Semantic Segmentation of Aerial Images with Shuffling Convolutional Neural Networks

Kaiqiang Chen, Kun Fu, *Member, IEEE,* Menglong Yan, *Member, IEEE,* Xin Gao, Xian Sun, *Member, IEEE,* and Xin Wei

***Abstract*—Semantic segmentation of aerial images refers to assigning one land cover category to each pixel. This is a challenging task due to the great differences in the appearances of ground objects. Many attempts have been made during the past decades. In recent years, convolutional neural networks (CNNs) have been introduced in the remote sensing field, and various solutions have been proposed to realize dense semantic labeling with CNNs. In this letter, we propose shuffling convolutional neural networks (SCNNs) to realize semantic segmentation of aerial images in a periodic shuffling manner. This approach is a supplement to current methods for semantic segmentation of aerial images. We propose a naive version and a deeper version of this method, and both are adept at detecting small objects. Additionally, we propose a method called FoV-Enhancement that can enhance the predictions. This method can be applied to various networks, and our experiments verify its effectiveness. The final results are further improved through an ensemble method that averages the score maps generated by models at different checkpoints of the same network. We evaluate our models using the ISPRS Vaihingen and Potsdam datasets, and we acquire promising results using these two datasets.**

***Index Terms*—Convolutional neural networks, Semantic segmentation, Aerial images, Remote sensing, Deep learning**

I. Introduction

Semantic segmentation of aerial images requires assigning one land cover category to each pixel. Conventional methods that are dependent upon hand-crafted features fail to reach state-of-the-art performances and are limited by the representation capacity of features. Ever since the work of [9] won the first prize in the ILSVRC-2012 contest [14], convolutional neural networks(CNNs) have been the most eminent methods in computer vision. Based on CNNs, a huge number of algorithms have been proposed during the last several years in order to solve challenging computer vision tasks such as image classification, object detection, super resolution and semantic segmentation.

In recent years, numerous attempts have been made to introduce convolutional neural networks into the field of remote sensing to tackle all kinds of tough missions. [4] simultaneously detects vehicles at the pixel level and classifies them into 9 categories. [7] extracts buildings through a combination of two deep deconvolution networks. In a similar way, [12] implements semantic segmentation of aerial images through an assembly of two identical parallel FCNs. [6] proposes rotation-invariant convolutional neural networks to advance the performance of object detection, which is achieved by introducing a rotation-invariant layer and a novel objective function. [18] proposes a unified annotation framework by combining discriminative high-level feature learning and weakly supervised feature transferring. [5] provides a comprehensive review of remote sensing image scene classification techniques and proposes a large-scale dataset. [17] designs three kinds of networks to realize dense semantic labeling of aerial images and proposes a dense prediction based on deconvolution (FPL) after comparing the performances of these networks. [15] proposes no-downsampling convolutional neural networks, which are extremely computationally expensive and require tremendous GPU memory resources.

The contributions of this letter can be summarized as follows. (1) We introduce the shuffling operator into semantic segmentation of aerial images, upon which we propose two networks. This approach is a supplement to current methods for semantic segmentation of aerial images. Specifically, these networks are adept at detecting small objects like cars. Compared with the networks in [17] and [15], our networks are more effective and more efficient. (2) We make a further study of how atrous spatial pyramid pooling (ASPP) [2] acts in shuffling convolutional neural networks (SCNNs) through experiments. The results are listed in the tables and analyzed in Section IV. (3) We propose a method called FoV-Enhancement in Section II-B that can enhance predictions. This method can be applied to various networks, and our experiments can verify its effectiveness. The final results are further improved through an ensemble method, which averages the score maps generated by models at different checkpoints of the same network.

II. Methods

A. Shuffling operator

The shuffling operator converts downsampled feature maps, which are generated by convolutional neural networks, into high-resolution feature maps. Considering a network, which is fed with patches of size $H \times W$, it generates feature maps of $c \times \frac{H}{r} \times \frac{W}{r}$, where c is the number of classes and r is the sampling rate. To generate predictions of the original size, one solution is interpretation, which is what [1] do. Alternatively, inspired by the work of [16], we can force the network to generate feature maps of $c \times (s \times r)^2$ channels instead of only c channels, where s is the scale factor, which means that we

This work was supported by the National Natural Science Foundation of China under Grant 41301493.

