CCML: A Novel Collaborative Learning Model for Classification of Remote Sensing Images with Noisy Multi-Labels

Ahmet Kerem Aksoy, *Student Member, IEEE,* Mahdyar Ravanbakhsh, *Member, IEEE,* Tristan Kreuziger, *Student Member, IEEE,* and Begüm Demir, *Senior Member, IEEE*

***Abstract*—The development of accurate methods for multi-label classification (MLC) of remote sensing (RS) images is one of the most important research topics in RS. Deep Convolutional Neural Networks (CNNs) based methods have triggered substantial performance gains in RS MLC problems, requiring a large number of reliable training images annotated by multiple land-cover class labels. Collecting such data is time-consuming and costly. To address this problem, the publicly available thematic products, which can include noisy labels, can be used for annotating RS images with zero-labeling cost. However, multi-label noise (which can be associated with wrong as well as missing label annotations) can distort the learning process of the MLC algorithm, resulting in inaccurate predictions. The detection and correction of label noise are challenging tasks, especially in a multi-label scenario, where each image can be associated with more than one label. To address this problem, we propose a novel Consensual Collaborative Multi-Label Learning (CCML) method to alleviate the adverse effects of multi-label noise during the training phase of the CNN model. CCML identifies, ranks, and corrects noisy multi-labels in RS images based on four main modules: 1) group lasso module; 2) discrepancy module; 3) flipping module; and 4) swap module. The task of the group lasso module is to detect the potentially noisy labels assigned to the multi-labeled training images, and the discrepancy module ensures that the two collaborative networks learn diverse features, while obtaining the same predictions. The flipping module is designed to correct the identified noisy multi-labels, while the swap module task is devoted to exchanging the ranking information between two networks. The proposed CCML method detects noisy multi-labels in the training set and corrects them through a relabeling mechanism. Unlike existing methods that make assumptions about the noise distribution, our proposed CCML does not make prior assumptions about the noise distribution in the training set. The experiments conducted on two multi-label RS image archives confirm the robustness of the proposed CCML under extreme multi-label noise rates. Our code is publicly available online: http://www.noisy-labels-in-rs.org**

***Index Terms*—Multi-label noise, collaborative learning, multi-label image classification, deep learning, remote sensing**

I. Introduction

REMOTE sensing (RS) images acquired by satellite-borne and airborne sensors are a rich source of information for monitoring the Earth surface, e.g., urban area studies, forestry applications, and crop monitoring [1]. As a result of recent advances in RS technology, huge amounts of RS images have been acquired and stored in massive archives, from which the mining of useful information is an important and challenging issue. In view of that, development of multi-label RS image scene classification methods that aim to automatically assign multiple land-cover class labels (i.e., multi-labels) to each RS image scene in an archive is a growing research interest in RS. This can be achieved by direct supervised classification of each image in the archive. In recent years, deep learning (DL) based approaches have attracted great attention also in multi-label classification (MLC) of RS images due to their high capability to describe the complex spatial and spectral content of RS images. As an example, in [2] convolutional neural networks (CNN) that contain the softmax function as the activation of the last CNN layer are presented. In [3], a radial basis function neural network with a multi-label classification layer is proposed. In [4], an attention-based long short-term memory (LSTM) network is used to sequentially predict classes one after another. An encoder-decoder neural network that includes: i) a squeeze excitation layer (which characterizes the channel-wise interdependencies of the image feature maps); and ii) an adaptive spatial attention mechanism (which models the informative image regions) is proposed in [5]. A multi-attention driven approach that contains: i) spatial resolution specific CNNs in a branch-wise architecture; and ii) a bidirectional LSTM network is presented in [6]. In [7], a study to analyze and compare different DL loss functions in the framework of MLC of RS images is presented.

Most of the above mentioned DL based approaches for the MLC of RS images require a sufficient number of high quality (i.e., reliable) training images annotated with multi-labels. This is crucial for accurate characterization of complex content of images with discriminate and descriptive features and thus for achieving accurate multi-label predictions. However, the collection of a sufficient number of reliable multi-labeled images is time-consuming, complex, and costly in operational scenarios, and can significantly affect the the final accuracy of the MLC methods [8]. To overcome this problem, a common approach is to employ DL models pre-trained on publicly available Computer Vision (CV) datasets (e.g., ImageNet [9]). Then, the pre-trained models are fine-tuned by considering a small set of multi-labeled RS images for the final classification task. This approach is not fully adequate for RS images due to the differences between the CV and RS image characteristics (e.g., Sentinel-2 multispectral images

A. K. Aksoy, M. Ravanbakhsh, T. Kreuziger, and B. Demir are with the Faculty of Electrical Engineering and Computer Science, Technische Universität Berlin, 10623 Berlin, Germany (emails : a.aksoy@tu-berlin.de, ravanbakhsh@tu-berlin.de, tristan.kreuziger@tu-berlin.de, demir@tu-berlin.de).

contain 13 spectral bands associated with varying and lower spatial resolutions compared to CV images). In addition, the semantic content (and thus the considered semantic classes) present in RS images is significantly different from that of CV images (see Fig. 1). Thus, DL models trained from scratch on a large RS training set annotated with multi-labels are required. An effective approach for constructing a large training set with zero-annotation effort is to exploit the publicly available thematic products (e.g., the Corine Land Cover [CLC] map, the GLC2000 and the GlobCover) in RS as labeling sources [10,11]. As an example, Sümbül et al. develop the BigEarthNet benchmark archive made up of Sentinel-2 multispectral images annotated by using the CLC map to drive the DL studies in RS MLC [8]. Constructing such large RS training sets with zero labeling cost is highly valuable. However, the set of land-use and land-cover class labels available in a given area through the thematic products can be incomplete or wrong (i.e., noisy). As an example, according to the validation report of the CLC, the accuracy is around 85% [12]. Using training images with noisy labels may result in uncertainty in the MLC model and thus may lead to a reduced performance on multi-label prediction. Accordingly, methods that allow reducing the negative impact of noisy annotations are needed in the framework of RS MLC. In detail, when training images are annotated by multi-labels, two types of noise can exist for a given RS image:

1) Noise on missing label: This type of noise appears when a land cover class label is not assigned to the image under consideration although that class is present in the given image (i.e., the label is missed in the label set of that image) [13].
2) Noise on wrong label: This type of noise appears when a land cover class label is assigned to the image under consideration although that class is not present in the given image (i.e., the label is wrong in the label set of that image) [14].

The number of missing or wrong class labels can vary depending on the labeling source. It is worth noting that the two types of noise can simultaneously appear associated to different spatial areas of a given RS image (e.g., a land-cover class label can be missed, while another land-cover class label is wrongly assigned in different spatial portions of the image). Since DL models can easily overfit to noisy labeled data [15,16], dealing with label-noise can significantly improve the MLC performance. Recently a couple of studies in RS are presented to learn from noisy labels in RS MLC. As an example, in [17], a semantic segmentation method that identifies label noise is presented to generate accurate land-cover maps by classifying RS images. This is achieved by simply evaluating the loss values since the noisy image labels are associated with the highest values of the loss. However, this method can only identify the wrong label noise and ignores the missing label noise problem. Hua et al. propose a regularization method to improve the MLC performance in RS under label noise [18]. The regularization is defined on the basis of a label correlation matrix constructed by the semantic word embedding of the labels considering that semantically similar classes are more likely to appear together. Construction of a reliable label correlation matrix for different RS applications is a complex task due to the difficulty in collecting text descriptions of class labels for properly modeling the correlation between all possible combinations of classes present in RS images. The performance of these two methods depends on the accurate estimation of noise distribution in the considered data. Thus, there is a need to make prior assumptions about the noise structure, which restricts the applicability and generalization capability of the methods for different MLC applications with different noise distributions. This is a critical limitation for complex RS MLC problems. Unlike RS, in the CV and multimedia communities, the development of noise-robust DL models is much more extended and widely studied (see Sec. II-A for the literature survey). However, most of the existing methods assume that each image is annotated by a single label associated with the most significant content of the considered image [19,20]. Adapting single label noise tolerance methods for multi-labeled images is a challenging task due to the complexity of modeling the above-mentioned two types of noise in multi-labeled images. This becomes more critical when the number of land-cover classes (and thus class combinations) increases.

