International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Incorporating Transfer Learning Strategy for improving Semantic Segmentation of Epizootic Ulcerative Syndrome Disease Using Deep Learning Model

Anbang[1] and Gede Putra Kusuma[1]

[1] Computer Science Department, Bina Nusantara University, Jakarta, Indonesia

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Automated fish disease detection can eliminate the need for manual labor and provides earlier detection of fish disease such as EUS (Epizootic Ulcerative Syndrome) before it further spreads throughout the water. One of the problems that is faced on implementing a semantic segmentation fish disease detection system is the limited size of the semantic segmentation dataset. On the other hand, classification datasets for fish disease detections are more common and available in larger sizes, which cannot be used in segmentation tasks directly since it lacks the necessary label for such tasks. In this paper, we propose a training strategy based on transfer learning to learn from both ImageNet and classification dataset before being trained on the segmentation dataset. Specifically, we first train the ImageNet pre-trained VGG16 on a classification task with the classification dataset, then we transfer the weights into a semantic segmentation architectures such as U-Net and SegNet, and finally train the segmentation network on a segmentation task with the segmentation dataset. We modify the U-Net architecture so that the pre-trained VGG16 weights can be transferred into the architecture. We used a classification dataset containing 304 images of fish diseases for classification task and a segmentation dataset containing 25 images of EUS-affected fishes for the segmentation task. The proposed training strategy is then compared with alternative training strategies such as training VGG16 on ImageNet alone or classification dataset alone. When applied to SegNet and U-Net, the proposed training strategy surpasses their respective architecture trained on ImageNet or classification dataset alone. Between these two architectures with all compared training strategies, the SegNet architecture trained with our proposed training strategy achieves the best performance with validation and testing mIoU of 66.53% and 63.46%, respectively.

Keywords: Fish Disease Detection, Semantic Segmentation, Transfer Learning, U-Net Model, SegNet Model.

1. Introduction

Due to the geography of the country, fish has become one of the major source of income in Indonesia. According to BPS, fish production has increased from 15.24 million tons in year 2017 to 16.12 million tons in year 2018 [1].

Like many other animals, most fish are susceptible to fish diseases. One of the diseases is Epizootic Ulcerative Syndrome (EUS), which is caused by *Aphanomyces invadans*. This disease is easily identified by the red spots that appear on the fish body, hence this disease is also known as Red Spot disease. Fish that are affected by EUS will start to lose appetite, thus consuming less feed and growing slower. The affected fish may also even die, with a mortality rate of around 20-80% [2]. Furthermore, EUS can spread from the infected fish to other healthy fishes in the same body of water. Therefore, early detection of fish disease can potentially prevent further infections and mortalities in the affected fish pond.

One conventional approach for detecting fish diseases is to manually monitor the fish. The person observes the fish in the fish tank and notices for any anomalies visible on the fish skin. However, this approach is time-consuming and requires an individual that is capable at identifying fish diseases.

Computer vision is a study that tries to mimic the human's capability of recognizing images through the use of a computer. While traditional computer vision methods rely on algorithm selected to extract features from images such as edge detection, more recent computer vision methods employ deep learning neural network to automatically extract features from images, without the need to manually selecting certain features.

There exists many works that are aimed at detecting fish diseases through image classification from hand-

E-mail: anbang@binus.ac.id, inegara@binus.edu

crafted features [3][4] and image classification using deep neural network [5][6][7]. There are also other works that also aims to detect fish disease through image segmentation [8][9], although image segmentation approaches are far less common than image classification approaches.

One of the problems faced by one previous work was the limited sample size of the semantic segmentation dataset, which resulted in the model performing worse on testing data split [8]. Semantic segmentation datasets are annotated with pixel-level labels. That is, the label is applied to each individual pixel by creating a mask of different classes in the image. Due to the complexity and level of expertise required to annotate pixel-level labels, they are more costly to annotate. Thus, they are less common and available in smaller sample sizes.

One common method used to overcome the dataset small sample sizes is to employ data augmentation. While data augmentation can help to increase the model performance slightly, the synthetic data created from data augmentation does not introduce new features and as such the model can still overfit from the limited features.

On the other hand, classification datasets are more common and available in larger sample sizes. In contrast to segmentation datasets, classification datasets are annotated with image-level labels. That is, the label is applied to the whole image rather than the individual pixels. image-level labels are easier to annotate than pixel-level labels. However, image-level datasets are incompatible with segmentation models which produce pixel-level outputs and cannot be directly used in segmentation tasks. Therefore, incorporating the features from the more abundant image-level datasets into segmentation models can be seen as a challenge.

An approach for a model to learn both image-level and pixel-level features is to build an architecture with one shared encoder and two outputs, one for classification, and another for segmentation. The shared encoder learns from input data by summing the losses from the respective label output. This approach is also known as Multi-task learning. Several works have proposed Multi-task learning architectures for classification and segmentation tasks [10][11][12]. Such architectures are more complicated to build.

On the other hand, there are several works that have applied transfer learning based on ImageNet for semantic segmentation problems [13][14][15]. Transfer learning is a method of reusing the knowledge on a different yet related domain to the target domain. This is done by transferring the weights from a pre-trained network to another compatible network.