The authors are with the Institude of Electronics and the Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Chinese Academy of Sciences, Beijing 100190, China. Kaiqiang Chen and Xin Wei are also with the University of Chinese Academy of Sciences. (e-mail: chenkaiqiang14@mails.ucas.ac.cn)

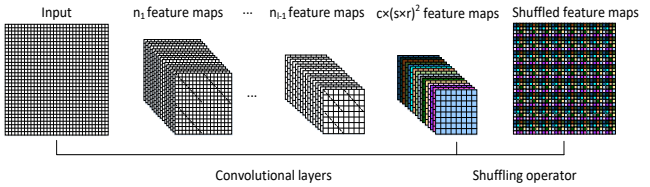


Fig. 1. An intuitive understanding of shuffling convolutional neural networks. The previous layers are common convolutional layers, and the last layer is a shuffling operator that converts $c \times (s \times r)^2$ feature maps of size $(\frac{H}{r}) \times (\frac{W}{r})$ into c feature maps of size $(H \times s) \times (W \times s)$. In this figure, $H = 32, W = 32, r = 4, s = 1, c = 1$.

want predictions of size $(H \times s) \times (W \times s)$. Hence, its feature maps are of size $\frac{H}{r} \times \frac{W}{r} \times c \times (s \times r)^2$. After periodic shuffling, the final feature maps are of size $(\frac{H}{r} \times r \times s) \times (\frac{W}{r} \times s \times r) \times c$. That is $(H \times s) \times (W \times s) \times c$. This periodic shuffling can be written as

$$I^{shuffled}(c_i, x, y) = I^{before}((c_i - 1) \times (sr)^2 + mod(x, sr) + mod(y, sr) \times sr, \lfloor \frac{x}{sr} \rfloor, \lfloor \frac{y}{sr} \rfloor), \quad (1)$$

where c_i should comply with $1 \leq c_i \leq c$. Intuitively, it can be seen in Fig.1.

Applying softmax to $I^{shuffled}(c_i, x, y)$ will generate score maps $I^{score}(c_i, x, y)$. If $s \neq 1$, the score maps are upsampled, noted as $I^{up\_score}(c_i, x, y)$, to match the original image using bilinear interpolation. The final predictions are

$$I^{pred}(x, y) = \arg\max_{c_i} I^{up\_score}(c_i, x, y). \quad (2)$$

B. Field of View Enhancement

Field of view plays a significant role in prediction. A field of view that is too small may lead to inferior results due to the lack of contextual information. For example, when patches only cover a part of buildings or roads and there is no other object in the patches, only textural information is provided. Due to the lack of sufficient information, it is difficult to distinguish objects with similar textural features.

Predictions of marginal pixels (which are near the edges of patches) are sometimes unreliable. Compared with pixels at the center of patches, marginal pixels have smaller fields of view. Hence, the limited field of view of the marginal pixels may result in inaccurate predictions.

Based on these two assumptions, we propose a method called FoV-Enhancement that can improve accuracy.

This method should be adopted when making inferences. First, the sizes of the patches fed into networks are enlarged when making inferences. Second, the patches are cropped from images in an overlapping manner. There are many methods for handling overlapping areas. In this letter, the final prediction of a pixel in an overlapping area is determined by the nearest patch. Though it is simple, it is effective.

C. Ensemble methods

Ensemble methods are widely used to enhance performance. Some algorithms assemble different algorithms with complementary characteristics. For instance, [13], which is adept at capturing the fine details of an object, fuses its own results with those of FCN [10], which is good at extracting the overall shape of an instance. Some methods may aggregate models trained with different parameters or various parts of the dataset. All these methods require training models at the beginning of the process.

In this letter, instead of training models at the beginning, we average the score maps generated by the models at different checkpoints of the same network. This ensemble method is efficient as there is no need to train an entirely new model from scratch, which is always extremely time-consuming. Our experiments demonstrate the effectiveness of this approach, as will be shown in Section IV-D5.

III. Shuffling convolutional neural networks

A. Rectified Deeplab Model

All our models are based on Deeplab Model [1], and we modified this model to serve as our baseline model, which we call Rectified Deeplab Model (RDM for short). The strides of the convolution layers right before pooling layers in [1] are changed to 2 to take responsibility for downsampling. And the pooling layers in [1] are all removed. One batch normalization [8] layer is appended right after each convolution layer. The other settings are the same as those in [1]. Another baseline model is the RDM-ASPP, which inserts an ASPP [1] into RDM as [1] does.

