




# Filtering Specialized Change in a Few-Shot Setting

Martin Hermann , Sudipan Saha , *Member, IEEE*, and Xiao Xiang Zhu , *Fellow, IEEE*

**Abstract**—The aim of change detection in remote sensing usually is not to find all differences between the observations, but rather only specific types of change, such as urban development, deforestation, or even more specialized categories like roadwork. However, often there are no large public datasets available for very fine-grained tasks, and to collect the amount of training data needed for most supervised learning methods is very costly and often prohibitive. For this reason, we formulate the problem of *few-shot filtering*, where we are provided with a relatively large change detection dataset and, at test time, a few instances of one particular change type that we try to “filter out” of the learned changes. For example, we might train on data of general urban change, and, given some samples of building construction, aim to only predict instances of these on the test set, all without any explicit labels for buildings in the training data. We further investigate a fine-tuning approach to this problem and assess its performance on a public dataset that we adapt to be used in this novel setting.

**Index Terms**—Change detection, deep learning, few-shot filtering, few-shot learning.

## I. INTRODUCTION

CHANGE detection, that is, segmenting a pair of images of the same region but taken at two different points in time into changed and unchanged pixels, is a well-known task in remote sensing, with many applications in disaster assessment, urban planning, forest monitoring, and other remote sensing domains [1]. Usually, for these applications we are not interested in every change that occurred between the images, but limit our attention to certain categories, such as building construction or destruction [2], deforestation [3] or flooding [4], and occasionally even finer subcategories like road construction [5], mining activities [6], or ship movement [7].

For this reason, *supervised learning*, where we can exactly specify our interests via annotated samples, is a natural choice for these specialized change detection tasks. However, the downside of this approach is that these methods usually require large amounts of training data, which is expensive or even prohibitive

Manuscript received 11 October 2022; revised 8 December 2022; accepted 14 December 2022. Date of publication 9 January 2023; date of current version 12 January 2023. This work was supported in part by the Munich Aerospace e.V. scholarship as part of the research group IMonitor, in part by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI lab “AI4EO – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond” under Grant 01DD20001, and in part by the German Federal Ministry for Economic Affairs and Climate Action in the framework of the “National Center of Excellence ML4Earth” under Grant 50EE2201C. (*Corresponding author: Xiao Xiang Zhu.*)

The authors are with the Chair of Data Science in Earth Observation, Technical University of Munich, 85521 Ottobrunn, Germany (e-mail: martin.hermann@tum.de; sudipan.saha@tum.de; xiaoxiang.zhu.ieee@gmail.com).

Code will be available at <https://gitlab.lrz.de/ai4eo/cd/-/blob/main/fewShotFilteringCd>.

Digital Object Identifier 10.1109/JSTARS.2022.3231915

to produce. Even though there are a number of large datasets publicly available, these often are annotated with rather general categories, such as *urban change* [8].

If we want to use these existing resources for more specialized tasks, such as the ones mentioned above, the resulting models detect a lot of unwanted changes in addition to those that are relevant, so filtering out the important information becomes a key step. In principle, it is possible to do so with a lot of additional data, or by manually adapting the training labels (one recent example for this can be found in the work of Li et al. [9]). However, this again requires a large amount of resources, the lack of which often is the reason for using a preexisting dataset in the first place. Hence, a solution that adapts to a specialized usecase with only a handful of annotated samples of this particular type of changes would be very desirable.

Pushing this idea even further, research often is an iterative process, and in many situations we might not have a clear definition of the change we are interested in from the beginning. For example, when investigating deforestation, we could realize after some time that in fact the most relevant type for our scenario is caused by wildfires, and we want to ignore, e.g., logging. On the other hand, we might decide to focus on human influences, and now look for newly built infrastructure close to the rain forest. To enable a flexible workflow and avoid long interruptions caused by retraining the network from scratch, it is useful to allow the specification of the change of interest only *after* training on the full dataset.

Therefore, we propose an approach to detect specialized changes that works top-down: First, we learn to classify a broader type of change in a binary classification task, for which ample training data are available, and then, try to *filter out* one particular subcategory via only a few examples, thereby entering the realm of few-shot learning [10], [11]. As few-shot learning deals with novel classes that the machine learning model has never seen before, whereas we in contrast try to specialize and split known classes, we propose the term *few-shot filtering* for this task, that we will describe in detail in Section III. To provide some background, we will shortly present few-shot learning, data efficient approaches to change detection and other related work in Section II, before we describe the methods we investigate in Section IV, detail our experimental setup and dataset in Section V and finally present the results in Section VI and discuss them in Section VII. Finally, Section VIII concludes this article.

In particular, our contributions are the following.

- 1) We formulate the problem of few-shot filtering for change detection, which differs from standard few-shot learning in that the query classes are not disjoint from the base classes during training. Instead, we focus on refinements of previously seen change.

- 2) We investigate a fine-tuning approach to this problem, together with two simple baselines, and compare their performance on several different few-shot tasks.
- 3) To evaluate these methods, we suggest a way to suitably adapt a semantic change detection dataset to this new setting of few-shot filtering and discuss its advantages and limitations.
- 4) We conduct a hyperparameter study to gain some insight into the effects on different types of specialized change, which is important as the conventional approach of optimizing on a validation set is not possible in the data scarce setting.

## II. RELATED WORK

In the following, we will give a brief overview over the different areas of research we touch upon and highlight important related work. The four main areas are few-shot learning, specialization and subcategorization, semantic change detection, and methods to deal with data scarcity.

### A. Few-Shot Learning and Few-Shot Segmentation

Few-shot learning is a very active area in machine learning in general and computer vision in particular. Similar to how a human can learn novel objects from only a few instances, it is interested in adapting a network to previously unseen classes from a small number of training examples (which are known as “shots”) [10].

There are several different popular approaches to this task [11], including meta-learning techniques, such as *MAML* [12] and metric-based ones, most notably *matching*- [13] and *prototypical networks* [14]. These methods generally use *episodic learning*. They simulate the few-shot setting already during training, optimizing the performance on a *query set* given a small amount of labeled data (the *support set*).

However, there is evidence that episodic learning might not be necessary for good performance [10], [15] and several methods only use support and query sets after training on the full dataset. Examples for this are the works of Gidaris and Komodakis [16] and Qi et al. [17]. There, the last layer classifies the embeddings produced by the rest of the network via cosine similarity and, during inference, the embeddings of the support set serve as a prototype for the new class in the query set. Closely related approaches are also investigated by Chen et al. [10] and Dhillon et al. [18] as competitive baselines, and like them, we also use fine-tuning on the support set as our main approach.

In general, most research focuses on few-shot classification; however, there also exists a considerable amount of literature on few-shot segmentation, both of images [19], [20], [21], [22] and of videos [23], [24]. This task is more challenging, but also closer to our problem of change detection, where labels have to be assigned to individual pixels instead of the whole image.

In summary, our setting is closely related to existing work on few-shot learning and segmentation, but we do not propose to solve it via the very common episodic training and aim for a simpler fine-tuning-based approach instead.

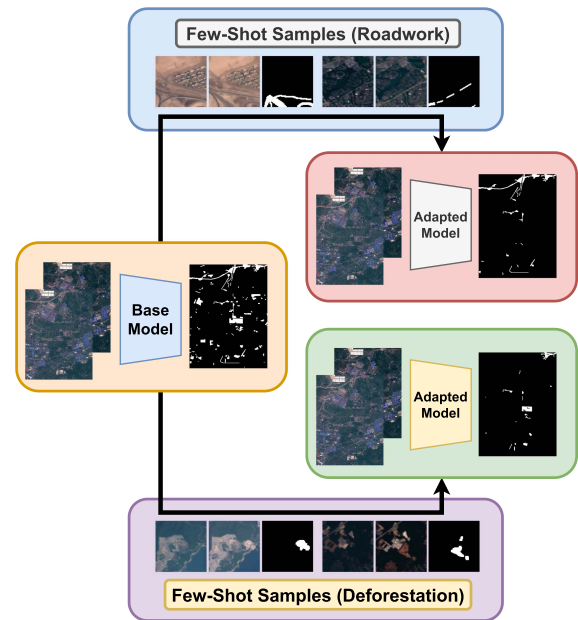


Fig. 1. Few-shot filtering task: Starting from a base model, trained on a broad set of changes, such as *urban development*, we want to quickly adapt to new, specialized categories, such as *roadwork* or *deforestation*. Examples are from the *OSCD* dataset [8], the filtered annotations are own work.

