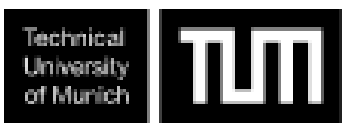
Cartography M.Sc.

Master Thesis

Semi-Supervised Virtual Support Vector Machines with Self-Learning Constraint for Remote Sensing Image Classification

Ozan Tuncbilek

2021



Semi-Supervised Virtual Support Vector Machines with Self-Learning Constraint for Remote Sensing Image Classification

Submitted for the academic degree of Master of Science (M.Sc.) conducted at the Department of Aerospace and Geodesy Technical University of Munich

Author:	Ozan Tuncbilek
Study Course:	Cartography M.Sc
Supervisors:	Dr. Christian Murphy(TUM), Dr. Christian Geiß (DLR) Prof. Dr.-Ing. Günter Strunz (DLR)
Reviewer:	Mahdi Khodadadzadh(ITC)
Cooperation:	German Aerospace Center (DLR)
Chair of the Thesis Assessment Board:	Prof. Dr. Liqiu Meng
Submission Date:	10.09.2021

Statement of Authorship

Herewith I declare that I am the sole author of the submitted Master's thesis entitled:

Semi-Supervised Virtual Support Vector Machines with Self-Learning Constraint for Remote Sensing Image Classification

I have fully referenced the ideas and work of others, whether published or unpublished. Literal or analogous citations are clearly marked as such.

Munich, 10.09.2021 Ozan Tuncbilek

Acknowledgments

First of all, I would like to thank my supervisors from the German Aerospace Center (DLR), Dr. Christian Geiss, Patrick Manuel Aravena Pelizari and Prof. Dr.-Ing. Günter Strunz who envisioned the thesis idea and assisted me through the whole process. Their expertise in machine learning and remote sensing field further supported and motivated me to enhance my methodology and pushed my work to a higher level.

Secondly, I would like to express my appreciation for Dr. Christian Murphy for his constant feedback and support, and his ideas from a cartographic perspective to combine the thesis with its further extensions. Additionally, I would also like to thank the Cartography thesis assessment board for their further guidance and comments during the process.

I would also like to extend my gratitude to TUM and DLR for providing their facilities and setting up a virtual working environment in these difficult times all over the world. Finally, I would like to thank my family and friends who were always supportive and motivating during this period.

Abstract

In real-world applications, it is difficult to collect labeled samples, and supervised learning methods rely on the quality of this labeled training data. Therefore, in this research, a semi-supervised learning approach is developed in order to benefit from the unlabeled samples that can be produced effortlessly. These semi-supervised methods are built on a popular machine learning technique called support vector machine, which is used to classify remote-sensing imagery in this thesis. Moreover, this work aims to enhance the accuracy of the methods in settings with very few labeled samples and deploy a constrained set of unlabeled samples with a self-learning strategy. Additionally, the aim includes model evaluation for existing support vectors and virtual samples. Moreover, the methodology is further extended with an active learning method. This extension involves uncertainty visualizations in order to increase the model accuracy by relabelling the uncertain samples in a prioritized way. To evaluate these models, experimental results were obtained over the city of Cologne, Germany, and the Hagadera Refugee Camp, Kenya from a very high spatial resolution multispectral data set. Results from newly proposed methods showed favorable performance properties, especially on the few labeled samples. Furthermore, the uncertainty of the models was compared with the active learning extension, and this extension also increased the accuracy.

Contents

List of Figures

List of Tables

List of Abbreviations

SVM	Support Vector Machines
SV	Support Vectors
VSVM	Virtual Support Vector Machines
VSV	Virtual Support Vectors
VSVM-SL	Virtual Support Vector Machines with Self-Learning
QP	Quadratic Programming
VC	Vapnik–Chervonenkis
SRC	Structural Risk Minimization
S3VM	Semi-supervised Support Vector Machines
GIS	Geographic Information Science
AL	Active Learning
VSVM-SL-U	Virtual Support Vector Machines with Self-Learning with Unlabeled Samples
VSVM-SL-VU	Virtual Support Vector Machines with Self-Learning with Virtual Unlabeled Samples
SVM-SL-U	Support Vector Machines with Self-Learning with Unlabeled Samples
OA	Overall Accuracy
AA	Average Accuracy

1 Introduction

1.1 Motivation and problem statement

In the last decades, with the advent of high spatial and spectral resolution remote sensing data, land cover classification applications have become one of the main subjects in remote sensing (Lu et al., 2016). Consequently, it triggered the development of many methods to derive thematic classes from image data, and as an outcome, supervised methods became one of the most preferred classification approaches because of their robust and accurate information extraction properties (Geiß et al., 2019) Although it is overly challenging to determine the best method from numerous of existing approaches for a classification problem, Support Vector Machines (SVM) attracted attention regarding the classification of multispectral remote sensing images. As a working principle, SVM set suitable hyperplanes on different classes of labeled data and those samples are projected through a nonlinear transformation from input space to a higher-dimensional space. In that space, support vectors (SV), which are the samples closest to the separating surface, are determined in subject to the optimal hyperplane that maximizes the margin (Geiß et al., 2019; Burges, 1998). Therefore, SVM showed excellent performance due to their, (i) ability to manage high-dimensional feature space; (ii) relevant generalization properties (iii); the uniqueness of the solution (Tuia et al., 2009)

SVMs, as any other supervised method, rely on the quality of the labeled training data. However, this constrains the training set and requires extensive manual efforts regarding human-machine interaction. That is why active learning methods and semi-supervised learning approaches which use unlabeled samples will benefit the classification results, especially with respect to poorly sampled remote sensing applications (Izquierdo-Verdiguier et al., 2012). It is here where we combine self-learning constraints on Virtual Support Vector Machines (VSMV) with a semi-supervised approach to indicate useful information about the underlying data distribution which eventually achieves higher accuracies, especially with small amounts of training data.

1.2 Research identification

1.2.1 Research objectives

To eventually enhance the accuracy properties of the Virtual Support Vector Machines with the self-learning (VSVM-SL) method in settings with very few labeled samples, the goal is to deploy a constrained set of unlabeled samples for model learning and for very high spatial resolution multispectral remote sensing images. As a result, the training set which the model is learning will be enriched by informative unlabeled samples. Those are jointly evaluated and selected with respect to existing support vectors and virtual samples. In addition, the generation of spatial visualization for the uncertainty of results is done by checking the distance of SVMs hyperplane from the model and further monitoring how the uncertainty changes with the newly developed methods. The spatial visualization will be displayed as land cover classification maps

showing corresponding thematic uncertainty. Subsequently, these spatial visualizations will benefit the active learning process by providing human-machine interaction on relabeling uncertain samples in a prioritized way and will use those samples to relearn the model and eventually obtain higher accuracies.

RQ1: To what extent does the new Virtual Support Vector Machines with self-learning constraints on a semi-supervised scheme (VSVM-SL- Unlabeled Samples) method provide better classification accuracy with few labeled samples when compared to other/older methods such as SVM, VSVM, and VSVM-SL?

- Comparison analysis will be made between newly proposed semi-supervised methods to previous methods by comparing overall and average accuracies, kappa value, and F1 score.
- Line graphs will be used in order to see mean kappa values and overall accuracies of the methods.

RQ2: Does visualizing the uncertainties of the models improve human-monitored active learning approaches on relabeling uncertain samples?

- Model quantifies the certainty of unlabeled samples by checking the distance of SVM hyperplane and shows which land cover classes they belong to. Consequently, this helps the user to label those uncertain samples and bring them back to model.
- Therefore, a case study will be applied in order to assess the effects and overall performance of relabeling with visualization of uncertainty on active learning. Accuracy results of newly developed methods plus the uncertainty visualizations will be compared to the results of newly developed methods without the uncertainty visualizations.

1.3 Innovation aimed at

The innovation of the research aims at developing a semi-supervised classification method based on a self-learning strategy. This will provide results with higher accuracy on sparsely sampled remote sensing imageries and will be adaptable in the future to the classification of hyperspectral data. These innovations are aimed at bringing a new outlook with the extension and combination of methods on remote sensing and the cartography fields.

2 Theoretical Background & Literature Review

2.1 Theoretical Background

In this chapter, the theoretical background is discussed in six main subsections. In the first section, the theoretical background of one of the robust machine learning methods, Support Vector Machines (SVM) is introduced. The second part gives information about the extension of the virtual support vector machines with self-learning constraints for remote sensing image classification research and explains the semi-supervised learning and especially semi-supervised support vector machines. In the third part, the fundamentals of uncertainty, as well as uncertainty visualization, are explained. At last, base methods that are used for setting semi-supervised environments are explained, including actual SVM learning, Virtual SVM (VSVM), and Self-Learning (SL) strategy for pruning virtual and unlabeled samples with under specific threshold.

2.1.1 Support Vector Machines

SVM are supervised learning models in machine learning, that can solve classification and regression problems. As a working principle, SVM creates a linear decision surface with a suitable set of parameters that sticks to the generalization principle by mapping the input vectors to higher dimensional future space Z (Cortes and Vapnik, 1995). In this feature space, the optimal separating hyperplane that maximizes the margin is defined in the Eq. 1, where x is the nearest data point to the plane, w is the normal vector to the hyperplane, and b as an absolute value.

$$w_0 \cdot x + b_0 = 0 \tag{1}$$

Moreover, maximized distance $\rho\left(w_0, b_0\right)$ of optimal separating hyperplane for projection of training vectors can be formulated in the Eq. 2 and the maximized distance is displayed as optimal margin in Fig. 1.

$$\rho\left(w_0, b_0\right) = \frac{2}{\mid w_0 \mid} = \frac{2}{\sqrt{w_0 \cdot w_0}} \tag{2}$$

Which further creates an optimization problem in order to maximize the distance $\frac{1}{||w||}$ in subject to the constraints $\min_{n=1,2,\ldots,N} \mid w^t \cdot b \mid = 1$ and same time minimizing $\frac{1}{2} w^T w$ subject to the constraints $y_n(w_t \cdot x^n + b) \geq 1$. Therefore, quadratic programming (QR) is required to construct the optimal hyperplane because optimal hyperplane that minimizes $w_0 \cdot w_0$ is the unique under the constraints.

The formula above is a convex programming problem since in optimization step, minimizing $f(x)$ with constraints to $g(x)$ includes convex functions. A solution to this problem is to use the Lagrange function in Eq. 3:

$$L(w, b, \Lambda) = \frac{1}{2} w \cdot w - \sum_{i=1}^{l} \alpha_i \left[y_i \left(x_i \cdot w + b \right) - 1 \right] \tag{3}$$

The margin further can be described by determining support vectors (SVs) which $y_i (w \cdot x_i + b) = 1$ in order to define the model. (Cortes and Vapnik, 1995; Geiß et al., 2019). It can be seen in the figure 1 that squared samples on the border that are closest to the separating the surface are SV's.

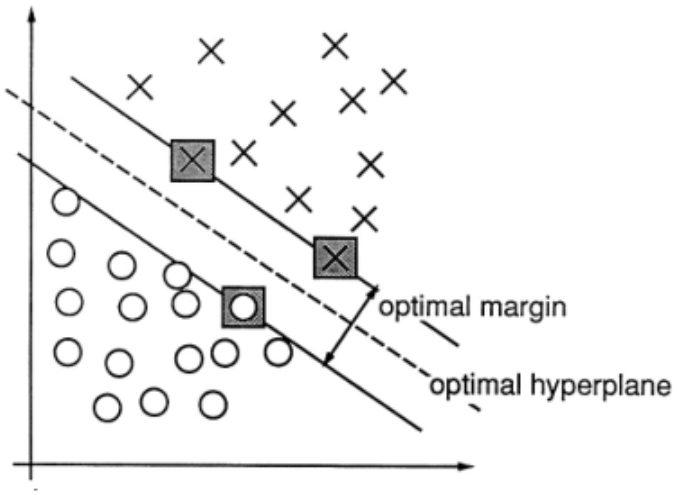


Figure 1: An example of linearly separable case. (Cortes and Vapnik, 1995)

In order to continue, Vapnik–Chervonenkis(VC) dimensions and structural risk minimization has to be further explained. In SVM it is important to control generalization ability since generalization indicates the trained model's performance on unseen data. Therefore, this has to be controlled by two factors so as error rate and capacity of the learning which are measured by its VC-dimensions (Cortes and Vapnik, 1995). Here, VC dimensions provide properties of a set of functions $f(\alpha)$ and can be defined for a two-class classification case. Moreover, if VC dimensions h is a set of functions, shattered points which are N data points that the learning model can produce 2^N distinct patterns on those data points will be one set of h points. Yet, it does not mean that every set of h points can be shattered (Burges, 1998) as it is shown in Fig 2.

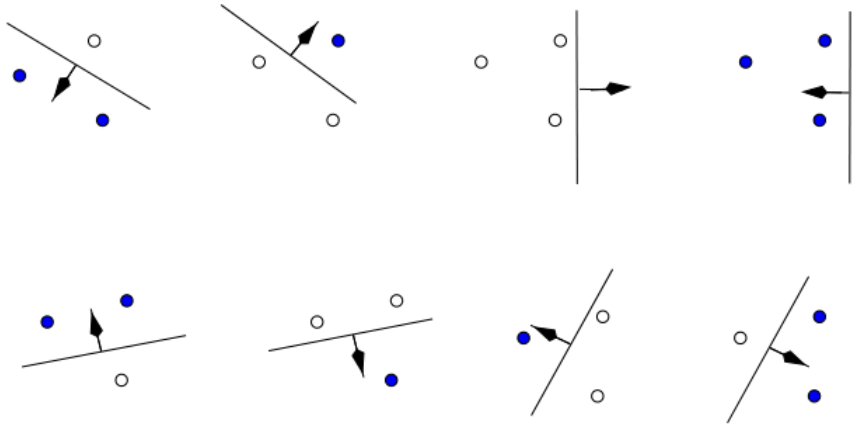


Figure 2: Shattered three points by oriented lines. (Burges, 1998)

To summarize, VC dimensions are related to parameters of the models, and the class of functions relies on the VC confidence. However, here, the goal is to minimize empirical risk and VC confidence by finding the subset of the chosen set of the functions yet it cannot be achieved easily since h is an integer value. That is why structural risk minimization (SRM) has to be defined. SRM works by dividing the class of functions into subsets and finds the actual subsets of functions that minimize the bound of risk (Burges, 1998).

As another point, it is vital to mention SVM strategy on dealing with non-linearly separable cases as it is formulated in the Eq 4 and 5. In order to deal with the non-separable case, constraints have to be modified by adding positive slack variables: ξ_i to the constraints. Therefore, in this case, the upper bound of α is introduced to the model so-called C. Which is different from the previously mentioned optimal hyperplane. Therefore, the input vectors are mapped through a non-linear transformation ϕ to a high dimensional space.

$$\min_{w,\xi_i,b} \left\{ \frac{1}{2} \mid\mid w \mid\mid^2 + C \sum_{i=1}^{n} \xi_i \right\} \tag{4}$$

subject to:

$$y_i \left(\langle \phi \left(x_i \right), w \rangle + b \right) \geqq 1 - \xi_i \tag{5}$$

However, applying polynomial combinations of features to transform data into higher dimensional space is causing high computational cost, and applying kernel method that provides solution by acting as a modified dot product. See the figure 3.

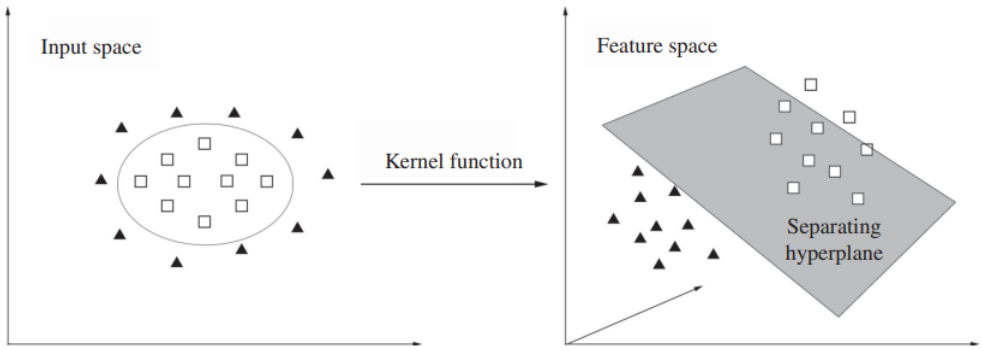


Figure 3: Transformation of input vectors to the higher dimensional space (Song et al., 2012)

To summarize, SVM are a highly robust method because of their general advantages in the field. These advantages are; its effectiveness in high dimensional spaces, effectiveness in cases where dimensions are higher than samples size so-called the Hughes phenomenon, and its overall strategy on using SV in the decision function which eventually makes it memory efficient in the sense of increasing computational speed. In addition to several researches that have been done on remote sensing field about SVM are including many wide ranges of areas on remote sensing application domain (biophysical tasks, land cover , and use tasks), change detection, and focusing on algorithmic advancements (Mountrakis et al., 2011).

2.1.2 Semi-Supervised Support Vector Machines

As an overview, semi-supervised learning is a combination of labeled data so-called "the training set : $L = \{(x_1, y_1) + (x_2, y_2) + ... + (x_l, y_l)\}$" and unlabeled data so-called "the working set : $U = \{x_1^{'}, x_2^{'}, ..., x_u^{'}\}$" and it only occurs when both training and working sets are valid. The reason that semi-supervised learning methods are trending is that in real-life data that have been generated are mostly unlabeled; for this reason, a lot of work has to be put on labeling those data and some of the occasions few labeled samples decrease the generalization ability and cause models not to adapt unseen data .(Bennett et al., 1999; Ding et al., 2017)

Moreover, this part of the chapter will mostly consider a semi-supervised learning method, called semi-supervised support vector machines (S^3VM) which is an extension of the standard form of the SVM method and useful when SVM are not beneficial when the above-mentioned situations on labeled data occur. (Ding et al., 2017)

As it has been mentioned, S^3VM follows the standard form of SVM and aims for finding the maximum margin to separate two classes by using training and working sets. Here, one of the aims is to obtain the lowest generalization error on unlabeled data on predicting outcomes with unseen data that comes within the new separating hyperplane. Another aim is to find the function that attains the minimum classification error and this can be done by applying two restrictions. The first restriction presumes that the working set belongs to class one and another restriction presumes that the working set belongs to class two. Furthermore, by calculating the error rate, these restrictions are satisfied. Additionally, for the nonlinear separable cases, so as SVM, usage of the kernel functions are suitable for S^3VM (Ding et al., 2017; Li and Zhou, 2014).

Decision function of S^3VM is formulated in Eq. 6 and 7. Here, C is a misclassification penalty and has to be greater than zero and it can be solved by using integer programming by optimizing when all variables are restricted to be an integer values.

$$\min_{w,b,\eta,,\xi,z} C\left[\sum_{i=1}^{l} \eta_i + \sum_{j=l+1}^{l+k} min\left(\xi_j, z_j\right)\right] + \| w \| \tag{6}$$

subject to:

$$\begin{aligned} y_i\left(w \cdot x_i + b\right) + \eta_i \geq 1 \qquad \eta_i \geq 0 \qquad i = 1, ..., l, \\ w \cdot x_i - b + \xi_i \geq 1 \qquad \xi_j \geq 0 \qquad j = l+1, ..., l+k \\ -\left(w \cdot x_i + b\right) + z_j + Md_j \geq 1 \qquad z_j \geq 0 \qquad d_j = \{0, 1\} \end{aligned} \tag{7}$$

To sum up, the semi-supervised approach on SVM brings promising results since it is more suitable on real-life data and it improves the performance of SVM model and with this extension, it extends to more application areas such as image processing. Yet, it still has some downsides because large sets of unlabeled data cause higher complexity in computation which eventually takes long time. However, it is possible to overcome through the extension of S^4VM method developed by Li and Zhou (2014); Ding et al. (2017)

2.1.3 Uncertainty

Here in this part, uncertainty will be defined and causes of uncertainty in remote sensing and image classification methods will be discussed. According to the Goodchild (2008) uncertainty is the contrast between contents of the geospatial database and corresponding phenomena in the real world and it can be also defined as a component of data quality in geospatial data depending on the purpose and the application (Lucieer, 2004).

Moreover, due to several factors, for example; atmospheric conditions, geometric calibration, sensor sensitivity, and sensor resolutions, different types of uncertainty in remote sensing exists and it became an important factor to investigate since remote sensing data is highly used in geographic information science (GIS) and many application fields of remote sensing. Regarding the image classification methods on remote sensing data, uncertainty is mostly caused by the pixel values, fuzzy boundaries, transition zones, and misclassified data points. For an instance, Fig. 4 demonstrates the situation where objects in the landscape are below the resolution of the image and this causes the mixed pixels situation. Moreover, fuzzy classification is also introduced for this problem. Instead of assigning the pixel with its label as Boolean values 0 or 1, values between 0 to 1 can be assigned to show its probable class as its is displayed in Fig 5 (Bastin et al., 2002)

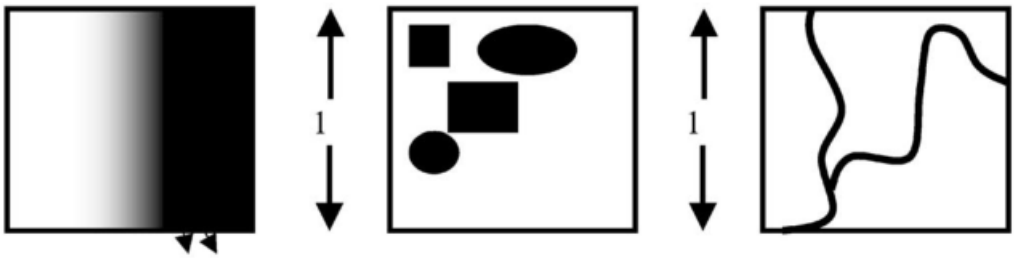


Figure 4: Example of mixed pixels causing uncertainty (Bastin et al., 2002)

.

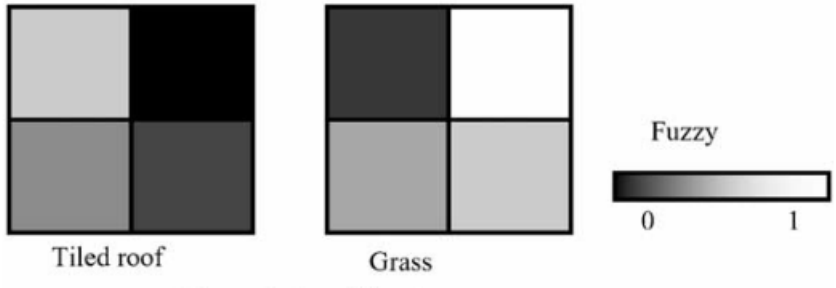


Figure 5: Example of fuzzy classification (Bastin et al., 2002).

Furthermore, to evaluate the image classification's overall quality, a confusion matrix is needed and it is determined through the reference labels. However, from the confusion matrix, it is not possible to see how the uncertainty is spatially distributed. Therefore, thematic and spatial uncertainty, where the error or lack of knowledge is coming from its geographic position or its thematic class, have an essential role in finding the quality of classification (Lucieer, 2004; Bastin et al., 2002).

In this part, the uncertainty of classification method, SVM will be taken care of. In the SVM models uncertainty is mainly occurs when data points are not labeled right and when they are closer to the margin. The condition is formulated below by (Wang and Pardalos, 2014) in Eq. 8 and 9. Here, uncertain data points are presented as x_i and it is free within the centered at $\overline{x_i}$ with Δx_i radius and could move towards in any direction within the uncertainty set.

$$\min_{w,b,\xi_i} \frac{1}{2} \mid\mid w \mid\mid_2^2 + C \sum_{i=1}^{m} \xi_i \tag{8}$$

subject to:

$$\begin{aligned} y_i \left(w^T \left(\overline{x_i} + \Delta x_i \right) + b \right) \geq 1 - \xi_i, \qquad & \xi_i \geq 0, \qquad i = 1, ..., m \\ \mid\mid \Delta x_i \mid\mid_2 \leq \delta_i, \qquad & i = 1, ..., m \end{aligned} \tag{9}$$

In addition, it is important to further analyze the uncertainty by quantifying, exploring, presenting, and communicating. This can be done by visualizing the patterns and spatial behaviors of uncertainty. Different visualization techniques can be used in remotely sensed image classifications to explore and present these patterns. For instance, uncertainty visualization in land cover maps and graphic variables such as color saturation, lightness, noise annotation lines can be used (Kinkeldey, 2014; Lucieer, 2004).

