

# Marine Animal Classification Using Combined CNN and Hand-designed Image Features

Zheng Cao, Jose C. Principe

Department of Electrical and Computer Engineering  
University of Florida  
Gainesville, FL 32611 USA  
zca087@ufl.edu, principe@cnel.ufl.edu

Bing Ouyang, Fraser Dalglish, Anni Vuorenkoski

Harbor Branch Oceanographic Institute  
Florida Atlantic University  
Fort Pierce, FL 34946 USA  
[bouyang, fdalgley, adalgley]@fau.edu

**Abstract**—Digital imagery and video have been widely used in many undersea applications. Online automated labeling of marine animals in such video clips comprises of three major steps: detection and tracking, feature extraction and classification. The latter two aspects are the focus of this paper. Feature extracted from convolutional neural network (CNN) is tested on two real-world marine animal datasets (Taiwan sea fish and Monterey Bay Aquarium Research Institute (MBARI) benthic animal), and yields better classification results than existing approaches. Appropriate combination of CNN and hand-designed features can achieve even higher accuracy than applying CNN alone. The group feature selection scheme, which is a modified version of the minimal-redundancy-maximal-relevance (mRMR) algorithm, serves as the criterion for selecting an optimal set of hand-designed features. Performance of CNN and hand-designed features are further examined for images with lowered quality that emulates bad lighting condition in water.

## I. INTRODUCTION

Undersea applications such as hydrokinetic site monitoring and fishery stock assessment use extensively digital imagery and video to analyze the marine animals' behavior, seasonal distribution, and abundance. Such imagery and video are normally acquired by video camera or Lidar attached to a fixed installation [1] [2] or mounted on the unmanned underwater vehicles (UUVs) [3].

However, manually labeling such video clips is often not feasible considering the sheer size of collected data. Therefore, researchers seek an automated solution such that real-time animal recognition becomes possible. This procedure comprises of three major steps: detection and tracking, feature extraction and classification. Detection locates objects of interest. Tracking associates the blobs in consecutive frames that correspond to the same object, which provides animals' motion information. The outputs of detection and tracking are typically individual images with grouping information, i.e. an "event" includes images that belongs to the same animal. Afterwards, finding good features from the images is a crucial step towards classification.

Hand-designed features that explicitly describe objects shape, color or texture are often used in marine animals classification works [1] [2] [4] [5]. In [1] [2], Gabor filters and grey-level co-occurrence matrices are applied to extract texture features, while Fourier descriptors are used as shape features. Normalized color histogram is introduced in [2] for additional color features. In [4], texture properties are obtained via

vertical directive filtering. Shape and texture features have also been used in previous non-natural environment studies [6] [7]. In many cases, biological characteristics such as body part ratio [2] [8] can well distinguish between species when sufficient knowledge about the subjects are available. Other computer vision algorithm adopted in live fish recognition area include Haar-like feature [9] and local jets [3]. Performing contour extraction prior to feature extraction enables some features such as shape that are otherwise unavailable, and enhances some others by determining area of interest in advance. Recent advance in deep neural network has brought new light into computer vision. The DeCAF framework [10] discovers saliency in the natural image domain with a deep learning architecture. This approach has shown high accuracy and robust performance on different datasets. Furthermore, unlike many hand-designed features, it does not need any preprocessing on images.

In this paper, the DeCAF approach will be applied on two real-world datasets: Taiwan sea fish [2] and Monterey Bay Aquarium Research Institute (MBARI) benthic animal [3]. A novelty of this paper's work is to show that proper combination of CNN and hand-designed features can improve classification performance. Hand-designed features are usually outputs from empirical filters that have explicit physical interpretations, while the CNN can be regarded as a complicated filter whose parameters are learned from a large image dataset. It is possible that they describe different aspects of natural images and complement each other. However, this intuition alone does not guarantee that arbitrarily combined features will have more discriminative power. It is important [11] to select an optimum subset of feature types from all features (CNN and 1 or more types of hand-designed features). This will reduce computation complexity and ensure classification performance in the following feature combination stage. Furthermore, as a general rule the descriptive ability of certain feature may differ on distinctive datasets. Since underwater image quality varies with many factors such as lighting condition change and different hardware being used, it will be interesting to check how the trained neural network and hand-designed features respond to images with lowered quality.

