Deep Ensemble Learning Model Based on Covariance Pooling of Multi-Layer CNN Features

Sara Akodad, Lionel Bombrun, Maria Puscasu, Junshi Xia, Christian Germain, Yannick Berthoumieu

DEEP ENSEMBLE LEARNING MODEL BASED ON COVARIANCE POOLING OF MULTI-LAYER CNN FEATURES

Sara Akodad[1,2]*, Lionel Bombrun*[1]*, Maria Puscasu*[1]*, Junshi Xia*[3]*, Christian Germain*[1] *and Yannick Berthoumieu*[1]

[1] : Université de Bordeaux, CNRS, IMS, UMR 5218, Groupe Signal et Image, F-33405 Talence, France

e-mail: firstname.lastname@ims-bordeaux.fr

[2] : CNES, Centre National d'Etudes Spatiales, 18 Avenue Edouard Belin, F-31400 Toulouse, France

[3] : RIKEN Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo 103-0027, Japan; junshi.xia@riken.jp

ABSTRACT

Compared to standard deep convolutional neural networks (CNN) which include a global average pooling operator, second-order neural networks have a global covariance pooling operator which allows to capture richer statistics of CNN features. They have been shown to improve representation and generalization abilities. However, this covariance pooling is performed only on the deepest CNN feature maps. To benefit from different levels of abstraction, we propose to extend these models by using a multi-layer approach. In addition, to obtain better predictive performance, an end-to-end ensemble learning architecture is proposed. Experiments are conducted on four datasets and have confirmed the potential of the proposed model for various image processing applications such as remote sensing scene classification, indoor scene recognition and texture classification.

***Index Terms*—** Covariance pooling, multi-layer representation, ensemble learning, CNN.

1. INTRODUCTION

In recent years, deep learning has gained incredible popularity and many achievements can be found in literature [1]. Indeed, deep learning algorithms have been shown to outperform classic machine learning methods in a variety of contexts. For example, convolutional neural networks (CNN) have been employed successfully in image classification applications [2, 3]. They are composed of a series of hidden layers, each of which is in charge of extracting and learning specific features from the input images [4]. First-order pooling steps, such as average or max pooling operations, are common in CNN models. But recently, many authors have expressed interest in using a higher-order representation, such as second-order pooling. It basically consists in computing the covariance matrix of CNN features [5,6]. Since they are symmetric positive definite (SPD) matrices, covariance matrices are manifold-valued data. They hence lie on a Riemannian manifold and not on an Euclidean space. Information geometry tools are then required to process them. Since then, several authors have proposed various second-order neural network architectures to benefit from both second-order statistics and deep learning architectures [7–14]. The pooled covariance matrix from CNN outputs was one of the first attempts [7]. The Riemannian SPD matrix network (SPDNet) [10] is another way to use second-order statistics in a deep neural network. The goal of this network is to adapt the classical CNN fully connected (FC) convolution-like layers and rectified linear units (ReLU)-like layers to manifold-valued data. For that, the bilinear mapping (BiMap) layers and eigenvalue rectification (ReEig) layers were introduced. For SPDNet, the affine-invariant Riemannian metric is exploited. It involves matrix logarithm operation which may harm covariance pooling as observed in [15]. To overcome this issue, Wang *et al.* have proposed in [15] to exploit the power-Euclidean metric, allowing a robust covariance estimation by shrinking the largest sample eigenvalues and stretching the smallest ones. However, second-order representation is only used for the deepest layers in these models. To address this, He *et al.* published in [9] a multi-layer version: the multi-layer stacked covariance pooling (MSCP). Willing to exploit multi-layer CNN features richness and higher representations through second-order statistics in an ensemble learning approach, we have introduced in [16, 17] the ensemble learning covariance pooling (ELCP) architecture which consists in the extension of MSCP. This method aims at enhancing the classification performance by using different convolutional layer features of a CNN with various depth and combining different weak classifiers. This strategy allows to improve the classification accuracy by fusing the decision obtained on different subsets. Nevertheless, it is based on a transfer learning approach and cannot be trained from end-to-end. Based on these observations, the main contribution of this paper is to propose a deep learning model based on covariance pooling of multi-layer CNN features which can be trained from end-to-end.