Furthermore, it is possible with the use of active learning techniques to improve model performances by ranking the uncertain pixels, exploring their uncertainty, and relabeling them as it has been done in the research by Tuia and Munoz-Mari (2012).

2.1.4 Support Vector Machines Settings

In the model learning process of the research, as an initial SVM method, the Gaussian radial basis function (RBF) kernel has been used due to its practicality in image classification and especially in its remarkable performance in remote sensing applications (Volpi et al., 2013). Here, this RBF kernel function can be formulated below in the Eq. 10, x and $x^{'}$ represents the feature vectors in input space, and its Euclidean distance stays under the range of zero and one.

$$K\left(x_i, x_j\right) = \exp(-\frac{\mid\mid x_i - x_j \mid\mid^2}{\sigma^2}) \tag{10}$$

Moreover, learning RBF kernels with concurrence of SVM requires a cost parameter C and it also requires an additional parameter γ to define kernel-width and these parameters range in $C = \{2^{-4}, 2^{-3}, ..., 2^{12}\}$ and $\gamma = \{2^{-5}, 2^{-4.5}, ..., 2^{3}\}$. As an other remark, this mentioned SVM approach has been used to classify the inputs for binary and multi class problems with one one-against-one SVM architecture.

2.1.5 Virtual Support Vector Machines

First of all, it should be mentioned that the Virtual Support Vector Machines (VSVM) method is a modification of the SVM approach. For this reason, the VSVM's working principle is built on SVM models which are explained in chapter 2.1.1. Where the overall aim is to find a separating hyperplane with the maximum margin between labeled samples of different classes and eventually determine those SV that are the labeled points on the border of the margins. Furthermore, in the VSVM approach, SV are captured from the initial SVM model. These captured SV control the encoding of invariances by altering modeled objects (e.g. invariances of scale or shape of the objects) which are after added to the input space as virtual samples.

Working principle of VSVM is explained in Fig. 6 and 7. Figure 6-a shows the input vectors in the feature space and Fig. 6-b shows the seperating hyperplane with a maximum margin when the initial SVM model is applied.

Thereafter, virtual samples were generated through encoding the invariances in the context of object-based image analysis framework (Blaschke, 2010). Invariances are encoded by the segmentation algorithm with the respect to the shape of the object and scale such as the size of an object. This segmentation algorithm initially uses the SV that is obtained in Fig 6-b and finds the segmentation level that involves these SV in the image domain. Thereby, with those segmentation levels object features are computed and introduced to the model in Fig 7-c as virtual samples. (Geiß et al., 2019)

These introduced virtual samples together with SV's are used for learning the model once more and which eventually finalize by altering the hyperplane with maximum margin (Fig. 7-d). Finally, virtual samples as induced by SVs are called Virtual Support Vectors (VSV). As a final remark, hyperparameters of the VSVM model, which are used to handle the case of non-separable data, are optimized by the hold-out method (Foody, 2009). This was done to see how well the model works with unseen data by using training data that the model is trained with to deal with the very large data set.

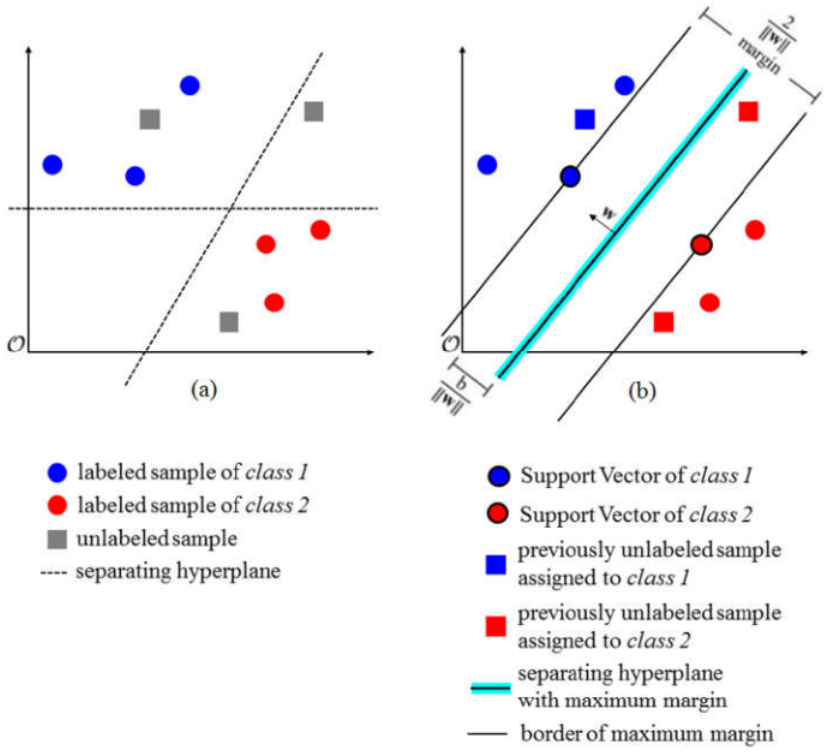


Figure 6: Working principle of Virtual Support Vector Machines (a),(b). (Geiß et al., 2019)

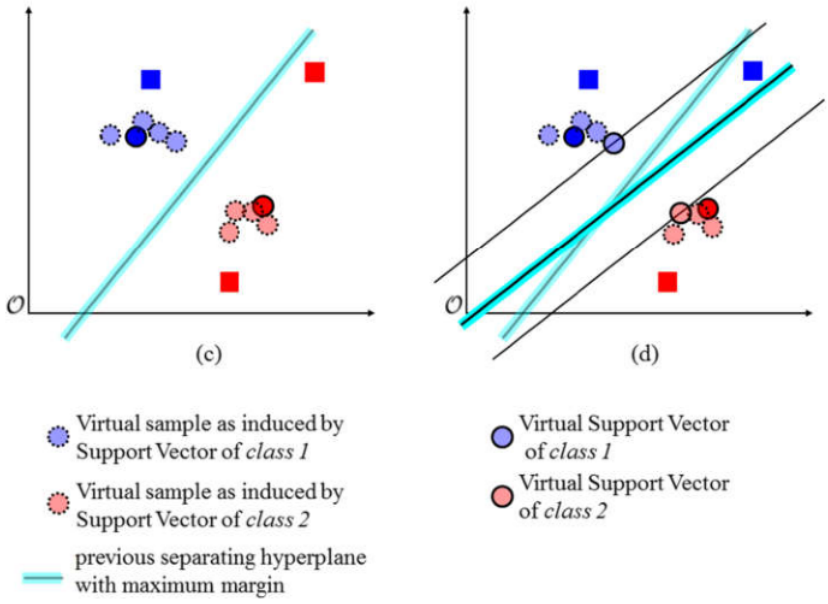


Figure 7: Working principle of Virtual Support Vector Machines (c), (d). (Geiß et al., 2019)

VSVM approach is described below in detail within the pseudocode under Algorithm 1 (Geiß et al., 2019). Here, the pseudocode explains the procedure that has been displayed in Fig 6 and 7.

Algorithm 1 (Virtual Support Vector Machines (VSVM))

Inputs:

Pool of labeled samples: X_{Train}, X_{Test}

Output:

SVM classifier retained with training set $\hat{X}_{Train}$

1: Learn initial SVM model with X_{Train}
2: Extract SVs and add them to a pool X_{Train}^{SV}
3: Perturb features based on X_{Train}^{SV} to generate a pool of virtual samples V^{SV}
4: Compile training set $\hat{X}_{Train} = X_{Train}^{SV} \cup V^{SV}$
5: Learn SVM model with $\hat{X}_{Train}$ and select model with optimal hyperparameters based on X_{Test}.

2.1.6 Self-learning strategy

In order to make VSVM approach more efficient, self-learning (SL) strategy is introduced. Overall goal of the SL strategy is to prune the uninformative virtual samples from the input space in under consideration of similarity and margin constraint (Fig. 8).

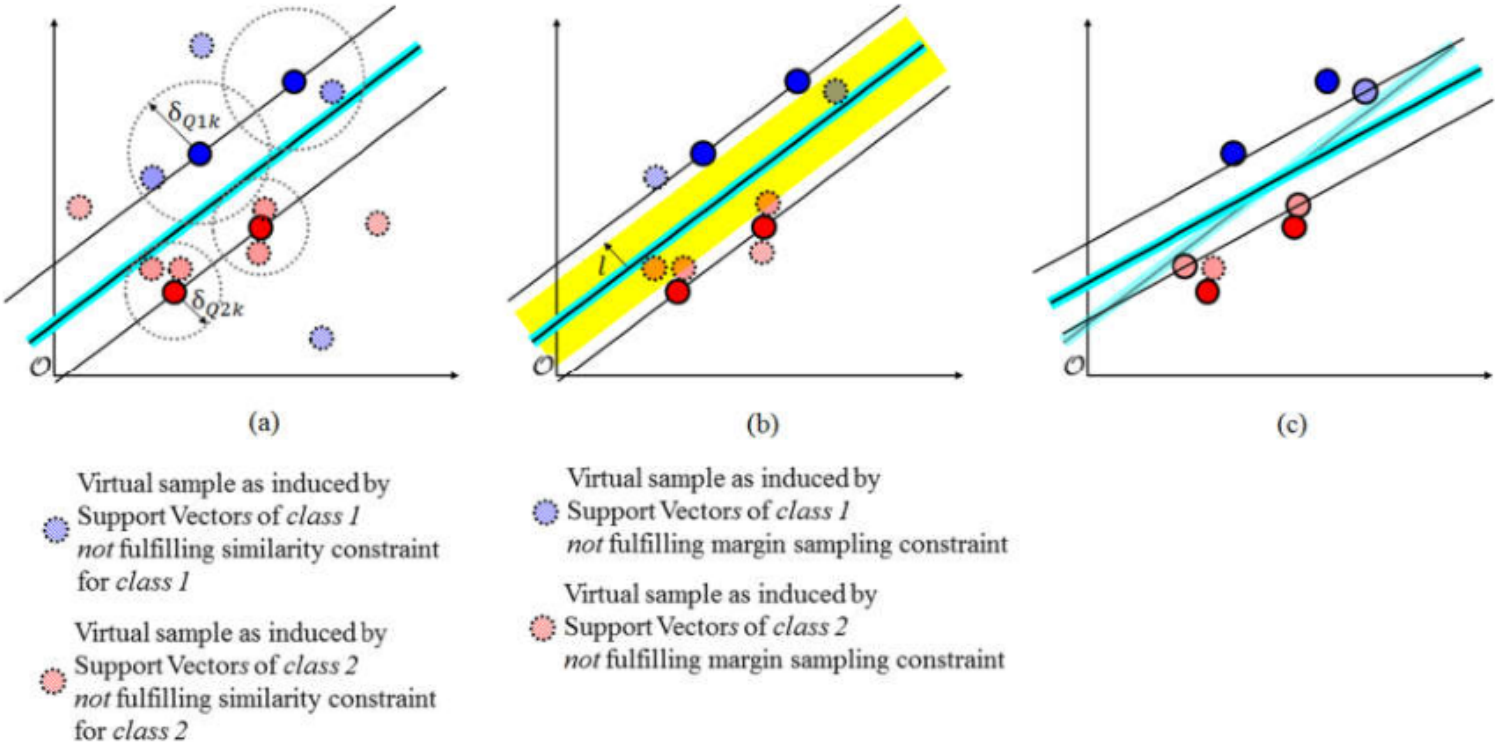


Figure 8: Working principle of Self-Learning strategy on VSV (Geiß et al., 2019)

In the similarity constraints approach, first of all, the Euclidean distance (d) between a virtual sample and its SV's is computed. Distance calculation is formulated below with the Eq. (11). In the equation, v stands for virtual samples and m for the features.

$$d_{ij} = \sqrt{\sum_{m} \left(v_{im}^{SV} - x_{jm}^{SV}\right)^2} \tag{11}$$

As a next step, a threshold δ is introduced to determine the uninformative samples. δ is calculated below in Eq. (12) , here, N_Q refer the the number of SV per thematic class Q.

$$\delta_Q = \frac{2}{N_Q\left(N_Q - 1\right)} \sum_{i=1}^{N_Q - 1} \sum_{j=i+1}^{N} \sqrt{\sum_{m} \left(x_{im}^{SV_Q} - x_{jm}^{SV_Q}\right)^2} \tag{12}$$

In the consideration of margin sampling constraint, it is aimed to prune those virtual samples that are far away from the hyperplane. This is an important step since those virtual samples that are not pruned with similarity constraint should be pruned for the reason that samples that are far from hyperplane are unlikely to be VSV. Therefore, the distance of a virtual sample to the hyperplane is calculated below in Equation 13.

$$f\left(v_{\delta_{Q_{kj}}}^{SV}\right) = \sum_{i=1}^{n} y_i \alpha_i K\left(x_i, v_{\delta_{Q_{kj}}}^{SV}\right) + b \tag{13}$$

Algorithm 2 (Self-learning strategy (SL))

Inputs:
Pool of SV: X^SV
Pool of virtual samples: $\hat{V}^{SV}$, X_{Test}
Output:
A pool of constrained virtual samples $\hat{V}^{SV}$

1: **for** i=1 to N in V^{SV} **do**
2: Compute Euclidean distance d_{ij} between v_i^{SV} and x_j^{SV};
3: **end for**
4: Compute class-specific maximum distance thresholds δ_{Qk} according to $\delta_{Qk} = \delta_Q \cdot k$ and Equation (12);
5: Remain virtual samples which satisfy $d_{ij} \leq \delta_{Qk}$ and prune the others from V^{SV} according to $V_{\delta_{Qk}}^{SV} = V^{SV} \cap \{V_i^{SV} | d_{ij} \leq \delta_{Qk}\}$ to establish a pool $V_{\delta_{Qk}}^{SV}$, which contains only virtual samples that lie within the radius of δ_{Qk} ;
6: **for** i=1 to N in V_{Qk}^{SV} **do**
7: Compute distance to hyperplane for class Q with decision function according to Equation (13);
8: **end for**
9: Remain virtual samples which satisfy the maximum acceptable distance l, and prune others from V_{Qk}^{SV} to establish a final pool of constrained virtual samples $\hat{V}^{SV}$

Figure 8 explains the use of the described constraints and shows how the hyperplane is adjusted after pruning the uninformative virtual samples. In Fig. 8-a similarity constraints approach is applied and after pruning the virtual samples, margin sampling constraint is applied in Fig. 8-b by calculating the distance of the remaining virtual samples to the hyperplane. In the last stage, the model is learned with remaining virtual samples and with SV, and the hyperplane is adjusted with the maximum margin Fig. 8-b.

Furthermore, self Learning strategy is described in detail within the pseudocode Algorithm 2 (Geiß et al., 2019). The algorithm briefly explains the framework in Fig 8 In the first stage, virtual samples among the SV are collected from the VSVM algorithm. After that, the euclidean distance is calculated between SV and the virtual samples. Virtual samples that are under a certain threshold are selected. The virtual samples that are far away are pruned from the model. In the second step, distance to the hyperplane is calculated between the remaining virtual samples, and samples that satisfied the maximum acceptable distance are selected. As a final product, a pool of constrained virtual samples are produced and these samples are used for model learning.

2.2 Literature Review

The literature review chapter is discussed in two main parts. The first part discusses the theory of support vector machines, semi-supervised SVM, active learning methods, and their applications mainly on the classification of remote sensing imagery. Furthermore, the first part involves some of the partial combinations of these methods. At last, the second part discusses uncertainty visualization and cartographic design.

2.2.1 Theory and the Applications of Machine Learning Methods

Cortes and Vapnik (1995) introduced a new machine learning method for classifying two-class problems, the so-called the support vector networks. This method is applied to very high-dimensional feature space and also enlarged to solve non-linearly separable training data. Empirical studies on the research made on optical character recognition data and further compared with various supervised algorithms such as linear classifier, k-nearest neighbors algorithm.

Burges (1998) further demonstrates and introduces a tutorial on support vector machines(SVM) and mainly focuses on its concepts on structural risk minimization and Vapnik–Chervonenkis dimension (VC dimension). The research explains the kernel mapping technique to classify non-linear data in Fig. 9 below. Figure shows how SVM transform the data from the original space into a higher dimensional feature space and make it linearly separable and show how well SVM show good generalization performance on very large VC dimensions.

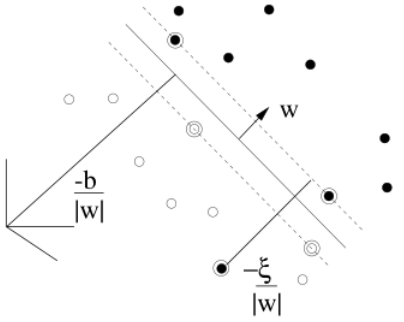


Figure 9: Linear separating hyperplanes for the non-separable case (Burges, 1998)

Bennett et al. (1999) proposed a new method called semi-supervised support vector machines (S^3VMs). The purpose of the method is to overcome the transduction problem by applying overall risk minimization (ORM) with the use of both labeled (training set) and unlabeled samples (working set). Since transduction does not produce a predictive model and when an unknown point is added to the set, it causes the transductive algorithm to be repeated to predict the label. Subsequently, this research (S^3VMs) tested on ten real-world application data sets and results showed that adding unlabeled data improves the generalization.

Semi-supervised support vector machines (S^3VM) are a highly favorable approach on few labeled data but when the obscure and misleading unlabeled samples are selected they can perform poorly. Therefore, Li and Zhou (2014) come up with a solution to improve S^3VMs safeness. Subsequently, two approaches that are developed named S^3VMs-us and S^4VMs (Safe S^3VMs) that work with multiple low-density separators eventually maximize the performance against inductive support vector machines. After empirical studies, it has been detected that new methods can decrease the risk of poor separation and moderately improves performance. Moreover, research from Ding et al. (2017) gives an overview to the S^3VM and they further investigate the previous methodology on S^3VM, discuss the extension of methods such as transductive support vector machine, Laplacian support vector machine, mean S^3VM, S^3VM based on cluster kernel to overcome main challenges of semi-supervised methods especially on computation costs on training the models. Research also indicates that extension of the methods truly improves the computation although there are still some obstacles to overcome in the future.

Fernández-Delgado et al. (2014) did detailed investigation on 179 different classifiers from 17 families including support vector machines, decision trees, rule-based classifiers, and so on. According to the experimental results on 121 real-world application data sets random forest (RF) and support vector machines (SVM) resulted in the best accuracy. Therefore, this paper gives a comprehensive explanation of these methods and their applications.

In the application are of machine learning, Melgani and Bruzzone (2004) approached the classification of hyperspectral remote sensing images problem by using support vector machines(SVM) and compares the results with non-parametric classifiers such as radial basis function neural networks and the K-nearest neighbor classifier. Moreover, research further analysis the issue of applying SVM binary separator to multi-class problems and discusses the one-against-all, the one-against-one, and two hierarchical tree-based strategies on a real Airborne Visible/Infrared Imaging Spectroradiometer hyperspectral data set. Results showed that SVM are effective compared to other non-parametric classifiers in hyper-dimensional spaces and can solve multi-class problems.

Foody and Mathur (2004) evaluate the performance of support vector machines (SVM) on multi-class image classification and compare to other classification methods such as discriminant analysis (DA), decision tree (DT), and feed-forward neural network (NN). According to the research, the most precise classification results were obtained with the SVM method. Additionally, the overall performance of SVM is also evaluated by taking the training samples size from 15 to 100 cases per class. Although SVM performs better on a small number of training-set, increasing training set size obtained decent accuracy.

Mountrakis et al. (2011) investigated the use of support vector machines in remote sensing over the year and further classifies the research that has been done on this topic. They conducted this SVM overview in remote sensing based on its application areas, platform type, spatial and spectral resolution, and last change detection. They further highlighted its upsides and downsides within previously published works and they provide a guideline for future improvements.

Izquierdo-Verdiguier et al. (2012) deals with remote sensing image classification using an invariance in Support Vector Machines (SVM) which refers to the robustness of the classifier to any changes in data and deals with characterizes of images. The method uses the initial SV that maximizes the margin and further includes invariances to reflections and rotations and then object scales and finally generates synthetic SV so-called Virtual Support Vectors. Obtained results demonstrate that method works efficiently with few labeled samples and appears to be a robust classifier.

Ul Haq et al. (2011) present a ℓ^1 minimization-based sparse representation approach for hyperspectral data classification with a few labeled samples. Unlike the supervised learning, method approaches the proposed model does not acquire training and testing phases use labeled samples to determine the representation of test samples. Additionally, the model overcomes the challenge of classifying a few sampled but high dimensional data as known as the Hughes phenomenon (Hughes, 1968). At last, experimental results showed promising results on four hyperspectral data sets compared to the traditional classification methods.

Geiß et al. (2017) investigated the effect of spatially non-disjoint training and test samples on model generalization in supervised classification with spatial features. Additionally, two different partitioning strategies for training and test sets followed to effects of spatial auto-correlation on spatial features. Thereby, the first strategy determined spatially random selected samples meanwhile, the second strategy determined spatially disjoint selection with topological constraints on multi- and hyperspectral acquisitions over urban areas.

Bruzzone and Carlin (2006) proposed a multilevel context-based system for the classification of very high spatial resolution images. To do so, they developed a feature extraction module which is depicted in Fig. 10 below. Figure investigates the image in multilevel and extracts the spatial context of every pixel and segments the image with a support vector machine (SVM) classifier with a tree-based hierarchical constraint. This allowed the feature-extraction module to work with different image scales. Moreover, experimental results on two different data sets, urban and rural areas of Pavia(Italy) and Trento(Italy) showed that overall accuracy does not change on a different number of levels, and the model can characterize the spatial context adaptively.

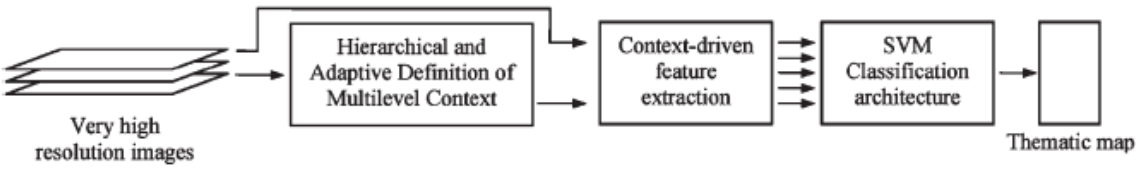


Figure 10: Scheme of the proposed approach of multi-level context-based classification strategy (Bruzzone and Carlin, 2006)

Bruzzone et al. (2006) introduced a method for the classification of remote sensing images. The method is based on semi supervised system that uses labeled and unlabeled samples in the training phase of SVM. Besides, the SVM algorithm is enriched by the transductive process which finds the best hyperplane iteratively. The proposed method showed that the transductive SVM approach can achieve stable and high classification accuracy results compared to standard inductive SVM.

Dópido et al. (2013) tailored a semi-supervised learning technique on the classification of hyperspectral remote sensing imagery. The reason is that hyperspectral imagery provides higher dimensional data but with a few labeled samples since it is costly and time-consuming which eventually ends up in Hughes phenomenon (Hughes, 1968). Therefore, with use of the unlabeled samples in a self-learning framework that selects the most informative unlabeled samples, has achieved favorable results in multinomial logistic regression (MLR) and a support vector machine (SVM) classifiers.