### B. Specialization and Subcategorization in Computer Vision

Hierarchical information, such as subcategories and general “coarse-to-fine” relations can be a valuable resource in computer vision for a range of different tasks. Learning on fine labels induces features relevant for coarse classification, as well as vice versa [25]. One subclass specific task in semantic segmentation is, e.g., hierarchical segmentation [26] [27], where multiple levels of the hierarchy are predicted simultaneously.

Going from a broad class to a specialized subcategory in transfer learning has been explored, e.g., for object category detection [28], and recent works formalized this in a few-shot manner [29], [30]. Bukchin et al. [29] call this setting *coarse-to-fine few-shot*, and this is essentially the classification variant of few-shot filtering. However, they do assume a set of mutually exclusive fine-grained classes, whereas we allow pixels to be relevant for different possible change types (cf. Fig. 1, where some pixels are relevant both for deforestation *and* roadwork). Ni et al. [31] explore the coarse-to-fine few-shot setting under the name *cross-granularity few-shot* with a medical application in mind, Xiang et al. [32] develop an incremental variant and, concurrently to the present work, Gong et al. [33] investigate taxonomy adaptive cross-domain semantic segmentation (e.g., also incorporating subclasses of known classes) also with only a few labeled shots. To the best of our knowledge, however, this is the first work exploring similar ideas specifically adapted to the context of change detection.

### C. Semantic and Multiclass Change Detection

Whereas binary change detection is just interested in whether some change occurs in the given time frame or not, the goal of *semantic* change detection (also known as *multiclass* change detection) [34], [35] is to further break this change down into

several classes. Often, these are identified by the land cover categories of both images, and the task is then to detect change as well as label it, e.g., as “*from low vegetation to building.*” As Yang et al. [36] point out, there are also changes that do not affect the land cover categories (such as the replacement of one building by another), which are ignored without additional change/nonchange information. Other works also consider much more fine-grained change types, such as the construction of *residential or industrial buildings or mega projects* [37]. Unsupervised approaches to this task can also work without defining change classes in advance and discover different types of change, for example via *deep change vector analysis* [38].

While we are also concerned with different change categories, unlike supervised semantic change detection methods we do not consider them fixed *a priori*, and instead want to allow a flexible specification after training. Similar to the unsupervised methods, we also do not have any labels on the change categories, but we *do* have access to binary change information. Nevertheless, unsupervised semantic change detection is the task closest related to our problem of few-shot filtering.

#### D. Data Scarcity in Change Detection

The problem of limited training data in change detection is not new. Commonly, this is approached by *semi-* [39], *self-* [40], or *unsupervised* methods [38], where no or only a small amount of annotated training data is needed. In contrast to few-shot learning, the focus lies on the information contained in many unlabeled images and not so much on quick adaptability to a few labeled ones.

Additionally, other techniques, such as *transfer* [41] or *active learning* [42] are employed to deal with data scarcity. Also, how a small amount of data can affect the performance of change detection algorithms has been investigated by Saha et al. [43]. Recently, there is also work exploring few-shot learning approaches to change detection [44], [45], showing that this is indeed a promising approach, which we hope to further expand by introducing the few-shot filtering setting. Tang et al. [46] use methods from few-shot segmentation—prototypes and masked average pooling—but apply them to a standard binary setting where a larger amount of data is available.

Finally, we also want to highlight recent work by Lenczner et al. [47], who add new classes to the segmentation of remote sensing images in a continual learning setting, resulting in what can loosely be described as the inverse of our task: although in the context of semantic segmentation instead of change detection.

All in all, while there have been various attempts to deal with data scarcity, our proposed setting is novel in its flexibility and differs from existing tasks that are designed with alternative applications in mind. We will now describe it in detail in the following section.

### III. FEW-SHOT FILTERING AS A SETTING IN CHANGE DETECTION

#### A. Binary Change Detection and Few-Shot Filtering

The aim of few-shot filtering for change detection is to train a model on a dataset that is annotated with a general category of

change, and then adapt it to a more fine-grained task with only a few new examples (the support set). This problem is illustrated in Fig. 1, and we will now formalize this setting.

The (binary) change detection task is concerned with a pair of images

$$\mathbf{I}_1, \mathbf{I}_2 \in \mathbb{R}^{H \times W \times C}$$

where  $H \times W$  are the spatial dimensions and  $C$  denotes the number of channels (which might be higher than the usual 3 in standard computer vision and can also include, e.g., near infrared bands). It is assumed that  $\mathbf{I}_1$  and  $\mathbf{I}_2$  depict the same region, but are taken at different points in time, often multiple months, or (such as, e.g., for applications in urban development) years apart. Our aim then is to derive a *change map*

$$C_{\mathbf{I}_1, \mathbf{I}_2} \in \{0, 1\}^{H \times W}$$

i.e., a segmentation of the input images into pixels that have changed in some meaningful way between  $\mathbf{I}_1$  and  $\mathbf{I}_2$  (denoted by a value of 1) and those that remain unchanged.

Note that this notation conceals the nature of the change: while for certain applications, a pixel belonging to a newly constructed building might be considered relevant, in other cases (such as for deforestation or agricultural domains) we might not be interested in this particular instance. Therefore, we additionally index the change map by a change type  $\mathcal{T}$  and try to produce  $C_{\mathbf{I}_1, \mathbf{I}_2, \mathcal{T}}$ . This is different from semantic change detection, where we are given  $K > 1$  different categories of change and the aim is to find

$$C_{\mathbf{I}_1, \mathbf{I}_2} \in \{0, 1, \dots, K\}^{H \times W}.$$

In our setting, we still are interested only in binary classification, albeit restricted to one particular category of changes  $\mathcal{T}$ .

In supervised learning, we assume a train set

$$D_{\mathcal{T}}^{\text{train}} = \{(\mathbf{I}_{1i}, \mathbf{I}_{2i}, C_{\mathbf{I}_{1i}, \mathbf{I}_{2i}, \mathcal{T}})\}_{i=1}^N$$

of image pairs and their corresponding change map. This train set is usually limited to one change type  $\mathcal{T}$ , such as urban development or the impact of natural disasters. A standard change detection task would now be to evaluate a model trained on  $D_{\mathcal{T}}^{\text{train}}$  on some test set  $D_{\mathcal{T}}^{\text{test}}$  that is sufficiently similar to the train set. For the few-shot filtering problem however, we assume an additional support set  $D_{\mathcal{T}'}^{\text{supp}}$  that is small (five  $256 \times 256$  patches in our experiments, but in other scenarios, a moderate amount of data—that can still be labeled with low cost—might also be adequate) and contains examples of a change type  $\mathcal{T}' \subset \mathcal{T}$ , meaning

$$C_{\mathbf{I}_1, \mathbf{I}_2, \mathcal{T}'} = 1 \Rightarrow C_{\mathbf{I}_1, \mathbf{I}_2, \mathcal{T}} = 1.$$

This is, e.g., the case for  $\mathcal{T}$  denoting general urban change, and  $\mathcal{T}'$  then signifying building demolition. The evaluation then happens on a query set  $D_{\mathcal{T}'}^{\text{query}}$ , where we are now only interested in finding change of the new, restricted type  $\mathcal{T}'$ .

#### B. Relation to Other Settings

One way to compare this definition to the standard few-shot learning setting is to look at  $\mathcal{T}$  as the base class, and  $\mathcal{T}'$  as the novel class. However, instead of the empty intersection between base and novel classes that is normally assumed, we in contrast

are interested in a subset relation. We could also consider our formulation as a form of weakly supervised learning [48] with the broad annotations of the train set acting as weak labels for the more specialized labels of the support and query sets. However, this view does not adequately highlight the few-shot nature of the task, where the support set in practice is only available after the training and consists just of a small amount of change instances. Another interpretation is to see it as a form of transfer learning [49], where we have a very strong relation between source and target task and we already know quite well about this connection.

