INVESTIGATIONS ON FEATURE SIMILARITY AND THE IMPACT OF TRAINING DATA FOR LAND COVER CLASSIFICATION

M. Voelsen[1]*, D. Lobo Torres[2], R. Q. Feitosa[2] F. Rottensteiner[1], C. Heipke[1]

[1] Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
(voelsen, rottensteiner, heipke)@ipi.uni-hannover.de
[2] Department of Electrical Engineering, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, Brazil
(daliana91, raul)@ele.puc-rio.br

Commission III, WG III/7

KEY WORDS: land cover classification, remote sensing, FCN, cosine similarity loss

ABSTRACT:

Fully convolutional neural networks (FCN) are successfully used for pixel-wise land cover classification - the task of identifying the physical material of the Earth's surface for every pixel in an image. The acquisition of large training datasets is challenging, especially in remote sensing, but necessary for a FCN to perform well. One way to circumvent manual labelling is the usage of existing databases, which usually contain a certain amount of label noise when combined with another data source. As a first part of this work, we investigate the impact of training data on a FCN. We experiment with different amounts of training data, varying w.r.t. the covered area, the available acquisition dates and the amount of label noise. We conclude that the more data is used for training, the better is the generalization performance of the model, and the FCN is able to mitigate the effect of label noise to a high degree. Another challenge is the imbalanced class distribution in most real-world datasets, which can cause the classifier to focus on the majority classes, leading to poor classification performance for minority classes. To tackle this problem, in this paper, we use the cosine similarity loss to force feature vectors of the same class to be close to each other in feature space. Our experiments show that the cosine loss helps to obtain more similar feature vectors, but the similarity of the cluster centers also increases.

1. INTRODUCTION

Pixel-wise classification of land cover is the task of assigning a class label to each pixel in an image. The classes correspond to different physical materials of the Earth's surface, e.g. *settlement* or *vegetation*. The most popular methods for this task are variants of Fully Convolutional Networks (FCNs) (Long et al., 2015) based on architectures such as U-Net (Ronneberger et al., 2015) or Deeplab (Chen et al., 2018).

Deep neural networks need a sufficient amount of labeled data for training (Krizhevsky et al., 2012). In remote sensing, it is hard to obtain enough reliable data as manual labeling is time consuming and costly, and existing datasets are limited in size (Zhu et al., 2017). This may lead to overfitting, so that the classifier does not generalize well to unseen data (Goodfellow et al., 2016). In remote sensing, large amounts of training data can be obtained automatically if the class labels are extracted from existing geospatial databases (called *maps* hereafter). However, some of these labels will be incorrect, e.g. due to temporal changes (Maas et al., 2019), i.e. the data will be affected by *label noise* (Frenay and Verleysen, 2014). Many strategies have been proposed to deal with label noise, e.g. the use of robust classifiers (Song et al., 2020) or training strategies (Maas et al., 2019). Drory et al. (2018) showed that FCN are robust to label noise to a certain degree if the noise is spread randomly and the errors are not concentrated in some classes. However, it is unclear whether this applies to the case when training labels are extracted from existing maps, where, for instance, errors occur in spatial clusters. Thus, the first question investigated in this paper is related to the generalization capabilities of a FCN for land cover classification trained on large amounts of noisy training data. We conduct experiments in which, starting from a large pool of data with noisy annotations, we vary the training data set with respect to size, composition and level of label noise and compare the results of the trained classifier to a reference to investigate the impact of these variations on the results.

Another common problem in training is an imbalanced distribution of the classes in the training data, which occurs frequently in remote sensing applications. Such an imbalance causes the classifier to focus on the majority classes and, consequently, leads to poor results for the underrepresented classes (Johnson and Khoshgoftaar, 2019). To cope with this problem one can adapt the training procedure to focus on the underrepresented classes. This can be achieved by weights in the loss function that force the classifier to focus on samples that are hard to classify (Lin et al., 2017). Another approach is to adapt the training strategy so that samples from different classes form distinct and separate clusters in feature space. Motivated by (Voelsen et al., 2020), where the imbalanced class distribution was identified to be one of the limiting factors for the classification of satellite images, we investigate the cosine similarity loss, e.g. (Yang et al., 2020), to ensure that feature vectors of the same class are close to each other in feature space. In this context, we also investigate which is the best FCN layer at which to apply this loss to obtain an optimal classification performance.

In our experiments, based on a variant of U-Net (Ronneberger et al., 2015), we use optical Sentinel-2 data covering the entire German state of Lower Saxony (47600 km^2) at 16 epochs. The training labels are derived from a topographic database and differentiate six land cover classes.

The scientific contribution of this paper can be summarized as follows: **(1)** We investigate the generalization capabilities of an

* Corresponding author

FCN trained using a very large set of noisy training data; **(2)** In this context, we assess the impact of the size and composition of the training dataset on the results to see how the selection of the epochs to be used for training affects the results. We also investigate the influence of different degrees of simulated label noise on the classifier; **(3)** We investigate the cosine loss as a strategy for increasing the classification accuracy for under-represented classes.

