Machine Learning (2023) 112:2211–2237
https://doi.org/10.1007/s10994-021-06008-4

Weakly supervised change detection using guided anisotropic diffusion

Rodrigo Caye Daudt[1,4,5] · Bertrand Le Saux[2] · Alexandre Boulch[3] · Yann Gousseau[4]

Received: 12 June 2020 / Revised: 22 February 2021 / Accepted: 24 May 2021 /
Published online: 10 June 2021

Abstract

Large scale datasets created from crowdsourced labels or openly available data have become crucial to provide training data for large scale learning algorithms. While these datasets are easier to acquire, the data are frequently noisy and unreliable, which is motivating research on weakly supervised learning techniques. In this paper we propose original ideas that help us to leverage such datasets in the context of change detection. First, we propose the guided anisotropic diffusion (GAD) algorithm, which improves semantic segmentation results using the input images as guides to perform edge preserving filtering. We then show its potential in two weakly-supervised learning strategies tailored for change detection. The first strategy is an iterative learning method that combines model optimisation and data cleansing using GAD to extract the useful information from a large scale change detection dataset generated from open vector data. The second one incorporates GAD within a novel spatial attention layer that increases the accuracy of weakly supervised networks trained to perform pixel-level predictions from image-level labels. Improvements with respect to state-of-the-art are demonstrated on 4 different public datasets.

Keywords Remote sensing · Change detection · Weak supervision · Neural networks · Anisotropic diffusion

Editors: Dino Ienco, Thomas Corpetti, Minh-Tan Pham, Sebastien Lefevre, Roberto Interdonato.

✉ Rodrigo Caye Daudt
rodrigo.cayedaudt@geod.baug.ethz.ch

[1] Photogrammetry and Remote Sensing, ETH Zurich, Zurich, Switzerland

[2] European Space Agency, ESRIN Φ-lab, 00044 Frascati, Rome, Italy

[3] valeo.ai, 75008 Paris, France

[4] Télécom Paris, Institut Polytechnique de Paris, Paris, France

[5] ONERA, DTIS, Université Paris Saclay, F-91123 Palaiseau Cedex, France

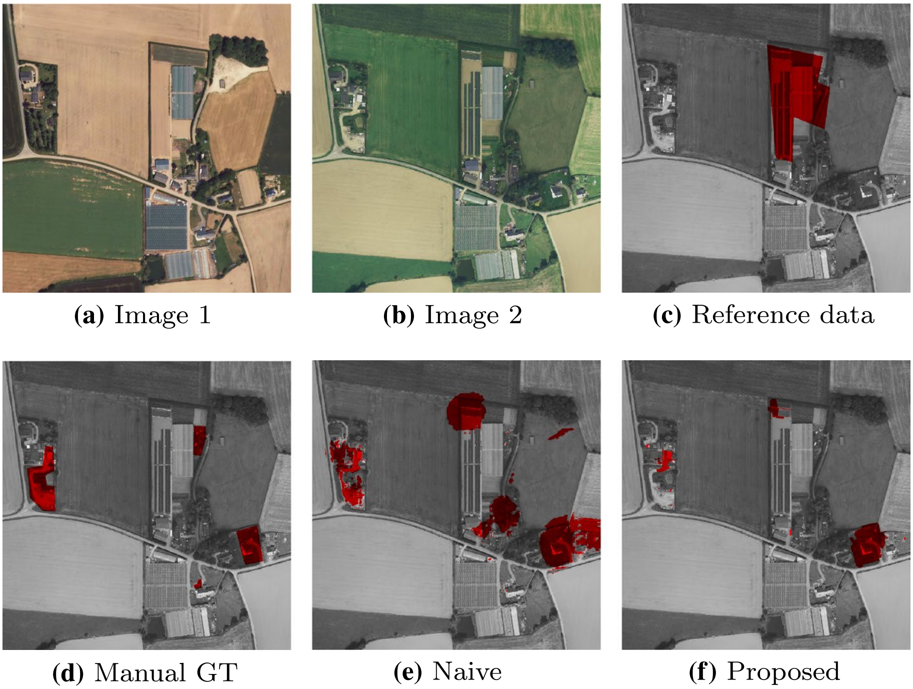
1 Introduction

Change detection (CD) is one of the oldest problems studied in the field of remote sensing image analysis (Hussain et al., 2013; Singh, 1989). It consists of comparing a pair or sequence of coregistered images and identifying the regions where meaningful changes have taken place between the first and last acquisitions. However, the definition of meaningful change varies depending on the application. Changes of interest can be, for example, new buildings and roads, forest fires, and growth or shrinkage of water bodies for environmental monitoring. Although exceptions exist, such as object-based and change vector analysis methods, it is common for change detection algorithms to predict a change label for each pixel in the provided images by modelling the task mathematically as a segmentation or clustering problem.

Many variations of convolutional neural networks (CNNs) (LeCun et al., 1998), notably fully convolutional networks (FCNs) (Long et al., 2015), have recently achieved excellent performances in change detection tasks (Chen et al., 2018; Daudt et al., 2018a, 2019c; Guo et al., 2018). These methods require large amounts of training data to perform supervised training of the proposed networks (LeCun et al., 2015). Unsupervised change detection methods with CNNs also exist, which circumvent the problem of scarce annotations (Alvarez et al., 2020; Luppino et al., 2020; Saha et al., 2019, 2020). Open labelled datasets for change detection are extremely scarce and are predominantly very small compared to labelled datasets in other computer vision areas. Benedek and Szirányi (2009) created the Air Change dataset which contain 13 small annotated images, divided into three regions. Daudt et al. (2018b) created the OSCD dataset from Sentinel-2 multispectral images, with a total of 24 fully annotated image pairs. While these datasets allow for simple models to be trained in a supervised manner, training more complex models with these data would lead to overfitting.

New datasets have recently appeared which change the scale of what is possible for machine learning approaches, but they also raise new issues which are illustrated by the two following examples. The High Resolution Semantic Change Detection (HRSCD) dataset (Daudt et al., 2019c) is a large scale change detection dataset that was generated by combining an aerial image database with open change and land cover data. Change maps and land cover maps were generated for 291 5000 × 5000 RGB image pairs, resulting in over 3000 times more annotated pixels than previous change detection datasets. This dataset, however, contains unreliable labels due to having been generated automatically. The effect of naively using these data for supervised learning of change detection networks is shown in Fig. 1. Inaccuracies in the reference data stem primarily from two causes: imperfections in the vector data at different semantic levels, and temporal misalignment between the annotations and the images. Naive supervision using such data leads to overestimation of the detected changes, as can be seen in Fig. 1e. Nevertheless, there is much useful information in the available annotations that, if used adequately, can lead to better CD systems. Due to the way the ground truth was generated, the labels in the dataset mark changes at a land parcel level with imprecise boundaries. That is often the case when labels are extracted from parcel polygons. While useful for global monitoring of changes in land cover, such labellings often do not delineate precise object-level changes.

Other change detection datasets rely on cross referencing data obtained by on-site surveys with available remote sensing imagery. Such is the case of the ABCD dataset (Fujita et al., 2017), which contains image pairs centered on buildings in a region that has been affected by a tsunami. Images before and after the event were taken with different sensors,



(a) Image 1 **(b)** Image 2 **(c)** Reference data

(d) Manual GT **(e)** Naive **(f)** Proposed

Fig. 1 **a, b** image pair, **c** change labels from the HRSCD dataset, **d** ground truth created by manually annotating changes, **e** result obtained by naive supervised training, **f** result obtained by our proposed method

and were registered and cropped around each known building in the area. Binary change labels for each image pair are available, but segmentation labels are not. This dataset contains over 8000 labelled image pairs, and is available in two versions: *fixed scale*, where the spatial resolution of the images is kept constant, and *resized*, where images are resized so that the length of the imaged building takes up roughly a third of the patch size.

We explore in this paper how to leverage and learn from imprecise or approximate labels for remote sensing image analysis. In particular, we propose the guided anisotropic diffusion (GAD) algorithm for label refinement. GAD is useful to improve the accuracy of prediction boundaries using the input images as guides, which is especially useful in a weakly supervised setting where accurate boundary predictions are often challenging to obtain. We validate its contribution in two weakly supervised learning settings that improve on previously proposed methods for semantic segmentation. First, we use GAD in a training scheme that harnesses the useful information in the HRSCD dataset for parcel-wise change detection, attempting to refine the reference data while training a fully convolutional network. By acknowledging the presence of incorrect labels in the training dataset (with respect to our fine grained objective), we are able to focus on good data and ignore bad ones, improving the final results as seen in Fig. 1f. A preliminary version of this idea has been proposed in Daudt et al. (2019a, 2019b). Second, we also assess GAD in a different task: performing pixel-level damage estimation in the ABCD dataset initially only designed for classification. GAD improves spatial attention weights, which are then used for classification and weakly supervised segmentation of changes. Finally, we evaluate the effectiveness of GAD as a postprocessing algorithm for enhancing the accuracy of region boundaries in semantic segmentation using fully convolutional networks.

2 Related work

Change detection has a long history, being one of the early problems tackled in remote sensing image understanding (Singh, 1989). It is done using coregistered image pairs or sequences, and consists of identifying areas in the images that have experienced significant modifications between the acquisitions. Many of the state-of-the-art ideas in pattern recognition have been used for change detection in the past, from pixel-level comparison of images, to superpixel segmentation, object-level image analysis, and image descriptors (Hussain et al., 2013). In this paper we treat change detection as a two class semantic segmentation problem, in which a label is predicted for each pixel in the input images. With the rise of machine learning algorithms for semantic segmentation, notably convolutional neural networks, many algorithms have attempted to learn to perform change detection. Most algorithms circumvented the problem of the scarcity of training data through transfer learning by using pretrained networks to generate pixel descriptors (El Amin et al., 2016, 2017; Sakurada & Okatani, 2015). Fully convolutional networks trained end-to-end to perform change detection have recently been proposed by several authors independently, usually using Siamese architectures (Chen et al., 2018; Daudt et al., 2018a, 2019c; Guo et al., 2018; Zhan et al., 2017). Unsupervised (Alvarez et al., 2020; Luppino et al., 2020; Saha et al., 2019, 2020) and semi-supervised (Saha et al., 2020) alternatives have also been proposed to cope with the scarcity of accurately labelled data.

