A Deep Learning Approach to UAV Image Multilabeling

Article *in* IEEE Geoscience and Remote Sensing Letters · March 2017

DOI: 10.1109/LGRS.2017.2671922

3 authors:

Abdallah Zeggada
Università degli Studi di Trento
10 PUBLICATIONS **34** CITATIONS
SEE PROFILE

Farid Melgani
Università degli Studi di Trento
185 PUBLICATIONS **5,779** CITATIONS
SEE PROFILE

Yakoub Bazi
King Saud University
76 PUBLICATIONS **2,099** CITATIONS
SEE PROFILE

A Deep Learning Approach to UAV Image Multilabeling

Abdallah Zeggada, *Student Member, IEEE*, Farid Melgani, *Fellow, IEEE*, and Yakoub Bazi, *Senior Member, IEEE*

***Abstract*— In this letter, we face the problem of multilabeling unmanned aerial vehicle (UAV) imagery, typically characterized by a high level of information content, by proposing a novel method based on convolutional neural networks. These are exploited as a means to yield a powerful description of the query image, which is analyzed after subdividing it into a grid of tiles. The multilabel classification task of each tile is performed by the combination of a radial basis function neural network and a multilabeling layer (ML) composed of customized thresholding operations. Experiments conducted on two different UAV image data sets demonstrate the promising capability of the proposed method compared to the state of the art, at the expense of a higher but still contained computation time.**

***Index Terms*— Convolutional neural networks (CNNs), image multilabeling, Otsu's algorithm, unmanned aerial vehicles (UAVs), urban monitoring.**

I. Introduction

THE ever-increasing interest witnessed in the acquisition and development of unmanned aerial vehicles (UAVs), commonly known as drones in the past few years, has paved the way to a very promising and effective technology. Since 2005, the number of countries that have acquired drones doubled from 40 to more than 75 [1]. UAVs have proven their effectiveness in collecting data over unreachable areas and limited coverage zones due to their small size and fast deployment. Moreover, their custom-made capacity allows them to collect information with a very high level of detail, leading to extremely high-resolution (EHR) images. UAVs were mainly created for military usage. However, in the last decade, they have being exploited in numerous civilian applications as well. For instance, in [2], a real time algorithm is introduced for classification, object detection and tracking from thermal UAV images acquired over the surface of the ocean. In [3], the authors present a UAV cloud system disaster surveillance system to reduce natural or man-made damages. Li *et al.* [4] introduce an unsupervised classification method for UAV images to detect earthquake triggered on rural houses. Furthermore, several works dealing with vehicle detection can be found in [5] and [6]. Kamate and Yilmazer [7] present a visual surveillance system for tracking moving objects in video sequences acquired by means of UAVs using Lucas-Kanade optical flow and continuously adaptive mean-shift techniques. In [8], a texture-based (i.e., energy, correlation, mean intensity, and lacunarity) classification method using Minkovski distance as a method of comparison was presented. In addition, UAVs have been used with promising results in various applications such as in the agricultural sector. In particular, Malek *et al.* [9] proposed an automatic method for palm tree detection using scale-invariant feature transform (SIFT) features and extreme learning machine (ELM) classifier. Moreover, Senthilnath *et al.* [10] describe a spectral-spatial method for the detection of tomatoes on UAV images, exploiting three different spectral clustering methods with spatial segmentation morphological operations applied on the target image.

In spite of the efforts being dedicated to UAV imagery classification and analysis within the remote sensing community, there is still a plenty of room for improvement. Indeed, as the spatial resolution increases, so does the need for new methods to process images with such high level of detail and rich information content, where traditional ways of classification such as pixel-based and segment-based descriptors may raise the problem of intraclass variability especially when dealing with several classes at the same time. Moreover, they dramatically increase the computational needs. Indeed, this makes the analysis of UAV imagery particularly challenging. In this letter, we deal with the problem of multilabel classification of EHR images acquired by means of UAVs using a coarse description approach. That is, instead of attributing a label to each individual spatial entity or segment region descriptor as in the traditional monolabel classification, we describe the considered entity by a list of object classes present in it. This approach first subdivides the image into a grid of equal tiles. Then using some specific tile representation and an opportune classification tool, to each tile, a vector of labels is assigned representing the object classes that are possibly present in it. Such a multilabel classification approach was first introduced in [11] for describing UAV images over urban areas with interesting results. In particular, the multilabel implementation derives benefits from exploiting local feature descriptors, such as SIFT, and histogram of oriented gradients (HoG), combined with a bag of visual words (BOW) compact representation.