To address this problem, we propose a novel Consensual Collaborative Multi-label Learning (CCML) method. Unlike the existing methods in the literature, the proposed method identifies, ranks, and corrects noisy multi-labels in RS images without making any prior assumption about the type of noise in the training set. To this end, the proposed CCML method contains four different modules: 1) group lasso module; 2) discrepancy module; 3) flipping module; and 4) swap module. To automatically identify the noise type, we propose a group lasso module that computes a sample-wise ranking loss and penalizes noisy labeled images. The discrepancy module ensures that the two collaborative networks learn different features, while obtaining consistent predictions. The flipping module flips the identified noisy labels, while the swap module exchanges the ranking information between the two collaborative networks. The main contributions of this work are summarized as follows:

- To the best of our knowledge, CCML is the first method that automatically identifies the two types of noise in multi-labeled RS images without any prior assumption about the noise distribution.
- CCML can learn from highly noisy training sets by re-weighting or correcting the noisy labels based on the noise type rather than excluding them from the training process.
- We adapt the collaborative paradigm to be suitable for multi-labeled images based on ranking training images according to their noise levels and exchanging the ranking information.
- The proposed CCML is evaluated on the two multi-labeled benchmark RS image datasets: i) the BigEarthNet dataset that consist of Sentinel-2 multispectral images [8]; and ii) the UC Merced Land Use dataset that contain images selected from aerial orthoimagery [21].

(a)

Discontinuous urban fabric, Coniferous forest, Mixed forest, Industrial or commercial units

Discontinuous urban fabric, Industrial or commercial units, Non-irrigated arable land

(b)

Person, Bicycle, Bus, Umbrella, Clock, Backpack

Person, Bicycle, Car, Truck, Dog

Fig. 1: An example of multi-labeled images (a) in RS [8]; and (b) in CV [22].

The rest of the paper is organized as follows. In Sec. II we survey the DL based methods presented in the CV and multimedia communities for learning from noisy single-labeled and multi-labeled training images. In Sec. III, we introduce the proposed CCML method in detail. Sec. IV describes the considered datasets and the experimental setup, while Sec. V represents the experimental results. Finally, Sec. VI concludes the paper by a short discussion on the results, observations by indicating the future works.

II. Related Work

In this section we review the methods that are robust to label noise and are presented in the CV literature in the context of image scene classification. We categorize the considered methods in two sub-sections as: 1) methods robust to noise on single-labeled images; and 2) methods robust to noise on multi-labeled images.

A. Methods Robust to Noise on Single-Labeled Images

When an image is annotated by a single label, there is only one type of noise associated to the wrong class label assignment. The methods in this category aim to model the noise distribution present in the data considering only this type of noise to reduce the influence of noisy labels on the result.

Patrini et al. propose a loss correction method by estimating the noise transition matrix of data [23]. The proposed method claims to fix the class-dependent label noise in the data, given the probability of each class being corrupted into another. However, as the number of classes increases, it becomes harder to estimate the noise transition matrix, making the method difficult to scale. In [24], the correct labels are considered latent variables in the presence of noisy labels, and the expectation-maximization [25] algorithm is applied to iteratively calculate the correct labels as well as the network parameters. This scheme is also extended to scenarios, where noisy labels are dependent on the features, by modeling noise via a softmax layer that connects correct labels to noisy labels. In [26], it is shown that overfitting avoidance techniques such as regularization and dropout can be partially effective when dealing with label noise. However, in these cases, label noise may still affect the quality of the classifier, resulting in an accuracy drop. Moreover, since noise injection is already an overfitting prevention technique that attempts to improve generalization performance [27], regularization should be carefully applied to avoid underfitting.

Sukhbaatar et al. explore the performance of CNNs trained on noisy labels. CNNs modified by including an additional fully connected layer at the end of their network architecture to adapt the predictions to the noisy label distribution of the data [28]. In [29], data is partitioned into multiple subsets and used to train different classifiers. If all of the classifiers agree on a label from the original training set, the label is updated with the agreed upon prediction. This process is repeated over several stages, which gradually improve the overall performance. Nonetheless, this process depends very much on the quality of the classifiers, and it can take much time when large and complex training sets are used. Dehghani et al. use a student model that is trained on noisy data along with a teacher model that exploits the noise structure that is extracted by the student model [30]. However, this approach needs training samples with both clean and noisy labels, which is not always available. On the other hand, [31] proposes a self-error-correcting CNN that can work on completely noisy data. The network swaps potential noisy labels with the most probable prediction of the network, while simultaneously optimizing the model parameters. Failing to distinguish hard samples from noisy ones is a common problem when fixing labels, which may result in an undesired situation where the model swaps labels by mistake. To address this issue, additional policies are often needed. Wu et al. use semantic bootstrapping to detect noisy samples and remove them from training [32]. Removing as few samples as possible is an important aspect of the noisy sample removal process to avoid unnecessary information loss. On the other hand, [33] trains a DNN on a noisy dataset and filters the noisy labels out while keeping the samples for the training in an unsupervised fashion. This prevents loosing training samples, and thus allows DNNs to learn from more examples. MentorNet [34] imposes a data-driven sample weighting curriculum learning on the student network to choose probably clean samples from the data. Han et al. propose a learning paradigm called Co-Teaching, which we predicate our collaborative model on [35]. Under Co-Teaching, two networks are trained simultaneously and choose batches including only clean training samples to feed each other. Each network back-propagates the mini-batch that is chosen by its peer network to update itself. Han et al. demonstrate that Co-Teaching is an effective method when

dealing with label noise.

In [36], the authors propose a Co-Teaching method with an additional discrepancy measurement to make the used networks diverge. Two peer networks that learn different features of the same probability distribution are used. then, both networks select clean samples for each other based on their predictions to improve performances mutually. Ren et al. propose a meta-learning algorithm, in which a weighting factor is assigned to every training sample based on its gradient direction [37]. Although this method achieves impressive results, it needs a clean validation set, which may not always be present. One of the advantages of the Co-Teaching label noise structure modeling methods is that they can be decoupled from the training process and the application domain. Such decoupled label noise estimation process allows Co-Teaching approaches to be used and tested with different network architectures. However, the success of the above-mentioned works mostly depends on the accurate estimation of noise distribution in data, which restricts the applicability and generalization capability of the methods.

B. Methods Robust to Noise on Multi-Labeled Images

Most of the above-mentioned works are designed to address only one type of noise (i.e., wrong class label assignment) and thus it is not easy to simply extend or apply these methods in the framework of the MLC of images without large modifications.There are also fw works presented in the literature to address the multi-label noise problems. As an example, Ghosh et al. study different loss functions such as categorical cross-entropy (CCE), mean square error (MSE), and mean absolute error (MAE) for noisy multi-label classification, and argue that MAE is more robust to label noise compared to the others [38]. In [39], it is stated that MAE shows poor performance with more complex training sets, and a set of robust loss functions that combine CCE and MAE is proposed. This is known as noise-robust alternative of CCE in the literature [38]. Meta-Learning and ensemble methods are other approaches proposed to address the label noise in MLC problems. Li et al. aim to find noise-tolerant model parameters using a teacher model along with a student model to make accurate predictions by optimizing a meta-objective, which encourages the student model to give consistent results with the teacher model after introducing synthetic labels [40].

Bucak et al. present a ranking-based multi-label learning method that exploits the group lasso to enhance the accuracy with incomplete class assignments [41]. The method first computes two error values, one associated with predicted classes in the multi-label set and one error for the unpredicted ones within the same multi-label set for each sample. The two errors are then combined to define the ranking error for each unpredicted class. This error value indicates the possible missing class for the related sample. Finally, all the ranking errors associated with that sample are summed to define a final ranking error. The group lasso is introduced by [42] as an extension to the regular lasso for grouping points of interest together for accurate prediction in regression. It is effective, because it groups variables together to be included or excluded completely, as opposed to lasso, which only selects variables individually. The group lasso is used to regularize the network in [41], giving empirically robust results against missing class labels. Durand et al. propose a modified binary cross-entropy loss, which reduces the negative effects of missing class labels on training [14]. Jain et al. address the problem of missing class labels by defining a novel loss function called propensity scored loss (PSL) [13]. PSL not only can prioritize the most relevant few labels over the irrelevant majority, but also provide unbiased estimation of true labels of a sample without omitting the missing labels completely.

As mentioned before, two types of noise (which are associated with wrong and missing class label assignments) may exist when training images annotated with multi-labels are considered. However, all the aforementioned methods are designed to overcome only one type of label noise in the training data (either missing or wrong class label assignments). Thus, these methods are not capable of identifying and correcting the two noise types simultaneously, which is an important limitation in MLC. In this paper to address this issue, we propose a novel collaborative learning method that aims at training a DL model robust to the two types of label noise based on a a group lasso module without any prior assumption in the framework of MLC of RS images.

III. PROPOSED CCML

The proposed CCML method consists of four main modules: 1) discrepancy module; 2) group lasso module; 3) flipping module; and 4) swap module. The discrepancy module is devoted to allow the two networks that are used collaboratively in the method to learn different feature sets, while ensuring consistent predictions. The group lasso module aims at identifying potential noisy class labels by computing a sample-wise ranking loss and penalizing noisy labeled images. The flipping module is devoted to flip the noisy labels approved by the two collaborative networks. Finally, the swap module aims at exchanging the ranking information between the networks by taking the Binary Cross Entropy (BCE) and the ranking loss functions into consideration. The pseudo-code of the proposed method is presented in Algorithm 1.