Semantic segmentation algorithms attempt to understand an input image and predict to which class among a known set of classes each pixel in an input image belongs. Change detection is modelled in this paper and many others as a semantic segmentation problem which takes as input two or more images. Long et al. (2015) proposed the first fully convolutional network for semantic segmentation, which achieved excellent performance and inference speed. Since then, several improvements have been proposed for CNNs and FCNs. Ioffe and Szegedy (2015) have proposed batch normalization layers, which normalize activations and help avoid the vanishing/exploding gradient problem while training deep networks. Ronneberger et al. (2015) proposed the usage of skip connections that transfer details and boundary information from earlier to later layers in the network, which improves the accuracy around the edges between semantic regions. He et al. (2016) proposed the idea of residual connections, which have improved the performance of CNNs and FCNs and made it easier to train deep networks.

Noisy labels for supervised learning is a topic that has already been widely explored (Frénay et al., 2014; Frénay & Verleysen, 2014). In many cases, label noise is random (by this we mean, following the literature terminology, independent of the data and not correlated) and is modelled mathematically as such (Natarajan et al., 2013; Rolnick et al., 2017; Xiao et al., 2015). Rolnick et al. (2017) showed that supervised learning algorithms are robust to random label noise, and proposed strategies to further minimize the effect label noise has on training, such as increasing the training batch sizes. In the case presented in this paper, the assumption that the label noise is random does not hold. Incorrect change detection labels are usually around edges between regions or grouped together, which leads the network to learn to overestimate detected changes as seen in Fig. 1e. Ignoring part of the training dataset, known as data cleansing (or cleaning), has already been proposed in different contexts (Guyon et al., 1996; Jeatrakul et al., 2010; John, 1995; Matic et al., 1992).

Weakly supervised learning is the name given to the group of machine learning algorithms that aim to perform different or more complex tasks than normally allowed by the training data at hand. Weakly supervised algorithms have recently gained popularity because they provide an alternative when data acquisition is too expensive. The problem of learning to perform semantic segmentation using only bounding box data or image level labels is closely related to the task discussed in this paper, since most methods propose the creation of an approximate semantic segmentation ground truth for training and dealing with its imperfections accordingly. Dai et al. (2015) proposed the BoxSup algorithm where region proposal algorithms are used to generate region candidates in each bounding box, before a semantic segmentation network is trained using these annotations and finally used to iteratively select better region proposal candidates. Khoreva et al. (2017) proposed improvements to the BoxSup algorithm that include using *ad hoc* heuristics and an ignore class during training. They obtained best results using region proposal algorithms to create semantic segmentation training data directly from bounding boxes. Lu et al. (2017) modelled this problem as a simultaneous learning and denoising task through a convex optimization problem. Ahn and Kwak (2018) proposed combining class activation maps, random walk and a learned network that predicts if pixels belong to the same region to perform semantic segmentation from image level labels. Zhou et al. (2016) proposed the class activation mapping technique, which allows the networks to localize what regions in the image contribute to the prediction of each class, which can be harnessed for generating pixel-level predictions from image-level labels.

Post-processing methods that use information from guide images to filter other images, such as semantic segmentation results, have also been proposed (Ferstl et al., 2013; Kopf et al., 2007; Petschnigg et al., 2004). A notable example is the Dense CRF algorithm proposed by Krähenbühl and Koltun (2011), in which an efficient solver is proposed for fully connected conditional random fields with Gaussian edge potentials. The idea of using a guide image for processing another is also the base of the Guided Image Filtering algorithm proposed by He et al. (2013), where a linear model that transforms a guide image into the best approximation of the filtered image is calculated, thus transferring details from the guide image to the filtered image. The use of joint filtering is popular in the field of computational photography, and has been used for several applications (Ferstl et al., 2013; Kopf et al., 2007; Petschnigg et al., 2004). One of the building blocks of the filtering method we propose in this paper is the anisotropic diffusion, proposed by Perona and Malik (1990), an edge preserving filtering algorithm in which the filtering of an image is modelled as a heat equation with a different diffusion coefficient at each edge between neighbouring pixels depending on the local geometry and contrast. However, to the best of our knowledge, this algorithm has not yet been used for guided filtering.

3 Method

The main contributions of this paper are: (1) the guided anisotropic diffusion algorithm, which uses information from the input images to filter and improve semantic segmentation results (Sect. 3.1), (2) an iterative training scheme that aims to efficiently learn from inaccurate and unreliable ground truth semantic segmentation data (Sect. 3.2), and (3) a learned spatial attention layer that improves classification and weakly supervised semantic segmentation for datasets whose images have been cropped using the geographical coordinates of objects of interest (Sect. 3.3). These contributions are described in detail below. While these ideas are presented in this paper in the context of change detection, the proposed methods' scope is broader and they could be used for other semantic segmentation problems, as we show in Sects. 3.4 and 4.3.

3.1 Guided anisotropic diffusion

In their seminal paper, Perona and Malik proposed an anisotropic diffusion algorithm with the aim of performing scale space image analysis and edge preserving filtering (Perona and Malik 1990). Their diffusion scheme has the ability to blur the inside of homogeneous regions while preserving or even enhancing edges. This is done by modelling the filtering as a diffusion equation with spatially variable coefficients. The corresponding equation is an extension of the linear heat equation, whose solution is mathematically equivalent to Gaussian filtering when diffusion coefficients are constant (Koenderink 1984). Diffusion coefficients are set to be higher where the local contrast of the image is lower.

More precisely, we consider the anisotropic diffusion equation

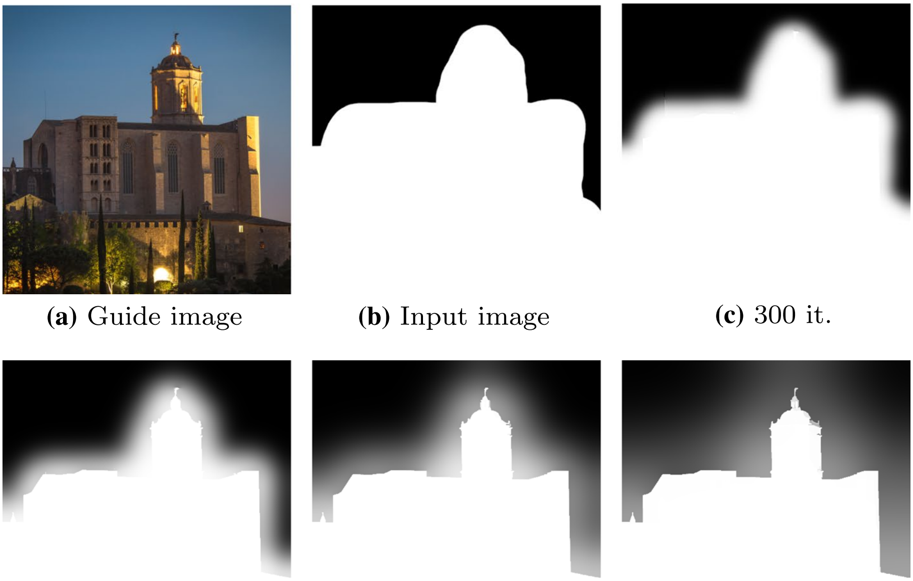
$$\frac{\partial I}{\partial t} = div(c(x, y, t)\nabla I) = c(x, y, t)\Delta I + \nabla c \cdot \nabla I \tag{1}$$

where I is the input image, $c(x, y, t)$ is the coefficient diffusion at position (x, y) and time t, div represents the divergence, ∇ represents the gradient, and Δ represents the Laplacian. In its original formulation, $c(x, y, t)$ is a function of the input image I. To perform edge preserving filtering, one approach is using the coefficient

$$c(x, y, t) = \frac{1}{1 + \left(\frac{||\nabla I(x,y,t)||}{K}\right)^2}, \tag{2}$$

which approaches 1 (strong diffusion) where the gradient is small, and approaches 0 (weak diffusion) for large gradient values. Other functions with these properties and bound in [0, 1] may also be used. The parameter K controls the sensitivity to contrast in the image.

In the *guided* anisotropic diffusion (GAD) algorithm, the aim is to perform edge preserving filtering on an input image, but instead of preserving the edges in the filtered image we preserve edges coming from a separate guide image (or images). Doing so allows us to transfer properties from the guide image I_g onto I_{input}, producing the filtered image I_f. An illustrative example is shown in Fig. 2, where the image of a cathedral (a) is used as a guide to filter the image of a rough segmentation (b). The edges from the guide image I_g are used to calculate $c(x, y, t)$, which in practice creates barriers in the diffusion of the filtered image I_f, effectively transferring details from I_g to I_f. These edges effectively separate the image in two regions, inside and outside the region of interest, and the pixel values in each of these regions experience diffusion, but there is virtually no diffusion happening between them.



(a) Guide image **(b)** Input image **(c)** 300 it.

(d) 1000 it. **(e)** 3000 it. **(f)** 10000 it.

Fig. 2 Results of guided anisotropic diffusion. Edges in the guide image (**a**) are preserved in the filtered image (**b**). **c–f** show results using different numbers of iterations

Algorithm 1 Guided Anisotropic Diffusion pseudocode.

```
Input: $I_{g,1}$, $I_{g,2}$, $I_{input}$, $N$, $K$, $\lambda$
Output: $I_f$
$I_f \leftarrow I_{input}$
for ($i \leftarrow 1; i \leq N; i++$) do
    for ($I_j = \{I_1, I_2\}$) do
        $\nabla I_j \leftarrow$ Calculate gradient of $I_j$
        $c_{I_j} \leftarrow$ Calculate diffusion coefficients using Eq. 3
        $I_j \leftarrow I_j + \lambda \cdot \nabla I_j \cdot c_{I_j}$
    end for
    $\nabla I_f \leftarrow$ Calculate gradient of $I_f$
    $c_f \leftarrow$ Calculate diffusion coefficients using Eq. 4
    $I_f \leftarrow I_f + \lambda \cdot \nabla I_f \cdot c_f$
end for
```

Our primary aim is to use this GAD algorithm to improve semantic segmentation results based on the input images. Weakly supervised learning methods are often used when there is an overestimation or underestimation of the target area: either the whole image is the starting point in classification to segmentation tasks, or the reference region is too large in parcel to region segmentation tasks, or a subset of pixels (points or squiggles) are used for supervision.