The rest of the paper is organized as follows: Section II gives a review of the DeCAF framework and describes steps for designing hand features. Section III introduces strategies for choosing appropriate hand-designed features and combining features in classification. Section IV presents experimental

results. Section V concludes the discoveries.

## II. FEATURE DESIGN

### A. DeCAF

Deep neural network has the ability of capturing salient aspects of given domains, e.g. natural scenes or man-made indoor objects. Training such neural network on larger image dataset will better catch the variability of objects in realistic settings. The ImageNet [12] has a collection of over 14 million high-resolution images in 22000 categories, all labeled by human using the Mechanical Turk. A subset of ImageNet with 1.2 million training and 150,000 testing images was used in the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), in which the winner group designed a new structure of deep convolutional neural network (CNN) [13]. The network has five convolutional layers and three fully-connected layers, with the last layer producing class label predicts. The CNN has several desirable features: rectified linear units (ReLU) nonlinearity enhances performance and eliminates the need for input normalization, optimized GPU implementation increases training efficiency, and the dropout technique helps reducing over-fitting.

DeCAF [10] adopts the CNN structure of [13] and uses the same train data, but takes a different perspective that the outputs from the first two fully-connected layers can serve as features. This kind of feature generalizes well to colored natural images outside ImageNet, such as Caltech-101 and Caltech-UCSD birds. DeCAF has also shown better semantic clustering of labels than conventional computer vision features, both within and outside the ImageNet dataset. Users of DeCAF can think it as a feature extractor that has been pre-trained on ImageNet in a supervised manner. This pre-training makes CNN competent in handling small training data, where over-fitting is often a problem.

The aforementioned properties make DeCAF an appealing tool in feature extraction for marine animal imagery. Plus, the abundance of fish (566 subcategories, 280,000 images in total) and other marine animals in ImageNet further guarantees appropriate representations of the inputs.

In this paper, DeCAF is implemented using the ConvNet Python module and pre-trained ImageNet parameters from the nolearn package (<https://pythonhosted.org/nolearn/index.html>). For any image, the 7th layer of ConvNet generates a 4096 dimensional CNN feature. On a computer with Intel i3-3240 CPU and 8GB RAM, the execution time for extracting a single image's CNN feature is about 0.15s, which does not vary with image size due to the pre-wrap stage.

### B. Hand-designed Features

The extraction of hand-designed features for Taiwan fish data is detailed in [2]. This section describes the feature extraction process for the MBARI data. First the image is preprocessed to increase the uniformity of the background and improve the object contrast. This is achieved by decreasing the saturation of the pixels with hue value between 0.18 and 0.4, since the background is mostly green tone in this dataset – common for underwater video. The saturation reduction is inversely proportional to the pixel hue value. GrabCut [14] is

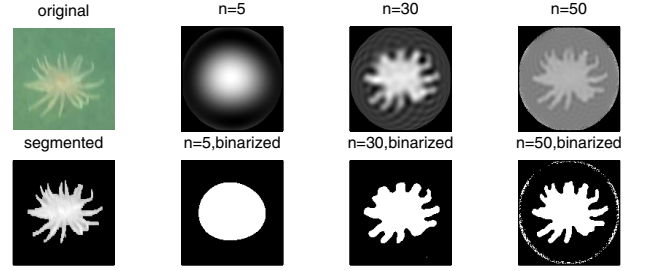


Fig. 1: Reconstructed image using Zernike moments.

then applied first to segment the object. An energy function is iteratively minimized for good separation of background and foreground:

$$E(\alpha, \mathbf{k}, \theta, \mathbf{z}) = U(\alpha, \mathbf{k}, \theta, \mathbf{z}) + V(\alpha, \mathbf{z}) \quad (1)$$

