December 2014

Satellite Scene Image Classification for Adaptive Target Tracking

Daniel Pasqual Banco
Worcester Polytechnic Institute

Satellite Scene Image Classification for Adaptive Target Tracking

A Major Qualifying Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

by

Daniel Banco, PH/ECE

December 23, 2014

Approved By:

Professor Edward A. Clancy, ECE Advisor,WPI

Professor Germano S. Iannacchione, PH Advisor, WPI

Abstract

This project proposes a basis for a flexible processing and tracking scheme based on image classification. The goal of this project was to test image features for classifying satellite scene images consisting of various background scenarios. Classification was completed using linear discriminant analysis and a support vector machine. Images were classified using the intensity mean, the intensity standard deviation, the intensity coefficient of variation, and the P-cell average intensity standard deviation, and the bag of keypoints descriptor.

Each of these image features were shown to be useful in classification. Increasing the number of image classes increased the number of features needed to achieve linear class separability. For visible band images, LDA can be used to classify scene images based on the intensity mean and 64-cell average intensity standard deviation with an accuracy of about 0.95 with 2 classes, 0.92 with 4 classes, and 0.75 with 9 classes. A support vector machine used the bag of keypoints descriptor to reach a classification accuracy of 1 with 2 classes, 0.92 with 4 classes and 0.83 with 9 classes. The bag of keypoints descriptor did not render significantly improved results for this dataset.

This methodology can be applied for imagery of any waveband. No parameters were chosen directly based upon the kind of images being classified. Target trackers modified to leverage this classification scheme may be allowed to track more successfully in a variety of background scenarios.

Contents

List of Figures

List of Tables

1 Introduction

Target tracking in video is impacted by the contents of the background. The background of video can contain clutter. Clutter interferes with target tracking by impeding the tracker's ability to discern the target from the background or by presenting false targets. If the target tracking situation is well-defined, that is, characteristics of the background clutter are known, then the tracker can take advantage of specialized processing which eliminates the impact of the background clutter. If the background clutter is known to vary, then a more complicated processing scheme may be required. The processing scheme would need to make a decision based on the contents of a single frame (an image). The contents of an image can be quantified.

In image processing, it is common to use simple statistical descriptions of images. The probability density function of the brightness, the average brightness, the standard deviation of the brightness, the coefficient of variation, the mode, and the signal-to-noise ratio all are examples of ways to quantify the contents of an image (Young, Gerbrands, & van Viliet, 2007). In computer vision, it is common to characterize the contents of an image by identifying localized features such as edges, corners, or other interest points (Szeliski, 2011). The scale-invariant feature transform (SIFT), for example, is a more complex algorithm which can be used to extract keypoints from an image and describe the pixels at keypoints in terms of a SIFT descriptor (Lindeberg, 2012). In target tracking, it is often of interest to characterize the clutter content of an image; however, no standardized clutter metric exists. Proposed clutter metrics are often based on the fundamental image properties described above. The clutter content of an image is often defined relative to the ability of an algorithm or human to identify a target in an image. A few specific examples will be discussed. Once the properties of an image have been quantified, some sort of algorithm is needed to make a decision based on the extracted information.

Classification algorithms are widely used to automate decision-making in a variety of applications. Classification can be viewed as a supervised machine learning problem. Some popular classification approaches are discriminant analysis, neural networks, support vector machines, decision trees, and boosting. Each of these approaches is discussed from an approachable, algorithmic perspective in Marsland (2009). The output of the classification algorithm would determine the processing/tracking steps to follow.

This project proposes a basis for a flexible processing and tracking scheme. A classification algorithm makes a decision based on extracted image features. The goal of this project is to test image features for classifying images of a dataset consisting of various background clutter scenarios.

2 Background

In tracking applications, one seeks to detect one or more targets amongst clutter in a scene. Clutter present in the frame affects the ability to track targets in video. As a result, the following section will define clutter and discuss metrics used to quantify clutter in target tracking literature. Examples of the use of classifiers in target tracking literature are also discussed briefly. This section describes the two classification approaches used in this project: linear discriminant analysis and a support vector machine. The bag of keypoints descriptor is also described.

