EFFICIENT TRAINING DATA GENERATION BY CLUSTERING-BASED CLASSIFICATION

Melanie Böge*, Dimitri Bulatov, Denis Debroize, Gisela Häufel, Lukas Lucks

Fraunhofer IOSB, Gutleuthausstrasse 1, 76275 Ettlingen, Germany,
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation

Commission III

KEY WORDS: Hierarchical Clustering, Classification, Training Data, CNN, GUI, Labeling

ABSTRACT:

Insufficient amount or complete absence of reference data for the training of classifiers is a general topic. Especially the state-of-the-art deep learning approaches have to deal with the availability or adaption of this reference data to produce the reliable results they are designed for. This paper will pursue different approaches according to the absence of training data for land cover classification from aerial images. First, we will analyze the performance of traditional classification in the absence of reference data using clustering techniques and salient features for the assignment of semantic labels. Second, we will transfer the results as training data to a DeepLabv3+ CNN with pre-trained weights to demonstrate the usability of the generated training data. Third, we expand the clustering approaches and combine them with a Random Forest classifier. Finally, if user interaction and manual annotation of training data are still necessary, we also introduce our labeling GUI that enables a simple, fast, and comfortable training data generation with only a few clicks. To evaluate our procedure, we used two datasets, including the Vaihingen benchmark, for which ground truth is available. Without any interactive steps except setting a few algorithm paremeters, we achieved an overall accuracy of 75% using the Deeplab method with image data only.

1. INTRODUCTION AND RELATED WORK

Modern supervised classification approaches require high amounts of labeled examples. In particular, deep-learning-based approaches are known to be data-hungry. Supervised learning in Airborne Remote Sensing often addresses land cover and material classification. These tools open the door to many applications, such as digital twins necessary for urban planning, crop monitoring, and change identification in disaster management and requirements analysis in military content. It is undesirable and waisting resources for such essential applications to manually carry out that time-consuming and labor-intensive annotation of training examples. Therefore, possibilities for quick labeling must be developed.

Overall, we identified four ways of automatized training data annotation: One is the application of already available benchmark datasets, such as patches from Vaihingen (Cramer, 2010) and Potsdam (Rottensteiner et al., 2012), carefully labeled within the ISPRS 2D semantic segmentation contest, but also the MUUFL dataset (Zare et al., 2016) for classification of hyperspectral images. Also, in Computer Vision, there are several annotated datasets (Demir et al., 2018). For the dataset at hand, transfer learning is typically used for semantic segmentation. Since the inputs (illumination, camera angle, number of channels) and the outputs (classes) of the dataset at hand are not necessarily the same as in the original dataset, many examples must be annotated to finetune the dataset previously obtained solution. However, this brings us again to the problem at the beginning. The second possibility is to rasterize vector data that is typically extractable from freely available shapefiles (e.g., Open Street Map data) (Mills et al., 2015), and to combine them with simple features. This approach was pursued in (Bulatov et al., 2019, Häufel et al., 2018, Huang et al., 2015, Kaiser et al., 2017), and has the advantage that model assumptions about classes in question are incorporated efficiently. For example, we start at tree trunk positions annotated by experts (e.g., by land surveying offices) and use the elevation data to create training examples for the class tree. The question here is to choose the elevation threshold. Most easy-to-classify points can be recognized even by shallow models, but regions close to the border between tree and background class will always be a misclassification source. The third possibility is to use generative techniques, where the input data to be classified can be transferred to the target domain (Bodensteiner et al., 2018, Mirza and Osindero, 2014). These techniques can produce artifacts in the input data, which can subsequently lead to incorrect classifications. Moreover, training generative networks are known to be cumbersome.

On these grounds, we opted for the fourth strategy, to use un-, semi-, and half-supervised approaches to create clusters of non-labeled data and assign it cluster-wise to one of the semantic classes. Also here, there are many sources in the literature (Häufel et al., 2019, Lasserre et al., 2006, Ma et al., 2018, Xiong et al., 2010). In (Santiago et al., 2020), sophisticated approaches were used for clustering, after which a deep-learning-based method has been used to assess confusion matrices. The work of (Lasserre et al., 2006) was probably the first one in establishing the theory of the "discriminatively trained" generative model. In other words, the quality of the generative model depends on the reality fidelity of the data distribution – an improvement that discriminative training may provide. To the same extent, the work of (Lasserre et al., 2006) establishes a connection to the world of generative methods, (Ma et al., 2018) could be counted as a (feature-based) domain adaptation workflow. Despite being denoted as unsupervised, it relies on the abundance of labeled data in the source domain. Using a sequence of spectral translations and covariance adaptations, the

*melanie.boege@iosb.fraunhofer.de

transfer into the target domain is accomplished. Similar is the purpose of (Tuia et al., 2014), where alignments from several corresponding points are employed, and (Zhong et al., 2009), where clusters in source and target domain are compared to learn rules for instance-based domain adaptation. Again, the disadvantage of these algorithms is the necessity of classes to coincide in both domains. (Romero et al., 2015) show how to obtain distinctive features from unlabeled data. The model assumption here is *sparsity of lifetime and population*. The former means that there are only a few classes, the latter limits the involvement of classes among the data points. The model assumption constitutes the loss function for a deep neural network that iteratively learns the layers' weights. This strategy is known as layer-wise greedy unsupervised pretraining network (Bengio et al., 2007). The features are forward-propagated by the weight and are subject to the Support Vector Machine, a classifier known to cope with only a few labeled points.

