A Consensual Collaborative Learning Method for Remote Sensing Image Classification under Noisy Multi-Labels

Ahmet Kerem Aksoy, Mahdyar Ravanbakhsh, Tristan Kreuziger, Begüm Demir

Remote Sensing Image Analysis (RSiM) Group, TU Berlin

IEEE International Conference on Image Processing 2021

Introduction

- ✓ Multi-label remote sensing (**RS**) image scene classification methods aim to automatically assign **multiple land-cover class** labels to each image.
- ✓ Deep neural networks (**DNNs**) have recently shown a great potential for multi-label classification problems.
- ✓ Most of the DNNs require a large amount of **annotated** multi-label training samples.

Problem: Collecting annotations is very **costly** in terms of human time/effort and needs **expertise**.

RS Images	Multi-Labels

Discontinuous urban fabric
Coniferous forest
Mixed forest
Industrial or commercial units

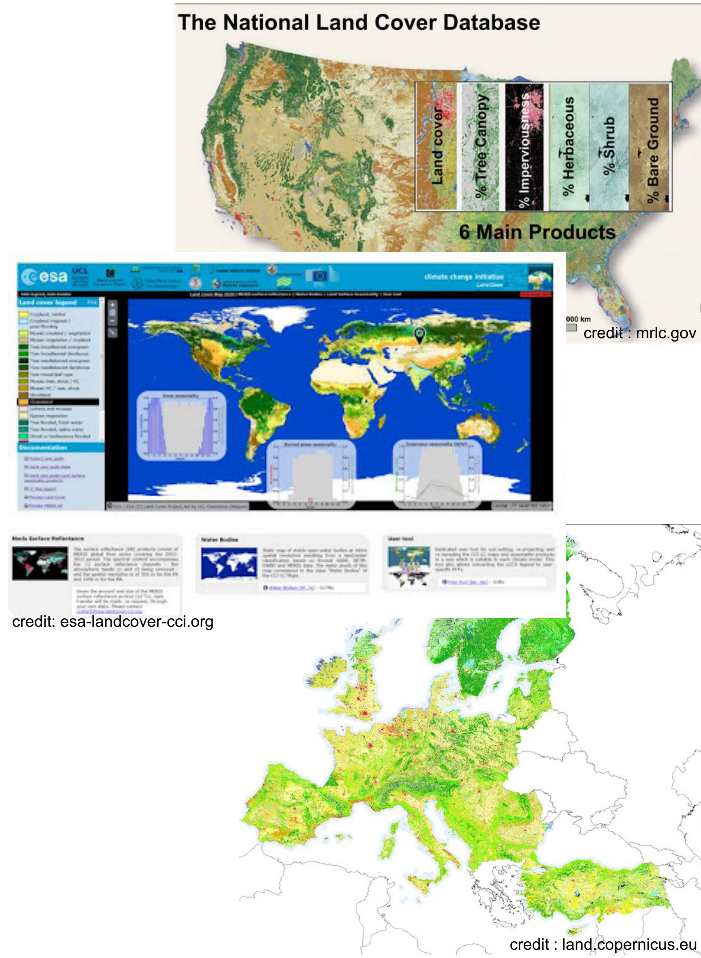
Discontinuous urban fabric
Industrial or commercial units
Non-irrigated arable land

Introduction

Solution: use the publicly available **thematic products** (digital maps) as labelling sources.

✓ Thematic products can be used to generate large scale training sets with zero-labeling-cost, covering wide areas, such as:

- **National scale,** e.g., American National Land Cover Dataset, DFD Land Use and Land Cover Product for Germany;
- **Global scale,** e.g., Global Land Cover, the ESA CCI-LC product;
- **Continental scale,** e.g., Corine Land Cover maps at European level.



credit : mrlc.gov

credit: esa-landcover-cci.org

credit : land.copernicus.eu

- G. Jaffrain, C. Sannier, A. Pennec, and H. Dufourmont, “Corine land cover 2012 - final validation report,” European Environment Agency, Tech. Rep., 2017.
- M. Hovenbitzer, F. Emig, C. Wende, S. Arnold, M. Bock, and S. Feigenspan. "Digital land cover model for Germany–DLM-DE." In Land Use and Land Cover Mapping in Europe, pp. 255-272. Springer, Dordrecht, 2014.
- ESA. Land Cover CCI Product User Guide Version 2. Technical Report, 2017.

Introduction: Noisy Labels

✓ However, thematic maps can produce noisy labels since:

1) Errors in the map due to different the strategies used to generate the map;
2) Changes in land-use/land-cover after the construction of the maps;
3) Geolocation errors due to the residual misalignment between a digital map and a satellite image.

Problem: Training sets with noisy labels are constructed, which can distort the learning process.

- G. Buttner, 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.
- 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.
- G. Jaffrain, C. Sannier, A. Pennec, and H. Dufourmont, “Corine land cover 2012 - final validation report,” European Environment Agency, Tech. Rep., 2017.

