DOI: 10.1051/itmconf/20171205012

Ship Detection and Classification on Optical Remote Sensing Images Using Deep Learning

Ying LIU[1], Hong-Yuan CUI[1], Zheng KUANG[1], Guo-Qing LI[1]

[1] *School of Computer and Control Engineering, University of Chinese Academy of Sciences, UCAS ,Beijing, China;*
Beijing University of Posts and Telecommunication, BUPT,Beijing, China;
Institute of Remote Sensing and Digital Earth Chinese Academy of Sciences, CAS,Beijing, China
yingliu@ucas.ac.cn, hongyuancui@163.com,kuangzheng2013212987@bupt.edu.cn, ligq@radi.ac.cn

Abstract—Ship detection and classification is critical for national maritime security and national defense. Although some SAR (Synthetic Aperture Radar) image-based ship detection approaches have been proposed and used, they are not able to satisfy the requirement of real-world applications as the number of SAR sensors is limited, the resolution is low, and the revisit cycle is long. As massive optical remote sensing images of high resolution are available, ship detection and classification on theses images is becoming a promising technique, and has attracted great attention on applications including maritime security and traffic control. Some digital image processing methods have been proposed to detect ships in optical remote sensing images, but most of them face difficulty in terms of accuracy, performance and complexity. Recently, an autoencoder-based deep neural network with extreme learning machine was proposed, but it cannot meet the requirement of real-world applications as it only works with simple and small-scaled data sets. Therefore, in this paper, we propose a novel ship detection and classification approach which utilizes deep convolutional neural network (CNN) as the ship classifier. The performance of our proposed ship detection and classification approach was evaluated on a set of images downloaded from Google Earth at the resolution 0.5m. 99% detection accuracy and 95% classification accuracy were achieved. In model training, 75× speedup is achieved on 1 Nvidia Titanx GPU.

1. Introduction

Ship detection and classification in remote sensing images is of vital importance for maritime security and other applications, e.g., traffic surveillance, protection against illegal fisheries and sea pollution monitoring. With the increasing volume of satellite image data, automatic ship detection and classification from remote sensing images is a crucial application for both military and civilian fields. However, the detection systems are faced with the need to process massive amounts of incoming data and the requirement of nearly real-time capacity of reaction. Many valuable studies have been carried out in this field, but these typical algorithms are usually effective only for common image analysis, not for the task of ship detection and classification in remote sensing images which often contains vast data and many background noises. Most of the conventional methods face difficulty in accuracy, performance and complexity.

In recent years, deep learning, or deep neural network has shown great promise in many practical applications. State-of-the-art performance has been reported in several domains, ranging from speech recognition [1], visual object recognition [2] to text processing [3]. In fact, it could be argued that the network's learning ability has been a crucial factor in the recent success of pattern recognition applications. It has also been observed that increasing the scale of deep learning with respect to the number of training examples or the number of model parameters, or both, can drastically improve ultimate classification accuracy [2]. Subsequently, the use of GPUs [1,2,4] is a significant advance in recent years that makes the training of modestly sized deep networks practical. Since the early work of ship detection and classification, it has been known that the variability and the richness of image data make it almost impossible to build an accurate detection and classification system entirely by hand.

In terms of the sparseness of ship distribution on the sea, regions of ship targets are small parts in remote sensing

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

DOI: 10.1051/itmconf/20171205012

images. In this paper, ship candidates are coarsely extracted by image segmentation methods first, then actual ships are detected from all the ship candidates and finally classified into 10 different ship classes by deep learning. The proposed method consists of preprocessing (ship candidates extraction), ship detection and ship classification model training. The specific contributions of this paper are as follows: 1) Cohen-Daubechies-Feauveau 9/7 (CDF 9/7) wavelet coefficients were extracted from the raw images and then ship candidates were extracted from the LL subband by conducting image enhancement, target-background segmentation and ship locating based on shape criteria; 2) a CNN model was implemented for ship detection and 99% accuracy was achieved; 3) for ship classification, using the proposed model, 95% accuracy was achieved; 4) up to 75× speedup was achieved on a server with a GTX Titanx GPU. The flow diagram of the proposed ship detection and classification approach is shown in Figure 1.