2.1 Clutter Measures in Target Tracking

The term "clutter" has different meanings in different fields, so it is important to establish a definition. In this application, the term "clutter" refers to complexity in a scene which causes false targets and impedes the detection of true targets (Sutherland, Montoya, & Thompson, 2002). Generally, features within a frame other than the target are considered to be clutter. Clutter typically appears in the background of the image with respect to the target. Occlusion of target occurs if clutter appears in the foreground with respect to the target. Other factors create noise in the image. Different sensors are subject to different kinds of noise. Noise artifacts can be caused by aliasing or quantization. Thermal and electronic noise can also contribute. Noise present in a digital image tends to follow a statistical distribution (Rahman & Jobson, 2003). Both clutter and noise interfere with extraction of the desired signal. Image resolution is also a factor (Reynolds, 1990). Many methods for quantifying clutter exist.

The paper (Meitzler, Gerhart, & Singh, 1998) mentions some existing clutter measures used in the infrared community. The first two mentioned are the radiance mean and radiance standard deviation. These are computed in exactly the same way as the average brightness and standard deviation brightness mentioned earlier. The measure proposed by (Schmieder &

Weathersby, 1983) is described as the most commonly used clutter measure. This measure is computed by averaging the variance of contiguous square cells over the whole scene (Meitzler et al., 1998). An appropriate square cell size was selected to be twice the length of the largest target dimension. This measure defines clutter relative to a known target of interest. This metric is used in (Sutherland et al., 2002) to study atmospheric effects on visible and infrared scene clutter characterization.

In more recent publications, clutter metrics are categorized as either global or local. The metrics mentioned above are all examples of global clutter metrics because they use information from all parts of the image equally. Local clutter metrics make a distinction between the target and the background (Salem, Halford, Moyer, & Gundy, 2009). Clutter metrics tend to be based on either mathematics or the human visual system (Salem et al., 2009). In order to determine the clutter level of images in a cognitive sense, one would test the ability of humans to identify targets in cluttered images. One recent 2009 publication proposes a clutter metric, the rotational clutter metric, that adapts to the definition of clutter as indicated by an expert. The expert labels features extracted from the Laplacian pyramid decomposition of clutter images. The rotational clutter metric generates new features based on the assigned labels in order to best separate input images into clutter levels. In this approach, information regarding what defines clutter and relationships between clutter and target size are learned (Salem et al., 2009). In this paper, a classification scheme was used to evaluate the usefulness of the rotational clutter metric. A set of N features was used to describe a set of N images in order to classify the images in a Laplacian pyramid decomposition feature space. The classification scheme used Bayesian decision theory to establish the class boundary separating low and medium clutter in the feature space (Salem et al., 2009). The paper demonstrates that a feature-based approach performed well in classifying clutter as either low or medium. The experiments conducted used 224 images labeled by an expert observer; 90% of the data were used for training and 10% was used for testing. Classifying the entire data set 10 times achieved an 89% classification rate for the training data; 87% for the testing data. The rotational clutter metric could prove to be worth investigating, but an implementation was not readily accessible. This project employs a similar methodology to (Salem et al., 2009) in order to test many image measures or features.

Four different clutter measures or image features were selected based on the clutter measures already being used in target tracking. The selected image features were the intensity mean, intensity standard deviation, intensity coefficient of variation, and the intensity P-cell average intensity standard deviation based on the measure from (Schmieder & Weathersby, 1983). The first two features appear in both the image processing and target tracking literature. The third feature encompasses the first two features. The fourth feature has been used widely in the target tracking for both visible and infrared band imagery. The simplicity of all four features make them appropriate for this project. These four features are described in Section 3.2. The rotational clutter metric proposed in (Salem et al., 2009) also motivated the selection of a fifth image feature. In computer vision, the bag of keypoints descriptor is used to visually categorize images. The feature is constructed by extracting many local features and characterizing the image based on the number of different types of local features. This image feature provides a higher level of information than the previous four image features. The bag of keypoints descriptor is described in Section 2.3. All five of these image features were be tested using classification algorithms.