The paper is structured as follows. Section 2 introduces the proposed model by detailing each building block. Then, Section 3 shows some experiments on four datasets including scene recognition and texture classification. An abla-

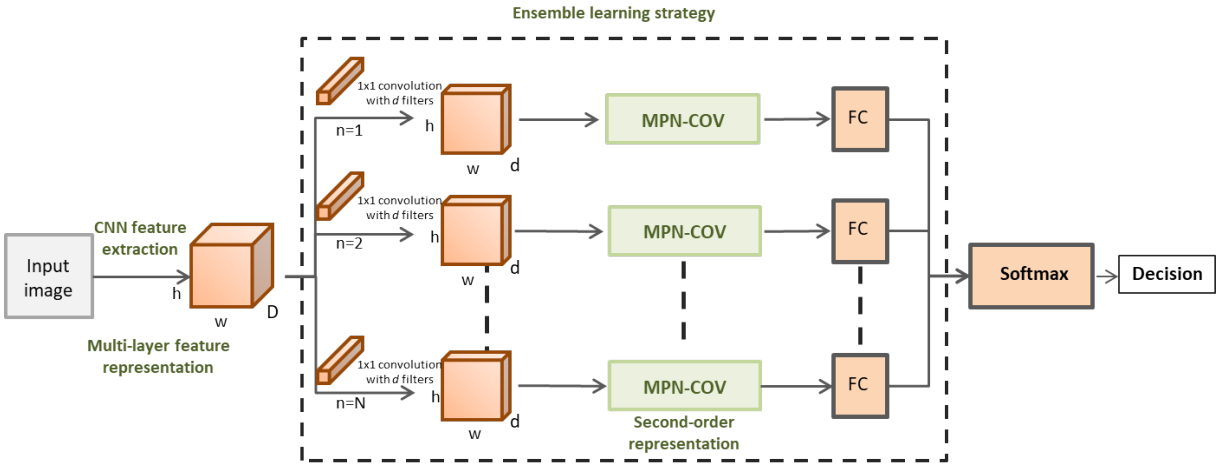

Fig. 1. Overview of the proposed deep ensemble learning model based on covariance pooling of CNN features.

tion study and a gradient-weighted class activation mapping (Grad-CAM) approach are performed to evaluate the added value of each part of the network. Finally, Section 4 gives some conclusions.

2. PROPOSED ARCHITECTURE

Fig. 1 shows an illustration of the proposed deep ensemble learning model based on covariance pooling of multi-layer CNN features. It is basically composed of a backbone which plays the role of a feature extractor. For that, a standard deep neural network can be employed such as the VGG-16 network. Then, three steps compose the architecture of the proposed model: a multi-layer feature extraction, a second-order representation of CNN features, and an ensemble learning strategy. The next subsections present in detail how these steps are employed.

2.1. Multi-layer feature extraction

CNNs are composed of multiple convolutional layers which allow to represent images at different levels of abstraction. Each layer provides a more advanced level of conceptual abstraction than the previous layer. In order to benefit from these different levels of abstraction, a multi-layer feature extraction approach is considered. It consists in the combination of CNN activation maps from different convolutional layers. Practically, the feature maps M_1, M_2 and M_3 produced by three deep convolutional layers ($conv_{3-3}$, $conv_{4-3}$ and $conv_{5-3}$) of the VGG-16 network are considered as shown in Fig. 2. However, CNN layers typically have varied spatial dimensions. Dimensions for the VGG-16 network are $M_1 \in \mathbb{R}^{56\times56\times256}$, $M_2 \in \mathbb{R}^{28\times28\times512}$ and $M_3 \in \mathbb{R}^{14\times14\times512}$. To stack the feature maps of these subsequent layers, a downsampling to the smallest spatial dimension is conducted using bilinear interpolation.

