Scalable Machine Learning Approaches for Neighborhood Classification Using Very High Resolution Remote Sensing Imagery*

Manu Sethi
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611
msethi@cise.ufl.edu

Yupeng Yan
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611
yupeng@cise.ufl.edu

Anand Rangarajan
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611
anand@cise.ufl.edu

Ranga Raju Vatsavai
NC State University and Oak Ridge National Laboratory
890 Oval Drive, Campus Box 8206
Raleigh, NC 27695
rrvatsav@ncsu.edu

Sanjay Ranka
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611
ranka@cise.ufl.edu

ABSTRACT

Urban neighborhood classification using very high resolution (VHR) remote sensing imagery is a challenging and *emerging* application. A semi-supervised learning approach for identifying neighborhoods is presented which employs superpixel tessellation representations of VHR imagery. The image representation utilizes homogeneous and irregularly shaped regions termed superpixels and derives novel features based on intensity histograms, geometry, corner and superpixel density and scale of tessellation. The semi-supervised learning approach uses a support vector machine (SVM) to obtain a preliminary classification which is then subsequently refined using graph Laplacian propagation. Several intermediate stages in the pipeline are presented to showcase the important features of this approach. We evaluated this approach on four different geographic settings with varying neighborhood types and compared it with the recent Gaussian Multiple Learning algorithm. This evaluation shows several advantages, including model building, accuracy, and efficiency which makes it a great choice for deployment in large scale applications like global human settlement mapping and population distribution (e.g., LandScan), and change detection.

*This work is partially supported by NSF IIS 1065081

KDD'15, August 11-14, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2788625.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; I.5 [**Pattern Recognition**]: Models—*Statistical*

Keywords

Remote Sensing, Segmentation, Neighborhoods

1. INTRODUCTION

The past twenty years have seen an explosion of interest in remote sensing technologies and machine learning algorithms aimed at representing, categorizing and classifying land cover. With applications ranging across agriculture, space exploration and urban planning, remote sensing is a technology with a huge upside with many more applications yet to be envisaged let alone executed. The various improvements in imaging resolution over the past two decades has led to fine-grained classification with several applications; biomass monitoring, urban settlement mapping, climate change projections, etc.

Remote sensing applications have certain characteristics that take them beyond routine application of machine learning algorithms for classification. First, remote sensing data live on a spatio-temporal grid and this fact is fundamentally at odds with the independent and identically distributed (i.i.d.) sample-based approach adopted in present-day machine learning. For this reason, it is vitally important to begin with an image representation which takes the gridded nature of the data into account. We adopt a segmentation-based representation which leverages the spatial remote sensing grid. Second, the range of queries in remote sensing are more complex. Instead of requiring pixel-level classification, newer applications requires us to classify homogeneous regions into single categories while maintaining clear-cut region boundaries between classes (e.g., urban versus forest). This is a far cry from merely labeling individual samples—the usual scenario in standard machine learning. Third, the

Figure 1: An example of the high resolution aerial images obtained from Google Earth that we use in our experiments for terrain classification. The above image contains more than 1 million pixels and represents an area of about 1 sq km on the ground.

underlying volume (terabytes to petabytes) and velocity (gigabytes to terabytes per day) of data produced by these applications is very large and is responsible for carrying us into the bigdata regime. Effective processing requires low complexity and multi-scale algorithms that can exploit modern architectures with deep memory hierarchies. This is an important challenge for existing machine learning techniques. Finally, expert interaction and the demarcation of training and test set regimes are very different in remote sensing. Experts may be called upon to label whole regions, for example, rather than individual pixels. And, since the entire image is available at the time of training, a semi-supervised approach which does not arbitrarily demarcate training and testing regimes is needed. These considerations shape our work presented below. In a nutshell, we leverage a segmentation-driven image representation and perform semi-supervised label propagation within this representation to achieve image classification in our remote sensing application. An example of the images that we are going to use in this paper can be seen in Figure 1:

A fundamental point of departure taken in the present work is the use of superpixel tessellation representations of remote sensing images. When previous work on remote sensing classification is examined in this light, we invariably find that either (i) a pixel-based classification approach is adopted with no consideration given to local region homogeneity, or (ii) rectangular patches of pixels are used, once again with no consideration given to local homogeneity and natural shape of the patch under consideration. In sharp contrast to these previous approaches, we begin with a superpixel tessellation representation of remote sensing images. Coherent local structures are a characteristic, common to remote sensing imagery, and adequately captured by superpixel tessellations. Superpixels are local homogeneous groupings of pixels and superpixel tessellations ensure that the image domain is covered by the superpixels with no overlap. Since region homogeneity is a function of scale, we employ a pyramid of superpixel tessellations with the coarser scales using local groupings of superpixels from finer scales. The result is a scale space of superpixel tessellations—a segmentation driven image representation, fundamental to our approach and rarely used by competing remote sensing classification approaches at the present time.

We now briefly delve into the specifics of the superpixel tessellation approach deployed here. Essentially, we adapt the popular ultrametric contour map (UCM) approach to image segmentation to obtain superpixel tessellations. We briefly summarize UCM and highlight its use as a feature extractor—a novel aspect we have not seen in other recent work. UCM begins by obtaining scale-space image gradient information at each pixel and for different orientations. Next, a graph is constructed by joining any pair of pixels which exhibit good evidence for a line segment connecting them in the actual image. Taking cues from recent developments in graph partitioning and spectral clustering, the top eigenvectors of the graph Laplacian are computed and rearranged in image space. Then, gradients are computed on the eigenvector images and combined with the original image gradients (from step one) to produce a contour descriptor at each pixel. A oriented watershed algorithm is executed on the gradient image to produce the lowest level superpixel tessellation. These superpixels are combined using grouping heuristics to obtain the final ultrametric contour map (UCM) [2] with superpixel containment across levels. The superpixel tessellation obtained is the image representation used in this work.