2.2 Classification

Classification algorithms sort items into separate classes. Objects are sorted into classes based on extracted mathematical characteristics or features. Datasets of datapoints with known class are used to test classifiers. The data are divided into training data and validation data. Classifiers learn different classes based on the set of examples provided by the training data. The learned information facilitates the classification of validation data. The classification accuracy is evaluated by comparing the assigned classes to the actual classes of the validation data. Classification techniques can generally be regarded as either statistical or structural. Cognitive methods, such as neural networks and genetic algorithms, borrow from both areas. Statistical techniques seek to quantify items with a statistical basis or with quantitative features. Structural techniques seek to take advantage of qualitative features describing structural or syntactic relationships. Of the two, statistical techniques are more popular; however, cognitive methods have gained popularity as well (Szeliski, 2011, p. 5).

Statistical approaches use specified or learned probability distributions of objects pertaining to each class to compute decision boundaries. Classification algorithms have a limited amount of prior information available depending on the application. As a result, a number of statistical approaches exist assuming differing amounts of prior knowledge. Figure 1 shows statistical classification techniques in order of decreasing available information from left to right (Szeliski, 2011, p. 18).

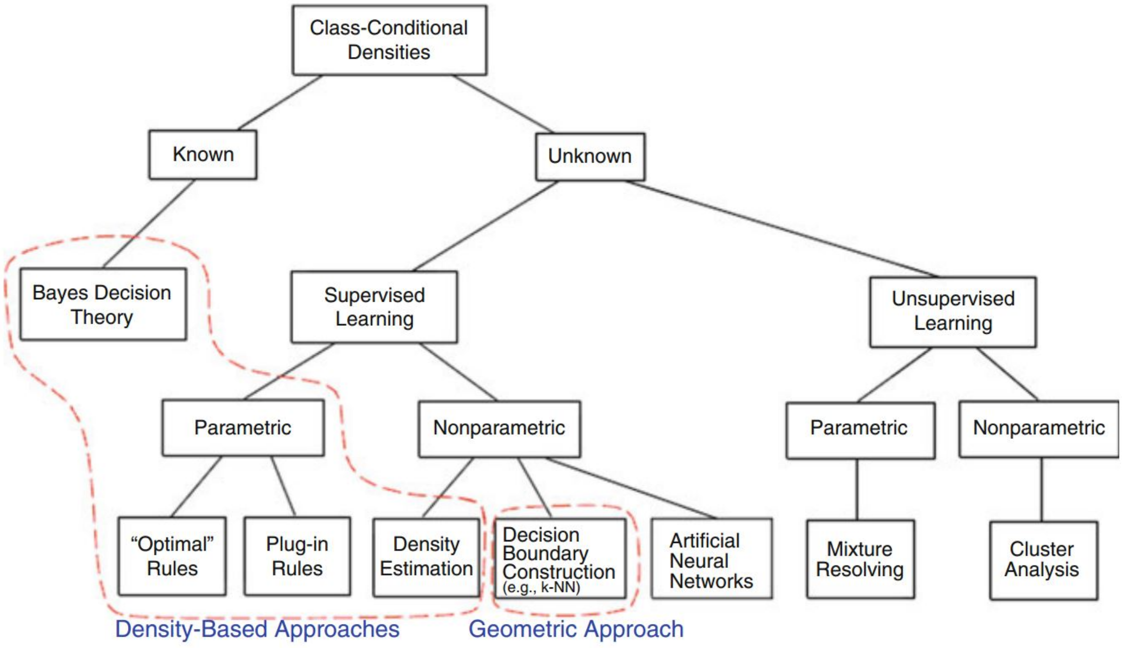


Figure 1: Various approaches in statistical pattern recognition (Szeliski, 2011, p. 19).

The k-nearest neighbor algorithm is one of the simplest examples. The method estimates a probability density function by building histograms that must contain k samples. The class of a sample is determined by the class of its nearest neighbors (Szeliski, 2011, p. 101). Although the method is simple and requires no learning, its efficacy depends largely on the selection of a k value and the estimates are susceptible to local noise.

