BEYOND HAND-CRAFTED FEATURES IN REMOTE SENSING

P. Tokarczyk, J. D. Wegner, S. Walk, K. Schindler

Institute of Geodesy and Photogrammetry, ETH Zürich

Commission III/4

KEY WORDS: classification, land cover, feature extraction, pattern recognition

ABSTRACT:

A basic problem of image classification in remote sensing is to select suitable image features. However, modern classifiers such as AdaBoost allow for feature selection driven by the training data. This capability brings up the question whether hand-crafted features are required or whether it would not be enough to extract the same quasi-exhaustive feature set for different classification problems and let the classifier choose a suitable subset for the specific image statistics of the given problem. To be able to efficiently extract a large quasi-exhaustive set of multi-scale texture and intensity features we suggest to approximate standard derivative filters via integral images. We compare our *quasi-exhaustive features* to several standard feature sets on four very high-resolution (VHR) aerial and satellite datasets of urban areas. We show that in combination with a boosting classifier the proposed *quasi-exhaustive features* outperform standard baselines.

1 INTRODUCTION

Automated classification of remotely sensed images is one of the fundamental challenges in remote sensing research. Here, the emphasis is put on urban areas because they are quickly evolving environments and changes are costly to monitor both in terms of monetary resources and time. What makes this task challenging is the complex scene structure where different object categories occur in complicated spatial layouts. Moreover, due to the nadir perspective one can hardly make simplifying assumptions about typical scene layouts such as ”the sky is at the top” or ”trees stand upright”, like commonly done for ground-level computer vision applications. Additionally, fine texture details become visible at the small ground sampling distance (GSD) of VHR sensors, thus increasing the within-class variability, while the between-class variability is relatively low.

Supervised classification involves three major steps: *(i)* feature extraction from raw image observations, *(ii)* training of a classifier using labeled ground truth and *(iii)* classification of test data with the trained classifier. While systematic efforts have gone into the latter two steps by adopting Machine Learning methods, for example, SVMs (Waske and Benediktsson, 2007) and fuzzy approaches (Bovolo et al., 2010) or ensemble methods such as Boosting (Briem et al., 2002) and Random Forests (Pal, 2005; Gislason et al., 2006), less attention has been paid to the design of suitable features for remote sensing. Thus, the choice of features still remains somehow based on educated guessing.

Which combinations of the raw intensity values contain the information to separate the target classes is in general not explicitly known. It is thus still common to directly use the radiometric pixel values of all bands. Often these raw values are augmented with features computed via Principal Component Analysis (PCA) or the Normalized Difference Vegetation Index (NDVI).

For the older task of large-area land cover classification in low- and medium-resolution imagery texture features were not as crucial, because the pixel-footprint was too large to resolve texture patterns (e.g., crop rows in agricultural fields, road markings etc.). Nowadays, airborne and spaceborne sensors are capable of mapping even small objects like cars, narrow roads, and single trees with several pixels, thereby vastly increasing the spectral intraclass variability. On the other hand, fine-grained height information from high-resolution stereo matching is often available as additional source of information.

One straightforward approach for exploiting texture instead of only per-pixel intensities is to use pre-defined filters in multiple scales and directions, as proposed for example by Leung and Malik (2001) and Schmid (2001). These filter banks typically contain variants of multi-scale first and second derivative filters, but the number of filter responses (i.e., feature dimensions) usually remains limited, both to maintain computational efficiency and because traditional classifiers (e.g., Gaussian maximum likelihood) could not handle high-dimensional feature spaces.

Here, we propose to let a discriminative classifier do feature selection directly from a comprehensive set of candidate features, consisting of intensity values and within-channel as well as across-channel differences computed at various scales and orientations. In spite of its high dimension such a candidate set can be extracted efficiently, by approximating averaging and derivative filters with integral images, as proposed by (Bay et al., 2008) for SURF. Our hypothesis is that such a feature set could serve as an almost universal solution for a rather large range of classification tasks. Note that the full high-dimensional feature set only needs to be extracted for the (typically small) training areas, whereas only the small subset selected during training is needed for testing. In order to evaluate the potential of such *quasi-exhaustive features*, we conduct a comparative study: standard feature sets and the proposed *quasi-exhaustive features* are extracted, fed into a boosting classifier and results are evaluated with manually labeled ground truth of four test scenes.

