When Underwater Imagery Analysis Meets Deep Learning: a Solution at the Age of Big Visual Data

Hongwei Qin, Xiu Li, Zhixiong Yang and Min Shang
Graduate School at Shenzhen, Tsinghua University
Shenzhen, China
qhw12@mails.tisnghua.edu.cn

Abstract

Underwater imagery processing is in great demand, while the research is far from enough. The unrestricted natural environment makes it a challenging task. On the other hand, prior to the advent of cabled observatories, the majority of deep-sea video data was acquired by remotely operated vehicles (ROVs), and was analyzed and annotated manually. In contrast, seafloor cabled observatories such as the NEPTUNE and VENUS observatories offer a 24/7 presence, resulting in unprecedented volumes of visual data. The analysis of underwater imagery imposes a series of unique challenges, which need to be tackled by the computer vision community in collaboration with biologists and ocean scientists. In this paper, we introduce how deep learning, the state-of-the-art machine learning technique, can benefit underwater imagery understanding at the age of big data.

1. Large scale visual content recognition and deep learning

For underwater imagery analysis, one of the most important problem is visual content recognition, which is a quite challenging task. Visual content varies because of intra-class variability, which may result from variable illumination, scales, views and non-rigid deformations. Conventional solutions for classification use manually designed low-level features. For example, SIFT and HOG features are used for object recognition, LBP and Gabor features are used for texture and face classification. The carefully handcrafted low-level features do achieve good performance for some specific data and tasks. However, effective features require domain knowledge and most of them cannot simply apply to new conditions. Besides, the generalization capability of many conventional machine learning tools like SVM, PCA and LDA, tend to saturate quickly as the the volume of the training set grows significantly.

Hinton *et al.* [10] proposes a method to learn features through deep neural networks (DNNs), which significantly influences the machine learning field in recent years. Deep learning aims to learn multiple levels of representation of the data, from low-level to high-level, to make sense of data such as images, sound and text. High-level representation gives more information of the semantics of the data.

Deep and large networks have shown impressive results when used with large amounts of training data and scalable computation resources (thousands of CPU cores and/or G-PU computing [14]). Many conventional computer vision tasks have benefitted from this technical progress. Most notably, Krizhevsky *et al.* [14] proves the effectiveness of deep convolutional neural networks trained on ImageNet [3] and achieves an excellent classification accuracy. Besides, deep learning brings many other computer vision tasks to a brand new stage, such as object detection [8, 9, 7, 20, 21, 15], image segmentation[18, 2, 1, 23], image enhancement [5], image caption [12, 19, 22], 3D reconstruction [6, 17], video classification [13, 4] and so on. Deep learning based methods, especially deep convolutional neural networks based methods are beating traditional methods with state-of-the-art results in almost all the computer vision research topics.

In the past, people were sharing source files of their algorithms. While with deep learning toolkits like Caffe [11], people are sharing pre-trained models and network architectures. Datasets are becoming more and more dominating, sometimes even more important than algorithms. Models trained for classification tasks with large datasets like ImageNet are widely used for other classification tasks with a technique called finetune. Even non-classification tasks are benefiting from these models. For example, the models and corresponding network architectures are widely used as feature extractors. The extracted features of different network stages are comparable with and often better than low-level and high-level features extracted by traditional descriptors.

In the following section, we will take object detection and recognition for underwater videos as examples to demonstrate how deep learning can bring large scale underwater imagery understanding to a brand new stage.

2. Fish Recognition

2.1. Dataset

We evaluate the effectiveness of the deep learning on the Fish Recognition Ground-Truth dataset made by the Fish4Knowledge [1] project. This underwater live fish dataset is acquired from a live video dataset captured from the open sea. There are totally 27370 verified fish images of 23 clusters and each cluster is presented by a representative species. The fish species are manually labeled by following instructions from marine biologists.

2.2. CNN Architecture

Recently, in many object recognition systems, feature extraction stages are generally composed of a filter bank layer, a non-linear transformation, and a feature pooling layer. One feature extractor may contain several such stages. For example, the recently widely noted convolutional neural networks (CNN) is a such system. A typical CNN consists of several layers, and can be regarded as a stage. A convolutional filter bank layer aims to extract local patterns. A nonlinear processing layer aims to form a non-linear complex model. A feature pooling layer aims to decrease feature maps' resolution. Deep CNN may contain several stages.

