Improving recognition of complex aerial scenes using a deep weakly supervised learning paradigm

Praveer Singh, Nikos Komodakis

Improving recognition of complex aerial scenes using a deep weakly supervised learning paradigm

Praveer Singh , Nikos Komodakis

***Abstract*—Categorizing highly complex aerial scenes is quite strenuous due to the presence of detailed information with large number of distinctive objects. Recognition happens by first deriving a joint relationship within all these distinguishing objects, distilling finally to some meaningful knowledge that is subsequently employed to label the scene. However, something intriguing is whether all this captured information is actually relevant to classify such a complex scene ? What if some objects just create uncertainty with respect to the target label, thereby causing ambiguity in the decision making ? In this paper, we investigate these questions and analyze as to which regions in an aerial scene are the most relevant and which are inhibiting in determining the image label accurately. However, for such Aerial Scene Classification (ASC) task, employing supervised knowledge of experts to annotate these discriminative regions is quite costly and laborious; especially when the dataset is huge. To this end, we propose a Deep Weakly Supervised Learning (DWSL) technique. Our classification-trained Convolutional Neural Network (CNN) learns to identify discriminative region localizations in an aerial scene *solely* by utilizing image labels. Using the DWSL model, we significantly improve the recognition accuracies of highly complex scenes, thus validating that extra information causes uncertainty in decision making. Moreover, our DWSL methodology can also be leveraged as a novel tool for concrete visualization of the most informative regions relevant to accurately classify an aerial scene. Lastly our proposed framework yields state-of-the-art performance on existing ASC datasets.**

***Index Terms*—Deep Learning, Aerial Scene Classification, Scene Complexity, Weakly Supervised, Multi-Instance Learning.**

I. Introduction

Scene Classification is of paramount importance in remote sensing community to automatically categorize images for further scrutiny. These images are later on utilized by expert annotators for varied roles, including detection or segmentation of objects of interests.

One characteristic feature that holds Remote Sensing (RS) scenes distinctively apart from natural scenes is the widespread scale at which the area of interest is captured. Consequently, the extent of distinctive objects captured in a RS scene are also quite large. One of the challenges encompassing RS scene recognition is the added complexity when these discriminative objects are present simultaneously in multiple scenes. For *e.g.*, in Fig. 1, a scene of Mountain (middle) has distinctive patches of grasslands (similar to Meadow on left) however the most representative parts of the scene are the white snowy tracts as would be illustrated in section IV.

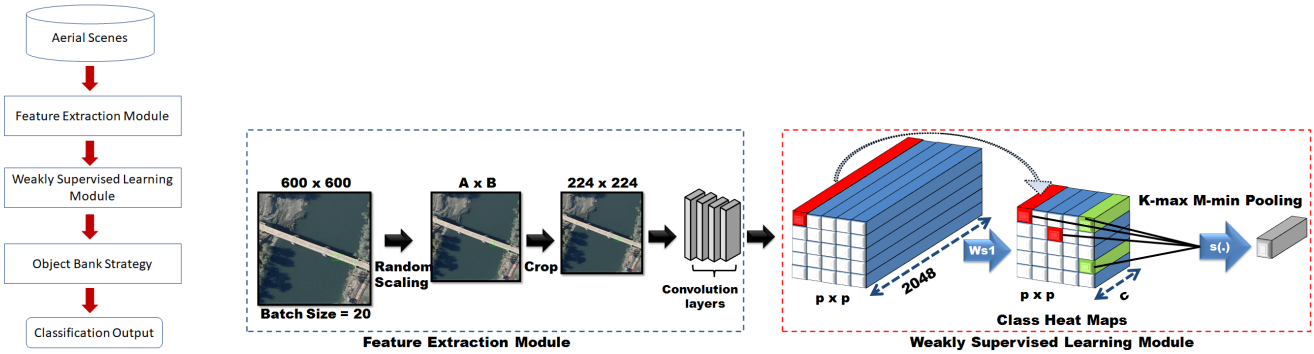
Recently deep learning models have shown significant improvement in performance for varied remote sensing tasks such as aerial scene segmentation [1, 2], hyper-spectral classification [3] or change detection [4]. Inspired by architectures of the primate visual cortex [5], these models interpret simultaneously various complex concepts which were lacking in conventional hand crafted methods [6]. [7] defined scene complexity, based upon how much attention a person devotes to understand a particular scene. For *e.g.*, in Fig. 1, a low complexity scene of Meadow is easy to interpret while comparatively more complex scenes of Mountain or Square have much more detailed information that needs to be distilled effectively and hence requires fairly larger period of attention.

