EPM: Meta-learning method for Remote Sensing Image Classification

Shiva Pundir[1] and Jerry Allan Akshay[1]

University Visvesvaraya College of Engineering, Bangalore, Karnataka, India
{shiva210199,jerry.allan.akshay}@gmail.com

Abstract. Remote sensing image scene classification is a challenging but fundamental task in understanding remote sensing images. Deep learning-based methods show enormous potential to understand remote sensing images. But, these models require a huge dataset in order to be trained efficiently and accurately without overfitting. Our proposed model, the EPM (Ensemble of Prototypical Networks and Model Agnostic Meta Learning), aims to tackle the aforementioned shortcomings posed by the traditional deep-learning networks by employing a meta-learning based approach to the remote sensing image classification problem. We evaluate the EPM model on five datasets (AID, NWPU-RESISC45, RSI-CB128, PatternNet, and UC-Merced), and compare the accuracy provided by 1-shot, 5-shot, and 10-shot classifications. We also provide comparisons between the two distance metrics employed in the EPM model.

Keywords: Few Shot Learning · Prototypical Networks · Model Agnostic Meta Learning · Remote Sensing.

1 Introduction

Remote sensing images are an important data source that can help us observe and measure structures on the surface of the Earth. The volume of remote sensing images is growing at an exponential rate due to the recent advancements in satellite technology. This gives an impetus to make full use of the increasing repository of remote sensing images that can be used for intelligent earth observations. Thus, understanding collections of remote sensing images is extremely important. Due to an increased interest to understand and interpret remote sensing images accurately and effectively, the field of scene classification of remote sensing images has become an active research area. In order to label the given remote sensing image data accurately with predefined semantic categories, remote sensing image scene classification is required. In the recent decades, extensive research has been done on remote sensing image scene classification, with various real-world applications such as Geospatial Object Detection [1, 2], Natural Hazards Detection [3,4], Environment Monitoring [5], and Urban Planning [6,7].

It is a known fact that good feature descriptions and representations are prerequisites to extract semantic information from remote sensing image data.

Feature engineering is an important step in remote sensing image classification. In the recent years, due to the exponential increase in the computing power available to the public, there has been a shift from crafting features manually to extracting features from the data samples via deep learning techniques; which are discussed in the next section of the paper.

However, these networks are usually prone to failure when there happens to be limited available annotated data. The failure occurs because these networks tend to overfit the input data. This consequently leads to the networks being unable to generalise well. These networks also tend to learn skewed class priors, and hence can be biased towards dominant classes of the distribution; and in cases where heavy-tailed class distributions exist, they do not generalize well. In juxtaposition, humans have the ability to quickly learn from a few examples by using already learnt prior knowledge. Such a capacity in fast adaptation and data efficiency can greatly expand the utilization of machine learning in various applications.

Hence, meta-learning techniques have emerged to facilitate learning from rather small amounts of available annotated data. These techniques allow the system to rapidly adapt to new environments and tasks with a small number of training examples.

In this research paper, we test and explore meta-learning techniques such as Prototypical networks [8] (distance metric based learner) and MAML [9] (gradient-based method) to help identify and label remote sensing images. To summarise, the contributions made in this paper are as follows:

- We propose to use an ensemble of meta-learning methods in order to help label and classify remote sensing image datasets that have long-tailed class distributions, with few available annotated data samples. We formulate the classification problem as a n-shot (1-shot, 5-shot or 10-shot) classification problem. We call the proposed network as the EPM model (Ensemble of Prototypical network and MAML).
- We explore MAML which is a gradient based method, Prototypical networks which are metric-learning based, and the ensemble of both; to classify remote sensing images in low-data scenarios.
- We also evaluate the EPM model on the publicly available remote sensing datasets such as AID [10], NWPU-RESISC45 [11], RSI-CB128 [12], PatternNet [13], and UC-Merced [14]. The results demonstrate that the EPM model outperforms prototypical networks and MAML working as a standalone model.

The rest of this research paper is organized as follows - section 2 presents the related work, section 3 presents the proposed methodology, section 4 presents the dataset and evaluation, section 5 presents the experiment and results, and section 6 presents the conclusion.

2 Related Work

In the recent decades, there has been extensive research on remote sensing image scene classification, driven by its real-world applications.

