Deep Learning for Multilabel Land Cover Scene Categorization Using Data Augmentation

Radamanthys Stivaktakis, Grigorios Tsagkatakis, and Panagiotis Tsakalides

***Abstract*— Land cover classification is a flourishing research topic in the field of remote sensing. Conventional methodologies mainly focus either on the simplified single-label case or on the pixel-based approaches that cannot efficiently handle high-resolution images. On the other hand, the problem of multilabel land cover scene categorization remains, to this day, fairly unexplored. While deep learning and convolutional neural networks have demonstrated an astounding capacity at handling challenging machine learning tasks, such as image classification, they exhibit an underwhelming performance when trained with a limited amount of annotated examples. To overcome this issue, this paper proposes a data augmentation technique that can drastically increase the size of a smaller data set to copious amounts. Our experiments on a multilabel variation of the UC Merced Land Use data set demonstrate the potential of the proposed methodology, which outperforms the current state of the art by more than 6% in terms of the F-score metric.**

***Index Terms*— Convolutional neural networks (CNNs), data augmentation, land cover, multilabel classification, remote sensing, scene categorization.**

I. Introduction

HIGH-resolution imaging sensors aboard the miniaturized satellite and aerial vehicles acquire large amounts of high-resolution imagery, which mandates the development of automated and sophisticated algorithms for reliably processing and deriving meaningful information. This can become more apparent in time-sensitive scenarios [1], [2] or in cases where temporal dynamics play a major role. Land cover classification remains one of the biggest challenges in the remote sensing discipline and a crucial component in monitoring physical and anthropogenic phenomena on a large scale. Semantic segmentation of satellite images has been widely applied [3]–[7], in a pixelwise manner, but as denoted in [8], there are certain limitations to that approach when dealing with the high-resolution images. Meanwhile, in higher level feature-based approaches [8]–[10], each image is processed as a whole, with a subsequent goal to be associated with a descriptive label of the scene content.

Manuscript received September 9, 2018; revised December 24, 2018; accepted January 12, 2019. Date of publication February 4, 2019; date of current version June 24, 2019. This work was supported in part by the DEDALE Project, under Contract 665044, within the H2020 Framework Program of the European Commission. *(Corresponding author: Radamanthys Stivaktakis.)*

R. Stivaktakis and P. Tsakalides are with the Institute of Computer Science, FORTH, 70013 Heraklion, Greece, and also with the Department of Computer Science, University of Crete, 71003 Heraklion, Greece (e-mail: stivakt@ics.forth.gr).

G. Tsagkatakis is with the Institute of Computer Science, FORTH, 70013 Heraklion, Greece.

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.2019.2893306

Existing approaches consider the *multiclass* classification scenario, where every image is categorized to a single class, an assumption that oversimplifies the annotation process since a given scene can depict more than one class. Associating each image with multiple labels is known as the *multilabel* classification problem, which unlike the multiclass case, has not been adequately explored as far as remote sensing, and land cover scene classification is concerned. Previous works in the natural image classification literature [11]–[13] expose the principal challenges that met in the multilabel classification case and propose potential solutions.

The extraction of meaningful and descriptive features from remote sensing imagery has been a critical step in the design of automated and sophisticated machine learning algorithms. Generic image features, such as scale invariant feature transform [14], although highly effective, suffer from an overreliance on heuristic optimizations and human intervention, while remote sensing tailored features, such as normalized difference vegetation index [15], are too closely coupled with particular types of observations.

In this letter, we consider the cutting edge framework of deep learning and, more specifically, the recently (re-) discovered methodology of convolutional neural networks (CNNs) for the problem of multilabel land cover scene categorization. CNNs have established themselves as the state of the art in numerous fields, from image enhancement [16] and video analytics [17] to spectral imaging [18] and remote sensing [9], [10]. However, the lack of adequately sized data sets can seriously limit the performance of deep learning models, necessitating alternative solutions, given that most multilabel land cover data sets are small. To that end, we exploit the CNNs' transformation invariance property, based on the fact that a CNN must be able to robustly classify inputs (in our case images), regardless of small alterations of their content. Specifically, we employ *data augmentation*, a technique that has proven to be especially effective in image classification. Data augmentation proposes a simple, yet powerful, framework where the size of the small labeled data set, derived in a limited set of conditions, can be artificially increased through a series of transformations, such as translations, flips, and rescaling. As previous studies in the single-label case have shown [1], [19], CNN classification with data augmentation can have a substantial impact on multiple remote sensing scenarios.