Fig. 1: Scenes representing various levels of complexity.

However, instead of pivoting simply on greater attention for more complex scenes [7], we rather propose an alternate strategy. We argue that by limiting the amount of information gathered from a scene, we tend to minimize the ambiguity at the time of knowledge distillation and thus yield higher performance. Earlier we had seen that the amount of information to be processed differs from scene to scene, with highly complex aerial scenes exhibiting more number of distinctive regions. We postulate that only some of these regions are relevant for characterizing an aerial scene while the other regions are either irrelevant or inhibit in the recognition performance. This is specifically pertinent in the case of more complex scenes where the number of distinctive regions are significantly large and often lead to confusion in the final decision of a network. We, therefore, propose to remove this ambiguity using a novel methodology which allows the network to learn to choose the most and least relevant regions in a scene that aids in improving the overall recognition accuracy.

Nevertheless, selecting important discriminative regions in aerial scenes is a tedious task. This is mainly because images are of very high resolution and require an expert annotator. We choose to automatize the task of selecting relevant regions by introducing an end-to-end deep weakly supervised learning model in the context of aerial scene classification. Precisely, we make the network learn to select the most relevant regions

The authors are affiliated to École des Ponts ParisTech & Université Paris Est, France (e-mail: {firstname.lastname}@enpc.fr).



(a) Pipeline

(b) Deep Weakly Supervised Learning (DWSL) network architecture

Fig. 2: Overall Pipeline and Network Architecture

in a scene that can predict the scene label with higher accuracy. Since we leverage only the class labels for localizing important regions in our aerial scene without object bounding box annotations, we call it weakly supervised. With further investigation, we also empirically showcase that RS scene recognition using this weakly supervised paradigm is much more beneficial for more complex scenes, thus substantiating our previous hypothesis of ambiguity removal.

Instances	% OA Resnet-DWSL (without object bank)
k=3,m=0	92.56 $\pm$ 0.22
k=3,m=3	**93.02 $\pm$ 0.57**

TABLE I: Accuracies without(m=0) & with(m=3) negative instances.

In a way, our proposed methodology is a relaxation of the prominent Multiple Instance Learning (MIL) [8] technique where an image is characterized as a bag of instances (region). These bags or images are assigned a label: positive if it contains at least one positive instance and negative if all the instances are negative (Negative Instances in Negative bag or NiN). NiN is a fairly strong assumption simply because for *e.g.*, absence of tennis court label doesn't imply that it is absent in the actual scene as seen in Square scene (rightmost in Fig. 1). It simply implies that the person who labeled the scene based his judgment upon the most predominant region in the scene. Thus, relaxing on the NiN presumption, our network tries to maximize the prediction of the correct class labels by utilizing both the top-K regions (positive instances) and bottom-M regions (negative instances).

Inspired from [9], we showcase our own mechanism to visualize the prediction scores by highlighting the most discriminative regions which help to correctly classify any aerial scene. In a way, it furnishes us with a nice interactive tool to visualize the most important regions in a scene as simple as in a single forward pass through our network.

Lastly, we test our proposed methodology on all the existing aerial scene classification datasets [6, 10, 11] as well as on a fairly new and challenging one [12], utilizing two prominent deep learning architectures namely Vgg-16 and Resnet-101. Experimental results clearly exhibit our Deep Weakly Supervised Learning (DWSL) method giving state-of-the-art (SoA) results both with Resnet-101 and Vgg-16 architecture. We also note that Resnet performs notably better than Vgg primarily because of preservation of spatial information throughout the network.

Fig. 3: Scenes from playground and stadium categories

In a nutshell,

1) We propose the first deep weakly supervised learning technique for aerial scene classification that automatically localizes most prominent regions using scene labels.
2) Our model removes ambiguity in complex aerial scenes by selecting only the most important objects out of a large number of distinctive objects that usually confuse the network in overall decision making.
3) We outperform SoA scene classification methods using only few relevant regions thus eradicating the need of utilizing entire scene as done in the past.
4) We present a compact tool for visualizing the most discriminative regions, thus giving us a better understanding of where network focuses upon while classifying a scene.