The idea of this work is based on the observations that most of the previous works related to transfer learning uses networks that were pre-trained on ImageNet, which is a large image-level dataset, to be transferred into a different network or architectures of a different task, such as segmentation tasks. Since the weights transferred to the segmentation network can originate from a classification network, it is possible to transfer the image-level features to the segmentation network by first training the classification network on a more abundant image-level dataset. After the weights have been transferred into a segmentation network, it can be further trained on a scarce pixel-level dataset to incorporate the pixel-level features. The resulting segmentation network will have the image-level features incorporated along with the pixel-level features. Additionally, ImageNet pre-trained network weights can be transferred into the classification model before training to incorporate ImageNet features into the network, which could further improve the model performance.

Therefore, this work proposes a training strategy based on transfer learning where a network is trained in classification tasks on ImageNet and classification dataset before the weights in the network are transferred to another network for segmentation tasks. We start with a classification network pre-trained on a large-scale dataset in a different domain such as ImageNet. The network is then trained on a fish disease classification dataset in the classification task. Finally, the network weights are transferred to a segmentation network and trained again on a fish EUS segmentation dataset in the segmentation task. The main contributions made in this work are as follows:

1) A proposed training strategy that can be applied to existing semantic segmentation models for improving fish disease detection. The proposed training strategy allows semantic segmentation models to be trained with both segmentation data and classification data without the need of a complicated architecture.

2) We demonstrated and compare the effectiveness of our proposed training strategy, which uses both ImageNet and fish disease classification dataset, with alternative training strategies, which only uses either ImageNet alone or fish disease classification dataset alone.

The rest of the work is structured as follows. In Section 2, various related works for fish disease detection and transfer learning are presented. The proposed training strategy and network architectures used on this work are shown in Section 3 as well as the datasets used. Section 4 presents the comparison and results of various training strategies including our proposed training strategy. We discuss our findings based on the results we presented in Section 5. Finally, we conclude our findings in Section 6.

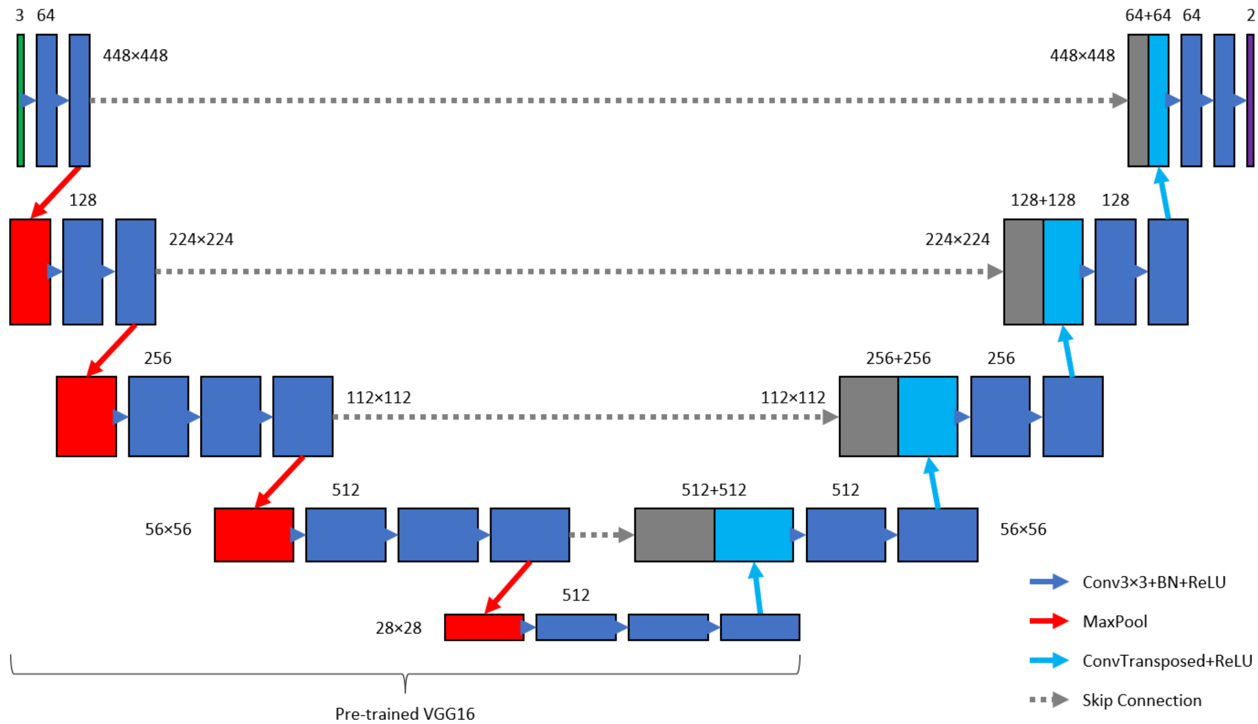

Figure 1. The U-Net+VGG architecture

2. Related Works

A. *Fish Disease Detection*

Chakravorty et al. [9] was one of the earliest work that suggested the use of semantic segmentation to detect fish disease from fish images. The work implemented a system to segment diseased areas using PCA and k-Means clustering. The results indicate that the implemented algorithm can work with above 90% accuracy. The authors suggested that more sophisticated approaches such as neural networks or SVM (Support Vector Machines) should be used for this problem. Continuing from this, Rachman [8] proposed and compared four different deep learning semantic segmentation models with backbone networks for detecting EUS. Although the performance was heavily affected by the very limited dataset, the U-Net with ResNet50 backbone network achieved the highest test score out of the four tested models with an mIoU (mean Intersection over Union) of 59.33%.

