Aerial scene understanding in the wild: Multi-scene recognition via prototype-based memory networks

Yuansheng Hua\textsuperscript{a, b}, Lichao Mou\textsuperscript{a, b}, Jianzhe Lin\textsuperscript{c}, Konrad Heidler\textsuperscript{a, b}, Xiao Xiang Zhu\textsuperscript{a, b, *}

\textsuperscript{a} Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Weßling, Germany
\textsuperscript{b} Data Science in Earth Observation (SiPEO), Technical University of Munich (TUM), Arcistr. 21, 80333 Munich, Germany
\textsuperscript{c} Electrical and Computer Engineering (ECE), University of British Columbia (UBC), V6T 1Z2, Canada

\textbf{ABSTRACT}

Aerial scene recognition is a fundamental visual task and has attracted an increasing research interest in the last few years. Most of current researches mainly deploy efforts to categorize an aerial image into one scene-level label, while in real-world scenarios, there often exist multiple scenes in a single image. Therefore, in this paper, we propose to take a step forward to a more practical and challenging task, namely multi-scene recognition in single images. Moreover, we note that manually yielding annotations for such a task is extraordinarily time- and labor-consuming. To address this, we propose a prototype-based memory network to recognize multiple scenes in a single image by leveraging massive well-annotated single-scene images. The proposed network consists of three key components: 1) a prototype learning module, 2) a prototype-inhabiting external memory, and 3) a multi-head attention-based memory retrieval module. To be more specific, we first learn the prototype representation of each aerial scene from single-scene aerial image datasets and store it in an external memory. Afterwards, a multi-head attention-based memory retrieval module is devised to retrieve scene prototypes relevant to query multi-scene images for final predictions. Notably, only a limited number of annotated multi-scene images are needed in the training phase. To facilitate the progress of aerial scene recognition, we produce a new multi-scene aerial image (MAI) dataset. Experimental results on variant dataset configurations demonstrate the effectiveness of our network. Our dataset and codes are publicly available\textsuperscript{1}.

\textbf{1. Introduction}

With the enormous advancement of remote sensing technologies, massive high-resolution aerial images are now available and beneficial to a large variety of applications, e.g., urban planning (Marmanis et al., 2018; Audebert et al., 2018; Marcos et al., 2018; Mou and Zhu, 2018b; Li et al., 2018; Qiu et al., 2020; Li et al., 2020b), traffic monitoring (Mou and Zhu, 2018c; Mou and Zhu, 2016), disaster assessment (Vetrivel et al., 2018; Lee et al., 2017), and natural resource management (Lucchesi et al., 2013; Weng et al., 2018; Cheng et al., 2017; Zarco-Tejada et al., 2014; Wen et al., 2017; Mou and Zhu, 2018a; Qiu et al., 2019). Driven by these applications, aerial scene recognition that refers to assigning aerial images scene-level labels is now becoming a fundamental but challenging task.

In recent years, many efforts (Zhu et al., 2017), e.g., developing novel network architectures (Murray et al., 2019; Cheng et al., 2020; Bi et al., 2020; Niazamardi et al., 2017; Lin et al., 2020; Zhu et al., 2018) and pipelines (Byju et al., 2000; Xu et al., 2020; Wang et al., 2019; Zhu et al., 2019), publishing large-scale datasets (Xia et al., 2017; Jin et al., 2018), introducing multi-modal and multi-temporal data (Hu et al., 2020; Tuia et al., 2016; Ru et al., 2020; Li et al., 2020a), have been deployed to address this task, and most of them treat it as a single-label classification problem. A common assumption shared by these researches is that an aerial image belongs to only one scene category, while in real-world scenarios, it is more often that there exist various scenes in a single image (cf. Fig. 1). Furthermore, we notice that aerial images used to learn single-label scene classification models are usually well-cropped so that target scenes could be centered and account for the majority of an

\textsuperscript{1} Corresponding author at: Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Weßling, Germany.

E-mail addresses: yuansheng.hua@dlr.de (Y. Hua), lichao.mou@dlr.de (L. Mou), jianzhe.lin@ece.ubc.ca (J. Lin), konrad.heidler@dlr.de (K. Heidler), xiaoxiang.zhu@dlr.de (X.X. Zhu).

\url{https://doi.org/10.1016/j.isprsjprs.2021.04.006}

Received 11 November 2020; Received in revised form 7 April 2021; Accepted 9 April 2021
Available online 16 May 2021

\url{https://github.com/Hua-YS/Prototype-based-Memory-Network}.
aerial image. Unfortunately, this might be infeasible for practical applications. Therefore, in this paper, we aim to deal with a more practical and challenging problem, multi-scene classification in a single image, which refers to inferring multiple scene-level labels for a large-scale, unconstrained aerial image. Fig. 1 shows an example image, where we can see that multiple scenes, e.g., residential, parking lot, and commercial, co-exist in one aerial image. We note that there is another research branch of aerial image understanding, multi-label object classification, which refers to the process of inferring multiple objects present in an aerial image. These studies (Sumbul and Demir, 2019; Zegeye and Demir, 2018; Hua et al., 2020; Khan et al., 2019; Hua et al., 2019; Zeggada et al., 2017; Koda et al., 2018) mainly focus on recognizing object-level labels, while in our task, an image is classified into multiple scene categories, which provides a more comprehensive understanding of large-scale aerial images in scene-level. To the best of our knowledge, multi-scene recognition in unconstrained aerial images still remains underexplored in the remote sensing community.