The overall architecture of our CNN is depicted in Fig. 1. The net contains six layers with weights, of which the first three are convolutional and the remaining three are fully connected. The output of the last fully-connected layer is fed to a 23-way Softmax which produces a distribution over the 23 class labels.

This net maps an input image x_i, via a series of layers, to a probability vector $\hat{y_i}$ over the 23 different classes. Each layer consists of the following operations: (i) convolution of the previous layer output (or, in the case of the 1st layer, the input image) with a set of learned filters (optimized weights); (ii) passing the responses through a rectified linear function $relu(x) = \max(x, 0)$; (iii) max pooling over local neighborhoods and (iv) a local contrast operation that normalizes the responses across feature maps.

As demonstrated in Fig. 1, the first convolutional layer filters the $47 \times 47 \times 3$ input image with 70 kernels of size $5 \times 5 \times 3$ with a stride of 1 pixel. The outputs are max pooled with kernel size 3×3, fed to ReLU layer and normalized. The second and the third convolutional layers repeat the above process, besides that the filters kernel size are both adjusted to 3×3. In detail, the second convolutional layer filters the (pooled, rectified, and normalized) outputs of the first convolutional layer with 110 kernels of size $3 \times 3 \times 70$, and the third layer has 180 kernels of size $3 \times 3 \times 110$ connected to the outputs of the second layer. The fully-connected layers have 200, 22, 23 neurons in turn. In total, the network has about 1 million parameters. We denote this method with Deep-CNN.

Method	Accuracy(%)
LDA+SVM	80.14
Raw-pixel SVM	82.92
Raw-pixel Softmax	87.56
Raw-pixel Nearest Neighbor	89.79
VLFeat Dense-SIFT	93.58
Deep-CNN	**98.57**

Table 1. Comparison of fish recognition accuracy (%) of various methods on the test set.

2.3. Experiments

Stochastic Gradient Descent (SGD) is used to learn such a network. In the experiment, we get a test accuracy of 98.57%, which achieves the state of the art. The comparisons are listed in Table. 1.

2.3.1 Feature and filter visualization

Now we have a close look at the features and weights by visualizing them.

The activations of the three convolutional layers during the forward pass are shown in Fig. 2. The corresponding convolutional filters (weights) are illustrated in Fig. 3.

The first layer activations looks relatively blobby and dense. In the following layers the activations become more sparse and localized. This is quite interpretable because we can infer the class-specific features. While in conventional hand-crafted methods, these features are often discovered and decided by experts with domain knowledge, for example, biologists.

As for the convolutional filters, the first layer is looking directly at the raw pixel data, so the filters looks relatively nice and smooth. The following two layers' weights are not as interpretable, but still well-formed.

3. Fish Detection

Recent high quality object detection approaches use similar scheme: a fast and accurate object proposal methods followed by classifier using deep convolutional neural networks. As mentioned before, RCNN [8], fast-RCNN [7] and faster-RCNN [20] are among the most notable works. We use fast-RCNN for fast and accurate fish detection and get an average precision (mAP) 9.4% higher than Deformable Parts Model (DPM), which was a most successful and widely used object detection at the time before deep learning. Fig. 4 is our fish detection pipeline, more details can be found in our concurrent paper [16].