Ahmed et al. [3] proposed a method of classifying infected salmon fish using SVM. Features such as statistical features and GLCM (Grey-Level Co-occurrence Matrix) is extracted from preprocessed salmon fish images and then fed to the SVM classifier. The highest accuracy obtained by the SVM classifier is 94.12% with area under ROC curve of 96.71%. Similarly, Mia et al. [4] proposed an expert system based on feature extraction of statistical feature and GLCM to detect fish diseases. Unlike the previous work, this work employs eight different machine learning algorithms. The Random Forest model performed the best with 88.66% accuracy and an area under ROC curve of 89.71%.

Waleed [16] proposed a system based on Raspberry Pi to detect fish disease using the deep learning approach. The images are preprocessed and segmented to obtain the diseased part of the fish, then a CNN (Convolutional Neural Network) model classifies the disease of the fish. The model AlexNet reaches 99.0446% accuracy when XYZ colorspace is used. Gupta et al. [5] proposed a modified VGG network for detecting lice and wound on salmon fish. The images are preprocessed to adjust contrast. The proposed model reached 96.7% accuracy, which is 3.89% higher than the unmodified VGG19 network trained on the same dataset. Y.P. Huang and Khabusi [6] proposes an architecture based on attention mechanism, multilayer fusion, and online sequential extreme learning machine to classify five different fish diseases. The images are preprocessed to adjust contrast and remove the image background. The proposed architecture reached 94.28% accuracy.

B. *Transfer Learning*

Pravitasari et al. [13] proposed the UNet-VGG16 architecture and applied the transfer learning method for segmentation of brain tumor. The proposed architecture

UNet-VGG16 was modified so that the encoder resembles the VGG16 network and the decoder to match the encoder layers. Imad et al. [15] proposes the use of transfer learning to semantically segment 3D objects from LiDAR data. The proposed network, which shape was inspired by the U-Net architecture, receives the pre-trained MobileNetV2 weights before being trained on pre-processed bird-eye view images from raw point clouds. Sakurai et.al. [17] proposed a two-step transfer learning for semantic segmentation of plants. The FCN-8s network was first pre-trained on ImageNet, then trained on a broader plant dataset, and lastly trained on a narrower plant dataset.

3. Materials and Methods

A. Proposed Training Strategy

The training strategy we propose is as follows. First, we obtain the pre-trained VGG16 network. This pre-trained network should be pre-trained on a large-scale dataset. As the network was previously trained on a different dataset, the number of classes on the network output will differ from the number of classes on our dataset. Therefore, we replace the very last dense layer with our own to match the number of classes in our classification dataset. The pre-trained network is then re-trained on the classification task with the classification dataset of the target domain. Re-training the network with a dataset that is similar to the target domain will introduce new features relevant with the target domain to the network.

After the VGG16 network has been re-trained on the classification dataset, we transfer the weights of the convolutional layers from the VGG16 into the encoder layers of the architecture used. The chosen architecture is then trained on a segmentation task with the segmentation dataset. The weights on both encoder and decoder layers are not frozen and allowed to change during training. Figure 2 shows the process of our training strategy.

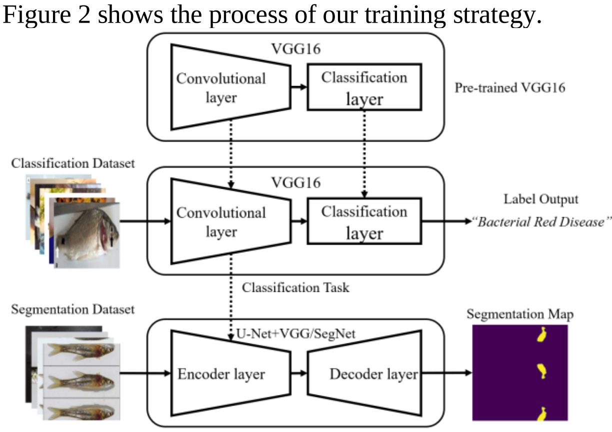

Figure 2. Our proposed training strategy.

B. Network Architectures

To test our proposed training strategy, we chose VGG16 as the classification network. We also chose SegNet and U-Net to be tested as the segmentation networks. The VGG16 network was chosen primarily due to the network being easy to adapt and transfer to the chosen segmentation networks.