II. Proposed Framework and Methodology

As illustrated in Fig. 2a, our proposed Deep Weakly Supervised Learning (DWSL) pipeline consists of 3 major steps:

1) Feature Extraction using a VGG16 or Resnet-101 deep learning architecture.
2) Weakly Supervised Learning to select the most discriminative regions and jointly pooling them.
3) Object Bank Strategy to concatenate features from multiple scales and training an SVM classifier on top for ASC.

We discuss each of the steps in more detail in following subsections.

A. Feature Extraction Module

Feature extraction module is employed to compute a fixed-size feature representation from the input image. We first perform a random scaling of the input image to a size between

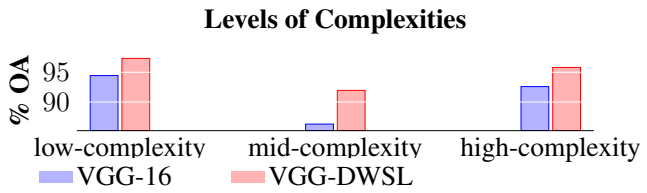


Fig. 4: Overall Accuracies (OA) for different levels of complexities

[224, 244]. This is followed by cropping the image to a fixed size of 224×224. We perform several other data-augmentation strategies like horizontal and vertical flip or rotation by a small angle theta to avoid over-fitting. The resulting image is then passed through the convolution layers of either Resnet-101 or Vgg-16. In Resnet-101 model, the final convolution layer results in an output of size $2048 \times 7 \times 7$ which can be treated as the first block of the Weakly Supervised Learning (WSL) Module (see Fig. 2b). In case of Vgg-16, the output from the final convolutional layer is of size $512 \times 14 \times 14$ which is then fed to WSL module as input.

Dataset	Arch.	Baseline	DWSL	% Gains
AID [12]	VGG	91.33 ±0.46	95.06 ±0.35	**4.11**
AID [12]	Resnet-101	95.44 ±0.26	96.96 ±0.34	**1.52**
WHURS19 [11]	Resnet-101	96.94 ±0.92	98.67 ±0.61	**1.73**
RSSCN7 [10]	Resnet-101	94.48 ±0.33	96.19 ±0.50	**1.71**
UCMercred [6]	Resnet-101	97.24 ±0.35	97.81 ±0.26	**0.57**

TABLE II: Results comparing baseline with our *DWSL* method

Motivated from [13], our WSL Module consists of two major components: Ws1 and K-max M-min Pooling (represented as s(.) in Fig. 2b).The Ws1 layer is similar to the fully connected layers and is often witnessed as terminal layers in most deep learning architectures. The only difference is that here we have used them in the form of multiple fully convolutional layers each of size 1×1. Further, these layers are applied individually to each of the $p \times p$ cells (as shown by the curved arrow) of the previous block. This operation yields us a $p \times p$ block but with a depth size equal to the class number c.

Now, we treat the new output block as stacked heatmaps for each individual class with dimension of $c \times 7 \times 7$ (corresponding to an input image of size 224). These class heatmaps are then fed to a K-max M-min Pooling layer $s(.)$ which (a) selects the top-K scoring activations in a particular class heatmap (highlighted by green cells in Fig. 2b, (b) caters to lowest-M scoring activations in the heatmap (red cells).

B. Weakly Supervised Learning Module

Both top and lowest scoring regions are projected onto the original image and visualized as green and red bounding boxes in Fig. 7. Finally, these cells are aggregated individually for each of the c class heatmaps using $s(c)$ which is given as: $s(c) = s_{top}(c) + s_{bottom}(c)$. The components $s_{top}(c)$ and $s_{bottom}(c)$ are formulated as: $s_{top}(c) = \sum_{k=1}^{K} t_k(c)$ and $s_{bottom}(c) = \sum_{m=1}^{M} l_m(c)$. The $t_k(c)$ represents the k^{th} top-scoring activation value for c^{th} class heatmap. Similarly, $l_m(c)$ is the m^{th} lowest-scoring activation value for the same c^{th} class heatmap. We set $K = 3$ and $M = 3$ as default settings for our DWSL model. At the end after applying $s(.)$ operation, we obtain one $c \times 1 \times 1$ layer representing the classification scores for each class.