Once the hierarchical image representation is in place, we can focus on superpixel classification. As mentioned earlier, we prefer to classify superpixels (at a suitably chosen level) instead of classifying individual pixels of rectangular regions (which lack local homogeneity). A semi-supervised machine learning approach is most suitable in remote sensing and in this context of superpixel tessellation representations. The remote sensing expert is tasked with labeling $O(1)$ superpixels (at a suitably chosen level) with the machine learning algorithm subsequently labeling all the other superpixels. Assume that the expert labels are in place. We now extract features at every superpixel which are consonant with the kind of discrimination required in this application: the separation of remote sensing image data into urban, slum, forest and other labels. To this end, we discovered a novel feature stemming from our chosen UCM-based image representation. The number and type of superpixels (at a suitably chosen level) can function as discriminating features. It turns out that regularly shaped superpixels are better indicative of urban regions and the density of superpixels correlates well with slums. These novel features are combined with more standard color histogram-based features into a superpixel classification algorithm using standard support vector machines (SVMs). Since SVMs do not leverage the underlying spatial grid and indeed have a fundamental limitation as sample-based classifiers, we execute a Laplacian graph-based label propagation algorithm to improve on the result of SVM-based classification.

In this paper, we utilize the above approach for semi-supervised labeling of regions into different land-use land-cover (LULC) classes (urban, slums, forests, sea, sand) in high resolution aerial images. Our approach leverages new features as the traditionally available features such as HOG and SIFT may not be effective in discriminating among different LULC classes pictured in aerial images.

1.1 Application Significance

The applications centered around very high resolution (VHR) imagery have gained huge impetus recently, primarily because images with sub-meter resolution are now available

easily—an outcome of a surge in the launch of satellites by private companies like Digital globe (e.g., WorldView-2 in late 2009). Evidently, such imagery provides extensively new avenues for the automatic classification of both natural regions (forest, sea, different kinds of terrain) and man-made structures (residential and commerical buildings, for instance) worldwide. We are developing various new approaches to efficiently process this imagery for monitoring natural and man-made infrastructures (including neighborhoods, nuclear and other energy infrastructures). At the Oak Ridge National Laboratory (ORNL), the Computational Science and Engineering division produces the LandScan [21] high-resolution population database that is widely used by both government agencies and commercial entities across the world. One of the critical inputs to this model is a thematic layer consisting of different neighborhoods generated from the VHR imagery that spans the globe. Generating a global scale neighborhood map at a sub-meter resolution is a daunting task. For the past several decades, ORNL is developing accurate and computationally efficient methods to process VHR imagery. Likewise, a few international agencies like European Commission (EU) Joint Research Center (JRC) are involved in global scale settlement mapping [26] using VHR imagery.

Despite great efforts by the research community across the globe, neighborhood mapping is a challenging task. First of all, neighborhoods are not well defined, which is reflected in the quote by Galster [12]—"Urban social scientists have treated 'neighborhood' in much the same way as courts of law have treated pornography: a term that is hard to define precisely, but everyone knows it when they see it." There is no consistent nomenclature across the countries regarding neighborhoods and no consistent ground-truth, making it very difficult to build machine learning models for global scale problems. In addition, most of the neighborhoods are made up of complex objects (consisting of different types of objects, not just buildings and roads) which makes it very difficult to obtain ground-truth data from images.

Despite these problems and limitations, mapping neighborhoods, especially informal settlements is crucial in terms of national security and on humanitarian grounds as well. This is because these informal settlements arise unplanned and unauthorized in the most hazardous regions and lack basic services, and therefore, pose several challanges to the nations. Even though numerous studies sponsored by World Bank and United Nations emphasize the significance of poverty maps in designing better policies and interventions, manually mapping the slums of the world is a daunting task. The framework provided in this paper is an advancement in terms of computational efficiency and accuracy in the current technology available to detect new settlements (across the globe), and as well as characterize different neighborhoods which is a key input to other programs like LandScan [21]. In addition, neighborhood maps have several other applications, including but not limited to health [17], energy [10], and crime [16].

The rest of the paper is organized as follows. In the next section we briefly discuss the related work in classifying remote sensing image datasets using semi-supervised approaches. Section 3 describes our approach in detail. Section 4 provides experimental validation and section 5 presents conclusions.

2. PREVIOUS WORK

The major steps involved in remote sensing image classification can be abstracted into: (i) extraction of features from the image, (ii) collection of ground-truth (training/test) data for a few sample locations, (iii) building a classification model (e.g. naïve Bayes, decision trees, MLPs), and (iv) predicting labels for the entire image. Most existing classification approaches work with spectral features (e.g., blue, green, red, thermal infrared) and derived features (e.g., texture, band ratios like Normalized Difference Vegetation Index (NDVI), Histogram of Oriented Gradients (HOG)), extracted at each pixel (spatial location). These classification approaches are called pixel-based or single instance learning (SIL) algorithms. A review of these techniques can be found in [29, 14]. Most classification schemes model the correlations in feature space while the spatial locations of those features are often ignored. An improvement over per-pixel classification schemes is to incorporate spatial locations such as MRF [23]. This combination of the two where spatial correlations and feature correlations are modeled simultaneously leads to what is known as spatial classification schemes. This results into much smoother class distributions in the final classified image. However, it should be noted that spatial classification methods are also essentially single instance learners. One way to overcome the single instance limitation is to look at additional features beyond spectral features. For example, features that exploit spatial contextual information have proved quite useful in classifying very high-resolution images. Recent studies [29, 27, 14] show the improved performance of SIL methods when the spectral features are combined with a broad set of extended features such as morphological, texture, and edge density. Although these studies showed that the extended features which exploit spatial contextual information resulted in improved SIL accuracy, the underlying image complexity and interpixel relationships are still not fully exploited.

