MULTI-SCALE CONVOLUTIONAL SVM NETWORKS FOR MULTI-CLASS CLASSIFICATION PROBLEMS OF REMOTE SENSING IMAGES

Gabriele Cavallaro[1], *Yakoub Bazi*[2], *Farid Melgani*[3] *and Morris Riedel*[1,4]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[2] Department of Computer Engineering, King Saud University, Saudi Arabia
[3]Department of Information Engineering and Computer Science, University of Trento, Italy
[4]School of Engineering and Natural Sciences, University of Iceland, Iceland

ABSTRACT

The classification of land-cover classes in remote sensing images can suit a variety of interdisciplinary applications such as the interpretation of natural and man-made processes on the Earth surface. The Convolutional Support Vector Machine (CSVM) network was recently proposed as binary classifier for the detection of objects in Unmanned Aerial Vehicle (UAV) images. The training phase of the CSVM is based on convolutional layers that learn the kernel weights via a set of linear Support Vector Machines (SVMs). This paper proposes the Multi-scale Convolutional Support Vector Machine (MCSVM) network, that is an ensemble of CSVM classifiers which process patches of different spatial sizes and can deal with multi-class classification problems. The experiments are carried out on the EuroSAT Sentinel-2 dataset and the results are compared to the one obtained with recent transfer learning approaches based on pre-trained Convolutional Neural Networks (CNNs).

Index Terms— Multi-scale convolutional support vector machine (MCSVM) network, supervised feature generation, multiclass classification, sentinel-2, remote sensing

1. INTRODUCTION

Passive optical sensors are a class of remote sensing instruments able to collect natural radiation from the Earth's surface and convert it into imagery [1]. A wide variety of resolutions, ranging from panchromatic to hyperspectral images, are nowadays available to serve different thematic applications. Among all the possible products that can be derived from remote sensing images, classification products are among the most frequently utilized. Supervised classification algorithms can be used to distinguish between different types of land-cover classes (e.g., streets, houses, grass, etc.) in order to interpret processes, such as monitoring of urban growth, land cover mapping, road network extraction, impacts of natural disasters, crop monitoring, object detection, etc. [2].

Recently, deep learning has brought in revolutionary achievements in many applications, including the classification of remote sensing images [3]. The state-of-the-art results have been achieved by the Convolutional Neural Networks (CNNs) [4, 5] due to their sophisticated hierarchical structure able to extract more hidden and deeper features than classic machine learning methods based on handcrafted features (i.e., shallow classifiers). The performance of a CNN classifier is considerably dependent on the amount of available training data (i.e., the larger is the training set the lower is the chance that overfitting occurs). Despite the recent advances in Earth observation benchmark data creation (e.g., RSI-CB128 with 36,000 images and 45 annotated classes [1]), the gap with the data size of the computer vision datasets (e.g., ImageNet with 14,197,122 images [2]) is still a key limiting factor for the development of effective deep network classifiers for remote sensing data. The scarcity of large, reliable and open-source annotated datasets is largely due to the inherent interpretation complexity of remote sensing data (e.g., RADAR images) and the time effort and cost involved in the collection of training samples. One effective approach to handle small/medium datasets is transfer learning [6]. It consists of acquiring the knowledge from networks that were pre-trained on an auxiliary recognition task with a higher number of labeled data (e.g., ResNet [7]) instead of performing the training from scratch.

Bazi *et al.* [8] proposed the Convolutional Support Vector Machine (CSVM) network for binary classification of datasets that have limited number of annotated samples. The CSVM adopts a learning strategy based on SVMs [9] that is alternative to the standard backpropagation algorithm [5]. The novelty was the introduction of the SVM convolutional layers that learn the kernel weights via a forward supervised learning strategy based on a set of linear SVMs. This paper introduces the Multi-Scale CSVM network for multiclass

The results of this research were achieved through the Human Brain Project PCP Pilot Systems at the Juelich Supercomputing Centre and the DEEP-EST project, which received co-funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No. 604102 and No. 754304, respectively.