CCML follows the principle of a collaborative framework called Co-Training to exchange loss information between networks and to select samples associated with small loss values from the training set. Co-Training is a semi-supervised learning technique that was first proposed in [43] in order to overcome the labeled data insufficiency. Inspired by [36], we use two CNNs with the same architectures, which are enhanced with discrepancy modules to make them learn independent set of features, while attaining the same class distribution. Therefore, it becomes easier for the networks to find their potential faults. This improves the ability of selecting training images with clean labels immensely, since two networks are forced to learn different features and correct each other by exchanging their loss information. It is worth noting that, our proposed collaborative model is architecture-independent, since it does not rely on any specific network architecture. However, each deep neural network has different

characteristics, therefore the position of discrepancy module may differ from network to network. Another important property of the proposed algorithm is associated to its modularity. The flipping and the swap modules can be completely decoupled from the selected classification algorithm and from each other. Thus, these modules are applicable in the framework of any classification algorithm.

Algorithm 1: Consensual Collaborative Multi-label Learning

```
for epoch = 1,...,E do
    for mini-batch = 1,...,B do
        Calculate the ranking loss;
        if flipping = True then
            Flip the noisy labels;
            Recalculate the ranking loss;
        end
        Calculate the BCE loss;
        Combine the BCE loss and the ranking loss;
        Get samples associated with low loss values;
        Swap the ranking information;
        Update θ and θ̂;
    end
end
```

A. Problem Formulation

We consider a multi-label image classification problem with a training set $\mathcal{D} = \{(x_1, y_1), ..., (x_n, y_n)\}$, where x_i denotes the i^{th} training image labeled by a multi hot vector $y_i = (y_i^1, ..., y_i^m) \in \{0, 1\}^m$ representing the corresponding label over m classes, where $y_i^k = 1$ if $k \in y_i$, and $y_i^k = 0$ if $k \notin y_i$. We use two identical CNNs that are represented as f and g with parameters θ and $\hat{\theta}$, respectively. We employ BCE as our classification loss function for each network, which are denoted by L_f and L_g as shown below:

$$
\begin{aligned}
L_f(x_i) &= -\sum_{k=1}^{m} y_i^k \cdot \log\left(f_\theta(x_i^k)\right) + (1 - y_i^k) \cdot \log\left(1 - f_\theta(x_i^k)\right) \\
L_g(x_i) &= -\sum_{k=1}^{m} y_i^k \cdot \log(g_{\hat{\theta}}(x_i^k)) + (1 - y_i^k) \cdot \log(1 - g_{\hat{\theta}}(x_i^k)),
\end{aligned}
\tag{1}
$$

where $f_\theta(x_i^k)$ and $g_{\hat{\theta}}(x_i^k)$ are the predictions of the corresponding networks for class k in x_i.

In a collaborative learning framework, the networks must learn independent features while predicting the same class distribution. The networks are required to be capable of fixing mistakes of each other in the training process by exchanging information. If they do not learn diverse features on the same mini-batch, they cannot enhance predictions with each other mutually. To achieve the desired diversity, we propose a discrepancy module embed in our method, which includes two loss functions: discrepancy loss and consistency loss. The detail of the discrepancy module is described in the following section.

B. Discrepancy Module

The discrepancy module consists of two loss functions: i) the disparity loss (L_D); and ii) the consistency loss (L_C). The former makes sure that the networks learn distinct feature, and it is inserted in-between the chosen layers of each network. The layers are to be chosen depending on the network architecture, and the disparity loss function can be placed anywhere in the network. On the other hand, the consistency loss makes sure that the final predictions of the networks are similar. The networks should learn diverse and distinct features while predicting the same class distribution. Hence, we locate the consistency loss component at the end of the networks to reduce the discrepancy and achieve comparable predictions. Therefore, the final loss functions that are minimized by f and g are defined as:

$$
\begin{aligned}
Loss_f^b &= \frac{\sum_{i=1}^{R} L_f(x_i)}{R} + \lambda_1 \cdot L_C - \lambda_2 \cdot L_D \\
Loss_g^b &= \frac{\sum_{i=1}^{R} L_g(x_i)}{R} + \lambda_1 \cdot L_C - \lambda_2 \cdot L_D \ ,
\end{aligned}
\tag{2}
$$

where b represents the mini-batch, and R represents a set of samples selected based on low loss values as calculated in (1). λ_1 and λ_2 represent the weights for L_C and L_D, respectively. The disparity component gets the logits of the layers present before the component as input and calculates the discrepancy loss L_D. It is worth noting that the discrepancy module is placed between the two networks. Thus, the input is the logits of both of the networks. The calculated loss is defined as:

$$
\begin{aligned}
L_D &= M(\hat{F_i}, \hat{G_i}) \ , \\
\hat{G_i} = g_{\hat{\theta}(1:l)}(X_i) \ , \hat{F_i} &= f_{\theta(1:l)}(X_i) \ ,
\end{aligned}
\tag{3}
$$

where M is the choice of discrepancy module, and $\hat{F_i}$ and $\hat{G_i}$ represent the logits of the last layers before the component. l denotes the layer where the discrepancy component is placed, and $\theta(1:l)$ and $\hat{\theta}(1:l)$ stand for the parameters of networks until the layer l. X_i represents a batch of training images. The consistency module is placed after the last layer of the networks, and it gets the logits of the last layer as input:

$$
L_C = M(F_i, G_i) \ , \tag{4}
$$

where F_i and G_i denote the logits of the last layer of the networks, analogous to (3). In general, any statistical distance function that measures the difference between two probability distributions can be used as the discrepancy module M. Distance measures have a wide range of application areas. However, in machine learning, they are commonly used as loss functions that measure the distance between the model predictions and the labels [44,45]. Algorithms such as Maximum Mean Discrepancy (MMD) [46], and Wasserstein metric [47] are commonly used for diverging two distributions from one another. Since it is shown that MMD can be used to disentangle probability distributions [36], we exploit the same principles for MMD in our discrepancy modules. In detail, MMD is a distance measure between two distributions with a quadratic computational cost, and it is defined as the

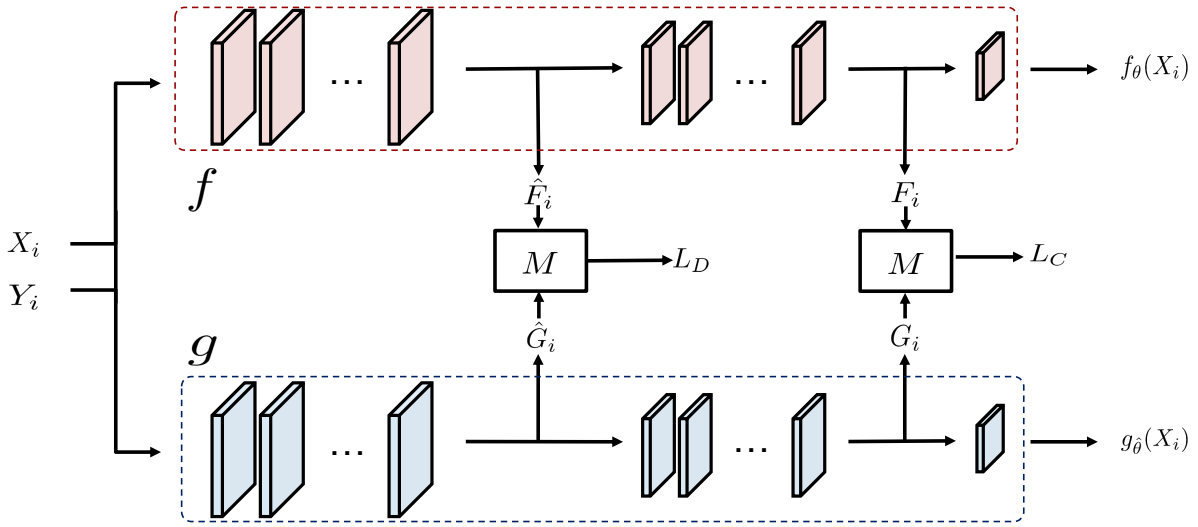

Fig. 2: Block diagram of the proposed discrepancy module. f and g represent the networks; red boxes are the inputs and the yellow boxes are the outputs. $\hat{F}_i$ and $\hat{G}_i$ stand for the intermediate logits of the networks. F_i and G_i stand for the logits of the last layers. M is the discrepancy component.