Complex object recognition requires investigation of spatial regions or image patches. Object based classification schemes [20, 4] exploit the spatial and spectral features in order to group the pixels into coherent regions called objects. One can then use these objects to build a meta classifier on the features (shape, geometry, etc.) that describe the whole object and not just a particular pixel. Another approach has also been to simply aggregate all features for all pixels belonging to a particular object into a single feature vector and then apply any single instance learning algorithm. However, all these approaches lose important structural and spatial properties in the aggregation process. In order to overcome some of the limitations of single instance learning schemes, multiple instance learning (MIL) methods have been developed. The seminal work of Dietterich *et al.* [9], Diverse Density [18], and Citation-KNN [30] are some notable approaches for MIL. In general MIL methods are shown to perform better than single instance learning schemes, and therefore, have seen application in remote sensing image classification as well. For example, in [5], the authors have developed an MIL based binary classification scheme for identifying targets (landmines) in Hyperspectral (HS) imagery. The high computational cost of Citation-KNN has led to the development of an efficient Gaussian Multiple Instance (GMIL) [28] learning algorithm. Both of these algorithms are shown to perform better than most well-known SIL approaches, however leveraging them for

global scale problems is difficult due to their computational complexity. We believe that our work which utilizes irregular patches or superpixels (which are mainly homogeneous) along with novel and parallelizable machine learning techniques have the potential to address the scale requirements of target applications. In addition our approach eliminates the need for determining an appropriate grid size which impacts the performance—both in terms of computation and accuracy.

Finally, we summarize the evolution—mainly in the past decade—of graph-based semi-supervised learning (SSL) methodologies. Note that there is no general literature of graph-based SSL for gridded data. Early work on SSL focused on optimization [8] and relationships to transductive inference [15, 25] and multi-view learning [19]. Since then, the use of graphs in SSL has become standard [13]. Graph-based SSL methods attempt to assign node labels using a weighted combination of the neighbors. Different methods use different principles to design objective functions for label propagation. For example, a popular approach [31] iterates a function of the graph affinity matrix until convergence and then uses the sign of the function at each node. A different method adapts the Jacobi iteration for linear systems and obtains a somewhat different weighted combination subsequently used for prediction. Other influential methods [3] use regression to determine the weighted combination. First, they compute the graph Laplacian followed by eigenvector computation. Then a regression objective estimates a weighted combination of the principal eigenvectors on the training samples which is utilized for prediction at the unlabeled nodes. Other methods draw upon random walks on graphs to perform label prediction [7].

Our work mostly extends the work of [22] to color images and adds superpixel density at multiple scales of the UCM hierarchy as a new feature. This is an important augmentation to the feature descriptor needed for terrain classification as we will describe in the following sections.

3. APPROACH

In this paper, we describe an efficient image representation and machine learning approach for analyzing large scale remote sensing imagery to support a wide range of earth science applications. The application involves the labeling of remote sensing imagery given expert labels on a small fraction of data. Typically, the classes are (i) forestry, (ii) slums, (iii) urban and (iv) other (not above). The approach has the following steps:

1. Tessellation of the data into superpixels: This step converts the data into irregular (but coherent) patches called superpixels. Superpixels correspond to coherent patches or areas in 2D. These coherent superpixels reduce the data complexity since processing is moved to the superpixel level from the pixel level. Superpixels also have a huge advantage over partitioning the image into regular patches because regular patches ignore the local variability of the underlying data w.r.t. the grid.

2. Generating multi-pixel features for each superpixel: At this step we generate features which are effective in discriminating between terrains present in the spatiotemporal image data. We exploit intensity, geometry and scale of tessellation to arrive at these features.

3. Building a superpixel graph: This step constructs a superpixel graph with edges corresponding to the spatiotemporal (grid). The nodes of the graph are the superpixels with the edges being the connections between them.

4. Label initialization of superpixels: Labels are only available for a small number of pixels. This information is used to derive the labels for a subset of superpixels. A classifier is then built using these features to provide an initial label for all the nodes of the superpixel graph.

5. Label refinement: The rudimentary labels obtained from the previous step are further smoothed using semi-supervised learning techniques.

These steps are described in detail in the next subsections. We will use the image in Figure 1 as a running example. The size of this image is roughly 1 million pixels.

3.1 Superpixel formation

Since the advent of normalized cuts [24] and graph-cuts [11, 6], there has been considerable interest in segmenting an image into sets of superpixels. There are several techniques available in the image processing literature. The ultrametric contour map (UCM) [2] is a popular method which uses local and global cues to produce a hierarchy of tessellations at different scales ranging from fine to coarse. These tessellations respect the containment property, that is every finer scale tessellation is contained within the next higher (or coarser) scale tessellation. We use UCM in two different ways – (i) The first usage is more traditional and direct in the sense that a finer scale tessellation is obtained which is used for classifying (or labeling) each superpixel (as against each individual pixel). (ii) This second usage is more subtle because we obtain the tessellation at a coarser scale and use it as a feature for classifying the superpixels at the finer scale selected in (i). For example, for the target application, the tessellation at a coarser scale mostly picks up prominent boundaries thus increasing the probability of detecting urban regions. Furthermore, the density of superpixels is greater in slum regions than in urban, forested regions making it a suitable feature for slum detection. In Figure 2, we show the superpixels obtained for the image in Figure 1 corresponding to fine and coarse scales.