The reminder of this paper is organized as follows. Section Ⅱ overviews the related work about this research; section Ⅲ describes the processing of ship candidates extraction; section Ⅳ explains our proposed CNN model for ship detection and classification; section Ⅴ demonstrates the experiments and analysis about the results; section Ⅵ conducts this paper.

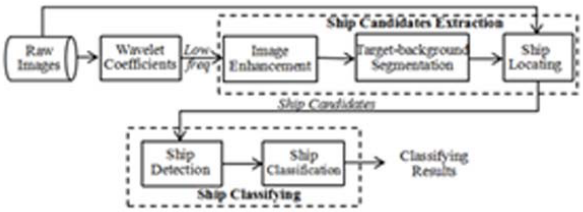

Figure 1. Flow diagram of the proposed ship detection and Classification approach.

2. Related Work

As SAR images have advantages which mainly include relatively little influence of weather and time, ship detection in SAR images has extensively been studied [5~9]. The most common algorithms of ship detection are based on a constant false-alarm rate (CFAR) detector with a certain SAR images' background distribution such as Gauss distribution [6], k-distribution and Gamma distribution [8] or other combination [9]. Han and Chong [5] took a brief review of ship detection algorithms in polarimetric SAR images. Greidanus et al. [7] compared the performance of eight ship detection systems based on spaceborne systems by running a benchmark test on RADARSAT images of various modes. However, ship detection based on SAR has limitations. First, with a limited number of SAR satellites, the revisit cycle is relatively long and it cannot meet the needs of the application of real-time ship monitoring. Second, the resolution of most satellite SAR images is often not high enough to extract detailed ship information.

For ship detection and classification on optical images, traditional methods were widely studied [10~13]. Zhu [10] and Antelo et al. [11] extracted manually designed features from images such as shapes, textures and physical properties while Chen [12] and Wang [13] exploited Dynamic Bayesian Network to classify different kinds of ships, however, they cannot overcome the images' variability and big volume problems. Recently, as the emergence of deep learning architectures, an autoencoder-based deep neural network combined with extreme learning machine was proposed [14] and it outperformed some other methods in detection accuracy. Tang [14] used SPOT-5 spaceborne optical images for ship detecting to verify this idea and achieved relatively satisfying results. However, it has some limitations: 1) as the image resolution is 5m, the extracted features are good enough to detect ships from waves, clouds and islands. But higher resolution is needed for recognizing different types of ships and the model should be expanded to improve its feature-representing ability which will need much more

computation; 2) as autoencoder model uses full connection totally which leads to a large number of nodes and large computation; 3) autoencoder-based deep learning has no concept of local features, so it cannot extract enough detail features when the model is expanded; 4) research indicates that autoencoder-based deep learning is more appropriate for speech recognition [18], hand-writing recognition [19] and texture classification [20]. It only works with simple image data set, so cannot satisfy the requirement of real-world applications.

3. Ship Candidates Extraction

Ship candidates extraction is the first step of our proposed method which preprocesses the raw images and extracts all ship candidates. First of all, CDF 9/7 wavelet coefficients are extracted from images. Instead of reducing processing time by passively cutting an image into tiles or scaling to a low-resolution version, extracting wavelet features can impressively increase the detection efficiency. After a 2-D discrete wavelet transform (DWT), the original image is decomposed into a low-frequency subband (denoted as LL) and horizontal/vertical/diagonal high-frequency subbands (denoted as LH, HL, and HH). The wavelet coefficients in different subbands tend to reflect different properties of the original image. Generally speaking, the low frequency contains most of the global information, while the high frequency represents local or detail information. Ship candidates are extracted from the low-frequency subband LL by conducting image enhancement, target-background segmentation and shape criteria-based ship locating.

DOI: 10.1051/itmconf/20171205012

3.1 Image Enhancement

In image enhancement, in order to remove uneven illumination, a morphological operator, i.e., top-hat transform (THT), is used for ship candidates extraction and background suppression. As ships are usually brighter than their surroundings, the white THT is employed in the proposed work [shown in Figure 2(a)]. The mathematical definition of white THT is as follows [14]:

$$Tw(f) = f - f \circ b \tag{1}$$

where f is the input LL coefficients of the original image, $\circ$ denotes opening operation, and Tw is the enhanced image. In the simulations, b is set as a circular structuring element with a radius of 12.