To achieve this task, huge quantities of well-annotated multi-scene images are needed for the purpose of training models. However, we note that such annotations are not easy in the remote sensing community. This could be attributed to the following two reasons. On the one hand, the visual interpretation of multiple scenes is more arduous than that of a single scene in an aerial image, and therefore, labeling multi-scene images requires more work. On the other hand, low-cost annotation techniques, e.g., resorting to crowdsourcing OpenStreetMap (OSM) through keyword searching (Xia et al., 2017; Jin et al., 2018; Long et al., 2020), perform poorly in yielding multi-scene datasets owing to the incompleteness and incorrectness of certain OSM data. Examples of erroneous OSM data are shown in Fig. 2. In addition, manually rectifying annotations generated from crowdsourcing data are inevitable due to error-proneness. Such a procedure is quite labor-consuming, as every scene is required to be checked in case that present ones are mislabeled as absent. Aiming to solve the aforementioned limitations, in this work, we propose to train a network for recognizing complex multi-scene aerial images by using only a small number of labeled multi-scene images but a huge amount of existing, annotated single-scene data. Our motivation is based on an intuitive observation about how humans learn to perceive complex scenes being composed of multiple entities (National Research Council, 2000; Liu et al., 2008; McLaren and Suret, 2000): we first learn and memorize individual objects (through flash cards for example) when we were babies and then possess the capability of understanding complex scenarios by learning from only a limited number of hard instances (cf. Fig. 1). We believe that this learning process also applies to the interpretation of multi-scene aerial images. Driven by this observation, we propose a novel network, termed as prototype-based memory network (PM-Net), which is inspired by recent successes of memory networks in natural language processing (NLP) tasks (Sukhbaatar et al., 2015; Miller et al., 2016) and video analysis (Shi et al., 2019; Park et al., 2020; Lai et al., 2020). To be more specific, we first learn the prototype representation of each aerial scene from single-scene aerial images and then store these prototypes in the external memory of PM-Net. Afterwards, for a given query multi-scene image, a multi-head attention-based memory retrieval module is devised to retrieve scene prototypes that are associated with the query image from the external memory for inferring multiple scene labels (see Fig. 3).
The contributions of this work are fourfold.

- We take a step forward to a more practical and challenging task in aerial scene understanding, namely multi-scene classification in single images, which aims to recognize multiple scenes present in a large-scale, unconstrained aerial image. Such a task is in line with real-world scenarios and capable of providing a comprehensive picture for a given geographic area.
- Given that labeling multi-scene images is very labor-intensive and time-consuming, we propose a PM-Net that can be trained for our task by leveraging large numbers of existing single-scene aerial images and a small number of labeled multi-scene images.
- In order to facilitate the progress of multi-scene recognition in single aerial images, we create a new dataset, multi-scene aerial image (MAI) dataset. To the best of our knowledge, this is the first publicly available dataset for aerial multi-scene interpretation. Compared to existing single-scene aerial image datasets, images in our dataset are unconstrained and contain multiple scenes, which are more in line with the reality.
- We carry out extensive experiments with different configurations. Experimental results demonstrate the effectiveness of the proposed network.

The remaining sections of this paper are organized as follows. Section 2 reviews studies in memory networks and prototypical networks, and the architecture of the proposed prototype-based memory network is introduced in Section 3. Section 4 describes experimental configurations and analyzes results. Eventually, conclusions are drawn in Section 5.

2. Related work

Since very few efforts have been deployed to this task in the remote sensing community, we only review literatures related to our algorithm in this section.

2.1. Memory networks

A memory network takes as input a query and retrieves complementary information from the external memory. In Sukhbaatar et al. (2015), the memory network is first proposed and utilized to address question-answering tasks, where questions are regarded as queries, and statements are stored in the external memory. To retrieve statements for predicting answers, the authors compute relative distances between queries and the external memory through dot product. In the following work, Miller et al. (2016) improves the efficiency of retrieving large memories by pre-selecting small subsets with key hashing. Moreover, the memory network is further applied in video analysis (Shi et al., 2019; Park et al., 2020; Lai et al., 2020) and image captioning (Cornia et al., 2020). In Shi et al. (2019), the authors devise a dual augmented memory network to memorize both target and background features of an video, and use a Long Short-Term Memory (LSTM) to communicate with previous and next frames. In Park et al. (2020), the authors propose a memory network to memorize normal patterns for detecting anomalies in an video. As an attempt in image captioning, Cornia et al. (2020) devise a learnable memory to learn and memorize priori knowledge for encoding relationships between image regions. Inspired by these works, we devise a memory network and store scene prototypes in the memory for recognizing scenes present in multi-scene images.

2.2. Prototypical networks

Prototypical networks are characterized by classifying images according to their distances from class prototypes. In learning with limited training samples, such networks are popular and achieved many successes recently (Snell et al., 2017; Guerriero et al., 2018; Yang et al., 2018; Huang et al., 2020; Zhang et al., 2020; Tang et al., 2019). To be specific, Snell et al. (2017) propose to first learn a prototype representation for each category and then identify images by finding their nearest category prototypes. Guerriero et al. (2018) aim to alleviate the heavy expense of learning prototypes by initializing and updating prototypes with those learned in previous training epochs. Yang et al. (2018) propose to combine prototypical networks and CNNs for tackling the open world recognition problem and improving the robustness and accuracy of networks. Similarly, Huang et al. (2020) propose to integrate prototypical networks and graph convolutional neural networks for learning relational prototypes. Albeit variant, most existing works share a common way to extract prototypes, which is taking average of training samples, such networks are popular and achieved many successes recently (Snell et al., 2017; Guerriero et al., 2018; Yang et al., 2018; Huang et al., 2020; Zhang et al., 2020; Tang et al., 2019). To be specific, Snell et al. (2017) propose to first learn a prototype representation for each category and then identify images by finding their nearest category prototypes. Guerriero et al. (2018) aim to alleviate the heavy expense of learning prototypes by initializing and updating prototypes with those learned in previous training epochs. Yang et al. (2018) propose to combine prototypical networks and CNNs for tackling the open world recognition problem and improving the robustness and accuracy of networks. Similarly, Huang et al. (2020) propose to integrate prototypical networks and graph convolutional neural networks for learning relational prototypes. Albeit variant, most existing works share a common way to extract prototypes, which is taking average of samples belonging to the same categories. Therefore, we follow this prototype extraction strategy in our work.

3. Methodology

3.1. Overview

The proposed PM-Net consists of three essential components: a prototype learning module, an external memory, and a memory retrieval module. Specifically, the prototype learning module is devised to encode prototype representations of aerial scenes, which are then stored in the external memory. The memory retrieval module is responsible for retrieving scene prototypes related to query images through a multi-head attention mechanism. Eventually, retrieved scene prototypes are
utilized to infer the existence of multiple scenes in the query image.

3.2. Scene prototype learning and writing

Following the observation introduced in Section 1, we propose to learn and memorize scene prototypes with the support of single-scene aerial images. The procedure consists of two stages. We first employ an embedding function to learn semantic representations of all single-scene images. Then, feature representations belonging to the same aerial images. The procedure consists of two stages. We first employ an embedding function to learn semantic representations of all single-scene images. Then, feature representations belonging to the same scene category are encoded into a scene prototype and stored in the external memory.