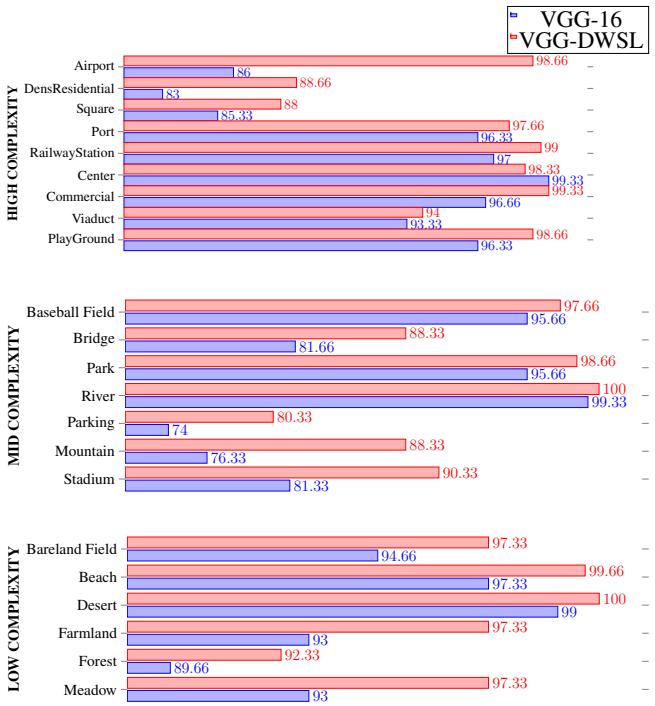


Fig. 5: Class-wise accuracies for VGG-16 (baseline) and VGG-DWSL (our method) for the AID dataset.

C. Object Bank Strategy

As a final step, we adopt an object bank strategy as highlighted in Fig. 2a. Similar to [13], we concatenate features computed from multiple scales of the input followed by training a support vector machine (SVM) classifier on top. This trained SVM model is then used to classify scenes from test dataset thus yielding final aerial scene classification score.

Hence, at the time of training, the K-max M-min pooling trains the network weights to accurately localize both the most discriminative regions (positive instances) that can correctly classify the scene plus those regions which have no correlation with the class (negative evidences). The reason for choosing negative instances is that some classes which have small inter-class variability (Playground and stadium class in Fig. 3) might result in high classification scores for both by only using positive instances, while classifying a scene lets say of stadium class. This is due to the fact that there is presence of positive instances (playground field and stadium roof tops respectively) for both the classes. In such a scenario, negative instances provide with complementary information (stadium roof giving clear evidence of absence of playground class) thus resulting in correct classification of the scene as stadium. This is elucidated empirically for with and without negative instances for our Resnet-DWSL model without object bank strategy (table I). Boost in overall classification score clearly highlights the necessity of using negative instances.

Method	Overall Accuracy (%)
Pretrained GoogLeNet [12]	85.84 ±0.92
Hierarchical Coding [14]	86.4 ±0.7
Pretrained VGG-VD-16 [12]	87.18 ±0.94
Pretrained CaffeNet [12]	88.25 ±0.62
Deep filter banks [15]	90.4 ±0.6
Two-stage deep feature fusion [16]	92.37 ±0.72
Our DWSL method	**96.19 ± 0.50**

TABLE III: Performance Comparison Over RSSCN7

Method	Overall Accuracy (%)
Pretrained GoogLeNet [12]	86.39 ±0.55
Pretrained CaffeNet [12]	89.53 ±0.31
Pretrained VGG-VD-16 [12]	89.64 ±0.36
$salM^3LBP - CLM$ [17]	89.76 ±0.45
Combining 2 FC Layers [18]	91.87 ±0.36
Two-stage deep feature fusion [16]	94.65 ±0.33
Our DWSL method	**95.06 ± 0.35**

TABLE IV: Performance Comparison Over AID dataset

III. Training and Implementation Details

A. Datasets

We conduct our experiments on 4 different datasets namely, AID[12], UCMerced[6], WHURS19 [11], RSSCN7 [10]. AID is a fairly recent large scale dataset composed of 10000 images coming from 30 different classes, drawn from Google imagery captured using multiple imaging source. An additional level of complexity arises from the larger intra-class variations as each sample is collected from different locations over varied time period and seasons. The other 3 datasets are limited in number and somewhat saturated in terms of performance.