[1]https://github.com/lehaifeng/RSI-CB
[2]http://www.image-net.org/

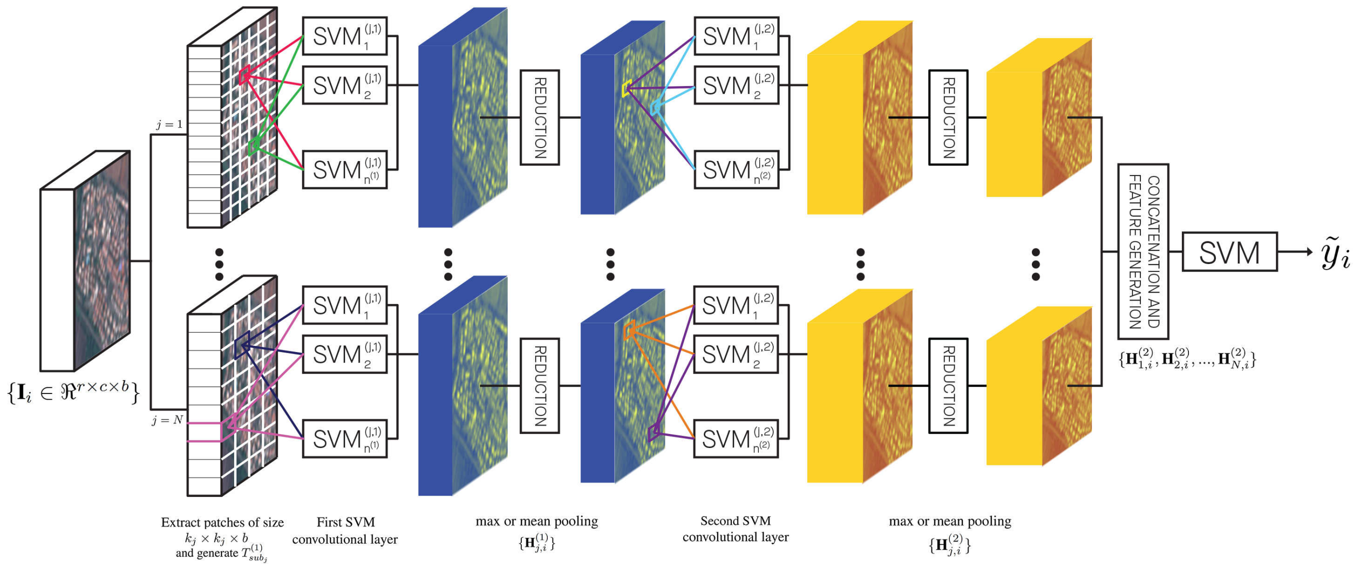
$\{\mathbf{I}_i \in \Re^{r \times c \times b}\}$
$j = 1$
$j = N$
$\mathrm{SVM}_1^{(j,1)}$
$\mathrm{SVM}_2^{(j,1)}$
$\mathrm{SVM}_{n^{(1)}}^{(j,1)}$
REDUCTION
$\mathrm{SVM}_1^{(j,2)}$
$\mathrm{SVM}_2^{(j,2)}$
$\mathrm{SVM}_{n^{(2)}}^{(j,2)}$
REDUCTION
CONCATENATION AND FEATURE GENERATION
$\{\mathbf{H}_{1,i}^{(2)}, \mathbf{H}_{2,i}^{(2)}, ..., \mathbf{H}_{N,i}^{(2)}\}$
SVM
$\tilde{y}_i$
Extract patches of size $k_j \times k_j \times b$ and generate $T_{sub_j}^{(1)}$
First SVM convolutional layer
max or mean pooling $\{\mathbf{H}_{j,i}^{(1)}\}$
Second SVM convolutional layer
max or mean pooling $\{\mathbf{H}_{j,i}^{(2)}\}$

Fig. 1. Example of a MCSVM network that receives as input batches of two different spatial sizes. The architecture includes two SVM convolutional layers, two reduction layers, a concatenation and feature generation step, and a classification layer placed on the top.

classification problems. It is an ensemble of CSVM networks, where each CSVM takes in input a set of batches at a defined spatial size. The experiments are conducted on the EuroSAT Sentinel 2 dataset which consists of classifying 10 land cover classes. The comparison of the preliminary results with the one obtained with pretrained CNNs confirms the effectiveness of the SVM convolutional layers for multi-class classification problems.