Recently, the computer vision community has reported a very promising generation of neural networks, called convolutional neural networks (CNNs) [12]. They show that CNNs can overcome traditional classification methods in very complex vision tasks [13], [14]. In this letter, we propose an alternative to the new classification problem raised in [11] by: 1) representing tiles with CNN features and 2) substituting the matching paradigm adopted in [11] with a multilabel classification model based on a radial basis function (RBF) neural network (RBFNN) [15]. In particular, a multilabeling

Manuscript received September 19, 2016; revised December 28, 2016; accepted January 31, 2017. Date of publication March 22, 2017; date of current version April 20, 2017.

A. Zeggada and F. Melgani are with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy (e-mail: abdallah.zeggada@.unitn.it; melgani@disi.unitn.it).

Y. Bazi is with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia (e-mail: ybazi@ksu.edu.sa).

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.2017.2671922

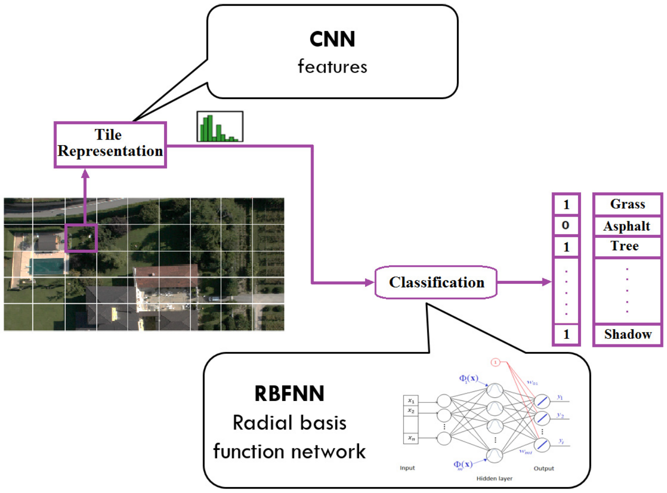


Fig. 1. Flowchart of the multilabel classification method.

layer, relying on a set of simple thresholding operations [16], is integrated on the top of the whole architecture to improve the obtained outcomes.

The remaining part of this letter is organized as follows. Section II details the proposed multilabel classification method. In Section III, we present the experimental results obtained on two real UAV image data sets acquired over urban areas and compare them with state-of-the-art results. Finally, Section IV draws some conclusions of the work and elaborates future developments.

II. Proposed Method

A. Image Coarse Description

Let us consider a three-channel red, green and blue (RGB) EHR image (I) acquired by means of UAVs. We start by subdividing it into a grid of tiles of equal sizes. The size of each tile is defined according to the spatial resolution of I and the expected sizes of objects that one aims at recognizing. The multiclass tile-based approach is composed of two main stages: 1) a suitable tile representation strategy and 2) a tile classification/matching method. A query tile is labeled either with a binary vector of the most similar tile present in the training library using a matching strategy, where the closest tile in the feature space from the training library has likely the same list of objects of the query tile, or as proposed in this letter it is labeled by means of a classification paradigm (Fig. 1).

In particular, our method starts with the extraction of features using the GoogLeNet pretrained CNN [13]. CNN is a feedforward hierarchical neural network implementing a set of convolutional and subsampling operations, followed by a softmax classifier. The main underlying idea behind CNN is to look for a pattern supposed to be invariant to spatial translations. The major difference between convolutional networks and a multilayer perceptron (MLP) is that, unlike MLP, the internal order of inputs and hidden units is relevant in convolutional networks, and each unit is connected with a specific spatial position of the input image. CNN architecture starts with a convolutional layer that is usually a 3-D volume of neurons. It consists of a set of feature detectors which refer to a convolution with a mask that is consistently convolved across the width and height of the input layer in order to extract robust features against noise and translation. The repetition of this mask all over the input layer allows to share the same weights all over the input layer which reduces the number of free parameters learned. In fact, this simplifies the computational requirements of training the network on large data sets building much efficient and powerful networks. After the convolutional layer, several operations are performed such as the activation function and subsampling, called also pooling. This last forms a nonlinear down-sampling layer that reduces the spatial size, and thus the number of parameters to be computed for the next layer. The most common pooling technique is Max-Pooling. It divides the input into a set of nonoverlapping blocks, and assigns to each block its maximum value. In order to increase sparseness, an elementwise activation function rectified linear units (ReLUs) layer can be applied after any convolutional layer. The ReLU layer deals also with the vanishing gradient problem in the error backpropagation phase. Thereafter, a set of convolutional and subsampling layers, comes the classification layer. It is a fully connected layer that has full connections to all activation units in the previous layers with a loss function (e.g., softmax).

