Mackerel Fat Content Estimation using RGB and Depth Images

SHUYA SANO¹, TOMO MIYAZAKI¹, (Member, IEEE), YOSHIHIRO SUGAYA¹, (Member, IEEE), NAOHIRO SEKIGUCHI², and SHINICHIRO OMACHI¹, (Senior Member, IEEE)

¹Graduate School of Engineering, Tohoku University, Japan, 9808579
²Tohto C-tech Corporation, Japan, 9830047

Corresponding author: Tomo Miyazaki (https://tomomiyazaki.github.io).

This work was partially supported by JSPS KAKENHI Grant Numbers 20H04201, 19K12033, and 19K11848.

ABSTRACT We propose a method for estimating the fat content of mackerels from their images. The market value of fish varies greatly depending on the fat content. For example, mackerels with high-fat content are of high priority for business transactions in Japanese fisheries. The fat content is commonly measured manually with special equipment using the near-infrared spectroscopy, which increases costs and reduces productivity. It is ideal to estimate the fat content automatically using inexpensive equipment such as ordinary cameras. However, fat content estimation from fish images is a challenging task because the difference in fat content appears only as a slight difference in their appearance. To tackle this problem, we propose to use not only RGB images but also depth images to utilize shape information as well as the textures. To detect subtle differences in texture and shape, we propose a convolutional neural network that extracts and concatenates features from part images, such as the head, body, and tail of a mackerel image. Color-texture and three-dimensional shape features extracted from RGB and depth images, respectively, are combined to estimate the fat content. Experimental results show that the proposed method estimated fat content with 2.25 points at mean absolute error.

INDEX TERMS Fish image analysis, fat content estimation, neural network, RGB image, depth image, fishery industry

I. INTRODUCTION

The fat content of fish is one of the important factors that determines its market value, and it is important to accurately estimate the fat content. The fat content is now commonly measured manually with special equipment using the near-infrared spectroscopy. However, this method requires additional labor power and special equipment, which increases costs and reduces productivity. A method for automatically estimating the fat content in a non-destructive manner with inexpensive equipment such as ordinary cameras is required. On the other hand, some experts identify high-fat content fish without cutting and checking the cross-section of the fish. They can estimate fat content based on subtle differences in appearance from years of experience. This means that it is not impossible to estimate the fat content from fish images.

In this study, we propose a method for estimating the fat content of fish from their images, targeting mackerels. Mackerel is a popular fish caught in Japan, and its market value varies greatly depending on the fat content. The fat content estimation from fish images is a challenging task because the difference in fat content appears only as a slight difference in their appearance. To tackle this problem, we propose to use not only RGB images but also depth images to utilize shape information as well as the textures. It is known that the body shape of fish changes and the pattern on the body surface also changes by accumulating fat in their body. The shape features from the depth image can improve estimation performance. We also propose to extract features from the head, body, and tail of a mackerel image to detect subtle differences in texture and shape. This feature extraction strategy enables us to focus on local features of texture and shape. Color-texture and three-dimensional shape features extracted from RGB and depth images, respectively, are combined to estimate the fat content. To this end, we propose neural networks to merge these features. Consequently, we can acquire features suitable for fat content estimation.

To show the effectiveness of the proposed method, ex-
periments were conducted to evaluate the accuracy of the estimated fat content. It was shown that the proposed method achieved an absolute error of approximate 2.2 % compared to the values measured by the NIR spectroscopy sensor.

II. RELATED WORKS

A. FISH FAT CONTENT ESTIMATION

The near-infrared (NIR) spectroscopy is a general measurement method for fish fat content in a non-destructive inspection. Almendingen et al. used a NIR spectroscopy to determine fat content in homogenized diets [1]. The determined fat content was accurate, and its processing time was more rapid than the traditional technique [2]. Zhang et al. applied linear regression to raw data measured by the NIR spectroscopy to determine moisture, protein, and fat content in fish meals [3]. The reported processing time was less than 3 minutes.

Although the NIR spectroscopy is a standard measurement method, it requires a special equipment and additional labor power which increases costs and reduces productivity. It is ideal to estimate the fat content automatically using inexpensive equipment such as ordinary cameras.