Applications of classification in target detection and tracking are discussed below. The examples of classifiers used in the papers discussed below show applications of Bayesian decision theory as well as a plug-in rule classifier.

In multiple target tracking, classification algorithms can be used to discern possible tracks from clutter (clutter rejection). One such example used a minimum error Bayesian classifier to reject clutter. The classifier was trained using 230 target and 520 background samples (Yoon, Song, & Kim, 2013). The training data were used to establish likelihood probability based on Gaussian probability and probabilities based on prior information. The probability information established regarding the target class and background class was compared to a potential target to ultimately classify it as target or clutter. Bayes' decision theory was used to classify images as low clutter or medium clutter before proceeding with additional processing (Yoon et al., 2013).

In another relevant target detection problem, clutter classification was used to inform the the decision to use one of two target detection methods (Wang & Zhang, 2011). A rule-based fuzzy system with eight "if-then" rules was used to determine the image to be low clutter or high clutter. The fuzzy degree of each feature determined the state of each feature to be low or high. Each of the eight rules depended on the states of three features. The three features used in this publication were the standard deviation map, segmentation region complexity, and region space distribution complexity (Wang & Zhang, 2011). Each of these features pertain specifically to maritime scenarios because they assume the presence of a sea-sky horizon. This study demonstrates the efficacy of a rule-based approach. The rule-based approach allows the detection system to take into account many metrics.

Since the definition of clutter is ambiguous, especially between infrared and visible band imagery, this study compares the use of statistical image features and one complex image descriptor for classifying scenes. Classification was done using linear discriminant analysis and a support vector machine.

2.2.1 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a method for transforming multivariate observations to univariate observations such that the separation between the univariate observations is maximized (Li, Zhu, & Ogihara, 2006, p. 5). LDA will be described using a two-class example. Some equations are shown such that they can be generalized to more classes. Figure 2 shows a boundary computed using LDA which separates two classes. Suppose that we have 2 sets of 100 training images and 2 features have been extracted from each image. The two features extracted from each image could be those shown in Figure 2.

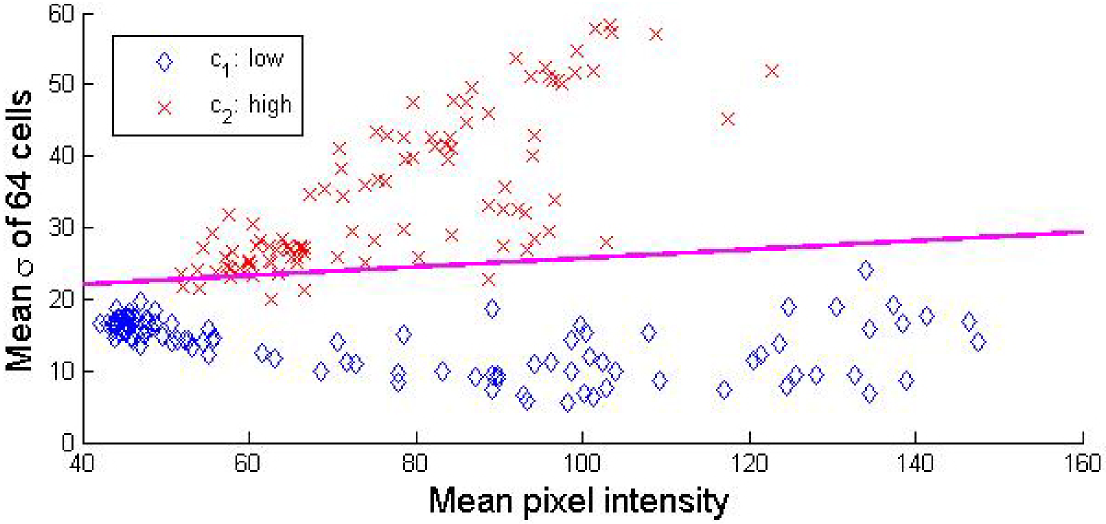


Figure 2: An example of two class LDA classification of images based on the mean standard deviation pixel intensity of P cells and the mean pixel intensity. The violet line represents the class separation boundary.

