# Learning and Transferring Convolutional Neural Network Knowledge to Ocean Front Recognition

Estanislau Lima, Xin Sun, *Member, IEEE*, Junyu Dong, Hui Wang, Yuting Yang, and Lipeng Liu

***Abstract*— In this letter, we investigated how to apply a deep learning method, in particular convolutional neural networks (CNNs), to an ocean front recognition task. Exploring deep CNNs knowledge to ocean front recognition is a challenging task, because the training data is very scarce. This letter overcomes this challenge using a sequence of transfer learning steps via fine-tuning. The core idea is to extract deep knowledge of the CNN model from a large data set and then transfer the knowledge to our ocean front recognition task on limited remote sensing (RS) images. We conducted experiments on two different RS image data sets, with different visual properties, i.e., colorful and gray-level data, which were both downloaded from the National Oceanic and Atmospheric Administration (NOAA). The proposed method was compared with the conventional handcraft descriptor with bag-of-visual-words, original CNN model, and last-layer fine-tuned CNN model. Our method showed a significantly higher accuracy than other methods in both datasets.**

***Index Terms*— Convolutional neural networks (CNNs), fine-tuning, ocean front recognition, transfer learning.**

## I. Introduction

IN RECENT years, remote sensing (RS) has witnessed a gradual improvement in spatial resolution. The improvement of spatial resolution provided the RS images with detailed information related to spatial arrangement information and textural structures. With the advance of new technologies, researchers have described numerous methods engaging computer vision techniques to classify satellite image scenes. The bag-of-visual-words (BOVW) model [8], a common handcraft visual descriptor in computer vision that has been the state of the art for several years in the community, is considered one of the most popular approaches to solving the problem of scene classification. Due to the specificities of remotely sensed data, many of these traditional methods are not applicable in the RS domain. In recent years, new methods that are able to effectively encoding spectral and spatial information have been proposed. Deep learning methods, especially convolutional neural networks (CNNs) [2], have gained popularity over the past years, due to their powerful ability to learn image representations. CNNs have been widely studied not only for classic problems, such as object recognition and detection, but also in many other practical applications, including RS imaging. It has obtained state-of-the-art results in many different RS applications, e.g., oil spill [3], [4]. The success of CNNs is due to their natural ability to effectively encoding spectral and spatial information based on mainly the data itself.

Ocean front recognition is vital when it comes to providing enlighten information concerning the properties and dynamics of the oceans and atmosphere. Thus, ocean front constitutes a fundamental key to understanding the majority of the oceanographic processes, namely, climate changes. Therefore, the aim of this letter is to investigate how deep learning methods, in particular CNNs, can be successfully applied as a new method to ocean front recognition. Applying CNNs to ocean front recognition is a challenging task since the specific training data are very scarce. We overcame this challenge using a CNN model and a sequence of transfer learning steps via fine-tuning. The core idea is to extract deep knowledge of the CNN model from a large data set and then transfer the knowledge to our task of limited training data. To the best of our knowledge, this letter presents the first work to apply the deep CNNs method to the ocean front recognition task.

## II. Related Work and Background Concepts

This section presents some fundamental concepts needed to understand this work.

### A. Ocean Front Recognition

Ocean fronts are sharp boundaries between different water masses and different types of vertical structure, which are usually accompanied by enhanced horizontal gradients of temperature, salinity, density, nutrients, and other properties [5]. In order to understand the oceanographic processes, ocean fronts have been a subject of study for many years. The literature gives a variety of methods and algorithms that have been proposed to address the problem of ocean front. The most popular methods include the gradient algorithms and the edge detector and entropy algorithms [6]. Due to the development of technological innovation and new instruments over the past decade, RS data such as high-resolution satellite imagery have gained popularity and is readily available and inexpensive. Consequently, with large quantities of data, new algorithms and methods for ocean front recognition and detection in satellite imagery have been proposed [7], [8].

### B. Convolutional Neural Networks

Deep learning methods [9], more specifically CNNs, are stated as the most advanced computer vision application for

Manuscript received September 19, 2016; revised November 15, 2016; accepted December 14, 2016. Date of publication January 23, 2017; date of current version February 23, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 41576011 and Grant 61401413, in part by the Natural Science Foundation of Shandong Province under Grant ZR2014FQ023, and in part by the NVIDIA Academic Hardware Grant. *(Corresponding authors: Junyu Dong and Xin Sun.)*

E. Lima, X. Sun, J. Dong, Y. Yang, and L. Liu are with the Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: dongjuyun@ouc.edu.cn; sunxin@ouc.edu.cn).