Lu et al. (2016) developed a new framework on semi-supervised learning for hyperspectral and panchromatic remote sensing image classification which eventually obtains improved overall classification accuracy with few labeled samples compared to other supervised algorithms. The framework that is displayed in Fig. 11 also uses an active learning approach for a self-learning strategy on image segmentation and automatically selects unlabeled informative samples by avoiding extra human-machine interaction. Therefore, in each iteration of the self-learning step, the candidate set chooses the samples with the same predicted labels with object labels until the stop criteria are met.

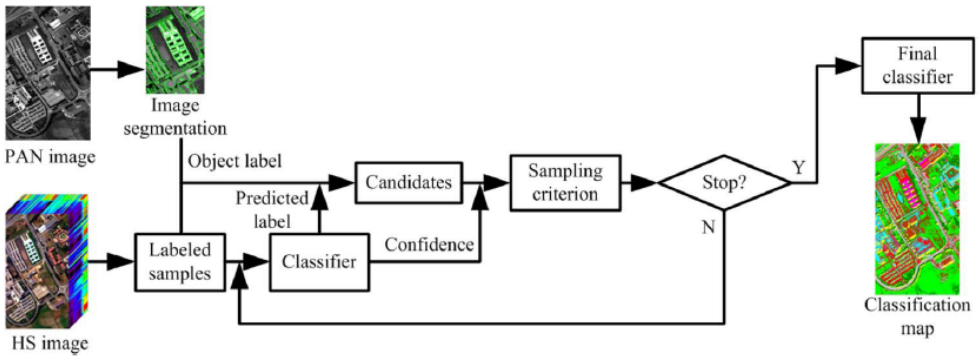


Figure 11: Main procedures of the proposed synergetic classification approach (Lu et al., 2016)

Tuia et al. (2009) evaluated two active learning strategies on both hyperspectral and very high-resolution remote sensing images with support vector machines classifier. Since SVM perform better with the high-quality labeled data, it turns into a difficult task to determine this set and active learning methods come into action to find the accurate set with human-image interaction. Therefore, results showed that applying active learning strategies gives consistent accuracy and methods can reach the same quality as large data sets.

Tuia et al. (2011) investigated active learning algorithms such as committee, large margin, and posterior probability on remote sensing image classification in order to create an optimal training set. They evaluate the working principle of active learning, adding new labels and selecting the ones that are beneficial for the model in an iterative way. Therefore, the evaluation process continued with challenging remote sensing scenarios and tested on both very high spatial resolution and hyperspectral images. Experimental showed that active learning algorithms have a great potential in the remote sensing area and they are robust on noise.

Demir et al. (2010) explored batch-mode-active-learning methods for the classification of remote sensing images. Additionally, uncertainty and diversity are selected as two parameters to investigate techniques. In this sense, a newly proposed novel query function that is based on a kernel clustering compared with state-of-the-art methods for very high-resolution multispectral and hyperspectral images. Results showed better accuracy on the newly proposed method for both data sets.

Pan et al. (2018) introduced a collaborative method on combining semi-supervised learning and active learning to classify hyperspectral imagery with limited labeled samples. Therefore, they follow a strategy so-called collaboratively integrated using clustering (CLUC) and this strategy clusters unlabeled samples, additionally, by calculating the uncertainty, most uncertain unlabeled samples are labeled manually. Experimental results showed that pseudo labeling strategy showed an important role in proposed algorithms and the used methods are open for future alternative extensions.

2.2.2 Uncertainty Visualization and Active Learning Methods

Kinkeldey (2014) explored major challenges in the analysis of land cover classification that are mainly caused by inaccuracies of multiple data sets such as bias in sensors or spatial disintegration. Therefore, Kinkeldey emerges an approach to quantifying and visualizing uncertainty in order to overcome this challenge. By doing so two elements are discussed in change detection and analysis steps, and visual communication of uncertainty within a case study. Example in Fig 12. shows uncertainty change per pixel with the real image in the bottom part and it is demonstrated as black (0.0) as no uncertainty to white (1.0) to maximum uncertainty. Uncertainty measure for land cover change is determined by the fuzzy membership values μ. For each scene, μ values are conducted by the minimum operator.

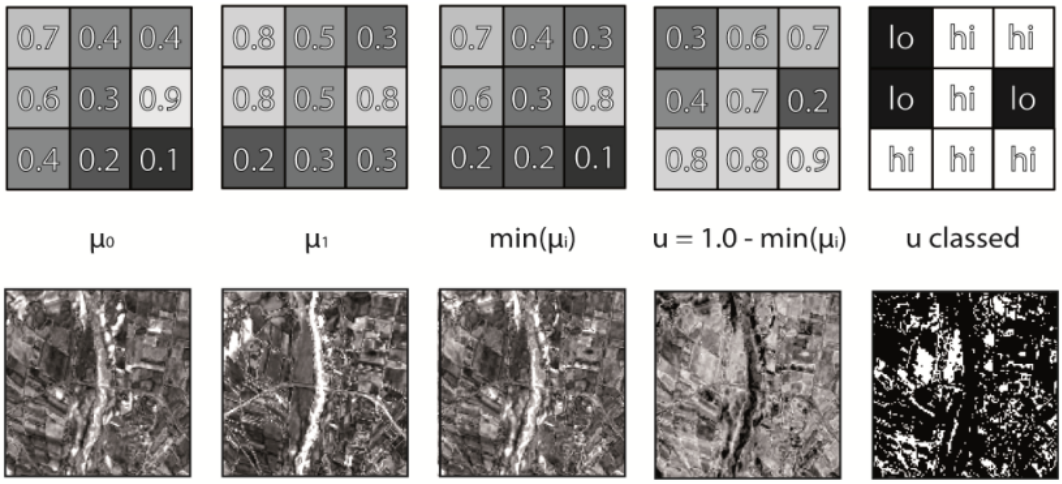


Figure 12: Example showing the change in uncertainty (Kinkeldey, 2014)

Tuia and Munoz-Mari (2012) discusses active learning (AL) scenarios in remote sensing based on the uncertainty of the pixels. AL methods are taken into consideration in terms of ranking the pixels by their uncertainty and users confidence in labeling depending on the user's experience and knowledge, example is shown in Fig 13. Moreover, experimental data sets consisted of different levels of resolution in order to test user's confidence in labeling after determining the areas by uncertainty and results showed that the area of AL has great potential and opens up for many future pieces of research.

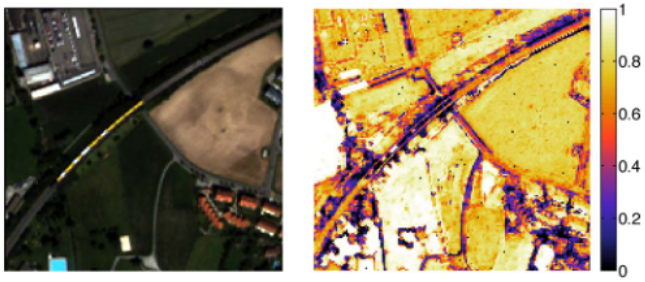


Figure 13: Images into confidence map after 5th iteration (Tuia and Munoz-Mari, 2012)

Murphy (2015) introduced a visualization option in the cartographic medium by combining remote sensing imagery with cartographic symbolization. Therefore, a suitable approach followed the idea of applying visual hierarchy on individual image objects and highlighting specific objects or classes. Research shows that this approach enhanced the user's visual attention and also the user's visual communication with image maps.

Since uncertainty is a major factor in processing geospatial data and ignoring the uncertainty may result in unusable results. For this reason, Kinkeldey et al. (2014) proposed a technique for visualizing uncertainty with the use of noise annotation lines. Therefore, two experiments are made to assess the method. It showed that noise annotation lines can be used in 6 uncertainty levels and can demonstrate a qualitative comparison of constant uncertainty.

Lucieer (2004)'s paper investigates uncertainties in remote sensing image classification and focuses on its visualizations. Research further separated into three parts. The first part is about developing and implementing a visual exploration tool; the second part is about developing, implementing, and applying image segmentation techniques for identifying objects and quantifying their uncertainty. The third part is about developing and implementing visualization techniques to explore the relationship between uncertainty. More specifically, in the first part, a fuzzy classification algorithm has been explored in remotely sensed imagery with its related uncertainty. This algorithm was applied to the Landsat 7 ETM+ image from Southern France and achieved an overall of 88 percent accuracy. In the second part, the working area is selected from the IKONOS image. Data belong to an agricultural area in the Netherlands and the segmentation technique, the so-called split-and-merge algorithm is applied. The method was extended with two case studies. As an example in Fig. 14 texture-based segmentation of a figure is displayed with its related uncertainty. In the left side of the figure 14, the object is classified with five reference classes and the related uncertainty of the classes is displayed on the right side. At last, the third part of the research has shown the effect of uncertainty by adjusting the threshold and generally, research showed that interactive visualization tool has an important role in objects uncertainty in remote sensing.

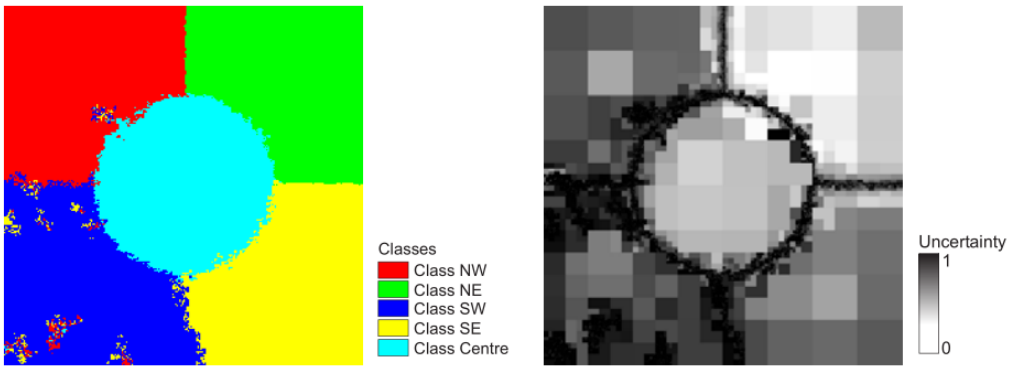


Figure 14: Supervised texture-based segmentation and related uncertainty for all object building blocks (Lucieer, 2004)

3 Methodology

In this section, the methodology is introduced with four new algorithms. The first three algorithms are created semi-supervised learning. These methods are based on the theory that is explained in the theoretical background section. At last, the fourth method demonstrates the uncertainty approach that boosts the models by visualizing and relabeling the uncertain samples.

3.1 Semi-Supervised Methods

This section explains three newly proposed semi-supervised methods on the SVM approach.The first method follows the initial idea of Virtual Support Vector Machines with Self Learning (VSVM-SL) strategy and combines unlabeled samples into model learning. The second method initializes a Virtual Unlabeled Samples approach by following the basic principles of the VSVM method. Finally, the third approach sets a benchmark method on a semi-supervised scheme by integrating unlabeled samples into the training data pool, and it learns the model with initial SVM method. Furthermore, in all these newly proposed approaches, the SL strategy is used in order to prune uninformative unlabeled and labeled samples. This prevented the use of further heuristics regarding which semi-labeled samples can be used.

3.1.1 VSVM-SL with Unlabeled Samples

As mentioned in the above section, VSVM-SL with the Unlabeled Samples method follows the idea of the VSVM-SL strategy and the overall aim is to eventually further increase the accuracy properties of the VSVM-SL in settings with few labeled samples and deploy a constrained set of unlabeled samples for training the model. By doing so the model is enriched by informative unlabeled samples and these informative unlabeled samples will be considered and selected with respect to the existing SVs and virtual samples.

In this model extension, Algorithm 3 starts with collecting unlabeled samples $x_1^*, x_2^*, ..., x_k^*$ among with the labeled training data $x_1, x_2, ..., x_n$ that will be used for initial model learning. As a second step, SV's are extracted from the initial SVM model and the VSVM approach is applied to collecting virtual samples by perturbing the features of these SVs. Moreover, at this stage, virtual samples and the selected unlabeled samples were forgathered in a pool for pruning uninformative VSV's and unlabeled samples. For that reason, by checking the similarity constraints (Eq. 12) and margin sampling constraints (Eq. 13), the model is left with the virtual samples that induced with unlabeled samples and new separating hyperplane with maximum margin has been found by the effect of the enriched unlabeled samples. The procedure is further described in the following figure 15. Here, in the figure 15, the blue color shows the gathered data, before or after any process. The red color shows the steps that use unlabeled samples. The circular cross represents the step when the training data is enriched by semi-labeled samples. At last, the green color stands for the model training steps.

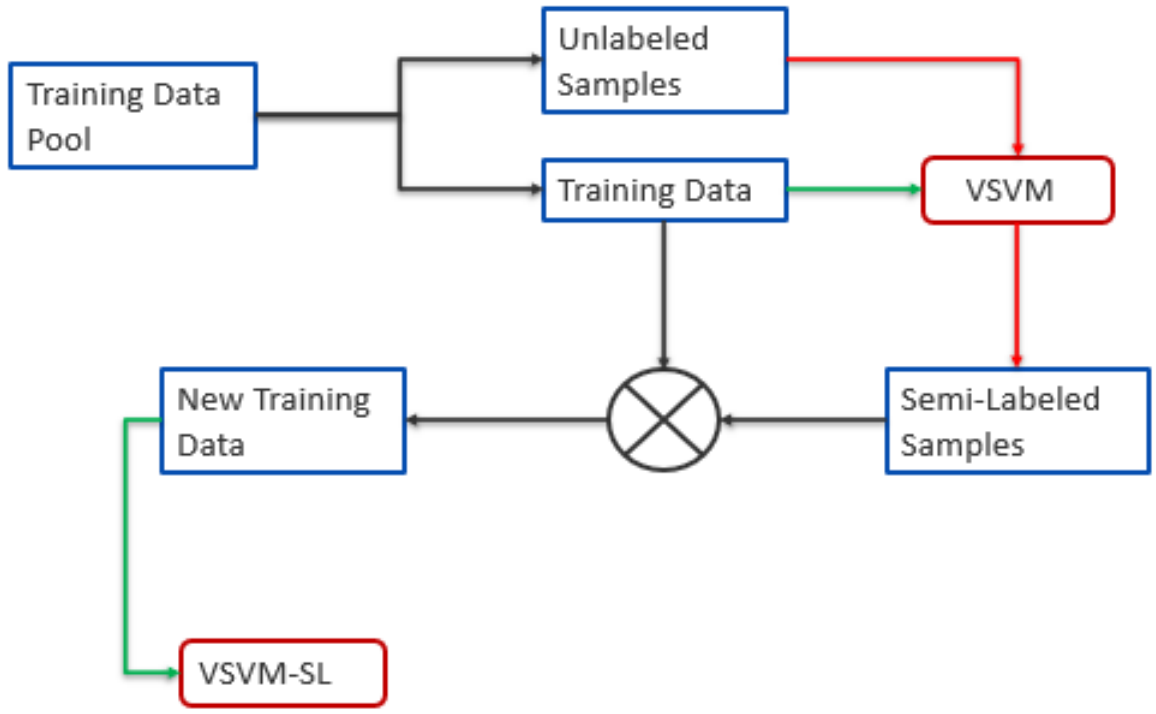


Figure 15: Working principle of VSVM-SL with Unlabeled Samples

VSVM-SL with Unlabeled Samples method is described below in detail within the pseudocode under Algorithm 3.

Algorithm 3 (VSVM + Unlabeled Samples+ SL Strategy)

Inputs:

Pool of labeled samples: X_{Train}, X_{Test}

Pool of unlabeled samples: $X_{TrainRemaining}$

Output:

SVM classifier retained with training set $\hat{X}_{Train}$

1: Select unlabeled samples from remaining training samples $X_{TrainRemaining}$
2: Learn initial SVM model with X_{Train}
3: Extract SVs and add them to a pool X_{Train}^{SV}
4: Perturb features based on X_{Train}^{SV} get virtual samples V^{SV}
5: Get SVs of unlabeled samples and perturb features of unlabeled samples based on $\hat{X}_{Train} = X_{Train}^{SV} \cup V^{SV}$
6: Apply self-learning strategy to V^{SV} and unlabeled samples which satisfy the maximum acceptable distance and prune others according to Algorithm (2).
7: Establish pool of constraint samples $\hat{V}^{SV}$
8: Compile training set $\hat{X}_{Train} = X_{Train}^{SV} \cup \hat{V}^{SV}$
9: Learn SVM model with $\hat{X}_{Train}$ select model with optimal hyperparameters based on X_{Test}.

3.1.2 VSVM-SL with Virtual Unlabeled Samples

The second semi-supervised approach, VSVM-SL with Unlabeled and Virtual Unlabeled samples, is motivated by VSVM method (chapter 2.1.5). It follows the idea that adding virtual unlabeled samples by integrating the invariances of shape and scale to already pruned unlabeled samples. With this approach, it is aimed to possibly increase accuracy properties of the model especially with the few labeled samples by benefiting the robust and sparse model solutions of virtual samples since these samples enhance generalization capabilities of the model. Therefore, method basically accompanies the approach on Fig. 15.

VSVM-SL with Virtual Unlabeled Samples method is described below in detail within the pseudocode under Algorithm 4.

Algorithm 4 (VSVM + Unlabeled Samples+ SL Strategy + Virtual Unlabeled Samples)

Inputs:
Pool of labeled samples: X_{Train}, X_{Test}
Pool of unlabeled samples:$X_{TrainRemaining}$
Output:
SVM classifier retained with training set $\hat{X}_{Train}$

1: Select unlabeled samples from remaining training samples $X_{TrainRemaining}$
2: Learn initial SVM model with X_{Train}
3: Extract SVs and add them to a pool X_{Train}^{SV}
4: Perturb features based on X_{Train}^{SV} get virtual samples V^{SV}
5: Get SVs of unlabeled samples and perturb features of unlabeled samples based on $\hat{X}_{Train} = X_{Train}^{SV} \cup V^{SV}$
6: Apply self-learning strategy to V^{SV} and unlabeled samples which satisfy the maximum acceptable distance and establish $\hat{V}^{SV}$.
7: Learn SVM model with $\hat{X}_{Train} = X_{Train}^{SV} \cup \hat{V}^{SV}$
8: Extract SVs from the model.
9: Perturb features of unlabeled samples based on $\hat{X}_{Train}^{SV}$ to get virtual unlabeled samples.
10: Prune virtual unlabeled samples with self learning strategy and establish constraint samples $\tilde{V}^{SV}$
11: Learn model again with $\tilde{X}_{Train}$ and select model with optimal hyperparameters based on X_{Test}

Here within the pseudocode, Algorithm 4 is displayed. In the algorithm, unlabeled samples $x_1^*, x_2^*, ..., x_k^*$ and labeled samples $x_1, x_2, ..., x_n$ are collected from the pool training samples. Thereafter, with the labeled samples the initial SVM model is learned to obtain SV for applying the VSVM approach. With the SL strategy, these unlabeled samples and virtual samples induced by unlabeled samples are pruned and the model has learned again as it is shown in Fig. 15. As a next step, the unlabeled samples that have become SV are extracted to obtain the virtual unlabeled samples. At this point, features of the extracted SV perturbed with the VSVM approach (chapter 3.1.2) and unlabeled virtual samples are established. As a final remark, the SL strategy (chapter 2.1.6) is applied to prune uninformative virtual unlabeled samples, and the model is learned again. As an output SVM classifier retained with a training set is generated.

3.1.3 Benchmark Method for Semi-Supervised Learning

This method is created as a modified and improved version of the basic function of SVM by basically including informative unlabeled samples. This method aims to increase the generalization abilities of decision functions when training information is not available. By doing so, the initial SVM is altered into a semi-supervised learning scheme and becomes a benchmark method for further semi-supervised learning methods. Furthermore, it follows simpler approach compared to the algorithms 3 and 4.

SVM with Unlabeled Samples approach is described below in detail within the pseudocode under Algorithm 5.

Algorithm 5 (SVM + Unlabeled Samples+ SL Strategy)

Inputs:
Pool of labeled samples: X_{Train}, X_{Test}
Pool of unlabeled samples:$X_{TrainRemaining}$
Output:
SVM classifier pruned with unlabeled samples

1: Select unlabeled samples from remaining training samples $X_{TrainRemaining}$
2: Learn initial SVM model with X_{Train} and extract SVs.
3: Prune the unlabeled samples with self learning strategy and establish constraint samples $\hat{X}^{SV}$
4: Learn the model again with $\hat{X}_{Train}$ and select model with optimal hyperparameters based on X_{Test}

In the Algorithm 5 , first of all, initial SVM is learned with training samples $x_1, x_2, ..., x_n$ and SV are extracted. Thereafter, with the selected unlabeled samples so as $x_1^*, x_2^*, ..., x_k^*$, SV are pruned with the indication of unlabeled samples with the constraints of chapter 2.1.6. As a final remark, the model has learned again with the training set that is enriched with pruned unlabeled samples. After, the best optimal hyperparameters are determined with the enhanced generalization ability by benefiting the semi-supervised learning approach.

3.2 Uncertainty Visualization

In this chapter, as an active learning method in the SVM approach, uncertainty visualization is introduced. As a primary idea uncertainty of the models is calculated and visualized in order to aim for higher accuracy of the developed methods. Therefore, this active learning approach initialized for the usage for all developed semi-supervised methods.

As a working principle, this approach follows the theoretical background of uncertainty. See in chapter 2.1.3. Therefore, the method measures the uncertainty of the SVM models. To do so, the optimal hyperplane that maximizes the margin has to be initially determined. By taking the margin constraints all the classified samples of certain classes are taken into account as input vectors and the distances of these samples to the hyperplane are aimed to be calculated with the margin constraints. Fig 16 shows

the calculated distances of samples to the hyperplane with l. SV is also displayed as the samples within the hyperplane in the figure.

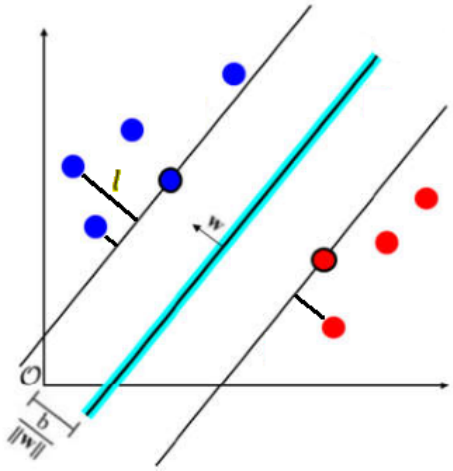


Figure 16: Illustration of distance to the hyperplane

Thereafter, these calculated distances are normalized in the range of zero to one to scale down the distance differences. Therefore, to determine the most uncertain samples, samples among the calculated hyperplane distances are visualized by stretching the uncertainty values along with the histogram. To improve the visual contrast, the tone of a single color is selected. With the help of visualizations and normalized distances, it can be considered that these samples closer to the hyperplane are most likely to be uncertain samples since SV are at the zero distance on the hyperplane. Moreover, in the relabeling procedure of these uncertain samples, an absolute number of most uncertain samples are selected, relabelled according to the reference labels, and the model has learned again by setting a new hyperplane with maximum margin. From this stage, calculating the uncertainty values once more and visualizing them makes it possible to monitor how the uncertainty of the models changes with this interactive relabelling process. Also, uncertainties of the introduced semi-supervised methods can be compared as a second parameter to the accuracy values. As an example, scheme of semi-supervised learning method, VSVM-SL with Unlabeled Samples is demonstrated with the integration of uncertainty visualization approach and it is shown in Fig. 17. In the figure, the blue color is used for showing data pools and the red color is used for steps that use unlabeled samples. Thereafter, training data that is enriched by semi-labeled samples are represented by the circular cross and the green color used for the model training steps. At last, the active learning process is showed with light green color. Furthermore, this active learning method eventually sets a collaborative learning method by combining this approach with semi-supervised learning methods.