2. RELATED WORK

While recent advances in the FCN architecture have led to vast improvements in different remote sensing applications; see (Zhu et al., 2017) or (Shi et al., 2020) for overviews, a major limiting factor is the lack of large representative datasets that are publicly available for training such networks. Most existing remote sensing datasets are limited in size or in the seasonal variation, or they are only relevant for some very specific task; see (Hoeser et al., 2020) for an overview. One possibility to create large amounts of training data without manual labelling is the automatic generation of class labels by using data from existing maps, assuming that most of the objects did not change between the generation of the map and the acquisition of the images to be classified, e.g. (Kaiser et al., 2017; Zhang et al., 2020). However, a certain amount of the class labels thus produced will be wrong for various reasons, e.g. temporal changes (Maas et al., 2019). Song et al. (2020) categorized seven different research directions to cope with label noise, including the use of robust architectures, regularization of loss functions and the selection of samples that are least likely to have wrong labels. In remote sensing, Mnih and Hinton (2012) proposed a convolutional neural network (CNN) for the binary classification of aerial images using training labels derived from Open Street Map (OSM) data. They proposed an error model tailored to the most frequent error types, relying on the availability of some error-free data in order to determine its parameters. Li et al. (2020) also used OSM to generate training labels. They developed a probabilistic noise model which is based on the dependencies between the input images, the noisy labels and the true labels and outperforms other state-of-the-art methods. Zhang et al. (2020) proposed a noise-adaptive FCN framework using noisy building footprints from a database. Their framework consists of the base FCN combined with a module that captures the relationship between the true labels and the noisy ones and is robust to label noise in their data. Maas et al. (2019) proposed a label-noise robust random forest classifier for image classification based on maps. Besides, OSM there are a lot of other possible data sources to obtain class labels: Ulmas and Liiv (2020) used Sentinel-2 images together with the CORINE Land Cover map 2018. Schmitz et al. (2020) combined information from OSM, CORINE Land Cover 2018, Global Surface Water and SAR data to create more reliable class labels than any of the single products can provide. Postadjian et al. (2017) used existing very high resolution land cover maps to train a simple FCN architecture. They trained and tested different models on different regions and conclude that the accuracy drops when the model is used to predict labels of another geographical area. Using a fine tuning step, the results are improved. However, none of these papers rely on the availability of a very large (state-level) dataset to train a model so that it remains unclear to which extent the results of these methods can be generalized.

Thus, before developing methods to cope with label noise, it is important to assess its impact on the classification results. Drory et al. (2018) showed that the impact of label noise on the performance of a neural network depends on its statistical properties. If the neighbourhood of noisy samples contains mostly correct samples in feature space and if it affects all classes in the same way, the influence of label noise is relatively low; otherwise, it has a clear negative effect on the results. Whether this is the case in the application envisaged in this paper is unclear. Kaiser et al. (2017) investigated the influence of noisy training data on a FCN, using OSM data and aerial images from Google Maps from five different cities. They show that the results of classification are affected in a negative way if no hand-labelled data are used at all for the imagery to be classified. However, using the OSM data to pre-train the network and fine-tune it using noise-free data from the imagery to be classified improves the classification accuracy considerably. It is unclear whether these conclusions also hold for the classification of multi-temporal satellite images with a coarser resolution, a more fine-grained class structure and labels that have different error characteristics (e.g., maps produced by crowdsourcing vs. by professional mapping agencies). Furthermore, the aspect of using data with noisy labels at the level of an entire state has not been considered so far. This paper investigates these questions based on multi-temporal Sentinel-2 data.

Another common problem in training is an imbalanced class distribution in the training data. Johnson and Khoshgoftaar (2019) differentiated methods that modify the data, e.g. by under- or oversampling, to solve this problem, and algorithmic approaches relying on modified training procedures. The latter approaches have the advantage that they do not require data preprocessing. Frequently, the training procedure is modified by considering weights in the loss function that force the classifier to focus on samples that are hard to classify. Examples for such loss functions are the focal loss (Lin et al., 2017) or its extension to multi-class problems (Yang et al., 2020), or the dice loss (Ren et al., 2020), the latter references giving applications in remote sensing. Another approach is to adapt the training strategy so that samples from different classes form distinct and separate clusters in feature space. If a FCN learns to produce such a representation, it might also be more likely for features from under-represented classes to form distinct clusters and, consequently, to be classified correctly (Wang et al., 2020). In order to do so, similarity measures such as the Euclidean distance or cosine similarity are applied to formulate additional loss function terms. Hadsell et al. (2006) proposed the contrastive loss that minimizes the Euclidean distance of similar pairs and maximizes the distance of dissimilar pairs. In the triplet loss of Schroff et al. (2015), triplets of positive and negative pairs are used to push feature vectors of positive pairs to be close to and those of negative pairs to be far away from each other. Yang et al. (2020) applied a cosine similarity loss for pixel-wise land cover classification from aerial imagery to ensure that features belonging to the same class are close to their centroids in feature space. Using this method they improved the average F1-score by 3%. However, it is unclear how such a loss performs in cases involving satellite data and in which the imbalance is more pronounced.

A high intra-class variability in combination with label noise, which is not present in (Yang et al., 2020), might make it impossible for the classifier to find separate distinct clusters in feature space. We investigate this question by using the cosine similarity loss in the training process based on multi-temporal Sentinel-2 data. Another question not dealt with by existing work is related to the definition of the feature representation to which such a loss should be applied; we try to answer this

question by applying this loss to various layers of the FCN and compare the results.

3. METHODOLOGY

3.1 Network Architecture

The network architecture used in this paper is a variant of U-Net (Ronneberger et al., 2015) designed for Sentinel-2 imagery and shown figure 1. The input layer consists of an image of size 256×256 with 10 spectral bands. The encoder is composed of four convolutional blocks, each consisting of two 3×3 convolutional layers followed by batch normalization (BN) (Ioffe and Szegedy, 2015) and a rectified linear unit (ReLU). To reduce the spatial dimension, we add a max-pooling layer after each encoder block. The encoder is linked to the decoder by another convolutional block without a downsampling layer. The decoder consists of four upsampling layers that use bilinear interpolation, each followed by another convolutional block. Similarly to U-Net, there are skip connections between corresponding layers of the encoder and the decoder; the corresponding features are concatenated before further processing. Finally, a 1×1 convolution maps the feature vectors to raw class scores, which are normalized by a softmax layer.

3.2 Training

Training is based on minimizing a loss function using stochastic minibatch gradient descent (Bishop, 2006). Our baseline method uses the cross entropy loss with class weights to counteract the imbalanced class distribution (Section 3.2.1). In addition, a cosine similarity loss forcing features of samples of the same class to be close to the class centroid (Yang et al., 2020) will be used in some experiments; it is described in Section 3.2.2.

3.2.1 Cross Entropy Loss: The weighted cross-entropy loss L_{CrEn} is based on the softmax predictions y_n^k for sample x_n to belong to class k,

$$L_{CrEn} = -\sum_n \sum_k C_n^k \cdot ln(y_n^k) \cdot cw_k. \quad (1)$$

In eq. 1, $C_n^k = 1$ if the n^{th} sample belongs to class k, otherwise $C_n^k = 0$. The class weights cw_k are based on the number of occurences n_k of class k in the training data (Patel, 2020):

$$cw_k = \frac{log(N) - log(n_k)}{\max_j(log(N) - log(n_j))}, \quad (2)$$

where N is the total number of pixels in all training patches. These weights are equal or near to one for the under-represented classes and lower for the majority classes. Thus, the impact of samples from a minority class with incorrect predictions on the loss is much higher, which compensates for the imbalance of the dataset up to a certain degree.