The main contribution of this letter lies in the use of a cutting edge methodology, namely CNNs with dynamic data augmentation, tailored for multilabel land cover scene classification. The proposed method marks a clear departure

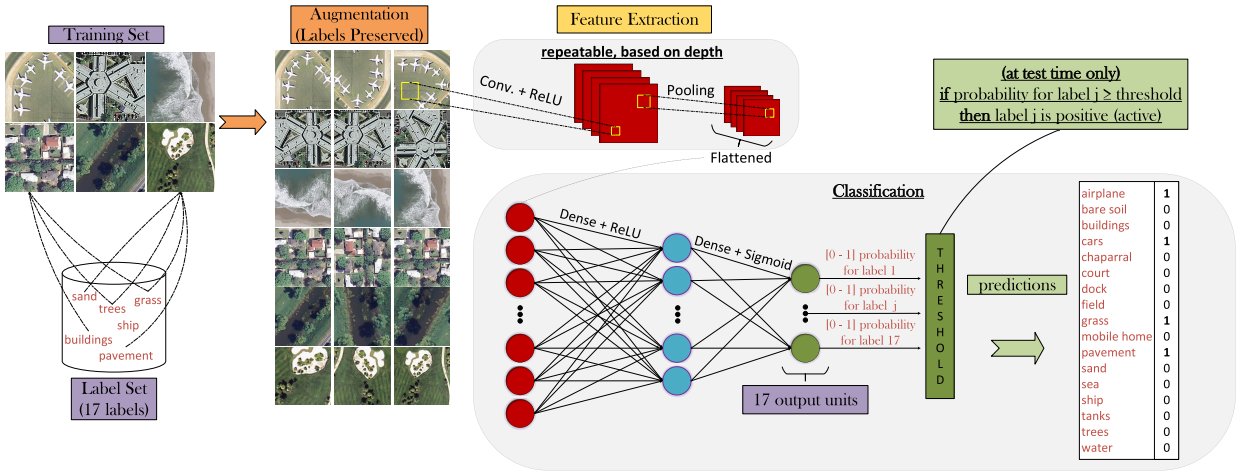

Fig. 1. Pipeline of the proposed methodology. Each batch of the training set is dynamically and randomly augmented at every iteration of the training process. The augmented data are fed into a deep CNN with sigmoid output units, which is trained with backpropagation based on a chosen loss function. Probability thresholding is used for the predictions only at test time. The test set is excluded from the data augmentation methodology.

for existing techniques, such as the current state-of-the-art graph-theoretic semisupervised approach [20], or a recent work [21] that exploits different types of features, either hand-crafted or derived via transfer learning, to calculate image distances and to obtain corresponding similarities. Our method, apart from being the first to employ a fully trainable, end-to-end deep learning model for the task at hand, manages, at the same time, to significantly outperform the state of the art on a redefined version, from a multilabel perspective, of the UC Merced Land Use data set [22]. The sequence of steps of our approach is shown in the block diagram of Fig. 1.

II. Proposed Approach

A. Convolutional Neural Networks

The main principles that define a CNN architecture can be handily derived if one breaks it apart to its respective components, commonly known as *layers*. The fundamental component of such a network, i.e., the *convolutional layer*, encodes the spatial correlations of a given input, by identifying appropriate 2-D filters. These trainable filters essentially map local, possibly overlapping, regions of the preceding layer to the units of the succeeding layer, resulting in local connectivity patterns. To be able to effectively form deeper, more complex CNN models, a nonlinearity needs to be introduced directly after each convolutional layer, in the form of a nonlinear activation layer like the *rectified linear unit* (ReLU): $f(x) = \max(0, x)$. ReLU is preferred over other commonly used choices (e.g., hyperbolic tangent or sigmoid), mainly because it is easily differentiable and less prone to saturation and the introduction of vanishing gradients. Finally, the *pooling* layer is responsible for introducing desired properties, such as scale invariance, through a form of nonlinear downsampling. The main intuition behind pooling lies in the fact that the exact location and orientation of a detected feature is less significant than its relative position to other features. With pooling, the processed input is commonly split into evenly sized, nonoverlapping local regions, and for each region, a given operation is executed (e.g., maximum, average, and L2-norm). Thus, the most relevant information is preserved, leading at the same time to a substantial reduction of the data dimensionality and, consequently, to an increased robustness against overfitting.