B. FISH CLASSIFICATION USING MACHINE LEARNING

It is essential to capture visual patterns in fish images for fish classification. Thus, there are many attempts to exploit machine learning techniques. Mohamed et al. classified fish images into tilapia and other species [4]. This method used both image processing and machine learning techniques simultaneously. Khotimah et al. developed an algorithm for classifying images into three species: bigeye tuna, yellowfin tuna, and skipjack tuna [5]. They used the gray level co-occurrence matrix (GLCC) [6] to extract features from the texture of fish images, and a decision tree was used for classification. Kitsato et al. used SVM as a classifier to classify chub mackerel and blue mackerel [7]. They used texture and shape features. The shape feature was measured the dorsal fin’s length from the first to the ninth spine in images. Hasija et al. developed a method for fish species classification using subspace-based graph matching [8]. Chuang et al. classified seven fish species using head size, eye texture, and the tail ratio to the whole body [9]. Hsiao et al. had developed a fish species classification by matching [10].

Convolutional neural network, CNN, becomes a common approach for fish classification [11]–[15]. Siddiqui et al. showed that CNN was effective for fish species classification in an underwater environment, including noise and blur [11]. Ge et al. extracted features using CNN and used Gaussian mixture models, GMMs, for fine classification of fish images [12]. Nagaoka et al. used CNN to recognize chub and blue mackerels [16]. Also, there are methods based on CNN for classification and detection of other animals [17]–[21]. The results of the methods based on CNN are faster than other methods. Besides, they are robust to noise.

C. REGRESSION ESTIMATION METHODS

Although not concerning fishes, many methods used regression techniques. For example, there are gender and age estimation from face images [22], friction coefficient and hardness estimation of an object [23], and ripeness estimation of a fruit [24]. Most methods used a pre-trained CNN model and trained only layers added to the model [22]–[25]. On the other hand, some methods trained a CNN as a feature extractor and perform regression estimation using decision trees [26], [27]. The methods mentioned used CNNs for feature extraction. The main difference is training CNN from scratch or use of a pre-trained model. Using a pre-trained model, we can train CNN on various datasets since the number of training parameters is limited. Whereas, in the case of training CNNs from scratch, training is not easy because of the large number of parameters.

There is a regression approach by classifying to discrete ranges [22]. The performance of this approach is comparable to the general regression approach. Therefore, we adopt the general regression approach.

III. IMAGE CAPTURE SYSTEM

We illustrate the image capture system in Fig. 1. We capture mackerel images moving on a conveyor belt. The input slope aligns mackerels. We suppose mackerels are isolated, and they should be left direction when they are put on the conveyor belt. We capture RGB image and depth values using an RGB camera and a Time-of-Flight camera (ToF camera), respectively. The distance from the cameras to the conveyor is 480 mm. Illuminance is 8000 lx at the center of the conveyor. The RGB and ToF cameras are Lucid Vision Triton TR050S-CC and Helios HLS003S-001, respectively. The focal length of the RGB camera is 8 mm. The precision of ToF camera is 0.69 mm. Considering the average thickness 76.5 mm in Table 1, the precision 0.69 mm is sufficient. We performed calibration to obtain pixel-level correspondences between RGB and ToF cameras. We create a depth map by retrieving the depth values corresponding to the RGB image. Consequently, all pixels are matched between the depth map and the RGB image. Also, their sizes are the same, 1024-pixels square. Table 1 shows statistics of four features, length, width, thickness, and weight. Note that we do not use these four features for fat content estimation in this study.