distance between the mean embeddings of the distributions in Reproducing Kernel Hilbert Space (RKHS) [48]. In other words, if the mean values of the functional mappings of the distributions to a high dimensional space are close to each other, the distance between the distributions is small. MMD to measure the difference between two distributions p and q can be defined as:

$$MMD(p,q) = \|\mu_p - \mu_q\|_H \quad . \tag{5}$$

where, μ_p and μ_q denote the expected values (means) of the distributions p and q, respectively. H represents the RKHS, and $\|\|_H$ represents the L_1 norm. Avoiding the explicit mapping of distributions with the kernel trick, an empirical estimate of MMD between p and q can be denoted as:

$$\begin{aligned} MMD(p,q) = \frac{1}{m^2}\Bigg(\sum_{i=1}^{m}\sum_{t=1}^{m} k(x_i^p, x_t^p) - \\ 2\cdot\sum_{i=1}^{m}\sum_{t=1}^{m} k(x_i^p, x_t^q) + \sum_{i=1}^{m}\sum_{t=1}^{m} k(x_i^q, x_t^q)\Bigg) \quad , \end{aligned} \tag{6}$$

where x_i^p and x_i^q are samples from respective distributions. Inspired by [36], we define the radial basis function kernel [49] as:

$$k(s_1, s_2) = \exp\left(-\frac{\|s_1 - s_2\|^2}{\sigma}\right) \quad . \tag{7}$$

C. Group Lasso Module

The BCE loss function is a widely used objective function in multi-label learning. However, while training set contains label noise, the networks may be biased toward noise in the training set and perform poorly. Thus, an additional mechanism is necessary to avoid the model misguided by noisy training sets. The additional mechanism can have many forms, such as regularization, noisy labeled image exclusion, or noise correction. Inspired by [41], we introduce a ranking error function capable of dealing with different types of noise in a multi-label scenario (missing and wrong class label assignments) without considering prior assumption. To this end, the proposed ranking error approach for missing class labels in [41] is extended to identify the wrong class label assignments as well. In addition, we do not use our ranking error function as a regularizer. Instead, we use it along with the BCE loss function to detect noisy labels within a mini-batch and exclude them from back-propagation. The exclusion of training images with noisy labels from back-propagation prevents overfitting to erroneous training set. The motivation behind using group lasso is to identify potential noisy labels in training set, given the opportunity to correct them. Furthermore, it provides information about label noise type. Let $E_{k,l}$ denotes the ranking error function:

$$E_{k,l}(x_i) = \max\Big(0, 2\cdot\big(f_l(x_i) - f_k(x_i)\big) + 1\Big), \tag{8}$$

where k represents a class index labeled by 1 and l represents a class index labeled by 0 for image x_i in the given batch. $f_k(x_i)$ and $f_l(x_i)$ denote the predictions of the classes k and l, respectively. The ranking error function gives a measure of potential noise in a class combination, which can be observed in Table I. In the case of a clean prediction the ranking error is equal to 0, otherwise, the ranking error function returns a positive value indicating a label noise.

The key idea behind the ranking error function is that if $k \in y_i$ and $l \notin y_i$, then $f_k(x_i) > f_l(x_i)$. However, the trustworthiness of the ranking error function depends very much on the prediction of the model. Hence, using $E_{k,l}(x_i)$ in the beginning of the training process, when the networks are unstable, may cause to misleading results.

The loss functions from class combinations are gathered by two loss terms using the group lasso to identify the potential label noise. The first loss term calculates an aggregated loss, considering missing class labels, and the second term calcu-

TABLE I: Ranking error function values

noise	f_l	f_k	E
clean label	0	1	0
noise on missing label	0	0	+1
noise on wrong label	1	1	+1
missing label and wrong label	1	0	+3

lates an aggregated loss for wrong class label assignments. This approach allows our algorithm to rank noisy training samples according to their noise rate and noise type by adjusting the importance factors (α and β) of the loss terms. The combined ranking loss per sample denotes as:

$$Lasso_f(x_i) = \alpha \cdot \sum_{l=a+1}^{m} \sqrt{\sum_{k=1}^{a} E_{k,l}^2(x_i)} \; + \\ \beta \cdot \sum_{k=b+1}^{m} \sqrt{\sum_{l=1}^{b} E_{k,l}^2(x_i)}, \tag{9}$$

where a and b denote the number of assigned labels and unassigned ones in an image label, respectively. $Lasso_g(x_i)$ is computed analogous to $Lasso_f(x_i)$.

Ranking errors per class is stored before summing them up into the ranking loss per sample, allowing noisy labels to be identified. The identified labels are afterward carried into the flipping module for correction. There may be more than one noisy label in an image, whether it is missing or a wrong label assignment. However, locating each of them correctly is a difficult task. When it is done wrongly, it may lead to detrimental effects on training. Thus, we select one label per image with the highest ranking-error for the flipping process. Finally, the group lasso module returns a ranking loss and a potential noisy class for each sample in a mini-batch.The returned ranking losses are added to the BCE loss by their importance factors to determine the noisy labels. Then the potential noisy labels are sent to the flipping module.

D. Flipping Module

The flipping module consists of two components: i) noisy class selector (NCS); and ii) noisy class flipper (NCF). The NCS takes the previously calculated ranking losses and potential noisy labels as input. When it comes to relabeling potential noisy labels, it is challenging to distinguish hard samples from noisy samples. To tackle this issue, the NCS component first compares the predicted labels by the two networks and eliminates the predictions that are not common, preventing relabeling false positives (i.e., labels that are not noisy). After both networks agree upon the noisy labels, the ranking losses of these noisy labels from the two networks are summed up. The NCS component then selects the samples with the largest ranking losses for flipping.

Let $l_i = (l_i^1, l_i^2, ..., l_i^n)$ and $\hat{l}_i = (\hat{l}_i^1, \hat{l}_i^2, ..., \hat{l}_i^n)$ be ranking losses, and $c_i = (c_i^1, c_i^2, ..., c_i^n)$ and $\hat{c}_i = (\hat{c}_i^1, \hat{c}_i^2, ..., \hat{c}_i^n)$ be the chosen potential noisy labels for every image in the i-th mini-batch with size n. The noisy labels C_i to be proceeded to NCF are chosen as follows:

$$I_i = \left\{ \frac{(l_i^k + \hat{l}_i^k)}{2} \middle| k \in \{1, ..., n\}, c_i^k = \hat{c}_i^k \right\}, \\ C_i = \{ c_i^a = \hat{c}_i^a | a \in D_i \} \;, \tag{10}$$

where D_i represents the set of indices of k largest elements in I_i.

The aim of the NCF component is to flip the labels of the identified noisy labels by the previous component. Flipping means that if a noisy class is labeled with 0, NCF turns the label into 1, and if the noisy class is labeled with 1, NCF turns the label into 0. After the flipping is applied, an additional group lasso component (see Sec. III-C) recalculates the ranking losses for the mini-batch with flipped labels. The flipped labels are also used for BCE loss calculation. An example for the mechanism of the flipping module is illustrated in Fig. 3.

As mentioned before, the success of the algorithm in finding and fixing the noisy labels depends on the accuracy of the network predictions, which are in general erroneous at the beginning of the training phase. Thus, if the label flipping module starts flipping noisy labels when the networks are unstable, it may flip the labels of hard but clean samples. To avoid that, the flipping module is initiated after the network is relatively stable. The value of this parameter can be set by analysing the learning curve of the networks. It is also important to point out that the number of labels that are flipped is significant for the success of the module. As the networks learn from each other's mistakes, the number of consented labels may increase, and a high flipping percentage may have negative effects.

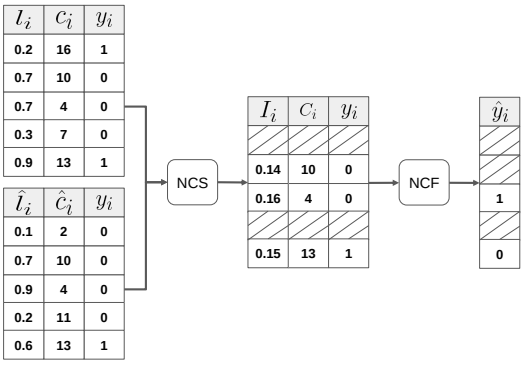

Fig. 3: A qualitative example of flipping noisy labels in the flipping module: l_i and $\hat{l}_i$ represent the ranking losses. c_i and $\hat{c}_i$ imply the potential noisy class indexes. NCS outputs the consented noisy labels C_i and sums up the ranking losses from both of the networks. NCF gets the labels y_i as input and flips a percentage of the consented noisy labels with the largest ranking losses. In this example the percentage of flipping is set to 2/3.