The U-Net architecture is divided into two parts, the encoder layers, and the decoder layers. Except for the last encoder layer, the encoder layers are connected to the respective decoder layers using skip connections. However, the VGG16 weights cannot be direcly transferred into U-Net architecture as the architecture does not contain the VGG16 network. Therefore, we modify the encoder layers to contain the VGG16 convolutional layers. This modified architecture will be referred to as U-Net+VGG throughout this paper. Unlike the architecture proposed in [13], the decoder layers of the U-Net+VGG are not modified and remains the same. The U-Net+VGG architecture is illustrated in Figure 1.

The SegNet architecture resembles the U-Net architecture with the main difference in the skip connection, which only transfers max-pool indices into the respective decoder layers rather than feature maps [18]. Since the encoder layers of the SegNet architecture resemble the VGG16 convolutional layers by design, no modifications are necessary. We directly transfer the VGG16 weights into the encoder layers.

All architectures are adjusted to receive images with an input size of 448×448 in order to preserve the pixel sizing between datasets. Batch normalization is applied after every convolutional layer and before the Rectified Linear Unit (ReLU) layer.

C. Datasets

1) Classification Dataset

The classification dataset contains 460 images of fish obtained from Kaggle [19]. The dataset contains pictures of fish that contracted various fish diseases, as well as pictures of healthy fish. Each image in this dataset is labeled by the class it is categorized in. The dataset contains 7 classes of fish disease, including healthy fish.

We discovered that the dataset contains several duplicated images within the same classes and between different classes. Additionally, the dataset overlaps with the segmentation dataset. To prevent leaking testing data to the model, image deduplication is performed through the use of a Python library to detect duplicated images and overlapping data from another dataset. The found duplicated images are then removed, prioritizing duplicate images from the largest class. After deduplicating the images, 304 images are left and split into training data, validating data, and testing data containing 213, 47, and 44 images respectively.

To match the segmentation network, the images are simply resized and stretched to 448×448 pixels. No image pre-processing was applied.

2) Segmentation Dataset

The segmentation dataset contains 26 images of fish obtained from Roboflow [20]. The dataset contains pictures of fish that contracted EUS. Each image is labeled with a segmentation mask. The dataset contains only EUS class.

Similar to the classification dataset, we also deduplicate the image using the same method, resulting in 25 images left in the dataset. The dataset is then split into training data, validating data, and testing data containing 15, 5, and 5 images respectively.

The segmentation dataset contains images with varying sizes. To ensure that the next image preprocessing method can work, the images are first resized to have the same width and varying height. We chose the width to be fixed at the median width of the segmentation dataset of 374 pixels.

Next, we employ an image preprocessing method to extend the height of the image using mirroring and padding. The method is as follows. For training image data, the image is extended by mirroring the original image. For validation and testing image data, the image is extended by padding with black pixels. The segmentation masks are also extended or mirrored along with the images. To further illustrate the method used, an example is given in Figure 3.

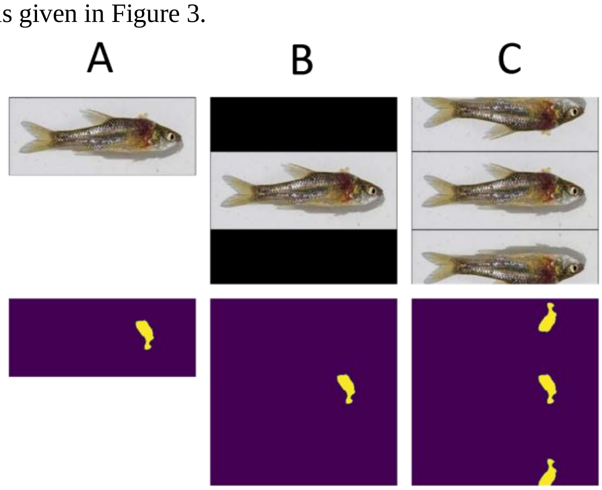

Figure 3. An example of the image processing method used for squaring images. A: Original image and mask. B: Padding with black pixels. C: Mirroring the original image

The resulting image size of this dataset is 374×374 pixels. To fit the image with the model input size, we further resize and stretch the images to 448×448.

D. *Data Augmentation*

During the training process in both classification and segmentation tasks, image augmentations are applied to images right before being fed into the network. The image augmentations are only applied to the training data splits. The validation and testing data splits do not receive any image augmentation whatsoever. Image augmentations are applied to all training strategies equally. The image augmentations used for this work are as follows, which are applied in order of appearance:

1) Affine transformation. The image is translated up to 20% up/down and left/right by a random value, and scaled up/down by a random value between 90% and 130%.

2) Horizontal and Vertical flipping. The image is flipped on the horizontal and vertical axis, with each axis flips applied independently.

3) Color shifting. The brightness, contrast, and saturation of the image is shifted by a random value between -15% and 15%.

4) Elastic deformations. The image is deformed by warping the image elastically [21].

5) Random rotation. The image is rotated to a random value between -90° to 90°.

E. *Evaluation*

To help describe the training strategies, we first introduce two VGG16 networks that will be used for transfer learning. The VGG16 networks are:

- VGG16-C: VGG16 network initialized with random weights and trained on classification dataset.
- VGG16-IC: VGG16 network initialized with ImageNet-1K VGG16 weights and trained on classification dataset.