Training a deep network usually requires a huge number of training images to avoid overfitting. Nevertheless, one may reasonably tackle this issue by transfer learning, which in our case consists of exploiting the weights of a model that is pertained on a large data set and making use of them while developing our data set-specific classifier. CNN features computed with pretrained networks have been used in many computer vision tasks, and have shown good results [17], [18].

One of the publicly available pretrained CNN is GoogLeNet. It was first trained over ILSVRC2014 data set, which contains over 1.2 million images, with a classification challenge of 1000 different classes, thus making GoogLeNet a promising candidate for generating powerful discrimination features. GoogLeNet is a 22-layers deep network excluding pooling layers, with a softmax loss layer as a classifier. The size of its receptive field is 224×224 of three channels (RGB) with zero mean. A ReLU activation function is used in all its convolutional layers. GoogLeNet generates a feature vector of size equal to 1024.

Since GoogLeNet is not directly adapted to multilabel classification tasks, in this letter, we substitute the softmax classifier with a RBFNN, which is a classifier that can fit our multilabeling requirement. Indeed, we will violate the principle that the sum of output targets should be equal to one (i.e., just one of them is active) ruling traditional (mono-label) classification problems. We will look at the outputs of the RBFNN no more as posteriors but as indicators of presence/absence of the corresponding object. This means that during the training phase the classifier will model which objects are present/absent in each training tile. During the prediction phase, the model will provide for each object a quantity (indicator) $f_i(X)$ from which we will need to infer the presence of absence of the considered object. Since during training, the values used to indicate the presence or the absence of an object are set to $f_i(X) = 1$ or 0, respectively, during the prediction an intuitive decision mechanism is "the object is present if $f_i(X) \geq 0.5$, otherwise it is absent." We propose to substitute this intuitive decision rule by integrating a multilabeling layer, which will be part of the architecture (Fig. 2). In particular, each indicator $f_i(X)$ will be viewed as a feature along which two hypotheses H_0 (absence) and H_1 (presence) are defined. The problem of discrimination between H_0 and H_1

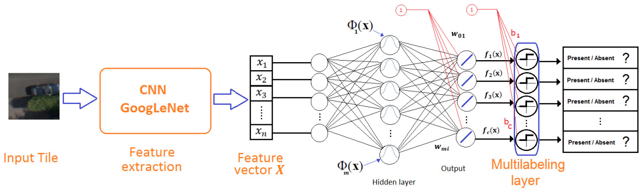


Fig. 2. Global flowchart of the proposed classification scheme.

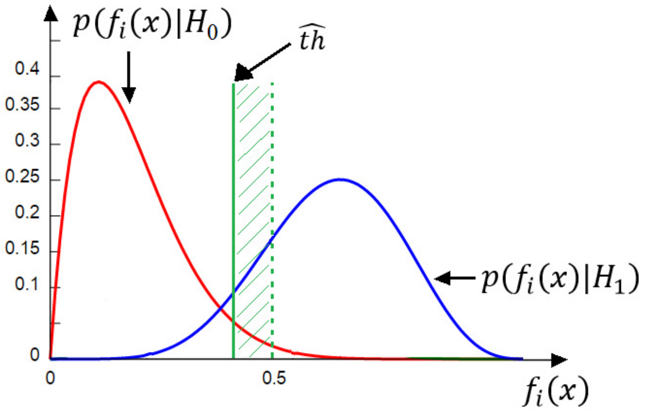


Fig. 3. Graphical histogram illustration of the OTSU thresholding technique.

can be seen as a simple thresholding problem. In the literature, there exist several algorithms for computing the best threshold between two classes. In the following, we briefly introduce a simple and fast algorithm, called Otsu's method [16], which will be exploited in this letter.