Introduction: Noisy Labels

✓ Two types of label noise can be present in a training image with multi-labels:

1. Missing labels;
2. Wrong labels.

Discontinuous urban fabric

Coniferous forest

Mixed forest

Missing label Industrial or commercial units

Discontinuous urban fabric

Industrial or commercial units

Non-irrigated arable land

Wrong label Coniferous forest

✓ Methods that are **robust** to the multi-label **noise** are required.

Aims of the Work

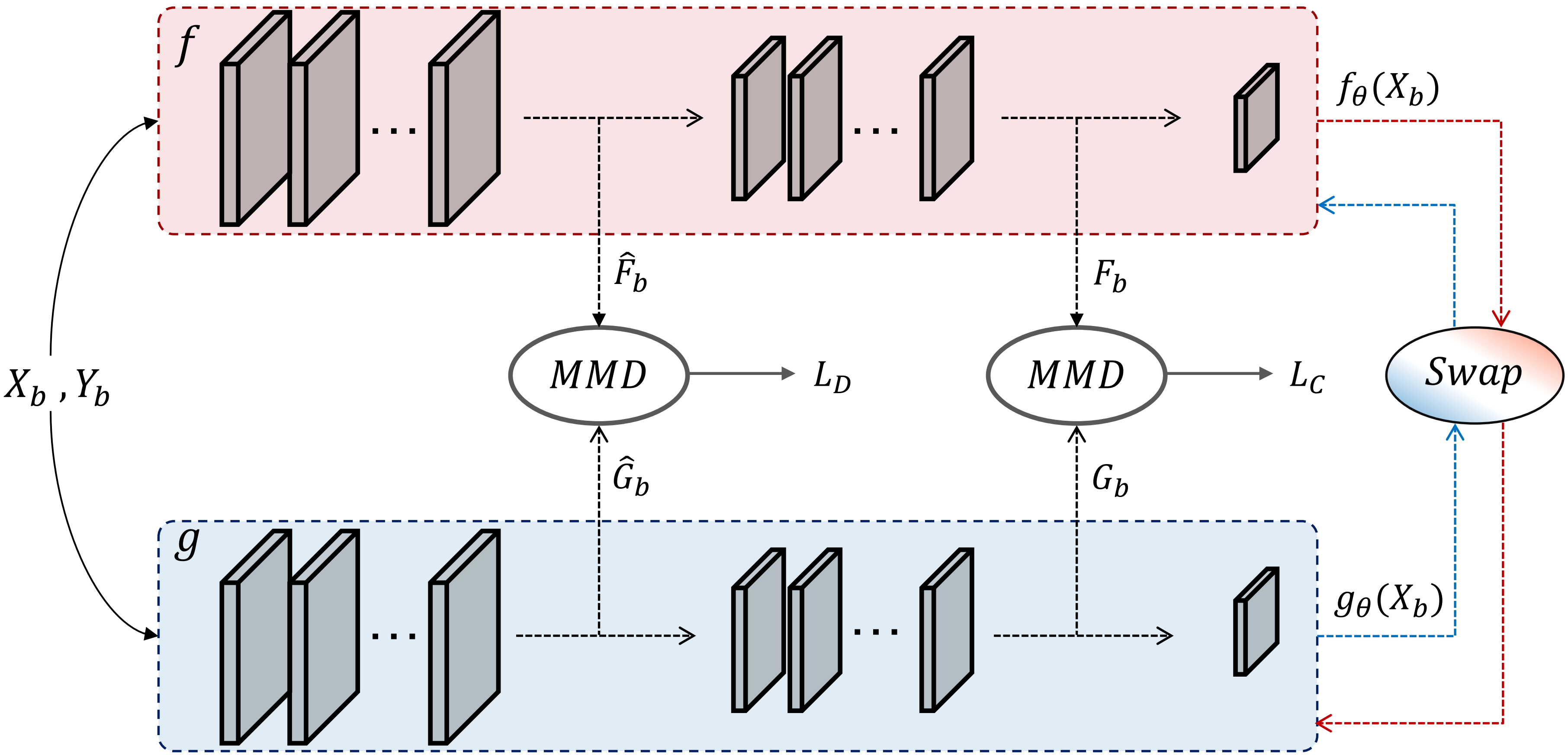
Goals:

- ✓ Automatically identify the samples with noisy labels without any prior assumptions.
- ✓ Train noise-robust classifiers with RS training images under label noise.

Solution:

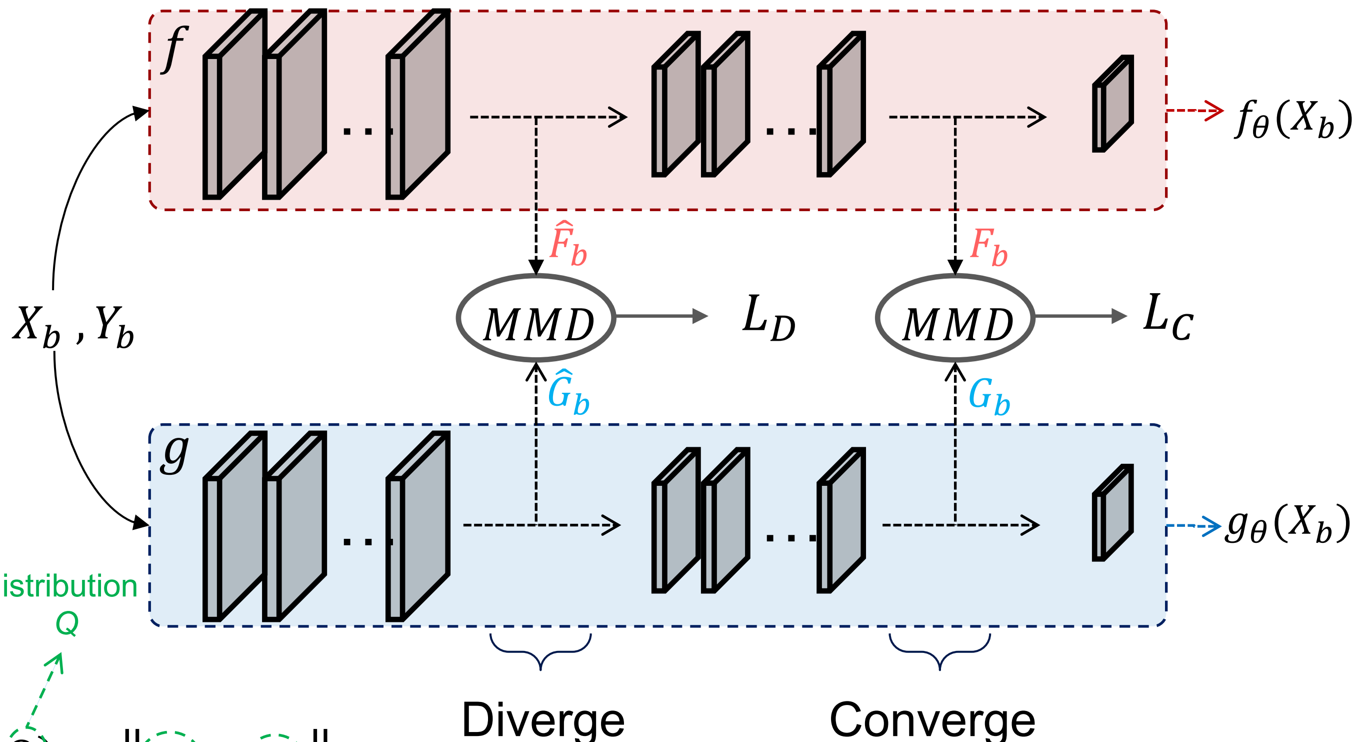
- ✓ We propose a novel consensual collaborative learning method images (**CCML**) which can:
  - **identify** the possible **noisy labels** by introducing a novel **ranking function** for identifying reliable labels,
  - estimate the label uncertainty based on the **aggregation** of two **collaborative networks**,
  - be used with **different classification** approaches to detect the potentially noisy labels assigned to the training images with multi-labels.

Proposed Consensual Collaborative Multi-Label Learning (CCML) Method



CCML: Discrepancy Module

- ✓ The discrepancy module aims at forcing the two networks to learn diverse features, while achieving consistent predictions.
- ✓ It includes: 1) Disparity loss (L_D); and 2) Consistency loss (L_C).



distribution P

distribution Q

$$\underbrace{MMD}_{\text{maximum mean discrepancy}}(\mathrm{P}, \mathrm{Q}) = \|\mu_P - \mu_Q\|_H$$

mean of P

mean of Q

CCML: Group Lasso Module

- This module has two main aims:
  - Identify potentially noisy labels in the training set by using the predictions of the two networks.
  - Identify the type of label noise by computing a sample-wise ranking loss as:

$$\mathrm{Lasso}_f(\mathrm{x}_i) = \underbrace{\alpha \sum_{\hat{c}=\mathrm{a}+1}^{m} \sqrt{\sum_{c=1}^{\overset{\text{assigned labels}}{a}} \epsilon}}_{\text{aggregated loss based on missing labels}} + \underbrace{\beta \sum_{c=b+1}^{m} \sqrt{\sum_{\hat{\mathrm{c}}=1}^{\overset{\text{unassigned labels}}{b}} \epsilon}}_{\text{aggregated loss based on wrong labels}}$$

CCML: Group Lasso Module

✓ This module has two main aims:

- Identify potentially noisy labels in the training set by using the predictions of the two networks.
- Identify the type of label noise by computing a sample-wise ranking loss as:

assigned labels

unassigned labels

$$\text{Lasso}_f(\mathrm{x}_i) = \underbrace{\alpha \sum_{\hat{c}=a+1}^{m} \sqrt{\sum_{c=1}^{a} \epsilon}}_{\text{aggregated loss based on missing labels}} + \underbrace{\beta \sum_{c=b+1}^{m} \sqrt{\sum_{\hat{c}=1}^{b} \epsilon}}_{\text{aggregated loss based on wrong labels}}$$

ranking error

$$E_{c,\hat{c}}(\mathbf{x}_i) = \max(0, 2[f_{\hat{c}}(\mathbf{x}_i) - f_c(\mathbf{x}_i)] + 1)$$