H. Wang is with the College of Physical and Environmental Oceanography, Ocean University of China, Qingdao 266100, China.

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2016.2643000

recognition and detection tasks. The idea of CNNs was first proposed by Fukushima [10]. Later on, LeCun *et al.* [11] designed a CNN model LeNet-5 to recognize hand-written digits. A typical CNN consists of three types of layers: the convolution layers, the pooling layers, and the fully-connected (fc) layers, with the latter being a classifier layer. It gained popularity over the past years due to the remarkable results in the ImageNet Challenge 2012 [2]. Thenceforward several researchers have explored the potential of the deep CNNs to outperform many classical approaches for object recognition and detection [12]. It also has been widely studied for many real practical applications including RS, image classification, and oil spill [3], [4].

### C. Transfer Learning

Transfer learning is preferable when the target data set is reasonably large, but not enough to train a new network from scratch. Transfer learning can be a powerful tool to enable training a large target network. A lot of work has been done with the CNNs using the transfer learning method in RS fields. In certain tasks, such as image classification [13] and poverty mapping [3], transfer learning methods have achieved great results. The transfer learning methodologies can be separated into two distinctive subdivisions.

*1) Feature Extractor:* A pretrained network can be used as a feature extractor for any image. For instance, one can take a CNN pretrained model, remove the last fc layer (classifier layer), and then treat the rest as a fixed feature extractor to adapt it to the new task. Features trained on ImageNet have already shown remarkable results in many applications, such as flower categorization and bird subcategorization [14].

*2) Fine-Tuning:* Fine-tuning is a good option to extract features, when the new data set is not large enough to fully train a new network. The fine-tuning strategy not only replaces and retrains the classifier to adapt to the new data set, but also includes weighing the pertained network by continuing the backpropagation. Fine-tuning can be applied to all layers of the model or some of the earlier layers can be kept fixed and only fine-tune some higher level portion of the network. Fine-tuning has been shown to be the best strategy in the field of RS for the task of image classification, with remarkable results [3]. More details about how to fine-tuning will be described in the following sections.

## III. Proposed Method

The goal of our work is to apply deep learning method, i.e., CNNs, to solve the ocean front recognition problem. The most challenge is the very scarce training dataset. We overcame the training data scarcity challenge using a sequence of transfer learning steps via fine-tuning to the CNN model. To the best of our knowledge, there is no other research on ocean front recognition based on the state-of-the-art deep CNN method.

The deep architecture proposed by Krizhevsky *et al.* [2] was applied to the fine-tuning procedure. It has 60 million parameters and 650 000 neurons. This network consists of two types of layers: convolutional layers and fc layers. The success of AlexNet popularized the application of large CNNs in visual recognition tasks. AlexNet has therefore become a baseline architecture of modern CNNs. In this letter, AlexNet was used instead of others deep models, e.g., GoogleNet, because it was originally designed to classify over ImageNet. On the other hand, GoogLeNet was designed to be a direct improvement over AlexNet for the task of classifying ImageNet. Compared with AlexNet of 8 layers, GoogleNet has 22 layers and need more computing power than AlexNet, even though the number of parameters in the model is purportedly 12 times smaller.

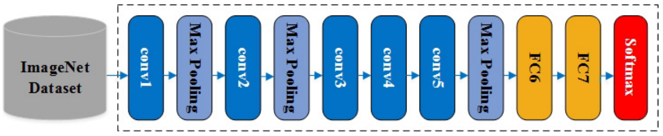

Fig. 1. Architecture of the CNNs used in our work (AlexNet): five convolutional layers (conv), two fc layers, and one classifier layer (softmax).

### A. Network Architecture

In this section, we briefly review the CNN architecture applied in our work. This architecture is presented in Fig. 1. The architecture takes a square 224 × 224 pixel RGB image as input and produces a distribution over the ImageNet object classes. The architecture is composed of five convolutional layers, three pooling layers, two fc layers, and finally a classifier layer. Very similar to the typical CNNs, the success of this architecture is based on several factors, such as availability of large data sets, more computing power, and availability of GPUs. The success of the network also depends on the implementation of additional techniques, such as dropout, data augmentation to prevent overfitting, and rectified linear units to accelerate the training phase.