3.2.2 Cosine Loss: As a further measure to counteract an imbalanced class distribution, we consider a constraint based on cosine similarity in training. The cosine similarity, i.e. the cosine of the angle between two vectors, can be used to measure feature differences. It forces feature vectors of samples belonging to the same class to be close to each other in feature space, which is assumed to help to produce well-formed clusters also for the minority classes and, thus, improve the results. In this context, the cosine similarity can be computed based on features from any layer of the FCN; in our experiments we compare four variants (cf. f1-f4 in figure 1). The cosine similarity loss obviously needs the class labels of the feature vectors. Thus, if it is applied to layers of lower resolution than the input, the corresponding feature maps are upsampled by bilinear interpolation before being passed on to the loss function, so that the class labels of the upsampled feature map can be taken from the reference.

The implementation of the cosine loss follows (Yang et al., 2020). First, the raw features $\vec{f^i}$ for each pixel i at the selected layer in the current minibatch are passed through the ReLU activation function, resulting in feature vectors $\vec{a}^i = ReLU(f^i)$. By using the class labels of the images, the number of pixels m_k of class k can be calculated for the minibatch. Then, the mean feature vector $\vec{u}^k$ is calculated using all feature vectors belonging to class k:

$$\vec{u}^k = \frac{1}{m_k} \sum_i^M C_i^k \vec{a}^i, \quad (3)$$

where $C_i^k = 1$ if feauture vector $\vec{a}^i$ belongs to class k and $C_i^k = 0$ otherwise and M is the total number of pixels in the minibatch. Next, the cosine similarity between each feature vector $\vec{a}^i$ and the corresponding mean feature vector $\vec{u}^k$ is computed:

$$cos(\vec{a}^i, \vec{u}^k) = \frac{\vec{a}^i \cdot \vec{u}^k}{||\vec{a}^i||_2 \cdot ||\vec{u}^k||_2}. \quad (4)$$

As it is the goal of using the cosine similarity to obtain a feature representation that forms compact clusters, the sum of cosine similarity of all pixels in the minibatch would have to be maximized. Thus, it cannot be used directly to define a loss function, because the loss has to be minimized in training. Consequently, the cosine similarity loss is defined according to:

$$L_{cos} = \frac{1}{M} \sum_i^M max(1 - cos(\vec{a}^i, \vec{u}^{c_i}) - t, 0), \quad (5)$$

where c_i is the class pixel i belongs to, M is the number of all pixels in the current minibatch and t defines a margin inside which the cosine similarity can vary without a negative effect on the loss (e.g. a margin of 0.1 would define a range of 0.9 - 1). For all experiments using the cosine similarity loss it is combined with the cross entropy loss, leading to a combined loss function L_{comb}:

$$L_{comb} = L_{CrEn} + \alpha \cdot L_{cos}. \quad (6)$$

The parameter α controls the trade-off between both losses.

4. EXPERIMENTS

4.1 Test Data and Test Setup

4.1.1 Test Data: The study site covers the whole area of the German federal state of Lower Saxony (47600 km^2). The dataset comprises Sentinel-2 images from 16 dates between May 2016 and November 2020, provided by the European Space Agency (ESA). We use Sentinel-2 Level-2A data, which contain georeferenced bottom-of-atmosphere reflectance and cloud masks from the top-of-atmosphere reflectance of every pixel

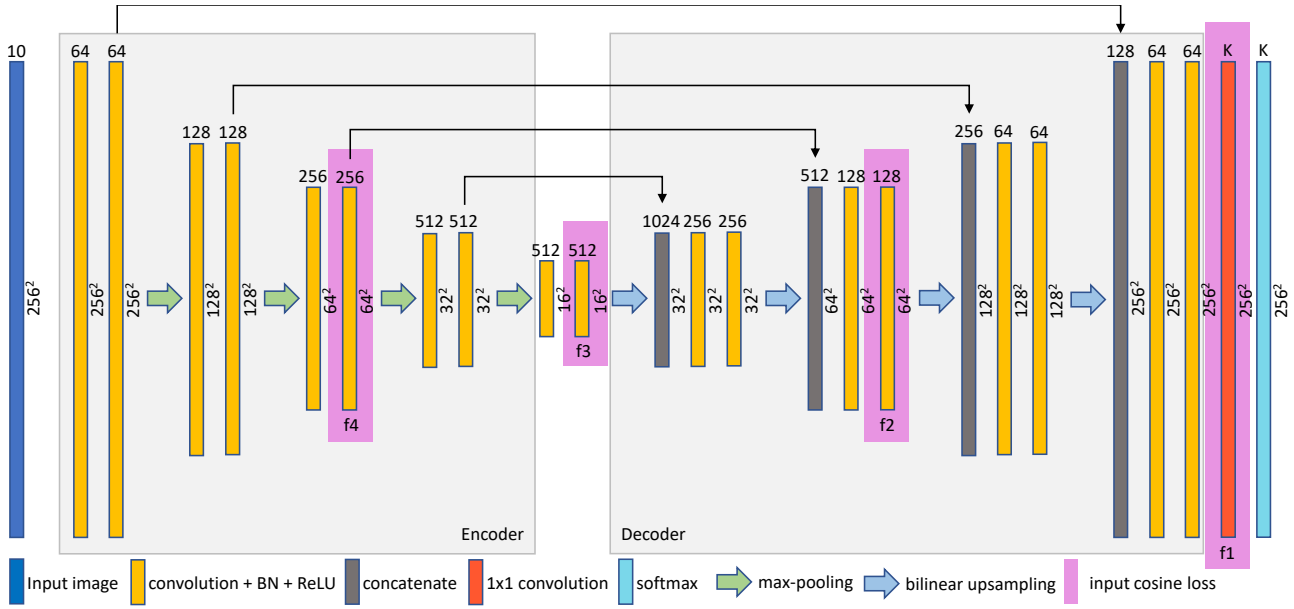


Figure 1. The network architecture. f1 - f4 identify layers that are used as input to the cosine similarity loss in some experiments. Squared numbers indicate the size of the feature maps in pixel, numbers on top of the layers indicate the number of filters.