One might also think that semantic change detection should solve the same problem: by breaking the binary change class into multiple subcategories, we can then simply select the one we are interested in. However, while the amount of such datasets is growing, a big part of existing resources is annotated with binary labels. More importantly, this only shifts the problem: the amount of different change types in semantic change detection problems is limited and *set in advance*, therefore in realistic applications, they might still not fit our needs exactly. In fact, we can even think of both approaches going hand in hand: using a semantic change detection dataset, we first choose the category that fits best our needs (such as *buildings*) and then use a few examples to further filter out exactly what we want (e.g., *high rise construction*). Also, categories in semantic change detection are usually assumed to be mutually exclusive, while this does not have to be the case for filtered specializations. Looking at Fig. 1, we can see that there are regions where *deforestation* happened in order to enable *roadwork*, so the changed pixels are relevant for both adapted models. Achieving this with semantic categories would need much more fine-grained categories than are currently common in semantic change detection or hierarchical or nonexclusive labels.

To avoid confusion, we should note that we use a semantic change detection dataset in this work to simulate a few-shot scenario, as described in Section V-A. The aim here is to use the high quality annotations of the dataset as a precise ground truth for the experiments in this work. In that section, we also shortly discuss the limitations of using the semantic categories as few-shot tasks, and the points raised there (independent pixels, no spatial structure, and limited granularity) also strengthen the argument above that semantic change detection alone cannot solve the problem that few-shot filtering addresses. The creation of a benchmark dataset designed specifically for few-shot filtering is a logical next step for further research.

### C. Discussion of the Setting

Few-shot filtering is a relevant setting in cases where we have access to an annotated binary dataset, but the annotations do not exactly fit our needs, as we are only interested in a particular subset of the change that is marked. One such scenario is the use of public benchmark datasets, where we often find, e.g., *urban change*. If we are investigating *roadwork*, it will be difficult to focus on the relevant instances, as most of the detected change probably consists of constructed buildings, which distracts from the events we want to study. Here, the few-shot filtering

framework applies as we want to get to the relevant information with only a limited amount of additional labeling effort.

A big advantage of the setting is the flexibility that is enabled by the separation of base training and adaption to the few-shot samples. Typically the former will take much longer than the latter, so we can perform the adaption quickly and repeatedly, allowing for an interactive and adaptive workflow. One such scenario was described in the introduction.

The clear limitation of the approach is that by design, the specialized change type is expected to be fully part of the original annotations used for base training, restricting the possible specialization depending on what data are available. We could extend the original formulation to allow for other relations between  $\mathcal{T}$  and  $\mathcal{T}'$ , such as just requiring a nonempty intersection, moving closer to the standard few-shot setting. However, we consider these tasks to be complementary: few-shot *learning* helps us *find new change*, few-shot *filtering splits known change* further. We will mainly focus on the latter in this work,<sup>1</sup> but in practice a combination of both tasks should be very beneficial.

## IV. METHODS

After defining the few-shot filtering setting, we will now investigate several methods to tackle it, that also give some insight on how the combination of different resources can be beneficial. In addition, we will discuss the practical issue of hyperparameters in this setting.

### A. Learning From a Single Data Source

*Base Training Only:* The most straightforward approach is to just use a standard change detection model that has been trained on the base training set and apply it on the query set without any kind of adaptations and without using the support set at all. Of course, we do not expect this to be a competitive approach, as the very idea of the few-shot filtering task is to specialize, and to limit the full output to only the interesting classes.

However, including these results in our experiments, we can gain an understanding of how much the additional information adds. In addition, it allows us to gauge the difficulty of the individual few-shot tasks and how much they vary. In general, we expect this approach to have a rather high rate of false positives, as we do not filter out the change that was relevant during training, but is not for the few-shot tasks.

*Support Set Only:* As the exact opposite of the previous method, we can also ignore the base training data (that has all changes annotated, not just the specialized type), and just train a change detection model on the (very small) support set. In general, this will likely overfit very heavily, and for better comparability, we are also using the same architecture as for the other tasks, which will only exacerbate this problem. A smaller model that is more tailored to this situation might achieve better results if we intend to use this method in practice, but for our experiments, this setup is well suited.

<sup>1</sup>We can see in the experiments that there are cases where we detect *more* of the desired change after fine-tuning. However, this is just a correction of mistakes of the base model: it should already find these, as they were part of the ground truth annotations, in contrast to completely new change types.

We will call these approaches *Baseline A* for the base training and *Baseline B* for using only the support set in our experiments. Together, both baselines can give an idea of how much information is already contained in the large, but general base training set (Baseline A), as well as in the small, but specific support set (Baseline B), and how much we can gain by combining both, i.e., by fully making use of the few-shot setting.

### B. Fine-Tuning

Fine-tuning amounts to combining both of the above baselines to learn a specialized model. For this, we first train the model on the base training set to learn a general notion of change, and then run additional epochs on the support set, to filter out the specialized type that we are interested in. Note that as the support set is small, this fine-tuning is comparably short, and we can use the same model that has been trained once on the full dataset to adapt to different few-shot tasks separately and quickly.

When learning on the support set, we retrain all layers; however, another common approach would be to freeze all but the last layer (which makes the final decision about whether an individual pixel has changed). This is known as *linear probing*, and some research suggests that it can for example be beneficial for out-of-distribution cases [50]. Based on initial experiments, we decided against it, but further research into the best training strategy during this phase might prove to be valuable.

We decide to investigate a fine-tuning approach instead of more sophisticated episodic techniques for several reasons: for one, it makes sense to first establish a basic performance level for this new setting, and to use a conceptually simple method to do so. Whether and what other methods might improve on this, and in particular whether episodic learning can be of advantage here is only the next step: in particular as the question how much can be gained from episodic learning is under discussion in the general few-shot literature [15]. Also, from a practical point of view, we assume that we use some existing binary data source, so episodic training would require additional filtered data, which defeats the purpose to have a data efficient method available. We could avoid this with a semantic change detection dataset, the same way that we also do this for testing in this article, but this limits the usability to domains where such data exist. Therefore, we propose to tackle our problem with a fine-tuning approach.

### C. Effect of Hyperparameters and Lack of Validation Data

Part of the training strategy during fine-tuning is the decision on a set of *hyperparameters*. The importance of these for the performance of a deep learning model is well known, and that optimizing them also for fine-tuning can be vital has been shown by Li et al. [51]. However, in our setting we only have a few annotated images for a new task, so we cannot simply use a validation set to determine the best values for these parameters. In addition, we also cannot expect that there is one set of parameters that will work well across all possible few-shot tasks, as they may differ in key aspects, such as how frequent the type of change is. The problem of no available validation set is also discussed, e.g., by Gulrajani and Lopez-Paz [52], however, in the context of domain generalization. Suggested solutions in literature include data augmentation on the support set [53].

It is desirable to gain some understanding on how different choices affect different scenarios, so that in practice we can, e.g., choose a suitable setting based on some heuristics, (similar to *BiT-HyperRule* [54]), such as how frequent we expect the specialized change to be or how difficult the task is. We will investigate the effects of parameters by varying the parameters on the actual few-shot tasks. In the terms introduced by [52], this would amount to using a test-domain validation set, which they do not view as suited for benchmarking. However, our goal is not to decide on a particular set of parameters, but rather to investigate their effect across different tasks, and the size of their effect in general. Still, there is some information leaking, as of course we also choose the ranges of the different parameters that we investigate based on initial experiments on both the test and validation sets, and there is no guarantee that this will transfer identically to different tasks or datasets. However, for the present study, this approach still gives valuable insights, and we consider more robust selection of parameters the goal of further research. We will discuss which hyperparameters are varied in Section V-B and give the exact parameters used in Section V-C.

## V. EXPERIMENTS

### A. Dataset and Few-Shot Tasks

The *semantic change detection dataset* (SECOND) [36] consists of 2968<sup>2</sup> image pairs of size 512 × 512, obtained from several Chinese cities. As it has been designed for semantic change detection, it also includes pixel-level annotations for land-cover classes of the changed areas in both images, with the categories *nonvegetated ground surface*, *tree*, *low vegetation*, *water*, *buildings*, and *playgrounds*.<sup>3</sup> As these annotations are provided for both time steps and include an additional *non-change* label, this enables also the description of change where the land-cover class stays the same, e.g., changes from one building to another.

As mentioned, we are not interested in semantic change detection as such. However, the structure of the dataset is very well suited to our task as well: during training, all change categories are grouped together, and we perform full binary change detection. Then, for the few-shot tasks, we can select individual change categories (e.g., *from any class to buildings*) and treat all others as unchanged. Fig. 2 shows an illustration of this process. A similar approach is also described by Liu et al. [55], who use the labels of the semantic change detection dataset HRSCD [34] to only select *cropland changes* as their binary labels.