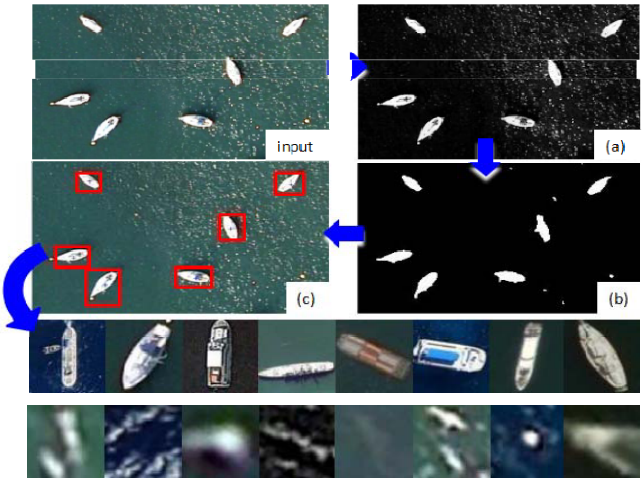

Figure 2. The process of ship candidates extracting.

3.2 Target-Background Segmentation

In target-background segmentation, each input image is binarized by the Otsu algorithm [15]. Otsu is a widely used method to automatically perform clustering-based image thresholding. The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), then it calculates the optimum threshold to separate the two classes so that their combined spread (intra-class variance) is minimal and their inter-class variance is maximal. After that, connected regions are labeled. As the binarized image usually remains small holes in the sea waves or clouds, then the median filtering, morphology dilation and erosion (circular structuring element with a radius of three) are applied to fill the isolated holes. Finally, the masks of sea waves, clouds, islands and ship candidates are segmented [shown in Figure 2(b)]. In the following, ship candidates will be further extracted by using the unique shape properties of ships.

3.3 Ship Locating

In ship locating, the ship candidates are further extracted by using the unique shape properties of ships, including the area, the major minor axis ratio and the compactness [14]. Area equals the number of pixels in the corresponding connected region. Area is used to cut off the clouds, sea waves and other obviously large/small false targets. Major minor axis ratio is defined as

$$R_{ls} = \frac{L_{axisL}}{L_{axisS}} \tag{2}$$

where L_{axisL} and L_{axisS} are the length of long and short axes of the bounding rectangle, respectively. Compactness measures the degree of circular similarity, and it is defined as

$$Compactness = \frac{Perimeter^2}{Area} \tag{3}$$

By using these shape criteria, we can obtain the coarse locations of ship candidates [shown in Figure 2(c)].

In the experiments, the size of the testing images is about 1000×1000 (in pixels) with resolution 0.5m. The size of ship candidates is supported to be larger than 20 in pixels. In this case, the regions with area smaller than 20 would be removed. Moreover, as the long axis of ship should be longer than the minor axis, the major minor axis ratio is selected as 1.2. Compactness is set as 80 to exclude the regions which are obviously irregular. Note that some of the pseudo-targets may be included in the extracted regions; however, they can be removed in the process of ship detection by CNN in Section Ⅳ.

4. Ship Detection and Classification by CNN

Ship detection by deep learning is the next step of our proposed method. It detects actual ships from all the ship candidates and then the actual ships are classified into different types by CNN.

The state-of-the-art ship detection approaches extract features using feature operators or feature descriptors, then use traditional machine learning methods for detection. Features extracted by these methods generally have some fundamental limitations in practical applications. For example, they may have poor performances when the images are corrupted by blur, distortion, or illumination which commonly exist in remote sensing images. So the processed images may contain various pseudo-targets, e.g., islands, clouds, sea waves, etc. Traditional machine learning algorithms, e.g., support vector machine (SVM), may have difficulties in efficiently handling such highly varying inputs. When dealing with highly variant conditions, the

computation is exponentially increased. Relatively, automatically learned features by deep learning from images can help to tackle these issues. Recent works have shown that the features extracted by deep learning outperform those manually designed ones on target detection.