As shown in Fig. 2, we created a pseudo-color image from depth data captured by the ToF camera using the following procedure. We converted the depth data into a gray scale image by

\[
g = 1 - \frac{d - D_{\text{max}}}{D_{\text{max}} - D_{\text{min}}},
\]

where \(d\) is the gray scale value, \(D_{\text{max}}\) is the maximum depth, and \(D_{\text{min}}\) is the minimum depth. Table 1. The statistics of 32 mackerels caught in December 2019

|       | Length | Width  | Thickness | Weight |
|-------|--------|--------|-----------|--------|
| Ave   | 777.6 mm | 54.0 mm | 76.5 mm   | 681.5 g |
| Min   | 310 mm  | 39 mm  | 60 mm     | 305 g   |
| Max   | 423 mm  | 65 mm  | 89 mm     | 990 g   |
where \( g \) represents gray scale value, \( d \) is measured depth. \( D_{\text{max}} \) and \( D_{\text{min}} \) are maximum and minimum depth values, respectively. In this study, we set \( D_{\text{max}} \) as 535 (mm), \( D_{\text{min}} \) as 420 (mm). The thick regions in the converted gray scale image will be bright. Then, we assigned zero to missing values in the depth data. Subsequently, we apply a median filter with \( 3 \times 3 \) kernel size to the gray image to remove noise in the gray image since the conveyor belt absorbs infrared radiation from the ToF camera. Finally, we converted the gray image into a pseudo-color image using a jet color map. Hereafter, the depth image denotes the pseudo-color image, which is input to the neural networks. Since it is difficult to extract the shape information from the RGB image, the depth image complements the RGB image. We resize the original image size, 1024 pixels square, to 224 pixels square since we adopt the VGG16 model [28] in this study.

IV. FAT CONTENT ESTIMATION
We show the overview of the proposed fat content estimation algorithm in Fig. 3. The proposed method is composed of three modules. The first module estimates the mackerel region in the input image. The second module generates global and local images of the mackerel using the estimated region. Finally, the third module estimates the fat content using the global and the local images.

A. MACKEREL REGION ESTIMATION
The only object in the input image should be a mackerel. However, some parts of the image capture system, such as the conveyor belt, exists in the image. To crop only the mackerel image as accurately as possible, we estimate the mackerel region accurately in the image.

In this study, we utilize the VGG16 model [28] for the region estimation, which is trained on ImageNet [29] to capture 1000 classes of objects in various situations. A wide range of applications uses the VGG 16 model since it can extract features from objects with various shapes and colors. Moreover, the model can fit various datasets, even a small one. The model contains 13 convolutional layers, and each convolutional layer extracts high dimensional features by increasing the number of channels while decreasing the image size. Specifically, we use the first layer’s feature map to maintain the mackerel’s resolution. This feature map activates the locations of the object. As shown in Fig. 4 (a), the feature map focuses strongly on mackerel. Therefore, we can accurately estimate the mackerel region in the image using the feature map.

We describe details of the algorithm for mackerel region estimation. We experimentally set the thresholds and other parameters.

1) Feature map extraction. We extract a feature map from the RGB image using the first layer of the VGG16 model. Subsequently, we reduce the number of channels of the feature map 64 to 1 by taking max in the channels. Fig. 4 (a) shows an example.
Sano et al.: Mackerel Fat Content Estimation using RGB and Depth Images

![Image of mackerel region estimation process]

**FIGURE 3.** The overview of the proposed method

**FIGURE 4.** The output examples of the mackerel region estimation

2) Binarization. We binarize the feature map using a threshold 9. The values in foreground and background become 1 and 0, respectively. An example is in Fig. 4 (b).

3) Noise suppression. We suppress the outer area of the predefined region by making the values to 0. The predefined region is a box with left-top (10, 30), width 90, and height 50. An example is shown in Fig. 4 (c).

4) Left and right position search. As shown in Fig. 4 (d), we count foreground pixels vertically. The left position is the first non-zero pixel from the most left. Likewise, the right position is the last pixel.

5) Upper and lower position search. As shown in Fig. 4 (e), we count foreground pixels horizontally. The upper position is the first non-zero pixel from the most top. Likewise, the lower position is the last pixel.

**B. GLOBAL AND LOCAL MACKEREL IMAGE GENERATION**

We generate global and local images of the mackerel. As shown in Fig. 5, we define the global image as the whole mackerel part. The local images are the head, body, and tail of the mackerel.

We produce the global image by cropping the estimated region. We resize the cropped image to 224-pixel square by adding margins. The VGG16 model is trained on 224-pixel square images. To take advantage of the performance of the trained VGG16, we adopt the same image size. The global image contains texture and shape features of the mackerel. However, local information may lose due to the low resolution caused by resizing.