In (1),  $U$  is the data term and  $V$  is the smoothness term. The data term is expressed by

$$U(\alpha, \mathbf{k}, \theta, \mathbf{z}) = \sum_n (-\log \pi(\alpha_n, k_n) + \frac{1}{2} \log |\Sigma(\alpha_n, k_n)| + \frac{1}{2} [\mathbf{z}_n - \boldsymbol{\mu}(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1} [\mathbf{z}_n - \boldsymbol{\mu}(\alpha_n, k_n)]) \quad (2)$$

In (2),  $\alpha$  specifies whether any  $3 \times 1$  pixel  $\mathbf{z}_n$  belongs to the background ( $\alpha_n = 0$ ) or foreground ( $\alpha_n = 1$ ), while  $k$  ranges from 1 to  $K$ ,  $K$  being the number of Gaussian Mixture Models (GMMs, described by  $\theta$ , or coefficient  $\pi$ , mean  $\boldsymbol{\mu}$  and covariance  $\Sigma$ ) in background or foreground. Thus, term  $U$  can be seen as the sum of the Mahalanobis distance between any pixel and its nearest GMM among all  $2 \times K$  GMMs. In this particular dataset,  $K$  is set to 2 since neither the background nor the foreground has complicated pattern. This helps increase the uniformity of the background and hence the object boundary becomes more apparent. After GrabCut, median filtering is applied to refine the boundary.

GrabCut is also applied in [1][2], but mainly as a means of refining the contour extracted during the tracking process. The fact that the camera is fixed makes it possible to initialize background area from consecutive frames and segment out the object of interest.

While there are many object contour based shape descriptors, the Zernike moments [15] are adopted to take advantage of its rotation invariance property. Although Zernike moments are neither scaling nor translation invariant, it is easy to resize the image such that the object occupies a certain percentage (20%) and align the objects mass center with the image center. An example of image reconstructions using various orders of Zernike moments orders ( $n=5, 30$  and  $50$ ) are shown in Figure 1. In this work,  $n$  is set to 30. Too much detail is unnecessary considering the intra-class variability.

The other feature extracted from the segmented image, color histogram, is relatively simple. Each pixels red and green channel values are selected and normalized by the sum of all 3 channel values. Then a normalized histogram is built for the hue component and the normalized RG channels.

### III. FEATURE COMBINATION

#### A. Feature Selection

The minimal-redundancy-maximal-relevance (mRMR) [16] is a popular tool for feature selection. It is an iterative search for individual feature vectors that have large mutual information  $I$  between itself and labels, and small  $I$  between itself and selected features:

$$\max_{\mathbf{x}_j \in F, \mathbf{x}_j \notin S_{N-1}} \frac{I(\mathbf{x}_j; \mathbf{c})}{\frac{1}{N} \sum_{\mathbf{x}_i \in S_{N-1}} I(\mathbf{x}_j; \mathbf{x}_i)} \quad (3)$$

For easier computation of  $I$ , the input to mRMR are discretized with the 3-modality discretization. For every normalized feature vector across all training instances, -1, 0 and 1 are given respectively to values below  $-0.5\sigma$ , between  $\pm 0.5\sigma$ , and above  $0.5\sigma$ .

An obstacle to directly apply mRMR on the stacked CNN and hand-designed feature set is the large number of feature vectors. A small subset of feature vectors does not have enough discriminative power, while choosing a larger subset makes incremental search costly. This problem can be dealt with the group feature selection scheme [17]. Suppose mRMR draws  $m$  features from all candidate features. The relative importance of a feature type (e.g. color histogram or shape moments) is characterized by the ratio  $\beta_k$

$$\beta_k = \frac{m_k}{m} \quad (4)$$

where  $m_k$  is the number of features selected from the  $k^{th}$  feature type. If  $\beta_k$  for some feature type is particularly small (e.g. far less than  $1/M_0$ ,  $M_0$  being the total number of feature types), the feature type will be discarded.