B. Otsu's Thresholding Algorithm

This algorithm is an unsupervised method, which finds a decision threshold $\widehat{th}$ between two hypothesis (classes) H_0 (absence of object), H_1 (presence of object) based on a discriminant criterion aiming at maximizing the separability between the two classes and thus minimizing their intraclass variance (Fig. 3). Let $f_i(X)$ be an output function (feature) of a deep network architecture represented by a 1-D histogram composed of M bins. This last is transformed by normalization into a probability function $p(f_i(X))$.

Let us assume that along $f_i(X)$ just two classes lie, namely H_0 and H_1. We are interested in finding a threshold value t that best separates the two classes. For a given threshold value t, the prior probabilities of H_0 and H_1 can be computed as follows:

$$P(H_0(t)) = \sum_{i=0}^{t-1} p(i) \tag{1}$$

$$P(H_1(t)) = \sum_{i=t}^{M-1} p(i). \tag{2}$$

The main idea behind Otsu's method is to select the threshold $\widehat{th}$ that minimizes the intraclass variance of the two classes H_0, H_1 which is but the weighted sum of variances of each cluster defined as

$$\sigma_W^2(t) = P(H_0(t))\sigma_0^2(t) + P(H_1(t))\sigma_1^2(t) \tag{3}$$

where $\sigma_1^2(t)$ and $\sigma_0^2(t)$ are the variance the pixels above and below the threshold t (thus an approximation of the variance of the classes H_0 and H_1), respectively. Alternatively, we may express the minimization process in terms of the between-class variance $\sigma_B^2(t)$, which is defined as the subtraction of the within-class variance $\sigma_W^2(t)$ from the total variance of their combined distribution σ^2 given by

$$\begin{aligned}\sigma_B^2(t) &= \sigma^2 - \sigma_W^2(t)\\ &= P(H_0(t))(\mu_0(t) - \mu_T)^2 + P(H_1(t))(\mu_1(t) - \mu_T)^2\\ &= P(H_0(t))P(H_1(t))[\mu_0(t) - \mu_1(t)]^2\end{aligned} \tag{4}$$

where the class means are

$$\mu_0(t) = \sum_{i=0}^{t-1} ip(i)/P(H_0) \tag{5}$$

$$\mu_1(t) = \sum_{i=t}^{M-1} ip(i)/P(H_1) \tag{6}$$

and the total mean level is defined as

$$\mu_T = \sum_{i=0}^{M-1} ip(i). \tag{7}$$

It can easily be verified that

$$P(H_0) + P(H_1) = 1, \quad P(H_0)\mu_0 + P(H_1)\mu_1 = \mu_T. \tag{8}$$

The best threshold $\widehat{th}$ that minimizes the within-class variance σ_W^2 is selected as

$$\widehat{th} = \underset{0 \le t \le M-1}{max} \sigma_B^2(t). \tag{9}$$

For every bin t (candidate value for the best threshold) of the histogram, we thus compute the between-classes variance $\sigma_B^2(t)$ and we choose the optimum threshold $\widehat{th}$ that maximizes it. This process is repeated for each output $f_i(X)$ of the RBFN network in order to find the best threshold value for each object and therefore to complete the training of the multilabeling layer. This means that: 1) for each output a histogram needs to be generated and 2) Otsu's algorithm is applied on each histogram to estimate the best decision threshold for the corresponding class.

III. Experimental Validation

In this section, we evaluate the classification performances of the proposed method on two real data sets of UAV images acquired over two different locations. The first set of images was taken over the Faculty of Science of the University of Trento (Italy) Nadir acquisition on October 3, 2011 at 12:00 A.M. The second set of images was acquired near the city of Civezzano (Italy) at different off-nadir angles, on October 17, 2012. Both Acquisitions were performed with a picture camera Canon EOS 550D characterized by a CMOS APS-C sensor with 18 megapixels. The UAV images are characterized by three channels (RGB) with a spatial resolution of approximately 2 cm. The image size is 5184 × 3456 pixels and the radiometric resolution is 8 b for both data sets. The first data set is composed of nine images, subdivided into two groups.