1.1 Related work

While there is a vast body of literature on feature design for computer vision applications, feature design and selection has been less of a topic in the context of VHR optical remote sensing data. In general, it has been acknowledged that one can no longer rely only on single-pixel values, but has to consider a certain local neighborhood (Gamba et al., 2011).

One technique for describing patterns is recording their responses to specific texture filters. Various texture filters have already been tested in remote sensing studies. Shao and Foerstner (1994) evaluate the potential of Gabor filters for aerial imagery, whereas Galun et al. (2003) perform landscape image segmentation with edge-based texture filters. Martin et al. (2004) have used a subset of the root filter sets (RFS) for the segmentation and Lazaridis and Petrou (2006) derive the texture features from the Walsh transform. That smooth filter banks can be approximated by differences of box filters has been exploited for example by Bay et al. (2008), who use box filters to approximate the responses of Hessian filters. They also utilize integral images for rapid computation of the filter responses. Drauschke and Mayer (2010) compare and assess the performance of several of the previously mentioned texture filters using images of building facades. They outline the needs for a more universal approach encompassing desired properties of all tested filter banks, because each single filter bank gives optimal results only for a specific dataset.

Other works propose to extract vast amounts of features and apply either genetic algorithms (van Coillie et al., 2007; Rezaei et al., 2012) or partial least squares for feature dimensionality reduction (Schwartz et al., 2009; Hussain and Triggs, 2010) prior to classifier training. In computer vision the closest related work to ours is probably the integral channel features method (Dollár et al., 2009). For object detection, they randomly sample a large number of box filter responses over the detection window and use AdaBoost to select the most discriminative features, also using integral images. They show improvement over methods that stick to hand-crafted feature layouts.

Deep belief networks (DBN) follow a similar line of thought in that they try to learn (non-linear) feature extractors as part of unsupervised pre-training (Hinton and Salakhutdinov, 2006; Ranzato et al., 2007). First steps have also been made to adapt them to feature learning and patch-based classification of high-resolution remote sensing data by Mnih and Hinton (2010, 2012). However, in a recent evaluation of Tokarczyk et al. (2012) DBNs as feature extractors did not improve the classification results compared to standard linear filter banks for VHR remote sensing data.

An alternative strategy is to model local object patterns via prior distributions with generative Markov Random Fields (MRF) or discriminative Conditional Random Fields (CRF). Usually, such methods are used to encode smoothness priors over local neighborhoods (Schindler, 2012), but some works exist that instead of applying smoothness constraints directly encode texture through the prior term (Gimel'farb, 1996; Zhu et al., 1997). Helmholz et al. (2012) apply the approach of (Gimel'farb, 1996) to aerial and satellite images as one of several feature extractors inside a semi-automatic quality control tool for topographic datasets.

2 METHODS

In line with recent machine learning literature, we pose feature extraction and classification as a joint problem. Instead of pre-selecting certain features that seem appropriate for describing a particular scene and classifying them in a subsequent step, feature selection is completely left to the classifier, such that those features are selected which best solve the classification problem for a given dataset. In the following we first describe hand-crafted features commonly used in remote sensing, which serve as our baseline, before turning to details about the proposed *quasi-exhaustive features*. Thereafter, the boosting algorithm for training (i.e., feature selection and weighting) and testing is explained.

2.1 Baseline features

We start by describing the baselines for our comparison, namely hand-crafted features commonly used for classifying aerial and satellite images. With hand-crafted we mean that a human expert makes an "educated guess" on what kind of features seem appropriate for classification of the data at hand. Note that this procedure runs the risk of losing important information, if an informative feature is not anticipated. We propose to circumvent manual pre-selection by computing a large number of intensity values and intensity differences over a range of scales, both per channel and between channels. The relevant subset is then automatically selected during classifier training.

2.1.1 15×15 pixel neighborhood State-of-the-art airborne data is captured with a GSD ≤ 0.2 m while space borne imagery can be acquired with a resolution of ≤ 0.5 m. In such very high spatial resolution imagery even small objects like single trees consist of several pixels, and on larger objects like building roofs sub-structures, like chimneys, dormers and tile patterns emerge. To cope with the resulting high intra-class variability of the radiometric signature, and to exploit class-specific texture patterns, one should thus consider also the intensities in a pixels' neighborhood. Hence, for each pixel also the intensities of its neighbors within a square window are added to the feature vector. Typical window sizes range from 3×3 to 21×21 depending on image resolution and object size. We have tested various different window sizes and found 15×15 patches to be sufficient, leading to a feature vector with 225 dimensions per channel. Thus we obtain a 675 dimensional feature space for our test images with three channels. Note, since the classifier is free to base its decision on the intensities of the central pixel, the method includes the case of not using any neighborhood. Note also, due to the strongly overlapping content of adjacent windows using the neighborhood can be expected to smooth the classification results.