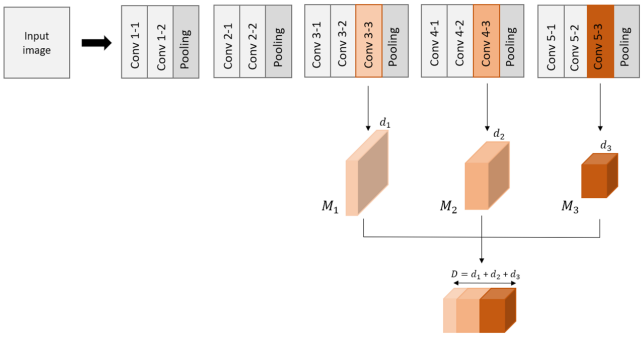

Fig. 2. Extraction of multi-layer features from three different VGG-16 convolutional layers.

2.2. Second-order representation

Many signal and image processing tasks, such as remote sensing scene classification or texture identification, have been demonstrated to benefit from second-order representation (i.e. covariance pooling) [9, 14, 16, 18, 19]. Motivated by these works and the success of deep neural networks, the proposed architecture integrates a covariance pooling operator. For that, the matrix power normalized covariance pooling (MPN-COV) structure introduced in [15] is employed. It is composed of the following operations shown in Fig. 3 and detailed hereafter.

Let's consider the output of the multi-layer feature extractor be a $h \times w \times d$ tensor with spatial height h, width w and channel d. After reshaping the tensor to a feature matrix $\mathbf{X}$ which consists of $n = wh$ features of d-dimension, the second-order pooling is performed by computing the sample covariance matrix as $\mathbf{\Sigma} = \mathbf{X}\mathbf{J}\mathbf{X}^T$ where $\mathbf{J} = \frac{1}{n}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T\right)$. $\mathbf{I}$ is the $n \times n$ identity matrix, $\mathbf{1}$ is the n-dimensional vector of one, and T denotes the matrix transpose operator. Then, a pre-normalization step is performed by dividing the covari-

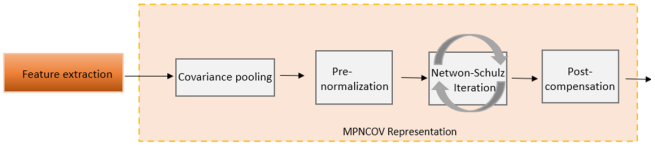

Fig. 3. Principle of the matrix power normalized covariance pooling (MPN-COV) operator [15].

ance matrix by its trace such that $\mathbf{A} = \frac{\Sigma}{tr(\Sigma)}$. The next step consists in using the power Euclidean metric to compare covariance matrices [20]. For that, the matrix square root normalization should be computed. Since $\mathbf{A}$ is a symmetric positive definite (SPD) matrix, it has a unique square root which can be computed by the eigen-decomposition. Indeed, $\mathbf{A} = \mathbf{U}\,\mathrm{diag}(\lambda_i)\mathbf{U}^T$, where $\mathbf{U}$ is an orthogonal matrix and $\mathrm{diag}(\lambda_i)$ is a diagonal matrix of eigenvalues λ_i of $\mathbf{A}$. Then $\mathbf{A}$ has a square root $\mathbf{Y} = \mathbf{U}\,\mathrm{diag}(\lambda_i^{1/2})\mathbf{U}^T$, with $\mathbf{Y}^2 = \mathbf{A}$. However, having a fast implementation on a GPU of the eigenvalue decomposition is still an open challenge. For this reason, an approximate iterative solution of the matrix square root is employed. For computing the square root $\mathbf{Y}$ of $\mathbf{A}$, these Newton-Schulz iterations are repeated five times:

$$\mathbf{Y}_k = \frac{1}{2}\mathbf{Y}_{k-1}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1}), \quad (1)$$

$$\mathbf{Z}_k = \frac{1}{2}(3\mathbf{I} - \mathbf{Z}_{k-1}\mathbf{Y}_{k-1})\mathbf{Z}_{k-1}, \quad (2)$$