(Fletcher, 2012). We use the four spectral bands with a ground sampling distance (GSD) of 10 m (red, green, blue, near infrared) and six bands with 20 m GSD. The latter are upsampled to 10 m using bilinear interpolation. The cloud mask is used to exclude parts of the images that contain more than 5% cloud coverage. The dataset contains images from all seasons, acquired at the following days: 2016-04-02, 2016-05-02, 2016-05-05, 2016-05-08, 2016-09-12, 2017-10-15, 2018-04-10, 2018-07-24, 2018-12-11, 2019-02-14, 2019-08-31, 2020-03-23, 2020-04-24, 2020-06-23, 2020-08-07, 2020-11-08.

To obtain the class labels to be used in training, information from the official German landscape model ATKIS is used (AdV, 2008). This database contains information about 64 different land use classes, which is too detailed for automatic classification. To define a suitable class structure for land cover, several land use classes from the database are merged, so that in the end, six classes are differentiated: *Building (bld.), Sealed area (sld.), Agriculture (agr.), Greenland (grl.), Water (wat.)* and *Forest (for.)*. In addition, the class *others* is used for areas without label information that occur due to errors in the database or for areas outside the state borders. This information is used to disregard samples of this class in training and evaluation. The database is updated at irregular intervals that can vary between a few days and three years. For the experiments reported in this paper, one reference label image at the geometrical resolution of the satellite imagery is created for every year, and each Sentinel-2 image is combined with the label image corresponding to the year of its acquisition. This will lead to some label noise, as some more recent changes will not yet be contained in the database.

For computational reasons, the available data is split into tiles of 8×8 km^2 (800 × 800 pixels), which leads to a total number of 950 tiles covering Lower Saxony (cf. figure 2). For one tile (shown in red in figure 2), the corresponding reference label image was corrected manually for two epochs (2016-05-05, 2020-04-24) to obtain a reference for the evaluation that is not affected by label noise. In this process, about 8% of the pixels were changed, which gives an indication to the amount of label noise to be expected in the remaining data. Figure 3 shows one of the two images and the reference for that dataset.

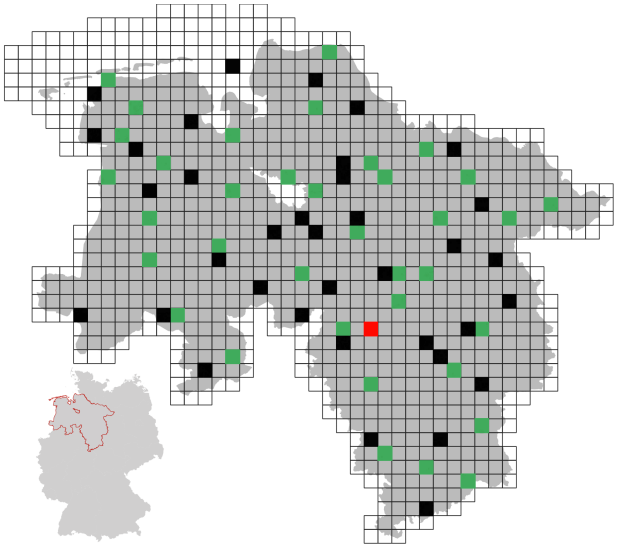


Figure 2. Overview of the available data tiles of 8×8 km^2. Grey / green: potential training / validation tiles. Red: test tile with corrected reference (dataset R_1). Black: test tiles without corrected reference (dataset R_2).

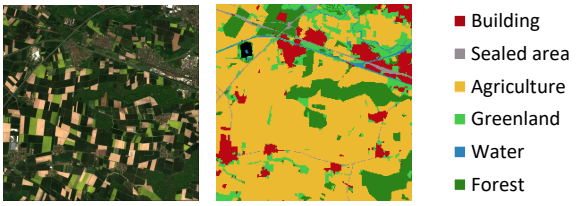


Figure 3. One Sentinel-2 image of size 8×8 km^2 (left) and corrected reference (middle) for dataset R_1.

4.1.2 General Test Setup: In our experiments, we compare results of the method described in Section 3 for different scenarios. For that purpose, 37 of the available tiles are set aside for testing (black and red tiles in figure 2), another 37 tiles are used for validation (green tiles in figure 2), and the remaining 876 tiles form a pool of training data. Training is based on the method described in Section 3.2. We randomly crop windows of 256×256 pixels from the available training tiles and apply random data augmentation, including rotations by 90°, 180°, 270°, horizontal and vertical flipping. As this results in a large set of training patches, the number of patches used in one epoch is restricted to 2000. Training continues for a maximum number of epochs of 250, but it is stopped earlier if the validation accuracy does not increase for 30 epochs. The minibatch size is set to 2. The training process is started with a learning rate of 0.01 that decreases by a factor of 0.7 after every 10 epochs. In the experiments involving the cosine similarity loss, the parameter α (equ. 6) is set to 1 and t (equ. 5) is set to 0.2.

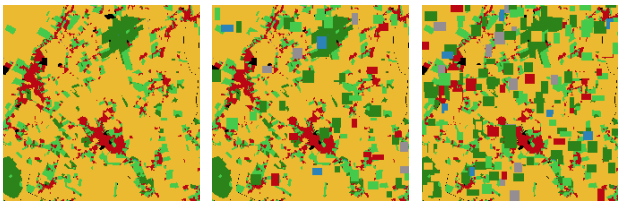
For the evaluation, the results of the FCN achieved for the test tiles is compared to the available reference and quality indicators are determined based on this comparison on a per-pixel level in all experiments. We report the Overall Accuracy (OA), i.e. the percentage of pixels with correctly predicted class labels, the F1-scores per class, i.e. the harmonic mean of precision and recall, and the average F1-score ($avg.F1$), i.e. the mean of the F1-scores for the individual classes, as a compound quality metric that is more susceptible to problems in underrepresented classes than OA. On the one hand, these indicators are determined on the basis of the tile with the corrected labels and the images from 2016-05-05 and 2020-04-24 (referred to as dataset R_1; red tile in figure 2). These numbers are not affected by errors in the reference, but they are only based on a small sample. Note that the images of the acquisition dates of the reference are not used for training in any of the experiments. In order to obtain indicators based on a larger set of samples, we use a second reference dataset R_2 consisting of data from 37 tiles (black in figure 2). However, these indicators will be biased due to the label noise present in the reference. We carried out three sets of experiments, investigating different aspects, as will be explained in the subsequent subsections.