The rudimentary stages of scene classification were mainly based on handcrafted features as demonstrated by [14–18]. In order to help design a variety of features and their combinations, these methods were dependent on the engineer's skill-set and their domain knowledge. These characteristics carry vital information that can be used to classify scenes. A variety of these techniques exist such as SIFT [19], HOG [20], and many more.

In recent years, unsupervised feature learning has emerged as a viable alternative to handcrafted features. Unsupervised feature learning can extract features automatically from images from unlabeled input data. There exists many unsupervised feature learning techniques such as sparse coding, principal component analysis (PCA), autoencoder, and K-means clustering. A few of these techniques can be integrated together, and few models can be stacked in order to form deeper unsupervised models. These techniques have led to considerable progress with unsupervised feature learning in the remote sensing image scene classification domain [21–25].

However, unsupervised feature learning techniques alone aren't enough to provide the accurate results since there is a lack of semantic information which is provided by class labels. Hence, labelled data is still needed in order to develop better and more efficient supervised feature learning methods.

When Salakhutdinov and Hinton [26] made a breakthrough in deep feature learning, there was a shift in the techniques used by the researchers. Researchers were now using deep multi-layer neural networks in order to obtain deep feature representations. These methods comprise of multiple processing layers which can be used to learn powerful deep feature representations of data with multiple levels of abstraction [27].

At present, a variety of deep learning models exist such as Deep Belief Nets (DBNs) [28], Convolutional Neural Networks (CNNs) [29, 30], Stacked Autoencoder (SAE) [31], and many other such deep learning models. But these models use a large volume of data, and require a large sample size in order to be trained accurately. Therefore, we apply the few shot learning paradigm to the remote sensing classification problem to tackle these limitations.

3 Proposed Methodology

3.1 Preliminaries

Few Shot Learning Paradigm Few-shot learning models are trained on a labeled dataset D_{tr} and tested on D_{te} with only a few labeled examples per class in the test set. The class sets are disjoint between D_{tr} and D_{te}. Therefore, few shot learning deals on the data $D = \{D_{tr}, D_{te}\}$ together. The few shot approaches rely on an *episodic* training arrangement where it is simulated by taking few samples from the large labeled set D_{tr} during training as shown Fig. 1. The K-shot, N-way episode training consists of each episode e taking a sample from N categories from the D_{tr} and then sampling two sets of images from these categories: (i) the *support* set $S_e = (s_i, y_i)_{i=1}^{N \times K}$ containing K examples for each of the N categories and (ii) the *query* set $Q_e = (q_j, y_j)_{j=1}^{Q}$ containing different examples from the same N categories.

Like every other training of a model in machine learning, the objective here is to minimize the loss on the data samples in Q_e, given the S_e. This loss function can be denoted as the negative log-likelihood of the true class of each query sample with parameter θ:

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{(S_e, Q_e)} - \sum_{t=1}^{Q_e} \log p_\theta(y_t | q_t, S_e) \tag{1}$$

where for episode e the sampled query and support set are $(q_t, y_t) \in Q_e$ and S_e respectively.

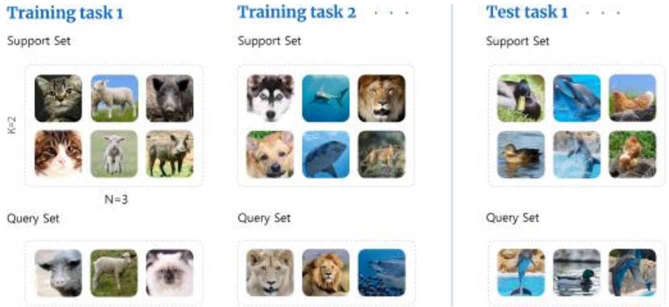

Fig. 1: Few shot learning episodes (Here with $K = 2$ and $N = 3$)

Prototypical Networks Prototypical network is a distance metric based meta-learning technique which calculates the representation of each class as a prototype vector, and this vector is the mean vector of the embedded support instances belonging to its class as shown in Fig. 2. For preparation of one training task, a subset of N classes are selected randomly. For each training task, a support set $S = (\mathbf{x}_1, y_1), ..., (\mathbf{x}_u, y_u)$ and a query set $Q = (\mathbf{x}_{u+1}, y_{u+1}), ..., (\mathbf{x}_{u+v}, y_{u+v})$ are created by sampling examples from the selected classes, where $\mathbf{x}_j$ are inputs and y_j represents corresponding labels. Here, the number of examples in support and query set are represented by u and v respectively. The representations of the inputs are calculated in the embedding space using $\mathbf{x}$ with an embedding function ϕ: $\mathbf{z} = \phi(\mathbf{x}, \boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$. For a class k, the prototype vector $\mathbf{c}_k$ is calculated as the mean of all the embedded vectors of the example inputs