[1] http://groups.inf.ed.ac.uk/f4k/

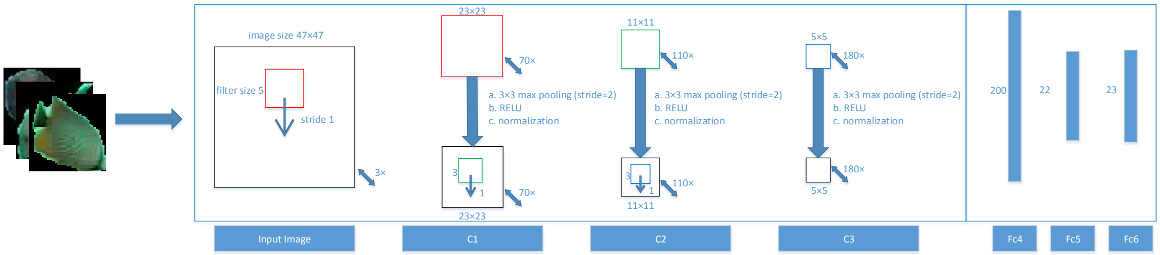
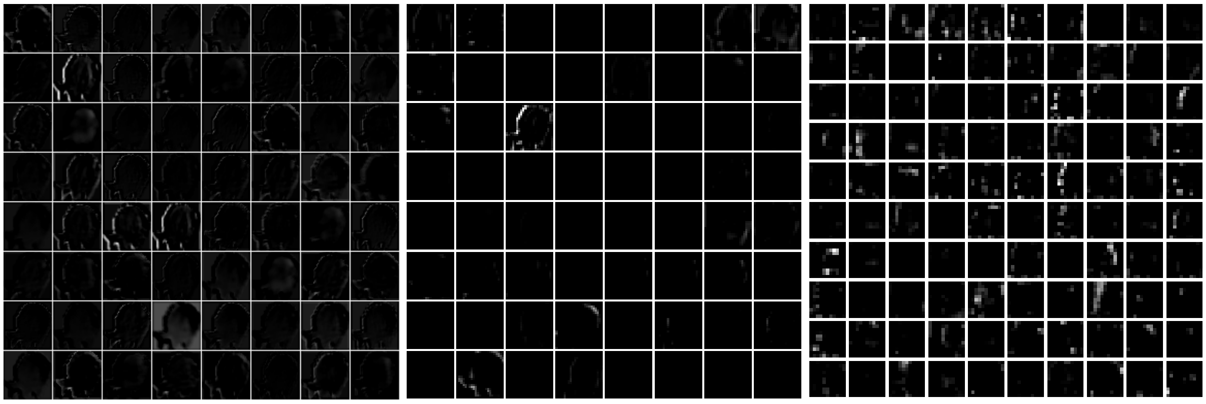


Figure 1. The architecture of our CNN, showing the definition of different layers. The network's input is 6627-dimensional, and the number of neurons of each layer is given by 37030-13310-4500-200-22-23.



(a) Layer 1

(b) Layer 2

(c) Layer 3

Figure 2. A typical visualization of the CNN architecture convolutional layer activations (rectified by ReLU) given an input image of fish. Each box shows an activation map corresponding to some filter. From left to right: (a) 64 outputs of size 23 × 23 by the first convolutional layer (64 out of 70 are shown). (b) 64 outputs of size 11 × 11 by the second convolutional layer (64 out of 110 are shown). (c) 100 outputs of size 5 × 5 by the second convolutional layer (100 out of 180 are shown). Notice that the last two layers' activations are mostly sparse (shown in black) and localized.

4. Plankton Classification

Plankton are critically important to our ecosystem, as they account for more than half the primary productivity on earth and nearly half the total carbon fixed in the global carbon cycle. They form the foundation of aquatic food webs including those of large, important fisheries.

Traditional methods for measuring and monitoring plankton populations are time consuming and dependent on humans, so they cannot scale to the granularity or scope necessary for large-scale studies. One possible solution is using an underwater imagery sensor to capture microscopic, high-resolution images over large study areas. The images can be post analyzed to assess species populations and distributions. While, to achieve automatic analysis, it is still a highly challenging computer vision task.

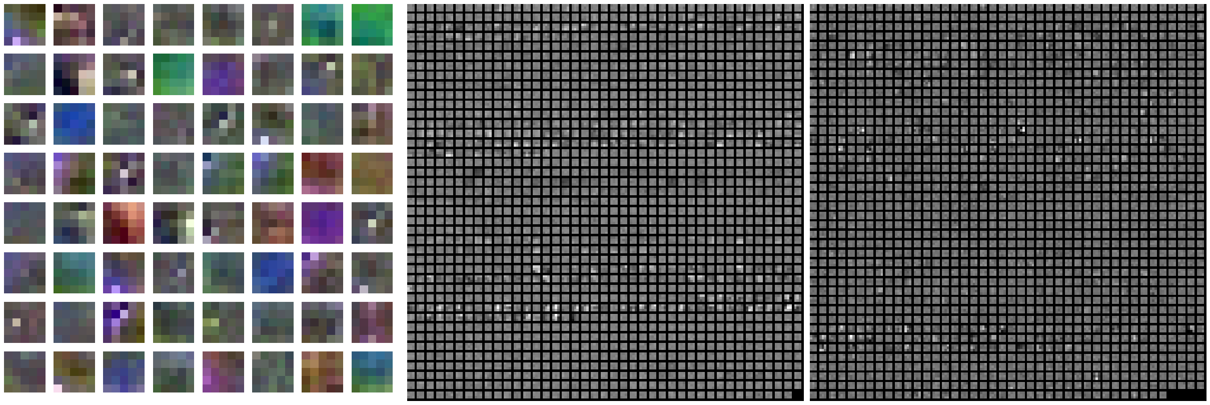
Recently, a Kaggle competition [2] was organized to classify plankton. The goal of the competition was to classify about 30000 grayscale images of plankton into one of 121 classes. In the final ranking list, all the leading teams used deep learning methods. Take the champion team [3] for example, they used Convnets as the model, and judicious combination of techniques to prevent overfitting such as dropout, weight decay, data augmentation, pre-training, pseudo-labeling and parameter sharing, which enabled them to train very large models with up to 27 million parameters on this dataset. The classification accuracy is 81.52% on the test set, which is much higher than results achieved by conventional methods.