Missing label	Wrong label	$f_{\hat{c}}$	f_c	$E_{c,\hat{c}}$
✗	✗	0	1	0
✓	✗	0	0	+1
✗	✓	1	1	+1
✓	✓	1	0	+3

CCML: Group Lasso Module

- This module has two main aims:
  - Identify potentially noisy labels in the training set by using the predictions of the two networks.
  - Identify the type of label noise by computing a sample-wise ranking loss as:

$$\mathrm{Lasso}_f(\mathrm{x}_i) = \alpha \underbrace{\sum_{\hat{c}=\mathrm{a}+1}^{m} \sqrt{\sum_{c=1}^{a} \epsilon}}_{\text{aggregated loss based on missing labels}} + \beta \underbrace{\sum_{c=b+1}^{m} \sqrt{\sum_{\hat{c}=1}^{b} \epsilon}}_{\text{aggregated loss based on wrong labels}}$$

assigned labels

unassigned labels

ranking error $E_{c,\hat{c}}(\mathbf{x}_i) = \max(0, 2[f_{\hat{c}}(\mathbf{x}_i) - f_c(\mathbf{x}_i)] + 1)$

Missing label	Wrong label	$f_{\hat{c}}$	f_c	$E_{c,\hat{c}}$
✗	✗	0	1	0
✓	✗	0	0	+1
✗	✓	1	1	+1
✓	✓	1	0	+3

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
x_1	5	2	3	8	1	5	2	3	6	3
x_2	9	3	5	2	2	7	3	5	2	2
x_3	3	1	2	7	1	3	9	2	1	4
x_4	3	3	3	1	2	6	3	1	1	2
x_5	3	7	1	3	4	1	2	5	8	3

CCML: Group Lasso Module

✓ This module has two main aims:

- Identify potentially noisy labels in the training set by using the predictions of the two networks.
- Identify the type of label noise by computing a sample-wise ranking loss as:

assigned labels

unassigned labels

ranking error

$$\text{Lasso}_f(\mathrm{x}_i) = \underbrace{\alpha \sum_{\hat{c}=a+1}^{m} \sqrt{\sum_{c=1}^{a} \epsilon}}_{\text{aggregated loss based on missing labels}} + \underbrace{\beta \sum_{c=b+1}^{m} \sqrt{\sum_{\hat{c}=1}^{b} \epsilon}}_{\text{aggregated loss based on wrong labels}}$$

$$E_{c,\hat{c}}(\mathbf{x}_i) = \max(0, 2[f_{\hat{c}}(\mathbf{x}_i) - f_c(\mathbf{x}_i)] + 1)$$

Missing label	Wrong label	$f_{\hat{c}}$	f_c	$E_{c,\hat{c}}$
✗	✗	0	1	0
✓	✗	0	0	+1
✗	✓	1	1	+1
✓	✓	1	0	+3

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
x_1	5	2	3	8	1	5	2	3	6	3
x_2	9	3	5	2	2	7	3	5	2	2
x_3	3	1	2	7	1	3	9	2	1	4
x_4	3	3	3	1	2	6	3	1	1	2
x_5	3	7	1	3	4	1	2	5	8	3

CCML: Group Lasso Module

- This module has two main aims:
  - Identify potentially noisy labels in the training set by using the predictions of the two networks.
  - Identify the type of label noise by computing a sample-wise ranking loss as:

$$\mathrm{Lasso}_f(\mathrm{x}_i) = \underbrace{\alpha \sum_{\hat{c}=a+1}^{m} \sqrt{\sum_{c=1}^{a} \epsilon}}_{\text{aggregated loss based on missing labels}} + \underbrace{\beta \sum_{c=b+1}^{m} \sqrt{\sum_{\hat{c}=1}^{b} \epsilon}}_{\text{aggregated loss based on wrong labels}}$$

assigned labels

unassigned labels

ranking error $E_{c,\hat{c}}(\mathbf{x}_i) = \max(0, 2[f_{\hat{c}}(\mathbf{x}_i) - f_c(\mathbf{x}_i)] + 1)$

Missing label	Wrong label	$f_{\hat{c}}$	f_c	$E_{c,\hat{c}}$
✗	✗	0	1	0
✅	✗	0	0	+1
✗	✅	1	1	+1
✅	✅	1	0	+3

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	l_{10}
x_1	5	2	3	8	1	5	2	3	6	3
x_2	9	3	5	2	2	7	3	5	2	2
x_3	3	1	2	7	1	3	9	2	1	4
x_4	3	3	3	1	2	6	3	1	1	2
x_5	3	7	1	3	4	1	2	5	8	3

$$N = \{4, 1, 7, 6, 9\}$$

potential noisy labels of each samples

CCML: Flipping Module

✓ The flipping module aims at flipping the identified noisy labels and includes:

- Noisy class selector (NCS) receives the ranking loss from two networks and identifies the samples with higher uncertainty;
- Noisy class flipper (NCF) selects the labels with the largest ranking loss to apply the flipping.