Formally, let $X_i^s$ denote the i-th single-scene image belonging to scene $s$, and $i$ ranges from 1 to $N_s$. $N_s$ is the number of samples annotated as $s$. The embedding function $f_b$ can be learned via the following objective function:

$$L(X_i^s, y^s) = -y^s \log \sum_{s} \exp(-g_b(f_b(X_i^s)))$$

where $\phi$ represents learnable parameters of $f_b$, and $y^s$ is a one-hot vector denoting the scene label of $X_i^s$. $g_b$ is a multilayer perceptron (MLP) with parameters $\theta$ and its outputs are activated by a softmax function to predict probability distributions. Following the overwhelming trend of deep learning, here we employ a deep CNN, e.g., ResNet-50 (He et al., 2016), as the embedding function $f_b$ and learn its parameters on public single-scene aerial image datasets. After sufficient training, $f_b$ is expected to be capable of learning discriminative representations for different aerial scenes.

Once $f_b$ is learned, the scene prototype can be computed by averaging representations of all aerial images belonging to the same scene (Snell et al., 2017; Guerriero et al., 2018; Yang et al., 2019). Let $p_s$ be the prototype representation of scene $s$. We calculate $p_s$ with the following equation:

$$p_s = \frac{1}{N_s} \sum_{i=1}^{N_s} f_b(X_i^s).$$

By doing so, in the single-scene classification, an image closely around $p_s$ in the common embedding space is supposed to belong to scene $s$. Similarly, in the multi-scene scenario, the representation of an aerial image comprising scene $s$ should show high relevance with $p_s$. After encoding all scene prototypes, the external memory $M$ can be formulated as follows:

$$M = [p_1, p_2, \ldots, p_S]^T,$$

where $S$ denotes the number of scenes. $[\ldots, \ldots]$ represents the concatenation operation. Given that $p_s$ is a D-dimensional vector, $M$ is a matrix of $S \times D$. Note that $D$ varies when using different backbone CNNs as embedding functions.

3.3. Multi-head attention-based memory retrieval

Inspired by successes of the multi-head self-attention mechanism (Vaswani et al., 2017) in natural language processing tasks (Radford et al., 2018; Radford et al., 2019; Devlin et al., 2018; Wolf et al., 2020), we develop a multi-head attention-based memory retrieval module to retrieve scene prototypes from the memory $M$ for a given query image $X$. Given a query multi-scene aerial image $X$, to retrieve relevant scene prototypes from $M$, we develop a multi-head attention-based memory retrieval module. In particular, we first extract the feature representation of $X$ through the same embedding function $f_b$ and linearly project it to an $L$-dimensional query $Q(X)$. Similarly, we transform the external memory $M$ into key $K(M)$ and value $V(M)$, and both are implemented as MLPs. The channel dimension of the key is $L$, while that of the value is $U$. The relevance between $X$ and each scene prototype $p_s$ can be measured by dot product similarity and a softmax function as follows:

$$R(X, M) = \text{softmax}(\frac{Q(f_b(X)) \cdot K(M)^T}{\sqrt{L}}).$$

The output is an $S$-dimensional vector, where each component represents a relevance probability that a specific scene prototype is related to the query image. Subsequently, the retrieved scene prototypes are computed by weight-summing all values with the following equation:

$$z = R(X, M) \cdot V(M).$$

Since the memory retrieval is designed in a multi-head fashion, the final retrieved prototype is reformulated as follows:

$$z' = [z_1, z_2, \ldots, z_n],$$

where $H$ denotes the number of heads, and each head yields a retrieved prototype $z_h$ by transforming $X$ and $M$ to the variant query $Q_h(f_b(X))$, key $K_h(M)$, and value $V_h(M)$. Eventually, the output $z'$ is fed into a fully-connected layer followed by a sigmoid function for inferring presences of aerial scenes.

3.4. Implementation details

For a comprehensive assessment of our PM-Net, we implement the embedding function with various backbone CNNs. Specifically, we conduct experiments on four CNN architectures, and details are as follows:

- **PM-VGGNet**: $f_b$ is built on VGG-16 (Simonyan and Zisserman, 2014) by replacing all layers after the last max-pooling layer in block5 with a global average pooling layer.
- **PM-Inception-V3**: Inception-V3 (Szegedy et al., 2015) is utilized, and layers before and including the global average pooling layer are employed as $f_b$.
- **PM-ResNet**: We modify ResNet-50 (He et al., 2016) by discarding layers after the last max-pooling layer and using the remaining layers as $f_b$.
- **PM-NASNet**: The backbone of $f_b$ is mobile NASNet (Zoph and Le, 2017). As with the modification in PM-ResNet, only layers before and including the global average pooling layer are used.

In our experiments, we train original deep CNNs on single-scene aerial image datasets and then take them as the embedding function $f_b$ following the aforementioned points. Subsequently, we yield scene prototypes $p_s$ and concatenate all of them along the first axis to form $M$.

4. Experiments and discussion

In this section, we introduce a newly produced multi-scene aerial image dataset, MAI dataset, and two single-scene datasets, i.e., UCM and AID datasets, which are used in experiments. Then network configurations and training schemes are detailed in SubSection 4.2. The remaining subsections discuss and analyze the performance of the proposed network thoroughly.

4.1. Dataset description and configuration

4.1.1. MAI dataset

To facilitate the progress of aerial scene interpretation in the wild, we yield a new dataset, MAI dataset, by collecting and labeling 3923 large-scale images from Google Earth imagery that covers the United
States, Germany, and France. The size of each image is $512 \times 512$ pixels, and their spatial resolutions vary from 0.3 m/pixel to 0.6 m/pixel. After capturing aerial images, we manually assign each image multiple scene-level labels from in total 24 scene categories, including apron, baseball, woodland, parking lot, and tennis court; commercial, parking lot, and residential; woodland, residential, river, and runway; river and storage tanks; beach, woodland, residential, and sea; farmland, woodland, and residential; apron and runway; baseball field, parking lot, residential, bridge, and soccer field.

Fig. 4. Example images in our MAI dataset. Each image is $512 \times 512$ pixels, and their spatial resolutions range from 0.3 m/pixel to 0.6 m/pixel. We list their scene-level labels here: (a) farmland and residential; (b) baseball, woodland, parking lot, and tennis court; (c) commercial, parking lot, and residential; (d) woodland, residential, river, and runway; (e) river and storage tanks; (f) beach, woodland, residential, and sea; (g) farmland, woodland, and residential; (h) apron and runway; (i) baseball field, parking lot, residential, bridge, and soccer field.