5. Conclusion

In this paper, we introduce how deep learning can benefit underwater imagery analysis at the age of big data. Examples of fish recognition, detection and plankton classification show that deep learning can achieve state-of-the-art results on these tasks. Besides, deep learning can be useful

[2] https://www.kaggle.com/c/datasciencebowl

[3] http://benanne.github.io/2015/03/17/plankton.html



(a) Filters 1 (b) Filters 2 (c) Filters 3

Figure 3. Visualization of the CNN architecture convolutional layer weights. Each box shows a filter. From left to right: (a) 64 convolutional kernels of size $5\times5\times3$ learned by the first convolutional layer on the $47\times47\times3$ input images (64 out of 70 are shown). (b) 24 convolutional kernels of size $3 \times 3 \times 70$ learned by the second convolutional layer on the outputs of the previous layer (24 out of 110 are shown). (c) 16 convolutional kernels of size $3 \times 3 \times 110$ learned by the second convolutional layer on the outputs of the previous layer (16 out of 180 are shown). Notice that the kernels in (b)(c) are reshaped for visualization.

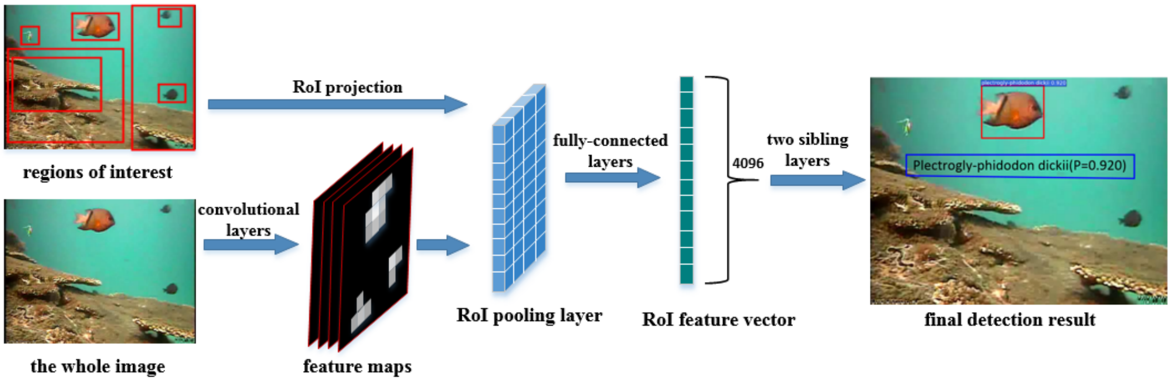


Figure 4. Overall architecture of our fish detector with fast-RCNN.

in many other underwater imagery analysis tasks, such as image and video enhancement. Other than improving the deep learning methods accuracy, improving the efficiency is another promising research topic. In conclusion, deep learning may not be the ultimate solution to computer vision, however, it can be a practical solution at the age of big visual data in ocean observation.

References

[1] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. *arXiv preprint arXiv:1503.01640*, 2015.

[2] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3992–4000, 2015.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014.

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014*, pages 184–199. Springer, 2014.

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[7] R. Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*

(CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision–ECCV 2014*, pages 346–361. Springer, 2014.

[10] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.

[13] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1725–1732. IEEE, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. *arXiv preprint arXiv:1505.02146*, 2015.

[16] 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-Washington, DC*. IEEE, 2015.

[17] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. *arXiv preprint arXiv:1411.6387*, 2014.

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[19] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[21] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014.

[22] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015.

[23] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv preprint arXiv:1502.03240*, 2015.