From group lasso module
f

Ranking loss	***N***	label
0.2	**4**	**1**
0.7	**1**	**0**
0.7	**7**	**0**
0.3	**6**	**0**
0.9	**9**	**1**

From group lasso module
g

Ranking loss	***N***	label
0.1	**2**	**0**
0.7	**1**	**0**
0.9	**7**	**0**
0.2	**11**	**0**
0.6	**9**	**1**

NCS

Ranking loss	***N***	label
0.14	**1**	**0**
0.16	**7**	**0**
0.15	**9**	**1**

NCF

Flipped label
1
0

CCML: Flipping Module

✓ The flipping module aims at flipping the identified noisy labels and includes:

- Noisy class selector (NCS) receives the ranking loss from two networks and identifies the samples with higher uncertainty;
- Noisy class flipper (NCF) selects the labels with the largest ranking loss to apply the flipping.

From group lasso module f

Ranking loss	*N*	label
0.2	4	1
0.7	1	0
0.7	7	0
0.3	6	0
0.9	9	1

From group lasso module g

Ranking loss	*N*	label
0.1	2	0
0.7	1	0
0.9	7	0
0.2	11	0
0.6	9	1

NCS

Ranking loss	*N*	label
0.14	1	0
0.16	7	0
0.15	9	1

NCF

Flipped label
1
0

CCML: Flipping Module

- ✓ The flipping module aims at flipping the identified noisy labels and includes:
  - Noisy class selector (NCS) receives the ranking loss from two networks and identifies the samples with higher uncertainty;
  - Noisy class flipper (NCF) selects the labels with the largest ranking loss to apply the flipping.

From group lasso module
f

Ranking loss	*N*	label
0.2	4	1
0.7	1	0
0.7	7	0
0.3	6	0
0.9	9	1

From group lasso module
g

Ranking loss	*N*	label
0.1	2	0
0.7	1	0
0.9	7	0
0.2	11	0
0.6	9	1

NCS

Ranking loss	*N*	label
0.14	1	0
0.16	7	0
0.15	9	1

NCF

Flipped label
1
0

RSiM

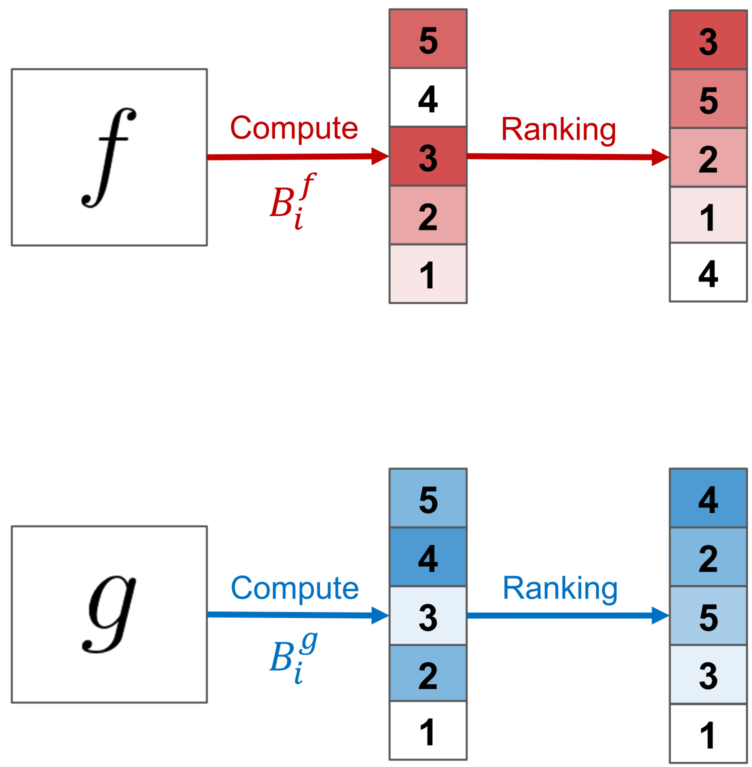
CCML: Swap Module

- ✓ This module:
  - aims at exchanging the ranking information between the networks,
  - inserted between the two collaborative networks,
  - takes the Binary Cross Entropy (BCE) and ranking losses into consideration to eliminate the detected noisy samples from back-propagation.