The two-dimensional observations are represented by the column vector

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}, X_i = [x_{i1}, x_{i2}, ...x_{iN}] \tag{1}$$

where $n = 2$ and $N = 200$ since 2 features are extracted from 200 images. Each image is known to belong to one of two classes, c_1 or c_2. The probability that an image (other than the 200 images) with features $\mathbf{x}$ belongs to class c_1 or c_2 is given by $p(\mathbf{x}|c = c_1)$ and $p(\mathbf{x}|c = c_2)$ These probability density functions are both assumed to be normally distributed with mean and covariance parameters $(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})$ and $(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})$ corresponding to the 100 images of class c_1 and 100 images of class c_2. This mean vector and covariance matrix must be computed for each class of known training images. In this case, each class has a mean vector of two elements and a 2x2 covariance matrix. The probability density function $f(\mathbf{x})$ for the population of images belonging to each class is fit to a multivariate normal (Gaussian) density function

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\mu)'\Sigma^{-1}(\mathbf{x}-\mu)}. \tag{2}$$

The allocation rule used to separate two classes in LDA is based on the posterior probability of $\mathbf{x}$ being classified into a class c_k. The probability density functions for each class given by Equation 2 can be used to represent this posterior probability using Bayes' theorem

$$P_k(\mathbf{x}) = p(c = c_k|\mathbf{x}) = \frac{p_k f_k(\mathbf{x})}{\sum_{l=1}^{K} p_l f_l(\mathbf{x})} \tag{3}$$

where K is the total number of classes (2 in this case) and where p_k is an *a priori* probabilitiy corresponding to each class (Krzanowski, 1988, p. 340). These values are $\frac{1}{2}$ where no *a priori* knowledge exists. A distinct covariance matrix is computed for each class in quadratic discriminant analysis; however, in LDA each class is assumed to have the same covaraince matrix. A single pooled covaraince matrix,

$$\mathbf{S} = \frac{1}{N_1 + N_2 - 2}\left[\sum N_1 i = 1(\mathbf{x}_i^{(1)} - \mu_1)(\mathbf{x}_i^{(1)} - \mu_1)' + (\mathbf{x}_i^{(2)} - \mu_2)(\mathbf{x}_i^{(2)} - \mu_2)'\right], \tag{4}$$

is used for two classes where $N_1 = N_2 = 100$ since we have 100 images of each class. Using this single pooled covariance matrix for both classes allows us to reduce Equation 3 to

$$P_k(\mathbf{x}) = \frac{p_k f_k(\mathbf{x})}{p_1 f_1(\mathbf{x}) + p_2 f_2(\mathbf{x})} = \frac{p_k e^{-\frac{1}{2}(\mathbf{x}-\mu)'\mathbf{S}^{-1}(\mathbf{x}-\mu)}}{p_1 e^{-\frac{1}{2}(\mathbf{x}_i^{(1)}-\mu_1)'\mathbf{S}^{-1}(\mathbf{x}_i^{(1)}-\mu_1)} + p_2 e^{-\frac{1}{2}(\mathbf{x}_i^{(2)}-\mu_2)'\mathbf{S}^{-1}(\mathbf{x}_i^{(2)}-\mu_2)}} \tag{5}$$

The expression $(\mathbf{x}-\mu)'\mathbf{S}^{-1}(\mathbf{x}-\mu)$ or z_k^2 is the squared Mahalanobis distance from a data vector $\mathbf{x}$ to the mean of group k (Krzanowski, 1988, p. 342). Substituting into Equation 5 and assuming $p_1 = p_2 = \frac{1}{2}$ gives

$$P_k(\mathbf{x}) = \frac{e^{-\frac{1}{2}z_k^2}}{e^{-\frac{1}{2}z_1^2} + e^{-\frac{1}{2}z_2^2}}. \tag{6}$$

The 200 training images are used to compute the pooled covariance $\mathbf{S}$ and means μ_1 and μ_2 so that the posterior probability that an image with features $\mathbf{x}$ belongs to a class c_k can be computed according to Equation 6. The image is assigned to that class with the greatest posterior probability, in this example, $P_1(\mathbf{x})$ or $P_2(\mathbf{x})$.