We create local images of the mackerel’s parts, such as the head, body, and tail, to maintain the resolution of the details. We crop $h$-pixel squares from the estimated region, where $h$ is the height of the estimated region. Precisely, we extract body square so that its center is the first position that maximizes the vertical count of foreground pixels as shown in Fig. 4 (d). The head and tail squares are adjacent to the body square. Finally, we resize the $h$-pixel squares to 224-pixel squares. We can extract features from the local images that complement features from the global image. Also, we produce depth global and local images using the locations of the RGB local images.

**C. FAT CONTENT ESTIMATION NETWORK**

We use neural networks to estimate fat content from the global and local images of RGB and Depth. We show the configuration of the network in Fig. 6. We apply the VGG16 model without the fully connected layers to all the input images to extract feature maps, resulting in feature matrices $\mathbb{R}^{7\times 7\times 512}$. Every fully connected layer has one hidden layer.

We use the VGG16 model as a feature extractor to extract texture features instead of using hand-crafted texture features, such as LTP [30] and LQP [31]. Then, we gradually merge features and produce the final feature to estimate fat content. Specifically, we use flatten layer to transform the feature matrices into one-dimensional vector $\mathbb{R}^{1\times 25088}$. We apply six fully connected layers that produce feature matrices $\mathbb{R}^{1\times 1\times 24}$
and $\mathbb{R}^{1 \times 1 \times 32}$ for global and local images, respectively. Then, we merge and produce features using concatenation and fully connected layers. The VGG16 is the configuration D model trained on ImageNet.

**Importance for Samples**

The fat content is biased in the dataset. There are only few samples of extremely low- and high-fat content. They are the minority. A simple training tends to learn the fat contents that are majorities in the dataset. However, it is difficult to learn the fat contents of minorities, such as extremely low- and high-fat content. To solve this problem, we assign importance to each sample to learn all samples’ fat content. Specifically, we assign high importance to minority samples and train them with a high learning rate to encourage the networks to learn the minorities. On the other hand, the majority samples are assigned with low importance to suppress learning them. Consequently, we can mitigate the bias of the dataset and prevent overtraining.

We calculate importance $W_i$ for sample $i$ by (2). Specifically, we normalized the fat content to $[0, 1.0]$ and created a histogram of fat content from the dataset$^1$. Then, we divide the bin value $m_i$ by the maximum value $m_{\text{max}}$ of all bins.

$$W_i = \left( \frac{m_i}{m_{\text{max}}} \right)^{-1} = \frac{m_{\text{max}}}{m_i}$$

(2)

**Normalization**

To facilitate training, we normalize the fat content to $[0, 1]$. We define the normalization used in this study as Eq. (3). We obtain the maximum $E_{\text{max}}$ and minimum $E_{\text{min}}$ from training data when we normalize fat content. We use the maximum value $E_{\text{max}}$ in the training data to normalize RGB and depth images as defined in Eq. (4).

$$y_{\text{norm}} = \frac{y - E_{\text{min}}}{E_{\text{max}} - E_{\text{min}}}$$  

(3)

$$x_{\text{norm}} = \frac{x}{E_{\text{max}}}$$

(4)

**IMPLEMENTATION DETAILS**

As we described, the feature extractor is the VGG16 model trained on ImageNet. We freeze the parameters of the VGG16 model. Thus, we use the fixed parameters to extract features during training and test. The number of parameters in the entire estimation network is 21,407,585, while, the number of training parameters is 13,772,321. We describe the hyperparameters used in this study below. The batch size is 32, the number of epochs is 100, the loss function is mean squared error (MSE), the optimization algorithm is SGD (stochastic gradient descent optimizer), the learning rate is $2.5 \times 10^{-3}$, and the momentum is 0.9. Also, we used 20% of the training data as validation data. We determine the best model using the validation data over the epochs. Tensorflow 1.9.0, a framework for machine learning, was used for the implementation, and the official Tensorflow Docker image file was used to build the environment. We use a GeForce GTX1080Ti GPU.