B. Naive Shuffling Convolutional Neural Network

The first shuffling convolutional neural network we propose is called the naive shuffling convolutional neural network (Naive-SCNN) as it only inserts a shuffling layer between the last convolution layer and $softmax$ in the RDM. All other layers are the same as in the RDM. Compared with the RDM, the only extra cost of this structure is that it learns more weights in in the last convolution layer. Intuitively, it can be seen in Fig.1. Its corresponding ASPP version (Naive-SCNN-ASPP) can be constructed by inserting a shuffling layer between the ASPP and $softmax$ in the RDM-ASPP. The number of learnable parameter increases about 17.59M after adding ASPP. This number is got as $(3 \times 3 \times 512 \times 1024 + 1 \times 1 \times 1024 \times 1024 + 1 \times 1 \times 1024 \times 6 \times 16) \times 3$.

C. Deeper Shuffling Convolutional Neural Network

Inspired by the idea that deeper structures are beneficial in enhancing accuracy, we make many attempts to increase the depth of the Naive-SCNN. To distinguish this network from the Naive-SCNN, we call it the Deeper-SCNN. Fifteen extra convolutional layers, which are followed by batch normalization [8] layers and ReLU layers, are appended after the last convolution layer in the Naive-SCNN. The numbers of learnable parameters for the different networks are listed in Table I.

IV. Experiments and analysis

A. Dataset

Our models are evaluated using the Vaihingen set and the Potsdam set, which are provided in the ISPRS 2D semantic

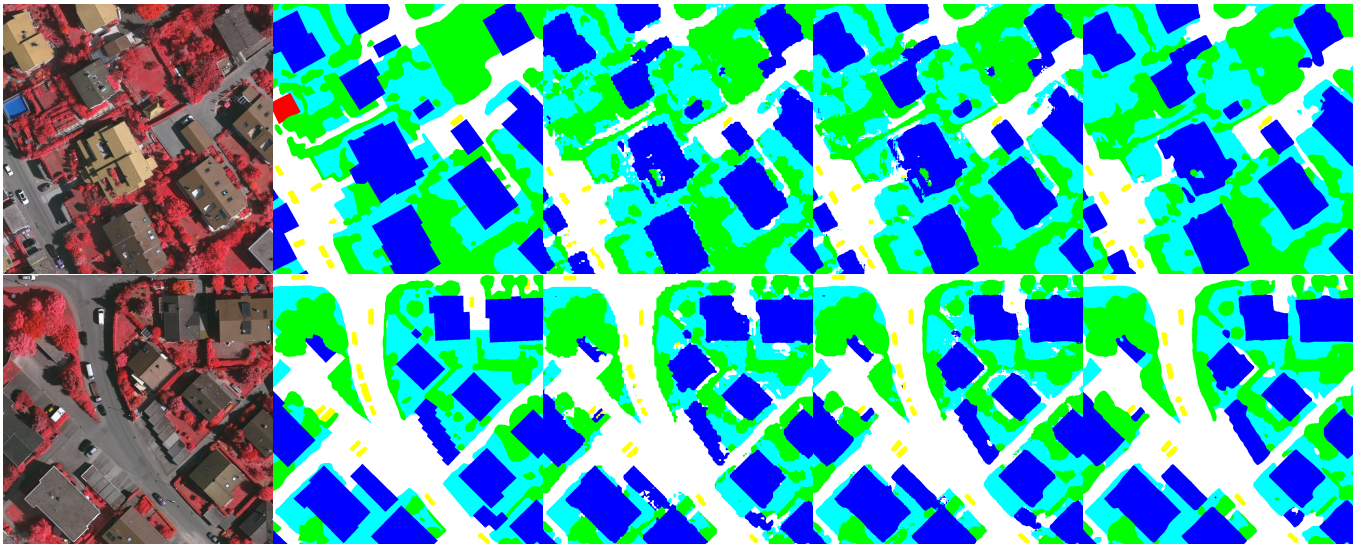


Fig. 2. Intuitive results of different networks. The five figures in each row correspond to an aerial image, the ground truth and predictions of the RDM, the Naive-SCNN and the Deeper-SCNN, respectively. From these figures, it can be found that predictions of the Deeper-SCNN are much smoother than those of the RDM and the Naive-SCNN.