B. Training

Our experimental setup is somewhat similar to [12]. We randomly draw 50% of our dataset as training set and rest is kept for testing. We repeat this step *thrice* to avoid the influence of randomness and compute precision accuracy for each run. We report overall mean and standard deviation over all runs in table II. We perform training on VGG-16 and Resnet-101 architectures as mentioned in section II.

For the **baseline models** (VGG-16 and Resnet-101), we first clip the final softmax layer + fully connected layer. Then, we add a fully connected layer with outputs equal to the number of classes present in the target dataset, finally appended with a softmax layer. Additionally, we pre-train these models on ImageNet and then, fine-tune them on target dataset.

For proposed **DWSL models**, we clip the last max pooling layer (just after convolution layers) + all the fully connected layers including soft max layer. Moreover, we simply add the Ws1 layer followed by K-max M-min pooling for Resnet-101 model. In case of VGG-16, an additional convolution layer of $7 \times 7 \times 2048$ convolutions is added before the Ws1 layer. We add a softmax layer at the end for both the models. Our training loss is a log loss function which depends on the probabilities computed by softmax layer.

To effectively fine tune our networks, we utilize a dampening factor layer (with dt=10) to divide the back-propagating gradient just before it reaches the feature extraction module. This is mainly to assure that the pre-trained feature extraction weights are not tampered much. We train our models using Adam [19] which computes adaptive learning rates for each parameter over the course of iteration. We set the initial learning rate to $1e-4$. Each run consists of 50 epochs and we choose the best epoch based upon the test accuracy.

Fig. 6: Scenes from Center and Viaduct categories

IV. Results and Discussions

A. Quantitative Results

Firstly, we compare the performance of our proposed model (DWSL) with the baseline model through overall mean accuracy over all runs as reported in Table II. Our weakly-supervised learning technique outperforms the current baselines by a considerable margin over all datasets. Especially in the case of VGG architecture, we witness a significant performance gain over the much recent AID [12] dataset.

Additionally for completeness, we also show comparisons with other state-of-the-art methods on AID and RSSCN7 benchmark datasets where we outperform all other hand-crafted as well as deep-learning methods by considerable margin. Our results are summarized in tables III and IV. Our method is superior to second best results by a margin of 3.8% and 0.4% on RSSCN7 and AID datasets respectively.

The results clearly emphasize that learning to pay attention to few discriminative regions in a scene and recognizing using only these regions is much more beneficial than recognition by visualizing the entire scene. Another conclusion that can be drawn here is since the design of Resnet architecture naturally preserves spatial information throughout the network, it tends to learn more meaningful discriminative regions as compared to VGG network thus contributing to better performance.

Since Resnet architecture gives state of art performance on [12] which is a fairly challenging dataset, therefore, we stick to Resnet to demonstrate the effectiveness of our method for all other well known datasets.

Next, we study the recognition accuracies from the perspective of scene complexities as depicted in Fig. 4. We observe that enhancement in performance is significantly higher in case of medium (5.66 point) and high complexity (3.22 point) regions than in case of low-complexity (2.88 point) regions. Delving deeper into class-wise predictions (in Fig. 5), we witness huge improvements in recognizing mid-level or high level complexity scenes such as Bridges (6.66 points), Parking (6.33 point), Mountain (12 point), Stadium (9 point), Airport (12.66 point) and Dense Residential (5.66 point). Thus we can fairly conclude that our weakly supervised methodology assists in minimizing ambiguity while recognizing more complex scenes where there is the presence of much more distinctive regions.

Point to note here is for the accuracy test results for high complexity scenes, we find that most of them have, on an

average gain of at least 1.3 - 2 percentage points which is quite significant considering that for many of them the baseline performance is already quite high. Only in two of the cases, *i.e.*, for center and viaduct the performance becomes comparable. In case of Center class (Fig. 6), we observe that most of the scenes in this category have one big roof structure in the center of the image instead of large number of scattered objects throughout the scene. Thus it is more straightforward to classify a scene from the complete image rather than selecting few relevant regions. Similarly in case of viaduct, there are large-scale loopy structures located in the center of image which again makes both our method and baseline that uses full image competing enough.