### B. Ocean Front Recognition Task

Applying deep learning methods to ocean front recognition is a challenging task, because fronts have significant visual similarities and are indistinguishable on color and shape. In this letter, we focused on an interesting property of modern CNNs, which is the first few convolutional layers tend to learn features that resemble edges, lines, corners, shapes, and colors, independent of the training data. More specifically, earlier layers of the network contain generic features that should be useful to many tasks. Since we can define and characterize front as edges, lines, and corners, modern CNNs can be trained to recognize ocean front. The task is to train a CNN model to extract generic features that can be useful to our work. Generally, to train a full CNN, we need a large data set. However, our data set was not large enough to train the full CNNs; therefore, fine-tuning becomes the preferred option to extract the features.

### C. How to Fine-Tune the Network

Fine-tuning a network is a procedure based on the concept of transfer learning [15]. Specifically, it is a process that adapts an already learned model to a novel classification model. There are two possible approaches of performing fine-tuning in a pretrained network: first is to fine-tune all the layers of the CNNs. The second approach is to keep some of the

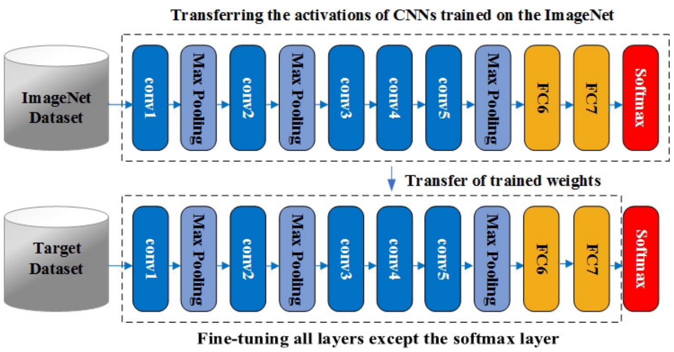

Fig. 2. Fine-tuning process. All layers are fine-tuned; basically, the last layer (softmax classifier layer) is ignored and only the layer used to extract the features needs to be defined.

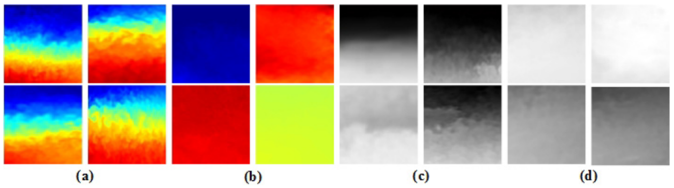

Fig. 3. Examples of the two data sets colorful and gray-level. (a) Class front from colorful data. (b) Class no-front colorful data. (c) Class front gray-level. (d) Class no-front gray-level.

earlier layers fixed (to avoid overfitting) and fine-tune only the higher level layers of the network. In the first approach, the classifier layer is removed from the pretrained CNNs and the rest of the CNNs are treated as a fixed feature extractor. In the second approach, the initial layers are frozen to keep the generic features already learned and the final layers are adjusted for the specific task. In other words, fine-tuning uses the parameters learned from a previous pretrained network on a specific data set and then adjusts the parameters from the current state for the new data set. In this letter, we fine-tuned all layers and assumed that the features from all layers were important for our task. The workflow for this approach is illustrated in Fig. 2.

## IV. Experiments

To evaluate the effectiveness of the proposed method, experiments were conducted on two different RS image data sets, illustrated in Fig. 3.

### *A. Data Sets*

We applied two challenging RS data sets with different properties to better evaluate the robustness and effectiveness of the proposed method described above. The data obtained from the National Oceanic and Atmospheric Administration (NOAA) contains data from different region and different years and are posteriorly processed and labeled using MATLAB. Two distinct data, colorful and gray-level data, were processed and each type of data contained two classes, front and no-front. The primary difference recorded between these two data sets is the color property, of which the gray-level images are original images. The colorful images were processed by interpolation algorithms. In order to validate the effectiveness of the proposed method in real applications, we used the original gray sea surface temperature (SST) images to appraise the proposed method.

This data set is particularly challenging to be labeled as front and no-front. To label our data, we first calculated the gradient for each pixel. For a gradient higher than two, the pixel was set as a fixed pixel. Afterward, the gradient of the next ten neighborhoods were calculated and described as indicated above. Results higher than two were cut into small pictures and labeled as front. Due to lack of knowledge and the difficulty of the task, the pictures were validated by professional oceanographers.