The uncertainty visualization approach is demonstrated in detail within the pseudocode under Algorithm 6. Here, in the algorithm, as a first step, a semi-supervised SVM method is selected. Model is trained with its related methodology and labels are predicted and the initial accuracy values. Thereafter, distances of samples to the hyperplane are calculated and normalized. Uncertainty visualizations are created with the normalized values and most uncertain samples are determined. After the relabeling process, as

a last step accuracy and uncertainty are calculated and compared with the previous values.

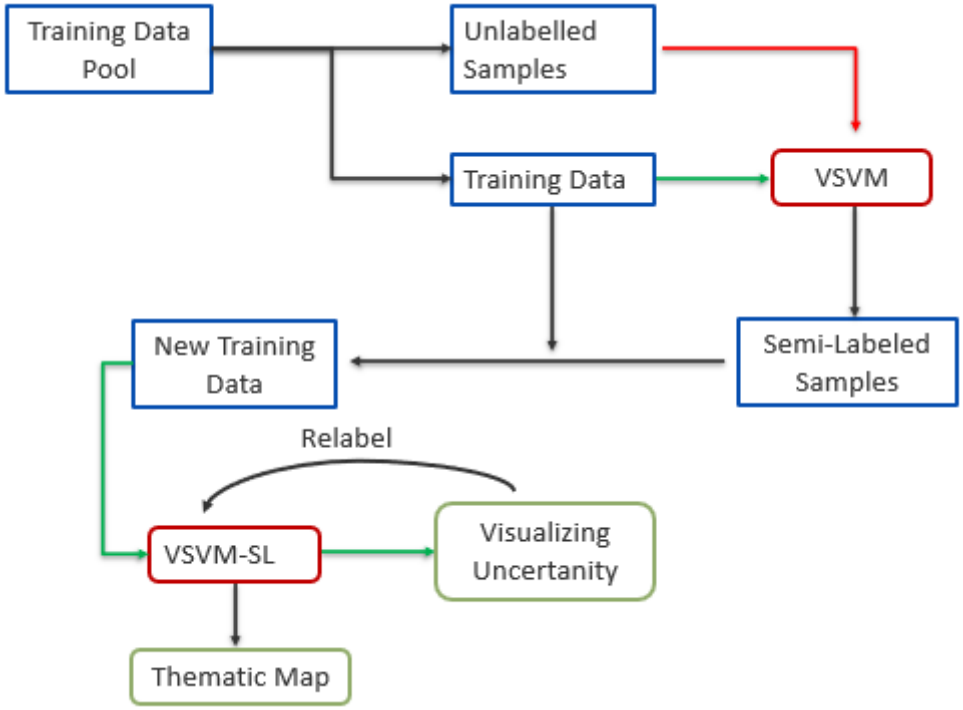


Figure 17: Working principle of Active Learning Approach with Uncertainty on VSVM-SL-U Method

Algorithm 6 (Active Learning Approach with Uncertainty)

Inputs:
Pool of labeled samples: $\hat{X}_{Train}$
Output:
Pool of labeled and relabeled samples : $\hat{X}_{Train}$, $\hat{X}'_{Train}$

1: Train the model with the selected SVM approach
2: Predict the labels the data set
3: Calculate the distance to hyperplane for each class Q
4: Normalize the distance in range $\{0,1\}$
5: Visualize the distances to determine most uncertain samples.
6: **for** i = 1 to N in $\hat{X}_{Train}$ **do**
7: Select most uncertain top 100 labeled samples;
8: Relabel them according to reference label;
9: **end for**
10: Recalculate accuracy and uncertainty

4 Results & Discussion

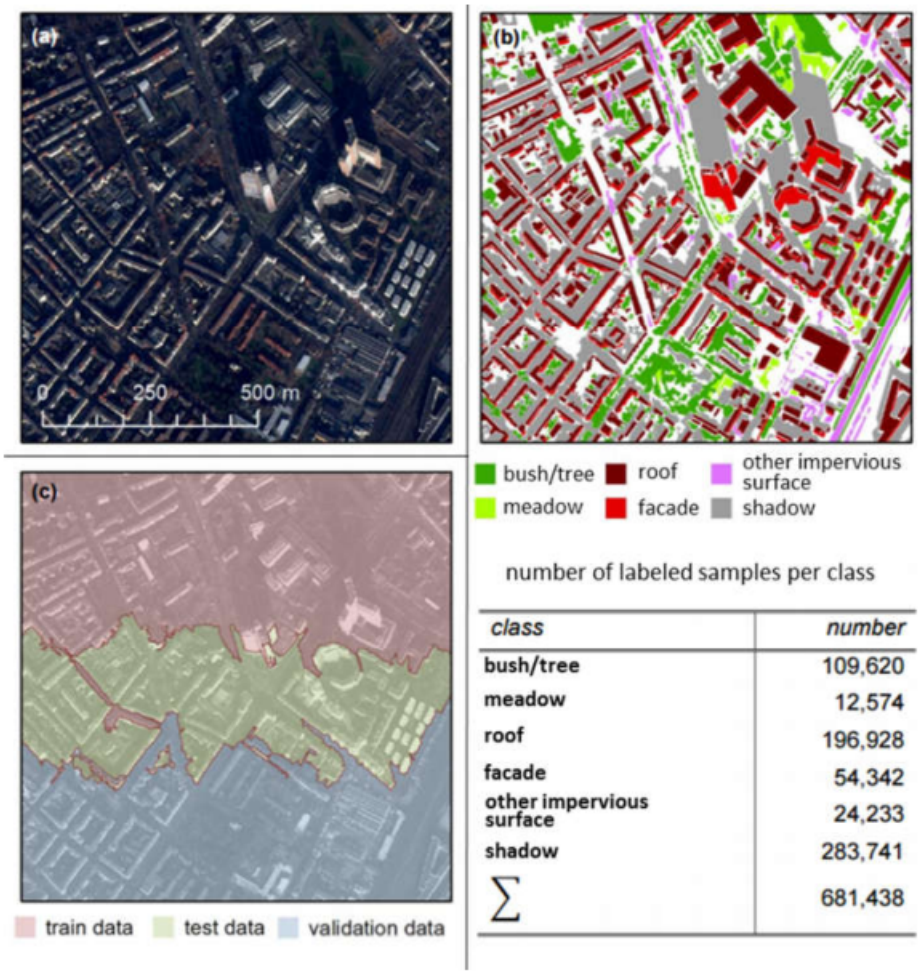
In the evaluation of newly proposed semi-supervised algorithms, VSVM-SL-U, VSVM-SL-VU, and SVM-SL-U, two different data sets are used with their configurations. So that these methods are evaluated with invariances to scale and shape for both of the data sets. Additionally, binary and multi-class classification settings are further added to evaluation processes. Furthermore, data is explained with its following thematic classes for each data sets in the next sub-chapter. In the model learning processes of binary and multi-class settings, labeled and unlabeled samples are selected randomly from the training and testing data pool. Without biased quantification, to evaluate the accuracy measurements properly, 20 independent realizations were made for generating graphs, and tables and standard deviations of these runs are added into the results as well. Besides the accuracy measurements, to further assess the model performance, k statistics, weighted mean $\bar{F}_1$ of F_1 calculations, overall accuracies (OA), average accuracies (AA), individual class accuracies are calculated. At last, visualizations of binary and multi-class settings are added for the specific number of labeled and unlabeled samples.

Furthermore, to increase the accuracy and explore the uncertainties of the newly proposed models, the uncertainty visualization approach is implemented as an active learning extension. In this process, the Cologne data set that is explained in chapter 4.1 is used with its binary classification settings. Uncertainty visualizations are created and with the help of the visualizations, most uncertain samples have been relabeled by and new uncertainties and labels have been displayed for comparison. Additionally, all the uncertainties of the newly proposed models have been compared with their statistical values for a certain amount of samples and all the uncertainty values have been scaled down for the consideration of all methods.

4.1 Data

In this section, two data sets that have been used in the experimental setup will be discussed. The first data set covers the geographic area of 1000x1000 meters and the second data set covers the area of 2000x2000 meters. They were captured by Very High Resolution (VHR) multispectral imagery with blue, green, red, and near-infrared bands from the WorldView-II sensor. Additionally, images have a geometric resolution of 0.5 meters.

The first data set was obtained in the city of Cologne, Germany on January 31, 2004. Moreover, it shows an urban area, mainly dominated by buildings. That is further shown in Fig. 18 below. The image also consists of shadow areas and facades of singular buildings that can be further seen through the direction of sensor view due to the of-nadir acquisition. Furthermore, pixels are organized into six thematic classes such as “bush/tree”, “roof”, “meadow”, “facade”, “shadow”, and “other impervious surface” in Fig. 18 (b).

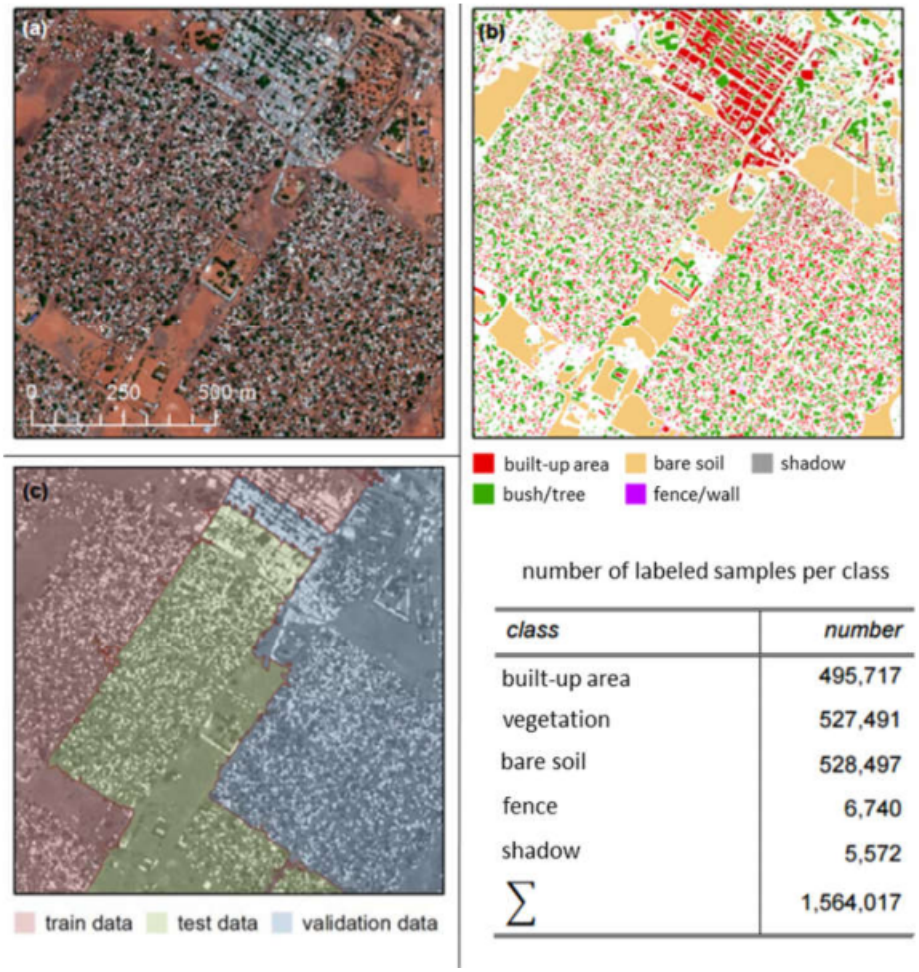


number of labeled samples per class

class	*number*
bush/tree	109,620
meadow	12,574
roof	196,928
facade	54,342
other impervious surface	24,233
shadow	283,741
$\sum$	681,438

Figure 18: (a)WorldView-II scene of Cologne, Germany; (b) Available labeled samples per class; (c) spatially disjoint training, testing, and validation areas.

The second data set was captured over the Hagadera refugee camp in Kenya, on March 01, 2012 with a 2000×2000 spatial resolution that is shown in Fig. 19. Moreover, data was spatially splited into train, test and validation areas and selection was made on determining a heterogeneously distributed settlement area. Therefore, this separation was made based on the objects in the image so that the image object stayed within the certain area as train, test, or validation. As a second consideration, it was aimed to split the area based on the idea of not overlapping over the features. This area also consists buildings that have various types of fences, walls, shadows and open spaces. At last, data was divided into five thematic classes such as “built-up area”, “bush/tree”, “bare soil”, “fence/wall”, and “shadow” in Fig. 19(b).



number of labeled samples per class

class	*number*
built-up area	495,717
vegetation	527,491
bare soil	528,497
fence	6,740
shadow	5,572
$\sum$	1,564,017

Figure 19: (a)WorldView-II scene of Hagadera Refugee Camp, Kenya; (b) Available labeled samples per class; (c) spatially disjoint training, testing, and validation areas.

4.2 Results from Data set I: Cologne

The methodology is applied to the Cologne data set. Results are presented in two different graphs and figure types. Semi-supervised methods (i.e., VSVM-SL-U, VSVM-SL-VU) are mainly compared with the VSVM with and without self-learning constraints strategy, for the reason that these methods are built upon these strategies. Additionally, benchmark semi-supervised methods (i.e., SVM-SL-U) are compared with the initial SVM model and for the SVM-M model where all the invariances were treated as features of the data and did not consider as virtual samples (Geiß et al., 2019).

In the first section of the Cologne data set results, graphs for binary classification settings are discussed. These graphs differ due to the invariances to scale and shape, and they were generated for 20 independent realizations considering the different number of labeled samples per class. In the figure creation, unlabeled samples are taken into the models as a hyperparameter, and the realization process is repeated for various amounts of unlabeled sample sizes, and from these unlabeled sample sizes so as 20,40,60 for per class, the best resulting sizes were selected. Therefore, in these figures, highlighted light blue (VSVM-SL-VU) and blue color (VSVM-SL-U) represent

the semi-supervised methods, green-colored lines present VSVM approaches. Dashed green lines for VSVM-SL and green lines for VSVM approach. At last, black colored lines present SVM methods, black line for SVM, dashed line for SVM-M and dashed and dotted line for the benchmark SVM-SL-U method. Additionally, graphs show mean accuracy values on the vertical plane where tables consider multiple statistic measurements. Moreover, the number of labeled samples varies in the range of 10 to 200 per class for each graph.

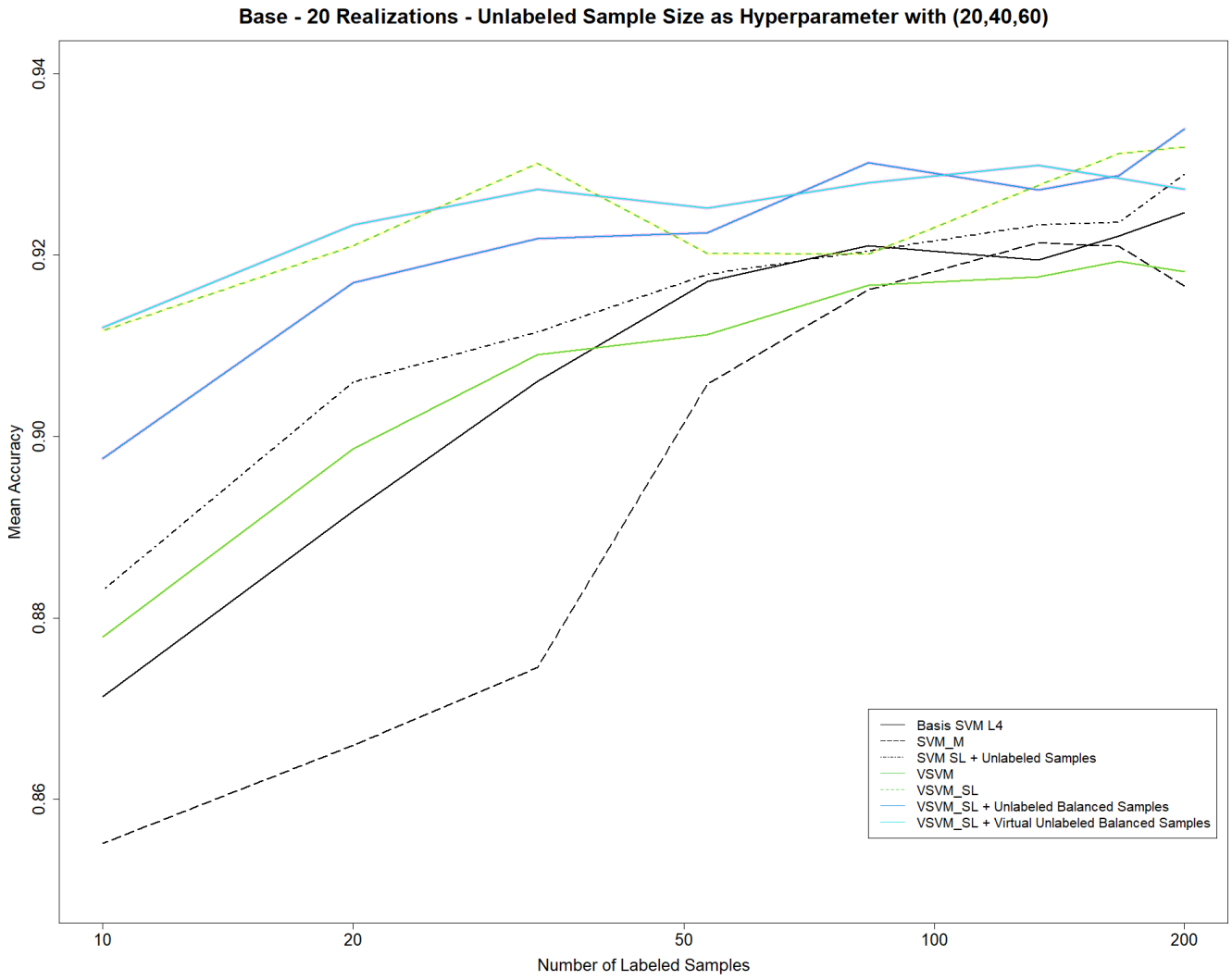


Figure 20: Cologne- Binary classification setting- invariance to scale- Mean Accuracy

Moreover, Fig 20- 22 demonstrates the overall accuracies of the semi-supervised and other methods for varying numbers of samples used in model learning for Cologne binary classification settings. Figure 20 shows that VSVM-SL-VU approach has the best overall accuracy measurements for the few labeled samples and it is followed by the VSVM-SL and VSVM-SL-U methods. As another result, it can be seen that the benchmark method SVM-SL-U performs better compared to initial SVM models for the few labeled samples. In Fig 22, the best accuracies archived by VSVM-SL method, followed by semi-supervised methods for the few labeled samples.

Additionally, in both of the figures, all the methods reach a plateau when the number of labeled samples increased. However, semi-supervised methods remain on top compared with supervised methods. In Fig 21- 23, kappa values are displayed instead of overall accuracy for the Cologne binary classification settings, and the results are supported with similar patterns from the kappa measurements. As an only difference, the benchmark method, SVM-SL-U, achieves higher kappa measurements for the few labeled samples and the higher number of samples.

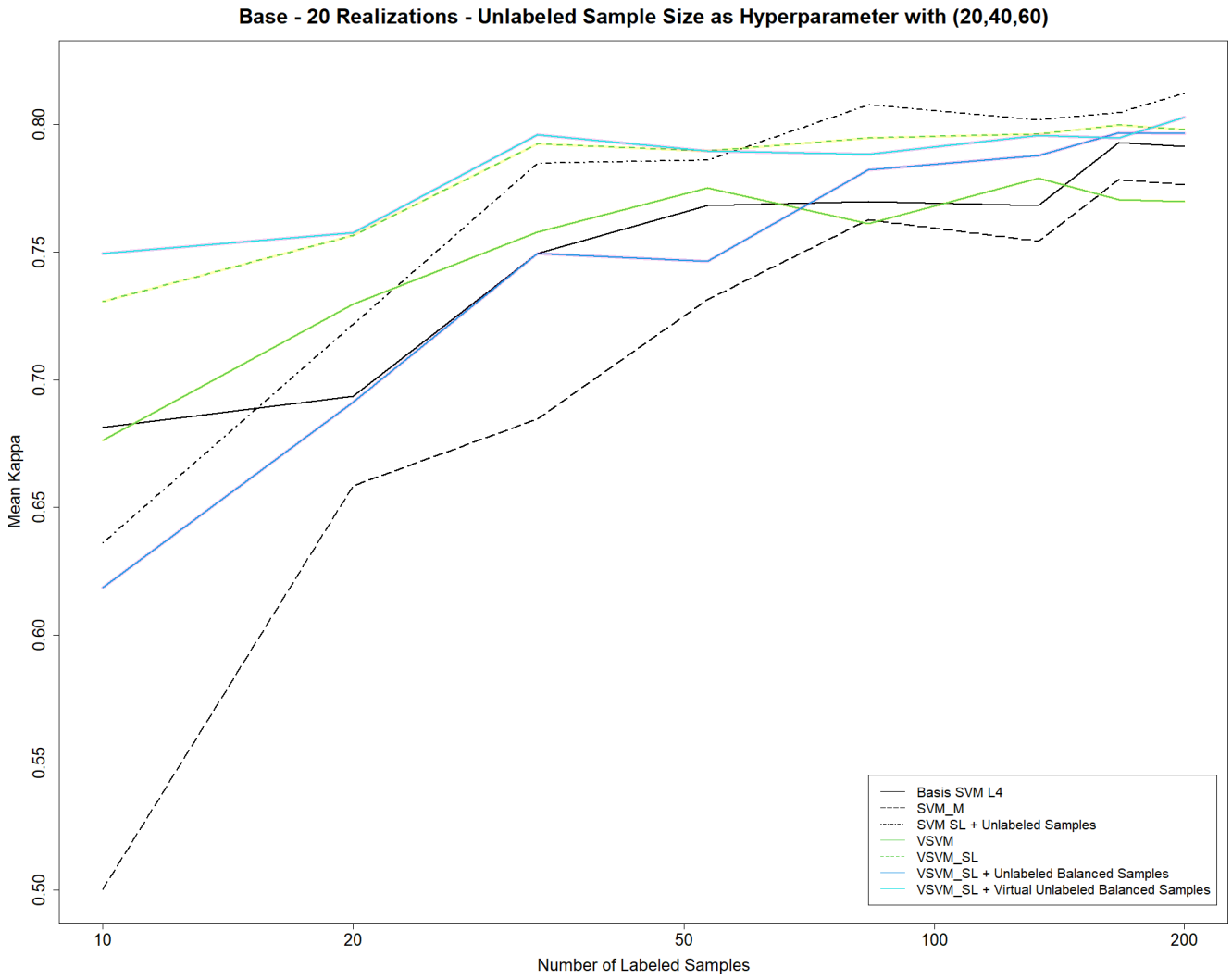


Figure 21: Cologne- Binary classification setting- invariance to shape- Kappa Statistics

Results with 40 labeled samples and 20 unlabeled samples with 20 realizations are displayed in tables 1 - 2 for the Cologne binary classification settings. As a remark, in terms of binary classes, SVM-SL-U method had better F1 and accuracy values with lower standard deviation especially in the invariance of shape settings. However, SVM and SVM-M methods performed better in overall statistics. Meanwhile, semi-supervised methods VSVM-SL-U and VSVM-SL-VU had a dominant performance in terms of overall accuracy and kappa statistics. Also, these methods had an overall good classification performance regarding the class bush/tree. Therefore, it can be later seen in the classification results Fig. 25