The functionality of the layer-wise greedy unsupervised pretraining resembles sparse autoencoders (Abdi et al., 2017, Li et al., 2016, Licciardi et al., 2009, Yao et al., 2016) and many others. For example, the weakly supervised approach of (Yao et al., 2016) starts by decomposing a satellite image (dataset) into non-overlapping fragments (tiles, for which only a subset of classes is annotated), and these in turn into regions. Only a very few regions classified pixel-wise are available. The autoencoder determines features and is controlled on the one hand by a consistency term penalizing the difference between the input and the decoder output and, on the other hand by a possible sparse coding. This coding is repeated for many iterations until the resulting 720 entries can easily separate the classes. Another two measures extend this observation. First, the weak classification should support the tile contents. A class not appearing in the annotation of the parent tile should not be assigned to a class. Second, a spatial consistency is built, encouraging the geometrically neighboring regions and those possessing similar codes to be assigned to the same class.

1.1 Contribution

Since the analysis of land-covering entities is an essential task for many types of applications, land cover classification methods are widely applied. In our work, we are interested in four main classes: building, road, high vegetation (trees), and low vegetation (meadow), representing the prevailing object types in urban land cover map.

As mentioned in Sec. 1, efficient classification tools most commonly depend on the quality and amount of training data. But often, this training data is either not available or can only be produced manually, tying expensive resources. In this paper, we will introduce approaches to deal with the absence of training data. We use salient features to perform classification without reference data. Further, we show the usability of this approach by transferring the generated training data to a DeepLabv3+ CNN. Nevertheless, to meet the demand for manually labeled, more data-specific reference data, we present an approach that reduces the necessary user interaction and makes it very convenient.

We address four different tasks:

1. In the case of insufficiency or absence of training data, we will analyze how traditional cluster-based and hierarchic classification performs on semantic assignments using rules for salient features.
2. Further, we will assess whether these results can be reliably used as training data by transferring them to a CNN. So, we will produce a sufficient amount of training data for these strong networks without reference data.
3. Land cover classification using image data exhibiting shaded regions may result in misclassifications concerning these areas. Shadow indices will be introduced.
4. In assigning semantic classes by user interaction, we will introduce a method that allows high-speed semantic labeling of the clusters and makes the necessary user interaction very small and convenient by using our GUI.

1.2 Organization

We present in Sec. 2 the main tools allowing us to assign semantic classes to large portions of input data, which may include different channels (multi-spectral, elevations, gradient-based features), but is supposed to be geo-referenced. The results are presented in Sec. 3. Finally, Sec. 4 summarizes our contribution and outlines future research directions.

2. METHODOLOGY

In Sec. 2.1, we introduce an approach of hierarchic clustering to group similar parts of the image data to differentiate between the classes of interest. To accelerate the process and to be able to include also neighboring properties, we work not on the pixel level of the image but a segmentation level. Arbitrary segmentation method could be thinkable, but we preferred a superpixel segmentation (e.g., compact superpixels due to (Veksler et al., 2010)). This method ensures that the superpixels' borders approximately coincide with those of the instances to be classified and since their sizes are very similar, they will be given equal treatment within classification apporach. With superpixels' features at hand, standard clustering algorithms are used, after which we need to assign a class to every cluster, in order to produce usable training data. One way to assign labels to a cluster is to use salient features, as we explain in detail in Sec. 2.2 and 2.4. However, suppose some classes are difficult to assign (which can happen, for example, in the case of a high intra-class variation). In that case, an interactive labeling tool is presented in Sec. 2.3. Direct integration of the labeling results into the deep learning algorithm (such as DeepLabv3+) is described in Sec. 2.5. We mainly focus on the possibility of transferring (semi-) automatically generated training data to a CNN and demonstrating a solid evaluability of classification. For that reason, we run DeepLabv3+ only with the training data from clustering and semantic assignments using salient features. An alternative, sensible for the situation there is still not sufficient training data available, represents a Random Forest algorithm allowing to use of hand-crafted features, see Sec. 2.6. For a better overview of the methodology that we introduce in the next chapter, Fig. 1 depicts a flowchart.

2.1 Hierarchic Clustering

For clustering, we will make superficial assumptions about the present urban area. We want to get by with very few and intuitive properties in a first approach during the clustering. Besides the given color channels, we use the relative elevation, the planarity measure (Gross and Thönnessen, 2006), and Normalized Difference Vegetation Index (NDVI), which we concatenate into a six tuple – our feature vector. First of all, we

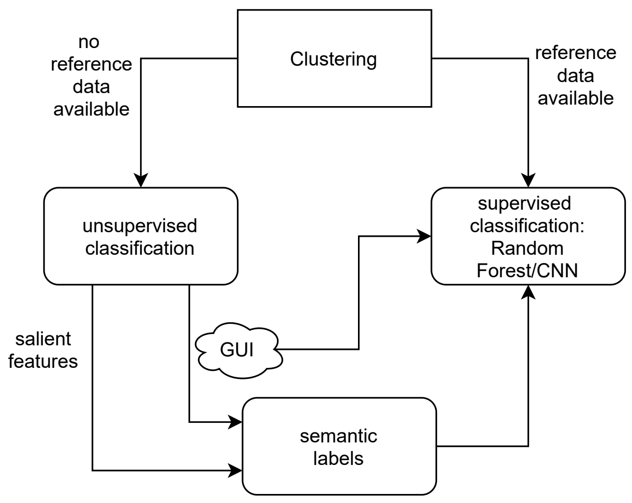

Clustering
no reference data available
reference data available
unsupervised classification
supervised classification: Random Forest/CNN
salient features
GUI
semantic labels

Figure 1: Flowchart of the following methodology.

assume that each urban area depends on two or three primary elevation levels. The actual number depends on the underlying data and can be chosen heuristically. Areas of homogeneous artificial objects like settlements of new buildings typically work fine with two elevation levels. Patchy areas of further building types like large buildings, garages, and sheds work better with three elevation levels.