This definition of few-shot tasks, while providing an easy way to adapt an existing dataset and make use of its high quality labels also for the few-shot setting, nevertheless has some shortcomings: first of all, we treat every pixel independently, and therefore have no way to determine if a removed “tree” pixel was part of a large forest, or an isolated roadside tree. Similarly, as there is no notion of spatial structure, we cannot, for example,

<sup>2</sup>The full dataset consists of 4662 pairs, however, we only had access to the train set. We then divided the data available to us into a new train-validation-test split with roughly an 8:1:1 ratio, making an effort to avoid overlap between the different sets.

<sup>3</sup>The last category mostly refers to sports fields.

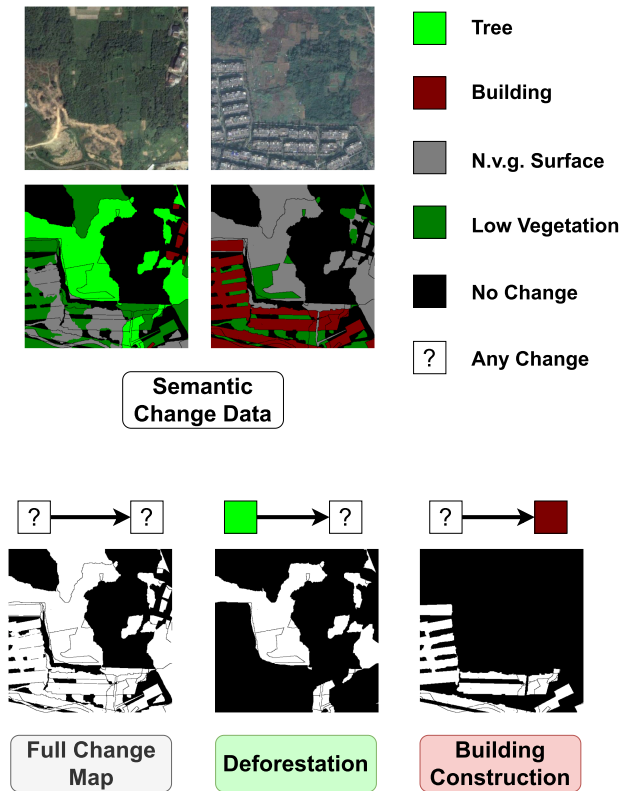


Fig. 2. Illustration of the process for dataset preparation: Starting from the semantic annotations from SECOND, we can create both the full binary change map, as well as different few-shot tasks by combining different sorts of transitions.

easily define *road construction* as a task in this way. Also, we are limited in granularity by the decisions of the original dataset, which means that we cannot determine which type of “building” we see, as it can be a skyscraper, a factory hall, a residential building, or anything else.

We choose four different of these tasks for our experiments: *surface change*, which is defined by either “from n.v.g. surface to low vegetation or from low vegetation to n.v.g. surface,” *deforestation* (“from tree to any class”), *building demolition* (“from building to any class”), and *building construction* (“from any class to building”). In each case, the support set was manually selected from the validation data to be representative of this change type and has a size of five  $256 \times 256$  patches. We show the first two support images for every task in Fig. 3. The choice and quality of the support images likely has a considerable impact on the performance of the few-shot methods, and exploring this might be a valuable target for further research.

The four tasks differ quite a lot, both in their difficulty and frequency of change: while it might be relatively easy to determine if a new building was constructed, the changes between nonvegetated and vegetated surface can be hard to assess even for a human. In addition, *deforestation* is very rare in the test set, with only 0.49% (0.86% on the validation set and 1.60% in the train set) of pixels from the images undergoing this change, compared to 1.71% (3.34% / 3.31%) for *building demolition*, 10.78% (6.79% / 8.52%) for *building construction*, and 6.26% (5.18% / 5.67%) for *surface change*. For reference, the amount of all changed pixels together is 20.70%, very similar to the

value of 20.19% on the train set, and slightly higher than the 16.91% on the validation set.

Therefore, we hope to have a somewhat broad representation of different scenarios one might want to use few-shot filtering in, and to be able to compare what works well in what circumstances.

### B. Hyperparameters Under Investigation

We will now shortly introduce the parameters for the fine-tuning phase that we investigate more closely (cf. Section IV-C). The concrete values used are given in Section V-C.

*Change Weight:* Change detection is an inherently imbalanced task, as there are always less changed pixels compared to unchanged ones, even for the more general base training set. Usually, we can solve this problem relatively easily by *weighted cross entropy*, where we use a weight term (that we call *change weight*) to give more importance to the rarer change instances and to avoid learning a network that simply predicts “no change” for every pixel.

In the typical change detection setting, and therefore also for the base training, we can take the frequency of change in the train set as a reference point to set this weight. However, in the fine-tuning phase, the support set will usually have a much higher amount of changed pixels. If we are interested in deforestation, for example, we will usually select images of forests or parks where some logging happened, to guide the few-shot process. This will not reflect the distribution in the actual test set, where also other scenes might be present. Therefore, we commonly have a mismatch between training and test data in terms of frequency of change during the few-shot phase.

Another issue is that we want to learn is specialized change, which is by definition not as frequent as the general change in the base training data. This implies that learning *unchanged* pixels (that is, “forgetting” change) is the most important part of fine-tuning, which will lead us to bias the training more toward unchanged pixels in this phase (or at least less strongly toward changed ones).

Both of these aspects suggest using a lower change weight in the fine-tuning phase. It also seems reasonable to assume that for few-shot tasks where change is very rare (in our case, deforestation is such a task), a lower weight might be sensible than for a relatively common type of change (such as building construction). Therefore, we investigate the impact of two different change weights, both lower than the one used during base training.

*Number of Fine-tuning Epochs:* How long we train during fine-tuning should determine the influence of the (specialized) support set compared to the (general) base training data. Initially, the information the model has learned from the full dataset should dominate, but over time, it will fit more and more to the specialized and narrow category from the few new images. Balancing this is therefore very important, and in addition, this also has a direct impact on performance. Using, e.g., 100 instead of 10 epochs will also increase the duration of fine-tuning around ten times. As with the change weight, we will investigate a shorter and a longer number of epochs in our experiments.

*Learning Rate:* The learning rate decides how much the model adapts in each step. Choosing a suitable value for this