4.1.3 Test Series 1 - Amount and Composition of Training Data: In the first set of experiments, described in Section 4.2, we want to assess the impact of varying the amount and the composition of the training data on the generalization performance of the FCN. To this end, we train the same classifier with training data varying in size, in the number of included Sentinel-2 dates, and in both aspects. For that purpose, we defined three sets of training data of different size (sets A, B and C in table 1, containing 100%, 20% and 1% of the area of Lower Saxony, respectively); in three of the experiments, images from 14 epochs were used for training, but using different numbers of tiles (i.e. all except for the two epochs from which the reference dataset R_1 was generated), in one experiment we only used the four epochs from 2020, and in two experiments we only used the data from one epoch (2020-06-23). Table 1 also shows the class distributions in the different datasets. First of all, it is obvious that this distribution is very imbalanced. In particular, *sealed area* is extremely underrepresented, covering only 0.7% of the pixels of the overall area (set A). There are also variations between the datasets, especially for class *water*.

4.1.4 Series 2 - Different Levels of Label Noise: In the second set of experiments, reported in Section 4.3, we evaluate the impact of different degrees of label noise on the generalization performance of the classifier. For that purpose, we use the entire training set (set A in table 1) with data from 14 epochs (all except those used for generating R_1), but we randomly change a certain percentage of the training labels, thus producing additional reference data sets with 5%, 10%, 20% and 30% of changed labels, respectively; at each level of additional label noise, we create two variants of the contaminated reference to see whether the spatial distribution has an impact on the results. As the original data already contain a certain amount of label noise the total amount of noise cannot be specified. Nevertheless, these experiments should give an indication for the direction of change of classification accuracy with increasing noise level. The noise is added by changing class labels in rectangles of random side lengths in the range of 20 and 50 pixels. To keep the class distribution approximately the same, the probability that the area inside the rectangle is assigned to a specific class is based on the class distribution of dataset A (e.g a rectangle is assigned to *Agriculture* with a chance of 38%, see Table 1). An example of different amounts of introduced label noise is shown in figure 4.

Set	N_{tiles}	Distribution of classes [%]					
		bld.	*sld.*	*agr.*	*grl.*	*wat.*	*for.*
A	950	8.7	0.7	38.0	21.5	12.9	18.2
B	159	9.0	0.8	42.0	19.8	1.3	27.1
C	9	15.4	1.7	52.4	14.1	4.9	11.6
R_1	1	9.4	1.7	61.7	11.4	1.4	14.3

Table 1. Number of tiles (N_{tiles}) of the different training datasets (A, B, C) and the reference R_1 used in the experiments, and the corresponding class label distributions.



(a) No noise (b) 10% noise (c) 30% noise

Figure 4. Examples for training data with different percentages of simulated label noise. Colour code: cf. figure 3.

4.1.5 Series 3 - Cosine Loss: The third set of experiments, presented in Section 4.4, evaluates the cosine loss as a strategy to increase the accuracy of underrepresented classes. We use different layers of the FCN as input features for the cosine loss to investigate the degree to which the quality of the results depends on this selection. We selected four candidate layers f1 - f4 (highlighted in figure 1) and use subsets to compute the cosine loss in different variants. An overview of the different input variants is shown in table 2. When f2, f3 or f4 is used as input, the number of feature maps is high (up to 512) and the cosine similarity computation becomes very slow. Thus, a selection step is integrated before passing the features into the cosine similarity calculation. For this selection the feature variance is calculated for every layer per class. Afterwards, the highest variances per layer are compared and the 10 features having the highest variance are used for cosine similarity calculation for a number of 100 minibatches before the selection process starts again. For these experiments, we also want to investigate the degree of feature similarity both between and across classes depending on whether the cosine loss is used for training or not. To do so, we calculate the mean feature vector

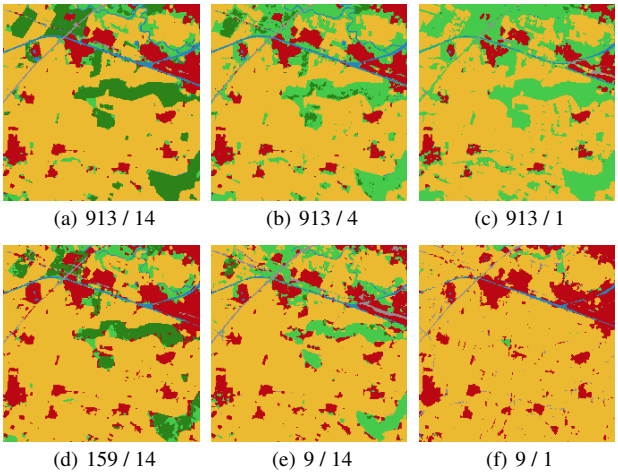
per class (eq. 3) and then the cosine similarity (eq. 4) between the individual feature vectors and the mean feature vector of the respective class. Afterward, the mean cosine similarity and its variance can be calculated for each class. In addition the cosine similarity between the mean feature vectors of each class is calculated. This evaluation should help to understand whether the goal of obtaining more distinct clusters for the individual classes is achieved and to see how compact these clusters are.

Variant	Loss	cosine loss input
Cr-En	L_{CrEn}	-
CL-f1	L_{comb}	f1
CL-f13	L_{comb}	f1, f3
CL-f124	L_{comb}	f1, f2, f4
CL-f1234	L_{comb}	f1, f2, f3, f4

Table 2. Variants for comparisons related to the cosine loss.