GAD provides a way to improve these semantic segmentation results by making them more precisely fit the edges present in the input images. A few design choices were made to extend the anisotropic diffusion from gray level images to RGB image pairs. The extension to RGB images was done by taking the mean of the gradient norm at each location

$$c_{I_g}(x, y, t) = \frac{1}{1 + \left(\sum_{C\in\{R,G,B\}} \frac{||\nabla I_{g,C}(x,y,t)||}{3\cdot K} \right)^2}, \tag{3}$$

so that edges in any of the color channels would prevent diffusion in the filtered image. To extend this further to be capable of taking multiple guide images simultaneously, which is necessary for the problem of change detection, the minimum diffusion coefficient at each position (x, y, t) was used, once again to ensure that any edge present in any guide image would be transferred to the filtered image:

$$c_{I_{g,1},I_{g,2}}(x, y, t) = min_{i\in\{1,2\}} c_{I_i}(x, y, t). \tag{4}$$

Guided anisotropic diffusion aims to improve semantic segmentation predictions by filtering the class probabilities yielded by a fully convolutional network. It is less adequate to correct for large classification mistakes, as opposed to non-local methods such as Dense CRF, but it leads to smoother predictions with more accurate edges. It can also be easily extended for any number of guide images by increasing the number of images considered in Eq. (4). The pseudocode for the GAD algorithm can be found in Algorithm 1. As mentioned in the original anisotropic diffusion paper, the algorithm is unstable for $\lambda > 0.25$ when using 4-neighborhoods for the calculations. For more information the reader can refer to the mathematical derivations presented in Aubert and Kornprobst (2006) and Perona and Malik (1990).

GAD parameters are tuned visually by performing anisotropic diffusion on guide images from the dataset. Each parameter offers different trade-offs:

- K allows us to choose the magnitude of gradients that should be considered as edges (and therefore diffusion should be restricted at that point).
- A larger number of iterations N allows for the diffusion to mix pixel values at longer ranges (the "receptive field" radius is equal to the number if iterations).
- If λ is set closer to 0 the algorithm approaches the continuous time solution of the equations, at the cost of diffusion speed. This has not been observed to improve results as long as λ is kept below the threshold of stability at $\lambda = 0.25$.

In the following sections, we show two ways to use GAD in approaches to learn to perform semantic segmentation with imperfect labels. First, we address in Sect. 3.2 the inaccurate labelling problem with an iterative data cleansing scheme. Second, in Sect. 3.3 we explore another use-case and use GAD to learn to segment changes from classification labels only.

3.2 Iterative training scheme

The first use-case we investigate for deploying GAD tackles the type of label noise present in parcel-based change detection datasets where pixel labels are generated from vector data . It is challenging due to its spatial structure and correlation between neighbors. In the taxonomy presented in Frénay et al. (2014) and Frénay and Verleysen (2014), this type

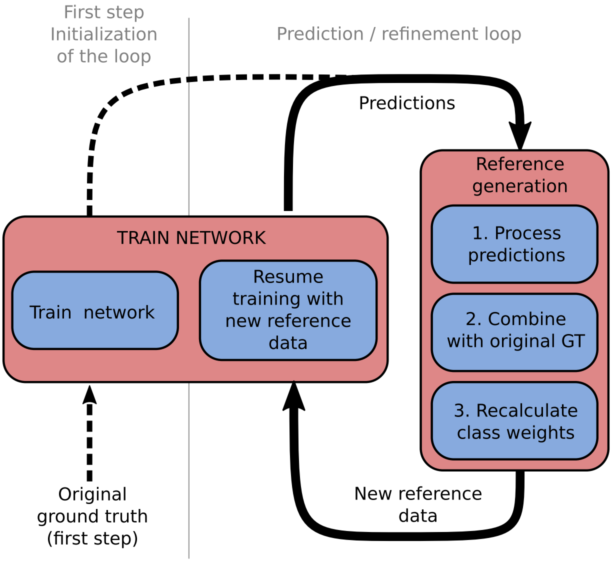


Fig. 3 Iterative training method: alternating between training and data cleaning allows the network to simultaneously learn the desired task and to remove bad examples from the training dataset

of label noise would be classified as "noisy not at random" (NNAR). NNAR is the most complex among the label noise models in the taxonomy. This is the classification applied to noise when the samples that are mislabelled are not randomly dispersed, and the type of noise is also not random. For labels generated from parcel polygons, label noise will be concentrated around region boundaries and the type of noise will be defined by the classes of the imaged objects. In the case of HRSCD, most errors can be attributed to one of the following reasons: the available information is insufficient to perform labelling, errors on the part of the annotators, subjectiveness of the labelling task, and temporal misalignment between the databases used to create the HRSCD dataset.

It is important to note that, as discussed by Frénay et al. (2014), label noise has an even more powerful damaging impact when a dataset is imbalanced since it alters the perceived, but not the real, class imbalance and therefore the methods used to mitigate class imbalance during training are less effective. In the case of change detection with the HRSCD dataset, the no change class outnumbers the change class 130 to 1, which means the label noise could significantly alter the calculated class weights used for training.

It has been noticed in Daudt et al. (2019c) and in the experiments presented in this paper that change detection networks trained directly on the HRSCD dataset had the capacity to detect changes in image pairs but tended to predict blobs around the detected change instances, as is depicted in Fig. 8c, likely in an attempt to minimize the loss for the training images where the surrounding pixels of true changes are also marked as having experienced changes. In many cases, it was observed that the network predictions were correct where the ground truth labels were not. Based on this observation, we propose a method for training the network that alternates between actual minimization of a loss function and using the network predictions to clean the reference data before continuing the training. A schematic that illustrates the main ideas of this method is shown in Fig. 3. For the remainder of this paper, the iteration cycles of training the network and cleaning of training data will be referred to as *hyperepochs*.

Algorithm 2 Iterative training pseudocode.

```
Input: $\mathcal{I}$: Image pairs, $GT_o$: Original unreliable ground truths, $N$: Number of hyperepochs,
$\Phi_r$: Initial random network weights.
Output: $\Phi_N$: Trained network weights.
$w_0 \leftarrow$ calculate class weights inversely proportional to number of class examples
$\Phi_0 \leftarrow$ Train network with $\mathcal{I}$ and $GT_0$ until convergence or fixed number of epochs
for $(i \leftarrow 1; i \leq N; i++)$ do
    $P_i \leftarrow$ generate predictions for training dataset with current network
    $P_{i,pp} \leftarrow$ Post-processing of predictions
    $GT_i \leftarrow$ Combine $P_{i,pp}$ with $GT_0$ to generate cleaner ground truth data
    $\Phi_i \leftarrow$ Continue training network from $\Phi_{i-1}$ using $\mathcal{I}$ and $GT_i$ until convergence
end for
```

Alternating between training a semantic segmentation network and using it to make changes to the training data has already been explored (Dai et al., 2015; Khoreva et al., 2017). Such iterative methods are named "classification filtering" (Frénay & Verleysen, 2014). The main differences between the method proposed in this paper and previous ones are:

1. *No bounding box information is available*: we work directly with pixel level annotations, which were generated form vector data;
2. *Each annotated region may contain more than one instance*: the annotations often group several change instances together;
3. *Annotations are not flawless*: the HRSCD dataset contains both false positives and false negatives in change annotations.

It has also been shown by Khoreva et al. (2017) that simply using the outputs of the network as training data leads to degradation of the results, and that it is necessary to use priors and heuristics specific to the problem at hand to prevent a degradation in performance. In this paper we use two ways to avoid degradation of the results with iterative training. The first is using processing techniques that bring information from the input images into the predicted semantic segmentations, improving the results and providing a stronger correlation between inputs and predictions. The GAD algorithm presented in Sect. 3.1 serves this purpose, but other algorithms such as Dense CRF (Krähenbühl & Koltun, 2011) may also be used. The second way the degradation of results is avoided is by combining network predictions with the original reference data at each iteration, instead of simply using predictions as reference data.

We propose three ways of merging the original labels with network predictions. When merging, each pixel will have a binary label from the original ground truth and a binary label from the network prediction. If these labels agree, there is no reason to believe the label for that pixel is wrong, and it is therefore kept unchanged. In case the labels disagree, the following options to decide the pixel's label are proposed:

1. *The intersection of predicted and reference change labels is kept as change*: this strategy assumes all changes are marked in both the reference data and in the prediction. It also

		Original GT	
		0	**1**
Pred.	**0**	0	0
	1	0	1

(a) Intersection

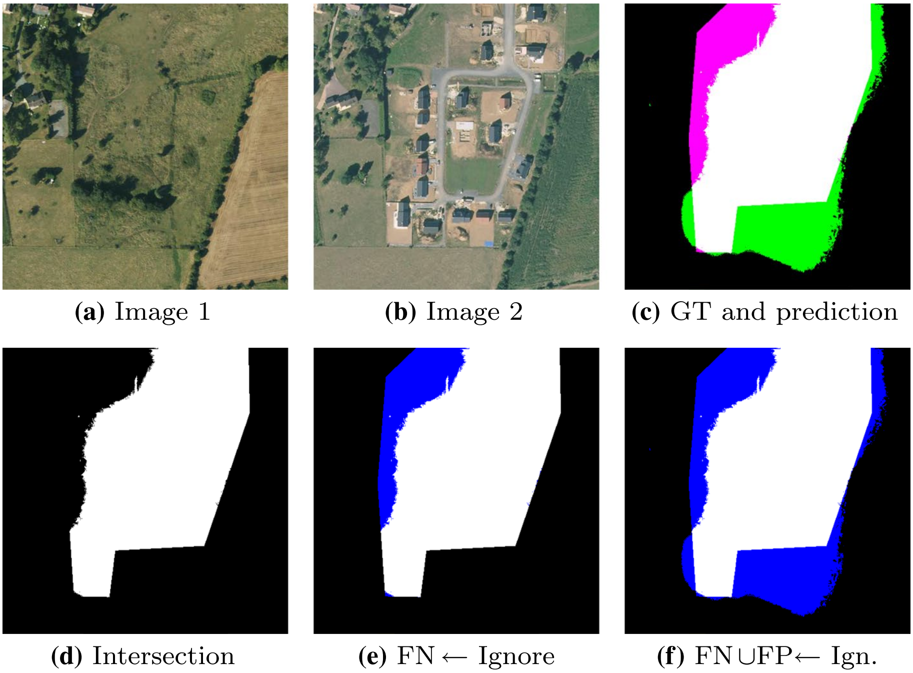
		Original GT	
		0	**1**
Pred.	**0**	0	2
	1	0	1