The aforementioned components are directly associated with the so-called "feature extraction" segment of a CNN. Meanwhile, for the classification part, the features that were derived in the final layer of the feature extractor must be connected with the nodes of a *dense layer*, in a fully connected manner. Multiple dense layers (typically with an ReLU activation) can be stacked together to form even deeper architectures, with the final predictive layer utilizing a probabilistic activation function, such as softmax or sigmoid. Given that softmax is normalized to strictly output probabilities that will always add up to one, it is considered an ideal choice for a single-label multiclass scenario, where classes are mutually exclusive, albeit not as good of a choice for the multilabel case. With softmax, as the trained system's confidence for the prediction of a specific class increases, there is a need to enhance the probability score of that specific class and simultaneously decrease the respective probabilities of the remaining classes. This is an undesirable property for multilabel classification where multiple labels are associated with each example. Instead of selecting the single label with the maximum probability score, the network must select all those labels with a score large enough that renders them active. To that end, for each individual output unit of the CNN, we must be able to efficiently transition from its predicted score to the binary decision of designating a label as active or inactive. Considering that the number of active labels is different for each observation, there are no guarantees that a sufficiently high softmax probability score for a certain label, for a given sample, will also be regarded as high for another sample.

To address this challenge, the proposed method employs the sigmoid activation function $f(x) = 1/(1 + e^{-x})$ at the output layer, yielding probability scores without constraints concerning their sum. During inference, translating the probabilities associated with each output node into a binary prediction for each label requires the utilization of an appropriately defined threshold, such that a label is considered active if the associated score exceeds the threshold. Finally, to train the proposed CNN architecture, we employ the binary cross

entropy (BCE) loss function, given by

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}[\, y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

where the scalar value n represents the number of training samples associated with each training batch, $y^{(i)}$ corresponds to the ground-truth label vector of the ith sample of the batch, and $\hat{y}^{(i)}$ corresponds to the predicted score vector for the same sample.

B. Image Data Augmentation

The state-of-the-art deep learning architectures are characterized by massive amounts of trainable parameters, in the order of tens of millions. Optimizing the performance of such complex models is challenging and can lead to model overfitting to the training data set if the number of training examples is not sufficiently large. For an image analysis problem, data augmentation can be an effective solution, which can significantly increase the total number of available annotated training examples through a variety of transformations. These transformations include the rotation of the image by different amounts, image rescaling, horizontal and vertical flips, translations to the x- and y-axes, and the addition of noise. By modifying the representation of each image without affecting its semantic information, thus preserving the associated annotations, CNNs are effectively trained on a significantly larger data set than the one initially available. In this letter, we employ an online approach for data augmentation where each training batch is dynamically augmented at every iteration. Compared with an off-line alternative, dynamic augmentation removes the memory requirements associated with larger static data sets and reinforces the generalization capabilities of the network since the CNN will rarely or never process the same example twice.

III. Data Set

The UC Merced Land Use data set [22] includes aerial images extracted from larger images of the USGS National Map Urban Area Imagery Collection and has been widely used in various remote sensing applications. It is a high-resolution data set that contains 2100 different images (of 256×256 pixels) evenly split among 21 unique classes. UC-Merced has been considered on many different land cover categorization methodologies, which, however, focus on the single-label scenario. To demonstrate the merits of the proposed approach, we have utilized UC-Merced with a completely redesigned labelset[1] [20], which is suited for the case of multilabel classification. Specifically, the new labelset consists of 17 different labels, listed in Table I, so that each image in UC-Merced is associated with multiple labels, ranging from one to seven, in accordance with its content.