Fig. 5. Statistics of the proposed MAI dataset for multi-scene classification in single aerial images.

States, Germany, and France. The size of each image is $512 \times 512$, and spatial resolutions vary from 0.3 m/pixel to 0.6 m/pixel. After capturing aerial images, we manually assign each image multiple scene-level labels from in total 24 scene categories, including apron, baseball, beach, commercial, farmland, woodland, parking lot, port, residential, river, storage tanks, sea, bridge, lake, park, roundabout, soccer field, stadium, train station, works, golf course, runway, sparse shrub, and tennis court. Notably, OSM data associated with the collected images cannot be directly employed as reference owing to the problems presented in Section 1. Such a labeling procedure is extremely time- and labor-consuming, and annotating one image costs around 20 s, which is ten times more than labeling a single-scene image. Several example multi-
scene images are shown in Fig. 4. Numbers of aerial images related to various scenes are reported in Fig. 5. Among existing datasets, BigEarthNet (Sumbul et al., 2019) is one of the most relevant datasets, which consists of Sentinel-2 images acquired over the European Union with spatial resolutions ranging from 10 m/pixel to 60 m/pixel. Spatial sizes of images vary from $20 \times 20$ pixels to $120 \times 120$ pixels, and each is assigned multiple land-cover labels provided from the CORINE Land Cover map. Compared to BigEarthNet, our dataset is characterized by its high-resolution large-scale aerial images and worldwide coverage.

4.1.2. UCM dataset

UCM dataset (Yang and Newsam, 2010) is a commonly used single-scene aerial image dataset produced by Yang and Newsam from the University of California Merced. This dataset comprises 2100 aerial images cropped from aerial ortho imagery provided by the United States Geological Survey (USGS) National Map, and the spatial resolution of the collected images is one foot. The size of each image is $256 \times 256$ pixels, and all image samples are classified into 21 scene-level classes: overpass, forest, beach, baseball diamond, building, airplane, freeway, intersection, harbor, golf course, runway, agricultural, storage tank, mobile home park, medium residential, sparse residential, chaparral, river, tennis courts, dense residential, and parking lot. The number of aerial images collected for each scene is 100, and several example images are shown in Fig. 6. To learn scene prototypes from these single-scene images, we randomly choose 80% of image samples per scene category to train and validate the embedding function and utilize the rest for testing.

4.1.3. AID dataset

AID dataset (Xia et al., 2017) is another popular single-scene aerial image dataset which consists of 10000 aerial images with a size of $600 \times 600$ pixels. These images are captured from Google Earth imagery that is taken over China, the United States, England, France, Italy, Japan, and Germany, and spatial resolutions of the collected images vary from 0.5 m/pixel to 8 m/pixel. In total, there are 30 scene categories, including viaduct, river, baseball field, center, farmland, railway station, meadow,
bare land, storage tanks, beach, mountain, park, bridge, playground, church, commercial, desert, forest, parking, industrial, square, sparse residential, pond, medium residential, port, resort, airport, school, stadium, and dense residential. The number of images in different classes ranges from 220 to 420. Similar to the data split in the UCM dataset, 20% of images are chosen from each scene as test samples, while the remaining images are utilized to train and validate the embedding function. Some example images of the AID dataset are exhibited in Fig. 7.

### 4.1.4. Dataset configuration

In order to widely evaluate the performance of our method, we utilize two variant dataset configurations, UCM2MAI and AID2MAI, based on common scene categories shared by UCM/AID and MAI. Specifically, the UCM2MAI configuration consists of 1600 single-scene aerial images from the UCM dataset and 1649 multi-scene images from our MAI dataset. 16 aerial scenes that are commonly included in both two datasets are considered in UCM2MAI, and numbers of their associated images are listed in Table 1. Besides, the AID2MAI configuration is composed of 7050 and 3239 aerial images from the AID and MAI datasets, respectively. 20 common scene categories are taken into consideration, and the number of images related to each scene is present in Table 1. Although such configurations might limit the number of recognizable scene classes, we believe this limitation can be addressed by collecting more single-scene images by crawling OSM data and producing large-scale multi-scene aerial image datasets. We select only 90 and 120 multi-scene aerial images from UCM2MAI and AID2MAI as training instances, respectively, and test networks on the remaining multi-scene images. For rare scenes (e.g., port and train station), we select all associated training images, while for common scenes, we randomly select several of their training samples. It is noteworthy that we yield the scene prototype of residential by taking an average of high-level representations of aerial images belonging to scene medium residential and dense residential. Besides, although the UCM and AID datasets do not contain images for sea, their images for beach often comprise both sea and beach (cf. (c) in Fig. 7). Therefore, we make use of training samples labeled as beach to yield the prototype representation of sea.

### 4.2. Training details

The training procedure consists of two phases: 1) learning the embedding function $f_\theta$ on large quantities of single-scene aerial images and 2) training the entire PM-Net on a limited number of multi-scene images in an end-to-end manner. Thus, various training strategies are applied to each phase and detailed as follows.

In the first training phase, the embedding function $f_\theta$ is initialized with the corresponding deep CNNs pretrained on ImageNet (Deng et al., 2009), and weights in $g_\phi$ are initialized by a Glorot uniform initializer. Eq. (1) is employed as the loss of the network, and Nestrov Adam (Dozat, 2015) is chosen as the optimizer, of which parameters are set as

![Fig. 8. The influence of the number of heads on both dataset configurations.](image)

**Table 1**

| Scene Category | UCM2MAI | AID2MAI |
|---------------|---------|---------|
| apron         | 100     | 194     | 360   | 54   |
| baseball field| 100     | 75      | 220   | 235  |
| beach         | 100     | 94      | 400   | 130  |
| commercial    | 100     | 607     | 350   | 1391 |
| farmland      | 100     | 680     | 370   | 983  |
| woodland      | 100     | 762     | 250   | 1312 |
| parking lot   | 100     | 708     | 390   | 1777 |
| port          | 100     | 3       | 380   | 9    |
| residential   | 200     | 958     | 700   | 2082 |
| river         | 100     | 209     | 410   | 686  |
| storage tanks | 100     | 89      | 360   | 193  |
| sea           | 100*    | 51      | 400*  | 59   |
| golf course   | 100     | 75      | -     | -    |
| runway        | 100     | 230     | -     | -    |
| sparse shrub  | 100     | 336     | -     | -    |
| tennis court  | 100     | 114     | -     | -    |
| bridge        | -       | -       | 360   | 878  |
| lake          | -       | -       | 420   | 756  |
| park          | -       | -       | 350   | 638  |
| roundabout    | -       | -       | 420   | 281  |
| soccer field  | -       | -       | 370   | 302  |
| stadium       | -       | -       | 290   | 136  |
| train station | -       | -       | 260   | 9    |
| works         | -       | -       | 390   | 186  |
| All           | 1600    | 1649    | 7050  | 3239 |

* indicates that the number of images is not counted in total amounts, as the scene prototype of beach and sea are learned from the same images.