**V. EXPERIMENTS**

We show the dataset’s specification in the experiments in Table 2. The number of mackerels used in the dataset was 287, with a minimum and maximum fat contents of 10.53% and 33.17%, respectively. To ensure that the training and test datasets are independent, we used the images taken in October 2019 for training data, and the test data are taken in February 2020.

We measured ground truth of mackerel fat content using a NIR spectroscopy sensor, NIR-GUN$^3$. We put the NIR spectroscopy sensor to a position where a few millimeters from the anus to the tail. The abdomen of a mackerel accumulates fats in a short time. On the other hand, the anus needs a long time to accumulate fat since there are no organs on the tail side of the anus. Therefore, the measurement of the anus is more stable than that of the abdomen.

We used four evaluation criteria: mean absolute error (MAE), root mean square error (RMSE), R2-score, and correlation coefficient. We describe each criterion using ground

---

$^1$ We experimentally set bin width to 0.01

$^2$tensorflow/tensorflow:1.9.0-gpu-py3

$^3$FQA-NIR GUN (Food Quality Analyzer) by FANTEC Co., Ltd.
FIGURE 6. Fat content estimation network

truth \( y \), estimated value \( y' \), and the mean of all ground truth \( \bar{y} \). The MAE is defined as (5). MAE averages error between the \( y \) and \( y' \), where the smaller the error, the more accurate the estimation. RMSE is defined as (6). RMSE considers large errors as more important. Compared to the MAE, RMSE is sensitive to outliers with large gaps between ground truth and estimated values. R2-score in (7) is ranging from zero and one. The closer to one, the performance is better. The correlation coefficient evaluates the correlation between the estimated values and the ground truth. The correlation coefficient is equal to the root of R2-score.

\[
MAE = \frac{1}{n} \sum |y_i - y'_i| \quad (5)
\]

\[
RMSE = \sqrt{\frac{1}{n} \sum (y_i - y'_i)^2} \quad (6)
\]

\[
R^2 = 1 - \frac{\sum_i (y_i - y'_i)^2}{\sum_j (y_j - \bar{y})^2} \quad (7)
\]

A. RESULT ON FAT CONTENT ESTIMATION

We compared the proposed model to typical regression and deep learning models. We used VGG16 and VGG19 as the deep learning models, which are regarded as baselines. We replace the existing fully connected layers of the VGG16 and VGG19 with a new fully connected layer. The input
for the baselines is an RGB image obtained by the image capture system, which is shown in Fig. 2. We trained only the replaced fully connected layer.

The typical regression models are Support Vector Regression (SVR) [32], random forest (RF), and gradient boosting (GB) [33], [34]. We extracted feature vectors from 224-pixel square images using Histogram of Oriented Gradients (HOG) descriptor [35]. The dimension of a HOG descriptor is 54756. We used radial basis function kernel in SVR. The random forest and gradient boosting models used ten weak classifiers.

We illustrated the evaluation results using the proposed method in Table 3. The baselines and the proposed method obtained more than 0.7 points at correlation coefficient and less than 3.0 points at MAE. Furthermore, the proposed method outperformed the baselines in all the evaluation criteria. In particular, the RMSE of the proposed method was less than 3, whereas the RMSE of the baselines was more than 3.2. The results indicate the effectiveness of the proposed method.

We showed a scatter plot of the evaluation results in Fig. 7. Also, Fig. 8 shows a histogram of the errors. The maximum error was 12%. The number of test samples in less than 4% error was about 1700, which is more than 84% of the test samples.

We investigated the effect of epochs. Specifically, we train the proposed model using epoch 500. Then, we evaluated the models at a 50 epoch period. As shown in Fig. 9, the losses converged until epoch 50. The mean absolute errors were comparable after epoch 50. Therefore, epoch 100 is sufficient.

### B. COMPARISON ON FEATURE EXTRACTORS

The proposed method uses the VGG16 model as a feature extraction CNN. The VGG16 plays a vital role in range estimation in the proposed method. However, there are various CNN models other than the VGG16. We carried out experiments using other models as feature extractors to search for a better feature extractor.