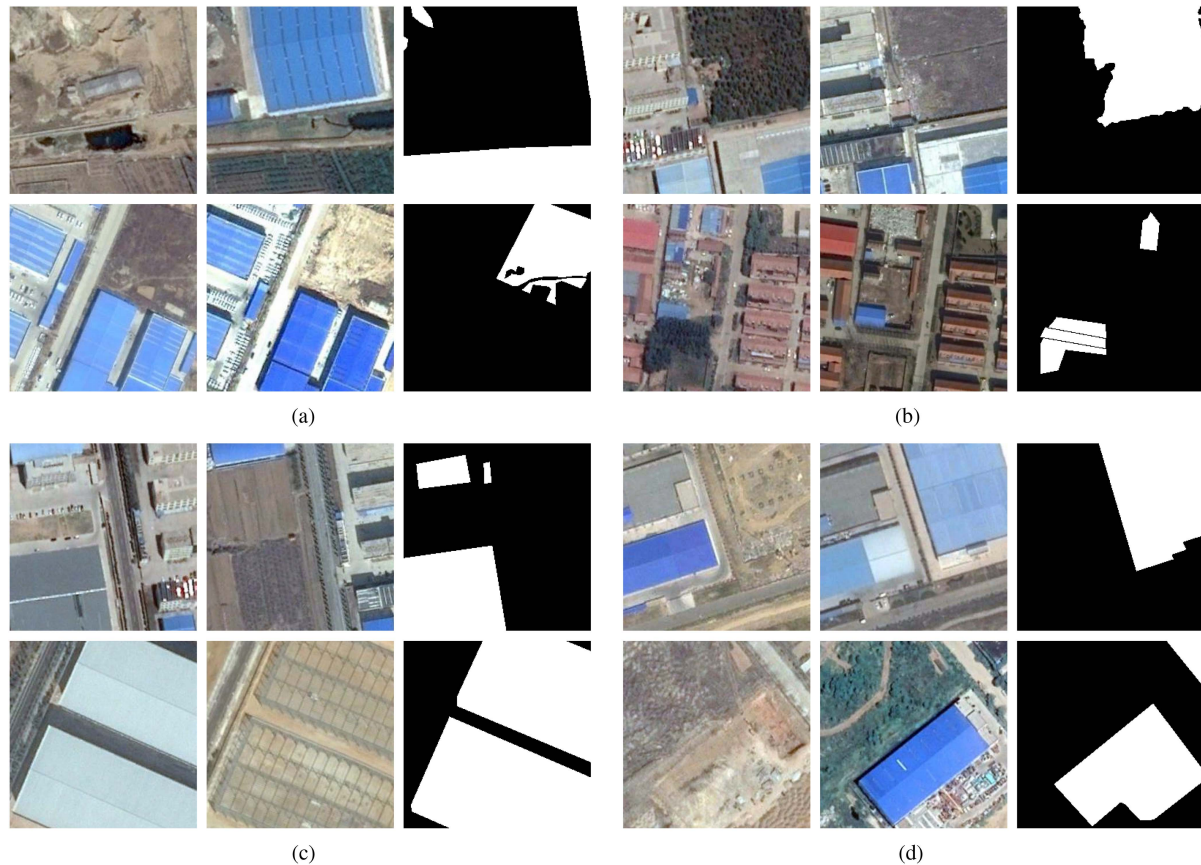


Fig. 3. First two image pairs from the support set of all four few-shot tasks. One thing we can notice, for example, is how the second image for *building demolition* shows a change from one building to another, as this also fits the definition for this task. (a) Surface Change. (b) Deforestation. (c) Building Demolition. (d) Building Construction.

is necessary for the information in the support set to improve the overall performance. However, in order to keep the scope of the investigations in this study reasonable and to not impede the analysis by too many variables, we decided on keeping this at a fixed value across all experiments.

*Dropout:* We do not use dropout during base training, as initial experiments suggested a slightly worse performance on the validation set when considering all change as relevant. However, during fine-tuning, dropout might be very valuable to avoid overfitting to the very small support set. We found that the effect is different for different tasks, which is why we include it in the parameters we investigate separately. Note that this is not the same as using dropout during test time, as it is often done in order to assess the uncertainties of the model [56]. During inference, it is turned OFF as usual.

### C. Setup

*Backbone:* We use a standard change detection backbone, namely, the *FC-Siam-Conc* by Daudt et al. [57], as implemented by the *TorchGeo* Python package [58]. We adapted it slightly from this implementation, removing the final block (which consists of a  $3 \times 3$  transposed convolution, batch normalization, an ReLU activation and a dropout layer), the dropout from the second to last block, and replacing all that by a  $1 \times 1$  convolutional layer. Also, we disabled dropout during training, as initial experiments showed slightly better results for the base

change detection task. Dropout during fine-tuning is part of the hyperparameters investigated.

*Base Training:* The model is trained with Adam (using standard parameters), an initial learning rate of  $5 \cdot 10^{-4}$  that decays exponentially with a  $\gamma$  of 0.95 and a weight decay factor of  $1 \cdot 10^{-4}$ . The maximum number of epochs is 100, but we choose the one with the lowest validation loss. Each  $512 \times 512$  image is split into 9 patches of size  $256 \times 256$ , with overlap to reduce edge effects due to padding, and the batch size is  $B = 32$ . For the change weight, we choose  $0.5 \cdot (1/p_{\text{change}} - 1)$ , where  $p_{\text{change}} = 0.2019$  is the fraction of changed pixels in the train set images.

*Fine-tuning:* For fine-tuning, we also use Adam, a learning rate of  $5 \cdot 10^{-4}$ , and a weight decay factor of  $1 \cdot 10^{-4}$ . However, as the epochs are much shorter, we do not use learning rate scheduling. The number of fine-tuning epochs, the change weight, and whether to use dropout are varied as part of the parameter studies, and we use values 25 and 75 for the epochs, 1 and 0.1 for the weight and a dropout probability of 0.2 where it is applied.

*Training of Baseline B:* In the case of Baseline B (i.e., training directly on the support set), we decided on individual hyperparameters by tuning on the validation set. Of course, the discussion of Section IV-C remains true, and we cannot do this in practice. However, the goal of this baseline is to see how much information is contained in the small support set and can be recovered with the backbone, so it serves more as a

TABLE I  
MAIN RESULTS

Method	IoU	Precision	Recall	F1
Base Task (Full Change)				
Backbone	50.3 (0.5)	59.4 (1.1)	76.6 (1.0)	66.9 (0.4)
Few-Shot Task 1 (Surface Change)				
Baseline A (Base model only)	13.0 (0.2)	14.1 (0.2)	<b>62.7 (1.9)</b>	23.1 (0.4)
Baseline B (Support set only)	12.8 (1.7)	16.3 (2.7)	40.9 (14.0)	22.6 (2.8)
Fine-tuning	<b>18.2 (0.9)</b>	<b>26.7 (1.7)</b>	36.7 (3.9)	<b>30.7 (1.3)</b>
Few-Shot Task 2 (Deforestation)				
Baseline A (Base model only)	0.7 (0.0)	0.7 (0.0)	<b>42.8 (2.5)</b>	1.5 (0.1)
Baseline B (Support set only)	4.1 (0.5)	4.7 (0.9)	30.0 (10.6)	7.9 (0.9)
Fine-tuning	<b>5.9 (0.9)</b>	<b>6.5 (1.2)</b>	40.9 (4.2)	<b>11.1 (1.6)</b>
Few-Shot Task 3 (Building Demolition)				
Baseline A (Base model only)	4.4 (0.1)	4.4 (0.1)	<b>71.0 (2.4)</b>	8.3 (0.3)
Baseline B (Support set only)	5.8 (1.4)	6.7 (1.8)	33.9 (9.2)	11.0 (2.5)
Fine-tuning	<b>11.3 (1.9)</b>	<b>15.9 (4.2)</b>	30.4 (5.1)	<b>20.3 (3.2)</b>
Few-Shot Task 4 (Building Construction)				
Baseline A (Base model only)	35.6 (0.9)	37.3 (1.0)	<b>89.1 (0.8)</b>	52.5 (1.0)
Baseline B (Support set only)	33.1 (3.5)	45.3 (5.7)	56.0 (6.5)	49.7 (4.0)
Fine-tuning	<b>49.3 (1.9)</b>	<b>62.9 (3.6)</b>	69.7 (3.8)	<b>66.0 (1.7)</b>

Best results are given in bold.

lower boundary than a real practical suggestion. In addition, experiments suggest that this baseline is much less sensitive to hyperparameter changes, and indeed using the same settings for all tasks yields results that are almost identical. The values used are a change weight of 0.1, a learning rate of  $3 \cdot 10^{-4}$ , trained for 3000 epochs<sup>4</sup> in the case of surface change, a weight of 2, learning rate of  $2 \cdot 10^{-4}$  and 5000 epochs for deforestation, a change weight of 5, learning rate of  $4 \cdot 10^{-4}$  trained for 7500 epochs for building demolition and a change weight of 1, learning rate of  $5 \cdot 10^{-4}$  trained for 7500 epochs for building construction. Dropout with probability 0.2 was used for all tasks except surface change. All other aspects of the training are done as in the fine-tuning case.

*Number of runs:* In order to account for the stochastic nature of training in deep learning, we perform ten training runs with different random seeds. In addition, when performing fine-tuning, we run each of these models two times to account for variance there. Similarly, for Baseline B, we also train 20 models, to have the same number of evaluations in the end.

*Metrics:* For qualitative comparison, we use the standard metrics in change detection and semantic segmentation: *Intersection over Union* (IoU) and *Precision* (Prec), *Recall* (Rec) and *F1-Score* (F1), computed from the binary annotation masks and ground truth, both of the full task and the reduced ones.

## VI. RESULTS

The main results are shown in Table I (with standard deviations in brackets) and example outputs can be found in Fig. 4. Additional parameter variations are recorded in Table II. For the main results, in each task we use the parameters that give the highest F1-score (which should be taken with a bit of care, since,

as mentioned above, we cannot choose these parameters based on a validation set in practice).

The first thing we can notice is the considerable difference between individual tasks, with a very low precision and overall performance on the *deforestation* task being the most noticeable. The most straight forward explanation for this is the fact that this change type is very rare, which more easily leads to a higher number of false positives. Indeed, we find that ranking the tasks by their F1 scores, their precision values and the percent of changed pixels (cf. Section V-A) all yield the same order, suggesting a clear connection there.