Table 2 Differences between two training phases.

| Phase          | Learnable Module | Dataset    | Memory      |
|----------------|------------------|------------|-------------|
| 1 prototype learning | ImageNet      | UCM/AID   | updated     |
| 2 memory retrieval  | MAI             | frozen     |             |

from the UCM dataset and 1649 multi-scene images from our MAI dataset. 16 aerial scenes that are commonly included in both two datasets are considered in UCM2MAI, and numbers of their associated images are listed in Table 1. Besides, the AID2MAI configuration is composed of 7050 and 3239 aerial images from the AID and MAI datasets, respectively. 20 common scene categories are taken into consideration, and the number of images related to each scene is present in Table 1. Although such configurations might limit the number of recognizable scene classes, we believe this limitation can be addressed by collecting more single-scene images by crawling OSM data and producing large-scale multi-scene aerial image datasets. We select only 90 and 120 multi-scene aerial images from UCM2MAI and AID2MAI as training instances, respectively, and test networks on the remaining multi-scene images. For rare scenes (e.g., port and train station), we select all associated training images, while for common scenes, we randomly select several of their training samples. It is noteworthy that we yield the scene prototype of residential by taking an average of high-level representations of aerial images belonging to scene medium residential and dense residential. Besides, although the UCM and AID datasets do not contain images for sea, their images for beach often comprise both sea and beach (cf. (c) in Fig. 7). Therefore, we make use of training samples labeled as beach to yield the prototype representation of sea.

4.2. Training details

The training procedure consists of two phases: 1) learning the embedding function $f_\theta$ on large quantities of single-scene aerial images and 2) training the entire PM-Net on a limited number of multi-scene images in an end-to-end manner. Thus, various training strategies are applied to each phase and detailed as follows.

In the first training phase, the embedding function $f_\theta$ is initialized with the corresponding deep CNNs pretrained on ImageNet (Deng et al., 2009), and weights in $g_\phi$ are initialized by a Glorot uniform initializer. Eq. (1) is employed as the loss of the network, and Nestrov Adam (Dozat, 2015) is chosen as the optimizer, of which parameters are set as
Table 3
Numerical Results on UCM2MAI (%).

| Model                        | m. F1 | m. F2 | m. p\textsubscript{e} | m. r\textsubscript{e} | m. p\textsubscript{r} | m. r\textsubscript{r} |
|------------------------------|-------|-------|-------------------------|------------------------|------------------------|------------------------|
| VGGNet\footnote{Simonyan and Zisserman, 2014} | 32.16 | 32.79 | 35.08                   | 34.35                  | 21.74                  | 22.57                  |
| VGGNet\footnote{Simonyan and Zisserman, 2014} | 51.42 | 49.04 | 62.00                   | 48.38                  | 36.80                  | 27.44                  |
| Mem-N2N-VGGNet (Sukhbaatar et al., 2015) | 52.16 | 50.93 | 57.26                   | 50.73                  | 20.79                  | 22.58                  |
| K-Branch CNN (Sumbul and Demir, 2019) proposed PM-VGGNet | 47.04 | 43.15 | 64.57                   | 41.83                  | 37.93                  | 22.28                  |
| ResNet\footnote{He et al., 2016} | 51.39 | 48.31 | 65.33                   | 47.37                  | 51.89                  | 30.48                  |
| ResNet (He et al., 2016) | 51.39 | 48.31 | 65.33                   | 47.37                  | 51.89                  | 30.48                  |
| Mem-N2N-Inception-V3 (Szegedy et al., 2015) | 58.56 | 58.06 | 64.17                   | 58.73                  | 46.44                  | 26.47                  |
| Mem-N2N-Inception-V3 (Szegedy et al., 2015) proposed PM-Inception-V3 | 58.56 | 58.06 | 64.17                   | 58.73                  | 46.44                  | 26.47                  |
| ResNet\footnote{He et al., 2016} | 54.31 | 51.45 | 63.97                   | 50.31                  | 44.33                  | 24.58                  |
| ResNet (He et al., 2016) | 54.31 | 51.45 | 63.97                   | 50.31                  | 44.33                  | 24.58                  |
| Mem-N2N-ResNet (Sukhbaatar et al., 2015) | 54.31 | 51.45 | 63.97                   | 50.31                  | 44.33                  | 24.58                  |
| Mem-N2N-ResNet (Sukhbaatar et al., 2015) proposed PM-ResNet | 56.89 | 54.11 | 69.85                   | 53.38                  | 55.93                  | 29.76                  |
| NASNet\footnote{Zoph and Le, 2017} | 43.64 | 39.34 | 58.96                   | 38.39                  | 46.01                  | 19.69                  |
| NASNet (Zoph and Le, 2017) | 43.64 | 39.34 | 58.96                   | 38.39                  | 46.01                  | 19.69                  |
| Mem-N2N-NASNet (Sukhbaatar et al., 2015) | 55.17 | 53.05 | 64.71                   | 52.65                  | 49.60                  | 29.14                  |
| Mem-N2N-NASNet (Sukhbaatar et al., 2015) proposed PM-NASNet | 60.13 | 59.57 | 67.04                   | 60.42                  | 58.60                  | 35.04                  |

CNN\textsuperscript{*} is initialized with weights pretrained on ImageNet. CNN, Mem-N2N, and PM-Net are initialized with parameters pretrained on the UCM dataset. m. F1 and m. F2 indicate the mean F1 and F2 score. m. p\textsubscript{e} and m. r\textsubscript{e} indicate mean example-based precision and recall. m. p\textsubscript{r} and m. r\textsubscript{r} indicate mean label-based precision and recall.