(b) FN← Ignore

		Original GT	
		0	**1**
Pred.	**0**	0	2
	1	2	1

(c) FN∪FP← Ignore

Fig. 4 Proposed methods for merging original labels and network predictions. Classes: 0 is no change, 1 is change, 2 is ignore. **a** Intersection between original and detected changes. **b** Ignore false negatives from the perspective of original labels. **c** Ignore all pixels with label disagreements



(a) Image 1

(b) Image 2

(c) GT and prediction

(d) Intersection

(e) FN ← Ignore

(f) FN∪FP← Ign.

Fig. 5 Example case of the three proposed merge strategies. In **c**, black is true negative, white is true positive, magenta is false negative, and green is false positive. In **d–f** blue represents the ignore class

puts pixels with uncertain labels in the no change class, where they are more easily diluted during training due to the class imbalance.

2. *Ignore false negatives*: using an ignore class for false negatives attempts to keep only good examples in the change class, improving the quality of the training data. It assumes all changes are marked in the original labels provided.
3. *Ignore all disagreements*: marking all label disagreements to be ignored during training attempts to keep only clean labels for training at the cost of reducing the number of training examples. This approach is the only one that is class agnostic.

In practice, the ignored pixels are marked as a different class that is given a class weight of 0 during the training. Tables for the three proposed methods can be found in Fig. 4, and an example can be found in Fig. 5.

3.3 Scene-invariant spatial attention layer

Our second change detection use-case for GAD-enhanced weak supervision addresses the more challenging task of inferring segmentation masks from image-level classification labels. Many datasets in remote sensing contain georeferenced data, such as patches cropped from large images using the coordinates of known objects for which a label is known. In such cases, objects to which the labels refer are located in the center of the images, while the characteristics of their surroundings are not directly related to the available labels. Pooling techniques such as max pooling and average pooling that are very often used in CNNs are invariant with respect to the image position. These operations fail to make use of the heuristic described above, and do not learn to prioritize some areas of the image over others when making classification predictions.

To increase the localization capability through the global average pooling operation, we propose here a learned spatial attention layer that can be used to allow the network to learn which positions of the images are more discriminative and should be prioritized over others when making inferences. We also propose to use the GAD algorithm to further focus the attention of the network on the most relevant features. Let's assume that a feature map x of size $C \times M \times N$ is obtained after any number of convolution, pooling and other operations from an input image (or images in the case of change detection), where C is the number of channels and M and N are spatial dimensions. We define a matrix A of size $M \times N$ which will be learned by the network, and will serve as attention weights given to spatial locations. The attention operation $f(x)$ can then be defined as

$$f(x)_{c,i,j} = \alpha \cdot x_{c,i,j} \cdot \sigma(a_{i,j}), \tag{5}$$

where $a_{i,j}$ denotes the element of A in position (i, j), σ denotes the sigmoid function and α is a normalization term defined as

$$\alpha = \sum_{i=1}^{M} \sum_{j=1}^{N} \sigma(a_{i,j}). \tag{6}$$

The sigmoid function is used to ensure the attention weights given to each spatial location is in the range (0, 1). The matrix A is initialized as a null matrix so that all spatial locations have equal attention values of $\sigma(a_{i,j}) = 0.5$. Random initialization of A is neither necessary nor recommended.

The proposed attention layer is designed to be used after a softmax operation and before a global average pooling (GAP) layer. Doing so will force the network to produce per-pixel classification predictions, which are then put through a weighted average operation whose weights are learnable parameters which depend only on the spatial position of each feature. Global average pooling is preferable to max pooling at the end of a network when we want the network to be able to localize objects, as was discussed in Zhou et al. (2016). Note that the number of learnable parameters introduced by this attention layer is only $M \cdot N$, which is extremely small in the context of deep neural networks. At inference time, the learned attention weights can be further adapted to the input images by using the GAD algorithm

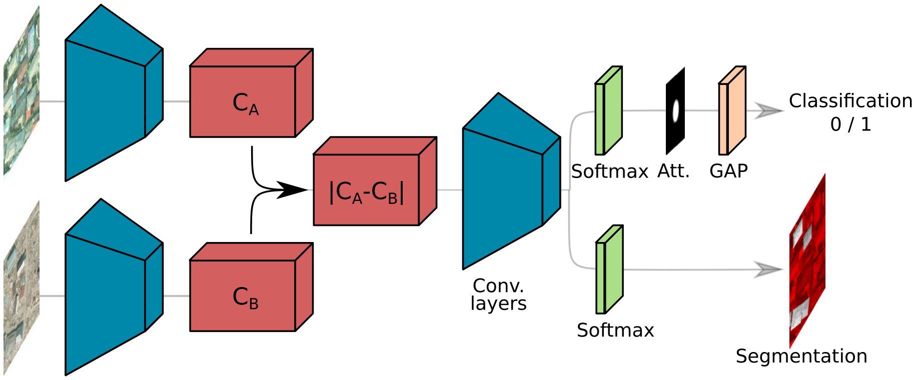


Fig. 6 Basic schematic of the network used for weakly supervised change detection. Two paths can be taken: the classification path uses the proposed attention layer and global average pooling to produce a classification of the image, while the segmentation path avoids these steps to output pixel-level predictions. Supervision is only available on the classification path

proposed in Sect. 3.1. This helps the network to focus its attention at the building at the center of the image pair, further increasing classification performance.

In this paper, we incorporate this attention layer into the classification branch of the architecture depicted in Fig. 6. The images are processed by two convolutions with stride $\frac{1}{2}$ and 5 residual blocks before their features are merged, and 4 residual blocks after. This architecture allows us to perform either classification or segmentation by choosing either of the paths at the end. This architecture is a straightforward Siamese extension of the ideas presented in Zhou et al. (2016). Supervision is only available for the classification branch, but the structure of the network allows us to apply equivalent classification operations at each spatial locations by avoiding the attention and global average pooling layers, effectively performing semantic segmentation.

3.4 Edge enhancement using guided anisotropic diffusion for segmentation upsampling

We further propose an experiment to evaluate the efficacy of the proposed guided anisotropic diffusion algorithm in a more general semantic segmentation setting. In this case, we attempt to compensate for when the network is supervised with images with a lower ground sample distance (GSD) when it is applied to images with a higher GSD. In this case, a network that is supervised with images at a lower resolution struggles to predict region boundaries at higher resolutions.

To apply a CNN trained with images with a smaller GSD to images with higher GSDs, it is first necessary to downsample the higher resolution images. This is necessary because the network has learned to detect objects at a given scale, and changing the scale of the input images would pose additional problems due to the shift in dataset statistics. The downsampled image is then segmented using the fully convolutional network, which produces softmax activations, i.e. class probabilities, for each pixel. To bring these low resolution predictions back to the higher GSD, these class probabilities are upsampled back to the original resolution. This procedure leads to semantic region boundaries not being very

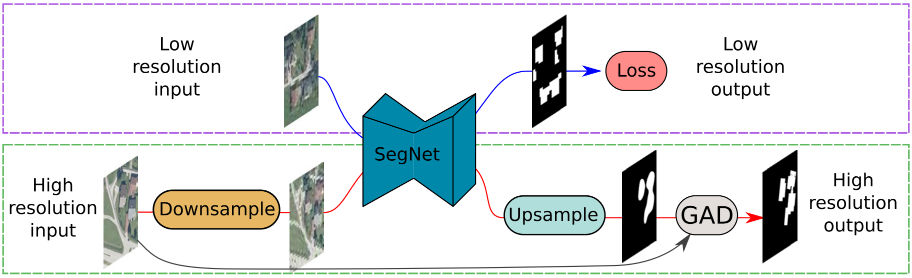


Fig. 7 Schematic for the experiments performed to evaluate if GAD is able to compensate for supervision using images at a lower resolution and improve prediction accuracy around region boundaries

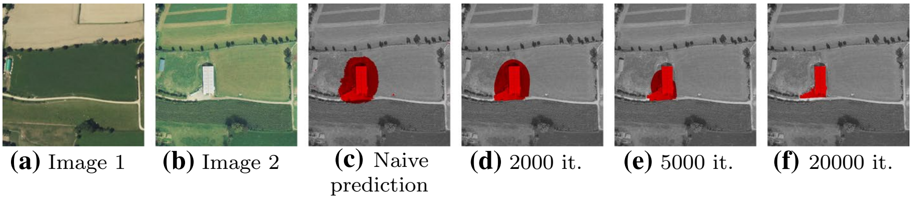

(a) Image 1 **(b)** Image 2 **(c)** Naive prediction **(d)** 2000 it. **(e)** 5000 it. **(f)** 20000 it.

Fig. 8 Guided anisotropic diffusion for filtering a real example of semantic segmentation. The diffusion allows edges from the guide images to be transferred to the target image, improving the results (change area in red)

accurate. In our experiments, we use the proposed guided anisotropic diffusion algorithm to recover boundary information using the high resolution as a guide for where the semantic boundaries should be located. A diagram that illustrates this procedure can be found in Fig. 7.