2.1.2 Augmented 15×15 pixel neighborhood This feature set represents a standard "educated guess" in the optical remote sensing domain. In addition to the local neighborhood of each single pixel, we add the NDVI channel, as well as linear combinations of the raw data found by PCA. Given N input channels, all N principal components are added to the feature vectors, because a-priori dimensionality reduction is not the goal here, while the classification stage performs feature selection anyway. PCA and NDVI are treated like additional channels, thus for each pixel we again put all values inside the 15×15 nieghborhood into the feature vector, thereby adding another 225 dimensions per channel. For input images with three channels plus three PCA channels and one NDVI channel we obtain a 1575-dimensional feature space.

2.1.3 Texture filters Here we use a filter bank wide-spread in semantic segmentation in the computer vision domain. Images are first converted to an opponent Gaussian color model (OGCM) in order to account for intensity changes due to lighting fluctuations or viewpoints (we have also tested several other color spaces, but OGCM yielded most stable responses). Normalized color channels are convolved with a set of filters adopted from (Winn et al., 2005) being a subset of the RFS filters. It contains three Gaussians, four first-order Gaussian derivatives, and four Laplacian-of-Gaussians (LoG). Three Gaussian kernels at scales $\{\sigma, 2\sigma, 4\sigma\}$ are applied. The first-order Gaussian derivatives are computed separately in *x*- and *y*-direction at scales $\{2\sigma, 4\sigma\}$ thus yielding four responses and the four LoGs have scales $\{\sigma, 2\sigma, 4\sigma, 8\sigma\}$. In total, 11 features are computed per channel leading to a 33-dimensional feature space for our test images with three channels. We tested multiple choices of σ and found $\sigma = 0.7$ to deliver best results. Note that by convolution of the channels with such filters each pixel's neigborhood is implicitly considered.

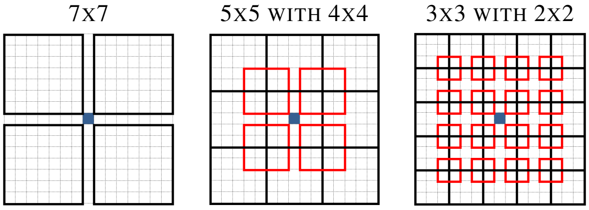


Figure 1: Multi-scale patches: Pixels are displayed grey, 2x2 and 4x4 patches red, 3x3, 5x5, and 7x7 black.

2.2 Quasi-exhaustive features

The main idea of the proposed *quasi-exhaustive features* is to avoid data-specific feature engineering altogether, by offering a redundant, comprehensive feature set. We do not attempt to learn a compact but complicated feature set, as for example deep learning (Hinton and Salakhutdinov, 2006; Ranzato et al., 2007) or generative texture priors (Gimel'farb, 1996; Zhu et al., 1997). Rather we propose to brute-force the feature extraction by computing a large, redundant set of simple features and selecting a small subset of those automatically during training. The data-specific selection allows one to adapt to specific sensor characteristics, lighting conditions and scene content, while the fixed, simple set of feature candidates makes it possible to extract them efficiently. The hope is that in this way one can mitigate the limitations of smaller filter banks, whose performance tends to vary a lot depending on data and scene content (Drauschke and Mayer, 2010). Our *quasi-exhaustive feature* bank includes (see Fig. 1):

- raw pixel intensities within a 15×15 neighborhood in three different scales: *(i)* individual pixel intensities, *(ii)* intensities averaged over 3×3 blocks, and *(iii)* intensities averaged over 5×5 blocks,
- pixel-wise intensity differences within each channel. These are only computed within a 9×9 neighborhood; using the full 15×15 neighborhood would only marginally increase the information relevant for the central pixel, but dramatically increase the amount of features to 25200 per channel, significantly increasing computation time.
- mean intensity differences between patches of size 2×2, 3×3, 4×4, 5×5, 7×7 inside the 15×15 neighborhood, both within and across channels. The patches are chosen to be *(i)* non-overlapping, *(ii)* symmetric with respect to the full window, to approximate gradient/texture filters for the central pixel (Fig. 1).