Recommended: \( \beta_1 = 0.9, \beta_2 = 0.999, \) and \( \epsilon = 1 \times 10^{-8}. \) The learning rate is set as \( 2 \times 10^{-4} \) and decayed by \( \sqrt{0.1} \) when the validation loss fails to decrease for two epochs.

In the second learning phase, we initialize \( f_0 \) with parameters learned in the previous training stage and employ the Glorot uniform initializer to initialize all weights in \( Q_0, V_0, K_0, \) and the last fully-connected layer. \( L \) and \( U \) are set to the same value of 256, and the number of heads is defined as 20. Notably, all weights are trainable, and the embedding function is tuned during the second training phase as well. Differences between two training phases are summarized in Table 2. Multiple scene-level labels are encoded as multi-hot vectors, where 0 indicates the absence of the corresponding scene while 1 refers to existing scenes. Accordingly, the loss is defined as binary cross-entropy. The optimizer is the same as that in the first training phase, but here we make use of a relatively large learning rate, \( 5 \times 10^{-4}. \) The network is implemented on TensorFlow and trained on one NVIDIA Tesla P100 16 GB GPU for 100 epochs. We set the size of training batch to 32 for both training phases.

4.3. Evaluation metrics

For the purpose of evaluating the performance of networks quantitatively, we utilize example-based \( F_1 \) (Wu and Zhou, 2016) and \( F_2 \) (Van Rijsbergen, 1979) scores as evaluation metrics and calculate them with the following equation:

\[
F_\beta = (1 + \beta^2) \frac{p_\beta r_\beta}{\beta p_\beta + r_\beta}, \quad \beta = 1, 2, \tag{7}
\]

where \( p_\beta \) and \( r_\beta \) denote example-based precision and recall (Tsoumakas and Vlahavas, 2007). We calculate \( p_\beta \) and \( r_\beta \) as follows:

\[
p_\beta = \frac{TP_\beta}{TP_\beta + FP_\beta}, \quad r_\beta = \frac{TP_\beta}{TP_\beta + FN_\beta}, \tag{8}
\]

where \( FN_\beta, FP_\beta, \) and \( TP_\beta \) represent numbers of false negatives, false positives, and true positives in an example, respectively. In our case, an example is a multi-scene aerial image, and by averaging scores of all examples in the test set, the mean example-based \( F_1 \) scores, precision, and recall can be eventually computed. In addition to example-based evaluation metrics, we also calculate label-based precision \( p_r \) and recall \( r_r \) with Eq. 8 but replace \( FN_\beta, FP_\beta, \) and \( TP_\beta \) with numbers of false negatives, false positives, and true positives in respect of each scene category. The mean \( p_r \) and \( r_r \) can then be calculated. Note that principle indexes are the mean \( F_1 \) and \( F_2 \) scores.

4.4. Results on UCM2MAI

For a comprehensive evaluation, we compare the proposed PM-Net with two baselines, CNN\textsuperscript{*} and CNN. The former is initialized with parameters pretrained on ImageNet, and the latter is pretrained on single-scene datasets. Besides, we compare our network with a memory network, Mem-N2N (Sukhbaatar et al., 2015). Since Mem-N2N was proposed for the question answering task, we adapt it to our task by replacing its inputs, i.e., embeddings of questions and statements, with query image representations \( f_\phi(X) \) and scene prototypes \( p_\psi, \) respectively. To
be more specific, we feed X to a CNN backbone and take its output as the input of Mem-N2N. Scene prototypes are stored in the memory of Mem-N2N and retrieved according to $f_{\phi}(X)$. The initialization of $f_{\phi}$ is the same as that of our network, and the entire Mem-N2N is trained in an end-to-end manner. Various backbones of embedding functions are tested, and quantitative results are reported in Table 3. Besides, we also compare with a multi-attention driven multi-label classification network, termed as K-Branch CNN (Sumbul and Demir, 2019). K-Branch samples images into K spatial resolutions and extracts their features with separate branches. Afterwards, a bidirectional recurrent neural network is employed to encode their relationships for inferring multiple labels. In our experiments, K is set as default, 3, and input sizes of the three branches are 224 × 224, 112 × 112, and 56 × 56, respectively. Here we analyze results from the following three perspectives.

4.4.1. The effectiveness of learnt single-scene prototypes

To demonstrate the effectiveness of the prototype-inhabiting external memory, here we focus on comparisons between PM-Net and standard CNNs. In Table 3, PM-VGGNet increases the mean $F_1$ and $F_2$ scores by 3.00% and 2.12%, respectively, with respect to VGGNet, and PM-ResNet obtains increments of 5.50% and 5.80% in the mean $F_1$ and $F_2$ scores compared to ResNet. Besides, it is interesting to observe that PM-NASNet achieves not only the best mean $F_1$ and $F_2$ scores (60.13% and 59.57%) but also relatively high example-based precision and recall in comparison with other competitors. This demonstrates that employing NASNet as the embedding function can enhance the robustness of PM-Net. Comparisons between PM-Inception-V3 with Inception-V3 show that the external memory module contributes to improvements of 4.60% and 6.78% in the mean $F_1$ and $F_2$ scores, respectively. To summarize, memorizing and leveraging scene prototypes learned from huge quantities of single-scene images can improve the performance of network in multi-label scene recognition when limited training samples are available. For a deep insight, we further conduct ablation studies on the prototype modality and embedding function.

Single- vs. multi-prototype representations. We note that images collected over variant countries show high intra-class variability, and therefore, we wonder whether learning multi-prototype scene representations could improve the effectiveness of PM-Net. Specifically, instead of yielding scene prototypes via Eq. 2, we partition representations of single-scene aerial images belonging to the same scene into several clusters and take cluster centers as multi-prototype representations of each scene. In our experiments, we test two clustering methods, K-Means (Lloyd, 1982) and Agglomerative (Zepeda-Mendoza and Resendis-Antonio, 2013), with PM-ResNet on both UCM2MAI and AID2MAI, and results are shown in Fig. 9. We can see that the performance of PM-ResNet is decreased with the increasing number of cluster centers either using K-Means or Agglomerative clustering algorithms. Explanations could be that there are no obvious subclusters within each scene category (cf. Fig. 13), and thus PM-Net does not benefit from fine-grained multi-prototype representations.