1) *Training Set:* Two images are selected as training images. We extracted randomly 1000 tiles of size 224 × 224 from both images. The two training images were chosen from the overall set in such a way

TABLE I

COMPARISON OF CLASSIFICATION ACCURACIES IN TERMS OF SENSITIVITY (SENS) AND SPECIFICITY (SPEC) BETWEEN THE DIFFERENT IMPLEMENTATIONS. COMPUTATIONAL TIME PER TILE IS ALSO REPORTED FOR EACH STRATEGY

	Dataset 1						Dataset 2					
	Accuracy(%)			Time(ms)			Accuracy(%)			Time(ms)		
METHOD	Spec	Sens	Average	Tile Representation	Classification / Matching	Total	Spec	Sens	Average	Tile Representation	Classification / Matching	Total
HCR-B [11]	92.2	60.7	76.4	**32**	7	**39**	91.9	61.4	76.6	**32**	7	**39**
GoogLeNet (MLR)	92.7	60.8	76.8	90	**0.3**	90	92.9	58.7	75.8	90	**0.3**	90
AlexNet(MLR)	94.5	60.4	77.5	40	0.8	41	94.0	58.0	76.0	40	0.9	41
GoogLeNet -RBFNN	95.4	63.1	79.3	90	2	92	**96.1**	58.7	77.4	90	2	92
AlexNet -RBFNN	96.2	60.6	78.4	40	5	45	95.4	58.3	76.9	40	5	45
GoogLeNet _SVM(linear)	**96.4**	59.0	77.7	90	41	131	**96.1**	58.1	77.1	90	53	143
GoogLeNet_SVM(RBF)	96.3	58.6	77.5	90	39	129	95.1	53.6	74.4	90	51	141
AlexNet -SVM(linear)	95.5	54.5	75.0	40	131	171	95.2	52.3	73.7	40	143	183
AlexNet -SVM(RBF)	95.7	52.5	74.1	40	124	164	95.2	52.3	73.8	40	144	184
GoogLeNet -RBFNN (ML)	90.3	**75.1**	**82.7**	90	2	92	92.6	**68.6**	**80.6**	90	2	92
AlexNet -RBFNN(ML)	93.0	**70.5**	**81.8**	40	5	45	93.2	**66.1**	**79.7**	40	5	45

they contain all predefined classes of objects, which are “Asphalt,” “Grass,” “Tree,” “Vineyard,” “Pedestrian Crossing,” “Person,” “Car,” “Roof 1,” “Roof 2,” “Solar Panel,” “Building Facade,” “Soil,” and “Shadow.”

2) *Test Set:* It is composed of seven images. We subdivided each test image into a nonoverlapping grid of equal tiles of 224 × 224 pixels as explained in the previous section. The second data set is composed by ten images, subdivided into two groups.
3) *Training Set:* Three images are selected as training. We extracted randomly 1000 tiles of size 224 × 224 from the three images. The three training images were chosen from the overall set in such a way they contain all predefined classes of objects, which are “Asphalt,” “Grass,” “Tree,” “Vineyard,” “Low Vegetation,” “Car,” “Roof 1,” “Roof 2,” “Roof 3,” “Solar Panel,” “Building Facade,” “Soil,” “Gravel,” and “Rocks.”
4) *Test Set:* It is composed of seven images. We subdivided each test image into a nonoverlapping grid of equal tiles of 224 × 224 pixels.

For both training sets, we rotated each tile randomly with one of the following four angle values: 0°, 90°, 180°, and 270°. Given the image resolution of 2 cm, the tile size covers 4.5 m × 4.5 m.