The steps in superpixel estimation using UCM are as follows: (i) feature extraction using oriented scale-space gradients, (ii) graph construction, (iii) eigenvector computation, (iv) scale-space gradient computation on the eigenvector image, (v) combination of local and global information and (vi) oriented watershed transform to produce non-uniform tessellations. While this sequence is somewhat of a simplification, the major steps have been highlighted. Note that the UCM approach obtains local and global contour information by combining information from the original image and weighted graph eigenvector "images." This perceptual grouping property is mainly responsible for obtaining good superpixel tessellations. We now detail the individual steps in the overall sequence:

Step 1: Scale space feature extraction from brightness, texture and wavelength channels: Oriented Gaussian derivative filters are executed at multiple scales to obtain

$$I_{\text{local}}(\mathbf{x},\theta) = \sum_{\lambda}\sum_{s(\lambda)}\sum_{i(\lambda)} w_{i(\lambda),s(\lambda)} G(\mathbf{x},\theta;\sigma(i(\lambda),s(\lambda))) \quad (1)$$

where $\left\{w_{i(\lambda),s(\lambda)}\right\}$ is a set of weights that depend on the channels and scales. The dependence between the Gaussian filters and the scales can range from the simple to the complex. Here, we have just executed the filters at multiple

(a) Superpixels obtained at the finer scale.

(b) Superpixels obtained at a coarser scale.

Figure 2: Superpixels obtained at different scales for Figure 1. Different scales capture different properties. Coarser scales mostly picks up strong boundaries while the finer scale can be leveraged to obtain the varying density of superpixels contained in coarser superpixels. Coarser superpixels contain higher density of finer superpixels near slum regions but lower density in urban and forest regions—a feature which is exploited to get binary urban and slum features as shown in Figure 3.

scales and orientations. This results in a set of *local* features at different orientations which integrates information from the different color channels.

Step 2: Weighted graph construction: A weighted graph (for the purposes of eigenvector computation) is constructed using the local filter responses above. Following the UCM strategy, pixels within a certain distance of each other are linked by a weighted edge using the relation

$$W(\mathbf{x},\mathbf{y}) = \exp\left\{-\alpha \max_{\mathbf{z}(\mathbf{x},\mathbf{y})} \max_{\theta} I_{\text{local}}(\mathbf{z}(\mathbf{x},\mathbf{y}),\theta)\right\} \qquad (2)$$

where α is a constant and $\mathbf{z}(\mathbf{x},\mathbf{y})$ is any point lying on the line segment connecting $\mathbf{x}$ and $\mathbf{y}$.

Step 3: Eigenvector computation from the weighted graph $W(\mathbf{x},\mathbf{y})$: Following the standard strategy of spectral clustering, the top eigenvectors of the weighted graph are computed. Since these eigenvectors are in location space, the result is a set $\{e_k(\mathbf{x})\}$ (usually rescaled using the eigenvalues of the weighted graph).

Step 4: Spectral information obtained from the top K eigenvectors: Since gradient information computed from the scaled eigenvectors can be expected to contain complementary spectral information [2], a set of gradient operations in different orientations are computed to obtain

$$I_{\text{spectral}}(\mathbf{x},\theta) = \sum_{k} \nabla_{\theta}(e_k(\mathbf{x})). \qquad (3)$$

Step 5: Combination of local and spectral information: We linearly combine the information in (1) and in (3) to obtain the final, global contour probability measure. A free parameter is used for the combination and this needs more careful validation (which we reserve for future work).

Step 6: Oriented watershed transform applied to the global contour measure: Since the global contour probability map may not always be closed and therefore may not divide the image into regions, we require another operation to extract closed contours. We have used the Oriented Watershed Transform (OWT) [2] to construct closed contours. Here, the orientation that maximizes the response of the contour detection approach is used to construct a set of regions and further build a hierarchical region tree from the input contours. Real valued weights are associated with each possible segmentation based on their likelihood to be a true boundary. For a specific threshold, the set of the resulting closed contours can be seen either as a segmentation or as the output of the super-pixellization. Further it can be seen that the uncertainty of a segmentation can be represented—at low thresholds, the image can be oversegmented respecting even very least probable boundaries and as you make the threshold higher only very strong boundaries survive (Fig. 3). This has the benefit of introducing a trade off between the extreme ends of the segmentation.

The resulting tessellation for the image in Figure 1 is given in Figure 2. It shows that areas of significant variation require smaller patch sizes while areas with less variations are captured by large patch sizes. We believe that this variability is one of the major strengths of the proposed approach.

3.2 Superpixel descriptor

Each superpixel at a finer level is described using three kinds of features—intensity histograms, corner density and a binary feature derived from the coarser levels. For the intensity histograms, we quantize the grayscale intensities into 52 bins and obtain a 52 dimensional feature vector for each superpixel.

We use the Harris corner detector to obtain a density measure—the number of corners per unit area for each superpixel. Corners are an important feature for discriminating between regions with buildings (for example, slums and urban area) and regions without (for example, forest and sea). To further distinguish between urban and slum dwellings, we compute the density of superpixels at a suitable level. This feature is based on the (visual) observation that slums have a higher superpixel density relative to forest and urban settlements at the same level of the superpixel pyramid. Another key feature for distinguishing between different kinds of human settlements (slums versus urban) is the presence of stronger boundaries around the superpixels representing the urban regions. Traditional features like dense SIFT, HOG etc. can be used to detect these regions but these suffer from the problem of determining the appropriate scale and orientation. Further these features are also not able to pick up the prominent boundaries as detected by UCM tessellations at coarser scales. As UCM inherently involves a combination of dense features like textons and color histograms and only shows stronger boundaries at coarser

(a) Regions overlayed with the binary urban descriptor. Regions overlayed as white correspond to 1 in our binary urban descriptor signifying a high probability of finding urban regions

(b) Regions overlayed with the binary slum descriptor. Regions overlayed as white correspond to 1 in our binary slum descriptor signifying a high probability of finding slums.