The two data sets colorful and gray-level are composed for 2000 images each, divided into two classes: 1) the front class contains 1279 images and 2) the no-front class contains 721 images. All high-resolution images referring to different regions and years were collected from NOAA. We processed all the images with a grid size of 2° for both data sets. The colorful date is RGB images with different sizes. Some samples of this dataset are presented in Fig. 3(a) and (b). The gray-level data used just one channel with different sizes. Some samples are shown in Fig. 3(c) and 3(d).

### *B. Training Procedure*

We created our preliminary training and validation sets by taking a stratified 10-fold of the provided training set, which split the provided training set into 90% for training and 10% for test, with two classes distributions as the provided training set [16]. To fine-tune the model, we kept the structure of pretrained model unchanged and removed only the last layer (classifier layer). The last layer of the original CNNs is a softmax classifier that computes the probability of 1000 classes of the ImageNet data set. After removing the last layer, we adjusted the others parameters to fit our goal. The pretrained CNNs required a fixed size (e.g., $224 \times 224$) as input image. Therefore, each image was resized to a fixed size in the network. We also changed the number of classes to just two, the front class and no-front class. For the training parameters, we ran the code for 5000 iterations and set the learning rate to a very small variations of 0.01 and 0.001. In addition, we fine-tuned the pretrained model with our training data to learning the weights inside the model. Then we used a support vector machine (SVM) classifier to classify the new learned feature, instead of using the softmax layer.

All our experiment sessions were carried out using the pretrained AlexNet model, available in the Caffe Model Zoo [17]. And we ran experiments on a dual eight-core Intel Xeon E5-2650 processor, with a Tesla K40 (2880 cores and 12 GB of RAM).

### *C. Comparison Methods*

The performance of the proposed method was compared with the original CNN model, the last-layer fine-tuned CNN model and the conventional handcraft descriptor with BOVW, which are the most popular approaches for image analysis and classification and have been widely used for general object recognition.

We conducted the experiments with both data sets and used the same configurations. In order to select the best

TABLE I
ACCURACIES OF THE FINE-TUNING PROCESS OF ALEXNET MODEL IN EACH DATA SET

| **AlexNet** | Fine-tuning all layers % |
|---|---|
| Colorful | 88 |
| Gray-level | 86 |

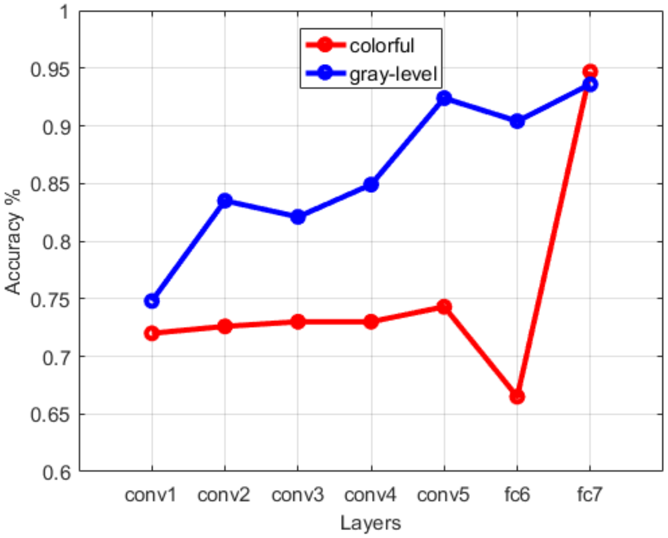

Fig. 4. Classification performance of the CNN features extracted from each layer of AlexNet. We used SVM to do the classification.

configuration for our experiment, we test BOVW in different configurations. The best BOVW configurations are based on dense sampling, soft assignment with max pooling, Scale Invariant Features Transform low-level descriptor, and a visual size of 50.

## V. RESULTS

The data set was different from the original data set and there were only two classes in the study. Furthermore, CNN models are designed and trained for generic object and/or scene recognition and not for RS images. Surprisingly, the results from the study were better than initially expected. As shown in Table I, the normalized accuracy of the proposed method of each data set in AlexNet reached an accuracy of 88% for the colorful data and 86% for the gray-level data.