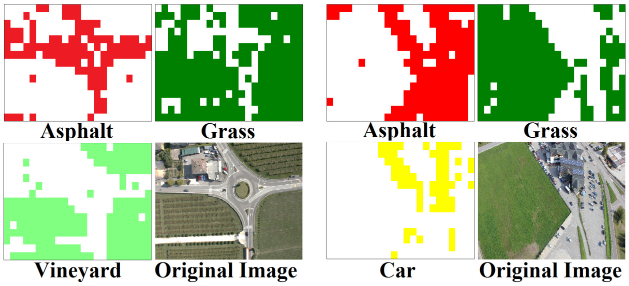
Regarding the accuracy evaluation, we adopted the sensitivity and specificity metrics in order to compare our method with a reference one [11]. This latter consists in using RGB and HoG features combined with a BOW for compact tile representation, and the chi-squared measure distance (χ^2) as a matching strategy (each test tile is labeled with the same binary vector of the most similar tile present in the training tiles library). As for comparing the proposed scheme with other state-of-the-art CNN-based models, we investigate also the AlexNet architecture [18] (Caffe version), which is another very popular pretrained model. It contains five convolutional layers with three fully-connected layers. AlexNet generates a feature vector of size equal to 4096. In order to deal with our multilabeling requirement, we substitute the original multinomial logistic regression (Softmax) in AlexNet and GoogLeNet with a multilabel logistic regression (MLR) prediction model, which consists of as many binary logistic classifiers as the number of labels (classes). In order to enrich further the comparative study, we added two other strategies, which are based on feeding the CNN features (i.e., AlexNet, GoogLeNet) to multiclass (one-against-all) support vector machines (SVMs), implemented with both linear and RBF kernels. All the experiments were conducted on an Intel Xeon E3-1246 CPU at 3.5 GHz with 32-GB RAM using a MATLAB platform.

The quantitative comparison results are summarized in Table I. As shown, the combinations of CNN features (i.e., AlexNet, GoogLeNet) with RBFNN classifier achieved in general better results in terms of average accuracy than the combination of the same features with the other three classifiers (i.e., MLR, linear SVM, and SVM with the Gaussian RBF kernel). GoogLeNet-RBFNN and AlexNet-RBFNN score 79.3% and 78.4% of average accuracy respectively for data set 1, and 77.4% and 76.9% of average accuracy for data set 2. Furthermore, this combination strategy overcomes the reference method [11] in data set 1 with an increment of around 3% for GoogLeNet and 2% for AlexNet (in terms of average accuracy), and slightly overcomes it in data set 2 with an increment of 0.8% and 0.3% of average accuracy for GoogLeNet and AlexNet, respectively. This stresses the promising discriminative capabilities of the deep and hierarchical feature representation process implemented by GoogLeNet and AlexNet.

A further refinement of the results has been possible thanks to the addition of a multilabeling layer of step functions at the top of the RBFNN. As explained earlier, the parameter (i.e., bias) of these functions is estimated by applying the very fast Otsu’s algorithm on the output of the training RBFNN. In particular, for each RBFNN $f_i(X)$ output, we constructed a histogram of 30 bins covering the range from -1.5 to $+1.5$ from the available training tiles. The obtained bias values of the multilabeling layer are reported in Table II, which shows that the values range from 0.14 to 0.49 depending on the class, stressing thus the importance of customizing the threshold value to each kind of object. The final results on the test tiles of data sets 1 and 2 are reported in Table I. As can be seen, a significant boost of accuracy was possible by adding the multilabeling layer on top of the RBFNN outputs. In particular, there is a clear average accuracy improvement that comes at the cost of some loss in the specificity. For instance, for GoogLeNet-RBFNN scheme in data set 1, despite the decrease in specificity of roughly 5% from 95.4% to 90.3%, there is a higher gain of almost 12% in terms of sensitivity from 63.1% to 75.1%. The same improvement can be noticed for all CNN feature combinations with RBFNN in both data sets. The multilabeling layer exhibits the advantage that it reduces the risk of missing objects (estimated threshold values are all less than 0.5) resulting thus in a globally better prediction. The

TABLE II
BEST THRESHOLD VALUES YIELDED BY THE OTSU'S METHOD FOR EACH OF THE RBFNN OUTPUT CLASSES

	Bias values													
class	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}
GoogLeNet (Dataset 1)	0.38	0.45	0.50	0.26	0.36	0.33	0.42	0.46	0.32	0.34	0.37	0.38	0.38	---
AlexNet (Dataset 1)	0.41	0.43	0.44	0.20	0.38	0.15	0.46	0.43	0.32	0.36	0.37	0.38	0.38	---
GoogLeNet (Dataset 2)	0.49	0.44	0.39	0.26	0.27	0.43	0.34	0.34	0.38	0.38	0.35	0.36	0.35	0.37
AlexNet (Dataset 2)	0.44	0.47	0.43	0.34	0.36	0.37	0.41	0.43	0.32	0.44	0.45	0.42	0.27	0.41



(a) (b)

Fig. 4. Partial multilabel classification maps obtained by the RBFNN (ML) method on one of the test images from (a) data set 1 and (b) data set 2. Note that a full multilabel map would report all classes.

obtained results clearly illustrate the usefulness of exploiting the distributions of the RBFNN outputs from the training tiles for customizing the decision process and therefore improving the results of the multilabel classification task. Fig. 4(a) and (b) illustrate some multilabel qualitative results yielded by the proposed method for data sets 1 and 2, respectively. The multilabel map allows depicting the spatial distribution of the object classes at a tile level.