2. MULTI-SCALE CONVOLUTIONAL SVM

2.1. SVM Convolutional Layer

The MCSVM network (see Figure 1) is composed by N CSVM networks that receive in input batches with different sizes $K = \{k_j\}_{j=1}^{N}$, with $k_j \in Z^*$. Each CSVM network includes many SVM convolutional layers, which are different than the convolutional layers of standard CNNs. In the following sub-sections, the structure of the first SVM convolutional layer (i.e., $SVM_1^{(j,1)}$, $SVM_2^{(j,1)}$, ... , $SVM_{n^{(1)}}^{j,1}$) of a jth CSVM network is described. The generalization to the next layers is plain.

2.1.1. Formation of the Global Training Set

Let $\{\mathbf{I}_i, y_i\}_{i=1}^{M}$ be the training set composed of M multi-bands images with $\{\mathbf{I}_i \in \Re^{r \times c \times b}\}$, where r, c and b refers to the number of rows, columns and bands of the image, respectively, while $y_i \in Z^*$ denotes the class label. From each image $\{\mathbf{I}_i\}$, a set of non-overlapped patches of size $k_j \times k_j \times b$ are extracted and reshaped as feature vectors $\mathbf{x}_i$ of dimension $d_j = k_j \times k_j \times b$. The result is a global training set $T_j^{(1)} = \{\mathbf{x}_i, y_i\}_{i=1}^{m^{(1)}}$ of size $m^{(1)}$.

2.1.2. Training the set of SVM Filter Banks

A set of linear SVM filters are learned on distinct sub-training sets $T_{sub_j}^{(1)} = \{\mathbf{x}_i, y_i\}_{i=1}^{l}$ by optimizing the unconstrained optimization problem described in [10] (i.e., with the L2-loss function). The l samples are randomly extracted from the global training set $T_j^{(1)}$. After training, the SVM filters $\{\mathbf{w}_{j,z}^{(1)}\}_{z=1}^{n^{(1)}}$ are computed. The term $\mathbf{w}_{j,z}^{(1)} \in \Re^{g \times d_j}$ refers to zth-SVM filter weight matrix, while $n^{(1)}$ is the number of filters. Each filter matrix $\mathbf{w}_j \in \Re^{[g \times d_j]}$ includes the weights that are assigned to the features, with $g = n\_class * (n\_class - 1)/2$ (i.e., the output attribute *coef_* for multiclass cases with the linear kernel [3]). The complete weights of this convolutional layer could be grouped into a filter bank $\mathbf{W}_j^{(1)} \in \Re^{g \times d_j \times n^{(1)}}$.

2.1.3. Generation of the Convolutional Feature Maps

Each training image $\{\mathbf{I}_i, y_i\}_{i=1}^{M}$ is convolved with the SVM filters to generate a set of 3-D hyper-feature maps $\{\mathbf{H}_{j,i}^{(1)}\}_{i=1}^{M}$. Here, $\{\mathbf{H}_{j,i}^{(1)}\} \in \Re^{r^{(1)} \times c^{(1)} \times b^{(1)}}$ is the new feature representation of image $\mathbf{I}_i$ composed of $n^{(1)}$ feature maps. To obtain the *zth* feature map $\mathbf{h}_{z,j,i}^{(1)}$, the *zth* SVM filter is convolved with

[3]https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

Table 1. Classification results of different pretrained CNNs and the proposed MCSVM for the EuroSAT dataset. The experiments of the first four columns use the three features Red, Green and Blue colors while the last column, includes 14 features (i.e., the original 13 bands of Sentinel-2 concatenated with the NDVI). For each configuration, the average and standard deviation in brackets of several metrics are reported (the values result from 5 different generated training sets).