BCE loss

Sample from mini-batch

$$B_i^f = L_f(\mathbf{x}_i) + \gamma Lasso_f(\mathbf{x}_i)$$



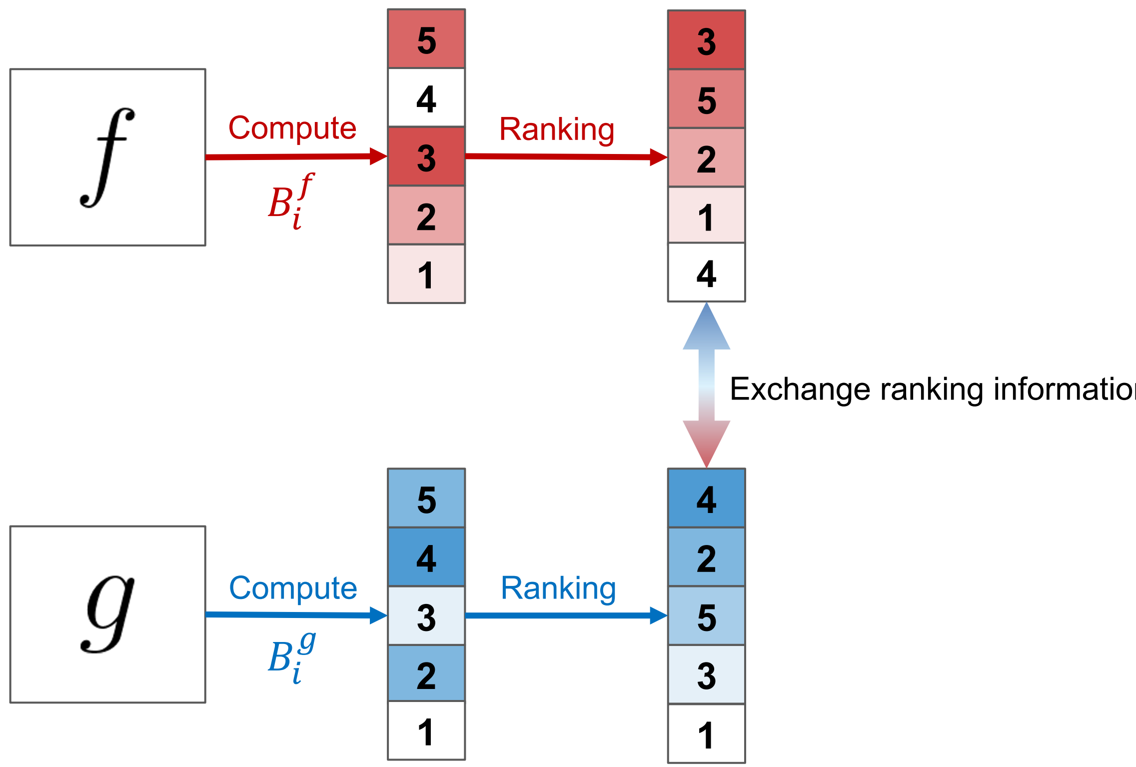
CCML: Swap Module

- This module:
  - aims at exchanging the ranking information between the networks,
  - inserted between the two collaborative networks,
  - takes the Binary Cross Entropy (BCE) and ranking losses into consideration to eliminate the detected noisy samples from back-propagation.

BCE loss

Sample from mini-batch

$$B_i^f = L_f(\mathbf{x}_i) + \gamma Lasso_f(\mathbf{x}_i)$$



CCML: Swap Module

- This module:
  - aims at exchanging the ranking information between the networks,
  - inserted between the two collaborative networks,
  - takes the Binary Cross Entropy (BCE) and ranking losses into consideration to eliminate the detected noisy samples from back-propagation.