E. Swap Module

The swap module adds the ranking losses, which are calculated by the (10), to the BCE loss, and selects R samples that are associated with the lowest loss values for each network. The ranking information of samples associated with the lowest loss value in R are exchanged between the two networks, where the network f uses samples associated with the lowest R that are found by the network g to update its weights, and vice versa. Swapping ranking losses can be denoted as follows:

$$\begin{aligned} B_i^f &= \frac{\sum_{i=1}^{R} L_f(x_i) + \alpha \cdot Lasso_f(x_i)}{R} \quad , \quad x_i \in m_g^R \\ B_i^g &= \frac{\sum_{i=1}^{R} L_g(x_i) + \alpha \cdot Lasso_g(x_i)}{R} \quad , \quad x_i \in m_f^R \quad , \end{aligned} \tag{11}$$

where B_i^f and B_i^g are the compound losses for each network, and m_f^R and m_g^R are the set of images that results in the R for the networks f and g respectively. α is the trade off parameter representing the strength of $Lasso_f$ and $Lasso_g$ for selecting samples with low loss values. A visualization of the process is illustrated in Fig. 4.

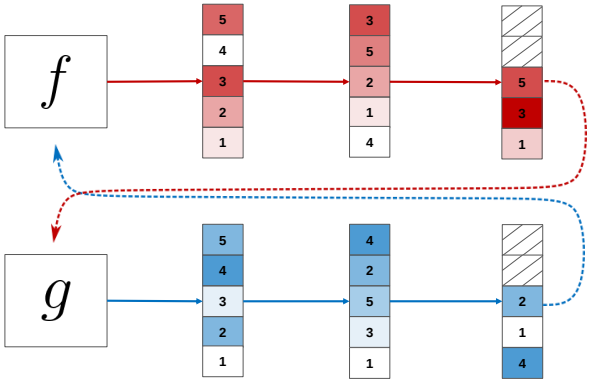

Fig. 4: A qualitative example to describe the swap module. The two networks exchange the ranking information. Then, each network updates its weights using the samples with smaller loss values that are found by the opposite network.

IV. Dataset Description and Experimental Setup

In this Section, we describe the datasets that are used in the experiments and present the experimental setup, including the network architecture, optimal values for hyperparameters and considered synthetic label noise injection approach.

A. Dataset Description

We conducted experiments on two different benchmark RS datasets. The first dataset is the Ireland subset of the BigEarthNet (denoted as IR-BigEarthNet) benchmark archive [8], which consists of 15894 Sentinel-2 multispectral images acquired between June 2017 and May 2018 over Ireland. Each image was annotated by multiple land-cover classes provided by 2018 CORINE Land Cover Map (CLC) inventory. Sentinel-2 images contain 13 spectral bands with varying spatial resolutions. Each image in IR-BigEarthNet is a section of: i) 120×120 pixels for the bands that have a spatial resolution of 10m; ii) 60×60 pixels for the bands that have a spatial resolution of 20m; and iii) 20×20 pixels for the bands that have a spatial resolution of 60m. In the experiments, we used the bands with 10m and 20m spatial resolutions (and thus 10 bands per image is used in total), excluding the two 60m bands due to their very low spatial resolution and small pixel size. The cubic interpolation was applied to 20m bands of each image to have the same pixel sizes associated with each band. In the experiments, we exploited the BigEarthNet-19 land-cover class nomenclature proposed in [50] and eliminated 5 classes which are represented with a significantly small number of images in the dataset, leading to 12 classes in total. The number of labels associated with each image varies between 1 and 7, while 97,4% of images contain less than 5 labels. IR-BigEarthNet was divided into a validation set of 3839 images, a test set of 3856 images, and a training set of 8192 images.In the experiments, the images that are fully covered by seasonal snow, cloud and cloud shadow were not used as suggested in [8]. An example of images from IR-BigEarthNet together with their multi-labels is given in Fig. 5. Table II shows the number of training, validation and test samples associated to each considered class in IR-BigEarthNet.

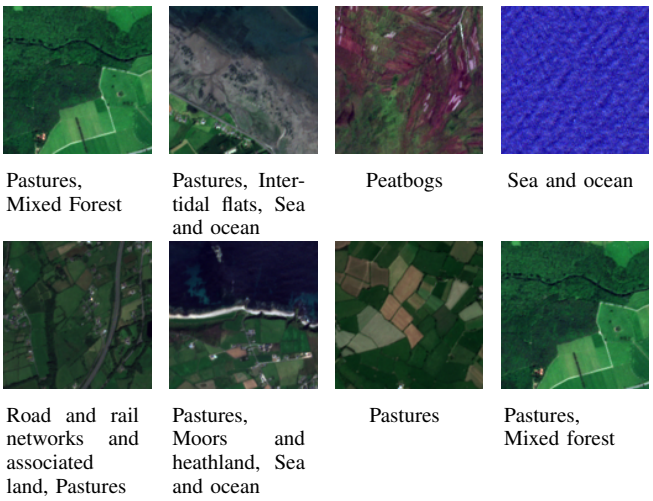

Fig. 5: An example of images with their multi-labels from the IR-BigEarthNet dataset.

The second dataset is the UC Merced Land Use (denoted as UCMERCED) archive that consists of 2100 images selected from aerial orthoimagery and downloaded from the USGS National Map of the following US regions: Birmingham, Boston, Buffalo, Columbus, Dallas, Harrisburg, Houston, Jacksonville, Las Vegas, Los Angeles, Miami, Napa, New York, Reno, San Diego, Santa Barbara, Seattle, Tampa, Tucson, and Ventura [21]. Each image is of size 256x256 pixels and has a spatial resolution of 30m. In the experiments, we used the multi-label annotations of UCMERCED images that were obtained based on visual inspection [51]. The total number of class labels is 17, while the number of labels associated with each image varies between 1 and 7. The number of training, validation and test samples in the multi-label UCMERCED dataset is given

TABLE II: Number of samples per class in training, validation and test sets of the IR-BigEarthNet dataset.

Class	Train	Val	Test
Urban fabric	818	369	371
Arable lands	2860	1404	1397
Pastures	5723	2725	2739
Complex cultivation patterns	510	240	226
Land principally occupied by agriculture, with significant areas of natural vegetation	896	420	414
Broad-leaved forest	410	205	199
Coniferous forest	1156	545	524
Mixed forest	588	273	296
Moors, heathland and sclerophyllous vegetation	467	217	219
Transitional woodland, shrub	708	345	369
Inland wetlands	811	400	415
Marine waters	2087	894	900

in Table III. Fig. 6 shows an example of images together with their associated multi-labels from the UCMERCED dataset.

TABLE III: Number of samples per class in training, validation and test sets of the UCMERCED dataset.

Class	Train	Val	Test
Airplane	70	15	15
Bare Soil	506	103	109
Buildings	482	102	107
Cars	631	125	130
Chaparral	77	21	17
Court	73	16	16
Dock	70	15	15
Field	73	15	15
Grass	683	145	147
Mobile home	70	17	15
Pavement	917	189	194
Sand	210	43	41
Sea	70	15	15
Ship	70	16	16
Tanks	70	15	15
Trees	702	155	152
Water	142	31	30

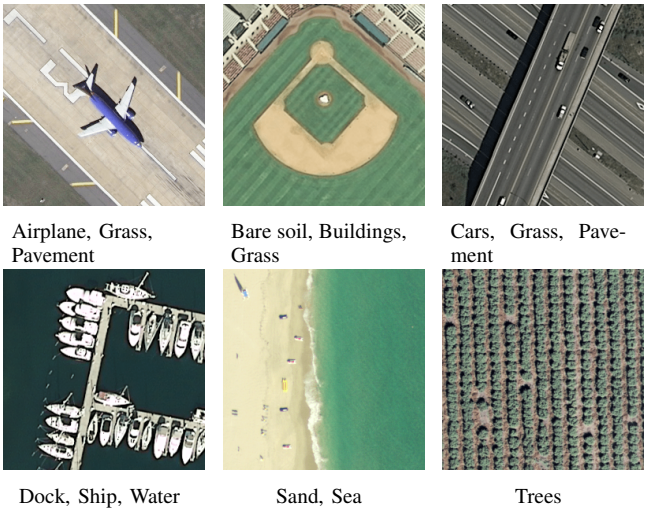

Fig. 6: An example of images with their multi-labels from the UCMERCED dataset.

B. Experimental Setup

In the experiments, we used ResNet [52], which is a well established network architecture. Among different versions of ResNet, each of which includes different numbers of layers, ResNet50 with 50 layers was chosen. Moreover, we used in our network improved residual units [53] that enhance the network's generalization performance and make training easier by introducing additional non-linearities. We utilized the Adam optimizer [54] with a learning rate of 10^{-3}. The batch sizes for IR-BigEarthNet and UCMERCED was set to 256 and 32, respectively. We reduce the training time and converge safer by avoiding noise with large batch sizes. We report the results of training obtained after 100 epochs. The hyperparameter tuning and the initial tests were conducted on NVIDIA Tesla P-100 GPU with 16 GB of RAM, while the model training and further experiments were conducted on a Tesla V100 GPU with 32 GB RAM.