DenseNet169	ResNet-50	VGG16	MCSVM	MCSVM	Metrics
3 features (RGB)				14 features (sentinel 2 + NDVI)	
77.52 (3.77)	90.65 (0.56)	92.85 (0.26)	76.28 (0.96)	93.60 (0.37)	OA
82.04 (1.49)	90.45 (0.55)	92.59 (0.28)	77.51 (0.63)	93.43 (0.34)	AA
75.01 (4.19)	89.59 (0.63)	92.04 (0.29)	73.56 (1.07)	92.88 (0.41)	Kappa
1043.17 (129.69)	405.12 (4.35)	277.20 (5.84)	2658.40 (8.08)	928.39 (11.05)	Train+Test time [s]

a set of sliding windows of size $k_j \times k_j \times b$ (with a predefined stride) over the training image $\mathbf{I}_i$

$$\mathbf{h}^{(1)}_{z,j,i} = f(\mathbf{I}_i * \mathbf{w}^{(1)}_{z,j}), z = 1, ..., n^{(1)} \quad (1)$$

where $*$ is the convolution operator and f is the ReLU activation function. The spatial size of the feature maps $\{\mathbf{H}^{(1)}_{j,i}\}^M_{i=1}$ is then reduced via a pooling operation (i.e., either max of mean) as it is done in a pooling layer of a CNN network.

2.2. Generation of the Convolutional Feature Maps

At the last SVM computing layer L (convolution or reduction depending on the architecture) of each CSVM, the hyper-feature maps $\{\mathbf{H}^{(L)}_{j,i}, y_i\}^M_{i=1}$ are outputted. These maps are then concatenated $\{\mathbf{H}^{(L)}_{1,i}, \mathbf{H}^{(L)}_{2,i}, ..., \mathbf{H}^{(L)}_{N,i}, y_i\}^M_{i=1}$. Subsequently, the feature generation step takes as input each concatenated hyper-feature map $\{\mathbf{H}^{(L)}_{1,i}, \mathbf{H}^{(L)}_{2,i}, ..., \mathbf{H}^{(L)}_{N,i}\}$ for the training image $\mathbf{I}_i$ and compute the mean or max value for each feature map. A non linear SVM classifier with the Radial Basis Function (RBF) kernel is finally trained over these features.

3. EXPERIMENTAL RESULTS

3.1. Dataset

The experiments have been carried out on the open-source EuroSAT dataset [11]. It includes two subsets of Sentinel-2 satellite image patches (i.e., with spatial size of 64×64 pixels) [4]: one with the RGB colors and the other with the all 13 original spectral bands acquired by the Sentinel 2 multispectral sensor. Both subsets consist of 10 classes with in total 27,000 labeled images. The Normalised Difference Vegetation Index (NDVI [5]) is also considered as additional feature for the experiments of the MCSVM network.

[4]http://madm.dfki.de/downloads

[5]https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm

3.2. Experimental Setup

Preliminary results are presented for a single layer MCSVM network architecture. The EuroSAT dataset is split into a training and test set at the ratio of 80:20. The MCSVM ensembles three CSVM networks which receive as input batches with the following dimensions $K = \{8, 10, 12\}$. Each CSVM has only one SVM convolutional layer which includes 7 linear SVM classifiers. The 2D convolutional operations are performed with the stride set to 5 and combined with a reduction layer (max pooling) with window shape and stride set to 3 and 2, respectively. Each SVM filter is trained with $l = 1000$ patches (100 for each class) that are randomly extracted. The estimation of the penalty parameter C for each SVM is done via a 5-fold cross-validation procedure in the range $[10^{-1}10^2]$.