S_k of the corresponding class k. Formally, $\mathbf{c}_k$ is computed as:

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_j, y_j) \in S_k} \phi(\mathbf{x}_j, \boldsymbol{\theta}) \tag{2}$$

For a query sample $\mathbf{x}$, the probability distribution of the predicted labels y is calculated using softmax over negative distances to the prototypes in the embedding space:

$$p(y = k|\mathbf{x}, \mathbf{c}_k) = \frac{exp(-d(\mathbf{z}, \mathbf{c}_k))}{\Sigma_{k'}\ exp(-d(\mathbf{z}, \mathbf{c}_{k'}))} \tag{3}$$

where $\mathbf{z} = \phi(\mathbf{x}, \boldsymbol{\theta})$. The learning begins by minimizing the negative log-probability of true label k, which can be formulated as:

$$J = -\log(p(y = y_j|\mathbf{x}_j, \mathbf{c}_k)) \tag{4}$$

which is computed using (2) with the estimated prototypes.

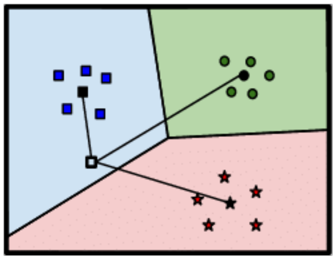

Fig. 2: The prototype c_k calculated as the mean of embedded examples for each class from support set.

Model Agnostic Meta-Learning Model Agnostic Meta-Learning (MAML) is a gradient-based meta-learning technique for few shot learning. The approach that MAML follows is to learn initial starting parameters of a network in such a way that it can quickly learn the optimized parameters pertaining to any few shot task in a proficient manner, which can then yield accurate results as demonstrated in Fig. 3.

We define the *base* model to be a neural network represented by f_θ with meta-parameters θ. The aim is to learn initial parameters θ_0 such that it can then be adjusted to obtain good task-specific parameters after only a few effective gradient descent steps on the data from a support set S_b. The parameters obtained after N steps is denoted by θ_N, and the steps are called *inner-loop* update processes.

The updated θ parameters after i steps on data from the support task S_b can be expressed as:

$$\theta_i^b = \theta_{i-1}^b - \alpha \nabla_\theta \mathcal{L}_{S_b}(f_{\theta_{(i-1)}^b}) \tag{5}$$

where α is the step size, θ_i^b are the parameters after i steps towards task b, $\mathcal{L}_{S_b}(f_{\theta_{(i-1)}^b})$ is the loss on the support set of task b after $(i-1)$ update steps. It is worth noting that θ_i^b is invariant to permutation of the samples. The *meta-objective* function with *task batch size* B can be denoted as:

$$\mathcal{L}_{meta}(\theta_0) = \sum_{b=1}^{B} \mathcal{L}_{T_b}(f_{\theta_N^b(\theta_0)}) \tag{6}$$

The qualitative value of initial parameter θ_0 is expressed as the sum of the losses throughout all the tasks. This transforms into a meta objective function which is minimized to optimize the initial parameter value θ_0. This optimization is called the *outer-loop* update process. It is this initial θ_0, that intrinsically holds the different task's knowledge. The resulting update for the meta-parameters θ_0 can be expressed as:

$$\theta_0 = \theta_0 - \beta \nabla_\theta \sum_{b=1}^{B} \mathcal{L}_{T_b}(f_{\theta_N^b(\theta_0)}) \tag{7}$$

where β is step size and $\mathcal{L}_{T_b}$ denotes the loss on the target set for task b.

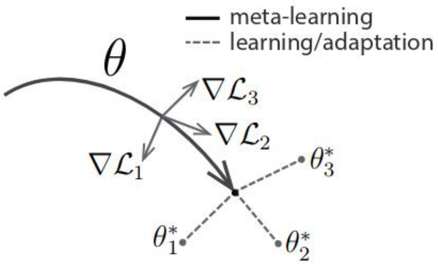

Fig. 3: Optimizing initial θ such that MAML can quickly adapt to new tasks.