TABLE I

MULTILABEL EXTENSION OF UC-MERCED AND THE NUMBER OF SAMPLES ASSOCIATED WITH EACH LABEL

label	# of samples	label	# of samples
airplane	100	mobile home	102
bare soil	633	pavement	1305
buildings	696	sand	389
cars	884	sea	100
chaparral	119	ship	102
court	105	tanks	100
dock	100	trees	1015
field	106	water	203
grass	977		

IV. Experimental Analysis and Discussion

A. Experimental Setup

In this letter, we propose a CNN architecture consisting of three Convolutional+ReLU+Max pooling layers, one Dense-ReLU layer, and one Dense-Sigmoid layer, with a global confidence threshold of 0.45 for all labels. An increasing number of 128, 256, and 512 trainable kernels has been deployed per convolutional layer, with a kernel size of 3×3 and a stride of 2×2 each. For the pooling operator, a nonoverlapping window size of 2×2 has been utilized on all applicable convolutional layers. Concerning the optimization process, the BCE loss has been adopted, along with the Adagrad optimizer [23].

Different experimental variations have been considered. In each variation, the same setting of the previous paragraph has been applied, altering, each time, only one of the available hyperparameters. Each experimental variation has been trained and tested five times, and for each adopted performance metric, an average value has been computed. Specifically, for each of these five experiments, the UC-Merced data set is randomly split into a training set of 1600 samples and a test set of 500 samples for evaluating the performance. In all the experiments, the network has been trained for 300 epochs, with a training batch size of 10. Furthermore, batch normalization has been examined, leading to a faster convergence of the training process, as well as a minor increase in the final performance.

The transformations that were used for dynamic data augmentation include image rotation, translation, and horizontal and vertical flips. Specifically, in the case of image rotation, we used a degree range for random rotations of $[-45, 45]°$, and for image translation, we performed random shifts in a maximum range of the 20% of the total height or width of the image. Considering that the augmentation of the training set is dynamic, the size of augmented data can be calculated by multiplying the initial size of the training set by the number of epochs that the network was trained. In our case, we end up with $1600 \times 300 = 480\,000$ training samples. In order to minimize any potential cross-contamination between the training and test sets, data augmentation is not performed on the test set.

To quantify the performance, the following metrics have been utilized, in order to provide a direct comparison with the current state of the art [20]:

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i|}, \quad \text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|}$$

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}, \quad \text{F-score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

[1] http://bigearth.eu/index.html

TABLE II

PERFORMANCE AND STANDARD DEVIATION (PARENTHESES) OF THE PROPOSED ARCHITECTURE, WITH DIFFERENT DROPOUT OPTIONS. DATA AUGMENTATION HAS BEEN UTILIZED

Dropout	Accuracy	Precision	Recall	F-score
No	81.2 (0.7)	87.3 (0.6)	88.5 (0.7)	87.9
0.25	81.3 (0.8)	88.0 (0.8)	89.3 (0.7)	88.6
0.50	**81.4 (0.4)**	**88.2 (0.5)**	**89.5 (0.6)**	**88.8**
0.75	79.7 (0.0)	86.4 (0.2)	89.1 (0.3)	87.7

TABLE III

PERFORMANCE AND STANDARD DEVIATION (PARENTHESES) OF THE PROPOSED ARCHITECTURE, WITH DIFFERENT DROPOUT OPTIONS. DATA AUGMENTATION HAS BEEN OMITTED

Dropout	Accuracy	Precision	Recall	F-score
No	68.0 (0.5)	80.4 (0.5)	77.6 (1.4)	79.0
0.25	73.7 (0.8)	85.1 (1.0)	81.1 (0.6)	83.1
0.50	75.7 (0.5)	85.4 (0.6)	83.3 (0.7)	84.3
0.75	**77.7 (0.2)**	**85.5 (0.1)**	**85.8 (0.4)**	**85.7**

where n corresponds to the number of samples in the evaluated data set (the test set), Y_i corresponds to the real labelset of the ith sample, and Z_i corresponds to the predicted labelset. The union ($\cup$) and the intersection ($\cap$) operators return a new set with the bitwise OR and the bit-wise AND of the elements of the two operand labelsets, respectively. Finally, the $|\cdot|$ operator counts the number of active labels (number of 1s) of the given set.