#### B. Feature Combination in Classification

The classifier used in this paper is one-versus-rest SVM. While a simple way to combine features is to directly stack them together, a more comprehensive method – multiple kernel learning (MKL) is evaluated in this work. MKL applies when different features are known to demand different kernels, or when only one feature type is involved but the optimal kernel type and parameters are unknown. The standard SVM has the following form [18]

$$y(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i \kappa(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (5)$$

where  $\alpha$  and  $b$  are trained SVM coefficients and bias,  $x_i, y_i$  are input data and corresponding labels and  $y(x)$  is the predicted output. For MKL, the kernel  $\kappa$  in (5) is a linear combination of  $M$  kernels, where  $M$  here is the number of feature types. The coefficients  $\beta_k$  is indicative of relative importance of individual kernels [18]. The optimization of  $\alpha, b$  and  $\beta_k$  is done using the LP-Norm MKL algorithm [19]. Alternatively, one can make a fast combination by modifying (5) into

$$y(\mathbf{x}) = \text{sign}\left(\sum_{k=1}^M \beta_k \sum_{i=1}^N \alpha_i y_i \kappa_k(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (6)$$

where  $\beta_k$  is derived from (4).

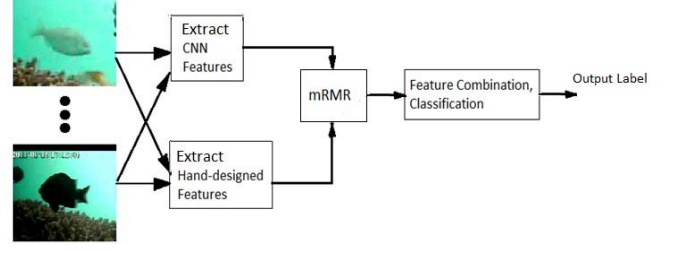


Fig. 2: Feature extraction, selection and classification.

For simplicity, only  $M = 2$  kernels are used: a histogram intersection (HI) kernel for CNN features and a linear kernel for the different types of hand-designed features. Both kernels are fast to compute and have no free parameters. When MKL is applied, the features need to be normalized: L1-norm normalization for HI kernel and L2-norm normalization for linear kernel. The whole procedure of feature extraction, selection and combination is summarized in Figure 2.

### IV. EXPERIMENTAL RESULTS

#### A. Taiwan sea fish data

The Taiwan sea fish data have been studied throughout the European Fish4Knowledge project, including works with a focus on feature extraction and classification [1][2][20]. This dataset has 23 types of fish and 27370 fish instances in total. The images (colored+mask) and code for these hand-designed features are from <http://sourceforge.net/projects/fish4knowledge/sourcecode/>. Feature dimensionality is 4096 for CNN and 2626 for hand-designed features. The latter is composed of 10 different feature types, whose respective dimensionalities are summarized in Table I. Each type of hand-designed features is preprocessed with z-normalization to eliminate the effect caused by difference in feature magnitude.

Five-fold cross validation is used for mRMR and later classification. Let  $m$  equal the number of features user selects using mRMR from concatenated CNN and hand-designed features. Figure 3 shows the top five feature types and their proportion among all  $m$  selected features. CNN feature is selected more often than all hand-designed features combined, yet some hand-designed feature are still competitive. Another interesting phenomenon is that the proportion of any feature type is quite constant for different  $m$ . The density may look like an exception, but this is due to the fact that it has only 12 dimensions. It is actually quite an important feature since 5 out of  $m=100$  features belong to density. Among the 10 feature types, the least selected ones are curvature shape ( $0/m=100$ ,  $0/m=1000$ ), PHOG ( $0/m=100$ ,  $1/m=1000$ ) and Fourier descriptor ( $0/m=100$ ,  $1/m=1000$ ).