These experiments illustrate how GAD could be used to help networks adapt to new applications with different data. Resolution differences is very common in remote sensing due to images coming from different satellite or aerial sources, and thus coping with such variations is important.

4 Experiments

The experiments presented in this paper have been divided into three sections. The first one, in Sect. 4.1, explores how the integration of GAD in the iterative training scheme of Sect. 3.2 allows us to refine approximate labels to obtain pixel-level change detection more accurately than through direct supervision. The second one, in Sect. 4.2, shows the effectiveness of GAD combined with a spatial attention layer (Sect. 3.3) in performing weakly supervised change detection using only image-level labels to perform pixel-level predictions. Finally, we evaluate in Sect. 4.3 the usage of GAD for adapting to multiple spatial resolutions for building and generic segmentation tasks, as defined in Sect. 3.4.

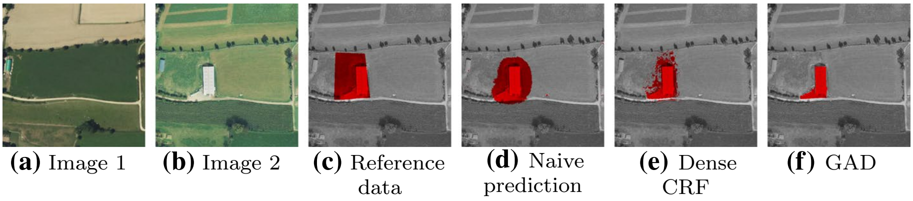


Fig. 9 Comparison between **c** original dataset ground truth, **e** prediction filtered by Dense CRF, and **f** prediction filtered with guided anisotropic diffusion for 20,000 iterations. (changes in red)

4.1 Label refinement through iterative learning

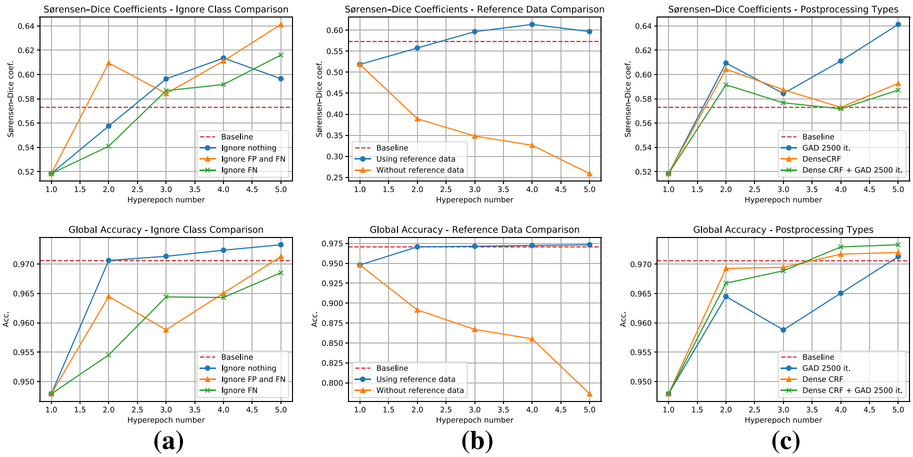
To validate the iterative training scheme proposed in Sect. 3.2 we adopted the hybrid change detection and land cover mapping fully convolutional network presented in Daudt et al. (2019c), since it was already proven to work with the HRSCD dataset. We adopted *strategy 4.2* described in Daudt et al. (2019c), in which the land cover mapping branches of the network are trained before the change detection one to avoid setting a balancing hyperparameter. The land cover mapping branches of the network were fixed to have the same parameter weights for all tests presented in this paper, and evaluating those results is not done here as the scope of this paper is restricted to the problem of change detection.

We applied the GAD algorithm to the predictions from a network trained directly on the reference data from HRSCD to evaluate its performance. In Fig. 8 is displayed an example of the obtained results. As noted before, we can see in (c) that the change is detected but that unchanged pixels around it are also classified as changes by the network. In (d)–(f) it can be clearly seen how the GAD algorithm improves the results by diffusing the labels across similar pixels while preserving edges from the input images in the semantic segmentation results. As expected, more iterations of the algorithm lead to a stronger erosion of incorrect labels. For these results, GAD was applied with $K = 5$ and $\lambda = 0.24$. In Fig. 9 we can see a comparison between GAD and the Dense CRF[1] algorithm (Krähenbühl and Koltun 2011). While the non-local nature of fully connected CRFs is useful in some cases, we can see the results are less precise and significantly noisier than the ones obtained by using GAD.

To perform quantitative analysis of results, it would be meaningless to use the test data in the HRSCD dataset. Using a GeForce GTX 1060 GPU, applying GAD to a 512×512 image for 100 iterations took approximately 230 ms. Indeed, we are attempting to perform a task which is not the one for which ground truth data are available since *i.e.* we are attempting to perform pixel-level precise change detection and not parcel-level change detection. For this reason we have manually annotated the changes as precisely as possible for two $10{,}000 \times 10{,}000$ image pairs in the dataset, for a total of $2{\cdot}10^8$ test pixels, or 50 km^2. The image pairs were chosen before any tests were made to avoid biasing the results. Due to the class imbalance, total accuracy, *i.e.* the percentage of correctly classified pixels, provides us with a skewed view of the results biased towards the performance on the class more strongly represented. Therefore, the Sørensen-Dice coefficient (equivalent to the F1 score for binary problems) from the point of view of the change class was used (Dice, 1945; Sørensen, 1948). The Sørensen-Dice coefficient score is defined as

$$Dice = (2 \cdot TP)/(2 \cdot TP + FP + FN) \tag{7}$$

[1] https://github.com/lucasb-eyer/pydensecrf.



(a) (b) (c)

Fig. 10 Ablation studies. **a** Comparison between strategies for merging network predictions and reference data. **b** Comparison between iterative training with and without the usage of original reference data. **c** Comparison between GAD and Dense CRF. Top row contains Dice scores, bottom row contains global accuracy curves

where TP means true positive, FP means false positive, and FN means false negative. It serves as a balanced measurement of performance even for unbalanced data.

All tests presented here were done using PyTorch (Paszke et al., 2017). At each hyperepoch, the network was trained for 100 epochs with an ADAM algorithm for stochastic optimization (Kingma & Ba, 2014), with learning rate of 10^{-3} for the first 75 epochs and 10^{-4} for the other 25 epochs. The tests show the performance of networks trained with the proposed method for 5 hyperepochs (iterations of training and cleaning the data), where the first one is done directly on the available data from the HRSCD dataset. For accurate comparison of methods and to minimize the randomness in the comparisons, the obtained network at the end of hyperepoch 1 is used as a starting point for all the methods. This ensures all networks have the same initialization at the point in the algorithm where they diverge. A baseline network was trained for the same amount of epochs and hyperepochs but with no changes done to the training data. This serves as a reference point as to the performance of the fully convolutional network with no weakly supervised training methods.

The first comparison, shown in Fig. 10a, compares the three methods proposed in Sect. 3.2 to combine the network predictions with the original ground truth from the HRSCD dataset. We notice that all three strategies surpass the baseline network using the proposed iterative training method, which validates the ideas presented earlier. In Fig. 10b we see a comparison between a training using the full training scheme proposed in this paper (without the usage of an ignore class) and the same method but without using the original reference data, *i.e.* using only network predictions processed by GAD to continue training at each hyperepoch. Our results, which corroborate the ones in Khoreva et al. (2017), show that referring back to the original data at each hyperepoch is essential to avoid a degradation in performance.

In Fig. 10c we show a comparison between using the proposed GAD algorithm versus the Dense CRF (Krähenbühl & Koltun, 2011) algorithm in the iterated training procedure, as well as using both together. We see that using the Dense CRF algorithm to process predictions leads to good performance in early hyperepochs, but is surpassed by GAD later

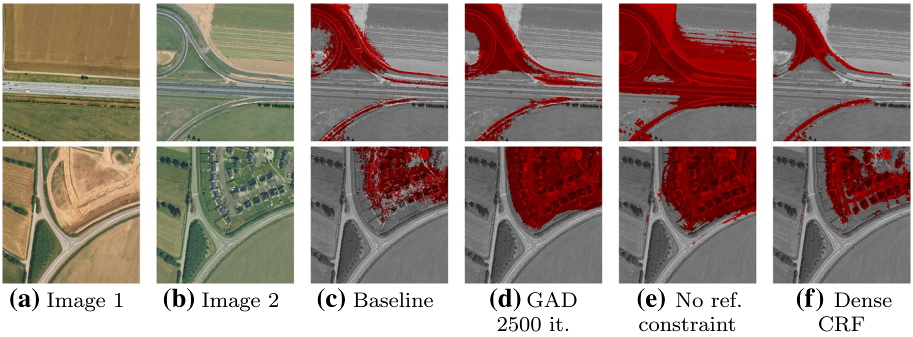


Fig. 11 Change maps obtained by using different methods on two image pairs. Detected changes are marked in red color

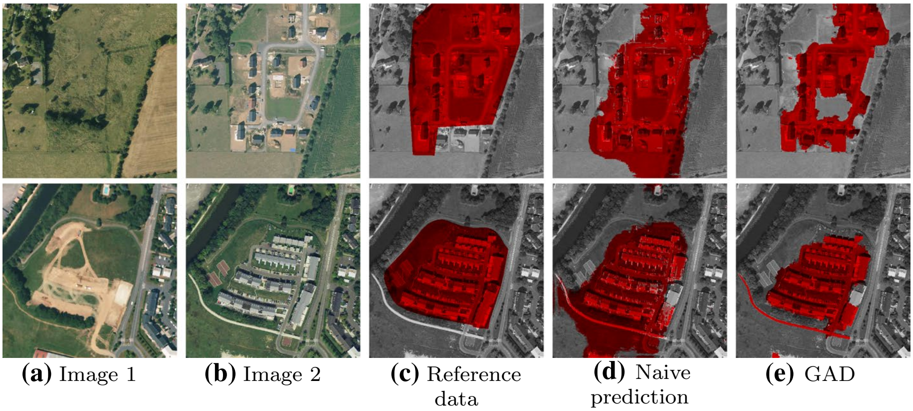


Fig. 12 Results using the complete inference pipeline (changes in red). GAD is used to improve predictions during the iterative training process as well as for improving the final segmentations

on. This is likely explained by the non local nature of Dense CRF and its ability to deal with larger errors, but its inferior performance relative to GAD for finer prediction errors.