Within the swap module of the proposed CCML method, the networks swap the calculated low loss sample information with each other. According to this information, each network chooses a certain amount of low loss samples of the opposite network and updates its weights by using them. The goal of this process is to remove the noisy samples from back-propagation. Without knowing the exact noise amount and distribution in the labels, optimizing this parameter is very difficult. In addition, excluding a very high number of samples from back-propagation degrades the accuracy of the predictions. Networks may cannot learn diverse features using insufficient number of samples. On the other hand, excluding as a very small number of samples also affects the accuracy adversely because the network may overfit to the noise present in the data. According to our observations, updating the weights using the 75% of the samples associated with small loss values at each iteration is the optimal solution to this tradeoff.

The discrepancy modules has an essential role in the success of the proposed CCML method. The ability of the network to learn diverse features from the data depends on the discrepancy modules. In our proposed CCML we used MMD for the discrepancy modules.The optimal value of the σ in the RBF kernel was determined as 10.000. We observed that low sigma values were ineffective for measuring the distance between the logits of the two networks used in the proposed method. Using the RBF kernel, the discrepancy module first maximizes the diversity to learn diverse features, and then minimizes it to learn the same class distributions. The module achieves this result by adding the respective losses, namely L_C and L_D, to the final loss terms of the networks, weighting them by λ_1 and λ_2 as follows. The value of λ_1 was set to 0.25, and the value of λ_2 was set to 0.5. The diversity component of the module was placed approximately within the second quarter of the convolutional layers. Forcing networks to diverge in an early stage and converging them thereafter teaches the networks distinct feature sets while keeping the predictions consistent.

An additional mechanism along with the BCE loss is necessary to identify the potential noisy samples. For this purpose, a ranking loss for every sample was calculated and

added to the BCE loss by a factor that was set to 0.2 as the result of our experiments. There are also parameters α and β that control the effects of two different noise types on the calculation of the ranking loss. A group lasso module with a higher α concentrates more on finding the missing class labels, whereas a higher β gives a heavier weight to detecting the extra class label assignments. An extra class label assignment was more harmful than a missing class label in our setting, where every sample was annotated with the minority of the present classes. Thus, we set α to 0.5 and β to 1.0.

As mentioned in Section III, it is crucial to choose the right time to start flipping potential noisy labels because the networks are not stable in a very early stage of the training, and thus the predictions may not be accurate. Therefore, the early initiation of the flipping module may cause erroneous flips. To prevent this, we started the flipping process after 90% of epochs was reached. Furthermore, the flipping module flips labels when the two networks agree that the label is noisy. As the networks learn from each other and get more stable, the number of agreed labels may increase. However, the networks may agree on the labels of hard classes instead of noisy ones, and flipping too much of them would decrease the performance of the model. The flipping module flips only the 5% of the agreed classes every iteration after it is initiated to avoid this problem.

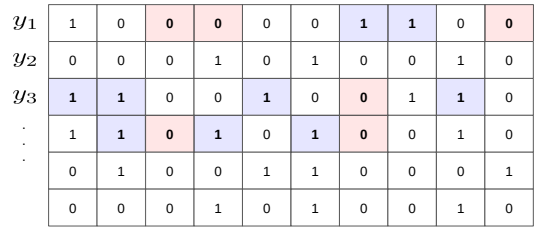

Fig. 7: The considered label noise injection approach: Random Noise per Sample (RNS) with a sampling rate 0.5 and class rate of 0.5. The colored cells represent the introduced artificial label noise. Cells in blue represent "0" value in ground truth labels that are flipped to "1", while those in red represent "1" value in ground truth labels that are flipped to "0".

To verify the effectiveness of the proposed method , we added synthetic noise to the labels of the IR-BigEarthNet and UCMERCED datasets. To ensure that both types of label noise (missing label and wrong label) are introduced to the multi-label training set, we designed a label noise injection approach called Random Noise per Sample (RNS). RNS chooses labels randomly from each mini-batch by a predefined percentage called the sampling rate. Afterward, to apply noise, RNS randomly flips a certain number of selected labels. The number of flipped labels driven by a parameter we called as class rate. An illustration of the considered label noise injection approach is shown in Fig. 7. The figure shows an example scenario for RNS with a sampling rate of 0.5 and a class rate of 0.5. Three samples out of six are selected, and label noise is randomly applied to half of their labels. In our experiments, the ratio of the introduced noise was varied from 0% to 50%.

The results of the experiments were provided in terms of three performance metrics: 1) precision (P); 2) recall (R); and 3) F_1-Score with the micro averaging strategy. For an explanation of these metrics, the reader is referred to [6]. Since the proposed CCML includes two networks that run simultaneously, the network with the best F_1 validation score has been chosen for evaluation. To further study the performance and the behavior of the proposed CCML, the class-based F_1 scores are reported in comparative plots.

We compared our proposed method with ResNet50 as the baseline. Since our proposed CCML is architecture-independent and can be used with any network architecture, we select the base model (ResNet50) due to its proven effectiveness. To have a fair comparison with the proposed method, we realized the CCML modules within the ResNet50 architecture. We trained CCML with the same number of training epochs and hyperparameters used in the baseline model.

V. Experimental Results

This section reports comparative results concerning the baseline model (ResNet 50) and the proposed CCML method. We report the experimental results on the IR-BigEarthNet and multi-label UCMERCED datasets, followed by class-wise performance analysis.

TABLE IV: Precision, Recall and F_1 scores obtained by the proposed CCML and the baseline [52] using IR-BigEarthNet dataset under different noise rates.

Noise	Precision (%)		Recall (%)		F_1 (%)	
Rate	Baseline	CCML	Baseline	CCML	Baseline	CCML
0%	79.4	**88.4**	**73.1**	69.9	76.2	**78.0**
10%	87.6	**89.1**	69.0	**69.3**	77.2	**78.0**
20%	87.8	**90.2**	**68.7**	**68.7**	77.1	**78.0**
30%	84.0	**88.2**	67.2	**68.9**	74.7	**77.4**
40%	76.4	**88.4**	65.1	**69.3**	70.3	**77.7**
50%	62.5	**87.5**	57.6	**62.1**	60.0	**72.6**

A. Results on Multi-Label IR-BigEarthNet Dataset

The results on the IR-BigEarthNet are presented in Table IV. By analyzing the results in Table IV, one can see that the proposed method outperforms the base model. Especially under extreme noise rates such as 40% and 50%, CCML achieves in average 10% better F_1 scores. The performance of both models does not decrease significantly by the increasing noise rates up to 30%. The reason is that the number of training images for each class is still sufficient for the networks to learn and predict them correctly, despite the introduced label noise.

For a better understanding, we select six representative classes from the IR-BigEarthNet and report their class-based F_1 scores in Fig. 8. The class-based F_1 scores in Fig. 8 reveal that both the baseline model and the proposed CCML learn the classes represented by sufficient training images better than the classes that do not include enough sufficient number of training images. The classes with a sufficient number of training images (e.g. "Marine waters") obtaining the highest F_1 scores