where $\mathbf{Y}_0 = \mathbf{A}$ and $\mathbf{Z}_0 = \mathbf{I}$ the identity matrix. A post-compensation step is next used to counteract the adverse effect of the pre-normalization, such that $\mathbf{C} = \sqrt{tr(\Sigma)}\mathbf{Y}_5$. And finally, a vectorization operator is performed to consider the upper triangular entries of the resulting symmetric matrix, which forms an $d(d+1)/2$-dimensional vector. All these blocks can be trained from end-to-end. For more information, on the back-propagation operations, the interested reader is referred to [15].

2.3. Ensemble learning strategy

Ensemble learning algorithms in machine learning focus on combining numerous weak classifiers to build a stronger one [21, 22]. Random forest classifier, for example, is an ensemble learning method that uses decision trees to train each model on a separate sample of the same training dataset. The ensemble members' predictions are then pooled to make the final decision using simple operations, such as a majority vote for a classification problem. The diversity in the ensemble, which is actually ensured by the differences within the data on which each base classifier is trained, is the key fundamental reason for the success of ensemble learning systems. Inspired by this principle, we propose a deep ensemble learning model as shown in Fig. 1. The set of multi-layer feature maps are first transformed into N subsets of d new features maps. But, instead of randomly selecting d features out of the D feature maps as it can be done in a random forest classifier, the transformation is learned. For that, a 1×1 convolutional layer is employed for each subset. This supervised learning allows to automatically find the transformation adapted to the final classification problem. Note that the learning of the 1×1 convolutional layer for the second subset depends on the learning of the 1×1 convolutional layer for the first subset since they are learned jointly, and not independently. Next, for each subset, the second-order representation of the multi-layer features are fed into a fully connected layer. Their outputs are then concatenated and passed to a last fully connected layer with the softmax activation function to obtain the final decision.

3. EXPERIMENTS

In this section, we illustrate the potential of the proposed deep ensemble learning model in three different image classification problems. For that, four datasets are considered. The first two ones are benchmark remote sensing datasets used for aerial scene classification, namely the UC Merced land use dataset [23] and the aerial image dataset (AID) [24]. They consist respectively of 2 100 and 10 000 images distributed in 21 and 30 classes of aerial scene types such as airport, baseball field, dense residential, etc. The third dataset is the describable texture dataset (DTD) which is composed by a collection of real-world texture images [25]. It has 47 texture classes with 120 images per category. And finally, the last application concerns an indoor recognition scene challenge with the Indoor dataset [26]. This latter has 67 categories and a total of 15 620 images.

For the evaluation of performance, the standard classification protocol mentioned in the four previously cited papers is employed on these datasets. 80% of images are used for training for the UC Merced, DTD and Indoor datsets and the remaining 20% of images are used for testing, while for the AID dataset, only 20% of images are used for training as suggested in [24]. For the following experiments, the backbone is the VGG-16 network pretrained on ImageNet dataset. The weights of all its convolutional layers are frozen. Only the parameters of the following layers (1×1 convolution, MPN-COV, FC, etc.) are learned. For the ensemble approach, $N = 10$ subsets are considered and the 1×1 convolution is of depth $d = 256$, *i.e.* covariance matrices are of dimension 256×256. These last two parameters have been tuned by preliminary experiments and are remaining fixed for all the datasets.

3.1. Ablation study

In this subsection, an ablation study is performed to evaluate the added value of each element in the proposed deep ensemble learning approach based on covariance pooling of CNN features. Table 1 shows the classification performance in term of overall accuracy. On the first row, the model reduces to a

Multi-layer approach	Second-order representation	Ensemble strategy	UC Merced	AID	DTD	Indoor
✗	✗	✗	91.96	81.26	65.54	62.49
✓	✗	✗	95.31	87.56	66.15	63.68
✗	✓	✗	91.07	80.39	65.36	66.00
✓	✓	✗	96.21	88.00	69.27	72.43
✓	✓	✓	**98.44**	**88.95**	**71.27**	**73.79**