Figure 3: Binary urban and slum descriptors.

scales, it greatly simplifies the task of discriminating the urban regions from the slums. The coarser scale UCM provides a binary feature for each finer superpixel as follows. The coarser scale UCM only keeps prominent boundaries and therefore outputs much larger superpixels. Among them, the superpixels which are smaller than a certain size threshold predominantly belong to urban regions. This is because urban regions are usually found with stronger boundaries and hence are more discriminating. The superpixels which are much larger than the size threshold are more likely to be a merger of several different types of smaller superpixels and are often not very discriminating. For example, these superpixels can be a merger of slums and forests or other similar looking regions which do not have as clearly demarcating boundaries as the urban regions. We label the superpixels below the chosen size threshold with ones and the superpixels above this threshold with zeroes in order to get binary features. These binary features are then percolated down the UCM hierarchy to the finer scale superpixels. All the finer superpixels contained in the larger superpixels get the same label as that of their larger parent superpixel.

Additionally, we also compute the density of finer level superpixels contained within each superpixel at the coarser level. This density is high in the regions corresponding to slums and low in the regions corresponding to urban and forest regions. This can be noticed in Figure 2. We choose a density threshold above which we mark all superpixels at the finer level as ones while the remaining as zeros. This approach further simplifies the task of discriminating slums from the rest of the image. Henceforth, we call these two discriminating features for urban and slum regions as binary urban descriptor and binary slum descriptor respectively. The significance of these two features is visually depicted in Figure 3 by overlaying them on top their corresponding regions of the image in Figure 1.

Further, we use average RGB values corresponding to each superpixel as another 3D feature. All these five different kind of features are then concatenated to form a 58 dimensional feature vector which describes each superpixel of the finer tessellation. Other features like HOG and dense SIFT can also be added to the above framework if needed.

3.3 Initializing the graph labels

Given the large size of the underlying datasets it is impractical to expect that the ground truth is available except at a small number of grid points since data sets scale but experts do not. Thus, practical approaches have to be semi-supervised (as opposed to supervised or unsupervised) with the focus restricted on methods with proven scalability. In this work, we achieve semi-supervised learning through a two stage process of (i) classification using either SVM or kNN followed by (ii) graph Laplacian smoothing. Our classification pipeline is similar to [1]. As mentioned above, the ground truth data is available only for a small number of superpixels as labeled by experts. We use this ground truth to train our classifier. A standard linear SVM was used for training. The model obtained from training is then used to determine preliminary labels for all other superpixels.

3.4 Label refinement

Because of the semi-supervised nature of the problem, the classification obtained from above is rudimentary because it is based on a classifier (SVM in our case) derived from limited ground truth data. This classification can lead to artifacts such that neighboring regions which belong to the same class may get labeled incorrectly. To correct this problem and after being inspired by [1], we apply the Laplacian propagation method as detailed here.

Let f_i denote the feature vector corresponding to the i^{th} superpixel and let X_i be the label that is required to be found from the Laplacian propagation. Let Y_i be the initial label as obtained from the first stage of either SVM or kNN. To perform Laplacian propagation, we construct a graph connecting adjacent superpixels in the spatial domain (and not in the feature domain). The edge weight is given as $W_{ij} = \exp\left(-\frac{\|f_i - f_j\|^2}{2\tau^2}\right)$. Our goal is to minimize the following objective function [1] :

$$C(X) = \sum_{i,j=1}^{N} W_{ij} \left| \frac{X_i}{\sqrt{D_{ii}}} - \frac{X_j}{\sqrt{D_{jj}}} \right|^2 + \sum_{i=1}^{N} \lambda \left| X_i - Y_i \right|^2 \tag{4}$$

where $D_{ii} = \sum_{j=1}^{N} W_{ij}$. The objective in (4) is optimized separately for each category in a one versus rest fashion.

$Y_i = 1$ if the superpixel belongs to the category and 0 otherwise. X_i can take a real value and after minimizing the objective in (4) for each category, we assign each superpixel to the category which corresponds to the maximum value of X_i. The objective function in (4) can be directly minimized by solving a linear system of equations [1]:

$$\left(I - (1 - \frac{\lambda}{1+\lambda})S\right) X = \alpha Y \tag{5}$$

where $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, and $\alpha = \frac{\lambda}{1+\lambda}$.

The image in Figure 4a below shows the preliminary labels obtained after SVM classification and its further refinement by using the graph Laplacian to obtain the final labels in Figure 4b.