We used Xception [36], Inceptionv3 [37], Resnet50 [38], and DenseNet [39]. The evaluation results using ResNet50 are 6.8057 at MAE and 0.078625 at the correlation coefficient. However, the learning processes of these models were not converged. The fail of training may be due to the large number of parameters to be trained. The dataset is insufficient to train them. We note that the same phenomenon has reported in [24]. Besides, the consumption of hardware resources and the computation time increased. Specifically, it costs 45.5 (ms/image) on VGG16, whereas 140 (ms/image) on ResNet50. Therefore, VGG16 is considered to be more practical for our system.

### C. VERIFICATION ON LOCAL IMAGES

To verify the effectiveness of using the local images, we compared the proposed method with and without the RGB...
and depth local images. We showed the experimental results in Table 4. The performance improved with the local images. Therefore, we confirmed that the local images contributed to the fat content estimation. Mackerels store fat in their skin to keep the body temperature. Thus, the skin textures extracted by VGG16 from RGB images are essential to estimate fat content. According to the results, the local images captured the texture. We successfully extract features from the local images for fat content estimation.

### D. VERIFICATION ON DEPTH IMAGES

We carried out the experiments to verify the effectiveness of the depth image. We evaluated the proposed method by removing the depth images and the related layers. Also, we replaced the depth images with negative images and edge images. Fig. 10 shows examples of the replaced images.

The experimental results are shown in Table 5. In all cases, the depth image marked the best accuracy. The results confirmed the significance of the depth image. We show the estimated fat content with and without depth image in Fig. 11. The results demonstrated the effectiveness of the feature extraction from depth images.

### E. DISCUSSION ON THE LENGTH OF MACKERELS

We discuss the effect of the length of mackerels on fat content estimation. The proposed method cropped mackerels and resized them into the fixed size, $224 \times 224$. Therefore, the proposed method omitted the actual length of the mackerels. We analyzed the relationships between length and fat content. The results are shown in Fig. 12. We used the 32 mackerels caught in December 2019, which are the same as Table 1.

We obtained the approximate line $y = 0.127x - 31.05$ using the least square method. The results show that there is a correlation between length and fat content. Therefore, we can expect further improvements by incorporating length information into the fat content estimation.

### F. DISCUSSION ON FISH DIRECTION

We investigated the directions of fishes in the dataset. The up and down directions are 47% and 53% in the training data, 43% and 57% in the test data. All fishes have left direction. For evaluation of fish direction, we conducted experiments using vertical and horizontal flips. Specifically, we trained models using data augmentation with vertical and horizontal flips to the training data. Then, we evaluated the trained models on the test data using the flips. Table 6 shows that the model trained with the original training data was suffered from the flipped test data. The performance improved using data augmentation on all test data. Therefore, data augmentation with the flips is effective for fat content estimation.

### VI. CONCLUSIONS

We proposed a method for estimating the fat content of mackerels from RGB and depth images. The proposed method estimates the mackerel region with a small computational cost using the feature map of the VGG16 model. The global and local images that contain the whole mackerel, head, body, and tail are extracted from the estimated region, and the features are extracted from the global and local images of RGB and depth. The extracted features are merged gradually and the fat content of the mackerel is estimated.

We conducted experiments to compare the estimated fat content with the values measured by the NIR spectroscopy sensor. The experimental results show the effectiveness of the proposed method. Introducing the proposed system to the fish market and assessing the effectiveness of the proposed method in a real situation is an important future work. In this study, we conducted experiments on mackerel, however, the proposed method can be used for other fish as well. It is also a future work to confirm the effectiveness of the proposed method for various kinds of fish.

### REFERENCES

[1] K. Almendingen, H. M. Meltzer, J. I. Pedersen, B. N. Nilsen, and M. Ellekjaer, “Near infrared spectroscopy—a potentially useful method for rapid determination of fat and protein content in homogenized diets,” European journal of clinical nutrition, vol. 54, no. 1, p. 20–23, 2000.