4.2 Evaluation: Amount and Composition of Training Data

To assess the impact of the size and composition of the training data on the classification performance we carried out experiments based on six different training datasets selected in the way described in Section 4.1.3. The results are shown in table 3. Figure 5 shows results for one of the epochs in the reference R_1. As can clearly be seen a classifier trained on a large amount of data which are also representative for the appearance of objects in various seasons has better generalization properties. Trained using all available data (experiment 0 in table 3), the FCN achieves an OA of 90% and a mean F1-score 75% on R_1. If data covering the entire area, but fewer epochs are used (experiments 1 and 2 in table 3), the OA drops considerably (9% if 4 epochs are used, 18% if only one epoch is used). This is mainly due to the inability of the classifier to differentiate *forest* and *grassland*, but also *sealed area* becomes much worse. It would seem that a combination of data from multiple epochs that would be more representative for vegetation classes in terms of covering more stages of plant development has a considerable stabilizing effect. However, the size of the area also matters: if the area is reduced, the classification accuracy is reduced by a similar margin even if all epochs are used (7% and 15% in experiments 3 and 4, respectively). In this case, the accuracy is also reduced for *building*; obviously, by using only a subset of the data, the variability of the appearance of settlements is no longer represented as well as before. For the smallest training dataset (experiment 5) the OA drops to 70% and the mean F1-score to 45%. As also shown in figure 5(f) the classifier can just separate coarse structures like rural and urban areas, but a differentiation between the different classes of vegetation is not possible. A classifier only trained on a very small dataset that only consists of imagery from one season does not generalize to the level of an entire state. To summarize, the performance of the classifier becomes much better with an increasing amount of data being used for training. Both, the size of the area and the variability of the acquisition dates have a high impact. Generally speaking, classes such as *forest* and *grassland*, the appearance of which varies between the seasons, are affected more by a reduction of the amount of training samples. For *water* it might be beneficial to separate sea and inland water bodies, these latter findings have to be taken with care, however, because they are based on a relatively small dataset. Table 3 also gives quality indices for the larger reference dataset R_2. On this dataset, the OA and the F1-scores are worse by approximately 10-15%. The actual numbers are not conclusive because this reference is affected by label noise in the order of the observed differences (8%; cf. Section 4.1.1). However, the observations w.r.t. the trend in the quality indices is confirmed: the larger the area and the more epochs are used, the better the classification results. Thus, the availability of free satellite data at high temporal frequency as well as the use of existing maps for the automatic generation of training labels can improve the prospects of classification considerably.



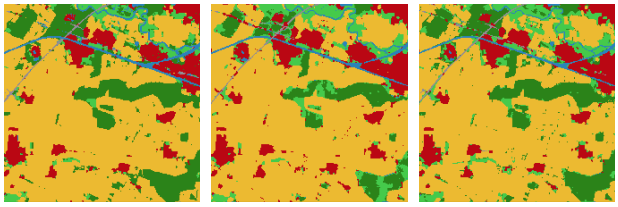
(a) 913 / 14 (b) 913 / 4 (c) 913 / 1

(d) 159 / 14 (e) 9 / 14 (f) 9 / 1

Figure 5. Exemplary prediction results on the tile corresponding to R_1 for different experiments from table 3. Captions: number of tiles / epochs used for training. Colour code: cf. figure 3.

4.3 Evaluation: Influence of Label Noise

To evaluate the impact of different amounts of label noise on the results, we produced eight variants of the reference with four different levels of simulated noise as described in Section 4.1.4. In all cases, we used training data from all tiles and 14 epochs. The evaluation results based on the corrected reference R_1 are shown in table 4. Figure 6 shows exemplary classification results for three noise levels.



(a) 5% (b) 10% (c) 30%

Figure 6. Exemplary prediction results for the tile in R_1 for three different levels of label noise. Colour code: cf. figure 3.

The results show a high level of robustness to increased noise levels. The maximum decrease is 4.3% in OA and 7.1% in mean F1-score compared to the results without simulated noise. Negatively affected classes are *forest*, with a decrease in F1-score of up to 15%, *sealed area* (up to 19%) and *grassland* (up to 14%). However, there is no clear pattern of decreasing accuracy with increasing label noise; for instance, the F1-score of *water* increases by up to 12% for most experiments and the one of *grassland* by up to 5% for some of the experiments. Classes that are difficult to classify (indicated by a low F1-score even without simulated label noise, e.g. *grassland* or *sealed area*) are affected to a slightly larger degree than others. In general, the results show that the distribution of noise has a larger impact on the results than the actual amount, which can be deduced from the fact that the variation of quality indices between experiments with the same amount of simulated label noise is larger

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-V-3-2021-181-2021 | © Author(s) 2021. CC BY 4.0 License.

Exp.	Tiles	Dates	F1-scores on R_1 [%]						R_1 [%]		R_2 [%]	
			bld.	*sld.*	*agr.*	*grl.*	*wat.*	*for.t*	avg. F1	OA	avg. F1	OA
0	913	14	86.7	45.6	96.3	63.9	66.6	91.4	75.1	90.4	65.0	78.5
1	913	4	87.3	38.7	95.4	50.3	74.0	43.9	64.9	81.1	59.1	70.7
2	913	1	87.4	25.2	91.5	43.1	71.2	5.4	54.0	72.6	43.8	55.0
3	159	14	73.3	37.1	95.0	46.3	81.6	69.4	67.1	83.6	47.7	68.5
4	9	14	74.7	53.9	92.1	32.9	76.2	14.2	57.3	75.8	42.3	62.2
5	9	1	68.0	30.4	84.6	13.9	70.5	0.0	44.6	70.8	33.6	53.1

Table 3. Evaluation of land cover classification with different training datasets. Exp.: Experiment number. OA: Overall Accuracy.