This feature set ensures that a large range of scales and texture frequencies will be covered and is a reasonable approximation of proper (e.g., Gaussian) derivative filters. Furthermore, the information based on the differences between the spectral channels of an image is exploited. The feature computation is done highly efficient by using integral images (Viola and Jones, 2001). Note that the dimension of the feature space compared to the baseline methods is large, for example, 14553 for our test images with three channels.

2.3 Boosting classifier

As a classification algorithm we choose a variant of discrete AdaBoost (Freund and Schapire, 1997) since it can perform feature selection and thus the computational effort of the testing phase can be reduced. Generally speaking, boosting is a method to improve the accuracy of predictions of learning algorithms. It works by combining many classification rules (called weak learners) that are inaccurate by themselves into an accurate (strong) classifier. As often done in image processing our weak learners are simple decision stumps (i.e., thresholds on single feature dimensions). AdaBoost selects the weak learners (and thus the features) in a greedy fashion.

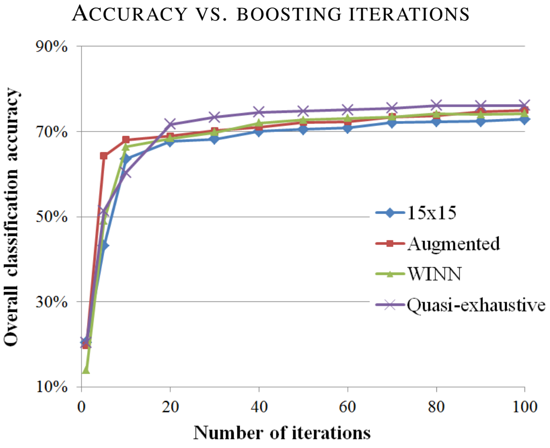


Figure 2: Classification accuracy versus training iterations of the boosting classifier for test scene VAIHINGEN (without nDSM).

As mentioned before, AdaBoost is an ensemble method. It combines many weak classifiers h_k into a strong classifier H by linear combination. The final classifier for a feature vector $\mathbf{x}$ is therefore built as:

$$H(\mathbf{x}) = \sum_k \alpha_k h_k(\mathbf{x}) \tag{1}$$

The sign of H is the predicted class ($+1$ or -1), its magnitude can be used as a confidence measure. The weak learners are added incrementally to the strong classifier. In each iteration AdaBoost tries to find the best weak learner to add to the strong classifier, by looking at the weighted error rate ϵ. The best weak learner is the one that maximizes $|\epsilon - 0.5|$. The weak classifier is assigned a weight α according to the relation:

$$\alpha = \frac{1}{2}\log\left(\frac{1-\epsilon}{\epsilon}\right) \tag{2}$$

This ensures that the classifier gets a higher weight when it operates farther away from chance. Then, the weight w_i of each instance is multiplied by $\exp\left(-\alpha y_i h(\mathbf{x_i})\right)$. This means that samples that were classified correctly (where $y_i = h(\mathbf{x_i})$) decrease in weight, while samples that were classified incorrectly increase in weight. The weights are then normalized to sum to 1 and the boosting procedure is repeated until the desired number of weak classifier is reached, or until no weak classifiers that improve the error rate are found. The re-weighting ensures that the algorithm focuses on hard samples that are misclassified by many weak classifiers.

AdaBoost is a binary classifier, meaning that it learns to distinguish a "positive" (label $y = +1$) from a "negative" (label $y = -1$) class. Since in this paper we have more than two classes (as is often the case in remote sensing), we employ AdaBoost.MH (Schapire and Singer, 1999), which is a multi-class extension to AdaBoost.[1] AdaBoost.MH effectively applies the one-vs-all strategy to create a set of binary problems from the multi-class problem (Friedman et al., 2000). In the case of K classes, $\mathbf{H}$ and $\mathbf{h}$ are vector-valued functions with K components. The label

[1] In the experiments, the software of Benbouzid et al. (2012) was used.

for a sample with class k is also a vector that has $+1$ in its kth component and -1 otherwise. Weak classifiers are of the form

$$\mathbf{h}(\mathbf{x}) = \mathbf{l} \cdot \eta(\mathbf{x}) \ , \tag{3}$$

where $\mathbf{l}$ is also a vector with K components with $+1$ or -1 in each component and η is a binary classifier. The components of $\mathbf{l}$ signal if the binary classifier η has a positive or negative correlation with the respective class label. In the multi-class case, each instance has one weight per classifier component. The weak classifier is selected by weighted averaging over the components and the weights are updated analogous to binary AdaBoost.