In classification, two strategies are compared when hand-designed features are involved: one is to use all, the other is to discard the 3 least selected feature types and use only the remaining 7. Table II shows that with reduced feature dimension, the time for computing the linear kernel is shortened, while time for optimizing MKL is not affected much. Table III lists the classification results. CNN features perform better than the hand-designed ones, but is beaten by the combination of both. Also, using partial features does not negatively affect

**TABLE I:** Hand-designed features and their dimensions.

|                 |                    |               |               |                  |
|-----------------|--------------------|---------------|---------------|------------------|
| Color Histogram | Curvature shape    | Density       | Co-occurrence | Moment invariant |
| 930             | 2                  | 12            | 720           | 42               |
| PHOG            | Fourier descriptor | Gabor feature | AMI           | Body part ratio  |
| 680             | 15                 | 160           | 63            | 2                |

**TABLE II:** Effect of dimension reduction on computation time (using partial hand-designed features / using all)\*100%

|                                 |                         |                       |
|---------------------------------|-------------------------|-----------------------|
| Hand-designed feature dimension | Kernel computation time | MKL optimization time |
| 73.46%                          | 75.57%                  | 96.95%                |

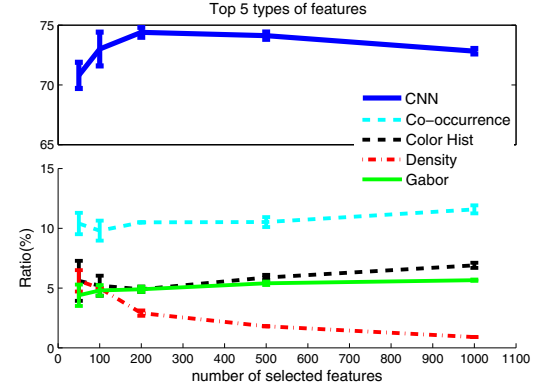
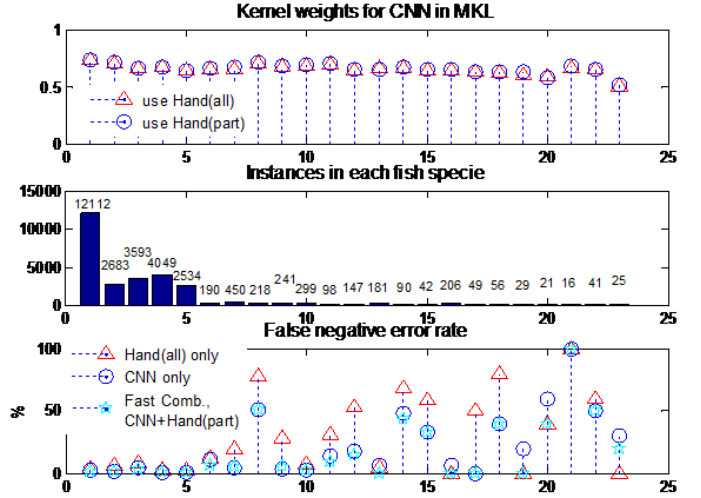
**TABLE III:** Classification error for Taiwan fish data.

| Method          | Overall error     | Average False Negative | Average False Positive |
|-----------------|-------------------|------------------------|------------------------|
| CNN(C)          | $2.95 \pm 0.47\%$ | $22.03 \pm 6.55\%$     | $0.19 \pm 0.03\%$      |
| Hand,all(Ha)    | $5.79 \pm 0.34\%$ | $30.50 \pm 6.00\%$     | $0.36 \pm 0.02\%$      |
| Hand,part(Hp)   | $5.96 \pm 0.51\%$ | $30.85 \pm 2.46\%$     | $0.38 \pm 0.03\%$      |
| C+Ha:Lp-MKL     | $2.09 \pm 0.48\%$ | $20.22 \pm 4.33\%$     | $0.15 \pm 0.04\%$      |
| C+Hp:Lp-MKL     | $2.15 \pm 0.38\%$ | $19.51 \pm 3.59\%$     | $0.15 \pm 0.03\%$      |
| C+Ha:Fast Comb. | $2.23 \pm 0.59\%$ | $18.74 \pm 4.72\%$     | $0.15 \pm 0.04\%$      |
| C+Hp:Fast Comb. | $2.09 \pm 0.51\%$ | $18.50 \pm 4.33\%$     | $0.14 \pm 0.03\%$      |