There are high objects in two primary classes, like buildings or trees, and low objects, like meadows and roads. With three initial classes, a middle layer is added. Further, buildings and streets are artificial objects with high planarity, while bushes, meadows, and trees feature naturally less planar values. Therefore, the first clustering step is to separate high from low objects by weighing the elevation parameter in the feature vector much more than the other parameters. For clustering, we use a k-means approach to separate the elevation levels.

In the following step, for every elevation cluster, the elevation characteristic is more or less negligible, while planarity and NDVI are given larger weights to separate plants from artificial objects. Since this clustering step aims to separate more clusters, we use a Gaussian-Mixture-Model (Reynolds, 2009) for clustering. The advantage is that the algorithm terminates if it cannot identify new clusters and is, in addition to that, less dependent on predefined parameters. So, we tested different numbers of clusters during the second classification step to analyze the difference between very few and a large number of target clusters. In addition to that, the feature weights can also be adapted according to their elevation level.

2.2 Salient Features for Semantic Assignments

Separating the data into only a few clusters allows us to revert to the initial clustering idea and apply rules for the features to assign a label to each class. We will address only the case of two primary classes, but in three primary categories, similar rules can be derived. We denote trees to be one of the clusters from high elevation with a maximum mean NDVI value. Buildings are denoted to be one of the clusters from high elevation with maximum mean planarity value. In the same way, we proceed for the low elevated clusters: Low elevation and maximum mean NDVI mean meadow, low elevation, and maximum mean planarity road. The more clusters are extracted, the more rules are necessary to assign a label for each class. With this relatively simple approach, we generated training data with a minimum of user pre-settings. We will transfer this training data to a DeepLabv3+ CNN to analyze its performance.

Nevertheless, the strategy of assigning semantic labels based on salient features includes three main obstacles: First, the necessary rules may become very data-specific. Second, the result of labeling depends on the order of treated clusters. Third, some clusters include segments that belong to different classes, e.g., segments lying in deep shadows. We will analyze an approach to deal with shadowy data later in Sec. 2.6.

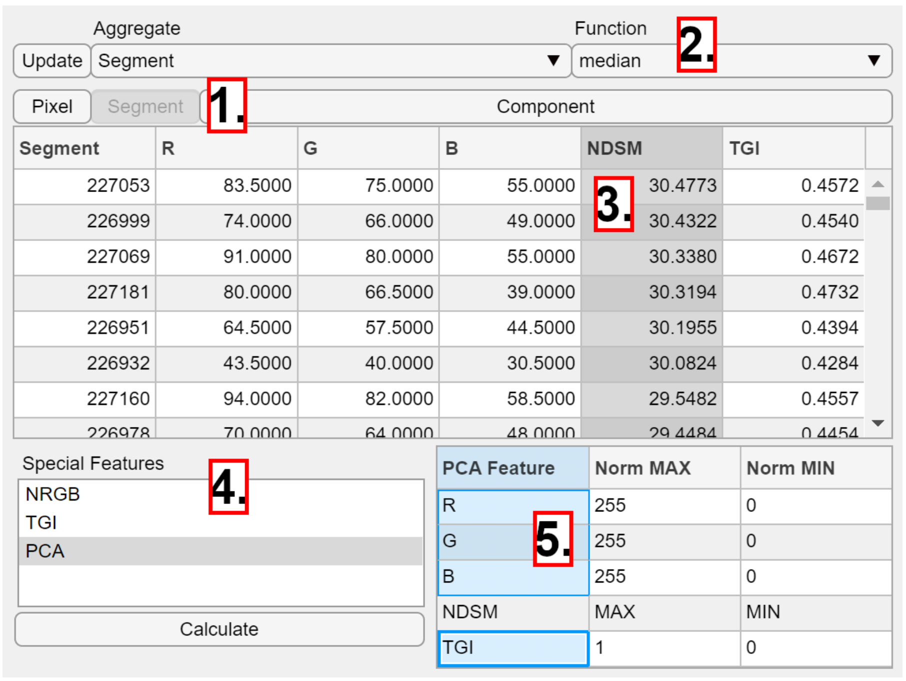
In advance, we want to address the issue that manual user interaction is still required to create (further) training data. Therefore, we decided to invest only a moderate and highly automatized interactive effort to create and multiply reliable training data using our GUI.

2.3 Interactive GUI-based application

As pointed out above, the determination of suitable features can be very data-specific. In some cases, that high-quality assignment data need to be labeled by hand. We designed a GUI workflow that supports this necessary manual interaction as comfortably and efficiently as possible. Our GUIs workflow follows the approach of hierarchic clustering with the subsequent creation of connected components. These components can be quickly assigned by sorting them by size and offering them successively to the supervising human. The assignment can be accomplished by simple keyboard shortcuts (*one-click-assignment*). This procedure is efficient for two reasons: Firstly, the descending ordering by size helps assign the most significant amount of pixels by the least required interaction. Secondly, as the offering is made cluster-wise, the list of expected labels is usually in the range of two to four labels (depending on the quality of the clustering and the complexity of the data as well as target classes). These supportive measures make it easier to focus on the detection of the correct label.