In order to compare the performance of our training strategy, we train several models with different training strategies.

- U-Net: U-Net architecture with encoder layers initialized with random weights.
- U-Net+VGG: U-Net+VGG architecture with encoder layers initialized with random weights.
- U-Net+VGG-C: U-Net+VGG architecture with encoder layers initialized with VGG16-C weights.
- U-Net+VGG-I: U-Net+VGG architecture with encoder layers initialized with ImageNet-1K VGG16 weights.
- U-Net+VGG-IC: U-Net+VGG architecture with encoder layers initialized with VGG16-IC weights. This is our proposed training strategy.
- SegNet: SegNet architecture with encoder layers initialized with random weights.

- SegNet-C: SegNet architecture with encoder layers initialized with VGG16-C weights.
- SegNet-I: SegNet architecture with encoder layers initialized with with ImageNet-1K VGG16 weights.
- SegNet-IC: SegNet architecture with encoder layers initialized with VGG16-IC weights. This is our proposed training strategy.

To measure the improvement between training strategies, we use mean Intersection over Union (mIoU) as a metric. The metric mIoU is widely used for evaluating the performance of semantic segmentation models. The mIoU is calculated by taking the mean of all the IoU value across all the classes (excluding the background class) and across all the images in the data split. Note that since there is only one non-background class in our dataset, we simply compute the mean across all the images. The IoU of a class of an image is computed by the following equation,

$$IoU = \frac{P \cap G}{P \cup G} \tag{1}$$

Where P is the predicted pixels in the image that belong to the class, and G is the ground truth pixels in the image that belong to the class.

4. Results

A. Experimental Setup

The training setup was implemented using PyTorch 2.1.1 library and written in Python programming language. The pre-trained VGG16 model was obtained from PyTorch Hub. The data augmentation pipeline was implemented with TorchVision 0.16 that comes with PyTorch itself. The models are trained with GPU hardware acceleration enabled in a computer with 10GB of Video RAM.

All weights in each training strategy, except the weights that have been initialized as per the training strategy, are initialized using Kaiming initialization with normal distribution [22]. Each training strategy was repeated 8 times with the same data split distributions, producing 8 models per training strategy. Out of the 8 models produced, the model with the best validation mIoU score is chosen to represent the result for the training strategy.

B. Classification Task

The VGG16-C and VGG16-IC network was trained on the classification dataset with SGD optimizer. The SGD uses a fixed learning rate of 0.0001 and momentum of 0.9. The batch size used for training VGG16 network is 8. Weighted cross-entropy loss was used for calculating losses, with each loss weights for each class set to the mean class output frequency divided by individual class output frequency. This is done to address the class imbalance over-representing the healthy fish class in the classification dataset. The networks were trained until the validation score shows no improvement for the last 100 epochs, with the exception of VGG16-IC, which is only trained until validation score shows no improvement for the last 10 epochs. TABLE 1 shows the performance results of VGG16 networks on the classification dataset.

TABLE 1. Performance results of VGG16 on the classification task

Training Strategy	**Epoch**	**Accuracy (%)**		
		Train	**Valid**	**Test**
VGG16-C	159	75.59	34.04	36.36
VGG16-IC	51	87.79	63.83	63.64

C. Segmentation Task

The U-Net and SegNet training strategies also used SGD optimizer during training. For U-Net, the fixed learning rate is 0.0001 and momentum of 0.99. For SegNet, the fixed learning rate is 0.1 and momentum is 0.9. Batch sizes used for training both U-Net and SegNet are 4 and 5, respectively. Dice loss was used for calculating losses. Both U-Net and SegNet training strategies are trained until the validation score shows no improvement for the last 50 epochs.

TABLE 2 shows the performance results for each training strategy employed at the end of training, with the performance calculated at the epoch when the model achieves its lowest validation score. The number of epoch when the model achived its lowest validation mIoU score is shown in the table.

TABLE 2. Performance results of all the training strategies tested on the segmentation task

Training Strategy	**Epoch**	**Mean IoU (%)**		
		Train	**Valid**	**Test**
U-Net	565	47.36	47.12	50.03
U-Net+VGG	449	46.10	47.24	52.17
U-Net+VGG-C	283	50.03	48.47	49.26
U-Net+VGG-I	572	66.62	55.41	53.37
U-Net+VGG-IC	334	**68.83**	**56.53**	**53.90**
SegNet	191	61.18	52.18	49.55
SegNet-C	113	55.36	57.27	51.69
SegNet-I	277	**78.72**	66.07	57.54
SegNet-IC	148	75.22	**66.53**	**63.46**

Comparing U-Net+VGG to U-Net+VGG-C, the training and validation mIoU score for U-Net+VGG-C has improved but the testing mIoU score has lowered. However, in the SegNet and SegNet-C case, the validation and testing mIoU score in SegNet-C was improved but the training mIoU score was lowered.

In the U-Net+VGG and U-Net+VGG-I case, there is a strong improvement in the mIoU score in all data splits for U-Net+VGG-I. The same improvement also happens for SegNet-I in the SegNet to SegNet-I case. This confirms that ImageNet transfer learning improves model performance.