TABLE I
LEARNABLE PARAMETERS FOR DIFFERENT NETWORKS

Network	Parameters
RDM	20.48M
RDM-ASPP	37.80M
Naive-SCNN	20.58M
Naive-SCNN-ASPP	38.17M
Deeper-SCNN	41.15M
FPL [17]	7.04M
NDFCN [15]	3.24M

segmentation contest[1]. The Vaihingen set contains 33 aerial images, of which 16 images are provided with full annotations. Eleven out of the 16 annotated images comprise the training set, and the remaining 5 (11, 15, 28, 30, 34) are used to evaluate the different networks. Each image consists of near-infrared, red and green channels. In addition, extra digital surface models (DSMs) are supplied. [11] provides normalized digital surface models (nDSMs). In our experiments, we feed patches of five bands, including DSMs, near-infrared, red, green and nDSMs, into the networks. The Potsdam set contains 38 aerial images of size 6000×6000, among which 24 images are fully annotated. Seventeen out of the 24 annotated images comprise the training set, and the remaining 7 (2_11, 2_12, 4_10, 5_11, 6_7,7_10,7_8) are used to evaluate the different networks. Each image in this set is provided with near-infrared, red, green and blue bands. Additionally, DSMs and nDSMs are offered. We make full use of all these data, and each patch fed into the networks contains six bands.

B. Experimental settings

We train and evaluate our models using MXNET [3] on NVIDIA TELSA K80 GPUs. The loss is the cross-entropy error, which is summed over all the pixels in a batch. To optimize this objective function, we use the standard SGD with momentum.

[1]http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

When training networks, samples in each batch are 224×224 in size. In contrast with the common strategy that prepares patches beforehand, and in which patches are regularly cropped from original large images and then saved on hard-disks before training, in our experiments, each sample is cropped randomly and temporarily. It saves storage and generates more samples in theory.

The RDM's predictions are only $1/8$ as large as the patches fed into the network. To generate predictions that match the input, the score maps generated by the RDM are rescaled using bilinear interpolation. Additionally, in our experiments, we set the scale factor s in Eq.1 to $1/2$, and the score maps generated by the Naive-SCNNs and the Deeper-SCNNs are rescaled to match the patches. The reason why we adopt $1/2$ is based on an assumption that the ground truth cannot be labeled very accurately. The effect of s can be seen in Fig.3, and we find that $s = 1/2$ is a good choice.

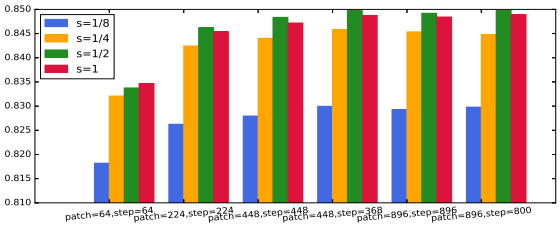


Fig. 3. The effect of s in Eq.1. In this experiment, we use Naive-SCNN on Vaihingen Set. It can be found that $s = 1/2$ is a good choice.

C. Effect of Field of View

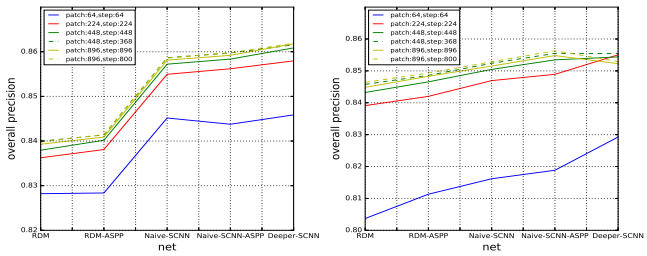
Field of view will affect accuracy, and FoV-Enhancement helps improve accuracy. We explored the effect of field of view on both the Vaihingen set and the Potsdam set, and the results can be seen in Fig.4. In our experiments, we use patches of 64,224,448 and 896. In general, the accuracy improves after enlarging the field of view. It is because that a field of view that is too small may lead to inferior results due to the lack of contextual information and edge information. Also, patches of 448 and 896 are explored further. We use