3.2 EPM

We present the remote sensing image classification problem as a few-shot problem. This is mainly due to two reasons. Firstly, the remote sensing image dataset can be skewed, and this causes the model to perform poorly on classes with less number of training examples. Secondly, training a large complex CNN is time-consuming, and CNNs in general are very data-hungry. It is a well known fact

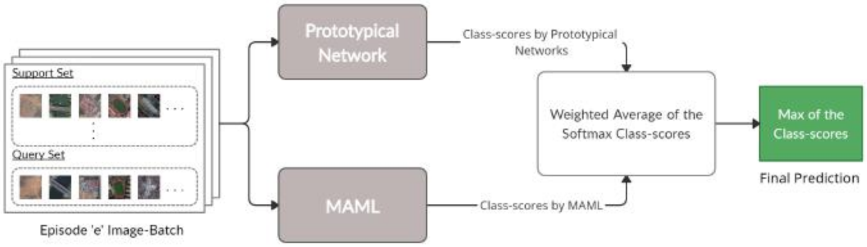

Fig. 4: Overview architecture of the EPM.

that ensemble methods tend to reduce the variance of estimators, and subsequently improve the quality of prediction as demonstrated in [32]. In order to gain accuracy from averaging, various data augmentation techniques or randomizations are typically used to encourage a high diversity of predictions [33, 34]. While individual classifiers of the ensemble network may perform poorly, the quality of the average prediction surprisingly turns out to be high sometimes. Hence, we develop an ensemble model called EPM model. An overview of the ensemble model architecture is shown in Fig. 4. The model broadly consists of two meta-learning methods, namely distance-metric based Prototypical Network (PN) and gradient-based MAML. For each episode e, the image-batch is input into both the models and the models output the class-scores. Then, the weighted-average of these class-scores are calculated using w_1 and w_2 to predict the final label of each image in the query set.

4 Dataset and Evaluation

We use five very well known datasets in the remote sensing image recognition field for a comprehensive result. We briefly describe each of the dataset used. Fig. 5 describes few examples from the datasets.

AID is a dataset published by Wuhan University that consists of 30 scene classes. Each scene class consists of roughly 220-420 images, and each image has a size of 600 X 600 pixels. The dataset consists of 10,000 images, and the pixel resolution of each image ranges from 0.5m to 8m.

NWPU-RESISC45 is a dataset which consists of 45 scene classes, with each class consisting of 700 images, and each image has a size of 256 X 256 pixels. The dataset consists of 31,500 images, and the pixel resolution of most images ranges from 0.2 m to 30m.

RSI-CB128 is a dataset that contains 6 categories with 45 sub-classes of more than 36,000 images. The dataset includes variable image samples but averages out to 800 image samples per class. The pixel resolution varies within 0.3 - 3 m.

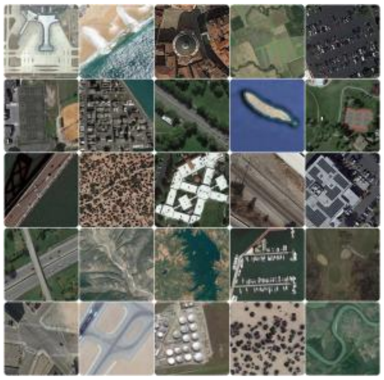

Fig. 5: Few examples collectively from the 5 dataset. Each row represents a dataset.

PatternNet is a high-resolution remote sensing image dataset assembled for RSIR. PatternNet is based on project "TerraPattern", which is an open-source tool for discovering "patterns of interest" in unlabeled satellite imagery. It consists of thirty-eight classes, with a total of 800 images, where each image has a size of 256 × 256 pixels. The spatial resolution varies from lowest - 0.062 m to highest - 4.693 m.

UC-Merced is a dataset published by the University of California Merced. This dataset consists of twenty-one scene classes, and each image has a fixed size of 256 X 256 pixels. The dataset consists of 2,100 images, and the pixel resolution of each image is 0.3 m.

5 Experiment and Results

We use the Deep Learning framework named PyTorch [35] to conduct the experiment. To get a comprehensive result, we run each experiment for 400 episodes and average over them. We tabulate the results into Table 1.