However, comparing U-Net+VGG-I to U-Net+VGG-IC, there is a slight improvement in the mIoU score in all data splits for U-Net+VGG-IC. In the SegNet-I and SegNet-IC case, there is a strong improvement in the testing mIoU score and a slight improvement in the validation mIoU score, but only a slight decrease in the training mIoU score. This shows that the proposed training strategy gives better performance than the alternative training strategies.

Figure 4 and Figure 5 shows the training loss curve for the U-Net and SegNet training strategies, respectively. The training loss curve shows how the loss of the model converges. Note that some models have shorter curves as these models were trained shorter due to their validation score showing no more improvement.

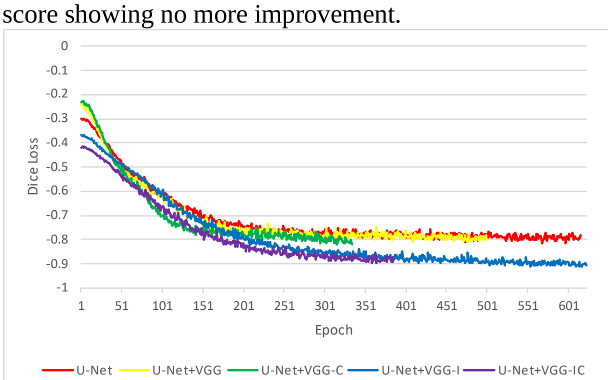

Figure 4. Training loss curves of the U-Net training strategies

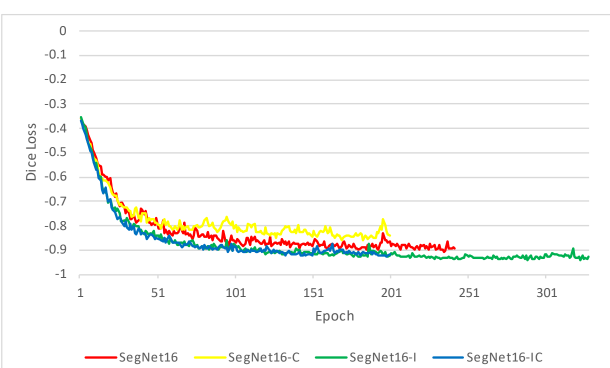

Figure 5. Training loss curves of the SegNet training strategies

From the training loss curves, it can be seen that the model U-Net+VGG-IC converges the fastest among the other U-Net training strategies. Although SegNet-IC has a slight decrease to the training mIoU score compared to SegNet-I, there is not a visible difference to the loss convergence between the two training strategies.

Among all the training strategies we tested, the SegNet-IC performed the best with a validation and testing mIoU score of 66.53% and 63.46%, respectively. This suggests that SegNet works the best in segmenting fish EUS disease.

To further validate the results, we also show two difficult segmentation cases, image A and B, in Figure 6. The areas marked in yellow show the EUS-diseased area, while the background class is marked in purple. As the segmentation masks show, the SegNet-IC produced the best segmentation masks for both images. U-Net+VGG-IC produced a slightly worse segmentation mask than U-Net+VGG-I in image A but still performs well in image B.

5. Discussion

We trained the classification network on a fish disease classification dataset in order to acquire the weights for testing different training strategies. The acquired weights are then transferred to the respective traning strategies. We then train and test different training strategies including our proposed strategy in order to compare the performance. We showed the training loss curves for different strategies to see how the model loss converges during training. We showed the mIoU scores of each training strategy in three different data splits.

The U-Net architecture is modified to allow the transfer of VGG16 weights. The modified U-Net architecture achieves a roughly similar performance compared to the unmodified U-Net architecture. This suggests that the model was not underfitting the data and only receives little benefit from the increased parameter count in the modified U-Net architecture.

The training strategy that we propose, which uses both ImageNet and classification dataset, are also compared with alternative strategies which only uses either ImageNet or classification dataset. This was done to confirm our beliefs that the inclusion of both ImageNet and classification dataset improves the performance of the network more than either ImageNet or classification dataset alone.

We tested the training strategies on both U-Net and SegNet to show that the training strategy that we propose can be applied to various conventional semantic segmentation architectures. As both SegNet and U-Net produced the best performance on the respective architectures when our proposed training strategy was applied, this confirms that the proposed training strategy can work on different segmentation architectures, without the need of a complicated Multi-task architecture.