TABLE II
EXPERIMENTS ON THE VAIHINGEN DATA SET

Method	Overall Precision	Impervious Surfaces	Building	Low Vegetation	Tree	Car
RDM	0.8400	0.8626	0.9186	0.7416	0.8357	0.6245
RDM-ASPP	0.8414	0.8657	0.9197	0.7428	0.8354	0.6339
Naive-SCNN	0.8587	0.8845	0.9356	0.7610	0.8497	0.7371
Naive-SCNN-ASPP	0.8597	0.8859	0.9357	0.7623	0.8511	0.7419
Deeper-SCNN	0.8615	0.8890	0.9363	0.7683	0.8519	**0.7741**
EDeeper-SCNN	**0.8623**	**0.8902**	**0.9369**	**0.7688**	**0.8521**	0.7738
FPL [17]	0.8470	0.8705	0.9175	0.7376	0.8487	0.7484
NDFCN [15]	0.8460	0.8696	0.9193	0.7425	0.8459	0.7268

steps of 368 and 448 for patches of 448 and steps of 800 and 896 for patches of 896. It can be found that cropping patches in an overlapping way always enhances the prediction accuracy because the predictions of marginal pixels are limited by their field of view. The experiments show that the FoV-Enhancement works.

The images in both test sets are large in size. For example, the images in the Vaihingen set are approximately 1800×2500, and those in the Potsdam set are 8000×8000. None of these images can be fed into the networks directly. Hence, when making inferences, these large images are cropped into patches, and the patches are fed into networks one by one. In the following experiments, these patches are 448×448, which is a different size from those that were cropped during the training, and they are cropped in an overlapping way with a step of 368. These two parameters are set empirically and not optimally, but this setting is always effective in our experiments. Finally, the predictions of these patches are recombined into complete large maps, upon which our final evaluations are made. All the following experiments adopt this configuration without additional explanation.



(a) Experiments on Vaihingen Set (b) Experiments on Potsdam Set
Fig. 4. Through altering the size of patches and overlapping areas, these two figures reveal that FoV-Enhancement is beneficial in improving accuracy.

D. Results and analysis

1) Analysis of SCNN: Table II shows the numerical results for the Vaihingen set. The metric for every category is the F1-score, which is defined as the harmonic mean of precision and recall. Because the number of pixels belonging to the clutter in the Vaihingen set is very small, this category is ignored during training. However, when calculating the metric of overall precision, they are still taken into account.

Among all the metrics, SCNNs outperform the RDM and the RDM-ASPP to a large extent. One of the most important reasons is that SCNNs learn to upsample, while RDMs employ only bilinear interpolation. Compared with RDM, the Naive-SCNN owns comparable learnable weights, and the extra cost can be ignored. Compared with the RDM-ASPP, the Naive-SCNN owns fewer learnable weights while achieving better performance. SCNNs outperform FPL [17] and NDFCN [15]. The performances of FPL and NDFCN are limited by the structures of the networks, which fail to learn more distinguishing features. The experiments on the Potsdam set reveal similar conclusions, as can be seen in Table III.

When the network goes deeper, we get more promising results. It is because that the deeper networks can learn more distinguishing features.

Qualitative results are presented in Fig.2. The results of the RDM are not smooth at all and are weak on the boundaries. This result may be caused by bilinear interpolation. Through learning upsampling, the Naive-SCNN alleviates this phenomenon, but there is still some splattering. With the Deeper-SCNN, we get smooth results.

2) Analysis of ASPP: ASPP does help improve performance of both the RDM and the Naive-SCNN. For the Vaihingen set, after employing ASPP, the overall accuracy of the RDM improves by 0.14%, and that of the Naive-SCNN improves by 0.1%. For the Potsdam set, after employing ASPP, the overall accuracy of the RDM improves by 0.28% and that of the Naive-SCNN improves by 0.31%.

Additionally, we have tried to introduce ASPP to our Deeper-SCNN, but we failed to make this network converge well. This is because simply appending extra convolutional layers does not work. We believe that the position of the ASPP does matter.

3) Small Objects: SCNNs are adept at detecting small objects like cars. For the Vaihingen set, SCNNs outperform RDMs by more than 10%, and for the Potsdam set, SCNNs outperform RDMs by more than 6%.