B. Experimental Results

The experimental setup described in Section IV-A was used as a baseline for the various experiments conducted. As a first comparison, in Tables II and III, we can observe that the use of data augmentation leads to a generous performance improvement with or without the use of regularization (i.e., dropout). Another important observation is that in the case where data augmentation is omitted, dropout can meaningfully improve the final outcome, given that, initially, the trained CNN overfits the small training data set. On the other hand, as shown in Table II, the impact of dropout greatly diminishes as the augmentation of the training set leads to a stronger mitigation of the effects of overfitting.

In Fig. 2, we explore the impact of the different sigmoid thresholds and how they are translated to the network's increased requirements for more confident predictions. The results demonstrate that low threshold values lead to overoptimistic (high recall, low precision) predictions, while high threshold values result to conservative (low recall, high precision) predictions. Nevertheless, all thresholds seem to result in reasonable F-score evaluations, with values between 0.3 and 0.4 qualifying as the optimal selections.

In Fig. 3, we perform a data-driven analysis on how the initial size of the given training set can affect the final performance of the trained CNN. In the case where data augmentation is employed, even though there is an obvious benefit with each increase, this benefit is not as pronounced as in the no-augmentation scenario. This result is in line with our intuition since with data augmentation, the transformation of the initial training examples leads to a fairly large data set, regardless of its original limited size, whereas without data augmentation, the initial size remains unaltered, rendering each increase far more impactful.

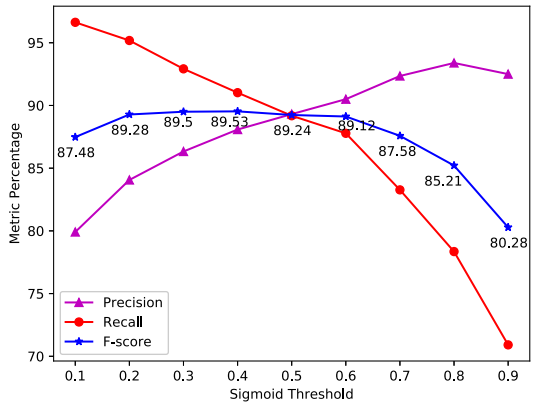

Fig. 2. Plot of the precision, recall, and F-score percentages for different sigmoid probability thresholds. The remaining hyperparameters stay unchanged. The extreme threshold values of 0 and 1 have been excluded.

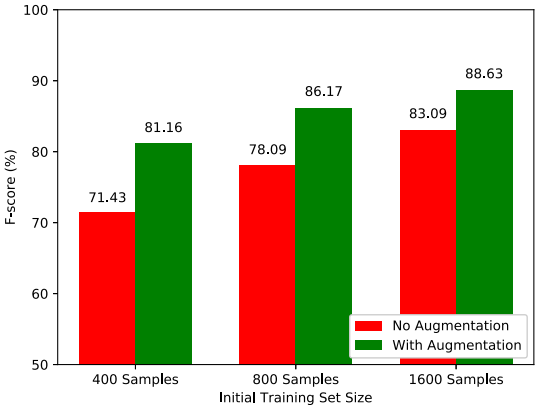

Fig. 3. Demonstration of the impact of the initial size of the training set, with and without the utilization of data augmentation. In the case of data augmentation (green bars), the initial size is projected into much larger numbers. Given that in all experiments, the network was trained for 300 epochs, the final size of the training data set is increased to 120 000, 240 000, and 480 000, indicated by the first, second, and third green bars, respectively.

Fig. 4 presents some indicative annotations inferred by the proposed model. We observe that the trained network manages to correctly predict the majority of the ground-truth labels of the tested images. Certainly, there are some cases where it fails to perceive the existence of certain objects. For example, in Fig. 4(a), it misses the building in the bottom-left corner, presumably because it has inferred that buildings are usually found in groups and rarely in maritime environments. In other cases, and for equivalent reasons, it attributes specific labels to the image that, in reality, are false positives. For example, in Fig. 4(b), the network is confident that it detects cars, given that in most freeway images in the data set, cars are present. At last, in Fig. 4(c), we can observe that the network might fail to distinguish between different objects that might share some common attributes. For example, the green color of the existing courts seems to confuse the trained CNN, which falsely decides the existence of the grass label, instead of that of the court.