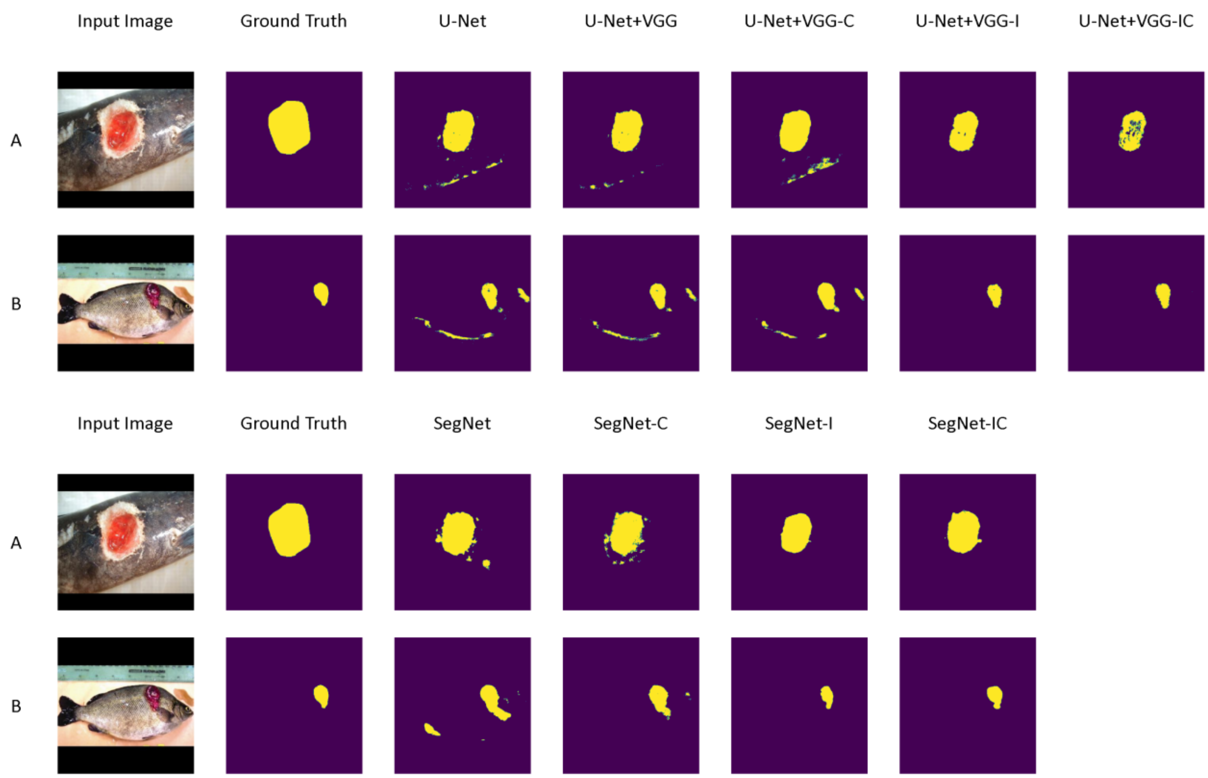

Figure 6. Comparison of segmentation maps produced by each training strategies

We have also looked at and evaluated other loss functions for improving semantic segmentation performance, including Cross-entropy and Focal loss. However, we did not see any significant improvements to the model performance with other loss functions.

We noticed that the results for each training strategy were influenced by the performance of the network its weights were transferred from. During the training, we further noticed that the performance of the VGG16 network differed as we resampled the classification dataset. This suggests that ensuring a good data split distribution is important as it will later affect the model performance in segmentation task.

This work is primarily focused on fish disease detection and uses datasets that contain images of fish diseases, especially EUS,. However, we believe that the proposed training strategy could also be applied for other datasets of different target domains.

6. Conclusion and Future work

We have proposed a training strategy based on transfer learning to improve the semantic fish EUS segmentation. The proposed training strategy involves training the classification network on both ImageNet and classification dataset before being transferred into the segmentation architecture. To test the proposed training strategy, we chose VGG16 as the classification network and both U-Net and SegNet as the segmentation architectures. Two datasets were used for the experiment, one for classification task, and another for segmentation task. In order to allow the VGG16 weights to be transferred into U-Net, we modified the architecture to include a VGG16 encoder layer. To compare and show the effectiveness of the proposed training strategy, we compare the proposed training strategy with alternative training strategies.

The results are then compared with different training strategies. Both architectures trained with our proposed training strategy performed better than the alternative training strategies we have shown, which shows the effectiveness of our proposed training strategy. The best performing training strategy is SegNet-IC with validation and testing mIoU score of 66.53% and 63.46%, respectively.

In the future, more recent semantic segmentation models can be researched to improve the segmentation quality of fish diseases. We will also look into working with object detection networks and region-level labeled datasets that are less difficult to annotate than pixel-level labeled datasets.

REFERENCES

[1] BPS, "Produksi Perikanan Budidaya Menurut Komoditas Utama," 2022. Accessed: Jun. 05, 2023. [Online]. Available: https://www.bps.go.id/indicator/56/1513/1/produksi-perikanan-budidaya-menurut-komoditas-utama.html.

[2] L. O. Haslan, *Buku Saku Hama dan Penyakit Ikan*. 2017.

[3] M. S. Ahmed, T. T. Aurpa, and M. A. K. Azad, "Fish Disease Detection Using Image Based Machine Learning Technique in Aquaculture," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5170–5182, Sep. 2022, doi: 10.1016/j.jksuci.2021.05.003.

[4] M. J. Mia, R. Bin Mahmud, M. S. Sadad, H. Al Asad, and R. Hossain, "An in-depth automated approach for fish disease recognition," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7174–7183, Oct. 2022, doi: 10.1016/j.jksuci.2022.02.023.