The experiments are performed on the JURON supercomputer system at the Jülich Supercomputing Centre. For implementing the CSVM network (i.e., with `Python 3.6.1`), the following software packages and respective versions were used: the `TensorFlow 1.7.0` (GPU) for the realization of the convolution operations, the ThunderSVM (GPU) for the SVM algorithm and the `h5py 2.8.0` for interfacing the Hierarchical Data Format (HDF5) files.

3.3. Evaluation

The performance of the MCSVM is evaluated in terms of the standard metrics Overall Accuracy (OA), Average Accuracy (AA) and Cohen's Kappa coefficient (Kappa), as depicted by Table 1. To rate the results, three popular pretrained CNNs are run with the same input setting: DenseNet169 [12], ResNet-50 [7] and VGG-16 [13]. For each pretrained network, the weights resulting from the training of the ImageNet dataset are frozen for all the layers (i.e., not trainable). Two additional layers are added on top (a fully connected layer followed by a softmax layer with 10 outputs) and trained with 20 epochs. The classification results of MCSVM that are obtained with the RGB dataset are not satisfactory, especially if they are compared with the one of the VGG16 network. However, with the 14 bands dataset (i.e., 13 bands of Sentinel-2 and the NDVI), the MCSVM is able to achieve competitive classification results. The pretrained networks are not evaluated with this dataset since their weights have been trained with 3-channel images and their original architectures have to be modified. Table 1 reports also the processing times (i.e., training plus test). The training time of the pretrained networks is only the fraction that was spent by the 20 epochs for learning the two top additional layers.

The good performance of the MCSVM network is especially encouraging if considered within the whole satellite image data framework. On the one hand, deep learning has proved capable of outperforming traditional machine learning classifier. On the other hand, the majority of the proposed deep networks operate with the limited RGB information. In the light of the fact that many earth observation programmes put a lot of effort and resources in designing sensors with higher resolutions, new methods have to be developed in order to avoid their underutilization.

4. CONCLUSIONS

This paper proposed a novel MCSVM network for multi-class classification problems of remote sensing images. Its architecture can include several CSVM networks that take as inputs batches with different spatial sizes. With this configuration the classifier can exploit both the spatial and spectral information and provide competitive classification results compared to recent solutions based on knowledge transfer from pretrained CNNs.

As perspectives, it is intended to investigate the effect of the different tunable parameters and additional layers on the performance of the network into more detail. Furthermore, thanks to the availability of distributed-GPU systems at Jülich Supercomputing Centre, it is planned to scale up the training phase by distributing the compute load (SVM training and convolution operations) over different GPUs.

5. REFERENCES

[1] W. G. Rees, *Physical Principles of Remote Sensing*, 3rd ed. Cambridge University Press, 2013, vol. 37.

[2] M. J. Canty, *Image Analysis, Classification and Change Detection in Remote Sensing: With Algorithms for ENVI/IDL and Python, Third Edition*. Taylor & Francis, 2014.

[3] J. E. Ball, D. T. Anderson, and C. S. Chan, "A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community," *SPIE Journal of Applied Remote Sensing (JARS), Special Section on Feature and Deep Learning in Remote Sensing Applications*, sep 2017. [Online]. Available: http://arxiv.org/abs/1709.00308

[4] A. Romero, C. Gatta, G. Camps-valls, and S. Member, "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1–14, 2015.

[5] L. Zhang, L. Zhang, and B. Du, "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, jun 2016.

[6] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] Y. Bazi and F. Melgani, "Convolutional SVM Networks for Object Detection in UAV Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3107–3118, 2018.

[9] C. Cortes and V. Vapnik, "Support-vector Networks," *Machine Learning*, vol. 20(3), pp. 273–297, 1995.

[10] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate Descent Method for Large-scale L2-loss Linear Support Vector Machines," *Journal of Machine Learning Research*, 2008.

[11] P. Helber, B. Bischke, A. Dengel, and D. Borth, "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," *Computing Research Repository - arXiv*, vol. abs/1709.0, aug 2017. [Online]. Available: http://arxiv.org/abs/1709.00029

[12] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," sep 2014. [Online]. Available: http://arxiv.org/abs/1409.1556