4. EXPERIMENTAL RESULTS

To evaluate the accuracy and the efficiency of our approach, we used VHR imagery data from three different geographic settings. The first five images are collected from the Google Earth from Rio, Brazil which is approximately around a million pixels corresponding to 1 square km of area. The Rio images represent two major types of neighborhoods—formal (high-rise apartments and commercial complexes), and informal (favelas). The other two images are from Madison and Milwaukee suburbs from Wisconsin, USA and the last image is from Sterlings Heights city, a suburb of Detroit, Michigan, USA. These images correspond to about 4 square km with the same resolution of 1m as the rio images. The Madison image represents two distinct neighborhoods of commercial complexes and suburban residential communities, while the Milwaukee image consists of downtown and residential neighborhoods. Sterling Heights is the second largest suburb of Metro Detroit and fourth largest city in the state of Michigan. The subregion chosen from Sterling Heights consists of commercial and residential neighborhoods. Apart from the major categories of neighborhoods, these images also contain forests (and isolated trees mixed with houses), grass fields and lawns, undeveloped areas (barren lands and rock outcrops), water bodies, and sandy areas along the shore. As can be seen from these three study regions, they represent diverse set of neighborhoods. The first five Rio images and their classification results are shown in Figure 4 and Figure 5. The Madison, Milwaukee, and Detroit images are shown in Figure 6.

Prior to obtaining superpixels using the method given in [2], we converted the color images to grayscale and performed Gaussian smoothing on the grayscale version of our color images. The support of the Gaussian filter was chosen to be 10 pixels wide and the standard deviation was set to 15. It is important to note that this heavy smoothing of image data as mentioned above was only done for computing UCM [2]. The high resolution detail provided by the images can lead to the generation of an enormous number of superpixels which are not needed for classification purposes. Hence, blurring the images to reduce the number of superpixels is a crucial step. Once UCM was obtained, the original images were used at every other stage of our pipeline, for example in computing features or for classification.

Grayscale intensity histograms, RGB values averaged over each superpixel at multiple levels of the UCM hierarchy, corner density, textons, and descriptors derived from the UCM hierarchy were extracted for each superpixel. As explained in section 3.2, simple descriptors (multilevel superpixel containment density and superpixel area at a coarser UCM level) were extracted from the UCM hierarchy which can be intuitively seen to distinguish urban (residential, commercial) regions, and slums.

Image	labeled data (%)	misclassification error (%)
Rio-1	1.21	8.49
Rio-2	1.49	7.81
Rio-3	1.14	6.58
Rio-4	1.38	11.46
Rio-5	1.30	7.40
Madison	1.01	9.82
Milwaukee	0.52	12.93
Detroit	0.26	16.71

Table 1: Quantitative results for the image in Figure 1 and the 4 different images shown in Figure 5.

Image	% error grids (GMIL)	% error superpixels (proposed)
Rio-1	15.6	9.2
Madison	6.8	3.09
Milwaukee	17.2	6.73
Detroit	15.39	8.73

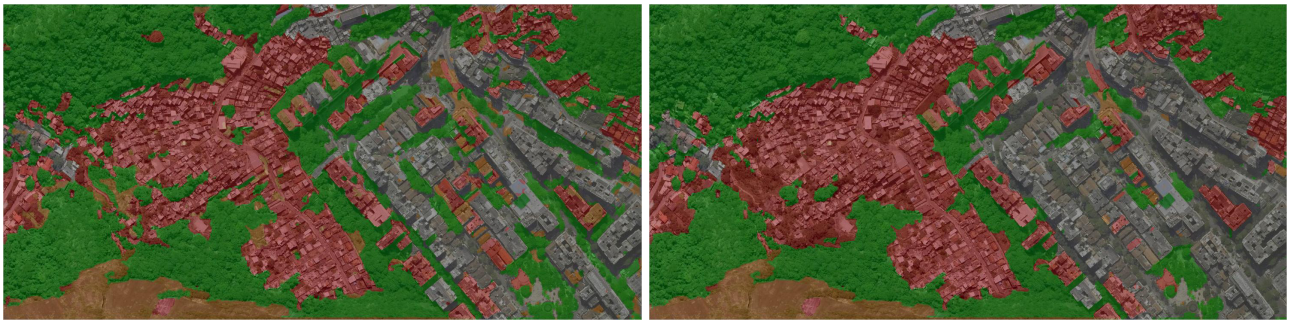
Table 2: Results obtained with GMIL for the first Rio image in Figure 1 and the 3 images belonging to Madison, Milwaukee, and Detroit as shown in Figure 6.

The intensity histograms were obtained by quantizing the intensities into 52 bins. The average RGB values comprised a 3-dimensional vector of average color values for each superpixel. The corner density feature obtained was a scalar which was multiplied by a factor of 100. The coarser scale UCM features and multilevel superpixel containment density features were binary valued. These weighted features were concatenated together to obtain a 58 dimensional feature vector describing each superpixel.

For training the SVM, the ground truth labels were provided to only about 1% (see Table 1) of the superpixels at the finest level. Note that this was even lesser for the Milwaukee (0.52%) and Detroit (0.26%) images. The SVM classifier was used to obtain the preliminary labels for all the other superpixels. This was then given as an initialization to the Laplacian propagation algorithm in order to obtain the final labels. The values of τ and λ were kept fixed to be 2 and 0.125 respectively. For all the images, the misclassification error was around 10% (Table 1). The misclassification error was computed by taking the weighted average of each misclassified superpixel where the weights for each superpixel were the ratio of the area covered by them in the image to the total image area.

The total time required for the overall processing (including UCM) was about 20 minutes on a sequential machine.

Our proposed framework is also compared against the recent Gaussian Multiple Instance Learning (GMIL) [28] algorithm which showed good improvement over standard perpixel based classification schemes. The accuracy estimates for these three images are summarized in Table 2. Results from our approach are comparable to GMIL, however, it should be noted that GMIL is computationally expensive



(a) Preliminary classification obtained after SVM.

(b) Final result obtained after Laplacian smoothing of the labels obtained by the SVM.

Figure 4: Resulting labels for image in Figure 1. The colors red, gray, green, and orange correspond to slum, urban, forest, and barren regions respectively.