For Prototypical networks, we use four convolution blocks connected sequentially to extract image features. Each convolution block consists of a number of neural network layers connected in a sequential manner. Each block of the convolution network comprises of a 64-filter 3×3 convolution layer followed by a batch normalization layer [36]. We use ReLU nonlinearity as the activation function, which is finally followed by a 2×2 max-pooling layer. This layer architecture results in a 64-dimensional output space when applied to images. All of our CNN models are trained via Stochastic Gradient Descent with Adam optimiser. We use learning rate of 10^{-3} as the initial value, and then cut the learning rate in half every 1000 episodes. We trained prototypical networks in the 5-way (1-shot, 5-shot, and 10-shot) scenarios with training episodes containing 10 query points per class.

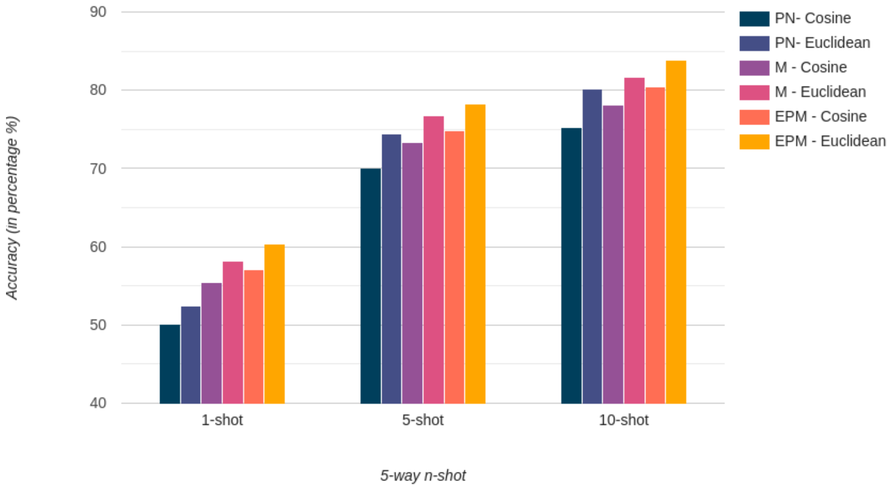

Fig. 6: Few examples collectively from the 5 dataset. Each bar represents a dataset.

For MAML, we use the same neural network architecture with a fully connected linear layer head that predicts the probability distribution of the classes for the images. Due to the computational limit, we only use 1st order MAML with meta batch size of 4 samples. Following the original paper, we keep the hyper-parameters same i.e we train the model using Adam optimiser and Cross Entropy Loss as the loss function. We use an initial learning rate of 10^{-3}, and reduce the learning rate to half for every 10 episodes if the loss doesn't decrease. We also keep the inner loop learning rate to be 0.01. Similar to prototypical networks, we provide results for 5-way (1-shot, 5-shot, and 10-shot) classification.

After experimenting we found $w_1 = 0.58$ and $w_2 = 0.42$ to be the useful values for calculating the weighted average of the class scores. We also evaluated the effects of distance metric and the number of training samples used while testing per episode on the performance of prototypical networks and MAML. We represent the results on a bar-graph in Fig. 6.

6 Conclusion

We propose the EPM model as a more efficient and accurate classification model to address the problem of remote sensing image classification, specifically for training classes with relatively small sample sizes. Through our tests and analyses on five different datasets, we confirm that the few shot learning method can be applied to the remote sensing image classification problem. We believe that further research in this path and direction can be propelled by incorporating newer state-of-the-art few shot learning algorithms that can be used as a standalone model, or even as an ensemble of different models.

Dataset Used	1-shot			5-shot			10-shot		
	PN	M	EPM	PN	M	EPM	PN	M	EPM
AID	52.4	58.2	**60.4**	73.4	76.8	**77.9**	80.2	81.7	**83.8**
NWPU-RESISC45	49.0	57.3	**62.7**	75.2	76.3	**78.3**	80.5	80.8	**83.5**
RSI-CB128	50.2	56.9	**61.5**	74.4	77.0	**78.2**	79.9	80.2	**83.7**
PatternNet	51.4	60.2	**61.9**	74.0	76.8	**79.3**	80.7	79.9	**84.1**
UC-Merced	51.9	58.5	**60.9**	75.1	77.1	**78.5**	81.0	81.3	**84.3**

Table 1: Average accuracy on 400 test episodes
PN: Prototypical Networks, M: Model Agnostic Meta Learning, EPM: Ensemble of Prototypical Networks and MAML

Acknowledgment

We would like to thank our mentor Dr. P Deepa Shenoy for her guidance, and Massimiliano Patacchiola for his helpful learning content. This work was supported by University Visvesvaraya College of Engineering.