The GUI supports various data handling and cross-compatibility functions to external reference data that can be loaded and adapted by the user. The user may load, inspect and transform the input data into the GUI. The amount of features to load is not limited and can be done for single-layer features (e.g., elevation) or multilayer features (e.g., hyperspectral data). In the case of multilayer data, each layer is handled as a single layer. The GUI offers a table view of all available features to verify successful import, see Fig. 2(a). Featurewise ordering (de- or ascending) of the table view offers a quick overview of extreme values and possible outliers. In addition, the user can analyze correlations between the features. In Fig. 2(a), the highest areas of the scene are around 30 m above ground. The Triangular Greenness Index (TGI) column on the right shows values around 0.45. This correlation suggests that the highest object in the scene is not a green tree but, more likely, a tall building. Especially for raw input features (like RGB or hyperspectral channels), it can create new features out of a transformation of existing features. Therefore the GUI provides a list of transformation techniques. Possible transformations are RGB-normalization (NRGB, for reducing the influence of lighting and shadows), triangular-greenness-index (TGI, to highlight green vegetation), and feature-specific principal-component-transformation (PCA, for feature reduction).

After loading superpixels, segment-wise features are computed by applying statistical measures (such as mean or median) to

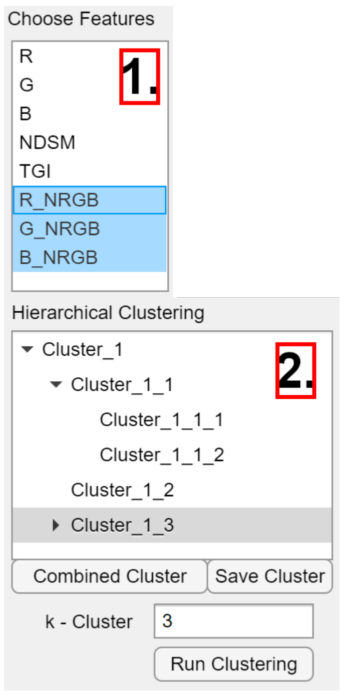


Segment	R	G	B	NDSM	TGI
227053	83.5000	75.0000	55.0000	30.4773	0.4572
226999	74.0000	66.0000	49.0000	30.4322	0.4540
227069	91.0000	80.0000	55.0000	30.3380	0.4672
227181	80.0000	66.5000	39.0000	30.3194	0.4732
226951	64.5000	57.5000	44.5000	30.1955	0.4394
226932	43.5000	40.0000	30.5000	30.0824	0.4284
227160	94.0000	82.0000	58.5000	29.5482	0.4557
226978	70.0000	64.0000	48.0000	29.4484	0.4454

Special Features
- NRGB
- TGI
- PCA

PCA Feature	Norm MAX	Norm MIN
R	255	0
G	255	0
B	255	0
NDSM	MAX	MIN
TGI	1	0

(a) Feature table view from the GUI. The 'Segment' tab is active (1), so the table shows the superpixel and the averaged feature values. Selection menu for switching between mean- and median-averaging (2). The NDSM was selected to sort the table (3). Available transformations are shown at (4) and (5).



Choose Features
- R
- G
- B
- NDSM
- TGI
- R_NRGB
- G_NRGB
- B_NRGB

Hierarchical Clustering
- Cluster_1
  - Cluster_1_1
    - Cluster_1_1_1
    - Cluster_1_1_2
  - Cluster_1_2
  - Cluster_1_3

k - Cluster: 3

(b) Feature selection and hierarchic tree structure. (1) selected features, (2) hierarchy tree with expanded sub-cluster 1.

Figure 2: Working with features in the GUI.

the pixel-wise features. For efficient computation, we refer to (Bulatov et al., 2019).

During the clustering procedure, the user can select single or multiple features to run a hierarchic clustering according to Sec. 2.1 but without optional weighting. For example, the user can cluster the whole scene only using the NDSM to separate objects of different elevation levels. Next, e.g., ground objects can be further separated along all NRGB features to divide grass, soil, and street. In the GUI, the hierarchic clustering is supported by feature and hierarchy selection with a visual output of the created sub-clusters. So, the user can test out multiple combinations of used features in a fast way (see Fig. 2(b)). As soon as the clustering satisfies the user's requirements, connected components are formed. For this, neighboring superpixels with the same cluster label are combined to one (super-)segment. To facilitate the data processing on multiple scenes, the user can save (and reload) all necessary parameters and intermediate results as a project file.

After the connected components' generation or import, the user can define the labels for the labeling process, see Fig. 3. The workflow guides the user cluster-wise from the largest component to the smallest, whereby the user can switch to the next cluster. The labeling is performed by clicking on the label name or by pressing the corresponding key. A convenient tool is the label recommendation, which gives an intelligent label prediction. The user can switch between different prediction modes like the "previously chosen" or "most frequent" label.

The GUI provides options to inspect the similarity of labels or clusters, depending on the number of assigned components or the number of assigned pixels in a dendrogram (Fig. 4). Depicting the clusters as a dendrogram (Fig. 4, top), it represents from top to bottom the levels and data separability of the hierarchical clustering. This reflects the suitability of the selected features for separating the data during clustering. The label dendrogram (Fig. 4, bottom) can further be used to check the similarity or separability of the different labels. In the example, superpixels labeled as "Soil" and "Clutter" are more similar concerning their occurrence in different clusters and therefore more difficult to separate than "Street", "House", and "Meadow". The "Tree" label is well separable from the other labels, which can be seen in Fig. 3. Only the "Tree" label is assigned to Cluster_2.

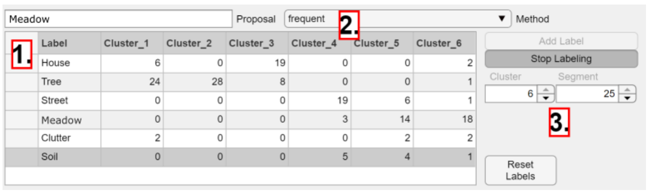
2.4 Semantic Assignments by Reference Data

In the case of available reference data, the resulting clusters could be labeled using the *winner takes all* strategy. We overlaid our achieved clustering results in the form of cluster-indexed superpixels with this reference data. Due to segmentation, each superpixel can cover reference data pixels of more than one land cover class. The superpixel label is determined using the *winner takes all* strategy.