classification accuracy. For MKL, the weights in every class for CNN kernel are shown in the upper plot of Figure 4. Most of them are between 0.6 and 0.7, which suggests that MKL treats CNN as more important than hand-designed features. For fast combination (7), the weights are chosen as  $\beta_1=0.73$  and  $\beta_2=0.27$  respectively for CNN and hand-designed features, according to Figure 3. The same  $\beta$  are used for all 23 one-versus-rest SVMs. Despite its extremely low computational cost, the results turn out to be quite good. The reason that false negatives are high is because of the huge imbalance in data (middle plot of Figure 4). The lower plot of Figure 4 compares the false negative rate of using CNN, hand-designed features and fast combination in every class.

### B. MBARI data

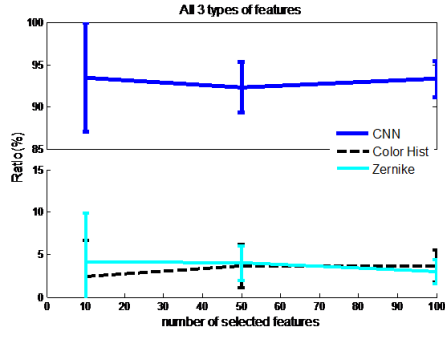
The MBARI benthic animal dataset has 4 different species, 48 events and 260 images. Previous study [3] achieved a 90% recall (true positive) rate on the Rathbunaster class using local jets [21] as feature, which is not very satisfying considering its visual distinctiveness. As mentioned before, the hand-designed features here are Zernike moments and color histogram, which have 255 and 63 dimensions respectively. All features are normalized the same way as in the previous example. Because of the small data size, 100 Monte-Carlo trials are conducted, with each trial randomly choosing half data for training and other half for testing. The proportion of selected features are shown in Figure 5. CNN is still the dominant feature, while the hand-designed features are also selected (about 3-4% for each). Therefore, both types of hand-designed features are kept

**Fig. 3:** Proportions of top 5 feature types selected for different  $m$ **Fig. 4:** Upper: Weights for the CNN HI kernel. Middle: Instance number for each class. Lower: Comparison of false negative error rate.**TABLE IV:** Classification error for Taiwan fish data.

| Method                          | Overall error     | Average False Negative | Average False Positive |
|---------------------------------|-------------------|------------------------|------------------------|
| CNN                             | $1.38 \pm 2.22\%$ | $2.04 \pm 3.49\%$      | $0.46 \pm 0.76\%$      |
| Hand,all                        | $5.04 \pm 4.03\%$ | $7.23 \pm 5.97\%$      | $1.61 \pm 1.26\%$      |
| MKL: CNN+Hand(all)              | $0.83 \pm 1.78\%$ | $1.33 \pm 2.97\%$      | $0.28 \pm 0.60\%$      |
| Fast combination: CNN+Hand(all) | $1.08 \pm 2.02\%$ | $1.50 \pm 3.03\%$      | $0.35 \pm 0.67\%$      |

in classification.

Classification results are listed in Table IV. For fast combination, the weights are chosen as  $\beta_1=0.93$  and  $\beta_2=0.07$ . The contribution of the hand-designed features are present in both MKL and fast combination. Compared with [3], the recall for Rathbunaster is much higher (no less than 99.5% for any single feature or feature combination). This dataset is arguably much simpler than the Taiwan fish one, since only about 130 training instances are used when such high classification accuracy is achieved.