[5] A. Gupta, E. Bringsdal, K. M. Knausgård, and M. Goodwin, "Accurate Wound and Lice Detection in Atlantic Salmon Fish Using a Convolutional Neural Network," *Fishes*, vol. 7, no. 6, p. 345, Nov. 2022, doi: 10.3390/fishes7060345.

[6] Y.-P. Huang and S. P. Khabusi, "A CNN-OSELM Multi-Layer Fusion Network With Attention Mechanism for Fish Disease Recognition in Aquaculture," *IEEE Access*, vol. 11, no. June, pp. 58729–58744, 2023, doi: 10.1109/ACCESS.2023.3280540.

[7] N. Hasan, S. Ibrahim, and A. Azlan, "Fish diseases detection using convolutional neural network (CNN)," *Int. J. Nonlinear Anal. Appl*, vol. 13, no. November 2021, pp. 2008–6822, 2022, doi: 10.22075/ijnaa.2022.5839.

[8] F. Rachman, M. Akbar, and E. Putera, "FISH DISEASE DETECTION OF EPIZOOTIC ULCERATIVE SYNDROME USING DEEP LEARNING IMAGE PROCESSING TECHNIQUE," in *international Conference on Fisheries and Aquaculture*, Jan. 2023, vol. 8, no. 1, pp. 23–34, doi: 10.17501/23861282.2023.8102.

[9] H. Chakravorty and R. Paul, "Image Processing Technique To Detect Fish Disease," *Int. J. Comput. Sci. Secur.*, vol. 9, no. 2, pp. 121–131, 2015.

[10] A. Amyar, R. Modzelewski, H. Li, and S. Ruan, "Multi-task deep learning based CT imaging analysis for COVID-19 pneumonia: Classification and segmentation," *Comput. Biol. Med.*, vol. 126, no. October, p. 104037, Nov. 2020, doi: 10.1016/j.compbiomed.2020.104037.

[11] S. Hong, J. Oh, H. Lee, and B. Han, "Learning Transferrable Knowledge for Semantic Segmentation with Deep Convolutional Neural Network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, vol. 2016-Decem, pp. 3204–3212, doi: 10.1109/CVPR.2016.349.

[12] Y. Sun *et al.*, "Weakly Supervised Instance Segmentation Based on Two-Stage Transfer Learning," *IEEE Access*, vol. 8, pp. 24135–24144, 2020, doi: 10.1109/ACCESS.2020.2969480.

[13] A. A. Pravitasari *et al.*, "UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 18, no. 3, p. 1310, Jun. 2020, doi: 10.12928/telkomnika.v18i3.14753.

[14] S. Baressi Šegota *et al.*, "Semantic Segmentation of Urinary Bladder Cancer Masses from CT Images: A Transfer Learning Approach," *Biology (Basel).*, vol. 10, no. 11, p. 1134, Nov. 2021, doi: 10.3390/biology10111134.

[15] M. Imad, O. Doukhi, and D.-J. Lee, "Transfer Learning Based Semantic Segmentation for 3D Object Detection from Point Cloud," *Sensors*, vol. 21, no. 12, p. 3964, Jun. 2021, doi: 10.3390/s21123964.

[16] A. Waleed, H. Medhat, M. Esmail, K. Osama, R. Samy, and T. M. Ghanim, "Automatic Recognition of Fish Diseases in Fish Farms," in *2019 14th International Conference on Computer Engineering and Systems (ICCES)*, Dec. 2019, pp. 201–206, doi: 10.1109/ICCES48960.2019.9068141.

[17] S. Sakurai, H. Uchiyama, A. Shimada, D. Arita, and R. Taniguchi, "Two-step Transfer Learning for Semantic Plant Segmentation," in *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, 2018, vol. 2018-Janua, no. Icpram, pp. 332–339, doi: 10.5220/0006576303320339.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Nov. 2015, [Online]. Available: https://arxiv.org/pdf/1511.00561.pdf.

[19] S. Biswas, "Freshwater Fish Disease Aquaculture in south asia Dataset." 2022, [Online]. Available: https://www.kaggle.com/datasets/subirbiswas19/freshwater-fish-disease-aquaculture-in-south-asia.

[20] testing, "Fish Disease 2 Dataset," *Roboflow Universe*. Roboflow, Aug. 2022, [Online]. Available: https://universe.roboflow.com/testing-dgymd/fish-disease-2.

[21] W. Weng and X. Zhu, "INet: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, vol. 9, pp. 16591–16603, May 2021, doi: 10.1109/ACCESS.2021.3053408.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, vol. 2015 Inter, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

Anbang received his bachelor's degree in Computer Science from Pelita Harapan University. He is currently pursuing his Master's Degree from Bina Nusantara University. His current research interests include deep learning, image segmentation, and weakly-supervised learning.

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Head of Department of Master of Computer Science, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R – A*STAR, Singapore. His research interests include computer vision, deep learning, face recognition, appearance-based object recognition, gamification of learning, and indoor positioning system.