Often, land cover classes are not equally distributed in the data. Hence, there are areas where one land cover class is less present than other land cover classes. This imbalance affects the decision of which cluster to use as training or test data for the subsequently performed classification. For each cluster, we compute the overlap percentage to the corresponding land cover class. Those clusters that are better covered by one referencing land cover class are assigned to the training data. The less covered clusters become test data.

2.5 Classification with DeepLabv3+

We demonstrate the usability of our generated training data for fine-tuning a DeepLabv3+ network (Chen et al., 2018), referred to as DeepLab in the following. As mentioned earlier, our main concern is to demonstrate the transferability of automatically generated training data into a CNN.



Meadow | Proposal: frequent | Method

Label	Cluster_1	Cluster_2	Cluster_3	Cluster_4	Cluster_5	Cluster_6
House	6	0	19	0	0	2
Tree	24	28	8	0	0	1
Street	0	0	0	19	6	1
Meadow	0	0	0	3	14	18
Clutter	2	0	0	0	2	2
Soil	0	0	0	5	4	1

Cluster: 6 Segment: 25

Figure 3: Labeling view: (1) Table with label names and number of assigned components for each cluster; (2) Recommendation and method selection; (3) Active component

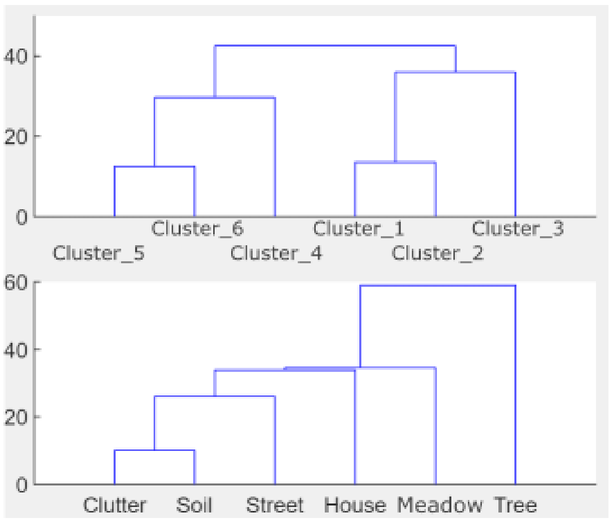


Figure 4: Dendrograms: (top) similarity of clusters; (bottom) similarity of labels

The DeepLab is one of the state-of-the-art networks in semantic image segmentation. The network contains parallel atrous convolution layers with different kernel sizes (Atrous Spatial Pyramid Pooling) to capture contextual information from multiple scales as an encoder. With that structure, the network can generate meaningful multi-scale features sampled over large receptive fields and having high context information. The DeepLab decoder contains deconvolutional and unpooling layers to refine the output of the encoder and other low-level features. This standard procedure leads to very plausible results without over-smoothing borders, mainly because the degree of downsampling remains moderate thanks to atrous convolutions. We use pre-trained weights of *pascal_voc* (Everingham et al., 2012).

2.6 Classification with Random Forest

Alternative to deep neural networks, we are interested in the case when a few salient features are available, which can be used for retrieval of pixels belonging to a certain class, compensating for a scarcity of training data. For Random Forest (Breiman, 2001), we considered as features the raw image channels including NDSM, in the case of availability of elevation data. Note that if only the DSM is available, the relative elevation is computed as in (Bulatov et al., 2014). We computed meaningful combinations of image channels. The most important example is NDVI (or pseudo NDVI, if no near infrared channel is available). Other examples are measures applied to improve the classification in the shadow areas. Here, we experimented with the component V of the HSV colour space; the ratio $(H + 1)/(I + 1)$ proposed by (Tsai, 2006); whereby I is the intensity, the sum of the L and B components of the LAB color space (Murali and Govindan, 2013); and, finally, the component C3 of the C1C2C3 color space, according to (Arévalo et al., 2008).

Besides, we included texture-based features, such as the planarity of the DSM, computed using the implementation of (Gross and Thönnessen, 2006). The entropy (Rafael et al., 2010) is extracted from the grayscale representation of the RGB image. Additionally, we compute morphological profiles for each shadow feature using the structuring elements of five sizes (1 m, 4 m, 10 m, 20 m, and 30 m) (Pesaresi and Benediktsson, 2001). For the superpixel-wise features, we extended the feature vector by the pixel-wise standard deviation, median, and maximum for each feature to increase the stability of classification (Häufel et al., 2019, Bulatov et al., 2019). Altogether, the feature vector has been extended to 54 features to train the Random Forest classifier with 100 decision trees.

3. RESULTS AND DISCUSSION

3.1 Datasets

To demonstrate the usability of our method, we will focus on two different datasets:

Vaihingen: The first dataset is the Vaihingen Benchmark (Cramer, 2010) that consists of 16 areas for training and 17 areas for testing. For all areas, the three channels (near infrared, red and green), the DSM as well as the ground truth data are available.

Ettlingen: We use an own dataset which includes a DSM and a digital orthophoto created by a commercial software (Agisoft Metashape) from high-resolution images. These show Ettlingen, a town in Southern Germany, in quasi-nadir views. The dataset has the resolution of 0.1 m and there no initial ground truth is available. Still, the dataset is very interesting because of large intra-class variation. The selected area contains parts of old town with many buildings of with different forms and textures with shadow areas between them, and at the same town the outskirts of the town, showing fields with different crops, forest paths, hilly areas of Black Forest, and a cemetery.