Putting aside the intertask variability, we can see that in every case, fine-tuning can considerably improve the performance compared to both baselines, showing that the few-shot filtering setting does indeed provide significant benefit for the detection of specialized change. However, regarding the comparison with the Baseline B, we have to note that while we optimize the hyperparameters for this on the *validation set*, for the fine-tuning approach, we choose the best performance on the *test set* (albeit over a much less extensive parameter range), so this has to be taken into account. Still, if we look at the worst hyperparameter choice for fine-tuning (which we are unlikely to pick when using the validation set in a comparable experiment), then we still beat Baseline B on three out of four tasks, albeit only by a slight margin on two of them, and for “reasonably adapted” choices (e.g., only choosing whether to use dropout during fine-tuning or not), the advantage is clear for all tasks.

Regarding hyperparameters, we notice that the best performing values are different for all four tasks, even in our restricted set of eight different combinations. This suggests that indeed some decision needs to be done based on the (expected) characteristics of each specialized change. Further, the effect of dropout is mixed. Generally, it increases precision and lowers recall, however, for the task of building demolition, it hurts both measures. Also, it shows a tendency to increase the variance between model runs. Considering the other parameters, increasing the

<sup>4</sup>These numbers might seem very high, but one epoch consists of only the five samples in the support set, which are treated as a single batch. Also, we can achieve quite good results already with a much lower number of epochs.



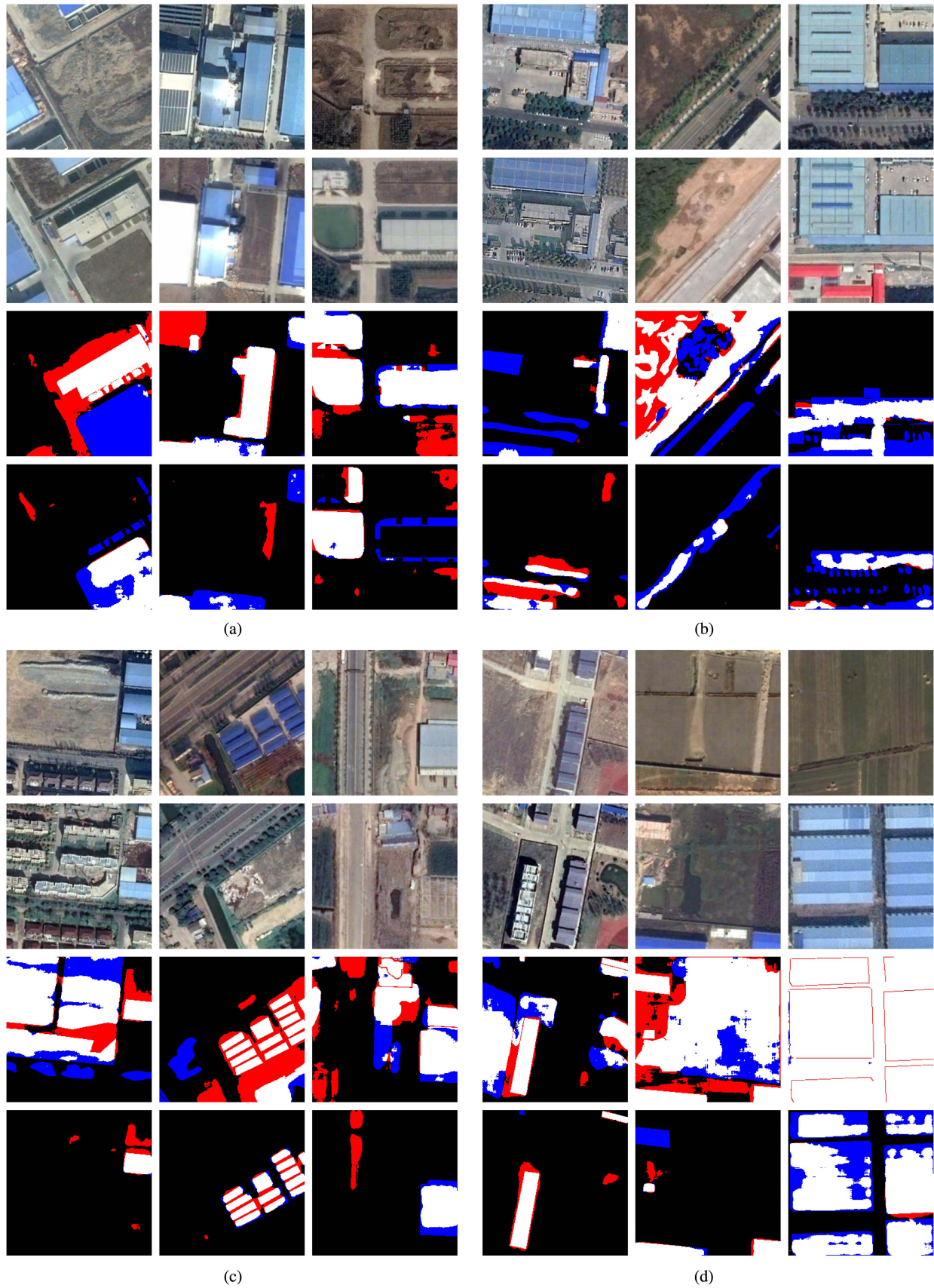


Fig. 4. Example results. For every task, from top to bottom, we have prechange images, then postchange, the output of the full model, and the filtered ones. True positives are colored white, false positives red, and false negatives blue. (a) Surface Change. (b) Deforestation. (c) Building Demolition. (d) Building Construction.

TABLE II  
ADDITIONAL PARAMETER STUDIES

Epochs	Weight	Dropout	IoU	Precision	Recall	F1
Few-Shot Task 1 (Surface Change)						
25	0.1	0.0	15.9 (0.6)	27.8 (2.4)	27.2 (2.4)	27.4 (0.9)
75	0.1	0.0	17.5 (0.9)	29.1 (2.3)	30.8 (2.4)	29.8 (1.2)
25	1.0	0.0	<b>18.2 (0.9)</b>	26.7 (1.7)	36.7 (3.9)	<b>30.7 (1.3)</b>
75	1.0	0.0	18.1 (1.0)	25.6 (2.0)	<b>38.7 (4.3)</b>	30.6 (1.4)
25	0.1	0.2	13.2 (2.7)	34.7 (4.7)	18.5 (5.9)	23.2 (4.3)
75	0.1	0.2	13.4 (3.4)	<b>37.3 (5.1)</b>	18.3 (6.8)	23.5 (5.3)
25	1.0	0.2	15.5 (2.2)	30.5 (3.1)	25.0 (6.1)	26.8 (3.3)
75	1.0	0.2	15.9 (2.4)	27.7 (3.3)	28.0 (6.2)	27.4 (3.6)
Few-Shot Task 2 (Deforestation)						
25	0.1	0.0	2.9 (0.3)	3.0 (0.3)	58.6 (3.7)	5.7 (0.5)
75	0.1	0.0	4.3 (0.4)	4.5 (0.5)	49.4 (2.6)	8.3 (0.8)
25	1.0	0.0	2.5 (0.3)	2.5 (0.3)	<b>72.7 (4.0)</b>	4.9 (0.6)
75	1.0	0.0	3.8 (0.3)	3.9 (0.4)	58.2 (4.2)	7.3 (0.6)
25	0.1	0.2	<b>5.9 (0.9)</b>	6.5 (1.2)	40.9 (4.2)	<b>11.1 (1.6)</b>
75	0.1	0.2	5.4 (1.7)	<b>8.2 (1.7)</b>	16.2 (8.0)	10.3 (3.1)
25	1.0	0.2	4.8 (0.8)	5.1 (1.1)	45.9 (7.7)	9.1 (1.5)
75	1.0	0.2	5.0 (1.3)	7.2 (2.2)	19.5 (10.6)	9.6 (2.4)
Few-Shot Task 3 (Building Demolition)						
25	0.1	0.0	<b>11.3 (1.9)</b>	<b>15.9 (4.2)</b>	30.4 (5.1)	<b>20.3 (3.2)</b>
75	0.1	0.0	11.2 (1.9)	15.3 (3.3)	30.6 (4.5)	20.0 (3.0)
25	1.0	0.0	10.5 (1.9)	12.2 (2.5)	<b>45.0 (5.2)</b>	19.0 (3.1)
75	1.0	0.0	10.6 (2.4)	13.2 (3.6)	37.3 (3.9)	19.2 (3.8)
25	0.1	0.2	10.5 (2.0)	15.0 (3.9)	27.5 (4.4)	18.9 (3.4)
75	0.1	0.2	9.8 (2.3)	13.0 (3.9)	29.8 (4.1)	17.7 (3.7)
25	1.0	0.2	8.8 (2.0)	10.2 (2.6)	42.1 (4.6)	16.2 (3.3)
75	1.0	0.2	7.5 (1.9)	8.7 (3.0)	37.4 (3.5)	13.9 (3.3)
Few-Shot Task 4 (Building Construction)						
25	0.1	0.0	44.1 (3.0)	59.8 (6.0)	63.2 (3.9)	61.2 (2.9)
75	0.1	0.0	43.3 (3.9)	63.1 (5.3)	58.3 (5.9)	60.4 (3.9)
25	1.0	0.0	40.8 (2.9)	46.1 (3.8)	<b>78.3 (3.8)</b>	57.9 (2.9)
75	1.0	0.0	42.7 (3.4)	51.1 (4.9)	72.4 (2.0)	59.8 (3.4)
25	0.1	0.2	45.6 (3.9)	74.9 (3.8)	54.1 (6.1)	62.5 (3.8)
75	0.1	0.2	39.4 (7.5)	<b>75.6 (5.7)</b>	45.9 (10.0)	56.2 (8.3)
25	1.0	0.2	<b>49.3 (1.9)</b>	62.9 (3.6)	69.7 (3.8)	<b>66.0 (1.7)</b>
75	1.0	0.2	46.2 (3.5)	69.3 (4.8)	58.2 (4.6)	63.1 (3.3)