4.1Convolutional Neural Networks

CNN is a kind of deep learning networks and it combines three architectural ideas to ensure some degree of shift, scale and distortion invariance: local receptive fields, shared weights and spatial or temporal sub-sampling. With local receptive fields, neurons can extract elementary visual features such as oriented edges, end-points and corners. These features are then combined by the subsequent layers in order to detect higher-order features. Units in a layer are organized in planes within which all the units share the same set of weights. The set of outputs of the units in such a plane is called a feature map. Units in a feature map are all constrained to perform the same operation on different parts of the image. A complete convolutional layer is composed of several feature maps (with different weight vectors), so that multiple features can be extracted from each location. An implementation of a feature map is equivalent to a convolution, followed by an additive bias and squashing function, hence the name convolutional neuron networks.

Once a feature has been detected, its exact location becomes less important. Only its approximate position relative to other features is relevant. Not only are the precise positions of each of those features irrelevant to identify the pattern, it is potentially harmful because the positions are likely to vary for different instances. A simple way to reduce the precision with which the positions of distinctive features are encoded in a feature map is to reduce the spatial resolution of the feature map. This can be achieved with the so-called pooling layers (or sub-sampling layers) which perform a local averaging or maximizing and a sub-sampling, reducing the resolution of the feature map, and reducing the sensitivity of the output to shifts and distortions. Successive layers of convolution and sub-sampling are typically alternated.

Since all the weights are learned with back-propagation, CNNs can be seen as synthesizing their own feature extractors. The weight-sharing technique can reduce the number of free parameters, thereby reducing the gap between test error and training error.

4.2Training for Ship Detection and Classification

In training, firstly, we solve a two-class (ship and non-ship) classification problem. We constructed a CNN consisting of four convolutional, three max-pooling and a fully-connection layer with a final 2-way softmax classifier for ship detection. Its structure is shown in Figure 3, where 64×64 is the size (in pixels) of each input image with RGB three channels, 5×5, 5×5, 3×3 and 3×3 are the sizes of convolution kernels, 30×30, 13×13 ,11×11 and 4×4 are the sizes of feature maps in each convolutional layer, 64, 128, 256 and 384 are the numbers of feature maps in each convolutional layer, 512 is the number of output units of fully-connection layer and the number '2' at the right end is for the 2-way softmax classifier in detection. After ship detection, all the actual ships are detected. To classify all the ships into 10 different types, this model is used for ship classification by changing the number '2' to '10' at the right end.

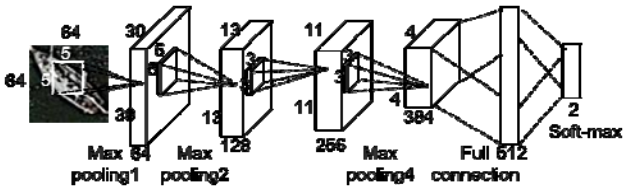

Figure 3. Our proposed CNN model.

The standard way to model a neuron's output f as a function of its input x is with $f(x) = \tanh(x)$ or $f(x) = (1+e^{-x})^{-1}$. In terms of training time with stochastic gradient descent, these saturating nonlinearities are much slower than the non-saturating nonlinearity $f(x) = \max(0, x)$. Following Nair and Hinton [16], we refer to neurons with this nonlinearity as Rectified Linear Units (ReLUs). Deep CNNs with ReLUs train several times faster than their equivalents with tanh units.

In order to reduce test errors, combining the predictions of many different models is a very successful way. But it appears to be too expensive for big neural networks. However, there is a very efficient version of model combination that only costs about a factor of two in training. The recently introduced technique, called 'dropout' [17], works by setting the output of each hidden neuron to zero with probability 0.5. The neurons which are 'dropped out' in this way do not contribute to the forward pass and do not participate in back-propagation. So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights. At test time, we use all the neurons but multiply their outputs by 0.5, which is a reasonable approximation to taking the geometric mean of the predictive distributions produced by the exponentially-many dropout networks. Without dropout, our network exhibits substantial overfitting. Dropout roughly doubles the number of iterations required to converge.