Figure 11 shows the predictions by networks trained by different methods on two example images. We see that the best results are obtained by using the full training scheme with GAD in (d)/(j), followed by Dense CRF, which also achieves good results shown in (f)/(l). The baseline results in (c)/(i), obtained by naively training the network in a supervised manner, and the ones without using the reference data as constraint in the iterative training scheme shown in (e)/(k) are significantly less accurate than those using GAD or Dense CRF. The final change maps that were produced by the proposed method for two test cases can be seen in Fig. 12.

4.2 Scene-invariant spatial attention layer

We tested the proposed method using the ABCD dataset proposed by Fujita et al. (2017). This dataset contains pairs of crops of images centered on buildings that have been surveyed to evaluate their destruction after a tsunami. We have followed the 5-fold cross

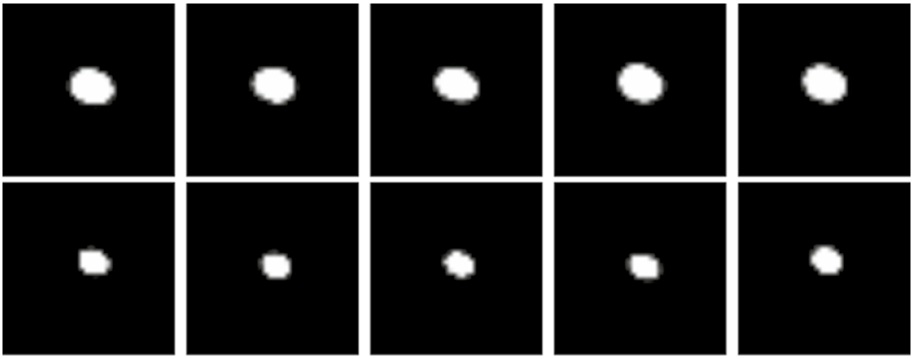


Fig. 13 Spatial attention weights that were learned in each of the cross-validation tests. Top row contains all 5 tests using fixed scale ABCD dataset, bottom row are the results using the rescaled version of the dataset. Note that the network was incredibly consistent in identifying the center of the images as most discriminative without any explicit knowledge. These attention matrices are of size 40 × 40

Table 1 Accuracy and standard deviation for each test on ABCD dataset using 5-fold cross validation

Method	Fixed scale	Resized
6-ch (Fujita et al., 2017)	94.5 ± 0.5	94.7 ± 0.3
siam (Fujita et al., 2017)	94.8 ± 0.3	94.9 ± 0.4
No attention	89.33 ± 0.79	90.96 ± 0.65
Attention	94.36 ± 0.26	94.88 ± 0.18
Attention + GAD	94.58 ± 0.27	94.90 ± 0.22

Fixed scale and resized variations of the ABCD dataset were tested. Results from methods proposed by Fujita et al. are included for comparison

validation that was defined by the dataset's creators. All networks were trained from scratch using only the ABCD dataset, using an initial learning rate of 0.005 for 10 epochs, then with a linearly decaying learning rate for 90 epochs for a total of 100 epochs. The classification results for these tests are presented in Table 1. These results show that our network with the attention module performed very similarly to the ones presented in Fujita et al. (2017). It is also clear that the proposed attention module improved the classification accuracy of the networks significantly. The obtained results also show that filtering the attention weights using the GAD algorithm further increases the classification performance of the proposed network, improving the quality of the attention weights by using the input images as guides.

Figure 13 show the learned spatial attention weights learned in each of the performed tests. We can clearly see how consistent the network was in identifying that the most discriminative region of the images was located in the center. It is also apparent that the network identified that the scale of this discriminative region is larger in the *resized* version of the dataset compared to the *fixed-scale* version.

Qualitative analysis of segmentation results show that the usage of the proposed spatial attention operation allowed the network to vastly increase its capacity to localize features in the input images, which led to much more accurate segmentation, as depicted in Fig. 14. The results also show how using the GAD algorithm for post-processing further increased

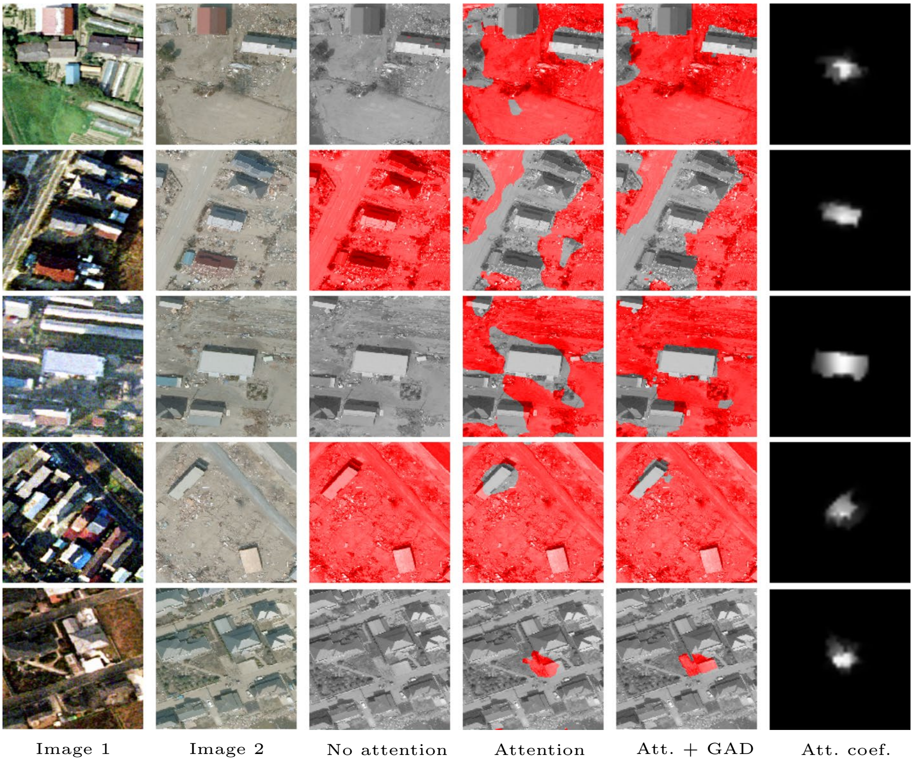


Fig. 14 Results obtained by using the proposed method. Note that when the attention layer is not used, the network does not learn to localize the features and tends to predict all pixels into the same class. The attention layer enables the network to localize features much more accurately, and the GAD post-processing further increases the spatial accuracy of such predictions. Changes are marked in red

the spatial accuracy of the segmentation results. In these images, the application of our algorithm without an attention layer or GAD (column "No attention") can be seen as a simple Siamese extension of the CAM technique for handling two input images (Zhou et al., 2016)

These results suggest that there is a positive feedback loop that happens during the training process between the network's ability to localize discriminative features and the spatial attention operation. Once the network develops the ability to roughly localize discriminative features, this allows the training of the spatial attention layer, which leads the network to learn even more local features, and so on.

Two notable examples can be seen in Fig. 14. The first one is the example in the fourth row, which shows that the network is not simply finding buildings in the second image and marking those as unchanged. In this example, a building is present in the second image but it is marked as a change nonetheless since it doesn't match the buildings in the first image. The second notable example is the one showed in the last row, where a very small change was detected in the center of the image, surrounded only by unchanged buildings. Since the

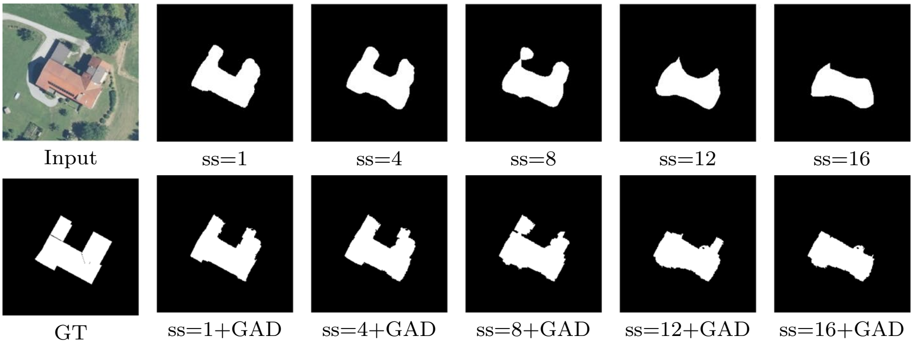


Fig. 15 Results obtained from experiments on the Inria Aerial Image Labeling Dataset. GAD successfully mitigated the accuracy loss in region boundaries at higher subsampling rates

position of this detected change coincided to the spatial attention position, the network was able to mark this image pair as a change, which is correct according to the ground truth label. The same was not accomplished by the network without the attention layer.

4.3 Edge enhancement for segmentation upsampling

To further validate the effectiveness of GAD as a postprocessing tool in a more general setting, we perform the experiments described in Sect. 3.4, where we study how GAD can be used for edge enhancement for upsampled softmax activations. To simulate a setting where data with different resolutions are available, we use subsampled images for training the network and study their predictions following the steps depicted in Fig. 7.

We performed these experiments using two datasets. The first one is the *Inria Aerial Image Labeling Dataset* (Maggiori et al., 2017), which contain RGB images of several urban areas in different countries and environments at a spatial resolution of 0.3 m/px, along with binary pixel-level labels that indicate the presence of buildings. The train/validation split that was proposed by the dataset creators (i.e. keeping the first five images for each location for validation) was used, which results in 155 images for training and 25 images for validation, all of size 5000×5000 pixels. The second dataset used for testing this approach was the *Vaihingen Dataset*,[2] which contains false color images for the urban area of Vaihingen at a spatial resolution of 0.09 m/px, as well as pixel-level semantic segmentation labels. We followed the train/validation split proposed by Audebert et al. (2017). The code for this work[3] was also used with the appropriate modification to perform the subsampling experiments. The standard SegNet architecture was used for all the experiments in this section (Badrinarayanan et al., 2017). The parameters for the GAD algorithms were tuned visually using a few example images before being applied to the validation dataset. For the experiments presented in this section, the parameters that were used were: $\lambda = 0.24$, $N = 1000$, and $K = 0.002$ (for images normalized between 0 and 1).