2.2.2 Support Vector Machines

The *support vector machine* (SVM) is a one of the most popular classification algorithms in machine learning. The complex algorithm will not be discussed in complete detail because one would typically use an open source implementation (Marsland, 2009, p. 119). Many SVMs can be trained to complete multi-class classification, but a single SVM can only separate two classes. A SVM identifies an optimal classification line between two classes such as the one shown in Figure 2. A classification plane or hyper-plane exists for classification in higher dimensional spaces (Figure 2 shows a two-dimensional space).

A SVM takes the datapoints which lie closest to the classification line to be support vectors. SVMs fundamentally differ from LDA because only the support vectors are used for classification. Suppose that a classifier can be defined by the line given by $\mathbf{w} \cdot \mathbf{x} - b = 0$, as shown in Figure 3, where $\mathbf{w}$ is a weight vector and $\mathbf{x}$ is an input vector. The classifying line has a margin on either side of length M. These margins pass through the datapoints that have been identified as support vectors and define the decision boundaries. A datapoint is classified as either 1 or -1 depending on whether it satisfies $\mathbf{w}\cdot\mathbf{x}-b \geq 1$ or $\mathbf{w}\cdot\mathbf{x}-b \leq -1$. We seek to maximize the margin M which separates the support vectors from the classification line. This is the same as minimizing $\mathbf{w}\cdot\mathbf{w}$. Since the two classes may not be separable, unlike the case shown in Figure 3, a parameter λ must be introduced to allow for misclassification. Now we seek to minimize

$$L(\mathbf{w},\epsilon) = \mathbf{w}\cdot\mathbf{w} + \lambda\sum_{i=1}^{R}\epsilon_i, \tag{7}$$

where R is the number of misclassified data points and ϵ_i is the distance to the correct boundary line for the misclassified point, and λ is a free parameter which weights the impact of the misclassified points. Now that misclassification is allowed, the constraints for each class are that points from class 1 satisfy $\mathbf{w}\cdot\mathbf{x}_i - b \geq 1-\epsilon_i$ and that points from class 2 satisfy $\mathbf{w}\cdot\mathbf{x}_i - b \leq -1+\epsilon_i$. This problem can be transformed and solved using quadratic programming (Marsland, 2009, p. 124). Transforming the form of the problem using Lagrange multipliers gives

$$L(\epsilon) = \max\left(\sum_{i=1}^{R}\alpha_i - \frac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{R}\alpha_i\alpha_j t_i t_j \mathbf{x}_i\cdot\mathbf{x}_j\right), \tag{8}$$

subject to the constraints $0 \leq \alpha_i \leq \lambda$ and $\sum_{i=1}^{R}\alpha_i\mathbf{x}_k = 0$ (Marsland, 2009, p.125). The paper (Burges, 1998) provides a more thorough explanation of support vector machines.