Best results are given in bold.

amount of fine-tuning epochs, as well as lowering the change weight both also have the effect of increasing precision and lowering recall, with some exceptions and differing size of the effect. In the case of training for more epochs, this confirms the hypothesis that we start with high recall and low precision on the base model, and training for longer adapts better to the support set, gaining precision but losing some change instances in the process. However, as noted, there are some exceptions (all in the surface change and building demolition tasks), and a larger number of epochs can even yield the opposite result. The effect of the change weight is more consistent and can also be easily explained, as a higher value gives relatively less importance to falsely identified change instances, therefore allowing for more false positives and a higher recall, but lower precision.

## VII. DISCUSSION

Considering the results described in the previous section, we see that combining two different data sources indeed performs better than each one on their own, showing that the basic assumption is reasonable. The large differences between the individual

tasks, however, show that—at least for the simple fine-tuning approach investigated in this article—change categories that are very rare in the base dataset do perform worse than relatively common ones. While these low performances might be an issue for some practical applications, we should also add that pixelwise statistics are not always the only relevant metrics. For example, when deciding on whether to update maps in certain regions, only a general measure of the amount of change is needed, and there the boost in precision might be very beneficial already.

Also, we restate the impact of a smart choice of parameters for every task, as for some settings, fine-tuning even performs worse than Baseline B. As a first observation, we have already seen that rare changes naturally have a low precision in the unadapted base model, therefore we should generally use a lower change weight, dropout, or more fine-tuning epochs in these cases. However, this is not the full picture, and, e.g., we find for all tasks that the best results are achieved with a lower number of epochs and that dropout hurts performance for the relatively rare building demolition. Investigating the interaction of the

individual hyperparameters, finding reasons for heterogeneous effects, such as that of the number of epochs, and exploring methods to find good values in a low data setting therefore are interesting lanes for further research.

We also see that Baseline B can perform surprisingly well, given that it only ever uses the five support images and has no access to the base training set. We cannot even exclude the possibility that, in particular with a very good choice of hyperparameters, a well-designed training strategy or a better suited network architecture, Baseline B can be competitive to fine-tuning, and this might be worth investigating in future work. However, we believe that even for such approaches, there might still be value in using the base training set as an additional information source in some way, and that by this, the few-shot filtering setting still is the right lens for these approaches.

### VIII. CONCLUSION

In this article, we have presented a new task for specialized change detection in low data regimes, investigated a fine-tuning approach to tackle it, compared it to two simple baselines, and studied the effect of hyperparameters that are difficult to assess in lack of validation data. In addition, we described a way to adapt existing semantic change detection datasets to act as a proxy for the few-shot tasks, enabling the use of trusted data sources for this new setting. While there is still some room for improvement regarding the overall quality of the results, the ideas here should be seen as a first concrete step into the direction of adaptive change detection using only a few samples, and we hope that our approach of filtering out change will lead to new applications in less explored domains or geographic regions for which currently no large public training corpora are available, helping uncommon usecases and underrepresented communities.

### ACKNOWLEDGMENT

M. Hermann would like to thank IABG for helpful discussions and further resources, and in particular Dr. Yasmine Israeli, Dr. Martin Willberg and Peter Schauer for their guidance and advice.

### REFERENCES

- [1] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sens.*, vol. 12, no. 10, 2020, Art. no. 1688.
- [2] S. Saha, F. Bovolo, and L. Bruzzone, "Building change detection in VHR SAR images via unsupervised deep transcoding," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 1917–1929, Mar. 2021.
- [3] P. de Bem, O. de C. Júnior, R. Guimarães, and R. Gomes, "Change detection of deforestation in the Brazilian Amazon using landsat data and convolutional neural networks," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 901.
- [4] B. Peng, Z. Meng, Q. Huang, and C. Wang, "Patch similarity convolutional neural network for urban flood extent mapping using bi-temporal satellite multispectral imagery," *Remote Sens.*, vol. 11, no. 21, 2019, Art. no. 2492.
- [5] D. Han, S. Lee, M. Song, and J. Cho, "Change detection in unmanned aerial vehicle images for progress monitoring of road construction," *Buildings*, vol. 11, no. 4, 2021, Art. no. 150.
- [6] S. Camalan et al., "Change detection of amazonian alluvial gold mining using deep learning and Sentinel-2 imagery," *Remote Sens.*, vol. 14, no. 7, 2022, Art. no. 1746.
- [7] L. Ma, W. Liu, Z. Han, J. Wang, and H. Chen, "Inshore ship change detection based on spatial-temporal saliency," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1641–1644.
- [8] R. Daudt, B. L. Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral earth observation using convolutional neural networks," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2018, pp. 2115–2118.
- [9] H. Li, F. Zhu, X. Zheng, M. Liu, and G. Chen, "MSCDUNet: A deep learning framework for built-up area change detection integrating multispectral, SAR and VHR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 5163–5176, 2022.
- [10] W. Chen, Y. Liu, Z. Kira, Y. Wang, and J. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–16.
- [11] Y. Wang, Q. Yao, J. Kwok, and L. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.
- [12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
- [13] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 3630–3638, 2016.
- [14] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.
- [15] S. Laenen and L. Bertinetto, "On episodes, prototypical networks, and few-shot learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 24581–24592, 2021.
- [16] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4367–4375.
- [17] H. Qi, M. Brown, and D. Lowe, "Low-shot learning with imprinted weights," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5822–5830.
- [18] G. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A baseline for few-shot image classification," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–20.
- [19] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–13.
- [20] X. Zhang, Y. Wei, Y. Yang, and T. Huang, "SG-One: Similarity guidance network for one-shot semantic segmentation," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3855–3865, Sep. 2020.
- [21] M. Siam, B. Oreshkin, and M. Jagersand, "AMP: Adaptive masked proxies for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5249–5258.
- [22] L. Yang, W. Zhuo, L. Qi, Y. Shi, and Y. Gao, "Mining latent classes for few-shot segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8721–8730.
- [23] S. Caelles, K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool, "One-shot video object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 221–230.
- [24] Y. Chen, J. Pont-Tuset, A. Montes, and L. V. Gool, "Blazingly fast video object segmentation with pixel-wise metric learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1189–1198.
- [25] M. Huh, P. Agrawal, and A. Efros, "What makes ImageNet good for transfer learning?," 2016, *arXiv:1608.08614*.
- [26] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, "Deep hierarchical semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1246–1257.
- [27] M. Mel, U. Michieli, and P. Zanuttigh, "Incremental and multi-task learning strategies for coarse-to-fine semantic segmentation," *Technologies*, vol. 8, no. 1, 2019, Art. no. 1.
- [28] Y. Aytar and A. Zisserman, "Tabula Rasa: Model transfer for object category detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2011, pp. 2252–2259.
- [29] G. Bukchin et al., "Fine-grained angular contrastive learning with coarse labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8726–8736.
- [30] C. Phoo and B. Hariharan, "Coarsely-labeled data for better few-shot transfer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9052–9061.
- [31] J. Ni et al., "Superclass-conditional Gaussian mixture model for learning fine-grained embeddings," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–20.
- [32] X. Xiang, Y. Tan, Q. Wan, J. Ma, A. Yuille, and G. Hager, "Coarse-to-fine incremental few-shot learning," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 205–222.