3 EXPERIMENTS

We compare classification results for the different feature sets for four classes: buildings, high vegetation (trees, bushes etc.), low vegetation (grassland, small bushes etc.), and streets. All ground truth was annotated manually. The ratio of training (25%) and testing pixels (75%) is kept constant across all four datasets. For training sample selection we simply take a strip on one side of the image, having in mind that each class should be represented with a reasonable amount of pixels.

We use 100 weak learners for training the MultiBoost classifier. The feature selection capability of boosting algorithms allows one to extract only the selected features during testing. This greatly reduces the computation time for testing the classifier.

3.1 Datasets

We evaluate the proposed method on four different VHR datasets (Fig. 3), three aerial photos and one satellite image.

Image Kloten (Switzerland) was acquired with an analogue aerial camera Wild RC30 and scanned. It depicts a part of Kloten airport in the vicinity of Zurich, Switzerland. The image has three spectral bands: red, green, and near infrared. For evaluation we only take a small subset of the scene of 1266×789 pixels at 8 cm GSD. Only a single image is available, thus neither DTM nor DSM can be computed.

Test image Graz (Austria) is a subset of a RGB aerial image of a large block acquired with a Microsoft Vexcel Ultracam D. Its size is 800×800 pixels at a GSD of 25 cm. A digital surface model (DSM) was computed via dense matching. Instead of generating a true orthophoto from the aerial image, the DSM was transformed to the geometry of the aerial image because manually labeled ground truth had been acquired in this geometry. Finally, a normalized DSM (nDSM) was computed via standard filtering techniques. Since only RGB channels exist for Graz a pseudo-NDVI was computed where the green channel replaces the near infrared channel.

Vaihingen (Germany) is a 1000×1000 pixels subset of a true orthophoto mosaic generated from an Intergraph DMC block with 8 cm GSD with red, green, and near infrared channels taken from publicly available benchmark data for urban object classification and 3D building reconstruction (Cramer, 2010; Rottensteiner et al., 2012). A nDSM was obtained by dense matching and subsequent filtering.

The satellite test image is a 1000×1000 pixels part of a stereo-scene of WorldView-2 acquired over Zurich (Switzerland). A pan-sharpened image of 50 cm GSD with three channels red, green, and near infrared was generated. The stereo configuration of the imagery allowed extraction of the DSM from the pan-sharpened channels and the DSM was upsampled to the resolution of the image. It should nonetheless be noted that the DSM quality is much lower than for aerial images (Graz and Vaihingen) because of the lower resolution.

3.2 Results and discussion

We present direct pixel-wise results of the boosting classifier based on the different feature sets (Fig. 4, Tab. 1) without any prior segmentation into superpixels, posterior smoothing via graph cuts or morphological cleaning, to compare only the effect of the features, without potential biases due to pre- or post-processing.

Classification results have been evaluated using two measures: the overall classification accuracy and the kappa index. By measuring the improvement over a chance agreement, as opposed to the one over a 100% wrong result that is measured by the overall accuracy, κ compensates frequency biases[2]. Table 1 summarizes results for all hand-crafted features and the proposed *quasi-exhaustive features*.

In order to quantify how much improvement is due to the nDSM, we compute two separate runs. The first considers all channels except the nDSM for evaluation of all four datasets. Secondly, evaluation is repeated with all channels plus nDSM for the three datasets Vaihingen, Graz, and WorldView-2. Recall that no height information was available for Kloten. In general, datasets augmented with relative height information achieve classification accuracies up to 10 percent points better (Tab. 1).

The proposed *quasi-exhaustive features* outperform almost all baselines in all tests. However, results are close to those of the ”Augmented 15×15 pixels neighborhood” and in the case of the ”GRAZ without nDSM” are worse. A closer inspection of this particular result reveals that it is due to over-fitting causing confusion of street and roofs with the same color as can be seen in the center of the images displayed in the second row of Fig. 4.