References

1. Y. Li, Y. Zhang, X. Huang, and A. L. Yuille, "Deep networks under scene-level supervision for multi-class geospatial object detection from remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 182–196, 2018.
2. K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296–307, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271619302825
3. T. Martha, N. Kerle, C. Westen, V. Jetten, and K. vinod Kumar, "Segment optimization and data-driven thresholding for knowledge-based landslide detection by object-based image analysis," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, pp. 4928 – 4943, 12 2011.
4. G. Cheng, L. Guo, T. Zhao, J. Han, H. Li, and J. Fang, "Automatic landslide detection from remote-sensing imagery using a scene classification method based on bovw and plsa," *International Journal of Remote Sensing*, vol. 34, no. 1, pp. 45–59, 2013.
5. T. Zhang and X. Huang, "Monitoring of urban impervious surfaces using time series of high-resolution remote sensing images in rapidly urbanized areas: A case study of shenzhen," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 8, pp. 2692–2708, 2018.
6. N. Longbotham, C. Chaapel, L. Bleiler, C. Padwick, W. J. Emery, and F. Pacifici, "Very high resolution multiangle urban classification analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1155–1170, 2012.
7. A. Tayyebi, B. C. Pijanowski, and A. H. Tayyebi, "An urban growth boundary model using neural networks, gis and radial parameterization: An application to tehran, iran," *Landscape and Urban Planning*, vol. 100, no. 1, pp. 35–44, 2011.

8. J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” *CoRR*, vol. abs/1703.05175, 2017. [Online]. Available: http://arxiv.org/abs/1703.05175
9. C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *CoRR*, vol. abs/1703.03400, 2017. [Online]. Available: http://arxiv.org/abs/1703.03400
10. G. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, “Aid: A benchmark data set for performance evaluation of aerial scene classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017.
11. G. Cheng, unwei Han, and X. Lu, “Remote sensing image scene classification: Benchmark and state of the art,” *CoRR*, vol. abs/1703.00121, 2017.
12. H. Li, C. Tao, Z. Wu, J. Chen, J. Gong, and M. Deng, “RSI-CB: A large scale remote sensing image classification benchmark via crowdsource data,” *CoRR*, vol. abs/1705.10450, 2017.
13. W. Zhou, S. D. Newsam, C. Li, and Z. Shao, “Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval,” *CoRR*, vol. abs/1706.03424, 2017.
14. 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*, ser. GIS ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 270–279.
15. S. Bhagavathy and B. S. Manjunath, “Modeling and detection of geospatial objects using texture motifs,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 12, pp. 3706–3715, 2006.
16. J. dos Santos, O. Penatti, and R. Torres, “Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification.” vol. 2, 01 2010, pp. 203–208.
17. E. Aptoula, “Remote sensing image retrieval with global morphological texture descriptors,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 3023–3034, 2014.
18. O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, “Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 44–51.
19. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.
20. N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” vol. 1, 07 2005, pp. 886–893.
21. M. Mekhalfi, F. Melgani, Y. Bazi, and N. Alajlan, “Land-use classification with compressive sensing multifeature fusion,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, 08 2015.
22. A. M. Cheriyadat, “Unsupervised feature learning for aerial scene classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 439–451, 2014.
23. X. Zheng, X. Sun, K. Fu, and H. Wang, “Automatic annotation of satellite images via multifeature joint sparse coding with spatial relation constraint,” *IEEE Geoscience and Remote Sensing Letters*, vol. 10, pp. 652–656, 07 2013.
24. G.-S. Xia, Z. Wang, C. Xiong, and L. Zhang, “Accurate annotation of remote sensing images via spectral active clustering with little expert knowledge,” *Remote Sensing*, vol. 7, pp. 15 014–15 045, 11 2015.

25. Y. Li, C. Tao, Y. Tan, K. Shang, and J. Tian, "Unsupervised multilayer feature learning for satellite image scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, 11 2015.
26. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
27. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.
28. G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
29. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 1097–1105.
30. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.
31. P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, p. 3371–3408, Dec. 2010.
32. G. Seni and J. Elder, *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*, 01 2010, vol. 2.
33. A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 05 2018, pp. 117–122.
34. B. Hu, C. Lei, D. Wang, S. Zhang, and Z. Chen, "A preliminary study on data augmentation of deep learning for image classification," *CoRR*, vol. abs/1906.11887, 2019.
35. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.
36. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.