Finally, in Table IV, we present a comparison between our best-trained model and the aforementioned works on the same topic. The increase of 6.65% on the F-score and the increase

(a) (b) (c) (d)

Fig. 4. Examples of inferred annotations where bold indicates the correctly identified labels, italics denotes the labels detected by the proposed method but not identified as active in the ground truth, and underlined are the ground-truth labels not found by the proposed method. (a) **Dock, ship, water**, and buildings. (b) **Bare soil, grass, pavement**, trees, and *cars*. (c) **Buildings, cars, pavement, trees**, court, and *grass*. (d) **Buildings, cars, chaparral, pavement, sand, trees**, *bare soil*, and *tanks*.

TABLE IV
COMPARISON BETWEEN THE MULTILABEL IMAGE RETRIEVAL MODEL [21], THE GRAPH-BASED APPROACH [20], AND OUR PROPOSED CNN WITH DATA AUGMENTATION

Metric	MLIR-CF [21]	GB [20]	CNN DA
Accuracy	61.88	74.29	**82.29**
Precision	68.13	85.68	**88.08**
Recall	81.77	80.25	**91.02**
F-score	74.33	82.88	**89.53**

of 8% on the multilabel accuracy, compared with the current state of the art, clearly show the capabilities of the proposed approach.

V. CONCLUSION

In this letter, we demonstrated the benefit of using deep CNN architectures along with data augmentation to efficiently address the problem of multilabel land cover scene classification. The performed experiments confirmed the impressive capabilities of the proposed methodology that managed to outperform the current state of the art by more than 6% in terms of the F-score in a multilabel modified version of the UC-Merced Land Use data set. This letter serves to further confirm the potential of deep learning for simultaneous feature extraction and classification, while in future work, we could explore the potential of using pretrained models and fine-tuning the architectures for multilabel classification.

REFERENCES

[1] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, Mar. 2016.

[2] C. Lippitt, D. Stow, and L. Coulter, Eds., *Time-Sensitive Remote Sensing*. New York, NY, USA: Springer, 2015.

[3] H. Lin, Z. Shi, and Z. Zou, "Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network," *Remote Sens.*, vol. 9, no. 5, p. 480, 2017.

[4] K. Karalas, G. Tsagkatakis, M. Zervakis, and P. Tsakalides, "Deep learning for multi-label land cover classification," *Proc. SPIE Image Signal Process. Remote Sens. XXI*, vol. 9643, p. 96430Q, 2015.

[5] J. Geng, J. Fan, H. Wang, X. Ma, B. Li, and F. Chen, "High-resolution SAR image classification via deep convolutional autoencoders," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2351–2355, Nov. 2015.

[6] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.

[7] G. Tsagkatakis and P. Tsakalides, "Deep feature learning for hyperspectral image classification and land cover estimation," in *Proc. ESA Symp.*, 2016, p. 179.

[8] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[9] D. Marmanis, M. Datcu, T. Esch, and U. Stilla, "Deep learning earth observation classification using ImageNet pretrained networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 105–109, Jan. 2016.

[10] G. Xu, X. Zhu, D. Fu, J. Dong, and X. Xiao, "Automatic land cover classification of geo-tagged field photos by deep learning," *Environ. Model. Softw.*, vol. 91, pp. 127–134, May 2017.

[11] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[12] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1609–1616.

[13] R. S. Cabral, F. Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for multi-label image classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 190–198.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1150–1157.

[15] S. W. Running, "Estimating terrestrial primary productivity by combining remote sensing and ecosystem simulation," in *Remote Sensing of Biosphere Functioning* (Ecological Studies). Cham, Switzerland: Springer, 1990.

[16] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 1725–1732.

[18] K. Fotiadou, G. Tsagkatakis, and P. Tsakalides, "Deep convolutional neural networks for the classification of snapshot mosaic hyperspectral imagery," *Electron. Imag.*, vol. 2017, no. 17, pp. 185–190, 2017. [Online]. Available: https://www.ingentaconnect.com/content/ist/ei

[19] W. Sun and R. Wang, "Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with DSM," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 474–478, Mar. 2018.

[20] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzon, "Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1144–1158, Feb. 2018.

[21] Z. Shao, K. Yang, and W. Zhou, "Performance evaluation of single-label and multi-label remote sensing image retrieval using a dense labeling dataset," *Remote Sens.*, vol. 10, no. 6, p. 964, 2018.

[22] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2010, pp. 270–279.

[23] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.