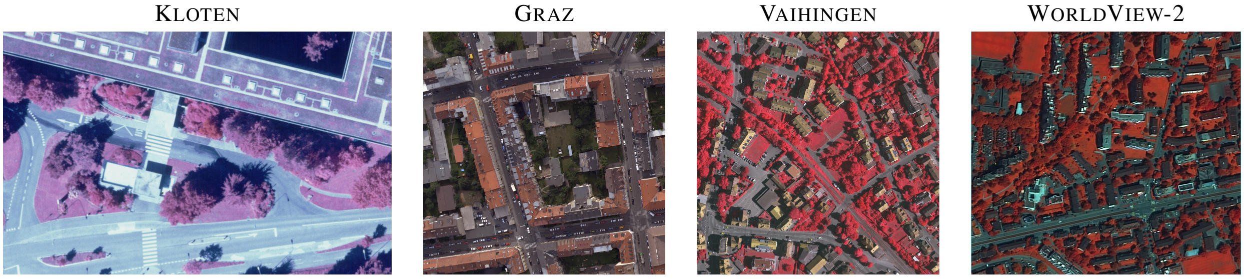
Regarding the WorldView-2 dataset, our method performs on the same level as the augmented features which is most probably due to less distinctive textural patterns in the pan-sharpened image, as well as the poor quality of the DSM.

We plot the classification accuracy versus the number of boosting training iterations in Fig. 2 for test scene Vaihingen (without nDSM). The red curve of the ”Augmented 15×15 pixels neighborhood” shows the steepest accuracy increase for the first five respective ten iterations because it immediately captures the NDVI and, less dominantly, the PCA. For example, for this particular run shown as red curve in Fig. 1 NDVI features ranked 1st, 2nd, 7th, and 9th while PCA features ranked 6th and 10th. *Quasi-exhaustive features* show a less rapid increase, but outperform all baselines after the 20th training iteration.

4 CONCLUSIONS AND OUTLOOK

We have investigated the need for feature engineering when classifying VHR remote sensing images from different sources and showing different scenes. We have demonstrated the power of a simple strategy: rather than trying to determine/guess the best feature set for a given classification problem, supply a *quasi-exhaustive feature set* capturing image intensity and texture at multiple scales over all channels, and let the classifier pick

[2] Formally, $\kappa = \frac{N\sum_i c_{ii} - \sum_i(\sum_j c_{ij} \cdot \sum_j c_{ji})}{N^2 - \sum_i(\sum_j c_{ij} \cdot \sum_j c_{ji})}$, where the c_{ij} are the entries of the confusion matrix and N is the number of pixels. Consider an image with 10% pixels of class A and 90% pixels of class B. A classifier which *always* returns B will have 90% overall accuracy, but κ = 0%.



KLOTEN GRAZ VAIHINGEN WORLDVIEW-2

Figure 3: Original optical images used for evaluation

Dataset	15x15		Augmented		WINN		Quasi-exhaustive	
	Ov	κ	Ov	κ	Ov	κ	Ov	κ
With nDSM								
GRAZ	77.4%	0.66	78.7%	0.68	77.4%	0.65	**80.0%**	**0.69**
VAIHINGEN	82.3%	0.76	83.4%	0.77	82.9%	0.76	**83.6%**	**0.77**
WORLDVIEW-2	77.2%	0.69	**78.7%**	**0.71**	76.1%	0.68	**78.7%**	**0.71**
Without nDSM								
KLOTEN	76.7%	0.67	82.5%	0.75	78.9%	0.70	**82.9%**	**0.76**
GRAZ	72.3%	0.57	**74.6%**	**0.60**	70.2%	0.53	70.9%	0.55
VAIHINGEN	72.9%	0.63	75.0%	0.65	74.2%	0.64	**76.2%**	**0.67**
WORLDVIEW-2	73.5%	0.64	75.0%	0.66	71.0%	0.61	**75.3%**	**0.67**

Table 1: Overall classification accuracies and kappa index for all four datasets and four feature sets

a suitable subset based on the statistics of the training data. To efficiently compute such a large comprehensive texture feature set we propose to resort to integral images, which allow one to evaluate block filters of arbitrary size in constant time, and in this way approximate smoothing and derivative filters.

In future work we plan to investigate in depth which features from our huge candidate set are actually picked by the classifier for different data sets. Moreover, the sizes of the box filters are currently pre-defined and their layout is fixed to a regular grid prior to extracting the *quasi-exhaustive features*. It would be interesting to test whether the performance can be further improved by randomly choosing patterns and box sizes, as is often done for object detection tasks, e.g. (Dollár et al., 2009).

References

Bay, H., Ess, A., Tuytelaars, T. and van Gool, L., 2008. Speeded-up robust features (SURF). CVIU 110(3), pp. 346–359.

Benbouzid, D., Busa-Fekete, R., Casagrande, N., Collin, F.-D. and Kgl, B., 2012. Multiboost: a multi-purpose boosting package. Journal of Machine Learning Research 13, pp. 549–553.