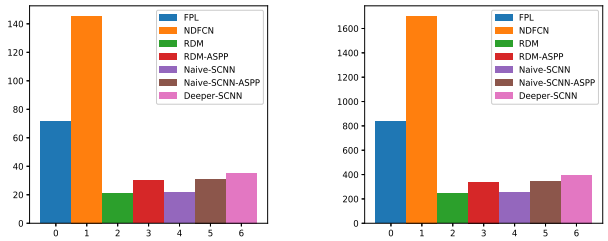
In detecting small objects like cars, the improvement offered by ASPP is limited. The F-Score of cars for the Vaihingen set is improved by only 0.84% for the RDM and by 0.48% for the Naive-SCNN. For the Potsdam set, the F-Score of cars is improved by only 0.33% for the RDM and by 0.51% for the Naive-SCNN. Though ASPP is designed to handle scale variability, its improvement in detecting small objects is not as remarkable as that of SCNNs and that of increasing the depth. After employing the shuffling operator, the F-score of cars is improved by 11.26% for the Vaihingen set and by 6.97% for the Potsdam set. After increasing the depth of the Naive-SCNN, the score is improved by 3.22% and 1.05% for the Vaihingen and Potsdam datasets, respectively.

4) Speed: Compared with networks in [17] and [15], SCNNs are much more efficient. The time costs of the different networks for both sets are shown in Fig.5. RDM is the most efficient, and Naive-SCNN is as efficient as RDM.

TABLE III
EXPERIMENTS ON THE POTSDAM DATASET

Method	Overall Precision	Impervious Surfaces	Building	Low Vegetation	Tree	Car	Clutter
RDM	0.8458	0.8693	0.9274	0.8300	0.7843	0.8098	0.7148
RDM-ASPP	0.8486	0.8719	0.9273	0.8335	0.7903	0.8131	**0.7167**
Naive-SCNN	0.8524	0.8767	0.9339	0.8368	0.7875	0.8795	0.7134
Naive-SCNN-ASPP	0.8555	0.8825	0.9348	0.8392	0.7916	0.8846	0.7138
Deeper-SCNN	0.8554	0.8885	0.9346	0.8421	0.7894	0.8951	0.6755
EDeeper-SCNN	**0.8578**	**0.8896**	**0.9363**	**0.8436**	**0.7944**	**0.8967**	0.6834
FPL [17]	0.8361	0.8612	0.9266	0.8174	0.7735	0.8632	0.6807
NDFCN [15]	0.8299	0.8599	0.9172	0.8094	0.7637	0.8582	0.6662

Naive-SCNN is approximately 3 times faster than FPL and approximately 7 times faster than NDFCN. After appending ASPP, it is still much faster than FPL and NDFCN.



(a) Experiments on Vaihingen Set (b) Experiments on Potsdam Set

Fig. 5. This figure shows the time costs of running the different networks for both sets.

5) Analysis of our ensemble method: An ensemble of models of different checkpoints contributes to improving accuracy. This improvement can be seen from both Table II and Table III. The ensemble is denoted as EDeeper-SCNN in Table II and Table III. After applying the ensemble method, the overall precision improves by 0.08% for the Vaihingen set and by 0.24% for the Potsdam set. All other entries are improved with one exception.

V. CONCLUSION

In this letter, we introduce the shuffling operator in semantic segmentation of aerial images. Based on this operator, we propose two shuffling convolutional neural networks, the Naive-SCNN and the Deeper-SCNN. Extensive experiments are designed to prove the effectiveness of the SCNNs. We evaluate our models using two publicly available data sets. All the experiments reveal that SCNNs outperform our baseline models to a large extent and that they are apt at detecting small objects like cars. Also, we propose the FoV-Enhancement method to enhance our predictions. This method can be applied to various networks, and extensive experiments have revealed the effectiveness of this method. Further, we explore how the ASPP acts in SCNNs, and experiments show that the improvement of ASPP is very limited. Lastly, we further improve model performance using an ensemble of models at different checkpoints.

REFERENCES

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016.

[3] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015.

[4] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 11(10):1797–1801, Oct 2014.

[5] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, PP(99):1–19, 2017.

[6] G. Cheng, P. Zhou, and J. Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, Dec 2016.

[7] Z. Huang, G. Cheng, H. Wang, H. Li, L. Shi, and C. Pan. Building extraction from multi-source remote sensing images via deep deconvolution neural networks. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1835–1838, July 2016.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[11] ITC Markus Gerke. Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen).

[12] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. Semantic segmentation of aerial images with an ensemble of cnns. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:473–480, 2016.

[13] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[15] Jamie Sherrah. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *arXiv preprint arXiv:1606.02585*, 2016.

[16] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[17] Michele Volpi and Devis Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, 2017.

[18] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo. Semantic annotation of high-resolution satellite images via weakly supervised learning. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3660–3671, June 2016.