Exp.	Noise [%]	F1-scores [%]						avg. F1 [%]	OA [%]
		bld.	*sld.*	*agr.*	*grl.*	*wat.*	*for.*		
0	-	86.7	45.6	96.3	63.9	66.6	91.4	75.1	90.4
6	5	87.2	39.3	95.4	54.1	62.2	83.7	70.3	87.9
		88.4	38.7	95.4	58.5	73.0	76.8	71.8	86.1
7	10	79.6	37.6	95.8	52.9	71.1	76.1	68.8	85.7
		81.3	41.1	96.0	64.5	73.4	91.4	74.6	89.3
8	20	86.7	43.0	96.3	68.9	70,4	91.2	76.1	90.4
		86.0	43.0	96.0	66.0	72.2	88.9	75.4	89.4
9	30	86.9	42.9	95.3	67.0	78.7	87.3	76.3	89.2
		77.6	26.6	94.9	49.3	74.4	85.0	68.0	86.7

Table 4. Evaluation of land cover classification different amounts of simulated noise based on reference R_1. The first row corresponds to experiment 0 in table 3. OA: Overall Accuracy.

than the one between the best results at each noise level. For example, the OA for the first experiment with 30% additional noise is only 1% worse than the one achieved without simulated noise, whereas the difference between this result and the one of the second experiment at that noise level is 2.5%. Our results indicate that the FCN is robust to noise to a relatively high degree, especially for classes with enough samples or that are clear to distinguish (like *agriculture* or *water*). The level to which the result is affected seems to depend more on the distribution of the label noise than on its actual amount. Again, these numbers have to be taken with care because they are only based on a relatively small reference dataset.

4.4 Evaluation: Cosine Loss for Feature Similarity

To evaluate the impact of the cosine loss, we conducted a set of experiments using different variants of the loss function as described in table 2. Table 5 shows the evaluation results based on reference R_1.

In general, the influence of the cosine similarity layer is relatively low. If the early layer f4 is included, the results are worse than in the other cases; it would seem that this early intermediate representation is not general enough for the network to be forced to form well-shaped clusters in feature space. If the cosine loss is applied in the layer having the lowest resolution (f3), the OA and the mean F1-score are identical to the one achieved without the cosine loss, but there is another distribution of class-specific F1-scores. Only the results achieved when the last convolutional layer (f1) is used as input to the cosine

Variant	F1-scores [%]						avg. F1 [%]	OA [%]
	bld.	*sld.*	*agr.*	*grl.*	*wat.*	*for.*		
Cr-En	86.7	45.6	**96.3**	63.9	**66.6**	91.4	75.1	**90.4**
CL-f1	**89.6**	**60.7**	**96.3**	66.0	70.2	86.0	**78.1**	89.5
CL-f13	85.1	34.7	96.0	63.2	79.5	92.4	75.1	**90.4**
CL-f124	82.3	13.6	93.5	62.4	**79.7**	91.9	70.6	87.0
CL-f1234	84.4	37.6	**96.3**	64.1	72.7	88.6	74.0	89.3

Table 5. Results for land cover classification of variants of the cosine loss on dataset R_1. OA: Overall Accuracy. Best scores are printed in bold font.

similarity loss show the desired effect of improving the results for the underrepresented classes, with an increase in the mean F1-score of 3%. The largest increase in F1-score is observed for *sealed area* (+15%), the class covering the smallest percentage of the area, followed by *water* (+3.6%), *building* (+2.9%) and *grassland* (+2.1%). Only *forest* decreases by 5.4%, which is responsible for the small decrease in OA of 0.9%.

We also analyse the distribution of the cosine similarities for some classes depending on the variant of the cosine loss used in training. Table 6 shows the mean cosine similarity and its variance for the classes *sealed area, agriculture* and *water* at layers f1 and f3 for three of the variants. In addition, table 7 shows the cosine similarity between the mean features vectors from f1 for all classes for experiments *CL-f1* and *CL-f13*.

Even without the cosine loss the features of a class have a high cosine similarity (between 0.79 and 0.98). The mean cosine similarity and its variance at the last layer (f1) are related to the class accuracies: *agriculture*, a class with high mean cosine similarity and a low variance has a high F1-score. A class with a lower mean cosine similarity and a higher variance, such as *sealed area*, achieves a low F1-score. As would be expected, using the cosine loss increases the similarity in the layer to 0.99 - 1.00 with a small variance; the other layers are also affected, but to a lesser degree. This is another indicator that the cosine loss does lead to well-defined clusters, which can support the classification if it occurs in the last layer of the network (f1). The cosine similarity between the mean feature vectors of the last layer (f1) can be interpreted as an indicator for the similarity of classes. For instance, table 7 shows a high cosine similarity between the mean feature vectors of *grassland* and *agriculture*, two classes that have a similar appearance at least in some parts of the vegetation cycle. Table 7 also shows that the cosine similarity between the mean feature vectors of the classes increases significantly if the cosine similarity loss is used. It would seem that the cosine similarity does not only lead to more compact clusters, but also to smaller differences between the clusters. However, as the variance of the similarities becomes even smaller, the separation of the clusters is still possible.

Layer	Variant	*sld.*		*agr.*		*wat.*	
		mean	var	mean	var	mean	var
f3	*Cr-En*	0.81	0.07	0.82	0.13	0.80	0.03
	CL-f1	0.87	0.06	0.88	0.07	0.87	0.02
	CL-f13	0.99	$5{\cdot}10^{-4}$	0.99	$4{\cdot}10^{-4}$	0.99	$2{\cdot}10^{-4}$
f1	*Cr-En*	0.79	0.26	0.98	0.08	0.98	0.02
	CL-f1	0.99	$3{\cdot}10^{-4}$	1.00	$1{\cdot}10^{-4}$	1.00	$5{\cdot}10^{-5}$
	CL-f13	0.76	0.24	0.98	0.06	0.95	0.06

Table 6. Mean and variance of cosine similarities for three classes with (*CL-f1*, *CL-f13*) and without (*Cr-En*) cosine loss.

Overall, this analysis indicates that some improvement for the underrepresented classes can be achieved if the cosine similarity loss is applied to the features just before the final classifica-

	bl.	*sl.*	*ag.*	*gr.*	*wt.*	*fr.*
bl.		0.68	0.26	0.48	0.18	0.24
sl.	0.99		0.73	0.82	0.20	0.32
ag.	0.97	0.99		0.90	0.12	0.20
gr.	0.99	1.00	1.00		0.22	0.47
wt.	0.96	0.97	0.96	0.97		0.31
fr.	0.98	0.99	0.97	0.99	0.96	

Table 7. Cosine similarities between mean feature vectors of layer f1 for *CL-f1* (blue) and *Cr-En* (green).

tion layer. Clustering in other layers does not help in that respect. It would seem that the clusters for the individual classes become more similar, but so do the cluster centres. It remains to be investigated whether other losses leading to more compact clusters should be preferred.