3.2 Evaluation

We use dataset *Vaihingen* for the evaluation of the initial hierarchic clustering. The visual impression of the cluster classification using few (six) target clusters is promising (see Fig. 5). By assigning semantic labels using salient features, the overall accuracy of all areas compared to the provided ground truth is approximately 75 to 77 %. The only user interaction is the choice of parameters for the clustering. Without any manual labeling, we generated reliable training data. Usually, land cover classification methods require a high amount of labeled data and large feature sets. Semi-supervised classification starts at limited labeled data and continues to learn with the newly emerging data, which must be appropriately prepared. We achieved these good results using only six simple features combined with intuitive rules without any manual labeling.

A result of using a approximately fifty target clusters (the number yielded after termination of the Gaussian-Mixture-Model) is shown in the first column of Fig. 6. We also include detailed views 1 and 2 (Fig. 6, 3rd column). Overall, the results for the Vaihingen dataset are satisfactory, with an overall accuracy between 74% and 79% for different patches yielded after application of Random Forest and additional reference data. The white bordered road area in the detailed view 1 splits up into

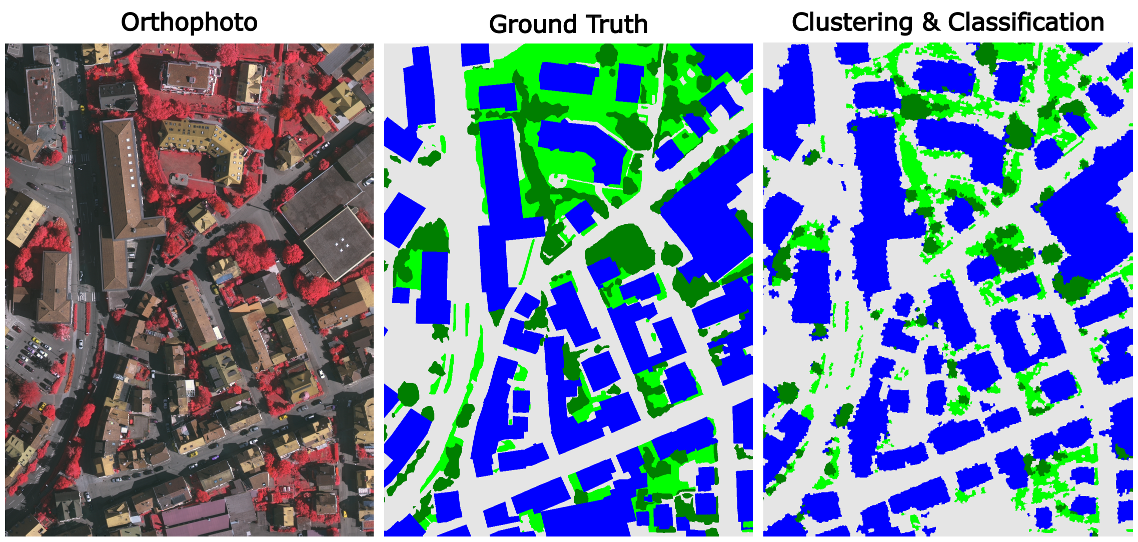


Figure 5: Classification of dataset *Vaihingen* (left) using a hierarchic clustering (middle) and assigning labels with salient features (right).

different clusters caused by deviations in the NDSM. A similar situation occurs in cutout 2 due to influence of shadow. In the reference data, no difference was made between shadowy and non-shadowy regions thus the shadow indices used for hierarchic clustering did not perform optimal.

By using a large number of target clusters (up to approximately fifty, that is the maximum number of Gaussian-Mixture-Model termination) and by using additional reference data and Random Forest classification, we can increase the accuracy to $74 - 79$ %. In Fig. 6, the cutouts of the two detailed views are displayed, showing the results after clustering, Random Forest classification, and the ground truth. Misclassifications often occurred in shaded regions.

Now we want to analyze the transferability of this training data to other test sets applying DeepLab classification. For these experiments, we use the results from only six target classes of Vaihingen dataset, described in the beginning of this chapter. The available ground truth allows evaluating the results of DeepLab. On DeepLab, we used just the color channels for classification. We also experimented on all six feature channels (color channels, elevation, planarity, and NDVI), but it turned out that the pre-trained weights from pascal_voc did not improve the results. We refer to future work for further experiments using other pre-trained weights in DeepLab to evaluate all available features. After 100 training epochs, DeepLab results in the maximum expected accuracy of about 75 % on the test areas and a visually appealing result (see Fig. 7). Also, with (semi-) automatically generated training data, the accuracy provided by the reference data can reliably be transferred to test data. In the case of dataset *Ettlingen*, no ground truth data was available. Thus, we used the GUI-labeled ground truth to train a Random Forest classifier. We also included the introduced shadow features and achieved an overall accuracy of 89 %. Fig. 8 displays a cut out of the classification result based on the GUI labeling. The outlined areas show an example of the correct classification of the shaded meadow and sidewalk.

4. CONCLUSIONS AND FUTURE WORK

We introduced a cluster-based classification that delivers sound results for land cover classification even if no training data is available. For this classification, we only use few distinctive and intuitive plausible features like the color channels, elevation, planarity, and NDVI. We showed that these results could be transferred to CNNs as reference data. In addition to that, we can reach the training data accuracy of approximately 80 % even for unknown test areas. If some kind of reference data (ground truth, OSM, etc.) is available, we can increase this accuracy to approximately 89 %, using supervised classification tools as Random Forest. Depending on the demanded target accuracy, we can include user interaction and manual labeling of training data with our GUI in a very efficient, resource-saving, and comfortable way.