For the other configurations of the network model, we trained our models using stochastic gradient descent with a batch size of 20 examples, momentum of 0.9, an equal learning rate 0.001 for all layers and weight decay of 0.0005.

DOI: 10.1051/itmconf/20171205012

5. Experiments

5.1 Platform Description

We conducted the training on a server with Intel Core i5-4460 CPU @3.20GHz, 8.00GB RAM and GTX TitanX card. Matlab2014a and cuda 7.0 were used.

5.2 Data Sets

Images shown in Figure 4(1) were downloaded from Google Earth and after ship candidates extraction, a dataset consisting 1200 images (containing ships, clouds, sea waves and islands) was obtained and used for performance evaluation each in 64×64 pixels, 4/5 used for network training and 1/5 for testing.

In order to be more convincing, another dataset shown in Figure 4(2), consisting 1500 images (10 categories of ships) each in 256×64 pixels with higher resolution was also downloaded and used for performance evaluation.

Figure 4. Datasets.

5.3 Experiment and Results

An equal learning rate 0.001 is used for all layers, meanwhile, values of the batch size, momentum and weight decay are set as 20, 0.9, 0.0005 correspondingly. 'Dropout' regularization method is used to reduce over fitting in fully-connected layers. This depth of the model seems to be important: we found that removing any convolutional layer resulted in inferior performance. Eventually, 99% detection accuracy and 92% classification accuracy were achieved in dataset (1), which is comparable with some state-of-the-art algorithms, such as SVM. In dataset (2), even higher classification accuracy 95% was achieved, which is closely related to the higher spatial resolution. Classifying error rate of each type of ship by CNN is shown in TABLE Ⅰ. Seventy-five times (75×) speedup was achieved on a GTX TitanX card.

TABLE I. CLASSIFYING ERROR RATE OF EACH TYPE OF SHIP BY CNN ON DATASET (2)

Ship class	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
Error rate	13%	6.5%	3.3%	10%	10%	6.5%	3.3%	0%	3.3%	3.3 %

As a comparison, Support Vector Machine (SVM) and Neural Network (NN) were used for classification on dataset (2), and achieved 87% and 81% accuracy respectively shown in TABLE Ⅱ.

TABLE II. CLASSIFYING ACCURACY ON DATASET (2)

Method	CNN	SVM	Neural Network
Accuracy	95%	87%	81%

6. Conclusion

Ship detection and classification is a widely studied topic both in civilian and military applications. Environmental complexity of marine makes it hard to extract ships from remote sensing optical images both effectively and efficiently. In this paper, we proposed a ship detection and classification method on remote sensing optical images. Firstly, CDF 9/7 wavelet coefficients were extracted from the raw images and the LL subband was used in ship candidates extraction to reduce the processing time. Then ship candidates are extracted by conducting image enhancement, target-background segmentation and ship locating based on shape criteria. Note that there are still nonship targets in the ship candidates, so in the next step a CNN was trained to detect actual ships from all candidates. Finally, using the CNN model, we classified all the actual ships into different ships. The model was trained on remote sensing images downloaded from Google Earth. Eventually, 99% detection accuracy and 95% classification accuracy were achieved, which is comparable with some state-of-the-art algorithms. Experiments showed that CNN, as a deep neural network is a good model for automatically feature learning and extraction. And up to 75× speedup was achieved on a server with a GTX TitanX GPU which indicates its potential for real-time processing.

References

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition," IEEE Transactions on Audio Speech & Language Processing, vol. 20, Jan. 2012, pp. 30-42, doi: 10.1109/TASL.2011.2134090.

[2] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition," CoRR, vol. 22, Nov. 2010, pp. 3207-3220, doi: 10.1162/NECO_a_00052.

[3] R. Collobert and J.Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," International Conference on Machine learning (ICML 08), ACM press, Jul. 2008, pp. 160-167, doi: 10.1145/1390156.1390177.

[4] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale Deep Unsupervised Learning Using Graphics Processors," International Conference on Machine Learning (ICML 09), ACM press, Jun. 2009, pp. 873-880, doi: 10.1145/1553374.1553486.