**Fig. 5:** Proportions of all 3 types of features selected by mRMR for different  $m$ .

### C. Simulated degradation

Simulated degradation are injected into the Taiwan sea fish images to investigate the impact of increased camera/object separation and camera noise. Images are corrupted by one or more of the following 4 operations: (i) Lowered resolution: images are subsampled with bicubic interpolation. (ii) Noise: uniformly distributed noise (maximum level is 50, whereas each of RGB channel ranges from 0 to 255) is added independently on every pixel. (iii) Attenuation: the formula for attenuation caused by color spectrum variation is [22]

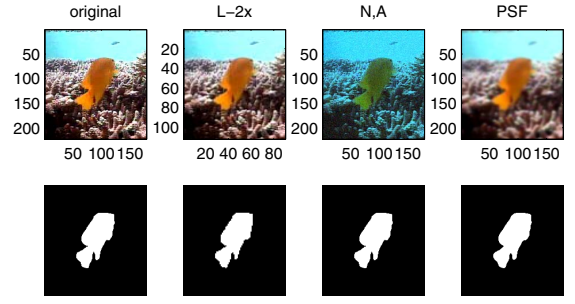
$$I_2 = I_1 * \exp(-cx) \quad (7)$$

where  $I_1$  and  $I_2$  are the original and altered intensities respectively,  $c$  is the absorption coefficient, and  $x$  is distance in meters. Absorption coefficients are  $c_{red}=0.3m^{-1}$ ,  $c_{green}=0.05m^{-1}$  and  $c_{blue}=0.015m^{-1}$  [23]. The camera/object separation is assumed to have been increased by 5 meters. (iv) Point spread function (PSF). Convoluting PSF with the image resembles lighting situation in turbid water. The PSF used here is a single, centered 2-D Gaussian kernel with  $\sigma=2$ . A practical problem is to decide how masks or the extracted fish contour changes after image degradation. For operations (i) and (iv), the same subsampling and PSF applied to the color images are applied to the masks. But operations (ii) and (iii) are performed only on the color images, because contour extraction is unavailable without original frames from the underwater video. Still, one can expect that the contours do not change much with Grabcut contour refinement. Original and degraded images are shown in Figure 6.

Table V summarizes the clear water results and results from degraded images. In general, hand-designed features based on segmentation are quite robust to the degradation, whereas CNN features are not as robust, possibly due to the fact that the CNN is not trained in a space that has images with such degradation. Also, the classification performances are in accordance with feature selection results. Higher proportion of hand-designed features usually indicates its better performance compared to CNN in a particular degradation setting.

### V. CONCLUSION

This paper proposes a practical approach to the problem of classifying marine animals from underwater imagery. For datasets with readily available hand-designed features, one can first decide whether all or part of the features are usable along



**Fig. 6:** Upper: color images. Lower: masks. L, N, A stand for lower resolution, noise and attenuation, respectively.

**TABLE V:** Classification error of images with degraded quality. Last column shows the proportion of hand-designed features in feature selection ( $m=100$ ).

| Image    | e%,CNN        | e%,Hand        | FN%,CNN        | FN%,Hand       | Hand%          |
|----------|---------------|----------------|----------------|----------------|----------------|
| Original | $3.0 \pm 0.5$ | $5.8 \pm 0.3$  | $22.0 \pm 6.6$ | $30.5 \pm 6.0$ | $27.0 \pm 1.4$ |
| L-2x     | $3.1 \pm 0.3$ | $4.7 \pm 0.3$  | $23.1 \pm 3.8$ | $25.7 \pm 4.8$ | $36.2 \pm 0.8$ |
| L-4x     | $4.0 \pm 0.6$ | $7.6 \pm 0.4$  | $31.6 \pm 6.2$ | $35.1 \pm 2.0$ | $33.6 \pm 0.5$ |
| N,A      | $4.3 \pm 0.6$ | $3.6 \pm 0.6$  | $29.9 \pm 4.2$ | $26.6 \pm 4.7$ | $39.6 \pm 1.1$ |
| L-2x,N,A | $6.2 \pm 0.8$ | $5.8 \pm 0.4$  | $35.4 \pm 3.3$ | $27.4 \pm 3.1$ | $47.2 \pm 0.8$ |
| L-4x,N,A | $9.1 \pm 0.8$ | $12.1 \pm 0.9$ | $45.6 \pm 4.8$ | $46.1 \pm 3.4$ | $41.4 \pm 0.5$ |
| PSF      | $3.7 \pm 0.6$ | $4.0 \pm 0.5$  | $28.9 \pm 4.9$ | $23.5 \pm 5.3$ | $42.8 \pm 1.3$ |

with the CNN features, then do classification with the combined features. For datasets without existing extracted features, either new features can be designed first, or the CNN features may be used alone, since it in general provides satisfying results. However, results have also shown that hand-designed features exhibit robustness when image quality is lowered. Using group feature selection not only provides knowledge about which features complement better with each other, but also helps reducing feature dimensionality without decreasing classification performance.