[2] B. Borgström, r. Nordén, B. Åkesson, and M. Jägerstad, “A study of the food consumption by the duplicate portion technique in a sample of the

---

**TABLE 4. Results with or without the local images**

|          | MAE   | RMSE  | R2   | Correlation |
|----------|-------|-------|------|-------------|
| Without  | 2.39  | 3.13  | 0.64 | 0.80        |
| With     | 2.25  | 2.91  | 0.69 | 0.83        |

**FIGURE 10. Replacements for the depth images**

(a) Edge images

(b) Negative images

**TABLE 5. Results for depth image removal and replacements**

|          | MAE   | RMSE  | R2   | Correlation |
|----------|-------|-------|------|-------------|
| Without  | 2.39  | 3.15  | 0.64 | 0.80        |
| Negative | 2.51  | 3.32  | 0.60 | 0.60        |
| Edge     | 2.55  | 3.32  | 0.60 | 0.78        |
| With Depth| 2.25  | 2.91  | 0.69 | 0.83        |

**TABLE 6. Mean absolute error with and without data augmentation**

| training data | Horizontal | Original | Vertical | Horizontal |
|---------------|------------|----------|----------|------------|
|               |            |          |          |            |
| -             | -          | 2.25     | 2.74     | 3.18       |
| ✓             | -          | 2.13     | 2.34     | 3.12       |
| -             | ✓          | 2.04     | 2.58     | 2.24       |
| ✓             | ✓          | 1.67     | 1.71     | 2.01       |
FIGURE 11. Estimated fat content w/wo depth image

(a) GT: 27.7, RGB: 19.2, RGB + Dep: 25.3

(b) GT 33.2, RGB: 20.2, RGB + Dep: 27.7

(c) GT: 30.0, RGB: 19.4, RGB + Dep: 29.2

(d) GT: 16.2, RGB: 24.0, RGB + Dep: 17.9

FIGURE 12. The scatter plot of length and fat content

Fat content (%) vs. Length (mm): $y = 0.127x - 31.05$

FIGURE 12. The scatter plot of length and fat content

Sano et al.: Mackerel Fat Content Estimation using RGB and Depth Images

dalby population.” Scandinavian journal of social medicine, vol. 10, p. 9–98, 1975.

[3] H.-Z. ZHANG, W. ZENG, M. RUTMAN, and T.-C. LEE, “Simultaneous determination of moisture, protein and fat in fish meal using near-infrared spectroscopy,” Food Science and Technology Research, vol. 6, no. 1, pp. 19–23, 2000.

[4] M. M. M. Foud, H. M. Zawbaa, N. El-Bendary, and A. E. Hassanien, “Automatic nile tilapia fish classification approach using machine learning techniques,” in Hybrid Intelligent Systems (HIS), 2013 13th International Conference on, 2013, pp. 173–178.

[5] W. N. Khotimah, A. Z. Artin, A. Yuniarti, A. Y. Wijaya, D. A. Navastara, and M. A. Kalbuadi, “Tuna fish classification using decision tree algorithm and image processing method,” in Computer, Control, Informatics and its Applications (IC3INA), International Conference on, 2015, pp. 126–131.

[6] P. Mohanaiah, P. Sathyanarayana, and L. GuruKumar, “Image texture feature extraction using glcm approach,” International Journal of Scientific and Research Publications, vol. 3, no. 5, p. 1, 2013.

[7] A. Kitasato, T. Miyazaki, Y. Sugaya, and S. Omachi, “Automatic discrimination between scomber japonicus and scomber australasicus by geometric and texture features,” Fishes, vol. 3, no. 3, p. 26, 2018.

[8] S. Hasija, M. J. Buragohain, and S. Indu, “Fish species classification using graph embedding discriminant analysis,” in Machine Vision and Information Technology (CMVIT), International Conference on, 2017, pp. 81–86.

[9] M.-C. Chuang, J.-N. Hwang, F.-F. Kuo, M.-K. Shan, and K. Williams, “Recognizing live fish species by hierarchical partial classification based on the exponential benefit,” in IEEE International Conference on Image Processing (ICIP), 2014, pp. 5232–5236.

[10] Y.-H. Hsiao and C.-C. Chen, “Over-atoms accumulation orthogonal matching pursuit reconstruction algorithm for fish recognition and identification,” in 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 1071–1076.

[11] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, E. S. Harvey, and H. editor: Howard Browman, “Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data,” ICES Journal of Marine Science, vol. 75, no. 1, pp. 374–389, 2017.