Qualitative and quantitative results for the experiments performed on the Inria dataset can be seen in Figs. 15 and 16, respectively. Figure 15 clearly shows the efficacy of GAD

[2] https://www2.isprs.org/commissions/comm2/wg4/benchmark/2d-sem-label-vaihingen/.

[3] https://github.com/nshaud/DeepNetsForEO.

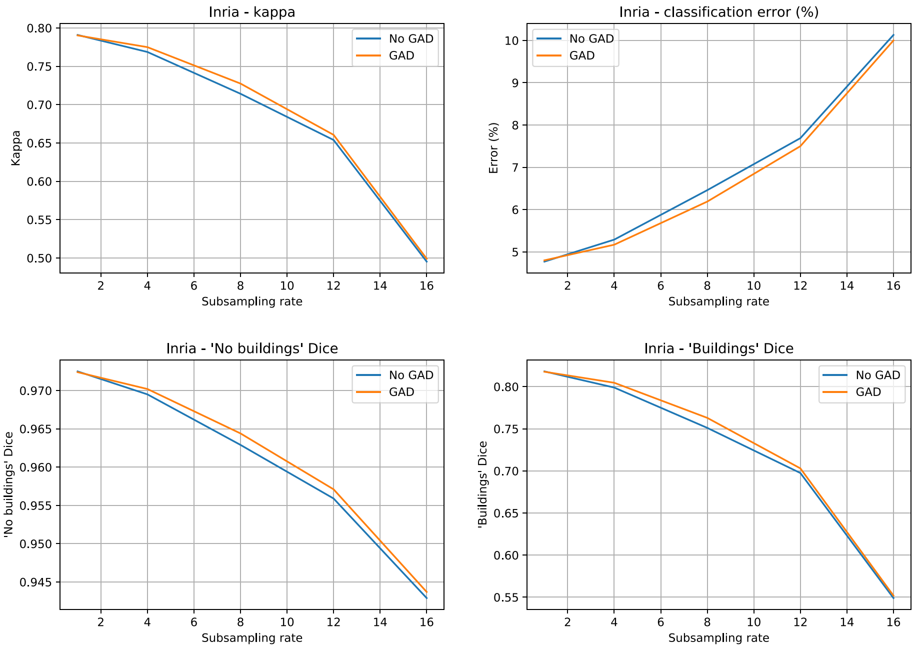
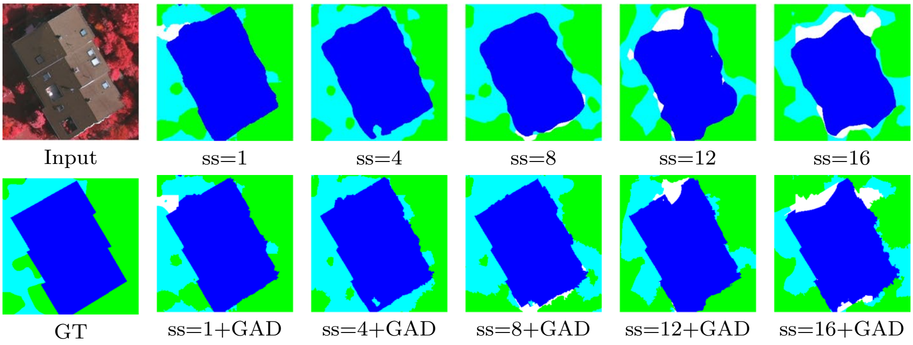


Fig. 16 Results for improving segmentation boundaries on the Inria Aerial Image Labeling Dataset. Consistent improvements across all metrics have been observed for all subsampling rates bigger than 1


Input ss=1 ss=4 ss=8 ss=12 ss=16

GT ss=1+GAD ss=4+GAD ss=8+GAD ss=12+GAD ss=16+GAD

Fig. 17 Results obtained from experiments on Vaihingen dataset. GAD performs well in larger objects with visible boundaries in the guide image. Classes with weak color differences (e.g. trees next to low vegetation) don’t provide the clear gradient maps that GAD needs to perform well

to transfer edges from the guide image onto the predictions. It also shows that, as a postprocessing algorithm, the quality of the output is strongly linked to the quality of the input, and GAD is not able to accurately fix large errors in the predictions if the inputs miss large parts of the objects. These images illustrate that higher subsampling factors led to worse results, as is expected. Nevertheless, GAD was able to improve the precision of predicted region boundaries in all cases.

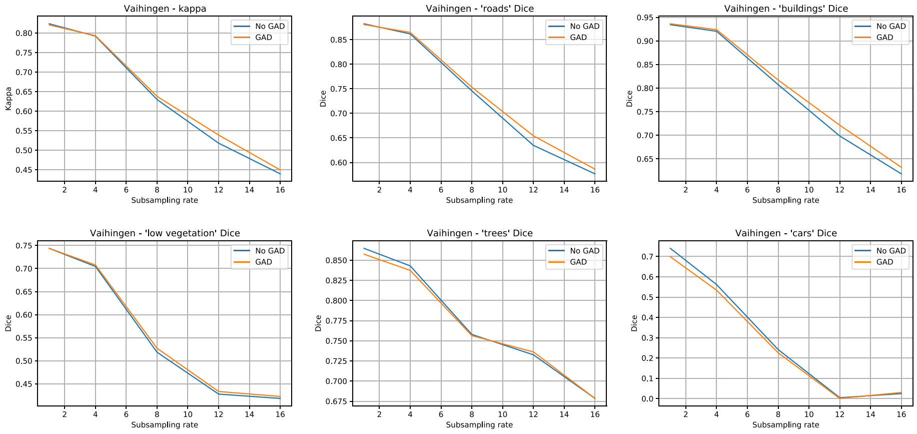


Fig. 18 Results for improving semantic segmentation on the Vaihingen dataset. Using GAD to postprocess the outputs consistently improves results for classes with larger objects and sharp visible edges

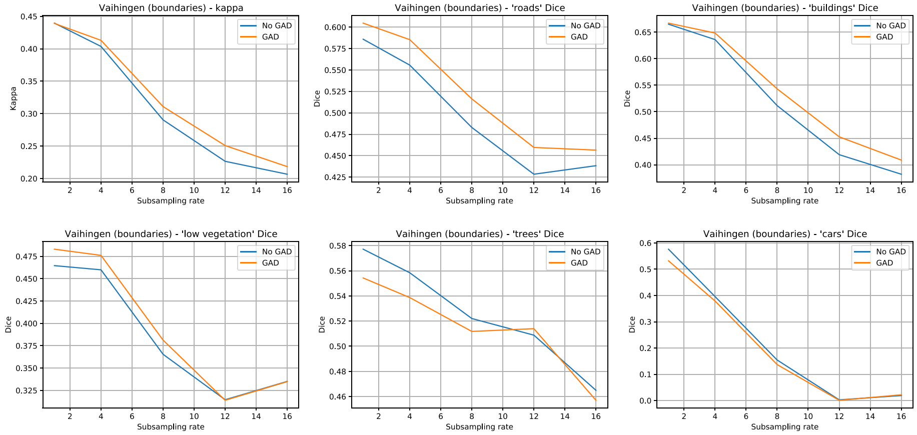


Fig. 19 Results for improving segmentation boundaries on the Vaihingen dataset considering only pixels around region boundaries. The effect of postprocessing results with GAD becomes clearer where one semantic region encounters another

The quantitative analysis presented in Fig. 16 shows a small but consistent improvement due to the GAD algorithm. The small scale of these quantitative improvements can be explained by the fact that GAD only affects region boundaries, which is itself only a small fraction of the total number of pixels. But the consistency that is observed in the improvement of these results show that GAD is clearly improving the predictions. It is also important to note that the Dice scores are improved for both classes. According to these results, the subsampling rate at which GAD leads to the biggest gain is $ss = 8$.

Results on the Vaihingen dataset can be seen in Figs. 17, 18, and 19. Two main conclusions can be drawn from Fig. 17. First, objects with well defined boundaries in the guide image (e.g. buidlings) benefit from GAD postprocessing. Second, objects with fuzzy borders or near objects with similar colors (e.g. trees next to grass) do not provide the necessary gradients that guide the anisotropic diffusion, and therefore do not benefit from GAD postprocessing.

Figure 17 shows that the classes “buildings”, “roads”, and “low vegetation” profit from GAD postprocessing similarly to the results on the Inria dataset. Once again, the small gains can be explained by the fact that GAD acts only on region boundaries. The “trees” and “cars” classes did not benefit from GAD postprocessing, likely for different reasons. The “trees” regions often had colours very similar to the ones of to the neighbouring regions, which did not lead to sharp gradient maps. The “cars” class contained objects that are relatively small, and therefore could easily be eroded away by the anisotropic diffusion. At higher subsampling rates, cars would only occupy one or two pixels in the image, which explains why the network itself failed to detect them at these scales.

To highlight the impact of GAD postprocessing around region boundaries, the same metrics were calculated using only pixels around region boundaries. The locations of such pixels were calculated using the complement of the “gts_eroded_for_participants” available with the Vaihingen dataset files. These results can be seen in Fig. 19. These results show that the effects of GAD postprocessing is much stronger around region boundaries, which is coherent with what was expected.

5 Analysis

The experiments presented in the previous section showed how GAD was successfully used in two different weakly supervised change detection settings. The results show an increase in performance in object-level segmentation from parcel-level labels through label cleaning, as well as the seldom explored task of weakly supervised image co-segmentation using classification labels.

The iterative training results made clear that it is of paramount importance to refer back to the ground truth data every time the training ground truth is being modified. Not doing so leads to a fast degradation in performance, since the network simply attempts to learn to copy itself and stops learning useful operations from the data. The results also showed that separating dubiously labelled pixels leads to a small increase in performance, likely due to the fact that we end up providing a cleaner and more trustworthy dataset at training time.