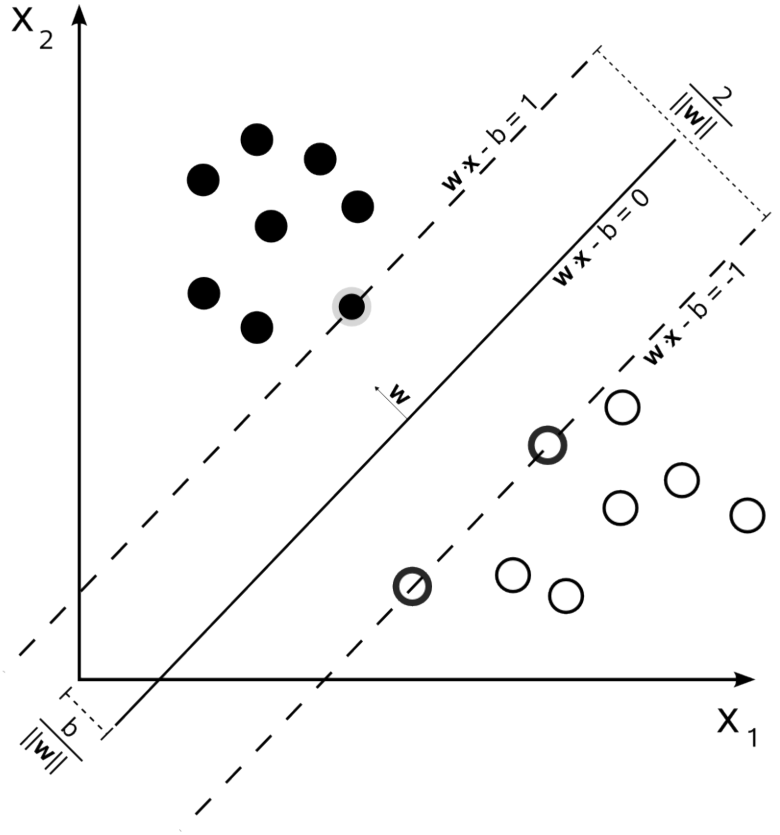


Figure 3: SVM defines a line with margins to establish class separation boundaries.

In order to linearly separate datapoints, SVMs use a kernel which adds dimensionality to the datapoints. A kernel can be selected based on knowledge of the or by simply testing. The problem is the same, except the datapoints have been transformed using a kernel function. Since the datapoints only appear as a part of an inner product in Equation 8, no computations actually need to be done in the higher dimensional space. This saves computation. One commonly used basis function is the radial basis funciotion expansions of the $x_k s$ with parameter σ and kernel:

$$\mathbf{K}(\mathbf{x}, \mathbf{y}) = \exp(-(\mathbf{x} - \mathbf{y})^2/2\sigma^2) \tag{9}$$

A single SVM only works for two classes, however, many can be used for N-class classification. N SVMs are trained to classify one class from all others. The SVM which makes the strongest prediction for a given input is used to classify that input. The strongest prediction is the one where the basis vector input point is the furthest into the positive class region.

2.3 Bag of Keypoints Descriptor

Visual categorization (or category recognition) is a challenging problem for which few high performing systems have been developed. One method for visually categorizing images is the bag of keypoints descriptor. The bag of keypoints descriptor characterizes an image based on the distribution of visual words found within it. Images are described using visual words from an established visual vocabulary The basic steps involved in computing the bag of keypoints descriptor, shown in Figure 4, are key patch detection (keypoint identification), feature extraction (feature descriptor computation), and histogram computation. The bag of keypoints descriptors for a set of images can be used for classification.

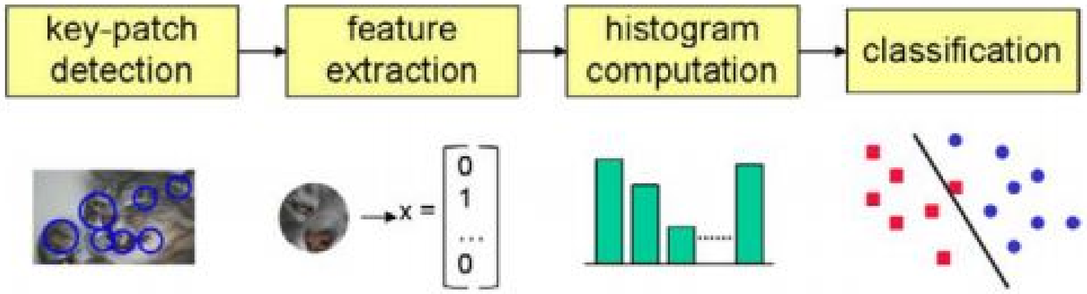


Figure 4: General steps of a bag of keypoints visual categorization algorithm (Szeliski, 2011, p. 613)

Before a bag of keypoints descriptor can be computed, a visual vocabulary must be constructed using a fixed number of training images. Training images are images with different known classes which are used to provide a classifier with knowledge of the characteristics associated with each class. A visual vocabulary receives basis from all of the feature descriptors present in all training images. A number of feature descriptors exist such as the scale-invariant feature transform (SIFT) and the Harris affine transform. Keypoints in the image are identified using an algorithm such as SIFT. Feature descriptors are then computed for all identified keypoints in each training image. The extracted descriptors are projected onto a feature space as shown in Figure 5(a). The feature space has one dimension corresponding to each dimension encoded by the chosen feature descriptor. The points in feature space are then clustered as shown in Figure 5(b); the cluster centers constitute the visual vocabulary. Clustering is accomplished using an algorithm such as k-means clustering or expectation maximization. Each cluster center is a visual word.