[12] Z. Ge, C. McCool, C. Sanderson, and P. Corke, “Modelling local deep convolutional neural network features to improve fine-grained image classification,” in IEEE International Conference on Image Processing (ICIP), 2015, pp. 4112–4116.

[13] G. Ding, Y. Song, J. Guo, C. Feng, G. Li, B. He, and T. Yan, “Fish recognition using convolutional neural network,” in OCEANS–Anchorage, 2017,
D. Rathi, S. Jain, and D. S. Indu, “Underwater fish species classification using convolutional neural network and deep learning,” arXiv preprint arXiv:1805.10106, 2018.

L. Meng, T. Hirayama, and S. Oyanagi, “Underwater-drone with panoramic camera for automatic fish recognition based on deep learning,” IEEE Access, vol. 6, pp. 17880–17886, 2018.

Y. Nagaoka, T. Miyazaki, Y. Sugaya, and S. Omachi, “Automatic mackerel sorting machine using global and local features,” IEEE Access, vol. 7, pp. 63767–63777, 2019.

W. Xu and S. Matzner, “Underwater fish detection using deep learning for water power applications,” arXiv preprint arXiv:1811.01494, 2018.

S. Choi, “Fish identification in underwater video with deep convolutional neural network: Snumbedinfo at lifesclef fish task 2015.” in CLEF (Working Notes), 2015.

X. Li, M. Shang, H. Qin, and L. Chen, “Fast accurate fish detection and recognition of underwater images with fast r-cnn,” in OCEANS 2015 - MTS/IEEE Washington, Oct 2015, pp. 1–5.

R. Mandal, R. M. Connolly, T. A. Schlacher, and B. Stantic, “Assessing fish abundance from underwater video using deep neural networks,” in International Joint Conference on Neural Networks (IJCNN), July 2018, pp. 1–6.

D. Zhang, G. Kopanas, C. Desai, S. Chai, and M. Piacentino, “Unsupervised underwater fish detection fusing flow and objectiveness,” in IEEE Winter Applications of Computer Vision Workshops (WACVW), March 2016, pp. 1–7.

K. Ito, H. Kawai, T. Okano, and T. Aoki, “Age and gender prediction from face images using convolutional neural network,” in Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, pp. 7–11.

Y. Mikawa, Y. Makino, and H. Shinoda, “Softness estimation based on images of pushing action using deep learning,” Transactions of the Virtual Reality Society of Japan, vol. 23, no. 4, pp. 239–248, 2018.

S. Tatemoto, Y. Harada, and K. Imai, “Image-based determination of plum tuuyuakane ripeness via deep learning,” Agricultural Information Research, vol. 28, no. 3, pp. 108–114, 2019.

Y. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, “Ordinal regression with multiple output cnn for age estimation,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4920–4928.

W. Li, J. Lu, J. Feng, C. Xu, J. Zhou, and Q. Tian, “Bridgenet: A continuity-aware probabilistic network for age estimation,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1145–1154.

E. Frank and M. Hall, “A simple approach to ordinal classification,” in Machine Learning: ECML, L. De Raedt and P. Flach, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 145–156.

K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in International Conference on Learning Representations, 2015.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” International Journal of Computer Vision (IJCV), vol. 115, no. 3, pp. 211–252, 2015.

S. Feki-Ershad, “Bark texture classification using improved local ternary patterns and multitlayer neural network,” Expert Systems with Applications, vol. 158, p. 113509, 2020.

L. Arni and S. Feki-Ershad, “Texture image classification based on improved local quinary patterns,” Multimedia Tools and Applications, vol. 78, no. 14, pp. 18995–19018, 2019.

C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, May 2011.

J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” The Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001. [Online]. Available: http://www.jstor.org/stable/2699986.

A. Alshahf, G. Azzopardi, B. Ducro, E. Hanenberg, R. F. Veerkamp, and N. Petkov, “Estimation of muscle scores of live pigs using a kinect camera,” IEEE Access, vol. 7, pp. 52238–52245, 2019.

N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, 2005, pp. 886–893 vol. 1.

F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807.