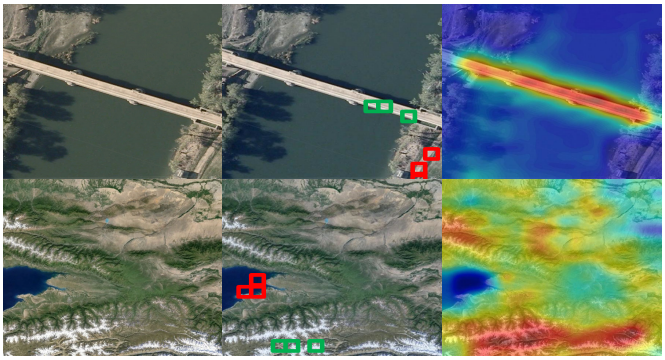


Fig. 7: Qualitative Results for Bridge and Mountain.

B. Qualitative Results

Fig. 7 demonstrates qualitative results for our methodology by depicting aerial scenes (column 1), most and least distinctive regions marked by green and red boxes respectively (column 2) and the overlaid resulting heatmaps (column3). It is quite evident that the network selects only those relevant regions which are characteristic feature of a particular class. At the same time it also picks up insignificant regions which best demonstrate the absence of a class.

For eg. in case of the bridge scene, a neural network might confuse it with a river if overlooking at the entire scene. However by providing a weak localization supervision using image labels in our method, we tend to make the network decide on which are the most representative regions in a scene that best describe a scene. Similarly in case of mountain scene, the network, by looking at the entire scene, might interpret it to be a Bareland or Forests or Meadows, however by focusing on just the snow covered regions it tends to correctly classify it as Mountains. Thus we can conclude that making the network to localize distinctive regions and only focusing upon them for recognizing the scene, considerably removes the ambiguity caused while visually inspecting the complete scene.

V. Conclusions

In this paper, a deep weakly supervised technique is proposed in the perspective of aerial scene classification. We illustrate how a network can learn to localize the most predominant regions in an image simply from the scene labels. We also conclude that using these discriminative regions instead of the entire scenes results in state-of-the-art performance while at the same time providing meaningful interpretable information that allows one to elucidate better the decision of a network. We substantiate empirically as to how, by adopting our weakly supervised paradigm, we tend to remove ambiguity in more complex aerial scenes, resulting in a boost in their performance. In addition to this, our experimental results also underline that Resnet architectures suits much better to our task due to its characteristic feature of spatial information preservation. Finally we showcase a nice visualization tool to highlight most relevant regions in aerial scenes.

References

[1] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 881–893, Feb 2017.

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre, "Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017.

[3] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, "Classification of hyperspectral imagery using a new fully convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 292–296, Feb 2018.

[4] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sensing*, vol. 8, no. 6, 2016.

[5] David H Hubel and Torsten N Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[6] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2010, GIS '10, pp. 270–279, ACM.

[7] Haifeng Li, Jian Peng, Chao Tao, Jie Chen, and Min Deng, "What do we learn by semantic scene understanding for remote sensing imagery in CNN framework?," *CoRR*, vol. abs/1705.07077, 2017.

[8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, Jan. 1997.

[9] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015.

[10] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep Learning Based Feature Selection for Remote Sensing Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp. 2321–2325, Nov. 2015.

[11] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Matre, "Structural high-resolution satellite image indexing," 2010.

[12] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang, "AID: A benchmark dataset for performance evaluation of aerial scene classification," *CoRR*, vol. abs/1608.05167, 2016.

[13] T. Durand, N. Thome, and M. Cord, "WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks," in *CVPR)*, 2016.

[14] Hang Wu, Baozhen Liu, Weihua Su, Wenchang Zhang, and Jinggong Sun, "Hierarchical coding vectors for scene level land-use classification," *Remote Sensing*, vol. 8, no. 5, 2016.

[15] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, "Deep filter banks for land-use scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1895–1899, Dec 2016.

[16] Y. Liu, Y. Liu, and L. Ding, "Scene classification based on two-stage deep feature fusion," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 183–186, Feb 2018.

[17] X. Bian, C. Chen, L. Tian, and Q. Du, "Fusing local and global features for high-resolution scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2889–2901, June 2017.

[18] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for vhr remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4775–4784, Aug 2017.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.