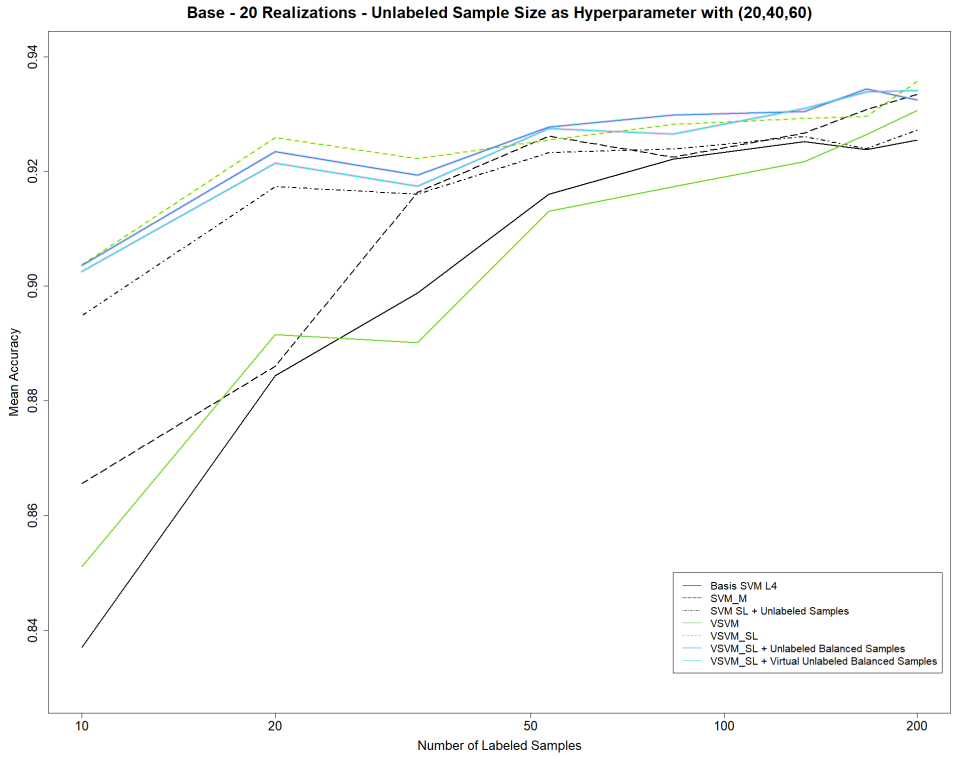


Figure 22: Cologne- Binary classification setting- invariance to shape-Mean Accuracy

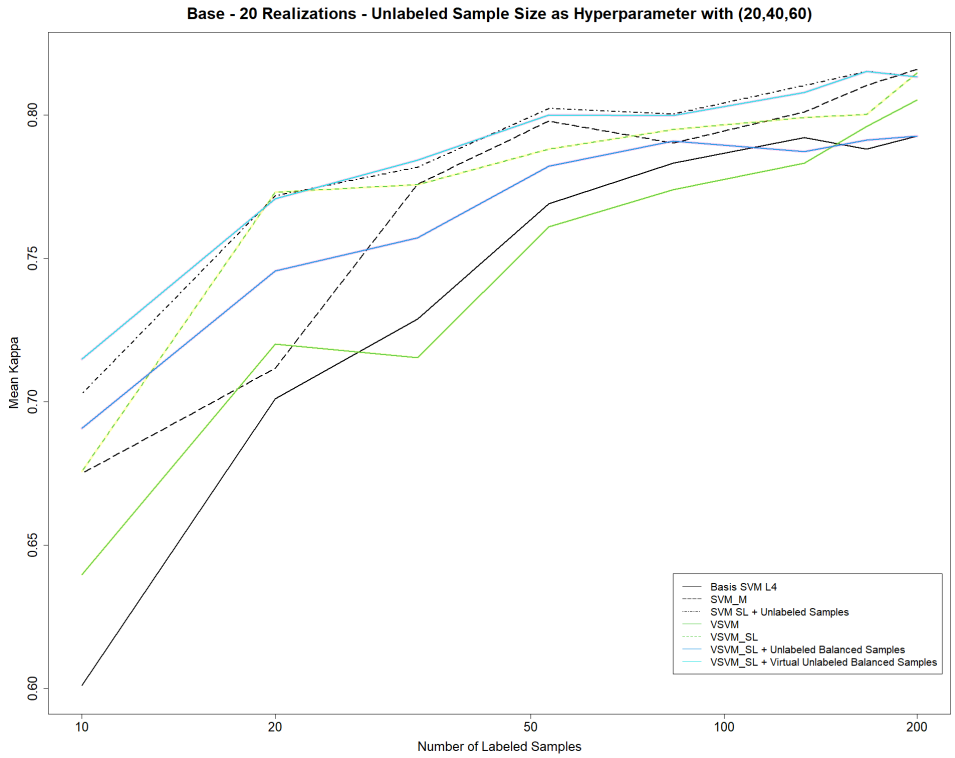


Figure 23: Cologne- Binary classification setting- invariance to shape-Kappa Statistics

Class	SVM		SVM-M		***SVM-SL-Un***	
	F1	Acc	F1	Acc	*F1*	*Acc*
Invariance to object scale; number of samples per class: 40 ; unlabeled samples : 20						
Bush/Tree	**82.82 (± 2.25)**	73.4 (± 4.7)	80.7 (± 1.65)	73.11 (± 2.82)	*80.96 (± 4.27)*	***75.45 (± 5.73)***
Other	**94.76 (± 1)**	**98.75 (± 0.92)**	94.37 (± 0.59)	97.41 (± 1.09)	*94.66 (± 1.33)*	*96.75 (± 1.4)*
Kappa	**77.69(±3.2)**		76.23(±2.71)		*75.93(±6.28)*	
F1	**92.78(±1.16)**		92.58(±0.87)		*92.32(±2.2)*	
OA	**91.89(±1.34)**		91.78(±0.99)		*91.42(±2.54)*	
AA	85.85(±1.79)		**86.16(±1.62)**		*85.51(±3.35)*	
Invariance to object shape; number of samples per class: 40 ; unlabeled samples : 20						
Bush/Tree	**82.83 (± 3.84)**	74.28 (± 4.9)	83.86 (± 1.91)	72.78 (± 2.89)	*81.78 (± 1.38)*	***76.45 (± 0.55)***
Other	94.89 (± 1.35)	98.3 (± 0.7)	94.78 (± 0.76)	**99.71 (± 0.02)**	***94.96 (± 0.27)***	*96.84 (± 0.79)*
Kappa	77.96(± 2.71)		**78.72(± 3.57)**		*76.1(± 3.85)*	
F1	92.89(± 0.96)		**93.01(± 1.34)**		*92.41(± 1.36)*	
OA	**92.02(± 1.12)**		92.1(± 1.57)		*91.53(± 1.57)*	
AA	85.99(± 1.53)		**86.33(± 1.98)**		*85.57(± 2.06)*	

Table 1: Cologne - Binary classification setting- Classification accuracies and other configurations compared to benchmark semi-supervised method.

Class	VSVM		VSVM-SL		***VSVM-SL-Un***		***VSVM-SL-Vun***	
	F1	Acc	F1	Acc	*F1*	*Acc*	*F1*	*Acc*
Invariance to object scale; number of samples per class: 40 ; unlabeled samples : 20								
Bush/Tree	79.89 (± 2.16)	69.07 (± 3.73)	80.9 (± 3.76)	70.93 (± 5.97)	*81.88 (± 4.26)*	***76.06 (± 5.15)***	***82.66 (± 2.06)***	*75.39 (± 5.74)*
Other	93.62 (± 0.97)	**98.64 (± 1.17)**	94.04 (± 1.5)	98.49 (± 0.72)	*94.91 (± 1.25)*	*97.07 (± 1.37)*	***94.92 (± 1.1)***	*97.92 (± 1.24)*
Kappa	77.15(±3.51)		78.15(±7.02)		***78.67(±4.92)***		*76.7(±3.87)*	
F1	92.58(±1.31)		92.95(±2.62)		***93.22(±1.73)***		*92.51(±1.35)*	
OA	91.65(±1.54)		92.1(±3.05)		***93.22(±1.99)***		*91.61(±1.56)*	
AA	85.58(±2.08)		86.4(±3.68)		***86.75(±2.58)***		*85.57(±2.05)*	
Invariance to object shape; number of samples per class: 40 ; unlabeled samples : 20								
Bush/Tree	**84.85 (± 0.83)**	75.32 (± 1.35)	83.54 (± 3.02)	76.54 (± 3.05)	*80.95 (± 3.18)*	***78.26 (± 3.45)***	*80.25 (± 5.41)*	*73.78 (± 7.28)*
Other	95.45 (± 0.3)	**99.23 (± 0.15)**	**95.3 (± 0.9)**	97.85 (± 0.81)	*94.96 (± 0.84)*	*95.86 (± 1.01)*	*94.28 (± 1.92)*	*96.94 (± 1.5)*
Kappa	78(± 3.68)		**78.4(± 4.49)**		*77.96(± 3.21)*		*78.35(± 4.12)*	
F1	92.85(± 1.38)		93.14(± 1.62)		*93.17(± 0.97)*		***93.24(± 1.38)***	
OA	91.95(± 1.61)		92.35(± 1.88)		*92.46(± 1.06)*		***92.5(± 1.57)***	
AA	85.95(± 2.11)		86.77(± 2.63)		***87.12(± 1.52)***		*87.1(± 2.21)*	

Table 2: Cologne - Binary classification setting- Classification accuracies and other configurations of semi-supervised methods

In the evaluation of the Cologne binary classification settings, single realization is picked from tables 1 and 2, and visualized in figure 24 and 25. Therefore, figure 24 supports that invariance to shape settings on SVM-SL-U approach has a better impact on statistical values, that is why classification of bush/tree class is distinguishable from the other methods. Also, in figure 25, semi-supervised methods perform slightly better although the statistical values are close to each other.

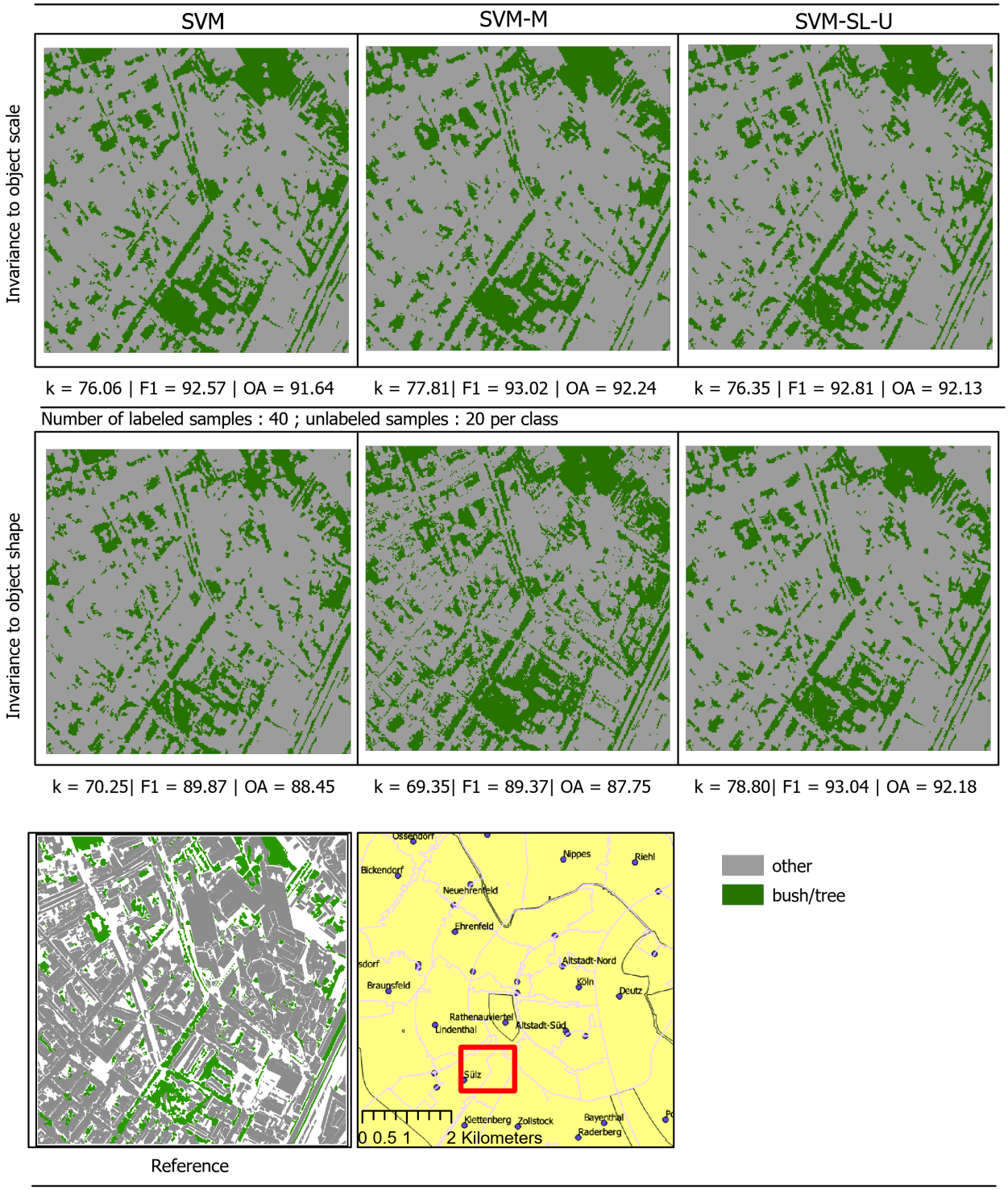


Figure 24: Visualization of results from single realization for binary classification setting for Cologne

Figure 25: Visualization of results from single realization for binary classification setting for Cologne

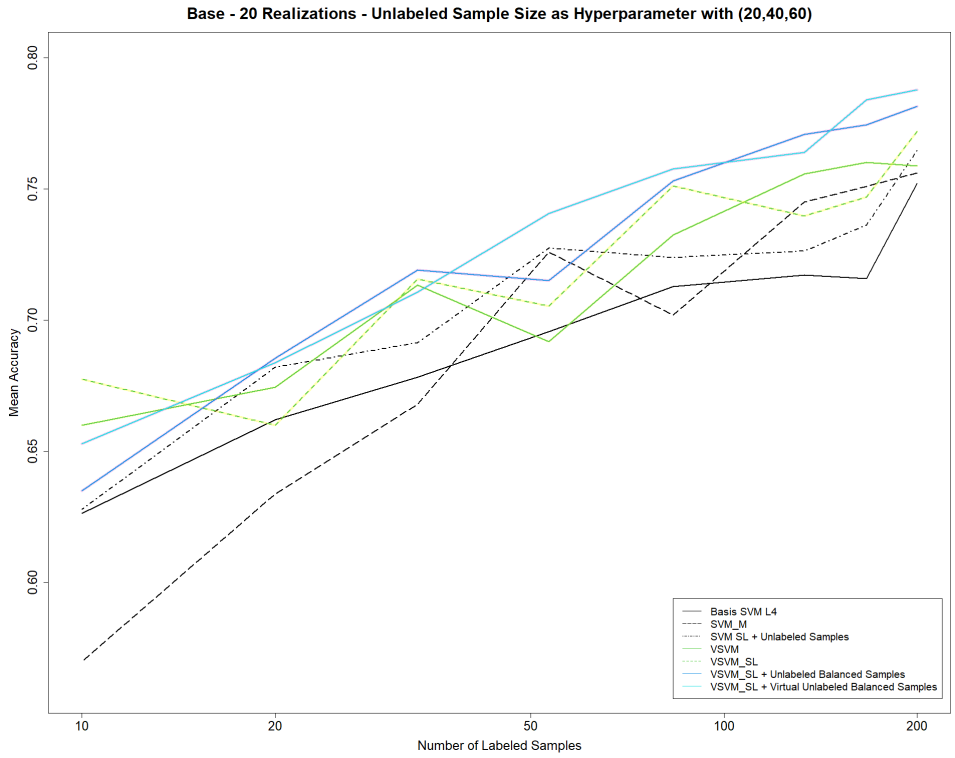


Figure 26: Cologne- Multi-class classification setting- invariance to Scale- Mean Accuracy

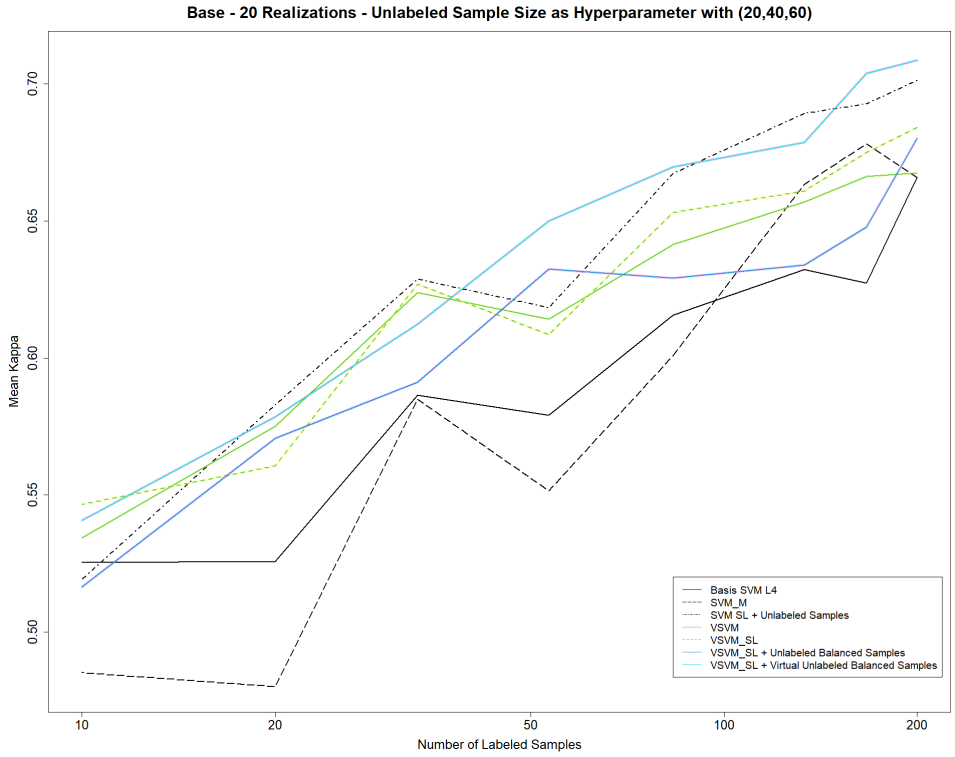


Figure 27: Cologne- Multi classification setting- invariance to scale- Kappa Statistics

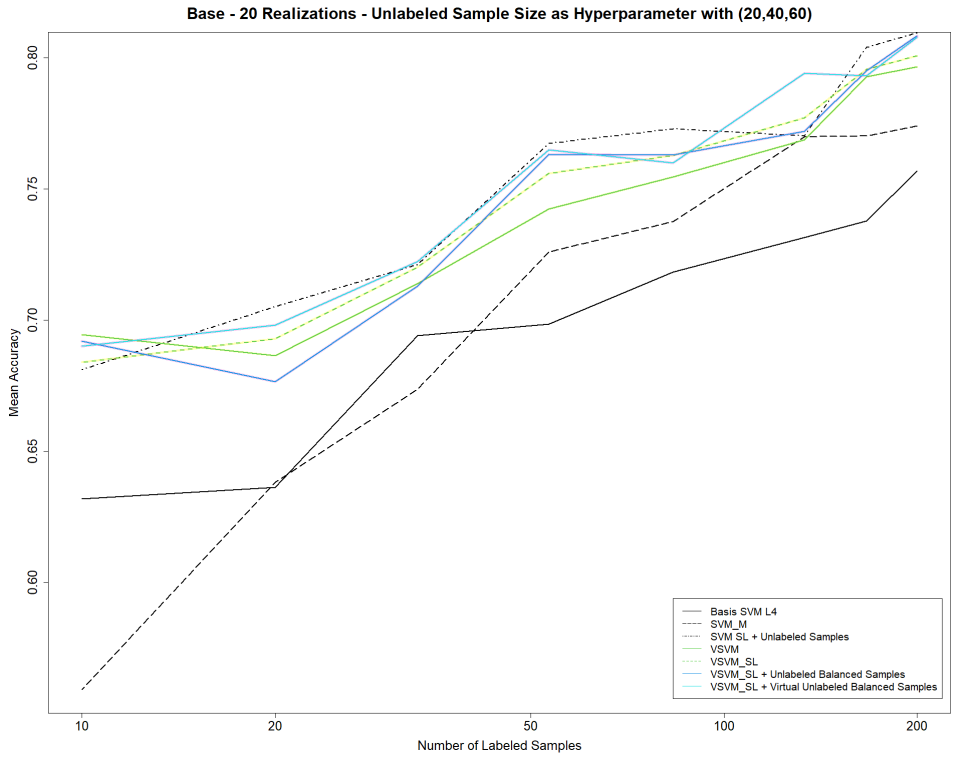


Figure 28: Cologne- Multi-class classification setting- invariance to shape-Mean Accuracy

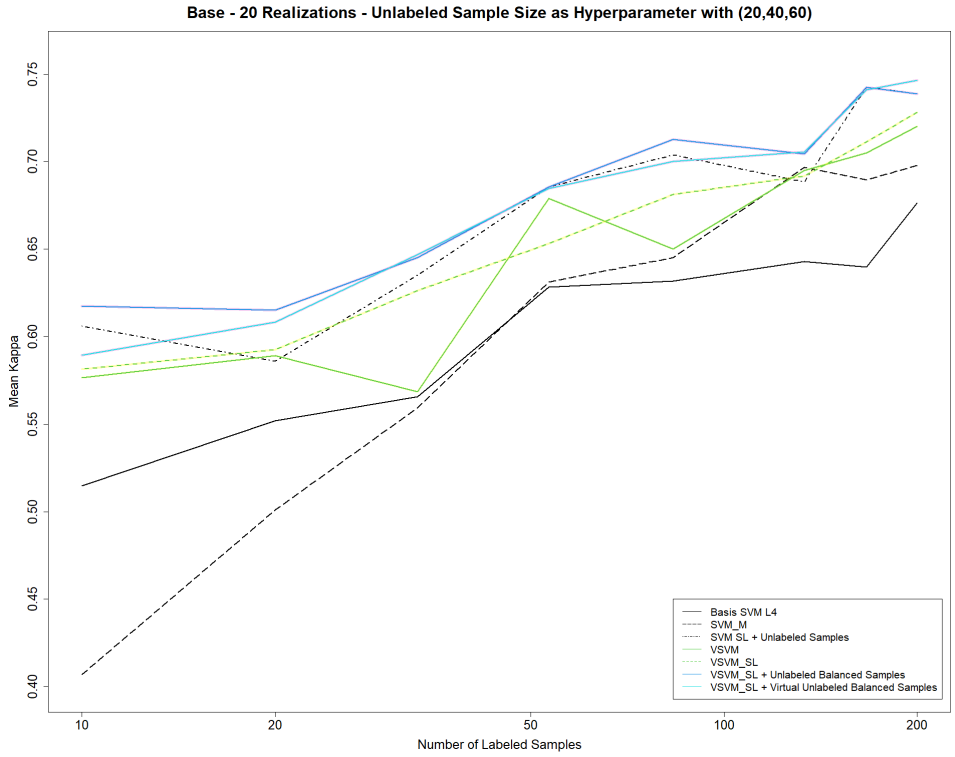


Figure 29: Cologne- Multi-Class classification setting- invariance to shape-Kappa Statistics

Class	SVM		SVM-M		***SVM-SL-Un***	
	F1	Acc	F1	Acc	F1	Acc
Invariance to object scale;number of samples per class: 40 ; unlabeled samples : 20						
Bush/Tree	82.08 (± 2.07)	80.19 (± 2.45)	81.99 (± 3.39)	**81.63 (± 3.11)**	***82.22 (± 2.75)***	*80.29 (± 3.42)*
Meadow	48.3 (± 5.67)	35.29 (± 6.82)	**55.38 (± 9.39)**	**48.63 (± 17.49)**	*51.47 (± 7.49)*	*37.86 (± 8.38)*
Roof	54.77 (± 8.22)	76.32 (± 4.88)	**67.72 (± 5.19)**	73.08 (± 4.75)	*58.04 (± 8.76)*	***78.2 (± 3.7)***
Facade	**57.38 (± 4.46)**	**50.62 (± 7.23)**	54.21 (± 5.3)	48.8 (± 10.98)	*56.16 (± 4.52)*	*49.22 (± 7.38)*
Other imp. surf.	35.69 (± 7.73)	25.03 (± 7.32)	18.4 (± 5.86)	24.56 (± 11.09)	***38.33 (± 8.08)***	***27.39 (± 7.98)***
Shadow	86.51 (± 1.55)	90.66 (± 1.57)	83.84 (± 1.62)	82.98 (± 2.74)	***86.75 (± 1.41)***	***91.19 (± 1.22)***
Kappa	60.91 (± 3.97)		**63.34 (± 3.44)**		*62.44 (± 4.56)*	
F1	71.79 (± 2.89)		**73.67 (± 2.27)**		*72.91 (± 3.31)*	
OA	70.64 (± 3.27)		**73.68 (± 2.79)**		*71.81 (± 3.8)*	
AA	59.69 (± 2.25)		59.95 (± 4.39)		***60.69 (± 2.32)***	
Invariance to object shape; number of samples per class: 40 ; unlabeled samples : 20						
Bush/Tree	81.54 (± 1.69)	78.99 (± 2.16)	84.03 (± 2.98)	**83.01 (± 3.48)**	***85.68 (± 1.97)***	*82.74 (± 2.72)*
Meadow	48.04 (± 4.37)	34.15 (± 5.07)	48.37 (± 7.01)	35.15 (± 8.27)	***57.01 (± 6.75)***	***49.73 (± 10.94)***
Roof	55.82 (± 7)	**78.96 (± 3.38)**	59.93 (± 10.69)	75.93 (± 5.25)	***70.74 (± 2.83)***	*76.5 (± 4.02)*
Facade	**58.27 (± 2.92)**	**52.26 (± 6.15)**	57.93 (± 3.27)	49.93 (± 4.98)	*54.25 (± 4.51)*	*47.34 (± 5.89)*
Other imp. surf.	35.06 (± 5.81)	24.41 (± 5.54)	42.71 (± 11.28)	31.21 (± 9.92)	***49.93 (± 8.51)***	***42.49 (± 8.26)***
Shadow	87.02 (± 1.32)	90.61 (± 0.99)	86.25 (± 4.22)	**91.96 (± 3.12)**	***87.41 (± 1.8)***	*91.41 (± 2.16)*
Kappa	61.39 (± 3.37)		63.74 (± 6.79)		***69.55 (± 2.48)***	
F1	72.26 (± 2.45)		73.78 (± 4.9)		***77.78 (± 1.58)***	
OA	71.03 (± 2.82)		72.65 (± 5.63)		***77.73 (± 1.88)***	
AA	59.89 (± 1.65)		61.2 (± 3.25)		***65.03 (± 2.46)***	

Table 3: Cologne - Multi-class classification setting- Classification accuracies and other configurations of benchmark semi-supervised method.