Figure 5: Results for 4 more images from Rio, Brazil. The left column shows the actual images. The middle column shows the preliminary results after SVM classification and the last column corresponds to the final refined results after Laplacian smoothing. The colors red, gray, green, blue, yellow, and orange correspond to slum, urban, forest, sea, sand, and farm regions respectively.

and also requires more ground truth training data compared to our technique.

Analysis

Careful analysis of the workflow and final classification results show following key advantages of the proposed method over GMIL [28].

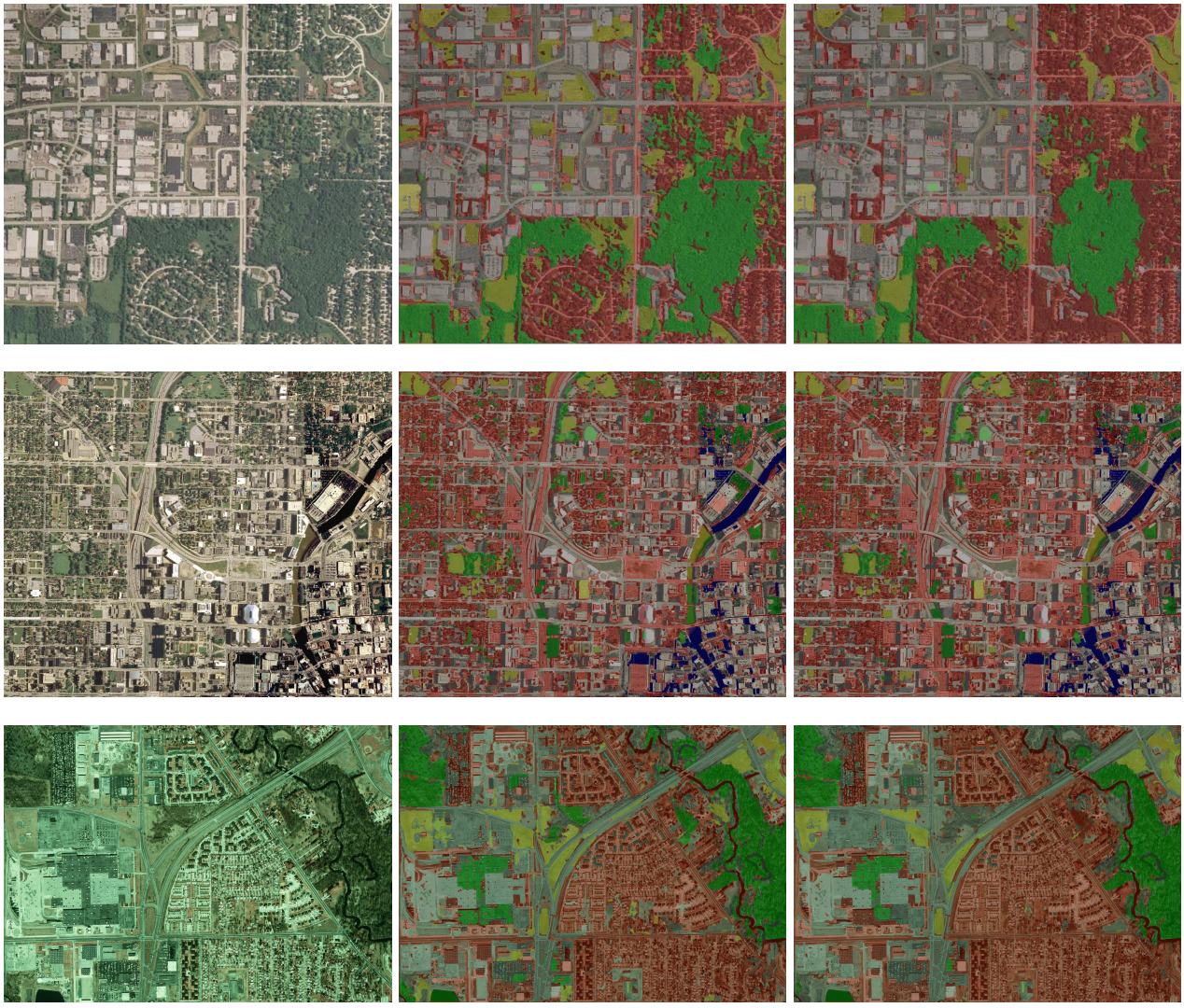


Figure 6: Results for images from Madison (top), Milwaukee (middle), and Detroit (bottom). The left column shows the actual images. The middle column shows the preliminary results after SVM classification and the last column corresponds to the final refined results after Laplacian smoothing. The colors red, gray, dark green, and light green, and blue correspond to residential areas, commercial areas, forests, and grass, and water.

Training One of the most important aspects of supervised learning is obtaining sufficient ground-truth data. GMIL requires a fair amount of ground-truth to capture representative blocks across the image space, as classification is based on patch-matching. On the other hand the proposed approach requires only a fraction of ground-truth points as compared to GMIL, as the ground-truth is for the superpixel. This is an important and notable advantage of the proposed method for global scale applications (e.g., ORNL/LandScan).

Segment vs. Block The basic unit of learning in GMIL is the block. One of the challenges in getting good results from GMIL is determining the appropriate block size. Though trade-offs are known for small and large blocks, determining the appropriate block size requires multiple experiments which is a bottleneck for global scale projects. On the other hand, the proposed method generates superpixel based segmentation, which is far more easier to control than GMIL block size.

Computational Cost GMIL requires each image block (million blocks per sq. km. image at 1m pixel resolution and 10 pixel block size) to be compared against all training blocks (typically on the order of 1000's). This operation is costly as similarity is determined by the Kullback-Leibler (KL) divergence which requires inversion of the covariance matrix. On the other hand, label propagation is computationally much simpler, and furthermore there are far more blocks to classify than the superpixels.