### *A. Performance of Different Convolutional Layers*

After the fine-tuned model was obtained, we conducted feature extraction based on the model as discussed before. We then built an SVM classifier on top of the deep features generated from the fine-tuned CNNs. The classification results are shown in Fig. 4. The classification accuracy for the colorful data increased very slowly in the first five convolutional layers. The accuracy then dropped in the first fc layer. However, during the last fc layer, the accuracy increased to the highest accuracy. The classification accuracy for the gray-level data increased faster than for the colorful data in first five convolutional layers, and the last fc layer was validated as the more accurate. The colorful data achieved better result during the last fc layer compared with gray-level data. We believe that the possible explanation for the better performance of AlexNet in the colorful data rather than the gray-level data was due to the particular intrinsic properties of each data set. The color properties are significantly important since the first convolutional layers tend to learn features that resemble color.

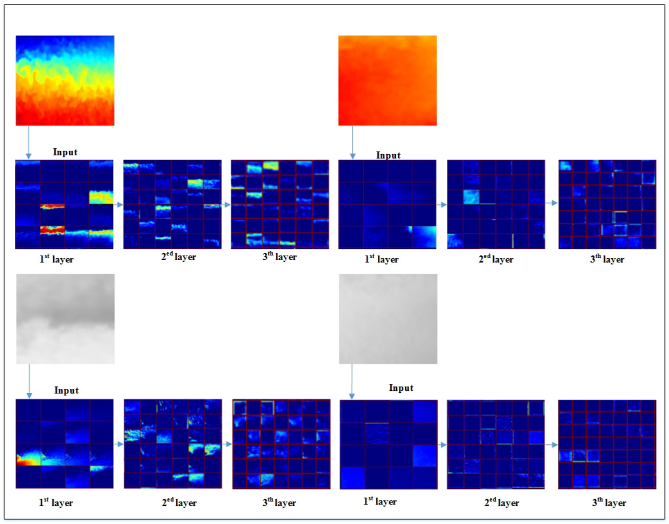

Fig. 5. Features visualization of convolutional layers of AlexNet. (Top) CNN features from colorful data. (Left) Visualization of class front. (Right) Visualization of class no-front. (Bottom) CNN features from gray-level data.

### *B. Features Visualization*

Understanding the operation of visualization of features learned by a CNN model requires interpreting the feature activity in intermediate layers. Fig. 5 shows feature visualizations from our proposed model once training was completed. From the results, we can see that the features from early layers 1–3 show better result compared with those from latter ones 4–5. We speculate that it may be due to the first convolutional layer tendency to learn features that resemble color, edges, lines, corners and shapes. The earlier layers of the network contain generic features that should be useful to the ocean front recognition task. The latter layers are closer to the label layer of the nature image classification problem, and thus they contained task-specific features that may not help our application.

### *C. Comparison With Different Baselines*

Fig. 6 shows the comparison in terms of normalized accuracy of our proposed method with different baselines: full training, fine-tuning higher layers, and BOVW. We first trained and tested the original AlexNet only with our data sets, and the results were very similar in both datasets. However, the full training performed better in the colorful data (83%) and in the gray-level data (81%). Comparing the results of the proposed method in relation to fine-tuning higher layers, we observe a big difference in the results, 55% in the colorful data and 45% in the gray-level data. One possible reason is that the earlier layers of the network tendency learn features that resemble color, edges, lines, corners, and shapes, which should be useful to the ocean front recognition task. Comparing the results of BOVW with those of our method, we can see that our proposed

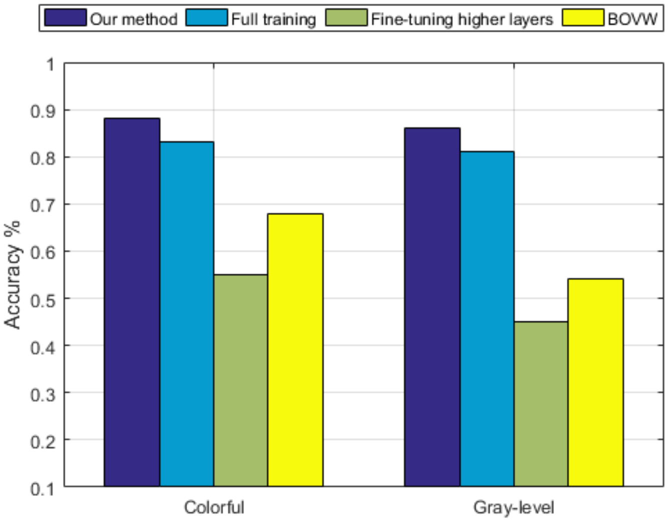

Fig. 6. Comparison of the proposed method with different baselines.

