[3] E. Yuliana, I. Farida, Nurhasanah, M. Boer, A. Fahrudin, and M. M. Kamal, "Habitat quality and reef fish resources potential in Karimunjawa National Park, Indonesia," AACL Bioflux, vol. 13, no. 4, pp. 1836–1848, 2020.

[4] I. Cáceres, E. C. Ibarra-García, M. Ortiz, M. Ayón-Parente, and F. A. Rodríguez-Zaragoza, "Effect of fisheries and benthic habitat on the ecological and functional diversity of fish at the Cayos Cochinos coral reefs (Honduras)," Mar. Biodivers., vol. 50, no. 1, p. 9, Feb. 2020.

[5] B. J. Boom et al., "Long-term underwater camera surveillance for monitoring and analysis of fish populations," Work. Vis. Obs. Anal. Anim. Insect Behav. (VAIB), conjunction with ICPR 2012, no. August 2015, pp. 2–5, 2012.

[6] S. Villon et al., "A Deep learning method for accurate and fast identification of coral reef fishes in underwater images," Ecol. Inform., vol. 48, no. August, pp. 238–244, 2018.

[7] S. Winiarti, F. I. Indikawati, A. Oktaviana, and H. Yuliansyah, "Consumable Fish Classification Using k-Nearest Neighbor," IOP Conf. Ser. Mater. Sci. Eng., vol. 821, no. 1, p. 012039, Apr. 2020.

[8] Z. Jin, J. Shang, Q. Zhu, C. Ling, W. Xie, and B. Qiang, "RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12343 LNCS, 2020, pp. 503–515.

[9] Y. C. Chang, K. H. Chang, and G. J. Wu, "Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions," Appl. Soft Comput. J., vol. 73, pp. 914–920, 2018.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-Augu, pp. 785–794, 2016.

[11] W. Jiao, X. Hao, and C. Qin, "The Image Classification Method with CNN-XGBoost Model Based on Adaptive Particle Swarm Optimization," Information, vol. 12, no. 4, p. 156, Apr. 2021.

[12] J. Brownlee, XGBoost With Python Gradient Boosted Trees With XGBoost and Scikit-learn. 2018.

[13] N. F. F. Alshdaifat, A. Z. Talib, and M. A. Osman, "Improved deep learning framework for fish segmentation in underwater videos," Ecol. Inform., vol. 59, no. May, p. 101121, 2020.

[14] S. Cui, Y. Zhou, Y. Wang, and L. Zhai, "Fish Detection Using Deep Learning," Appl. Comput. Intell. Soft Comput., vol. 2020, 2020.

[15] B. S. Rekha, G. N. Srinivasan, S. K. Reddy, D. Kakwani, and N. Bhattad, Fish detection and classification using convolutional neural networks, vol. 1108 AISC, no. July. Springer International Publishing, 2020.

[16] F. Kratzert and H. Mader, "Fish species classification in underwater video monitoring using Convolutional Neural Networks," 2018.

[17] D. Li, Z. Wang, S. Wu, Z. Miao, L. Du, and Y. Duan, "Automatic recognition methods of fish feeding behavior in aquaculture: A review," Aquaculture, vol. 528, p. 735508, 2020.

[18] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.

[19] J. Brownlee, Better Deep Learning. Train Faster, Reduce Overfitting, and Make Better Predictions, vol. 1.3, no. 0. 2019.

[20] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning: Machine Learning Book. 2016.

[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 7th Int. Conf. Learn. Represent. ICLR 2019, 2019.

[22] D. Masters and C. Luschi, "Revisiting Small Batch Training for Deep Neural Networks," pp. 1–18, 2018.

[23] B. J. Boom, P. X. Huang, J. He, and R. B. Fisher, "Supporting ground-truth annotation of image datasets using clustering," Proc. - Int. Conf. Pattern Recognit., no. January, pp. 1542–1545, 2012.

[24] L. N. Smith, "Cyclical learning rates for training neural networks," Proc. - 2017 IEEE Winter Conf. Appl. Comput. Vision, WACV 2017, no. April, pp. 464–472, 2017.