Bovolo, F., Bruzzone, L. and Carlin, L., 2010. A novel technique for subpixel image classification based on support vector machine. IEEE Transactions on Image Processing 19(11), pp. 2983 –2999.

Briem, G., Benediktsson, J. and Sveinsson, J., 2002. Multiple Classifiers Applied to Multisource Remote Sensing Data. IEEE TGRS 40(10), pp. 2291–2299.

Cramer, M., 2010. The dgpf-test on digital airborne camera evaluation overview and test design. Photogrammetrie – Fernerkundung – Geoinformation 2, pp. 73–82.

Dollár, P., Tu, Z., Perona, P. and Belongie, S., 2009. Integral channel features. In: BMVC.

Drauschke, M. and Mayer, H., 2010. Evaluation of texture energies for classification of facade images. In: IAPRS, Vol. 38.

Freund, Y. and Schapire, R., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), pp. 119–139.

Friedman, J., Hastie, T. and Tibshirani, R., 2000. Additive logistic regression: a statistical view of boosting. Annals of Statistics 38(2), pp. 337–374.

Galun, M., Sharon, E., Basri, R. and Brandt, A., 2003. Texture segmentation by multiscale aggregation of filter responses and shape elements. In: CVPR.

Gamba, P., Dell'Acqua, F., Stasolla, M., Trianni, G. and Lisini, G., 2011. Limits and Challenges of Optical Very-High-Spatial-Resolution Satellite Remote Sensing for Urban Applications. John Wiley & Sons, pp. 35–48.

Gimel'farb, G., 1996. Texture Modeling by Multiple Pairwise Pixel Interactions. PAMI 18(11), pp. 1110–1114.

Gislason, P. O., Benediktsson, J. A. and Sveinsson, J. R., 2006. Random forests for land cover classification. Pattern Recognition Letters 27(4), pp. 294–300.

Helmholz, P., Becker, C., Breitkopf, U., Bschenfeld, T., Busch, A., Grünreich, D., Müller, S., Ostermann, J., Pahl, M., Rottensteiner, F., Vogt, K., Ziems, M. and Heipke, C., 2012. Semi-automatic Quality Control of Topographic Data Sets. Photogrammetric Engineering and Remote Sensing 78(9), pp. 959–972.

Hinton, G. and Salakhutdinov, R., 2006. Reducing the Dimensionality of Data with Neural Networks. Science 313(5786), pp. 504–507.

Hussain, S. and Triggs, B., 2010. Feature Sets and Dimensionality Reduction for Visual Object Detection. In: BMVC.

Lazaridis, G. and Petrou, M., 2006. Image registration using the walsh transform. IEEE Transactions on Image Processing 15(8), pp. 2343–2357.