In terms of processing time required per single tile (see Table I), the reference method [11] reports to be 6 ms faster than the proposed AlexNet-RBFNN architecture (39 against 45 ms, respectively) and shows about two times faster than GoogLeNet-RBFNN (39 against 92 ms, respectively). Indeed, CNN networks, in particular GoogLeNet, perform much more computations due to the large number of layers and blocks composing each layer compared to the HCR-B strategy which uses fast local feature descriptors. As expected, AlexNet-RBFNN turned out to be two times faster than GoogLeNet-RBFNN. In fact, GoogLeNet architecture is deeper than that of AlexNet's, resulting in more processing needs.

IV. CONCLUSION

In this letter, we have proposed a new multilabel classification method for UAV images. The proposed model starts by subdividing the image into a set of equal tiles, each described by a list of objects present in it. CNN features are extracted from each tile using the GoogLeNet pretrained network. Then a RBFNN model is trained for the classification task. For the multilabeling issue, a multilabel layer has been integrated on top of the whole network to improve the obtained results. The proposed method can achieve substantial classification accuracy gains over the state of the art. From the results, one can infer that our method is rather promising for EHR UAV multilabeling applications. Moreover, we believe that this multilabel classification framework opens the door to exploit various other alternative solutions for the tile representation and the classification/matching steps.

REFERENCES

[1] *Nonproliferation: Agencies Could Improve Information Sharing And End-Use Monitoring On Unmanned Aerial Vehicle Exports*, U.S. Government Accountability Office, Washington, DC, USA, Jul. 2012.

[2] F. S. Leira, T. A. Johansen, and T. I. Fossen, "Automatic detection, classification and tracking of objects in the ocean surface from UAVs using a thermal camera," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, pp. 1–10, 2015.

[3] C. Luo, J. Nightingale, E. Asemota, and C. Grecos, "A UAV-cloud system for disaster sensing applications," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, Glasgow, Scotland May 2015, pp. 1–5.

[4] S. Li *et al.*, "Unsupervised detection of earthquake-triggered roof-holes from UAV images using joint color and shape features," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1823–1827, Sep. 2015.

[5] T. Moranduzzo and F. Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6356–6367, Oct. 2014.

[6] K. Liu and G. Mattyus, "Fast multiclass vehicle detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 9, pp. 1938–1942, Sep. 2015.

[7] S. Kamate and N. Yilmazer, "Application of object detection and tracking techniques for unmanned aerial vehicles," *Proc. Comput. Sci.*, vol. 61, pp. 436–441, Nov. 2015.

[8] D. Popescu and I. Loretta, "Image recognition in UAV application based on texture analySIS," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Cham, Switzerland: Springer, 2015, pp. 693–704.

[9] S. Malek, Y. Bazi, N. Alajlan, H. AlHichri, and F. Melgani, "Efficient framework for palm tree detection in UAV images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 12, pp. 4692–4703, Dec. 2014.

[10] J. Senthilnath, A. Dokania, M. Kandukuri, K. N. Ramesh, G. Anand, and S. N. Omkar, "Detection of tomatoes using spectral-spatial methods in remotely sensed RGB images captured by UAV," *Biosystems Eng.*, vol. 146, pp. 16–32, Jun. 2016.

[11] T. Moranduzzo, F. Melgani, M. L. Mekhalfi, Y. Bazi, and N. Alajlan, "Multiclass coarse analysis for UAV imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6394–6406, Dec. 2015.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[13] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[14] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1529–1537.

[15] C. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford Univ. Press, 1995, pp. 164–190.

[16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man., Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jun. 1979.

[17] R. F. Nogueira, R. D. A. Lotufo, and R. C. Machado, "Fingerprint liveness detection using convolutional neural networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1206–1213, Jun. 2016.

[18] N. Tajbakhsh *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 1106–1114.