Table 1. Ablation study on the four considered dataset for the proposed deep ensemble learning approach based on covariance pooling of CNN features.

fine tuning of a VGG-16 network. The second row consists in a multi-layer version of this VGG-16 network. In the next three rows, a second-order representation of the CNN features is employed. The original MPN-COV published in [15] corresponds to the third row, while the full proposition is the last line where our ensemble learning strategy includes both a multi-layer approach and a covariance pooling step. As observed in Table 1, the multi-layer approach allows to consistently improve the classification performance. A significant gain of 1% to 8% are observed on the four datasets. The second-order representation of the CNN features also allows to improve the overall accuracy. Finally, the best results are obtained for the proposed architecture when these two elements are used in an ensemble learning approach. Note also that compared to [17], a gain of 0.3% is obtained on the UC Merced dataset for our end-to-end approach.

3.2. Visual explanation of the proposed model

To better understand the potential of the proposed approach, a gradient-weighted class activation mapping (Grad-CAM) method is employed [27]. Grad-CAM is a well-known visualization technique that is useful for understanding how a model was led to make a decision. It consists of producing heat maps representing the image areas that were exploited on the input images to provide the final decision. As such, heat maps indicate the importance of each pixel related to the class of interest by increasing or decreasing the intensity of the pixel value. Here, three images from three different classes of the UC Merced dataset are used, namely, the airplane, residential and forest classes. Their corresponding maps are shown in Fig. 4. The classification probability obtained on the output of the Softmax are displayed in the top right of each image. As observed, the proposed architecture (last row) allows to better focus on the object of interest. For example, with the proposed model, contour of the aircraft is better delineated in the airplane image than with a simple VGG-16 network. In addition, for the residential and forest images, the attention is better widespread on the characteristic object of the scene such as the houses and trees. Moreover, classification probabilities are consistently improved when each element (multi-layer approach, second-order representation, ensemble strategy) of the proposed architecture are employed, illustrating the value of using them together.

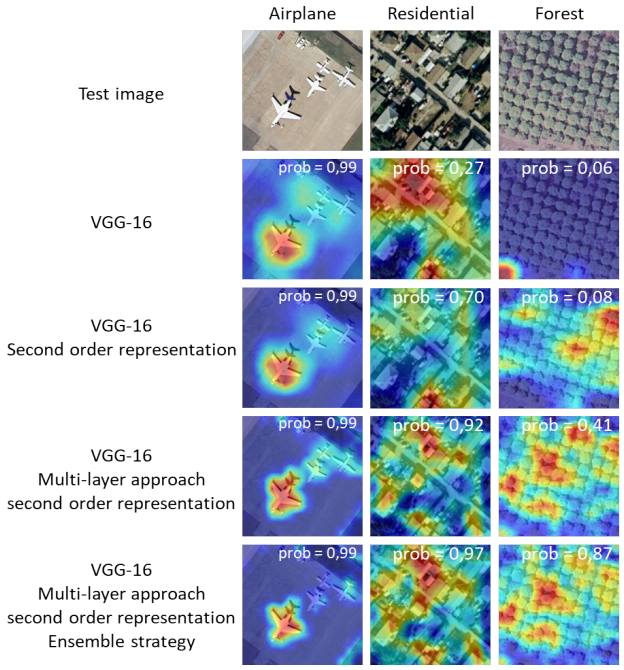

Fig. 4. Heat maps obtained with Grad-CAM for three images from the UC Merced dataset.

4. CONCLUSION

This paper has introduced a deep ensemble learning model based on the second-order representation of multi-layer CNN features. By considering a multi-layer approach, this model has allowed to represent images at different levels abstraction. Then, to model the dependencies between these activation maps, a global covariance pooling operator has been employed. Based on the Newton-Schulz iterations, the covariance matrix has been square root normalized and vectorized to obtain a vector representation. In addition, an ensemble architecture has been proposed to obtain better predictive performance. Experimental results on four dataset have confirmed the potential of the proposed model for various image processing applications such as remote sensing scene classification, texture classification and indoor scene recognition.