5. CONCLUSION

In this paper, we investigated how the generalization performance of a FCN can be improved by using large amounts of data affected by label noise or by an additional constraint for feature similarity in the loss function. The generalization performance of the model becomes better the more data is used during training. Both, the size of the area and the used acquisition dates have an equally high impact on the performance. If tested on a specific date this model achieves comparable results with a classifier trained on data of that specific date. Experiments with simulated label noise showed that the FCN is robust to a high degree of label noise. In our experiments the amount of noise was not correlated with the decrease of the model performance. We conclude that the noise distribution is more important, especially for classes that are difficult to classify anyway. The experiments with the cosine loss showed that, with the last convolutional layer as input, the results improve for the under-represented classes, a similar observation as in (Yang et al., 2020). The question, whether the cosine loss helps to achieve a better clustering remains open for future research, because in our experiments the inter-class similarity increased with the intra-class similarity, too.

Future research should investigate the integration of methods to cope with label noise coming from maps, e.g. by reducing the impact of uncertain samples in the loss function (Frenay and Verleysen, 2014). The goal is to use even larger amounts of training data to further increase the generalization performance and decrease the impact of label noise. We also plan to compare the cosine similarity loss with other constraints in the loss function which focus on increasing the intra-class similarity and also decrease the inter-class similarity, e.g. by using the Euclidean distance as a similarity measure. Such a constraint can further help to form compact clusters in feature space that, on the one hand, increases the accuracy for the minority classes and, on the other hand, allows to compare pixels based on the feature difference. These differences could be used, for instance, to detect class changes between pixels of the same area observed at different time steps and thus help to update outdated maps.

ACKNOWLEDGEMENTS

We thank the Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN) for providing the data of the geospatial database and for their support of this project.

References

Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (AdV), 2008. ATKIS®-Objektartenkatalog für das Digitale Basis-Landschaftsmodell 6.0. Available online (accessed 13 April 2021): http://www.adv-online.de/GeoInfoDok/GeoInfoDok-6.0/Dokumente/.

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning*. 1^{st} edn, Springer, New York (NY), USA.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834–848.

Drory, A., Avidan, S., Giryes, R., 2018. On the resistance of neural nets to label noise. arXiv:1803.11410.

Fletcher, K., 2012. *Sentinel-2: ESA's optical high-resolution mission for GMES operational services*. ESA SP-1322/2, ESA Communications, Noordwijk.

Frenay, 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.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press, Cambridge, MA, USA.

Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2, 1735–1742.

Hoeser, T., Bachofer, F., Kuenzer, C., 2020. Object detection and image segmentation with deep learning on Earth observation data: A review - Part II: Applications. *Remote Sensing*, 12(18). Paper 3053.

Ioffe, S., Szegedy, C., 2015. Batch normalization: accelerating deep network training by reducing internal covariant shift. *International Conference Machine Learning (ICML)*, 37, 448 – 456.

Johnson, J. M., Khoshgoftaar, T. M., 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1). Paper 27.

Kaiser, P., Wegner, J. D., Lucchi, A., Jaggi, M., Hofmann, T., Schindler, K., 2017. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11), 6054–6068.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems (NIPS)*, 1, 1097–1105.

Li, P., He, X., Qiao, M., Cheng, X., Li, Z., Luo, H., Song, D., Li, D., Hu, S., Li, R., Han, P., Qiu, F., Guo, H., Shang, J., Tian, Z., 2020. Robust deep neural networks for road extraction from remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 1-16. Early Access.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P., 2017. Focal loss for dense object detection. *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431 – 3440.

Maas, A., Rottensteiner, F., Heipke, C., 2019. A label noise tolerant random forest for the classification of remote sensing data based on outdated maps for training. *Computer Vision and Image Understanding*, 188. Paper 102782.

Mnih, V., Hinton, G., 2012. Learning to label aerial images from noisy data. *International Conference Machine Learning (ICML)*, 567–574.

Patel, M., 2020. Notes on implementation of cross entropy loss. https://github.com/meet-minimalist/Notes-on-cross-entropy-loss (accessed 29 April 2020).

Postadjian, T., Le Bris, A., Sahbi, H., Mallet, C., 2017. Investigating the potential of deep neural networks for large-scale classification of very high resolution satellite images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Science*, IV-1/W1, 183–190.

Ren, Y., Zhang, X., Ma, Y., Yang, Q., Wang, C., Liu, H., Qi, Q., 2020. Full convolutional neural network based on multi-scale feature fusion for the class imbalance remote sensing image classification. *Remote Sensing*, 12(21). Paper 3547.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 234–241.

Schmitz, S., Weinmann, M., Weidner, U., Hammer, H., Thiele, A., 2020. Automatic generation of training data for land use and land cover classification by fusing heterogeneous data sets. *Wissenschaftlich - Technische Jahrestagung der DGPF*, 29, 73–86.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815 – 823.

Shi, W., Zhang, M., Zhang, R., Chen, S., Zhan, Z., 2020. Change detection based on artificial intelligence: State-of-the-art and challenges. *Remote Sensing*, 12(10). Paper 1688.

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

Ulmas, P., Liiv, I., 2020. Segmentation of satellite imagery using u-net models for land cover classification. arXiv:2003.02899.

Voelsen, M., Bostelmann, J., Mass, A., Rottensteiner, F., Heipke, C., 2020. Automatically generated training data for land cover classification with CNNs using Sentinel-2 images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B3-2020, 767–774.

Wang, Y., Yao, Q., Kwok, J. T., Ni, L. M., 2020. Generalizing from a few examples: A Survey on few-shot learning. *ACM Computing Surveys*, 53(3). Paper 63.

Yang, C., Rottensteiner, F., Heipke, C., 2020. Investigations on skip-connections with an additional cosine similarity loss for land cover classification. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Science*, V-3, 339–346.

Zhang, Z., Guo, W., Li, M., Yu, W., 2020. GIS-supervised building extraction with label noise-adaptive fully convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 17(12), 2135-2139.

Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8–36.