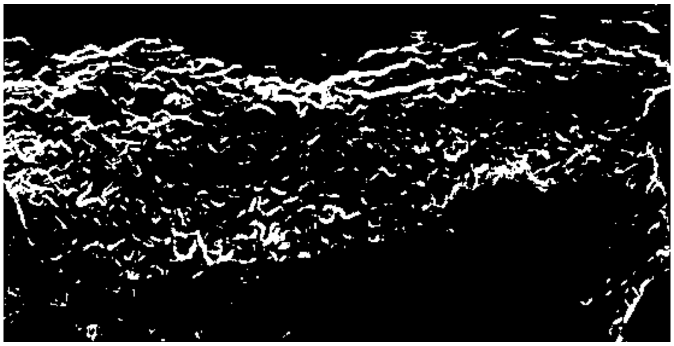

Fig. 7. Result of the traditional edge detection method for a representative SST image.

method provides better results than the BOVW that achieved 68% in the colorful data and 54% in the gray-level data. The results showed that our proposed method achieved the highest accuracy in both data sets, 88% in the colorful data and 86% in the gray-level data.

### D. Validation of the Proposed Method

To validate the effectiveness of the proposed method to an ocean front recognition task, we first used the traditional edge detection method to detect the front using SST images. Based on the result shown in Fig. 7, the images were collected with different scales, with front and no-front annotated by professional oceanographers. Then the evaluation of the final classification was performed using these images as input for our classification method and reached an accuracy of 87%.

Notwithstanding, the wrong classified images were those images with a small scale, which did not present any visually similarity with our training images. However, it is worth mentioning that the images with significantly small scale were not included during the fine-tuning process, but in the final classification performance as input images instead.

## VI. Conclusion

In this letter, we have investigated how to apply deep learning methods to the ocean front recognition task. The proposed method involves CNNs and transfer learning via fine-tuning. We demonstrated the capability of pretrained CNNs, transferred from the ImageNet data set via fine-tuning, to perform an ocean front recognition task, using RS images. In order to better evaluate our method, we conducted our experiments on two different RS image data sets, with different visual properties colorful and gray-level data. In future work we plan to design new deep architectures instead of the CNNs for ocean front recognition tasks.

## References

[1] J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proc. Int. Workshop Workshop Multimedia Inf. Retr.*, Bavaria, Germany, Sep. 2007, pp. 197–206.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[3] K. Nogueira, A. Octavio, B. Penatti, and A. Jefersson. "Towards better exploiting convolutional neural networks for remote sensing scene classification." Unpublished paper, 2016. [Online]. Available: https://arxiv.org/abs/1602.01517

[4] M. Fingas and C. Brown, "Review of oil spill remote sensing," *Spill Sci. Technol. Bull.*, vol. 4, no. 4, pp. 199–208, 2014.

[5] I. M. Belkin and P. Cornillon, "SST fronts of the pacific coastal and marginal seas," *Pacific Oceanogr.*, vol. 1, no. 2, pp. 90–113, 2003.

[6] J. Cayula and P. Cornillon, "Edge detection algorithm for SST Images," *J. Atmos. Ocean. Technol.*, vol. 9, pp. 67–80, Nov. 1992.

[7] P. I. Miller, W. Xu, and M. Carruthers, "Seasonal shelf-sea front mapping using satellite ocean colour and temperature to support development of a marine protected area network," *Deep Sea Res. II, Topical Studies Oceanogr.*, vol. 119, pp. 3–19, Sep. 2015.

[8] Y. Yang, J. Dong, X. Sun, R. Lguensat, M. Jian, and X. Wang, "Ocean front detection from instant remote sensing SST images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1960–1964, Dec. 2016.

[9] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2014, pp. 580–587.

[13] M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon. (Feb. 2016). "Transfer learning from deep features for remote sensing and poverty mapping." [Online]. Available: https://arxiv.org/abs/1510.00098

[14] Z. Ge, C. McCool, C. Sanderson, A. Bewley, Z. Chen, and P. Corke, "Fine-grained bird species recognition via hierarchical subset learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 561–565.

[15] J. Donahue *et al.*, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 647–655.

[16] X. Sun, Y. Liu, J. Li, J. Zhu, H. Chen, and X. Liu, "Feature evaluation and selection with cooperative game theory," *Pattern Recognit.*, vol. 45, no. 8, pp. 2992–3002, 2012.

[17] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.