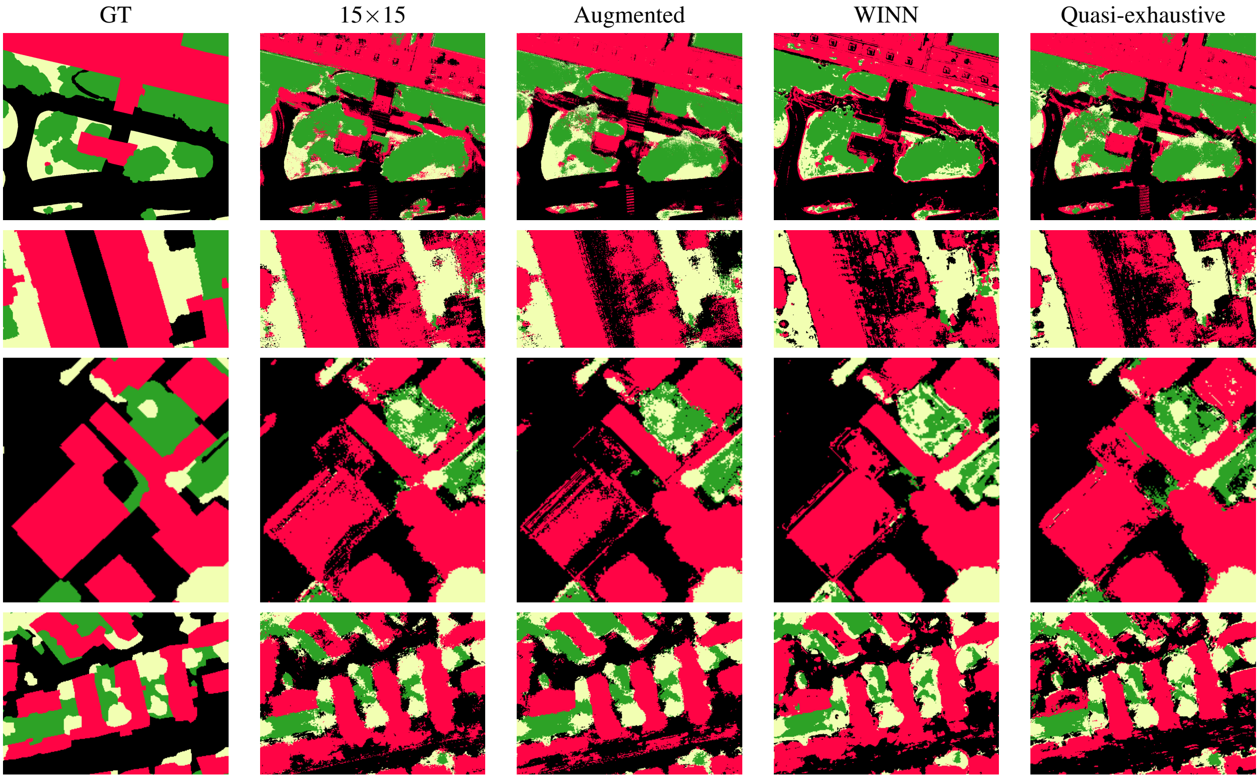


Figure 4: Ground truth (GT) and classification results of four different scenes using boosting (row-wise top to bottom: KLOTEN, GRAZ, VAIHINGEN, WORLDVIEW-2). Streets are displayed black, buildings red, high vegetation green, and low vegetation yellow. KLOTEN and GRAZ show results of the entire test area without the nDSM channel, VAIHINGEN and WORLDVIEW-2 show zoomed details of the results using the nDSM channel.

Leung, T. and Malik, J., 2001. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. IJCV 43(1), pp. 29–44.

Martin, D., Fowlkes, C. and Malik, J., 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE TPAMI 26(5), pp. 530 –549.

Mnih, V. and Hinton, G. E., 2010. Learning to detect roads in high-resolution aerial images. In: ECCV.

Mnih, V. and Hinton, G. E., 2012. Learning to label aerial images from noisy data. In: ICML.

Pal, M., 2005. Random forest classifier for remote sensing classification. International Journal of Remote Sensing 26(1), pp. 217–222.

Ranzato, M., Huang, F., Boureau, Y. and LeCun, Y., 2007. Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition. In: CVPR.

Rezaei, Y., Mobasheri, M., Zoej, M. V. and Schaepman, M., 2012. Endmember Extraction Using a Combination of Orthogonal Projection and Genetic Algorithm. GRSL 9(2), pp. 161–165.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S. and Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. In: ISPRS Annals, Vol. I-3.

Schapire, R. and Singer, Y., 1999. Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3), pp. 297–336.

Schindler, K., 2012. An Overview and Comparison of Smooth Labeling Methods for Land-Cover Classification. IEEE TGRS 50(11), pp. 4534–4545.

Schmid, C., 2001. Constructing Models for Content-based Image Retrieval. In: CVPR.

Schwartz, W., Kembhavi, A., Harwood, D. and Davis, L., 2009. Human detection using partial least squares analysis. In: ICCV.

Shao, J. and Foerstner, W., 1994. Gabor wavelets for texture edge extraction. In: ISPRS Commission III Symposium.

Tokarczyk, P., Montoya, J. and Schindler, K., 2012. An evaluation of feature learning methods for high resolution image classification. ISPRS Annals.

van Coillie, F., Verbeke, L. and Wulf, R. D., 2007. Feature selection by genetic algorithms in object-based classification of IKONOS imagery for forest mapping in Flanders, Belgium. Remote Sensing of Environment 110, pp. 476–487.

Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: CVPR.

Waske, B. and Benediktsson, J., 2007. Fusion of support vector machines for classification of multisensor data. IEEE TGRS 45(12), pp. 3858–3866.

Winn, A., Criminisi, A. and Minka, T., 2005. Object categorization by learned universal visual dictionary. In: ICCV.

Zhu, S., Wu, Y. and Mumford, D., 1997. Minimax Entropy Principle and Its Application to Texture Modeling. Neural Computation 9(8), pp. 1627–1660.