5. REFERENCES

[1] K. Li, W. Ma, U. Sajid, Y. Wu, and G. Wang, "Object detection with convolutional neural networks," 2019.

[2] Y. Le Cun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. Morgan-Kaufmann, 1990, pp. 396–404.

[3] 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. USA: Curran Associates Inc., 2012, pp. 1097–1105.

[4] N. Kriegeskorte, "Deep neural networks: a new framework for modelling biological vision and brain information processing," *bioRxiv*, 2015. [Online]. Available: https://www.biorxiv.org/content/early/2015/10/26/029876

[5] K. Yu and M. Salzmann, "Statistically motivated second order pooling," in *European Conference on Computer Vision*, 2018.

[6] P. Li, J. Xie, Q. Wang, and W. Zuo, "Is second-order information helpful for large-scale visual recognition?" 2018.

[7] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Matrix backpropagation for deep networks with structured layers," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2965–2973.

[8] S. Cai, W. Zuo, and L. Zhang, "Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 511–520.

[9] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "Remote sensing scene classification using multilayer stacked covariance pooling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 6899–6910, Dec 2018.

[10] Z. Huang and L. V. Gool, "A Riemannian network for SPD matrix learning," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 2036–2042.

[11] K. Yu and M. Salzmann, "Second-order convolutional neural networks," *CoRR*, vol. abs/1703.06817, 2017. [Online]. Available: http://arxiv.org/abs/1703.06817

[12] D. Acharya, Z. Huang, D. P. Paudel, and L. V. Gool, "Covariance pooling for facial expression recognition," *CoRR*, vol. abs/1805.04855, 2018. [Online]. Available: http://arxiv.org/abs/1805.04855

[13] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3019–3028.

[14] N. He, L. Fang, S. Li, J. Plaza, and A. Plaza, "Skip-connected covariance network for remote sensing scene classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1461–1474, 2020.

[15] Q. Wang, J. Xie, W. Zuo, L. Zhang, and P. Li, "Deep CNNs meet global covariance pooling: Better representation and generalization," *CoRR*, vol. abs/1904.06836, 2019. [Online]. Available: http://arxiv.org/abs/1904.06836

[16] S. Akodad, S. Vilfroy, L. Bombrun, C. C. Cavalcante, C. Germain, and Y. Berthoumieu, "An ensemble learning approach for the classification of remote sensing scenes based on covariance pooling of CNN features," in *27th European Signal Processing Conference*, La Coruña, Spain, Sep. 2019.

[17] S. Akodad, L. Bombrun, J. Xia, Y. Berthoumieu, and C. Germain, "Ensemble learning approaches based on covariance pooling of cnn features for high resolution remote sensing scene classification," *Remote Sensing*, vol. 12, no. 20, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/20/3292

[18] R. Rosu, M. Donias, L. Bombrun, S. Said, O. Regniers, and J. P. Da Costa, "Structure tensor Riemannian statistical models for CBIR and classification of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 248–260, Jan 2017.

[19] M.-T. Pham, G. Mercier, and L. Bombrun, "Color texture image retrieval based on local extrema features and Riemannian distance," *Journal of Imaging*, vol. 3, no. 4, p. 43, Oct 2017. [Online]. Available: http://dx.doi.org/10.3390/jimaging3040043

[20] I. L. Dryden, A. Koloydenko, and D. Zhou, "Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging," *The Annals of Applied Statistics*, vol. 3, no. 3, pp. 1102 – 1123, 2009. [Online]. Available: https://doi.org/10.1214/09-AOAS249

[21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[22] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[23] 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: ACM, 2010, pp. 270–279. [Online]. Available: http://doi.acm.org/10.1145/1869790.1869829

[24] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, X. Lu, and L. Zhang, "Aid: a benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, pp. 3965 – 3981, Feb 2017.

[25] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606–3613.

[26] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 413–420.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.