### ACKNOWLEDGMENT

This work was supported in part by US Department of Energy contract DE-EE0006787 and FAU/HBOI internal fund. The authors want to thank the collaborators from FAU Southeast National Marine Renewable Energy Center (SNMREC), Ms. Sue Skemp and Mr. Gabriel Alsenas for their assistance during this work.

### REFERENCES

- [1] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. J. Chen-Burger, R. B. Fisher, and G. Nadarajan. Automatic fish classification for underwater species behavior understanding. In *Proc. 1st ACM Int. Workshop ARTEMIS*, pages 45–50, 2010.
- [2] P. X. Huang, B. J. Boom, and R. B. Fisher. Underwater live fish recognition using a balance-guaranteed optimized tree. In *Proc. ACCV*, pages 422–433, 2013.
- [3] D. R. Edgington, I. Kerkez, D. E. Cline, J. Mariette, M. Ranzato, and P. Perona. Detecting tracking and classifying animals in underwater video. In *Proc. CVPR*, 2006.
- [4] A. Rova, G. Mori, and L. M. Dill. One fish, two fish, butterfly, trumpeter: Recognizing fish in underwater video. In *IAPR Conference on Machine Vision Applications*, 2007.

- [5] D. Semani, T. Bouwmans, C. Frlicot, and P. Courtellemont. Automatic fish recognition in interactive live video. In *Proceedings of the International Workshop on IVRCIA in the 6th World Multi-Conference on Systemics, Cybernetics, and Informatics*, July 2002.
- [6] R. Larsen, H. Olafsdottir, and B. Ersboll. Shape and texture based classification of fish species. In *SCIA09*, pages 745–749, 2009.
- [7] Y. Nagashima and T. Ishimatsu. A morphological approach to fish discrimination. In *MVA98*, 1998.
- [8] DJ. Lee, S. Redd, R. Schoenberger, X. Xu, and P. Zhan. An automated fish species classification and migration monitoring system. In *Proceedings of the 29th Annual Conference of the IEEE Industrial Electronics Society*, November 2003.
- [9] B. Benson, J. Cho, D. Goshorn, and R. Kastne. Field programmable gate array based fish detection using haar classifiers. In *American Academy of Underwater Science*, 2009.
- [10] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.
- [11] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [12] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [14] C. Rother, V. Kolmogorov, and A. Blake. Grabcut–interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 23(3):309–314, 2004.
- [15] B. Ouyang. *Watermarking based on unified pattern recognition framework*. PhD thesis, Southern Methodist University, 2007.
- [16] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *PAMI*, August 2005.
- [17] Z. Cao, J. C. Principe, and B. Ouyang. Group feature selection in image classification with multiple kernel learning. In *IJCNN*, 2015.
- [18] S. Sonnenburg, G. Raetsch, C. Schaefer, and B. Schoelkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.
- [19] M. Kloft, U. Brefeld, S. Sonnenburg, , and A. Zien. Lp norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, 2011.
- [20] P. X. Huang, B. J. Boom, and R. B. Fisher. Gmm improves the reject option in hierarchical classification for fish recognition. In *Applications of Computer Vision*, 2014.
- [21] M. Ranzato, P.E. Taylor, J.M. House, R.C. Flagan, Y. LeCun, and P Perona. Automatic recognition of biological particles in microscopic images. In *Pattern Recognition Letter*, volume 28, pages 31–39, 2007.
- [22] A. Morel and L. Prieur. Analysis of variations in ocean color. In *Limnology and oceanography*.
- [23] R.H. Stewart. *Introduction to Physical Oceanography*. Texas A & M University.