For future work, we will analyze the certainty of the clustered segment assignments. Segments with features that fit the estimated distribution during the clustering are more reliable than those differing significantly from the relevant cluster centers or distribution functions. By weighting the training data itself during the following classification approaches, the accuracy of the results could be increased further. We have found that the joint evaluation of color channels and additional feature channels in DeepLab does not improve. The reason is that the pre-trained weights of the *pascal_voc* used are designed for evaluating the color channels. In the future, we will analyze whether better results can be achieved with other starting weights for the CNN by evaluating all feature channels simultaneously.

For the future development of our GUI, the analyzer dashboard shall provide more tools for clustering evaluation of the certainty of labels and more extensive user support. This could be realized using internal histograms and outlier detection techniques. In addition, the implementation of machine learning models (like Random Forest) would support a better label recommendation or auto-completion.

REFERENCES

Abdi, G., Samadzadegan, F., Reinartz, P., 2017. Spectral–spatial feature learning for hyperspectral imagery classification using deep stacked sparse autoencoder. *Journal of Applied Remote Sensing*, 11(4), 042604.

Arévalo, V., González, J., Ambrosio, G., 2008. Shadow de-

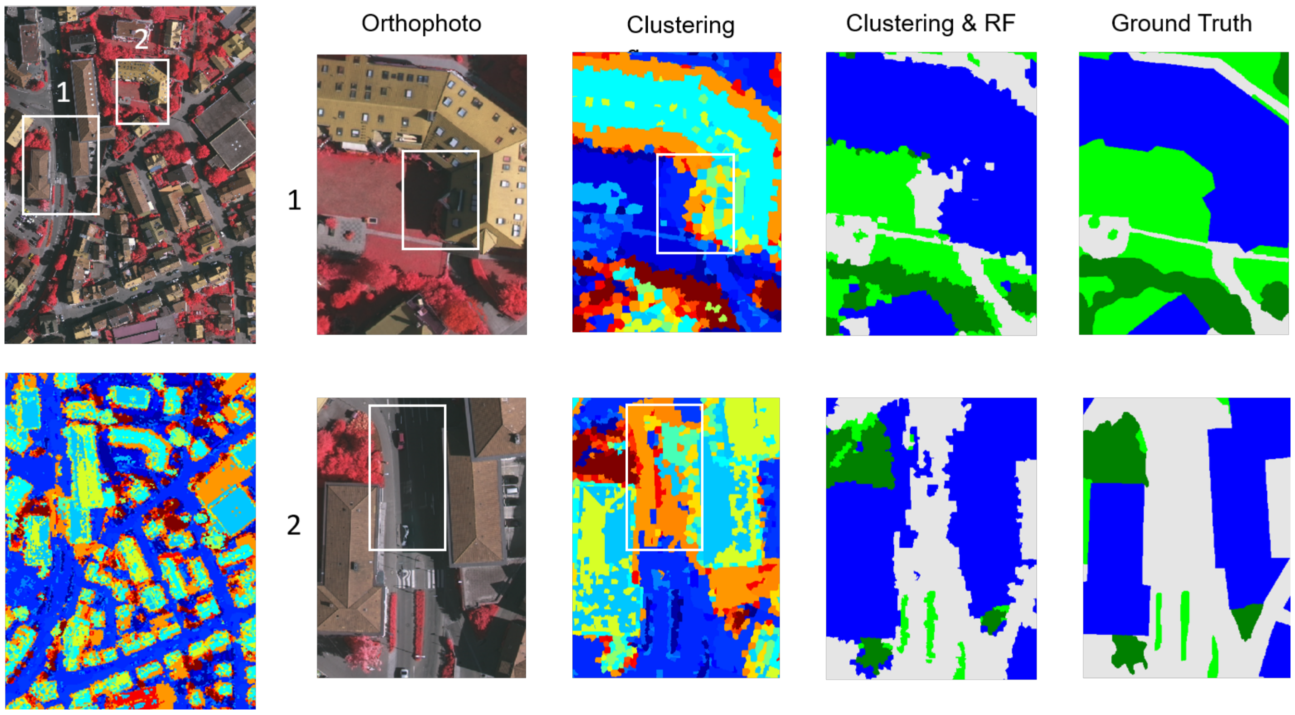
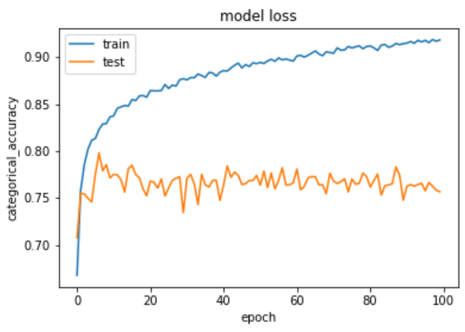
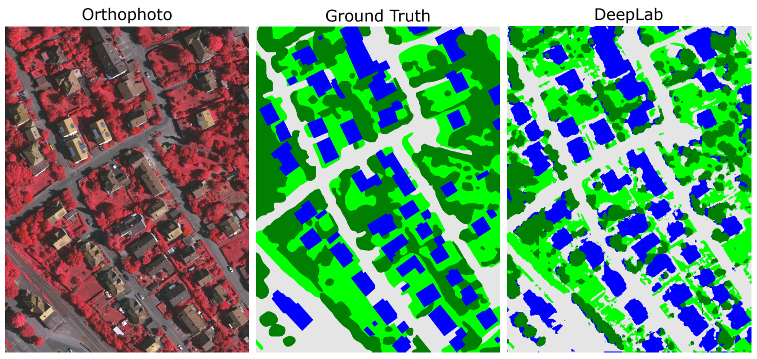


Figure 6: Clustering result including detailed views after hierarchic clustering (random colors) and Random Forest classification result



(a) Accuracy during training with DeepLab



(b) Results of DeepLab classification at test area 14.