- [33] R. Gong et al., "TACS: Taxonomy adaptive cross-domain semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 19–35.
- [34] R. Daudt, B. L. Saux, A. Boulch, and Y. Gousseau, "Multitask learning for large-scale semantic change detection," *Comput. Vis. Image Understanding*, vol. 187, 2019, Art. no. 102783.
- [35] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, Feb. 2019.
- [36] K. Yang et al., "Asymmetric siamese networks for semantic change detection in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–18, 2021.
- [37] S. Verma, A. Panigrahi, and S. Gupta, "QFabric: Multi-task change detection dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1052–1061.
- [38] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised deep change vector analysis for multiple-change detection in VHR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3677–3693, Jun. 2019.
- [39] D. Peng, L. Bruzzone, Y. Zhang, H. Guan, H. Ding, and X. Huang, "SemiCDNet: A semisupervised convolutional neural network for change detection in high resolution remote-sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5891–5906, Jul. 2021.
- [40] M. Leenstra, D. Marcos, F. Bovolo, and D. Tuia, "Self-supervised pre-training enhances change detection in Sentinel-2 imagery," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 578–590.
- [41] J. Liu et al., "Convolutional neural network-based transfer learning for optical aerial images change detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 127–131, Jan. 2020.
- [42] V. Růžička, S. D'Aronco, J. Wegner, and K. Schindler, "Deep active learning in remote sensing for data efficient change detection," in *Proc. MACLEAN: Mach. Learn. Earth Observ. Workshop Co-Located Eur. Conf. Mach. Learn. Princ. Pract. Knowl. Discov. Databases*, 2020, pp. 1–10.
- [43] S. Saha, B. Banerjee, and X. X. Zhu, "Trusting small training dataset for supervised change detection," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2021, pp. 1–4.
- [44] M. Khoshboresh-Masouleh and R. Shah-Hosseini, "Deep few-shot learning for bi-temporal building change detection," *ISPRS-Int. Arch. Photogrammetry Remote Sens. Spatial Inf. Sci.*, vol. 44, pp. 99–103, 2021.
- [45] R. Wang, W. Wang, P. Dong, W. Haojiang, L. Jiao, and J. Chen, "SAR image change detection via a few-shot learning-based neural network," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2021, pp. 5287–5290.
- [46] P. Tang, J. Li, F. Ding, W. Chen, and X. Li, "PSNet: Change detection with prototype similarity," *Vis. Comput.*, vol. 28, pp. 3541–3550, 2021.
- [47] G. Lenczner, A. Chan-Hon-Tong, N. Luminari, and B. Le Saux, "Weakly-supervised continual learning for class-incremental segmentation," *IEEE Int. Geosci. Remote Sens. Symp.*, pp. 4843–4846, Jul. 17, 2022.
- [48] Z. Zheng, Y. Liu, S. Tian, J. Wang, A. Ma, and Y. Zhong, "Weakly supervised semantic change detection via label refinement framework," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 2066–2069.
- [49] L. L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. Hershey, PA, USA: IGI Global, 2010, pp. 242–264.
- [50] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–42.
- [51] H. Li et al., "Rethinking the hyperparameters for fine-tuning," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–20.
- [52] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–29.
- [53] S. Hu, D. Li, J. Stühmer, M. Kim, and T. Hospedales, "Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9068–9077.
- [54] A. Kolesnikov et al., "Big transfer (BiT): General visual representation learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 491–507.
- [55] M. Liu, Z. Chai, H. Deng, and R. Liu, "A CNN-transformer network with multiscale context aggregation for fine-grained cropland change detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4297–4306, 2022.
- [56] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.
- [57] R. Daudt, B. L. Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *Proc. IEEE/CVF Int. Conf. Image Process.*, 2018, pp. 4063–4067.

- [58] A. J. Stewart, C. Robinson, I. A. Corley, A. Ortiz, J. M. Ferres, and A. Banerjee, "Torchgeo: Deep learning with geospatial data," in *Proc. 30th Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 1, 2022, pp. 1–12.



Mr. Hermann was the recipient of the Munich Aerospace scholarship in 2021.

**Martin Hermann** received the B.Sc. degrees in computational linguistics and mathematics from the Ludwig Maximilian University of Munich (LMU), Munich, Germany, in 2015 and 2018, respectively, the M.Sc. degree in mathematics in data science, in 2021, from the Technical University of Munich (TUM), Munich, Germany, where he is currently working toward the Ph.D. degree with the Data Science in Earth Observation Group, TUM.

His research interests include change detection, few-shot learning, and low data approaches.



**Sudipan Saha** (Member, IEEE) received the M.Tech. degree in electrical engineering from IIT Bombay, Mumbai, Maharashtra, India, in 2014, and the Ph.D. degree in information and communication technologies from the University of Trento, Trento, Italy, and Fondazione Bruno Kessler, Trento, in 2020.

He was an Engineer with TSMC Ltd., Hsinchu, Taiwan, from 2015 to 2016. In 2019, he was a Guest Researcher with the Technical University of Munich (TUM), Munich, Germany, where he has been a Post-doctoral Researcher, since 2020. His research inter-

ests include multitemporal remote sensing image analysis, domain adaptation, time-series analysis, image segmentation, deep learning, image processing, and pattern recognition.

Dr. Saha was the recipient of the Fondazione Bruno Kessler Best Student Award 2020. He is a Reviewer for several international journals. He served as a Guest Editor for *Remote Sensing* (MDPI) special issue on "Advanced Artificial Intelligence for Remote Sensing: Methodology and Application."



**Xiao Xiang Zhu** (Fellow, IEEE) received the master's (M.Sc.) degree, doctor of engineering (Dr.-Ing.) degree, and "Habilitation" degree in signal processing from Technical University of Munich (TUM), Munich, Germany, in 2008, 2011, and 2013, respectively.

She is the Chair Professor of Data Science with Earth Observation, Technical University of Munich, and was the Founding Head of the Department "EO Data Science," Remote Sensing Technology Institute, German Aerospace Center (DLR), Cologne, Germany. Since 2019, she has been a Cooordinator with the Munich Data Science Research School, Munich, Germany, ([www.mu-ds.de](http://www.mu-ds.de)).

Since 2019, she also heads the Helmholtz Artificial Intelligence – Research Field "Aeronautics, Space and Transport." Since May 2020, she has been the PI and Director of the international future AI lab "AI4EO – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond," Munich, Germany. Since October 2020, she also serves as a co-Director of the Munich Data Science Institute, TUM. She was a Guest Scientist or visiting Professor with the Italian National Research Council, Naples, Italy, Fudan University, Shanghai, China, the University of Tokyo, Tokyo, Japan, and the University of California, Los Angeles, CA, USA, in 2009, 2014, 2015, and 2016, respectively. She is currently a visiting AI Professor with ESA's Phi-lab. She is a member of the young academy (Junge Akademie/Junges Kolleg) with the Berlin-Brandenburg Academy of Sciences and Humanities, Berlin, Germany, and the German National Academy of Sciences Leopoldina, Halle, Germany, and the Bavarian Academy of Sciences and Humanities, Munich, Germany. Her research interests are remote sensing and Earth observation, signal processing, machine learning and data science, with their applications in tackling societal grand challenges, e.g., Global Urbanization, UN's SDGs and Climate Change.

Dr. Zhu serves on the scientific advisory board in several research organizations, among others the German Research Center for Geosciences and Potsdam Institute for Climate Impact Research. She is an Associate Editor for *IEEE TRANSACTIONS ON GEOSCIENCE* and *Remote Sensing* and serves as the Area Editor responsible for special issues of *IEEE Signal Processing Magazine*.