Class	VSVM		VSVM-SL		***VSVM-SL-Un***		***VSVM-SL-Vun***	
	F1	Acc	F1	Acc	*F1*	*Acc*	*F1*	*Acc*
Invariance to object scale;number of samples per class: 40 ; unlabeled samples : 20								
Bush/Tree	82.93 (± 3.02)	80.67 (± 3.17)	**83.5 (± 2.56)**	**80.91 (± 3.42)**	*83.27 (± 2.76)*	*80.74 (± 2.96)*	*81.76 (± 3.66)*	*78.44 (± 4.44)*
Meadow	50.53 (± 6.96)	37.58 (± 7.63)	**53 (± 6.45)**	**40.67 (± 7.81)**	*51.83 (± 6.63)*	*39.19 (± 7.61)*	*48.18 (± 9.16)*	*37.23 (± 8.63)*
Roof	63.4 (± 5.6)	**79.19 (± 4.05)**	63.96 (± 8.19)	76.84 (± 4.28)	*63.12 (± 8.12)*	*77.42 (± 4.78)*	**64.3 (± 5.17)**	*72.76 (± 8.27)*
Facade	**57.49 (± 3.14)**	**51.14 (± 6.8)**	57.43 (± 2.69)	49.86 (± 5.84)	*57.09 (± 2.95)*	*49.49 (± 6.7)*	*51.49 (± 5.93)*	*43.35 (± 7.29)*
Other imp. surf.	43.15 (± 7.82)	31.49 (± 7.45)	41.32 (± 7.37)	31.78 (± 7.97)	*41.73 (± 8.87)*	*31.8 (± 8.84)*	***43.58 (± 5.7)***	***34.06 (± 5.58)***
Shadow	**87.33 (± 1.37)**	**91.79 (± 1.53)**	86.57 (± 1.48)	91.13 (± 1.68)	*87 (± 1.5)*	*91.04 (± 1.53)*	*83.86 (± 3.76)*	*90.79 (± 1.71)*
Kappa	**65.24 (± 3.67)**		65.2 (± 3.92)		*65.06 (± 4.25)*		*63.09 (± 4.68)*	
F1	**75.08 (± 2.43)**		74.99 (± 2.81)		*74.86 (± 2.93)*		*73.2 (± 3.61)*	
OA	74.09 (± 2.96)		**74.16 (± 3.19)**		*74.03 (± 3.45)*		*72.55 (± 3.82)*	
AA	**61.98 (± 2.31)**		61.86 (± 2.42)		*61.62 (± 2.32)*		*59.44 (± 3.4)*	
Invariance to object shape; number of samples : 40 ; unlabeled samples : 20								
Bush/Tree	**86.62 (± 1.67)**	**83.43 (± 2.38)**	85.19 (± 1.7)	81.71 (± 2.78)	*84.72 (± 2.54)*	*81.44 (± 2.58)*	*84.87 (± 2.04)*	*81.58 (± 2.61)*
Meadow	**59.58 (± 6.01)**	50.71 (± 9.13)	55.11 (± 7.25)	46.79 (± 9.65)	*55.62 (± 7.42)*	*47.35 (± 12.23)*	***55.65 (± 7.76)***	*49.28 (± 14.96)*
Roof	68.02 (± 6.18)	**78.48 (± 3.33)**	70.37 (± 3.18)	76.31 (± 4.6)	***71.07 (± 2.49)***	*75.78 (± 4.69)*	*70.01 (± 2.81)*	*75.44 (± 5.64)*
Facade	**56.03 (± 5.94)**	**52.71 (± 5.93)**	53.84 (± 4.17)	45.65 (± 4.96)	*54.67 (± 4.33)*	*46.99 (± 4.81)*	*54.45 (± 3.71)*	*46.3 (± 5.25)*
Other imp. surf.	44.85 (± 10.61)	35.3 (± 9.85)	48.05 (± 9.96)	41.03 (± 6.67)	*46.12 (± 10.38)*	*41.35 (± 7.79)*	***49.83 (± 9.61)***	***41.94 (± 7.73)***
Shadow	**88.11 (± 1.75)**	90.61 (± 1.91)	87.08 (± 1.76)	91.61 (± 2.03)	*86.97 (± 2.11)*	*91.58 (± 1.79)*	*86.83 (± 2.37)*	***91.94 (± 1.62)***
Kappa	68.92 (± 3.85)		68.93 (± 2.56)		***68.94 (± 2.65)***		*68.7 (± 3.09)*	
F1	77.44 (± 2.56)		77.32 (± 1.65)		***77.45 (± 1.67)***		*77.18 (± 2.06)*	
OA	77.22 (± 3.02)		77.23 (± 1.95)		***77.24 (± 2.03)***		*77.03 (± 2.38)*	
AA	**65.21 (± 2.81)**		63.85 (± 2.79)		*64.08 (± 3.03)*		*64.41 (± 3.53)*	

Table 4: Cologne - Multi-class classification setting- Classification accuracies and other configurations of semi-supervised methods

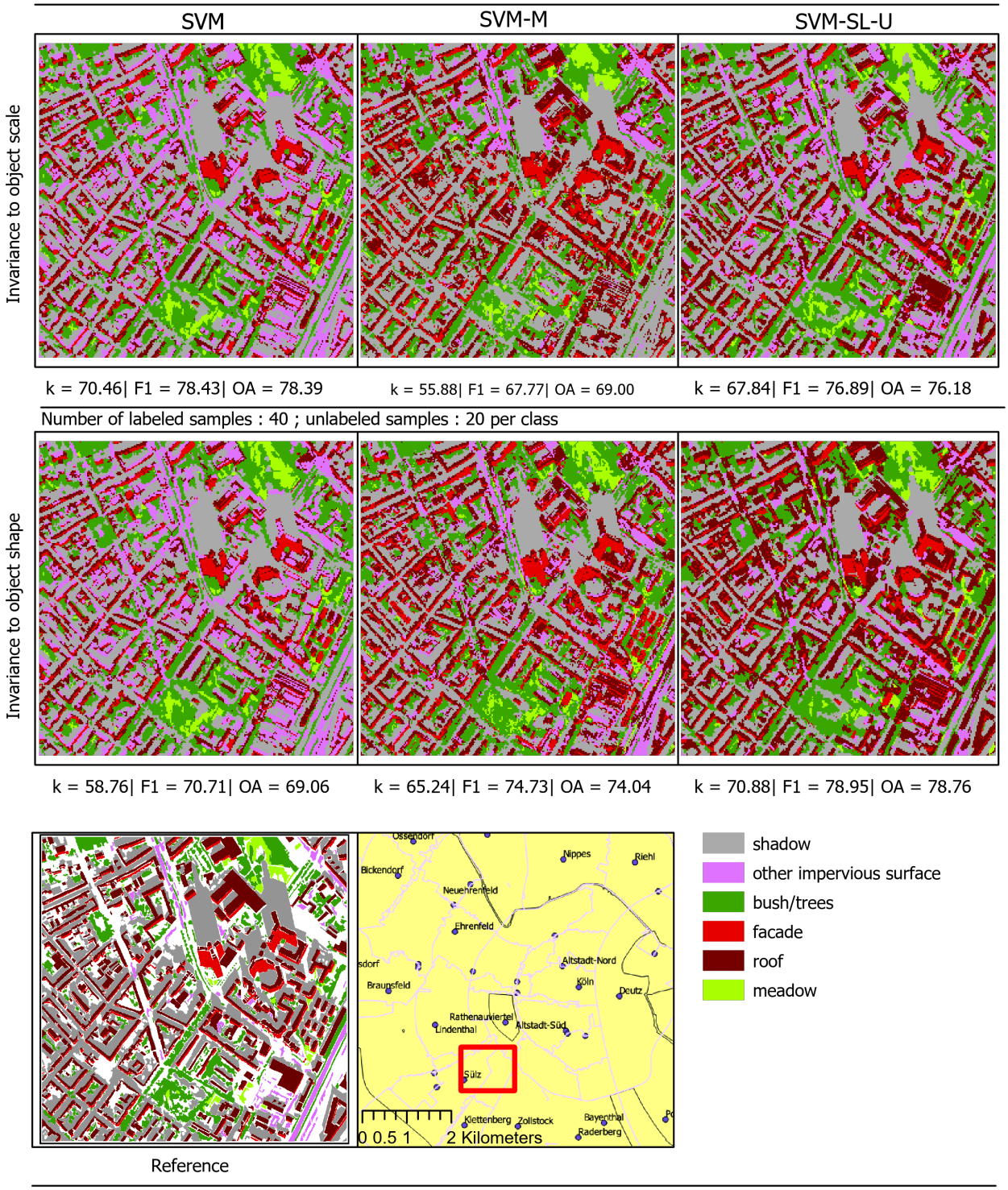


Figure 30: Visualization of results from single realization for multi-class classification setting for Cologne

Figure 31: Visualization of results from single realization for multi-class classification setting for Cologne

In the evaluation of multi-class classification settings, figure 26 and 27 for invariance to scale settings indicate the performance of the VSVM-SL-VU method for the whole range of the samples. Besides, the VSVM-SL-VU approach followed by VSVM-SL-U and VSVM-SL approach and semi-supervised methods remains on top. Additionally, in the kappa measurements of invariances of scale, the benchmark method SVM-SL-U shows a robust performance with the VSVM-SL-VU approach. Furthermore, table 3 results are supported by the visualization in figure 30 with a single run. The big majority of the classes have significantly better statistical values, and the difference can be seen in shadow and bush/tree classes. At last, in the invariances to shape settings (Table 4), semi-supervised methods (especially VSVM-SL-U) perform higher overall accuracy, kappa, and F1 scores, meanwhile, VSVM-SL-VU methods achieve favorable class performance with the VSVM method. This can be further seen in the classification map figure 31 where the spatial distribution of the classes is accurate.

Overall, binary, and multi-class settings of the Cologne data set support the accuracy improvements when unlabeled samples are fed into the model learning. Therefore, more favorable results obtained for the ranges of few labeled samples and also semi-supervised models stay on top when all the methods reach a plateau.

4.3 Results from Data set II: Hagadera

In the second part of the results section, the methodology is applied to the Hagadera data set. Binary and multi-class classification settings are used to evaluate the performance of semi-supervised methods on different data. Generated graphs (Fig 32, 33,34,35, 38,39,40,41) follows the same structure as Cologne data set. They include 20 realizations for the model runs and they involve the labeled sample range 10 to 200 per class. As the only difference in the hyperparameter selection, additional unlabeled sample sizes such as 80 and 100 are involved and further evaluated in the graphs. Graphs for binary settings indeed showed that semi-supervised methods (VSVM-SL-VU and VSVM-SL-U) have better overall accuracy values for the few labeled samples both for the invariances of scale and shape (Fig. 32 and 34). As a difference, in kappa statistic graphs, semi-supervised methods have slightly lower kappa values compared to the VSVM method, yet they reach higher values when all the models reach a plateau on higher labeled samples per class (Fig. 33 and 35). Furthermore, graphs have been followed by the tables (Table 5 and 6) indicating the results for 40 labeled samples and 20 unlabeled samples with the 20 realizations. Table results show that although semi-supervised methods perform slightly lower overall kappa, F1, and accuracy calculations than the previous methods, they have better class-based results and it is further noticeable in the visualization results in Fig. 36 and 37.

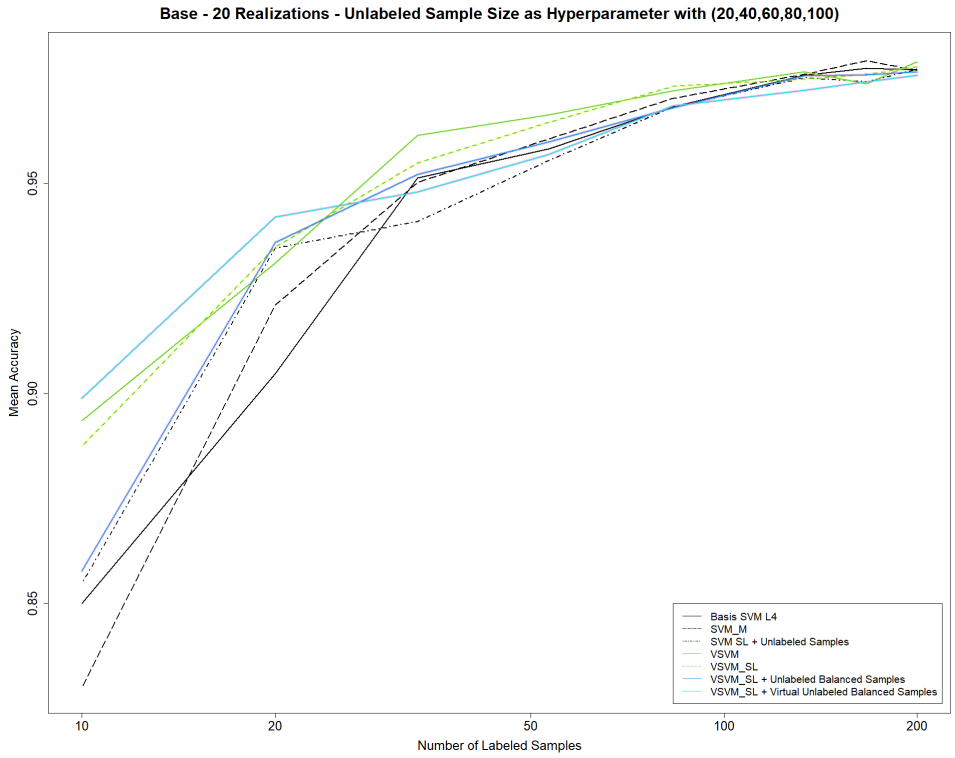


Figure 32: Hagadera- Binary- class classification setting- invariance to scale- Mean Accuracy

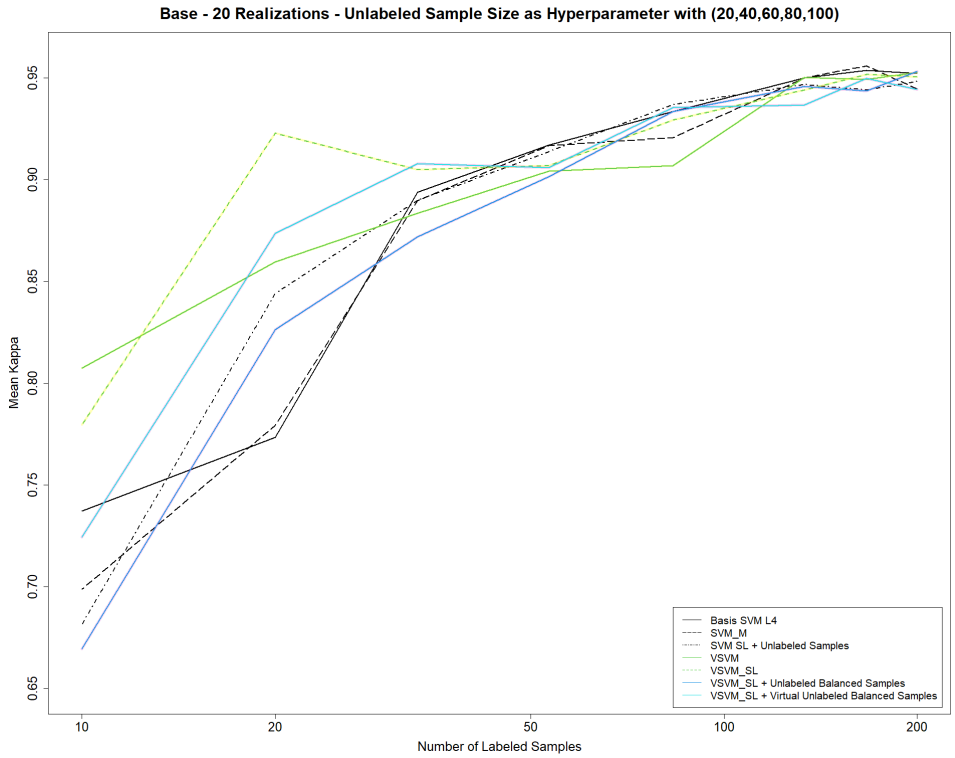


Figure 33: Hagadera- Binary- class classification setting- invariance to scale- Kappa Statistics

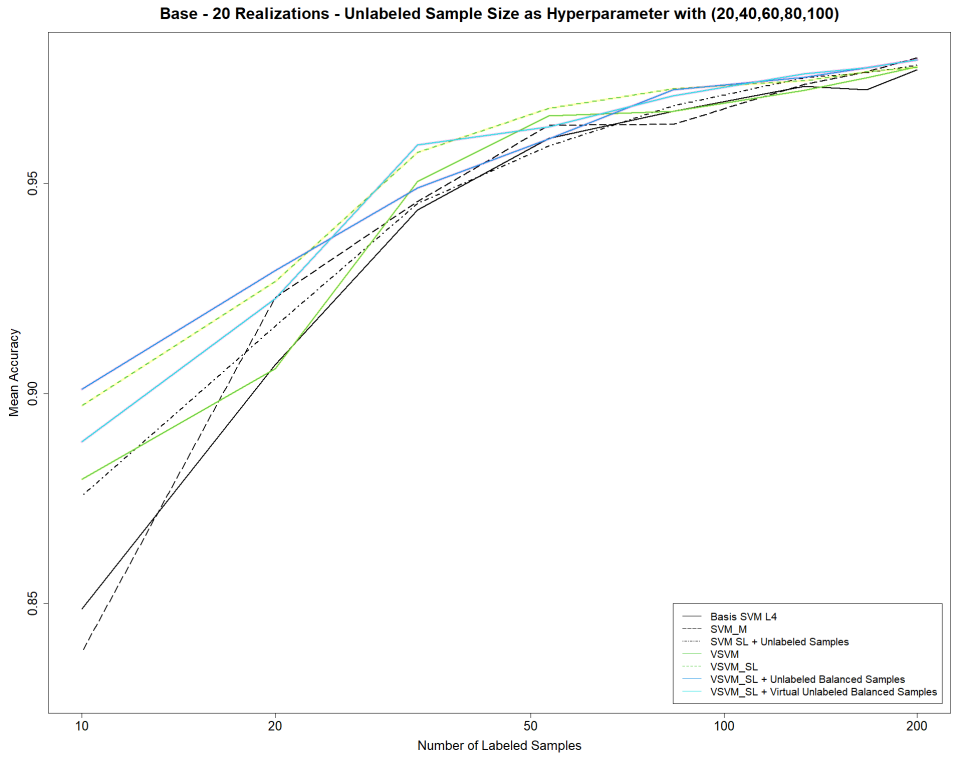


Figure 34: Hagadera- Binary- class classification setting- invariance to shape- Mean Accuracy

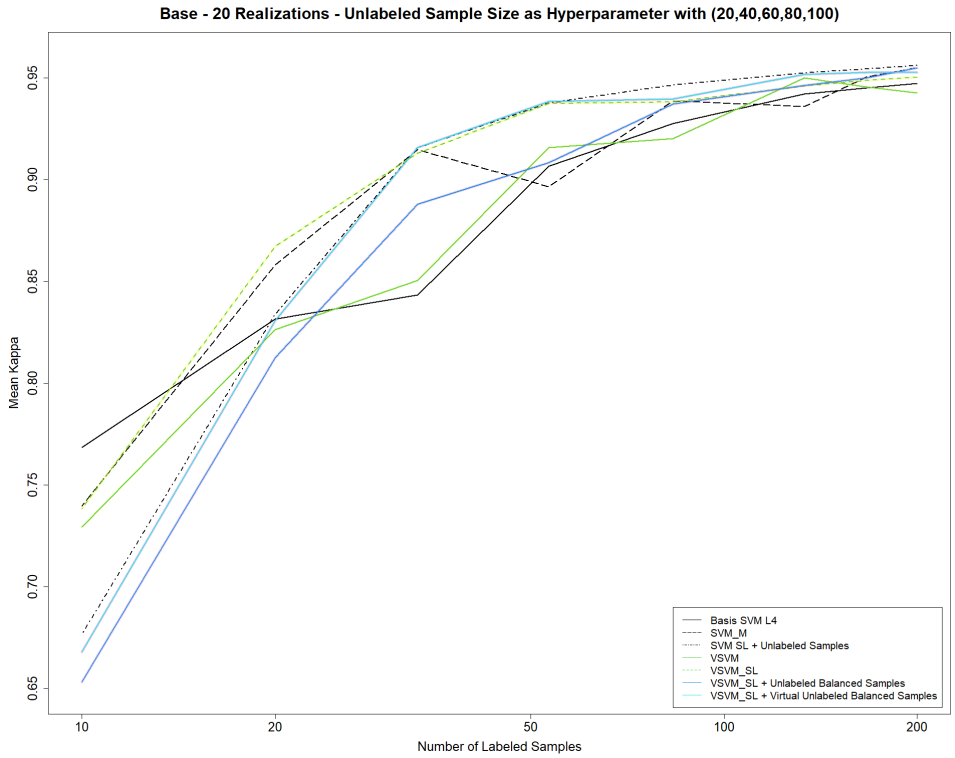


Figure 35: Hagadera- Binary- class classification setting- invariance to shape- Kappa Statistics

Class	SVM		SVM-M		***SVM-SL-Un***	
	F1	Acc	F1	Acc	*F1*	*Acc*
Invariance to object scale; number of samples: 40 ; unlabeled samples : 20						
Built-up area	**96.34 (± 0.79)**	94.6 (± 0.9)	94.6 (± 0.9)	84.19 (± 2.19)	*95.79 (± 0.65)*	***96.09 (± 1.32)***
Other	98.01 (± 0.41)	**99.01 (± 0.74)**	**99.01 (± 0.74)**	94.92 (± 0.87)	*97.79 (± 0.31)*	*97.65 (± 1.03)*
Kappa	**94.01(± 1.44)**		78.72(± 3.57)		*93.43(± 1.12)*	
F1	**97.29(± 0.66)**		91.07(± 1.6)		*97.05(± 0.49)*	
OA	**97.27(± 0.67)**		92.1(± 1.57)		*97.04(± 0.49)*	
AA	96.7(± 0.92)		86.33(± 1.98)		***96.89(± 0.49)***	
Invariance to object shape; number of samples : 40 ; unlabeled samples : 20						
Bush/Tree	96.21 (± 1.54)	**96.28 (± 0.71)**	73.31 (± 3.3)	84.19 (± 2.19)	***96.75 (± 0.17)***	*95.48 (± 1.26)*
Other	98.13 (± 1.46)	**98.04 (± 0.3)**	**99.7 (± 0.03)**	94.92 (± 0.87)	*97.02 (± 1.23)*	*97.66 (± 0.61)*
Kappa	**93(± 2.13)**		78.72(± 3.57)		*91.89(± 2.04)*	
F1	**96.83(± 0.99)**		91.07(± 1.6)		*96.37(± 0.9)*	
OA	**96.8(± 1.02)**		92.1(± 1.57)		*96.36(± 0.9)*	
AA	96.18(± 1.32)		86.33(± 1.98)		***96.41(± 0.89)***	

Table 5: Hagadera - Binary classification setting- Classification accuracies and other configurations of benchmark semi-supervised method.