Frozen vs. trainable embedding function. The embedding function plays a key role in both scene prototype learning and memory retrieval. In the former, we train the embedding function on single-scene images, while in the latter, the function is fine-tuned on multi-scene images. To explore the effectiveness of fine-tuning, we conduct experiments on freezing the embedding function when learning the memory retrieval module. The comparisons between PM-Net learned with frozen and trainable embedding functions are shown in Fig. 10. It can be observed that PM-Net with a trainable embedding function shows higher performance on both UCM2MAI and AID2MAI configurations. The reason could be that sources of single- and multi-scene images are variant, and fine-tuning can narrow their gaps.

Triplet vs. cross-entropy loss. Triplet loss (Schroff et al., 2015) is known as learning discriminative representations by minimizing distances between embeddings of the same class while pushing away those of different classes. To study its performance in our task, we train the embedding function by replacing Eq. 1 with the following equation:

$$\mathcal{L}(X'_i) = \max(||f_{\phi}(X'_i) - f_{\phi}(X'_{pos})||^2 - ||f_{\phi}(X'_i) - f_{\phi}(X'_{neg})||^2 + \alpha, 0),$$

where $X'_{pos}$ and $X'_{neg}$ denote positive and negative samples, i.e., images belonging to common and different classes, respectively, and $\alpha$ is set as default, 0.5. The trained embedding function is then utilized to extract scene prototypes and initialize $f_{\phi}$ in the phase of learning the memory retrieval module. Besides, all the other setups are remained the same. We compare the performance of PM-Net using embedding functions trained by the triplet loss, and $\text{brown}$ bars denote the performance of PM-Net with the cross-entropy loss as $\mathcal{L}$.
module VGGNet increases the mean F1 scores by 2.26% and 0.25%, respectively, compared to Mem-N2N-VGGNet. While taking ResNet as the backbone, the improvement can reach 2.58% in the mean F1 score. Besides, the highest increments of mean F1 and F2 scores, 4.96% and 6.52%, are achieved by PM-NASNet. These observations demonstrate that our memory retrieval module plays a key role in inferring multiple aerial scenes. An explanation could be that compared to the memory reader in Mem-N2N, our module comprise multiple scene prototypes, single-scene training samples can also benefit multi-label aerial image datasets. To summarize, except for learning scene prototypes, single-scene training samples can also benefit multi-label aerial image datasets. To summarize, except for learning scene prototypes, single-scene training samples can also benefit multi-label aerial image datasets.

4.4.2. The effectiveness of our multi-head attention-based memory retrieval module

As a key component of the proposed PM-Net, the multi-head attention-based memory retrieval module is designed to retrieve scene prototypes from the external memory, and we evaluate its effectiveness by comparing PM-Net with Mem-N2N. As shown in Table 3, PM-Net outperforms Mem-N2N with variant embedding functions. Specifically, PM-VGGNet increases the mean F1 and F2 scores by 2.26% and 0.25%, respectively, compared to Mem-N2N-VGGNet. While taking ResNet as the embedding function, the improvement can reach 2.58% in the mean F1 score. Besides, the highest increments of mean F1 and F2 scores, 4.96% and 6.52%, are achieved by PM-NASNet. These observations demonstrate that our memory retrieval module plays a key role in inferring multiple aerial scenes. An explanation could be that compared to the memory reader in Mem-N2N, our module comprise multiple heads, and each of them focuses on encoding a specific relevance between the query image and scene prototypes. In this case, more comprehensive scene-related memories can be used for inferring multiple scene labels. Moreover, we also conduct experiments on directly utilizing relevances for inferring multiple scene labels. Specifically, we set the number of heads to 1 and replace the softmax activation in Eq. 4 with the sigmoid function. Relevances between the query image and scene prototypes can then be interpreted as the existence of each scene. We compare it with our memory retrieval module on variant backbones, and results are shown in Fig. 12. We can see that utilizing relevances R(\(X, M\)) as weights for aggregating scene prototypes leads to higher network performance.

4.4.3. The benefit of exploiting single-scene training samples

Let’s start with the conclusion: exploiting single-scene images significantly contributes to our task. To analyze its benefit, we mainly compare CNNs* and CNNs. It can be observed that even with identical network architectures, the performance of CNN is superior to that of CNN*. More specifically, VGGNet achieves the highest improvement of the mean F1 scores, 19.26%, in comparison with VGGNet*. NASNet shows higher performance in all metrics compared to ResNet*, while other CNNs perform poorly in only the mean example-based precision with respect to their corresponding CNNs*. Besides, we visualize features of single-scene images learned by VGGNet on UCM and AID datasets via t-SNE, respectively. As shown in Fig. 13, extracted features are discriminative and separable in the embedding space, which demonstrates the effectiveness of learning the embedding function on single-scene aerial image datasets. To summarize, except for learning scene prototypes, single-scene training samples can also benefit multi-label scene interpretation by pretraining CNNs which are further utilized to initialize the embedding function.

We exhibit several example predictions of PM-ResNet trained on UCM2MAI in Table 4. False positives are marked as red, while false negatives are in blue. As shown in the forth example at the top row, we
Table 4
Example images and predictions on UCM2MAI.

| Ground Truths | Predictions |
|---------------|-------------|
| farmland, woodland, residential | farmland, woodland, residential |
| commercial, parking lot, residential | commercial, parking lot, residential |

Table 5
Numerical results on AID2MAI (%).