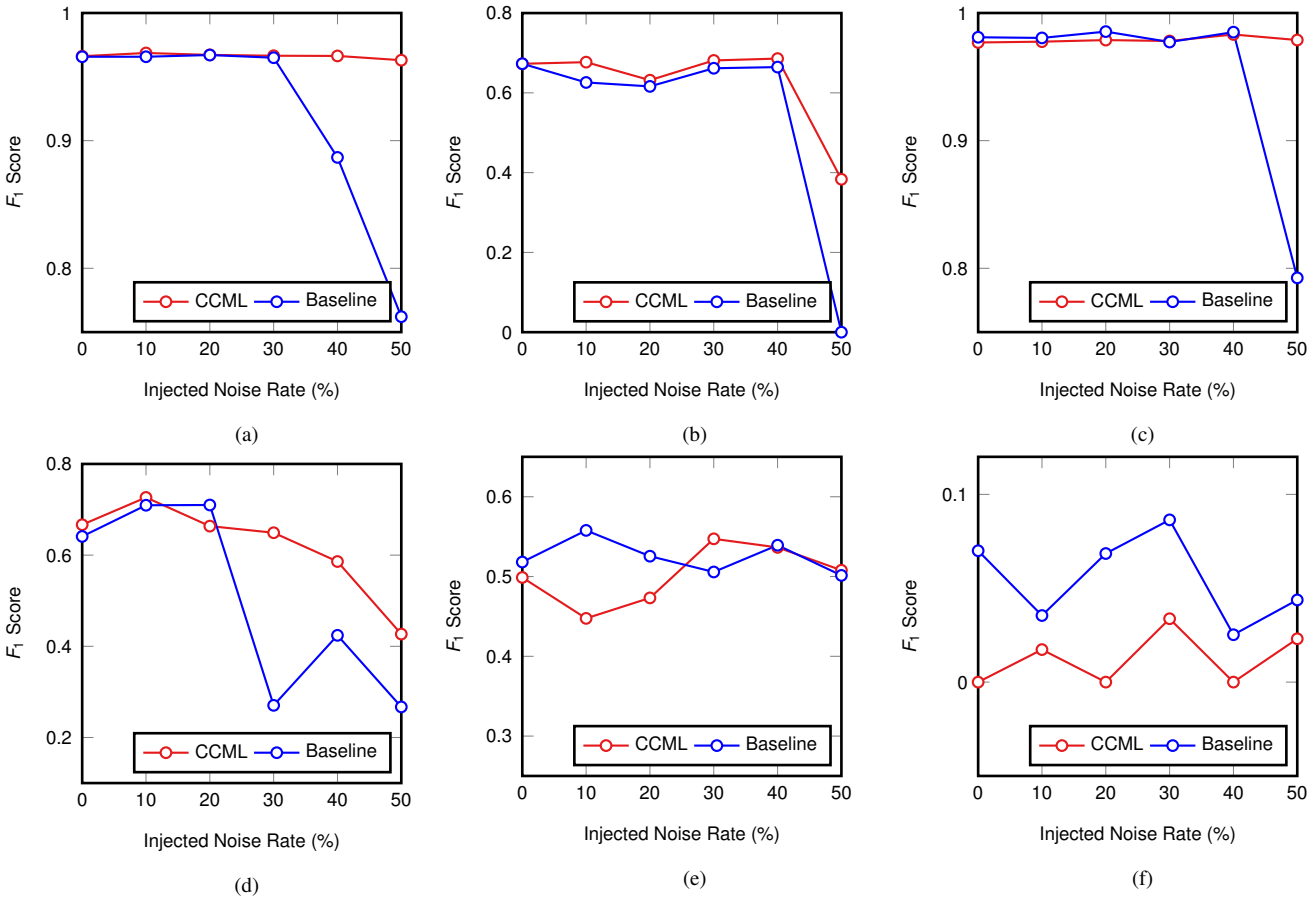

Fig. 8: Different noise rates versus class based F_1 scores obtained by the proposed CCML and the baseline for classes: (a) Pastures; (b) Urban fabric; (c) Marine waters; (d) Inland wetlands; (e) Moors, heathland and sclerophyllous vegetation; and (f) Complex cultivation patterns in IR-BigEarthNet dataset.

under different noise rates. Even high noise rates, such as 40%, do not significantly reduce the F_1 scores of these classes. The classes with the highest number of training images still have many images to learn from, despite applying high noise rates. On the other hand, the classes with insufficient training images (e.g., "Complex cultivation pattern", and "Moors, heathland and sclerophyllous vegetation") receive a low F_1 score over different noise rates.

The proposed CCML outperforms the baseline model in the majority of classes, especially under high noise rates, while obtaining comparative results against the baseline model. Both models perform poorly in the "Complex cultivation patterns" class, which is among the classes with no sufficient training images, with an average of 0.03% in F_1 score. In some cases (e.g., "Inland wetlands"), the baseline and CCML performances are comparable over low rates of label noises. However, by increasing the noise rate, the CCML holds the F_1 scores relatively high compared to the baseline, demonstrating the stability of CCML under extreme noise rates scenario.

B. Results on Multi-Label UCMERCED Dataset

The comparative evaluation results for the multi-label UCMERCED dataset are presented in Table V. In the experiments on multi-label UC Merced Land Use, increasing noise rates severely deteriorate the baseline model's learning process. As shown in Table V, noise creates instability on the baseline model. On the other hand, the proposed CCML provides relatively stable performance and achieves a certain degree of robustness against high label noise rates. Although the base model performs better than the proposed CCML under lower noise rates, the CCML achieves considerably higher F_1 scores with 40% and 50% noise rates. We should note that the ResNet50 is a powerful model, capable of tolerating even a relatively high noise rate, but it is not stable under extreme

TABLE V: Precision, Recall and F_1 scores obtained by the proposed CCML and the baseline [52] using UCMERCED dataset under different noise rates.

Noise Rate	Precision (%)		Recall (%)		F_1 (%)	
	Baseline	CCML	Baseline	CCML	Baseline	CCML
0%	**74.2**	69.6	**69.8**	64.1	**71.9**	66.7
10%	**71.9**	67.7	**70.8**	66.3	**71.4**	67.0
20%	**69.9**	68.5	62.3	**70.1**	65.7	**69.3**
30%	70.5	**72.2**	65.8	**70.6**	68.0	**71.4**
40%	58.7	**64.1**	61.4	**71.9**	59.7	**67.7**
50%	32.3	**61.2**	44.9	**68.2**	37.5	**64.5**

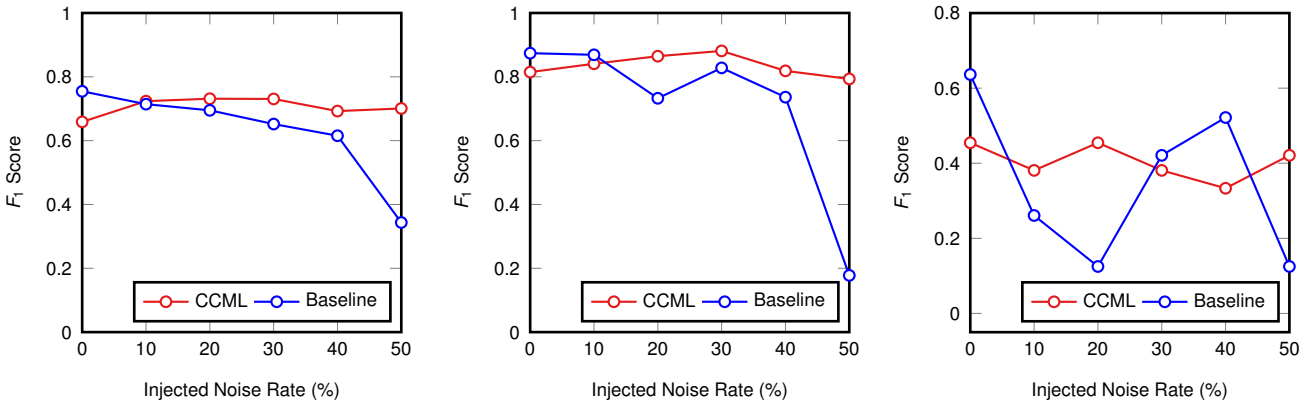

Fig. 9: Different noise rates versus class based F_1 scores obtained by the proposed CCML and the baseline for classes: (a) Cars; (b) Pavement; (c) Airplane in the UCMERCED dataset.

noise rate. However, employing ResNet50 as a classier with our proposed collaborative framework makes it more stable and robust against higher noise rates.

A comparison between Table IV and Table V reveals that the multi-label UCMERCED dataset achieves lower F_1 scores compared to the IR-BigEarthNet under the same experimental setup. The reason for this is the number of training images. The UCMERCED dataset includes 1470 training images, while IR-BigEarthNet includes 8192 training images. The availability of a large number of training images in the IR-BigEarthNet makes it easier for the classifiers to learn the underlying class distribution from the training set.

In Fig. 9, three representative classes from the UCMERCED dataset are chosen and illustrated to analyze the obtained class-based F_1 scores. The class-based F_1 scores demonstrate a similar pattern for the classes with a sufficient number of training images and classes that do not reach enough training images. For example, the "Airplane" class contains the lowest number of training images among the other classes, which is not considered a sufficient number of training images (70 images). The "Pavement" class contains the highest number of training samples in the training set (more than 900 images). The performance of the baseline model is not stable under different noise rates, as it can be seen in Fig. 9. For example, for the "Airplane" class, the F_1 scores are unstable for the baseline model, while the proposed CCML scores are relatively stable.

Similar to the IR-BigEarthNet results in Fig. 8, the F_1 scores of the classes with a sufficient number of training images are the highest among the other classes. The F_1 scores of the baseline model for the lower noise rates are higher than CCML since the baseline model uses all the training samples, while CCML excludes 25% of the training samples. This becomes more critical in the case of the UCMERCED dataset with a small number of training samples.