Class	VSVM		VSVM-SL		***VSVM-SL-Un***		***VSVM-SL-Vun***	
	F1	Acc	F1	Acc	*F1*	*Acc*	*F1*	*Acc*
Invariance to object scale; number of samples per class: 40; unlabeled samples: 20								
Built-up area	96.11 (± 1.39)	96.25 (± 1.23)	96.22 (± 1.46)	**97.05 (± 0.56)**	***97.75 (± 0.7)***	*93.75 (± 3.61)*	*96.12 (± 0.96)*	*95.73 (± 1.69)*
Other	97.96 (± 0.73)	97.79 (± 0.98)	**98.04 (± 0.7)**	97.64 (± 1.66)	*95.14 (± 3.29)*	*96.95 (± 1.58)*	*97.57 (± 1.27)*	***97.79 (± 0.85)***
Kappa	92.75(± 3.5)		**93.82(± 1.55)**		*92.43(± 2.67)*		*92.66(± 2.31)*	
F1	96.71(± 1.64)		**97.22(± 0.71)**		*96.6(± 1.19)*		*96.7(± 1.06)*	
OA	96.68(± 1.69)		**97.2(± 0.72)**		*96.59(± 1.19)*		*96.68(± 1.08)*	
AA	96.16(± 1.98)		**97(± 0.96)**		*96.44(± 1.24)*		*96.52(± 1.34)*	
Invariance to object shape; number of samples per class: 40; unlabeled samples: 20								
Bush/Tree	94.64 (± 1.08)	95.58 (± 1.07)	96.15 (± 0.83)	**95.84 (± 0.94)**	***97 (± 0.45)***	*95.72 (± 0.77)*	*96.24 (± 1.56)*	*95.11 (± 1.24)*
Other	**98.18 (± 1.29)**	97.64 (± 0.53)	97.66 (± 0.76)	97.82 (± 0.48)	*97.12 (± 0.58)*	***97.88 (± 0.39)***	*96.87 (± 0.62)*	*97.46 (± 0.66)*
Kappa	**94.09(± 1.16)**		94.03(± 1.58)		*93.42(± 1.6)*		*93.14(± 1.97)*	
F1	**97.34(± 0.53)**		97.32(± 0.71)		*97.05(± 0.71)*		*96.12(± 0.88)*	
OA	**97.31(± 0.53)**		97.3(± 0.71)		*97.04(± 0.71)*		*96.92(± 0.87)*	
AA	96.82(± 0.64)		97.08(± 0.79)		***97.09(± 0.67)***		*96.93(± 0.85)*	

Table 6: Hagadera - Binary classification setting- Classification accuracies and other configurations of semi-supervised methods

Figure 36: Visualization of results from single realization for binary classification setting for Hagadera

Figure 37: Visualization of results from single realization for binary classification setting for Hagadera

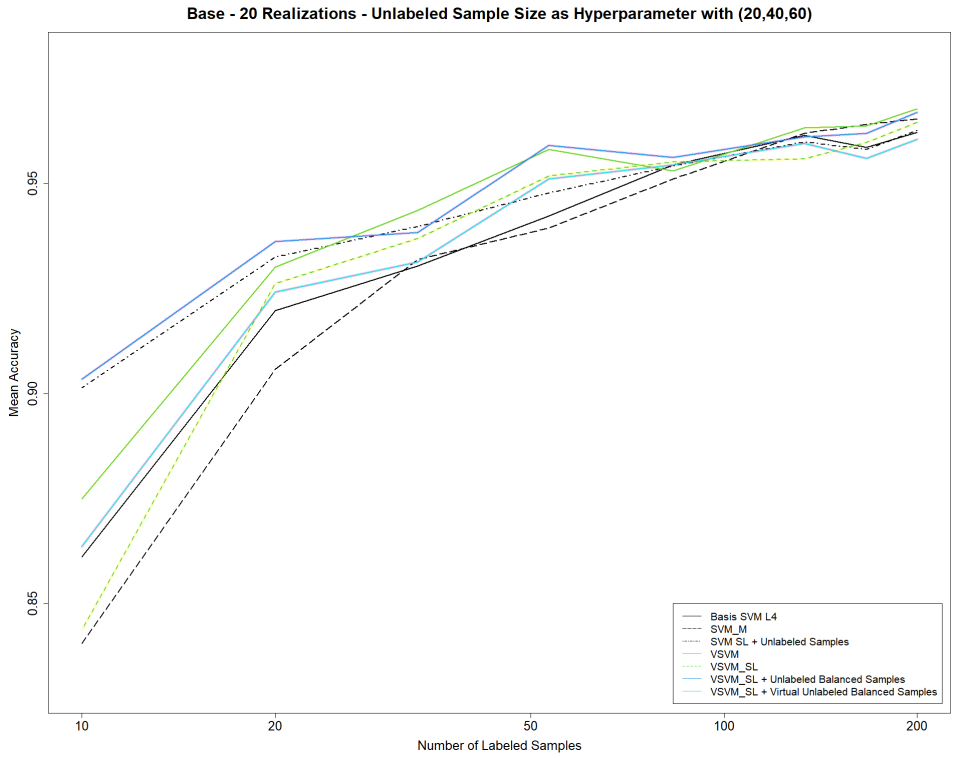


Figure 38: Hagadera- Multi-class classification setting- invariance to scale - Mean Accuracy

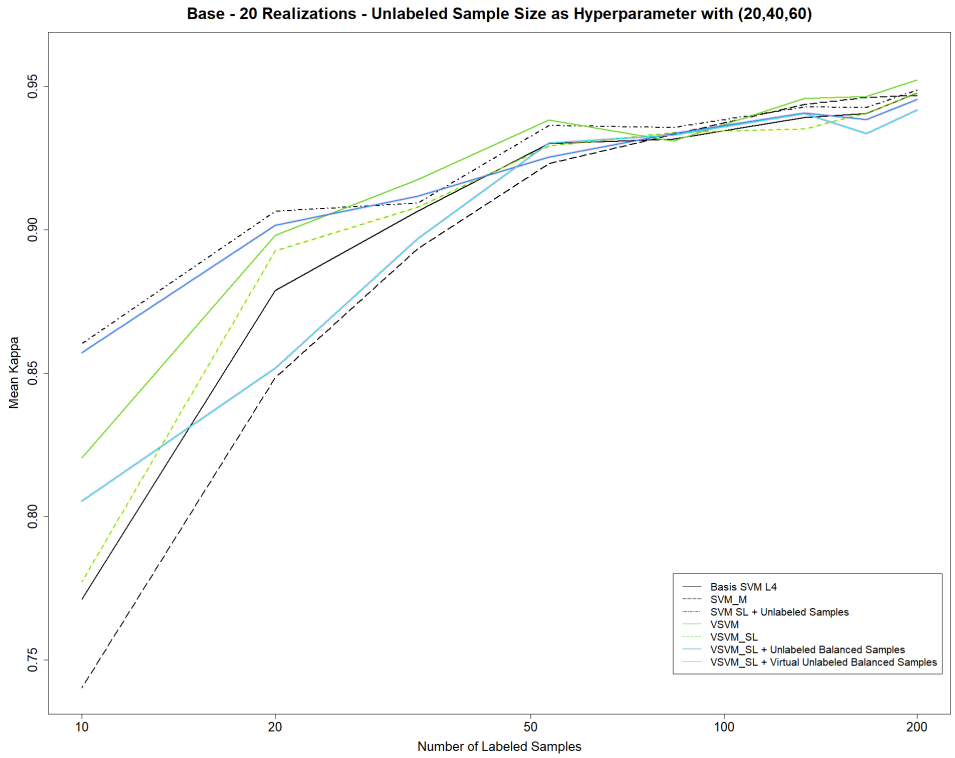


Figure 39: Hagadera- Multi-Class classification setting- invariance to scale - Kappa Statistics

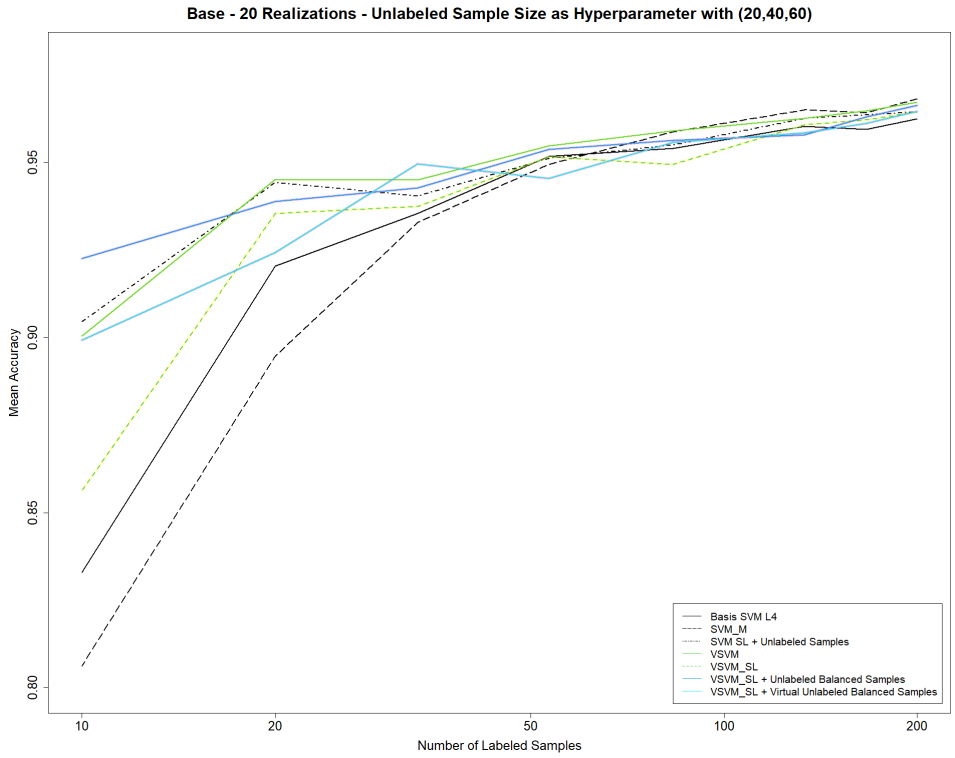


Figure 40: Hagadera- Multi-class classification setting- invariance to shape - Mean Accuracy

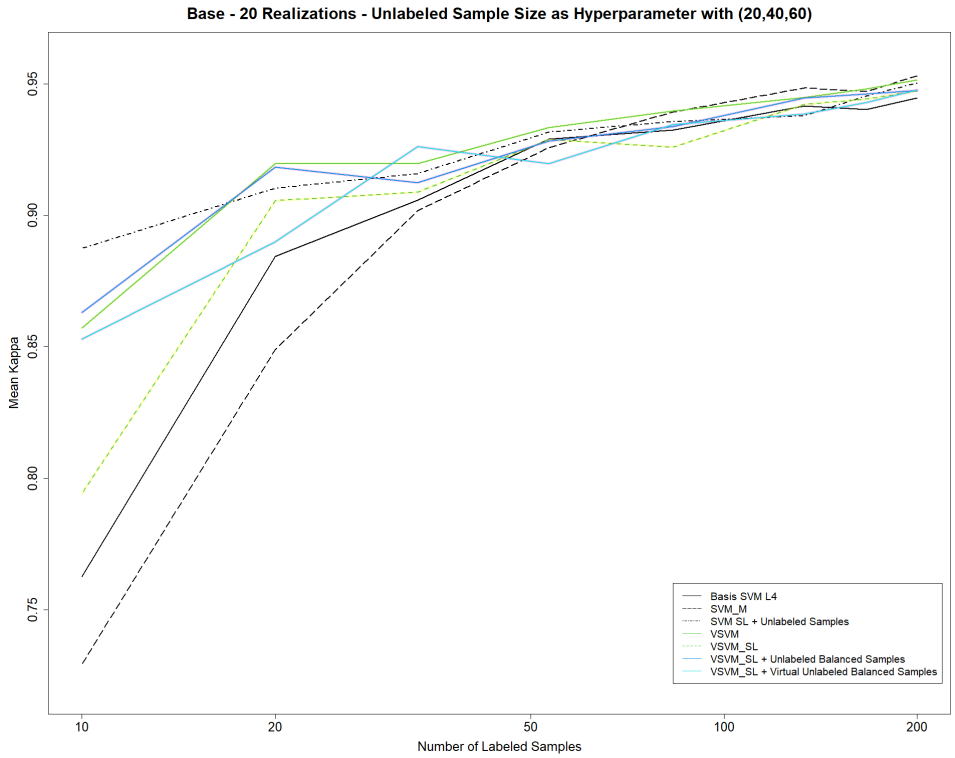


Figure 41: Hagadera- Multi-Class classification setting- invariance to shape - Kappa Statistics

Class	SVM		SVM-M		***SVM-SL-Un***	
	F1	Acc	F1	Acc	*F1*	*Acc*
Invariance to object scale; number of samples per class: 40 ; unlabeled samples : 20						
Built-up Area	93.46 (± 1.9)	98.3 (± 1.07)	**93.8 (± 1.81)**	**98.61 (± 0.8)**	*93.02 (± 2.36)*	*97.87 (± 2.27)*
Vegetation	97.11 (± 0.94)	99.55 (± 0.29)	97.45 (± 1.15)	**99.55 (± 0.25)**	***97.46 (± 0.87)***	*99.41 (± 0.35)*
Bare Soil	95.92 (± 2.01)	95.42 (± 2.23)	**97.19 (± 0.83)**	**96.95 (± 1.28)**	*96.03 (± 2.17)*	*95.57 (± 2.44)*
Fence	19.55 (± 8.09)	11.48 (± 5.67)	19.96 (± 4.4)	11.55 (± 3.03)	***20.02 (± 7.08)***	***11.8 (± 5.21)***
Shadow	**28.81 (± 7.28)**	**17.58 (± 5.46)**	25 (± 8.09)	15.04 (± 5.77)	*27.22 (± 7.32)*	*16.4 (± 5.41)*
Kappa	90.51 (± 2.12)		63.74 (± 6.79)		***90.62 (± 2.47)***	
F1	95.27 (± 0.98)		**96.47 (± 4.88)**		*95.19 (± 1.33)*	
OA	93.49 (± 1.51)		72.65 (± 5.63)		***93.57 (± 1.74)***	
AA	64.45 (± 1.74)		61.2 (± 3.25)		***64.62 (± 1.72)***	
Invariance to object shape; number of samples per class: 40 ; unlabeled samples : 20						
Built-up Area	93.39 (± 1.89)	98.47 (± 1.65)	93.64 (± 1.37)	**97.81 (± 1.37)**	***94.03 (± 2.52)***	*97.48 (± 1.78)*
Vegetation	97.34 (± 0.9)	99.39 (± 0.41)	**97.71 (± 0.67)**	99.63 (± 0.28)	*97.48 (± 1.25)*	***99.54 (± 0.37)***
Bare Soil	**96.74 (± 1.36)**	95.82 (± 2.47)	95.92 (± 1.2)	95.21 (± 1.61)	*96.22 (± 1.83)*	***97.03 (± 1.11)***
Fence	19.84 (± 5.39)	11.56 (± 3.68)	22.34 (± 6.77)	13.21 (± 4.77)	***23.48 (± 7.02)***	***13.95 (± 4.86)***
Shadow	**28.88 (± 9.52)**	**17.83 (± 7.25)**	27.73 (± 6.72)	16.97 (± 4.9)	*26.45 (± 7.73)*	*15.86 (± 5.44)*
Kappa	**90.09 (± 2.11)**		63.74 (± 6.79)		*89.75 (± 2.97)*	
F1	**94.91 (± 1.17)**		96.47 (± 4.88)		*94.47 (± 2.03)*	
OA	**93.2 (± 1.49)**		72.65 (± 5.63)		*92.99 (± 2.04)*	
AA	**64.13 (± 1.46)**		61.2 (± 3.25)		*64.09 (± 1.49)*	

Table 7: Hagadera - Multi-class classification setting- Classification accuracies and other configurations of benchmark semi-supervised method.

Class	VSVM		VSVM-SL		VSVM-SL-Un		VSVM-SL-Vun	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc
Invariance to object scale; number of samples used for model learning and selection per class: 40 ; unlabeled samples : 20								
Built-up Area	**94.25 (± 1.18)**	98.64 (± 0.75)	93.74 (± 1.77)	98.27 (± 1.15)	93.86 (± 1.56)	**98.66 (± 0.76)**	90.95 (± 6.6)	93.9 (± 10.7)
Vegetation	**97.74 (± 0.64)**	99.44 (± 0.35)	97.58 (± 0.75)	**99.53 (± 0.29)**	97.45 (± 0.78)	99.49 (± 0.34)	97.18 (± 0.94)	99.39 (± 0.67)
Bare Soil	**96.99 (± 0.74)**	**96.18 (± 1.26)**	96.27 (± 1.28)	95.68 (± 1.84)	96.49 (± 1.52)	95.73 (± 1.92)	87.77 (± 16.63)	93.92 (± 6.09)
Fence	**23.19 (± 7.08)**	**13.86 (± 5.04)**	20.44 (± 6.92)	11.99 (± 4.87)	20.84 (± 7.81)	12.44 (± 5.58)	18.95 (± 8.9)	11.25 (± 6.14)
Shadow	30.41 (± 6.2)	18.55 (± 4.6)	**32.59 (± 9.74)**	**20.49 (± 7.94)**	28.92 (± 7.18)	17.54 (± 5.17)	25.62 (± 11.88)	15.75 (± 8.37)
Kappa	**91.57 (± 2.26)**		90.92 (± 2.14)		90.44 (± 3.15)		87.49 (± 7.19)	
F1	**95.69 (± 0.99)**		95.37 (± 1.05)		95.02 (± 1.88)		93.16 (± 4.73)	
OA	93.23 (± 1.59)		93.78 (± 1.51)		**93.45 (± 2.2)**		91.34 (± 5.22)	
AA	**65.5 (± 2.21)**		64.82 (± 1.69)		64.85 (± 2.04)		63.83 (± 2.82)	
Invariance to object shape; number of samples used for model learning and selection per class: 40 ; unlabeled samples : 20								
Built-up Area	94.65 (± 1.46)	**98.54 (± 1.11)**	93.82 (± 1.93)	98.1 (± 1.52)	94.35 (± 1.85)	98.15 (± 1.18)	**95.08 (± 6.4)**	97.86 (± 10.42)
Vegetation	97.72 (± 0.72)	99.46 (± 0.4)	97.69 (± 0.65)	99.32 (± 0.37)	97.42 (± 1.05)	99.51 (± 0.41)	**98.44 (± 0.99)**	**99.63 (± 0.55)**
Bare Soil	**97.43 (± 0.85)**	**96.61 (± 1.86)**	96.78 (± 1.4)	95.99 (± 2.33)	96.54 (± 1.53)	96.09 (± 1.93)	96.64 (± 3.05)	96.53 (± 3.22)
Fence	24.67 (± 6.8)	14.71 (± 4.89)	23.03 (± 7.9)	13.83 (± 6.2)	22.76 (± 6.29)	13.4 (± 4.31)	**26.53 (± 8.49)**	**15.88 (± 6.08)**
Shadow	**35.4 (± 9.53)**	**22.82 (± 7.86)**	33.61 (± 9.75)	21.44 (± 8.17)	33.11 (± 9.44)	20.98 (± 7.44)	31.82 (± 8.91)	19.99 (± 6.03)
Kappa	92.18 (± 1.96)		90.54 (± 3.15)		91 (± 2.36)		**93.10 (± 11.59)**	
F1	95.92 (± 1)		94.93 (± 1.89)		95.23 (± 1.54)		**96.22 (± 9.63)**	
OA	94.66 (± 1.37)		93.53 (± 2.21)		93.85 (± 1.62)		**95.32 (± 7.93)**	
AA	**65.85 (± 1.9)**		64.82 (± 1.76)		65.2 (± 1.72)		65.98 (± 4.34)	

Table 8: Hagadera - Multi-class classification setting- Classification accuracies and other configurations of semi-supervised methods

Figure 42: Visualization of results from single realization for multi-class classification setting for Hagadera

Figure 43: Visualization of results from single realization for multi-class classification setting for Hagadera

Results from the multi-class classification are obtained and shown in Fig 38, 39, 40,41. As an overview in all figures, semi-supervised methods (especially SVM-SL-U and VSVM-SL-U methods) outperformed other methods in few labeled samples in all the variations of scale and shape. In these graphs, it can be observed that after 50 labeled samples, all the methods reach their maximum accuracy and kappa statistics, which is why all the methods hold a similar value. To sum up, the Hagadera data set can be summarized in a way that in settings to a various range of unlabeled samples, semi-supervised models indeed achieve higher accuracies for few labeled samples and demonstrates a clear classification map considering the generated tables for certain labeled and unlabeled samples.

4.4 Case Study of Extension on Active Learning

The last chapter of the results section focuses on an active learning approach by including uncertainty calculations and visualizations of the newly proposed semi-supervised methods. This extension on active learning is based on the theory in Chapter 3.3 and follows Algorithm 6. Therefore, a case study is applied on the Cologne data set with binary classification settings on invariances to scale modification. Additionally, for all the semi-supervised methods 40 labeled samples and 20 unlabeled samples per class are selected. In order to observe uncertainty visualization's impact on relabeling the most uncertain certain amount of samples, semi-supervised methods have been selected. As a result, new classification results and uncertainties are visualized, after normalizing the values once again. All the experiments are shown in Fig. 44, 45, and 46 among their binary classifications and overall accuracy, F1, and kappa statistics.