Accuracy Accuracy of both methods are comparable (see Table 2). A visual inspection however shows the proposed method produces a better classification map as

the superpixels confirms to natural boundaries of the objects, whereas GMIL produces jagged boundaries (due to arbitrary sized blocks).

Based on these advantages, we are confident that the proposed methods will become operational in global scale applications like LandScan.

5. CONCLUSIONS

We have developed a novel and scalable machine learning framework for classifying neighborhoods in VHR images. Accurate identification of neighborhoods is critical for many applications, including global scale high-resolution population databases (e.g., ORNL/LandScan), national security, human health, and energy. To meet these challenges, we also developed features which can effectively discriminate between different neighborhoods. This approach combines a superpixel image tessellation representation with semi-supervised label propagation (SVM followed by graph Laplacian-based propagation). The superpixel representation naturally coheres with local boundary information and is a major reason for obtaining good classification. Experimental evaluation on three different geographic settings showed good classification performance. The approach taken in this work is extensible to temporal data as well and will be the focus of our future work. In addition, we plan to focus on improvements to superpixel tessellation-based image representations along the lines of improved computational efficiency and contextual level selection.

6. REFERENCES

[1] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 811–818. IEEE, 2013.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[3] M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56(1-3):209–239, 2004.

[4] T. Blaschke, S. Lang, and G. Hay. *Object-Based Image Analysis*. Springer, 2008.

[5] J. Bolton and P. Gader. Application of multiple-instance learning for hyperspectral image analysis. *Geoscience and Remote Sensing Letters, IEEE*, 8(5):889 –893, sept. 2011.

[6] Y. Boykov and G. Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

[7] O. Chappelle, B. Schölkopf, and A. Zien, editors. *Semi-supervised learning*. Adaptive Computation and Machine Learning. The MIT Press, 2010.

[8] A. Demiriz and K. Bennett. Optimization approaches to semisupervised learning. In *Complementarity: Applications, Algorithms and Extensions*. Springer, 2001.

[9] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

[10] R. Ewing and F. Rong. The impact of urban form on u.s. residential energy use. *Housing Policy Debate*, 19(1):1–30, 2008.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[12] G. Galster. On the nature of neighbourhood. *Urban Studies*, 38(12):2111–2124, 2001.

[13] A. B. Goldberg, X. Zhu, and S. J. Wright. Dissimilarity in graph-based semi-supervised classification. In *AISTATS*, pages 155–162, 2007.

[14] J. Graesser, A. Cheriyadat, R. Vatsavai, V. Chandola, J. Long, and E. Bright. Image based characterization of formal and informal neighborhoods in an urban landscape. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(4):1164 –1176, Aug. 2012.

[15] T. Joachims. Transductive learning via spectral graph partitioning. In *Intl. Conf. Machine Learning*, pages 290–297, 2003.

[16] L. J. Krivo and R. D. Peterson. Extremely disadvantaged neighborhoods and urban crime. *Social Forces*, 75(2):619–648, 1996.

[17] J. Ludwig, L. Sanbonmatsu, L. Gennetian, E. Adam, G. J. Duncan, L. F. Katz, R. C. Kessler, J. R. Kling, S. T. Lindau, R. C. Whitaker, and T. W. McDade. Neighborhoods, obesity, and diabetes: A randomized social experiment. *New England Journal of Medicine*, 365(16):1509–1519, 2011.

[18] O. Maron and T. Lozano-Perez. A framework for multiple-instance learning. In *Advances In Neural Information Processing Systems*, pages 570–576. MIT Press, 1998.

[19] I. Muslea, S. Minton, and C. A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *Intl. Conf. Machine Learning*, pages 435–442, 2002.

[20] S. Nussbaum and G. Menz. *Object-Based Image Analysis and Treaty Verification*. Springer, 2008.

[21] ORNL. Landscan: High-resolution population database. http://web.ornl.gov/sci/landscan/.

[22] M. Sethi, Y. Yan, A. Rangarajan, R. R. Vatsavai, and S. Ranka. An efficient computational framework for labeling large scale spatiotemporal remote sensing datasets. In *Contemporary Computing (IC3), 2014 Seventh International Conference on*, pages 635–640, Aug 2014.

[23] S. Shekhar, P. Schrater, R. Vatsavai, W. Wu, and S. Chawla. Spatial contextual classification and prediction models for mining geospatial data. *IEEE Transaction on Multimedia*, 4(2):174–188, 2002.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[25] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *ICML*, pages 824–831, 2005.

[26] P. M. Thomas Kemper, Nale Mudau and M. Pesaresi. Towards a country-wide mapping & monitoring of formal and informal settlements in south africa. Science and Policy Report JRC92657, Joint Research Centre, European Commission, 2015.

[27] R. R. Vatsavai. High-resolution urban image classification using extended features. In *ICDM Workshops*, pages 869–876, 2011.

[28] R. R. Vatsavai. Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1419–1426. ACM, 2013.

[29] R. R. Vatsavai, E. A. Bright, V. Chandola, B. L. Bhaduri, A. Cheriyadat, and J. Graesser. Machine learning approaches for high-resolution urban land cover classification: a comparative study. In *COM.Geo*, page 11, 2011.

[30] J. Wang. Solving the multiple-instance problem: A lazy learning approach. In *In Proc. 17th International Conf. on Machine Learning*, pages 1119–1125. Morgan Kaufmann, 2000.

[31] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML*, pages 912–919, 2003.