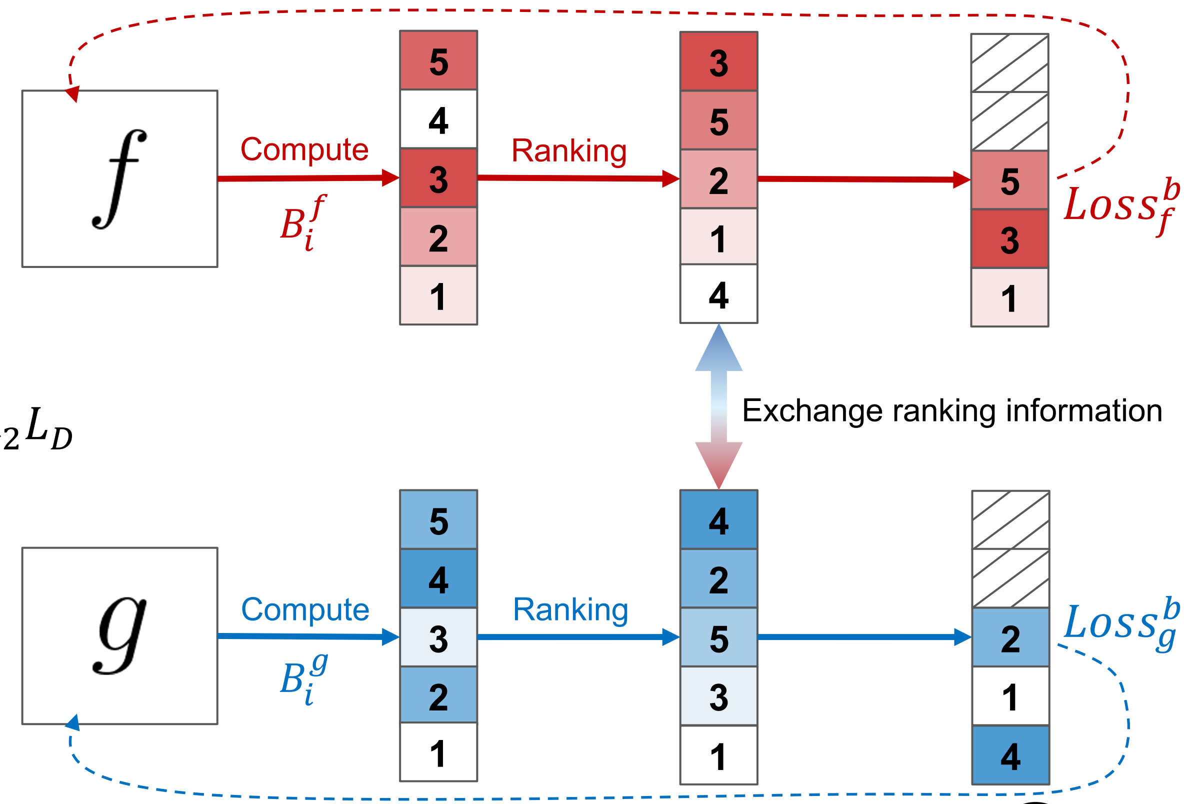
BCE loss

Sample from mini-batch

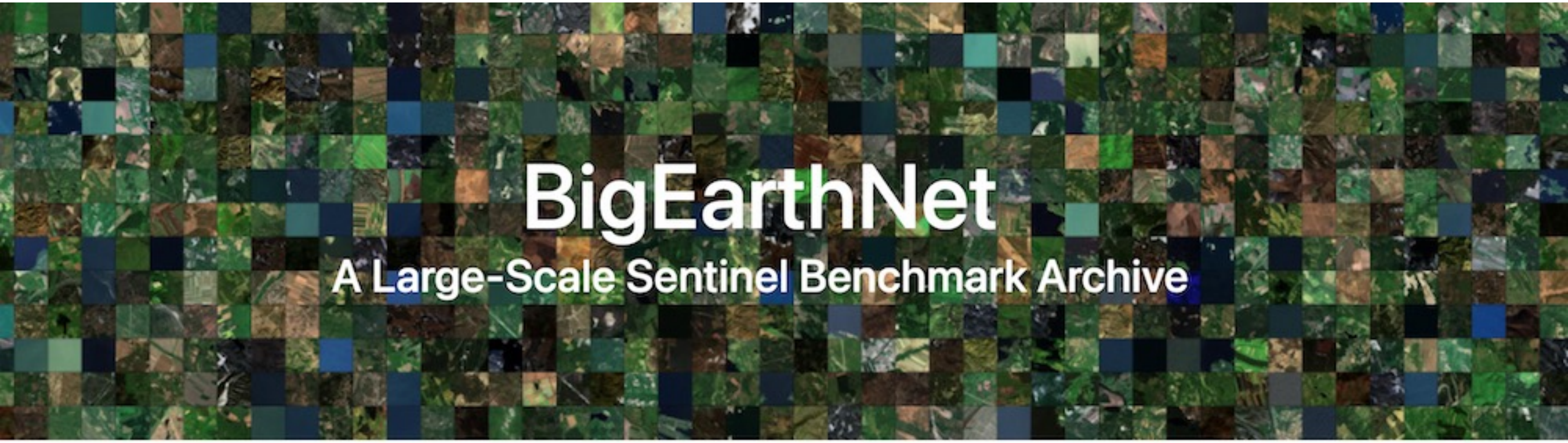
$$B_i^f = L_f(\mathbf{x}_i) + \gamma Lasso_f(\mathbf{x}_i)$$

$$Loss_f^b = \frac{1}{R}\sum_{i=1}^{R} L_f(\mathbf{x}_i) + \lambda_1 L_c + \lambda_2 L_D$$

BCE loss for R samples associated with low loss values identified by *g*



Dataset Description



- ✓ Experiments have been carried out on the Ireland subset of the BigEarthNet [1] benchmark archive, consisting of 15,894 Sentinel-2 images.
- ✓ Each image was annotated by multiple land-cover classes provided by 2018 CLC inventory.
- ✓ We used the land-cover class nomenclature proposed in [2].

[1] G. Sumbul, M. Charfuelan, B. Demir, V. Markl, "BIGEARTHNET: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding", IEEE International Conference on Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019.
[2] G. Sumbul, A. d. Wall, T. Kreuziger, F. Marcelino, H. Costa, P. Benevides, M. Caetano, B. Demir, V. Markl, "BigEarthNet-MM: A Large Scale Multi-Modal Multi-Label Benchmark Archive for Remote Sensing Image Classification and Retrieval", IEEE Geoscience and Remote Sensing Magazine, 2021, doi: 10.1109/MGRS.2021.3089174.

Experimental Setup

- ✓ Two architectures **ResNet** [3] and **DenseNet** [4] were used as **baselines** for comparison.
- ✓ The same architectures were considered as **backbones** for our **CCML**.
- ✓ Within the swap module of the CCML, we used **75%** of the samples associated with small loss values at each iteration for **swapping**.
- ✓ The **flipping** module was activated after reaching **90%** of epochs.
- ✓ Noise injection is applied by random selection of *n*% of samples from each mini-batch, and flipping randomly *n*% of the labels from the selected samples.
- ✓ The value of *n* was varied from 20 to 50 with a step size increment of 10.