In Fig. 44, the uncertainty of the initially calculated model is displayed on the top-left corner of the figure. Due to the uncertainty calculation, samples that are closer to the hyperplane get the value closer to 0. The samples which are further away from the hyperplane get the values closer to 1. Overall, the values that are closer to 1 are considered as safely labeled samples. Therefore, they are colored as white in the following figure. The values that are close to 0 are colored as red and considered as most uncertain samples and it is also indicated in the legend. Moreover, the figure shows that areas that are classified as bush/tree with an overall accuracy value of 91.10 have more uncertain values on the map. Especially, the areas on the right and left corner and middle-bottom parts of the map. Meanwhile, the second class ("other") has more confident areas, more specifically the areas that are classified as shadow and roof from the multi-classification settings of the Cologne data set. After determining the wrongly classified areas with the uncertainty visualizations, the most uncertain 100 samples have been selected. These samples have been assigned to their true classes from the reference labels and the overall accuracy, F1, and kappa statistics are calculated once again after re-running the model. Newly calculated statistics indeed showed that accuracy is increased only with relabeling the most uncertain 100 samples and also it can be further noticeable from the newly created uncertainty visualization. Although there is not a significant change between the two uncertainty visualizations due to the small number of newly labeled samples. Still, it can be noticed that on the top-left corner of the new uncertainty visualization, uncertainty has been decreased and those pixels became more confident.

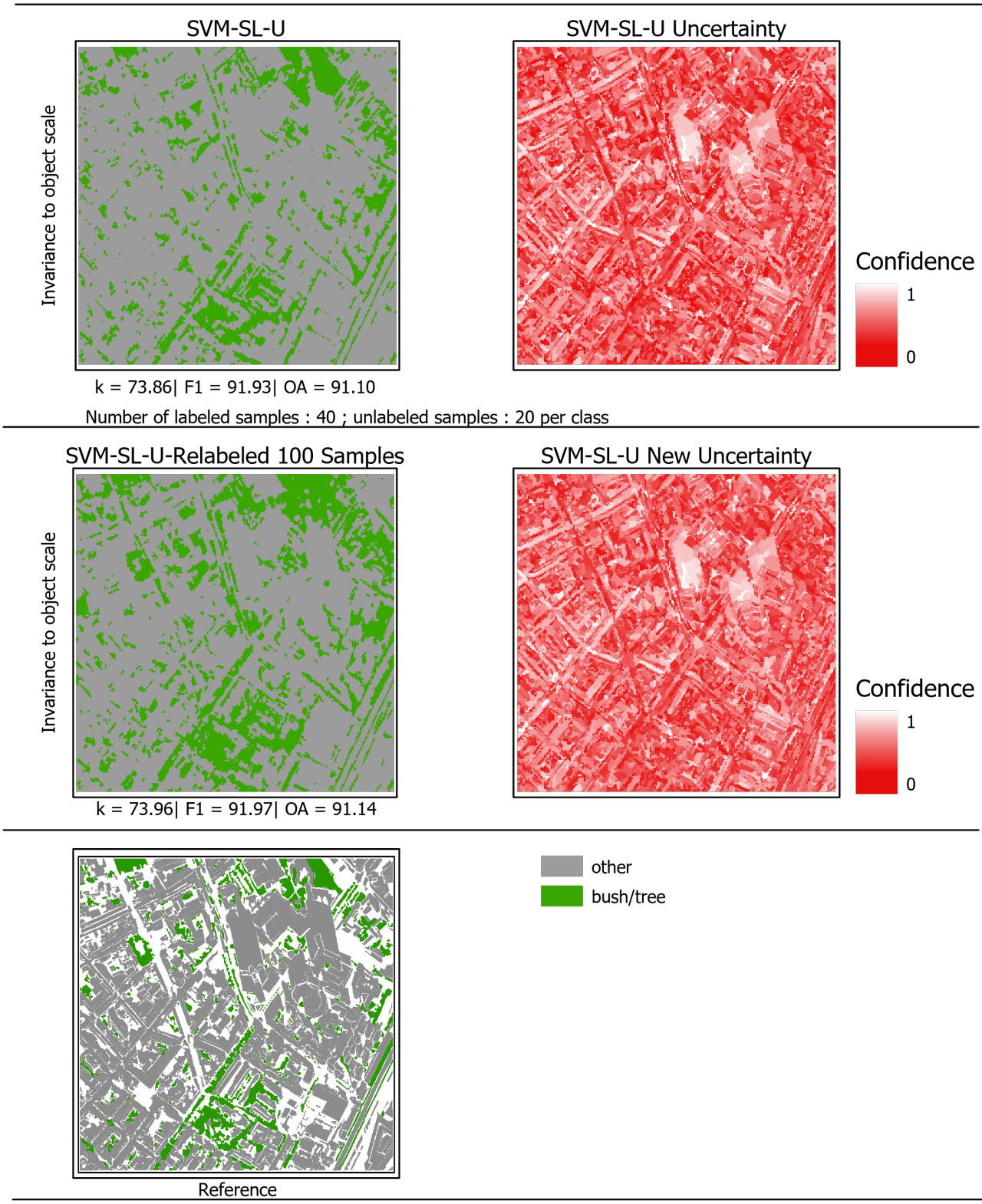


Figure 44: Visualization of uncertainty results on SVM-SL-U approach

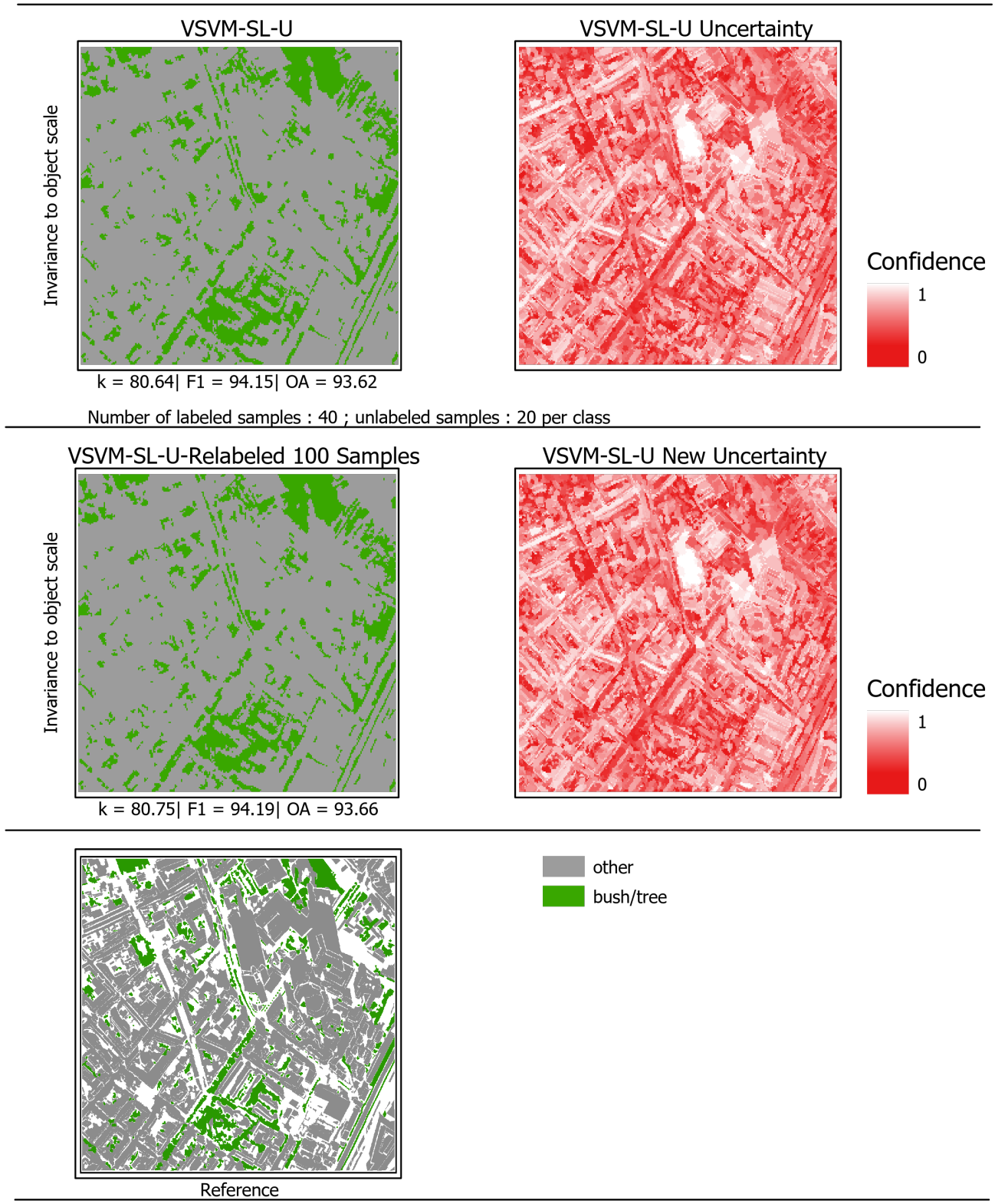


Figure 45: Visualization of uncertainty results on VSVM-SL-U approach

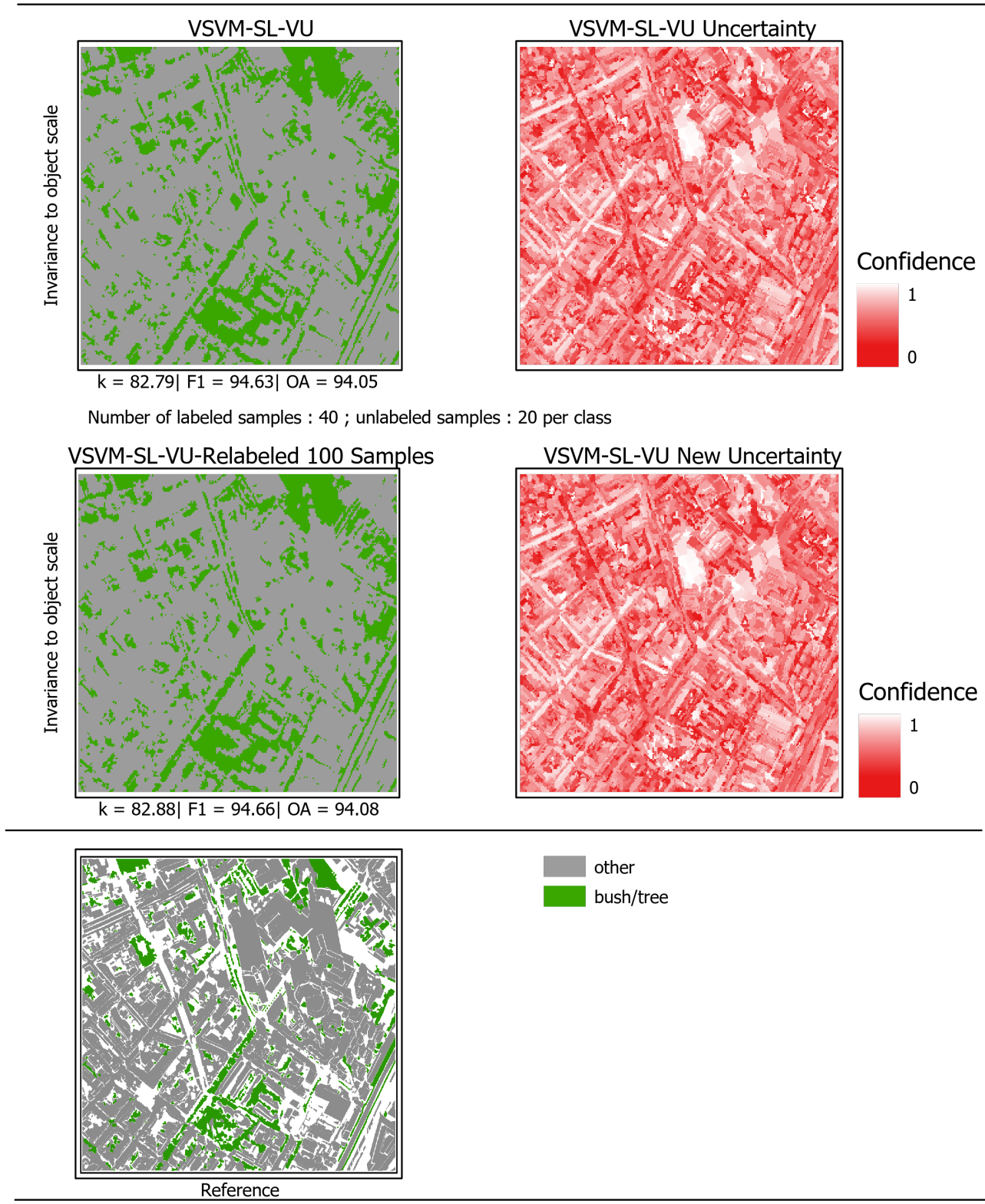


Figure 46: Visualization of uncertainty results on VSVM-SL-VU approach

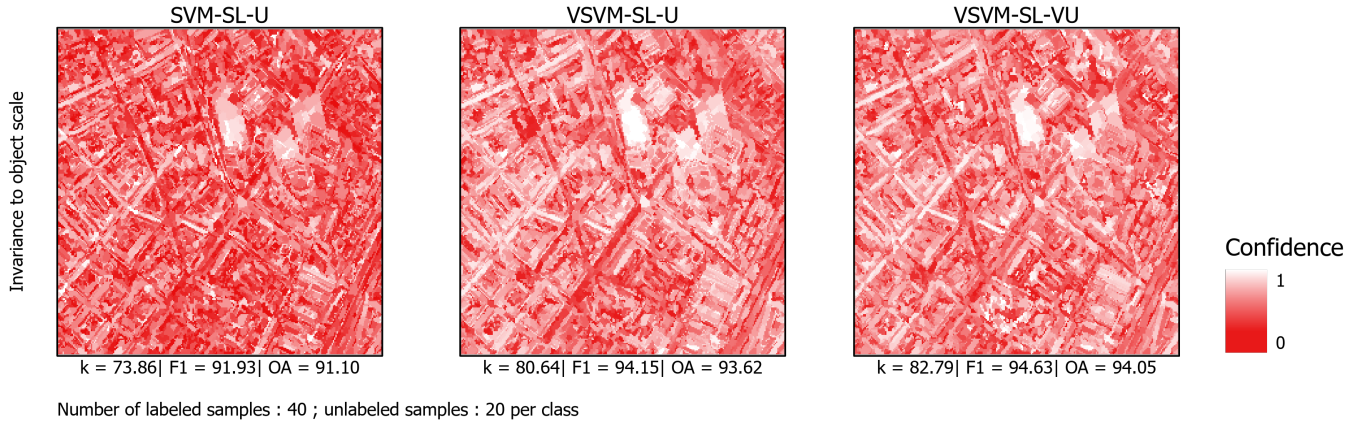


Figure 47: Comparison of uncertainty visualizations for semi-supervised methods on Cologne data set

In Fig. 45. uncertainty visualization is created for the VSVM-SL-U approach and displayed on the right-top part of the figure. Initial classification results showed that overall accuracy of 93.66 has been reached. Therefore, uncertainty visualization had brighter areas compared to the previous figure. In the relabeling process, 100 most uncertain samples have been selected from the bottom part, left and top-right corners of the area, where the class is mainly bush/tree and uncertainty is much higher compared to the middle sections. After relabeling, statistics have been calculated, and 0.0.4 increment is made on overall accuracy, 0.09 in kappa statistics, and 0.04 increment is achieved in F1 scores. New uncertainty values are normalized and visualized in the middle-right side of the figure. It can be seen those areas that are relabeled, had the lowest uncertainty values.

In Fig. 46 similar results with Fig. 45 are obtained because the overall accuracy values were higher than the SVM-SL-U approach. 100 samples have been selected from the areas where the confidence rate is low and after the rerunning, the model overall accuracy of 94.08 is observed with a 0.03 increment. Additionally, change in uncertainty is observed in the areas in the left part, and uncertainties are dropped after the relabeling process.

In the comparison of 3 models (SVM-SL-U, VSVM-SL-U, and VSVM-SL-VU) uncertainty values have been scaled down into the same level for all the methods to make a true comparison in Fig. 47. As expected, models with higher overall accuracies showed generally less uncertain areas, mainly in the middle part of the areas. Between the uncertainty maps of VSVM-SL-U and VSVM-SL-VU difference in uncertainty was only found near the right side of the visualizations.

To sum up, two inferences can be made from the active learning approach with uncertainty visualizations. First of all, uncertainty visualizations indeed increase the accuracies of the models especially when the overall accuracy of the model is already high.

Secondly, uncertainty visualizations give a good hint for the user to see the uncertain areas, and this eventually shows which class that sample belongs to, and it helps the relabeling process.

5 Conclusion

In this research, semi-supervised learning algorithms have been developed based on SVM. These new algorithms benefited from the virtual samples which are used to generate the decisions functions of a classification model invariant. Thereafter, a self-learning strategy is followed to prune non-informative virtual samples and unlabeled virtual samples. The baseline methods aimed to improve the learning abilities with the newly proposed semi-supervised algorithm. For developing the models, very high spatial resolution multispectral imagery, that is acquired by the WorldView-II sensor, is used.

Experimental results are used to solve binary and multi-class classification problems for areas in Cologne and Hagadera. These results showed that semi-supervised methods obtained higher accuracies compared to the baseline methods for both of the data sets, especially in the range for a few labeled samples. Results also indicated that involving unlabeled sample size as a hyperparameter into the model obtains higher accuracies and does not restrict the model to a certain number of unlabeled samples. Moreover, classification maps and tables underline the improvement of those semi-supervised methods regarding spatial consistency. Additionally, the SVM-SL-U as the benchmark method obtained better accuracies and kappa statistics in all data sets compared with the SVM approach. Therefore, it showed that informative unlabeled samples increase the accuracy of the benchmark methods and give robust measurements for all ranges of labeled samples.

Furthermore, an active learning approach has been implemented on semi-supervised methods. It was aimed to enhance the accuracy of the new methods by relabeling the most uncertain samples based on the uncertainty of the models. Experiments were conducted on the Cologne data set for the binary classification settings. Uncertainty visualizations are created to see how uncertain the models are based on their accuracy, F scores, and kappa statistics. Thereafter, uncertainty visualizations are used to relabel a specific amount of most uncertain samples. One of the most important conclusions is that research question one is successfully answered. From the semi-supervised approaches, VSVM-SL-U and VSVM-SL-VU achieved the highest accuracies for the range of a few labeled samples for all the classification settings. The SVM-SL-U approach reached higher accuracies compared to baseline SVM methods, and in kappa statistic graphs, it reached the highest among all the methods.

Research question two is also answered and it showed that uncertainty visualization of the models helps the user to pick and relabel the most uncertain samples. Consequently, the overall accuracy, F1 scores, and kappa statistics of all semi-supervised methods increase after rerunning the models with new labels. Uncertainty change can also be observed between the same models.

In future works, the same methodology can be applied for classifying hyperspectral data within an adequate processing framework. Additionally, an active learning approach with uncertainty visualizations can be integrated into all settings of the data sets. Moreover, this approach can be further adapted to the supervised methods as well. At last, a combination of the semi-supervised methods and active learning approach can be integrated better in a collaborative learning scheme.

6 References

Bastin, L., P. F. Fisher, and J. Wood
2002. Visualizing uncertainty in multi-spectral remotely sensed imagery. *Computers & Geosciences*, 28(3):337–350.

Bennett, K., A. Demiriz, et al.
1999. Semi-supervised support vector machines. *Advances in Neural Information processing systems*, Pp. 368–374.

Blaschke, T.
2010. Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1):2–16.

Bruzzone, L. and L. Carlin
2006. A multilevel context-based system for classification of very high spatial resolution images. *IEEE transactions on Geoscience and Remote Sensing*, 44(9):2587–2600.

Bruzzone, L., M. Chi, and M. Marconcini
2006. A novel transductive svm for semisupervised classification of remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3363–3373.

Burges, C. J.
1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Cortes, C. and V. Vapnik
1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Demir, B., C. Persello, and L. Bruzzone
2010. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 49(3):1014–1031.

Ding, S., Z. Zhu, and X. Zhang
2017. An overview on semi-supervised support vector machine. *Neural Computing and Applications*, 28(5):969–978.

Dópido, I., J. Li, P. R. Marpu, A. Plaza, J. M. B. Dias, and J. A. Benediktsson
2013. Semisupervised self-learning for hyperspectral image classification. *IEEE transactions on geoscience and remote sensing*, 51(7):4032–4044.

Fernández-Delgado, M., E. Cernadas, S. Barro, and D. Amorim
2014. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181.

Foody, G. M.
2009. On training and evaluation of svm for remote sensing applications. In *Kernel methods for remote sensing data analysis*, Pp. 85–109. Wiley.

Foody, G. M. and A. Mathur
2004. A relative evaluation of multiclass image classification by support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(6):1335–1343.

Geiß, C., P. A. Pelizari, L. Blickensdörfer, and H. Taubenböck
2019. Virtual support vector machines with self-learning strategy for classification of multispectral remote sensing imagery. *ISPRS journal of photogrammetry and remote sensing*, 151:42–58.

Geiß, C., P. A. Pelizari, H. Schrade, A. Brenning, and H. Taubenböck
2017. On the effect of spatially non-disjoint training and test samples on estimated model generalization capabilities in supervised classification with spatial features. *IEEE Geoscience and Remote Sensing Letters*, 14(11):2008–2012.

Goodchild, M. F.
2008. Imprecision and spatial uncertainty.

Hughes, G.
1968. On the mean accuracy of statistical pattern recognizers. *IEEE transactions on information theory*, 14(1):55–63.

Izquierdo-Verdiguier, E., V. Laparra, L. Gómez-Chova, and G. Camps-Valls
2012. Encoding invariances in remote sensing image classification with svm. *IEEE Geoscience and Remote Sensing Letters*, 10(5):981–985.

Kinkeldey, C.
2014. A concept for uncertainty-aware analysis of land cover change using geovisual analytics. *ISPRS International Journal of Geo-Information*, 3(3):1122–1138.

Kinkeldey, C., J. Mason, A. Klippel, and J. Schiewe
2014. Evaluation of noise annotation lines: using noise to represent thematic uncertainty in maps. *Cartography and Geographic Information Science*, 41(5):430–439.

Li, Y.-F. and Z.-H. Zhou
2014. Towards making unlabeled data never hurt. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):175–188.

Lu, X., J. Zhang, T. Li, and Y. Zhang
2016. A novel synergetic classification approach for hyperspectral and panchromatic images based on self-learning. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4917–4928.

Lucieer, A.
2004. *Uncertainties in Segmentation and their Visualisation*. PhD thesis, Utrecht University and International Institute for Geo-Information Science ....

Melgani, F. and L. Bruzzone
2004. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on geoscience and remote sensing*, 42(8):1778–1790.

Mountrakis, G., J. Im, and C. Ogole
2011. Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(3):247–259.

Murphy, C. E.
2015. Intellectual highlighting of remote sensing imagery for better image map design. In *Proc. 27th Int. Cartographic Conf.*

Pan, C., J. Li, Y. Wang, and X. Gao
2018. Collaborative learning for hyperspectral image classification. *Neurocomputing*, 275:2512–2524.

Song, X., Z. Duan, and X. Jiang
2012. Comparison of artificial neural networks and support vector machine classifiers for land cover classification in northern china using a spot-5 hrg image. *International Journal of Remote Sensing*, 33:3301–3320.

Tuia, D. and J. Munoz-Mari
2012. Learning user's confidence for active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):872–880.

Tuia, D., F. Ratle, F. Pacifici, M. F. Kanevski, and W. J. Emery
2009. Active learning methods for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2218–2232.

Tuia, D., M. Volpi, L. Copa, M. Kanevski, and J. Munoz-Mari
2011. A survey of active learning algorithms for supervised remote sensing image classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(3):606–617.

Ul Haq, Q. S., L. Tao, F. Sun, and S. Yang
2011. A fast and robust sparse approach for hyperspectral data classification using a few labeled samples. *IEEE Transactions on Geoscience and Remote Sensing*, 50(6):2287–2302.

Volpi, M., D. Tuia, F. Bovolo, M. Kanevski, and L. Bruzzone
2013. Supervised change detection in vhr images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation*, 20:77–85.

Wang, X. and P. M. Pardalos
2014. A survey of support vector machines with uncertainties. *Annals of Data Science*, 1(3-4):293–309.