Figure 7: DeepLab Classification: Results for test areas of dataset *Vaihingen* after 100 training epochs on corresponding training areas. As reference data we used the results of the hierarchic clustering classification.

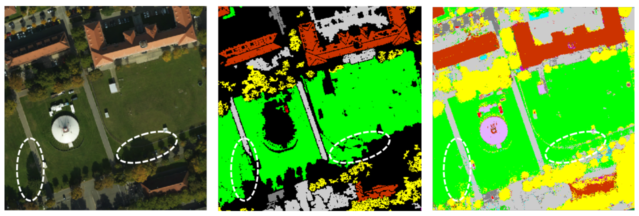


Figure 8: Random Forest classification result based on GUI ground truth labeling including white-bordered areas. Shadow areas are correctly classified as sidewalk and meadow.

tection in colour high-resolution satellite images. *International Journal of Remote Sensing*, 29(7), 1945–1963.

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H. et al., 2007. Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems*, 19, 153.

Bodensteiner, C., Bullinger, S., Arens, M., 2018. Multispectral matching using conditional generative appearance modeling. *15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 1–6.

Breiman, L., 2001. Random forests. *Machine learning*, 45(1), 5–32.

Bulatov, D., Haeufel, G., Lucks, L., Pohl, M., 2019. Land cover classification in combined elevation and optical images supported by OSM data, mixed-level features, and non-local optimization algorithms. *Photogrammetric Engineering & Remote Sensing*, 85(3), 179–195.

Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P., Wernerus, P., 2014. Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. *ISPRS Journal of Photogrammetry & Remote Sensing*, 93, 157–170.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 801–818.

Cramer, M., 2010. The DGPF-test on digital airborne camera evaluation–overview and test design. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation*, 2010(2), 73–82.

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., Hughes, F., Tuia, D., Raskar, R., 2018. Deepglobe 2018: A challenge to parse the earth through satellite images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 172–181.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Gross, H., Thönnessen, U., 2006. Extraction of lines from laser point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36 (Part 3/W49), 86–91.

Häufel, G., Bulatov, D., Helmholz, P., 2019. Statistical analysis of airborne imagery combined with GIS information for training data generation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Häufel, G., Bulatov, D., Pohl, M., Lucks, L., 2018. Generation of training examples using OSM data applied for remote sensed landcover classification. *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 7263–7266.

Huang, X., Weng, C., Lu, Q., Feng, T., Zhang, L., 2015. Automatic labelling and selection of training samples for high-resolution remote sensing image classification over urban areas. *MDPI Remote Sensing*, 7(12), 16024–16044.

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.

Lasserre, J. A., Bishop, C. M., Minka, T. P., 2006. Principled hybrids of generative and discriminative models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, IEEE, 87–94.

Li, E., Du, P., Samat, A., Meng, Y., Che, M., 2016. Mid-level feature representation via sparse autoencoder for remotely sensed scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(3), 1068–1081.

Licciardi, G., Del Frate, F., Duca, R., 2009. Feature reduction of hyperspectral data using autoassociative neural networks algorithms. *IEEE International Geoscience and Remote Sensing Symposium*, 1, IEEE, I–176.

Ma, L., Crawford, M. M., Zhu, L., Liu, Y., 2018. Centroid and covariance alignment-based domain adaptation for unsupervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4), 2305–2323.

Mills, G., Ching, J., See, L., Bechtel, B., Foley, M., 2015. An introduction to the WUDAPT project. *9th International Conference on Urban Climate, Toulouse*.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Murali, S., Govindan, V., 2013. Shadow detection and removal from a single image using LAB color space. *Cybernetics and Information Technologies*, 13(1), 95–103.

Pesaresi, M., Benediktsson, J. A., 2001. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(2), 309–320.

Rafael, C. G., Richard, E. W., Steven, L. E., Woods, R., Eddins, S., 2010. *Digital image processing using MATLAB*. Tata McGraw-Hill.

Reynolds, D. A., 2009. Gaussian Mixture Models. *Encyclopedia of Biometrics*, 741, 659–663.

Romero, A., Gatta, C., Camps-Valls, G., 2015. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3), 1349–1362.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I-3 (2012), Nr. 1*, 1(1), 293–298.

Santiago, J. G., Schenkel, F., Gross, W., Middelmann, W., 2020. An unsupervised labeling approach for hyperspectral image classification. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 407–415.

Tsai, V. J., 2006. A comparative study on shadow compensation of color aerial images in invariant color models. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6), 1661–1671.

Tuia, D., Volpi, M., Trolliet, M., Camps-Valls, G., 2014. Semisupervised manifold alignment of multimodal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(12), 7708–7720.

Veksler, O., Boykov, Y., Mehrani, P., 2010. Superpixels and supervoxels in an energy optimization framework. *European Conference on Computer vision*, Springer, 211–224.

Xiong, B., Zhang, X., Jiang, W., 2010. Semi-supervised classification based on Gaussian mixture model for remote imagery. *Science China Technological Sciences*, 53(1), 85–90.

Yao, X., Han, J., Cheng, G., Qian, X., Guo, L., 2016. Semantic annotation of high-resolution satellite images via weakly supervised learning. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6), 3660–3671.

Zare, A., Jiao, C., Glenn, T., 2016. Multiple instance hyperspectral target characterization. *arXiv preprint arXiv:1606.06354*.

Zhong, E., Fan, W., Peng, J., Zhang, K., Ren, J., Turaga, D., Verscheure, O., 2009. Cross domain distribution adaptation via kernel mapping. *Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1027–1036.