The guided anisotropic diffusion algorithm was compared against the Dense CRF algorithm for using information from the input images to improve semantic segmentation results. While both algorithms were successful when used in the proposed iterative training scheme, GAD outperformed Dense CRF at later hyperepochs for quantitative metrics. Both algorithms yielded visually pleasing results, each performing better in different test cases.

One possible criticism of the proposed iterative training method is that it would get rid of hard and important examples in the training dataset. It is true that the performance of this weakly supervised training scheme would likely never reach that of one supervised with perfectly clean data, but the results in Sect. 4 show that using the proposed method we can consistently train networks that perform better than those naively trained with noisy data directly.

The proposed spatial attention operation was showed to be useful in improving the classification and weakly supervised segmentation results for datasets which are cropped using object locations as reference points. While this is a particular case, such datasets are often available or can be easily generated for remote sensing applications, where georeferenced data is widely available. The proposed ideas have been only tested in a two-class problem, but there is nothing that indicates that such methods would not work just as well in a multi-class context. Filtering the attention weights with the GAD algorithm further increased the classification performance of the network by increasing the coherence between the attention weights and the region where the building of interest is located in each image.

Finally, the usage of GAD as an edge enhancing postprocessing algorithm was tested in a semantic segmentation setting using two remote sensing datasets to compensate for networks trained with lower spatial resolution images. The results were mostly positive, and showed that GAD is effective at improving segmentation boundaries for classes with well defined edges and strong gradients, as long as the objects are not too small. This shows that GAD is a versatile tool for enhancing segmentation edges in a variety of settings.

6 Conclusion

In this paper we have proposed the guided anisotropic diffusion algorithm for improving semantic segmentation results by performing a cross-image edge preserving filtering. It was shown to improve semantic segmentation results on two standard aerial datasets, leading to better boundary accuracy for semantic segmentation results. We have then proposed two GAD-based weakly supervised change detection methods to demonstrate how it can help to recover from inaccurate segmentation labels or go beyond the available classification labels.

We first proposed an iterative training method for training networks with noisy data that alternates between training a fully convolutional network and leveraging its predictions to clean the training dataset from mislabelled examples. We showed that the proposed method outperforms naive supervised training using the provided reference data for change detection. The GAD algorithm was used in conjunction with the iterative training method to obtain the best results in our tests. The GAD algorithm was compared against the Dense CRF algorithm, and was found to be superior in performance.

Finally, we proposed a spatial attention operation that can be easily incorporated into existing classification networks that significantly improve the classification and weakly supervised segmentation performances for datasets with object-aligned crops.

The proposed methods are useful when using data-based approaches in data-scarce domains, as is the case of change detection. We have observed improvement in all of our tests when approaching the problem from a weakly supervised perspective, as opposed to naive supervision. It would be interesting to test the efficacy of the proposed ideas outside the context of change detection. The proposed methods could be applied with minor adaptations to other applications to help mitigate the effects of data scarcity.

References

Ahn, J., & Kwak, S. (2018). Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4981–4990).

Alvarez, J. L. H., Ravanbakhsh, M., & Demir, B. (2020). S2-cGAN: Self-supervised adversarial representation learning for binary change detection in multispectral images.

Aubert, G., & Kornprobst, P. (2006). *Mathematical problems in image processing: Partial differential equations and the calculus of variations*, vol. 147. Springer.

Audebert, N., Saux, B. L., & Lefèvre, S. (2017). Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*. https://doi.org/10.1016/j.isprsjprs.2017.11.011.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495.

Benedek, C., & Szirányi, T. (2009). Change detection in optical aerial images by a multilayer conditional mixed Markov model. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(10), 3416–3430.

Chen, Y., Ouyang, X., & Agam, G. (2018). MFCNET: End-to-end approach for change detection in images. In *2018 25th IEEE international conference on image processing* (pp. 4008–4012). IEEE.

Dai, J., He, K., & Sun, J. (2015). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1635–1643).

Daudt, R. C., Chan-Hon-Tong, A., Le Saux, B., & Boulch, A. (2019a). Learning to understand earth observation images with weak and unreliable ground truth. In *IEEE International Geoscience and Remote Sensing Symposium* (pp. 5602–5605). https://doi.org/10.1109/IGARSS.2019.8898563.

Daudt, R. C., Le Saux, B., & Boulch, A. (2018a). Fully convolutional Siamese networks for change detection. In *2018 25th IEEE international conference on image processing* (pp. 4063–4067).

Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2018b). Urban change detection for multispectral earth observation using convolutional neural networks. In *International geoscience and remote sensing symposium* (pp. 2119–2122). IEEE.

Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2019b). Guided anisotropic diffusion and iterative learning for weakly supervised change detection. In *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*.

Daudt, R. C., Le Saux, B., Boulch, A., & Gousseau, Y. (2019c). Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding*, *187*, 102783. https://doi.org/10.1016/j.cviu.2019.07.003.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, *26*(3), 297–302.

El Amin, A. M., Liu, Q., & Wang, Y. (2016). Convolutional neural network features based change detection in satellite images. In *First international workshop on pattern recognition* (pp. 100110W–100110W). International Society for Optics and Photonics.

El Amin, A. M., Liu, Q., & Wang, Y. (2017). Zoom out CNNs features for optical remote sensing change detection. In *2017 2nd International conference on image, vision and computing (ICIVC)* (pp. 812–817). IEEE.

Ferstl, D., Reinbacher, C., Ranftl, R., Rüther, M., & Bischof, H. (2013). Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE international conference on computer vision* (pp. 993–1000).

Frénay, B., & Kabán, A., et al. (2014). A comprehensive introduction to label noise. In *European symposium on artificial neural networks*.

Frénay, B., & Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(5), 845–869.

Fujita, A., Sakurada, K., Imaizumi, T., Ito, R., Hikosaka, S., & Nakamura, R. (2017). Damage detection from aerial images via convolutional neural networks. In *2017 Fifteenth IAPR international conference on machine vision applications (MVA)* (pp. 5–8). https://doi.org/10.23919/MVA.2017.7986759.

Guo, E., Fu, X., Zhu, J., Deng, M., Liu, Y., Zhu, Q., & Li, H. (2018). Learning to measure change: Fully convolutional Siamese metric networks for scene change detection. *CoRR*. arXiv:1810.09111

Guyon, I., Matic, N., & Vapnik, V., et al. (1996). Discovering informative patterns and data cleaning. In *Association for the advancement of artificial intelligence*.

He, K., Sun, J., & Tang, X. (2013). Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(6), 1397–1409.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hussain, M., Chen, D., Cheng, A., Wei, H., & Stanley, D. (2013). Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, *80*, 91–106.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Jeatrakul, P., Wong, K. W., & Fung, C. C. (2010). Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *14*(3), 297–302.

John, G. H. (1995). Robust decision trees: Removing outliers from databases. In *KDD* (pp. 174–179).

Khoreva, A., Benenson, R., Hosang, J., Hein, M., & Schiele, B. (2017). Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 876–885).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Koenderink, J. J. (1984). The structure of images. *Biological Cybernetics*, *50*(5), 363–370.

Kopf, J., Cohen, M. F., Lischinski, D., & Uyttendaele, M. (2007). Joint bilateral upsampling. In *ACM Transactions on Graphics* (Vol. 26, p. 96). ACM.

Krähenbühl, P., & Koltun, V. (2011). Efficient inference in fully connected CRFs with gaussian edge potentials. In *Advances in neural information processing systems* (pp. 109–117).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Lu, Z., Fu, Z., Xiang, T., Han, P., Wang, L., & Gao, X. (2017). Learning from weak and noisy labels for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(3), 486–500.

Luppino, L. T., Hansen, M. A., Kampffmeyer, M., Bianchi, F. M., Moser, G., Jenssen, R., & Anfinsen, S. N. (2020). Code-aligned autoencoders for unsupervised change detection in multimodal remote sensing images.

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark. In *IEEE International geoscience and remote sensing symposium (IGARSS)*. IEEE.

Matic, N., Guyon, I., Bottou, L., Denker, J., & Vapnik, V. (1992). Computer aided cleaning of large databases for character recognition. In *International conference on pattern recognition* (pp. 330–333). IEEE.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., & Tewari, A. (2013). Learning with noisy labels. In *Advances in neural information processing systems* (pp. 1196–1204).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(7), 629–639.

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., & Toyama, K. (2004). Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, *23*(3), 664–672.

Rolnick, D., Veit, A., Belongie, S. J., & Shavit, N. (2017). Deep learning is robust to massive label noise. *CoRR*. arXiv:1705.10694.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Saha, S., Bovolo, F., & Bruzzone, L. (2019). Unsupervised deep change vector analysis for multiple-change detection in VHR images. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(6), 3677–3693. https://doi.org/10.1109/TGRS.2018.2886643.

Saha, S., Bovolo, F., & Bruzzone, L. (2020). Building change detection in VHR SAR images via unsupervised deep transcoding. *IEEE Transactions on Geoscience and Remote Sensing*, 1–13. https://doi.org/10.1109/TGRS.2020.3000296.

Saha, S., Mou, L., Zhu, X. X., Bovolo, F., & Bruzzone, L. (2020). Semisupervised change detection using graph convolutional network. *IEEE Geoscience and Remote Sensing Letters*, 1–5. https://doi.org/10.1109/LGRS.2020.2985340.

Sakurada, K., & Okatani, T. (2015). Change detection from a street image pair using CNN features and superpixel segmentation. In *British Machine Vision Conference* (pp. 61–1).

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. *International Journal of Remote Sensing*, *10*(6), 989–1003.

Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, *5*, 1–34.

Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2691–2699).

Zhan, Y., Fu, K., Yan, M., Sun, X., Wang, H., & Qiu, X. (2017). Change detection based on deep Siamese convolutional network for optical aerial images. *IEEE Geoscience and Remote Sensing Letters*, *14*(10), 1845–1849.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *The IEEE conference on computer vision and pattern recognition (CVPR).*