| Model | m, F₁ | m, F₂ | m, pₐ | m, rₑ | m, p₁ | m, r₁ |
|-------|-------|-------|-------|-------|-------|-------|
| VGGNet* (Simonyan and Zisserman, 2014) | 41.57 | 36.36 | 64.02 | 34.04 | 25.98 | 12.80 |
| VGGNet (Simonyan and Zisserman, 2014) | 48.30 | 50.80 | 48.53 | 54.19 | 32.89 | 44.75 |
| Mem-N2N-VGGNet (Sukhbaatar et al., 2015) | 45.92 | 43.17 | 56.16 | 42.22 | 23.10 | 18.76 |
| K-Branch CNN (Sembol and Demir, 2019) | 47.67 | 43.88 | 63.84 | 42.37 | 26.53 | 16.15 |
| proposed PM-VGGNet | 54.37 | 51.44 | 65.69 | 50.39 | 48.06 | 22.40 |
| Inception-V3* (Szegedy et al., 2015) | 45.92 | 40.76 | 66.17 | 38.43 | 39.56 | 14.71 |
| Inception-V3 (Szegedy et al., 2015) | 51.81 | 49.44 | 62.91 | 48.93 | 45.26 | 36.32 |
| Mem-N2N-Inception-V3 (Sukhbaatar et al., 2015) | 52.13 | 53.83 | 52.33 | 56.21 | 33.33 | 29.05 |
| proposed PM-Inception-V3 | 53.08 | 49.26 | 69.42 | 47.85 | 48.20 | 24.65 |
| ResNet* (He et al., 2016) | 50.06 | 46.88 | 64.32 | 45.98 | 39.48 | 22.34 |
| ResNet (He et al., 2016) | 54.74 | 52.76 | 65.54 | 52.62 | 47.54 | 40.23 |
| Mem-N2N-ResNet (Sukhbaatar et al., 2015) | 53.26 | 60.41 | 46.15 | 68.07 | 23.75 | 30.21 |
| proposed PM-ResNet | 57.42 | 54.34 | 70.62 | 53.33 | 55.34 | 29.55 |
| NASNet* (Zoph and Le, 2017) | 47.53 | 42.93 | 65.57 | 40.94 | 34.79 | 16.42 |
| NASNet (Zoph and Le, 2017) | 53.08 | 50.68 | 64.33 | 50.17 | 46.68 | 37.43 |
| Mem-N2N-NASNet (Sukhbaatar et al., 2015) | 39.27 | 40.72 | 38.52 | 42.38 | 20.03 | 20.41 |
| proposed PM-NASNet | 54.11 | 52.39 | 64.03 | 52.30 | 43.16 | 33.99 |

Blue predictions are false negatives, while red predictions indicate false positives.

CNN, Mem-N2N, and PM-Net are initialized with parameters pretrained on the AID dataset.
see that PM-Net can accurately perceive aerial scenes even in complex contexts, but unseen scene appearance (i.e. apron and runway in snow) can influence its prediction.

4.5. Results on AID2MAI

Table 5 reports numerical results on the AID2MAI configuration. It can be seen that the performance of PM-Net is superior to all competitors in the mean $F_1$ score. Compared to Mem-N2N-VGGNet, the proposed PM-VGGNet increases the mean $F_1$ and $F_2$ scores by 6.70% and 7.56%, respectively, while improvements reach 6.07% and 0.64% in comparison with VGGNet. PM-ResNet achieves the best mean $F_1$ score and example-based precision, 57.42% and 70.62, respectively. With NASNet as the backbone, exploiting the proposed memory retrieval module contributes to increments of 1.03% and 1.71% in mean $F_1$ and $F_2$ scores compared to directly learning NASNet on a small number of multi-scene samples.

We present some example predictions of PM-ResNet in Table 6. As shown in the top row, PM-ResNet learned with a limited number of annotated multi-scene images can accurately identify various aerial scenes even image contextual information is complicated. The bottom row shows some inaccurate predictions. It can be observed that although bridge and parking lot account for relatively small areas in last two examples at the top row, the proposed PM-Net can successfully detect them. Similar observations can also be found in the first and third example at the bottom row that residential and parking lot are recognized by our network, even they are located at the corner. In conclusion, quantitative results illustrate the effectiveness of our network in learning to perform unconstrained multi-scene classification, and example predictions further demonstrate it.

5. Conclusion

In this paper, we propose a novel multi-scene recognition network, namely PM-Net, to tackle both the problem of aerial scene classification in the wild and scarce training samples. To be more specific, our network
consists of three key elements: 1) a prototype learning module for encoding prototype representations of variant aerial scenes, 2) a prototype-inhabiting external memory for storing high-level scene prototypes, and 3) a multi-head attention-based memory retrieval module for retrieving associated scene prototypes from the external memory for recognizing multiple scenes in a query aerial image. For the purpose of facilitating the progress as well as evaluating our method, we propose a new dataset, MAI dataset, and experiment with two dataset configurations, UCM2MAI and AID2MAI, based on two single-scene aerial image datasets, UCM and AID. In scene prototype learning, we train the embedding function on most of single-scene images as we aim to simulate the real-life scenario, where massive single-scene samples can be collected at low cost by resorting to OSM data. To learn memory retrieval, our network is fine-tuned on only around 100 training samples from the MAI dataset. Experimental results on both UCM2MAI and AID2MAI illustrate that learning and memorizing scene prototypes with our PM-Net can significantly improve the classification accuracy. The best performance is achieved by employing ResNet as the embedding function, and the best mean P1 score reaches nearly 0.6. We hope that our work can open a new door for further researches in a more complicated and challenging task, multi-scene interpolation in single images. Looking into the future, we intend to apply the proposed network to the recovery of weakly supervised scenes.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is jointly supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. ERC-2016-StG-714087), Acronym: So2Sat), by the Helmholtz Association through the Framework of Helmholtz AI [Grant No.: ZT-1P-5-01] - Local Unit “Munich Unit @Aeronautics, Space and Transport (MASTy)” and Helmholtz Excellent Professorship “Data Science in Earth Observation - Big Data Fusion for Urban Research” and by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI lab “A14E0 – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond” (Grant No.: 01DD2001)

References

Audebert, N., Saux, B.L., Lefèvre, S., 2018. Beyond RGB: Very high resolution urban remote sensing with multispectral deep networks. ISPRS J. Photogramm. Remote Sens. 140, 20–32.
Bi, Q., Qin, K., Li, Z., Zhang, H., Xu, K., Xia, G., 2020. A multiple-instance densely-connected ConvNet for aerial scene classification. IEEE Trans. Image Process. 29, 4911–4926.
Byju, A., Sambul, G., Demir, B., Bruzzone, L., 2000. Remote sensing image scene classification with deep neural networks in JPEG 2000 compressed domain, arXiv: 2006.11529.
Cheng, G., Han, J., Lu, X., 2017. Remote sensing image scene classification: Benchmark and state of the art. Proc. IEEE 105 (10), 1865–1883.
Cheng, G., Xie, X., Han, J., Guo, L., Xia, G., 2020. Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 3725–3756.
Corina, M., Stefăni, M., Baraldi, L., Ciučiu, R., 2020. Meshed-memory transformer for image captioning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F., 2009. ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Devlin, J., Chang, M., Lee, K., Toutanova, K., 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. North American Chapter of the Association for Computational Linguistics.
Dozat, T., 2015. Incorporating Nesterov momentum into Adam, http://cs229.stanford.edu/notes/2015/05/02/report.pdf, online.
Guerrero, S., Caputo, B., M.T., 2018. DeepNCM: Deep nearest class mean classifiers.