[5] Z. Y. Han and J. S. Chong, "A Review of Ship Detection Algorithms in Polarimetric SAR Images," International Conference on Signal Processing (ICSP 04), IEEE press, vol. 3, Sept. 2004, pp. 2155-2158, doi: 10.1109/ICOSP.2004.1442203.

[6] K. Eldhuset, "An Automatic Ship and Ship Wake Detection System for Spaceborne SAR Images in Coastal Regions," IEEE Transaction on Geoscience and Remote Sensing, vol. 34, Jul. 1996, pp. 1010-1019, doi: 10.1109/36.508418.

[7] H. Greidanus, P. Clayton, N. Suzuki, and P. Vachon, "Benchmarking Operational SAR Ship Detection," International Geoscience and Remote Sensing Symposium (IGARSS 04), IEEE press, vol. 6, Dec. 2004, pp. 4215-4218, doi: 10.1109/IGARSS.2004.1370065.

[8] C. C. Wackerman, K. S. Friedman, and X. Li, "Automatic Detection of Ships in RADARSAT-1 SAR Imagery," Canadian Journal of Remote Sensing, vol. 27, Jul. 2014, pp. 568-577, doi: 10.1080/07038992.2001.10854896.

[9] D. J. Crisp, "The State of the Art in Ship Detection in Synthetic Aperture Radar Imagery," Organic Letters, vol. 35, May 2004, pp. 2165-2168.

[10] C. Zhu, H. Zhou, R. Wang and J. Guo, "A Novel Hierarchical Method of Ship Detection from Space-borne Optical Image Based on Shape and Texture Features," IEEE Transactions on Geoscience and Remote Sensing, vol. 48, Sept. 2010, pp. 3446-3456, doi: 10.1109/TGRS.2010.2046330.

[11] J. Antelo, G. Ambrosio, and C. Galindo, "Ship Detection and Recognition in High-resolution Satellite Images," International Geoscience and Remote Sensing Symposium (IGARSS 09), IEEE press, vol. 4, Feb. 2010, pp. 514-517, doi: 10.1109/IGARSS.2009.5417426.

[12] H. Chen and X. Gao, "Ship Recognition based on Improved Forwards-backwards Algorithm," International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 09), IEEE press, vol. 5, Dec. 2009, pp. 509-513, doi: 10.1109/FSKD.2009.336.

[13] Q. Wang, X. Gao, and D. Chen, "Pattern Recognition for Ship Based on Bayesian Networks," International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 07), IEEE press, vol. 4, Aug. 2007, pp. 684-688, doi: 10.1109/FSKD.2007.447.

[14] J. Tang, C. Deng, G.H. Huang, and B. Zhao, "Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine," IEEE Transactions on Geoscience and Remote Sensing, vol. 53, Jul. 2014, pp. 1174-1183, doi: 10.1109/TGRS.2014.2335751.

[15] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," 3rd ed. Knoxville: Gatesmark, 2007, pp. 742-745.

[16] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," International Conference on Machine Learning, (ICML 10), Proc icml, Jun. 2010, pp. 807-814.

[17] Krizhevsky A, Sutskever I and Hinton G, "ImageNet classification with deep convolutional neural networks," In NIPS, 2012.

[18] J. Gehring, Y. Miao, and A. Waibel, "Extracting Deep Bottleneck Features Using Stacked Auto-encoders," International Conference on Acoustics, Speech and Signal Processing (ICASSP 13), IEEE press, vol. 32, Oct. 2013, pp. 3377-3381, doi: 10.1109/ICASSP.2013.6638284.

[19] P. Vincent, H. Larochelle, and Y. Bengio, "Extracting and Composing Robust Features with Denoising Autoencoders," International Conference on Machine Learning (ICML 08), ACM press, Jul. 2008, pp. 1096-1103, doi: 10.1145/1390156.1390294.

[20] M. Chen and Z. Xu, "Marginalized Denoising Autoencoders for Domain Adaptation," International Conference on Machine Learning (ICML 12), Computer Science, 2012.