VI. Conclusion and Discussion

In this work, we have proposed a novel Consensual Collaborative Multi-label Learning (CCML) method to overcome the adverse effects of multi-label noise in the context of scene classification of RS images. The proposed method includes four main modules: 1) group lasso module; 2) discrepancy module; 3) flipping module; and 4) swap module. The group lasso module detects the individual noisy labels assigned to the training image, while the discrepancy module is devoted to ensuring that two collaborative networks learn different features while obtaining the same predictions. The proposed CCML method exploits the flipping module to correct the identified noisy labels and the swap module to exchange the ranking information between two networks. The CCML method automatically identifies the different multi-label noise types that are associated to missing and wrong class label annotations. To the best of our knowledge, CCML is the first method to simultaneously tackle the adverse effects of the two types of multi-label noise without making any prior assumption.

The performance of the proposed method was evaluated under different noise rates trained on two publicly available multi-label benchmark RS image archives. We used the IR-BigEarthNet and the UCMERCED archives. The experimental results confirm the effectiveness of the proposed method in specific settings where deterministic label noise is introduced to the multi-label training sets. Furthermore, CCML shows more robustness with respect to the baseline model under high-noise regimes (e.g., in the presence of high rate label noise such as 30% and more).

We would like to point out that developing efficient techniques for handling label noise in multi-label training sets is becoming more and more important. On the one side, due to the increased volume of RS image archives, manual large-scale image labeling is time-demanding and costly (and thus not fully feasible). On the other side, making use of zero-cost labeling by the use of available thematic products can introduce label noise to the training set. In this context, the proposed CCML is very promising as it allows to identify of the potential multi-label noise within the training set without considering any prior assumption. We emphasize that this is a very important advantage because one of the main objectives of noise-robust methods is to be applicable and

generalizable to any label noise distribution. Furthermore, the proposed method is intrinsically classifier-independent. Even if in our framework it is implemented in the context of CNNs (because of their efficiency for RS image classification), it can be adapted easily for any other classifiers or network architectures. This can be done by a proper adjustment on each module together with its input parameters.

As a final remark, it is worth noting that the proposed CCML provides much more accurate results, when an adequate number of training images related to each class is available. This may not be always possible and the datasets can be imbalanced in the operational RS MLC applications. As a future work, to address this issue we plan to develop an adaptive class-weighted loss function. Another solution can be considering the data augmentation techniques.

ACKNOWLEDGMENT

This work was supported by the European Research Council under the ERC Starting Grant BigEarth-759764.

REFERENCES

[1] L. Bruzzone and B. Demir, "A review of modern approaches to classification of remote sensing data," in *Land Use and Land Cover Mapping in Europe: Practices & Trends*, 1st ed., I. Manakos and M. Braun, Eds. Springer, Dordrecht, 2014, ch. 9, pp. 127–143.

[2] I. Shendryk, Y. Rist, R. Lucas, P. Thorburn, and C. Ticehurst, "Deep learning - a new approach for multi-label scene classification in planetscope and sentinel-2 imagery," in *IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 1116–1119.

[3] A. Zeggada, F. Melgani, and Y. Bazi, "A deep learning approach to UAV image multilabeling," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 694–698, 2017.

[4] Y. Hua, L. Mou, and X. X. Zhu, "Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional lstm network for multi-label aerial image classification," *Journal of Photogrammetry and Remote Sensing*, vol. 149, pp. 188–199, 2019.

[5] A. Alshehri, Y. Bazi, N. Ammour, H. Almubarak, and N. Alajlan, "Deep attention neural network for multi-label classification in unmanned aerial vehicle imagery," *IEEE Access*, vol. 7, pp. 119 873–119 880, 2019.

[6] G. Sümbül and B. Demir, "A deep multi-attention driven approach for multi-label remote sensing image classification," *IEEE Access*, vol. 8, pp. 95 934–95 946, 2020.

[7] H. Yessou, G. Sümbül, and B. Demir, "A comparative study of deep learning loss functions for multi-label remote sensing image classification," in *IEEE International Geoscience and Remote Sensing Symposium*, 2020.

[8] G. Sümbül, M. Charfuelan, B. Demir, and V. Markl, "Bigearthnet: A large-scale benchmark archive for remote sensing image understanding," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 5901–5904, 2019.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[10] G. Büttner, J. Feranec, G. Jaffrain, L. Mari, G. Maucha, and T. Soukup, "The corine land cover 2000 project," *EARSeL eProceedings*, vol. 3, no. 3, pp. 331–346, 2004.

[11] C. Paris and L. Bruzzone, "A novel approach to the unsupervised extraction of reliable training samples from thematic products," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–19, 2020.

[12] G. Jaffrain, C. Sannier, A. Pennec, and H. Dufourmont, "Corine land cover 2012 - final validation report," European Environment Agency, Tech. Rep., 2017. [Online]. Available: https://land.copernicus.eu/user-corner/technical-library/clc-2012-validation-report-1

[13] H. Jain, Y. Prabhu, and M. Varma, "Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 935–944.

[14] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep convnet for multi-label classification with partial labels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 647–657.

[15] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[16] K. Yi and J. Wu, "Probabilistic end-to-end noise correction for learning with noisy labels," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7017–7025.

[17] P. Ulmas and I. Liiv, "Segmentation of satellite imagery using u-net models for land cover classification," *arXiv preprint arXiv:2003.02899*, 2020.

[18] Y. Hua1, S. Lobry, L. Mou, D. Tuia, and X. X. Zhu, "Learning multi-label aerial image classification under label noise: A regularization approach using word embeddings," in *IEEE International Geoscience and Remote Sensing Symposium*, 2020.

[19] H. Song, M. Kim, D. Park, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *arXiv preprint arXiv:2007.08199*, 2020.

[20] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[21] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2010, pp. 270–279.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *European Conference on Computer Vision*, pp. 740–755, 2014.

[23] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1944–1952.

[24] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," 2016.

[25] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.

[26] C. M. Teng, "Evaluating noise correction," in *PRICAI 2000 Topics in Artificial Intelligence*, R. Mizoguchi and J. Slaney, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 188–198.

[27] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Advances in Neural Information Processing Systems*, 2017, pp. 5109–5118.

[28] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," *arXiv preprint arXiv:1406.2080*, 2014.

[29] B. Yuan, J. Chen, W. Zhang, H.-S. Tai, and S. McMains, "Iterative cross learning on noisy labels," in *IEEE Winter Conference on Applications of Computer Vision*, 2018, pp. 757–765.

[30] M. Dehghani, A. Mehrjou, S. Gouws, J. Kamps, and B. Schölkopf, "Fidelity-weighted learning," *arXiv preprint arXiv:1711.02799*, 2017.

[31] X. Liu, S. Li, M. Kan, S. Shan, and X. Chen, "Self-error-correcting convolutional neural network for learning with noisy labels," in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2017, pp. 111–117.

[32] X. Wu, R. He, Z. Sun, and T. Tan, "A light cnn for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.

[33] D. T. Nguyen, T.-P.-N. Ngo, Z. Lou, M. Klar, L. Beggel, and T. Brox, "Robust learning under label noise with iterative noise-filtering," *arXiv preprint arXiv:1906.00216*, 2019.

[34] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*, 2018, pp. 2304–2313.

[35] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems*, 2018, pp. 8527–8537.

[36] Y. Han, S. ROY, L. Petersson, and M. Harandi, "Learning from noisy labels via discrepant collaborative training," in *Winter Conference on Applications of Computer Vision*, 2020, pp. 3169–3178.

[37] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *arXiv preprint arXiv:1803.09050*, 2018.

[38] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," *arXiv preprint arXiv:1712.09482*, 2017.
[39] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems*, 2018, pp. 8778–8788.
[40] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5051–5059.
[41] S. S. Bucak, R. Jin, and A. K. Jain, "Multi-label learning with incomplete class assignments," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2801–2808.
[42] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.
[43] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Conference on Computational Learning Theory*, 1998, pp. 92–100.
[44] C. Liu and H.-Y. Shum, "Kullback-leibler boosting," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. I–I.
[45] B. Bigi, "Using kullback-leibler distance for text categorization," in *European Conference on Information Retrieval*. Springer, 2003, pp. 305–319.
[46] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.
[47] S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.
[48] M. A. Alvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review," *arXiv preprint arXiv:1106.6251*, 2011.
[49] J.-P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," *Kernel Methods in Computational Biology*, vol. 47, pp. 35–70, 2004.
[50] G. Sümbül, J. Kang, T. Kreuziger, F. Marcelino, H. Costa, P. Benevides, M. Caetano, and B. Demir, "Bigearthnet dataset with a new class-nomenclature for remote sensing image understanding," *arXiv*, pp. arXiv–2001, 2020.
[51] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, "Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method," *Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1144–1158, 2018.
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[53] K. He, X. Zhang, S. Ren, and S. Jian, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.