Noise injection with the rate of 50%

0 synthetically injected missing labels

1 synthetically injected wrong labels

y_1	1	0	**0**	**0**	0	0	**1**	**1**	0	**0**
y_2	0	0	0	1	0	1	0	0	1	0
y_3	**1**	**1**	0	0	**1**	0	**0**	1	**1**	0
y_4	1	**1**	**0**	**1**	0	**1**	**0**	0	1	0
y_5	0	1	0	0	1	1	0	0	0	1
y_6	0	0	0	1	0	1	0	0	1	0

[3] . He, X. Zhang, S. Ren, and J. Sun, "Deep residual learningfor image recognition," IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, 2016.
[4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2261–2269, 2017.

Experimental Results: ResNet

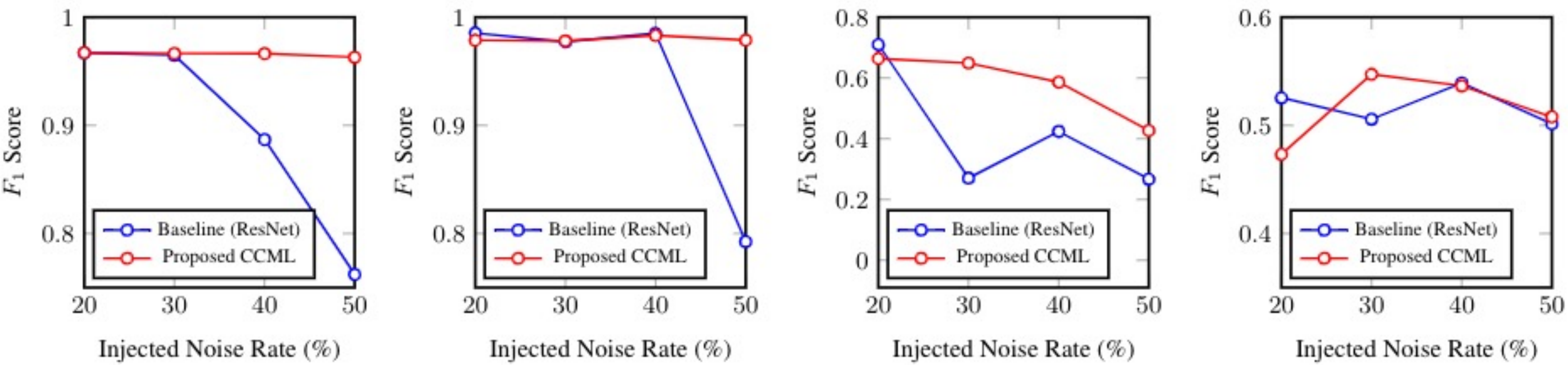
Injected Noise Rate	Precision (%)		Recall (%)		F_1 (%)	
	Baseline (ResNet)	Proposed CCML	Baseline (ResNet)	Proposed CCML	Baseline (ResNet)	Proposed CCML
20%	87.8	**90.2**	**68.7**	**68.7**	77.1	**78**
30%	84	**88.2**	67.2	**68.9**	74.7	**77.4**
40%	76.4	**88.4**	65.1	**69.3**	70.3	**77.7**
50%	62.5	**87.5**	57.6	**62.1**	60	**72.6**

Experimental Results: DenseNet

Injected Noise Rate	Precision (%)		Recall (%)		F_1 (%)	
	Baseline (DenseNet)	Proposed CCML	Baseline (DenseNet)	Proposed CCML	Baseline (DenseNet)	Proposed CCML
20%	89.2	**89.6**	68.4	**77.4**	77.1	**78.1**
30%	91.8	**92**	66.2	**66.7**	76.9	**77.3**
40%	85.6	**89**	68.5	**68.7**	76.1	**77.5**
50%	55.3	**85.1**	62.5	**66.4**	58.7	**74.6**

Experimental Results: Class-based Performance

Pastures | Marine waters | Inland wetlands | Moors, heathland and sclerophyllous vegetation



class represented by a **high** number of training images

class represented by a **small** number of training images

- ✓ In some classes, the baseline and the CCML performances are comparable over lower rates of label noise, but CCML maintains relatively **high performance** under **high noise rates**.
- ✓ CCML is **stable** under high noise rates.

Conclusion

- ✓ A novel Consensual Collaborative Multi-Label Learning (**CCML**) has been presented to overcome adverse effects of **multi-label noise** for the classification of **RS** images.

- ✓ The proposed CCML is promising since it:

  - is able to automatically identify two different types of multi-label noise (i.e., missing and wrong class label annotations) without making any **prior assumption**.

  - achieves **high accuracy** under a high (synthetically added) multi-label noise rates.

  - is **architecture-independent**, and thus can be used within different network architectures.

  - is applicable in a range of RS applications (e.g., large scale image retrieval, auto-labeling tools, etc.)

- ✓ As a future development, we plan to extend the proposed CCML with an **adaptive ranking loss function** to adjust the amount of sample removal.

Technische Universität Berlin

RSiM

Our code is available at:

https://git.tu-berlin.de/rsim/CCML

Follow us in twitter: @BigEarthERC

A. K. Aksoy , M. Ravanbakhsh , T. Kreuziger , B. Demir