The visual vocabulary is then used to compute the bag of keypoints for both the training images and the validation images. Validation images are used to test a classifier's ability to assign classes correctly using the training images. The algorithm computes the distribution histogram of visual words found in the validation image and compares the distribution to those found in the training images (Dougherty, 2013, p. 14). A histogram is then created for each training image based on the visual vocabulary defined based on all of the training images as shown in Figure 5(c). A classification algorithm is then trained using the histograms produced by training images. The same feature descriptors are computed for the keypoints of validation images and the descriptors are projected on the feature space to produce a histogram as shown in Figure 5(d). The histograms of validation images are then classified by the algorithm that was trained using the histograms of training images (Csurka, Dance, Fan, Willamowski, & Bray, 2004).

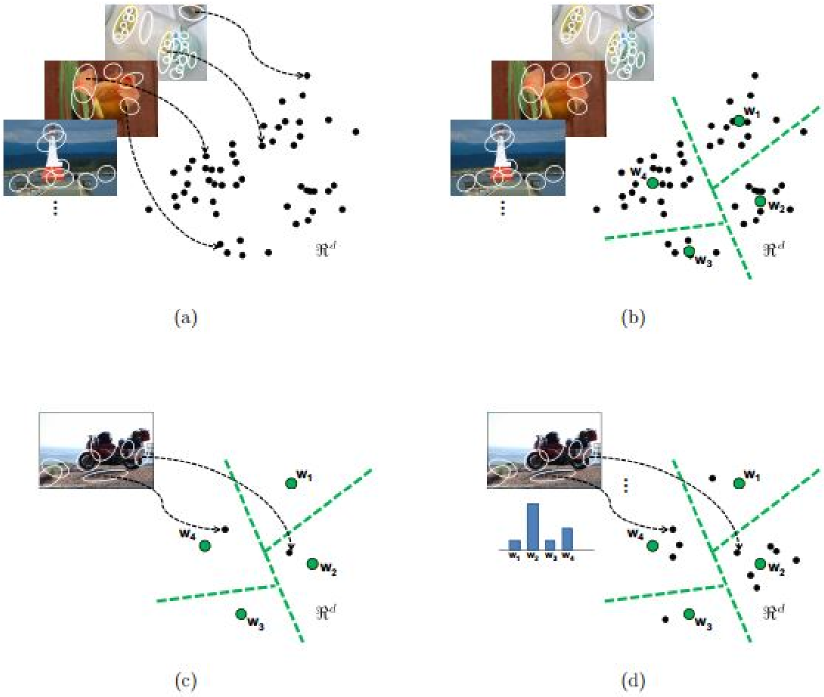


Figure 5: Projection of training image descriptors into feature space (a); clustering of points and identification of cluster centers to create a visual vocabulary (b); projection of image descriptors into feature space (c); histogram produced by comparing image descriptors to the visual vocabulary (d). The points w_1, w_2, w_3, and w_4 represent cluster centers or visual words. The green dashed lines are the decision boundaries associated with these four visual words. Each point in the scatter plot represents a single image feature. The image features are used to construct the visual words in (a) and (b). The visual words are then used to compute the bag of keypoints descriptor for an image in (c) and (d). In (d), it can be seen that one feature corresponded to visual word w_1, seven features corresponded to visual word w_2, 4 features corresponded to visual word w_4, and one visual word corresponded to visual word w_3. The histogram accurately reflects this distribution. This histogram is the bag of keypoints descriptor for the image. (Grauman & Leibe, 2011, p. 63)

3 Methods

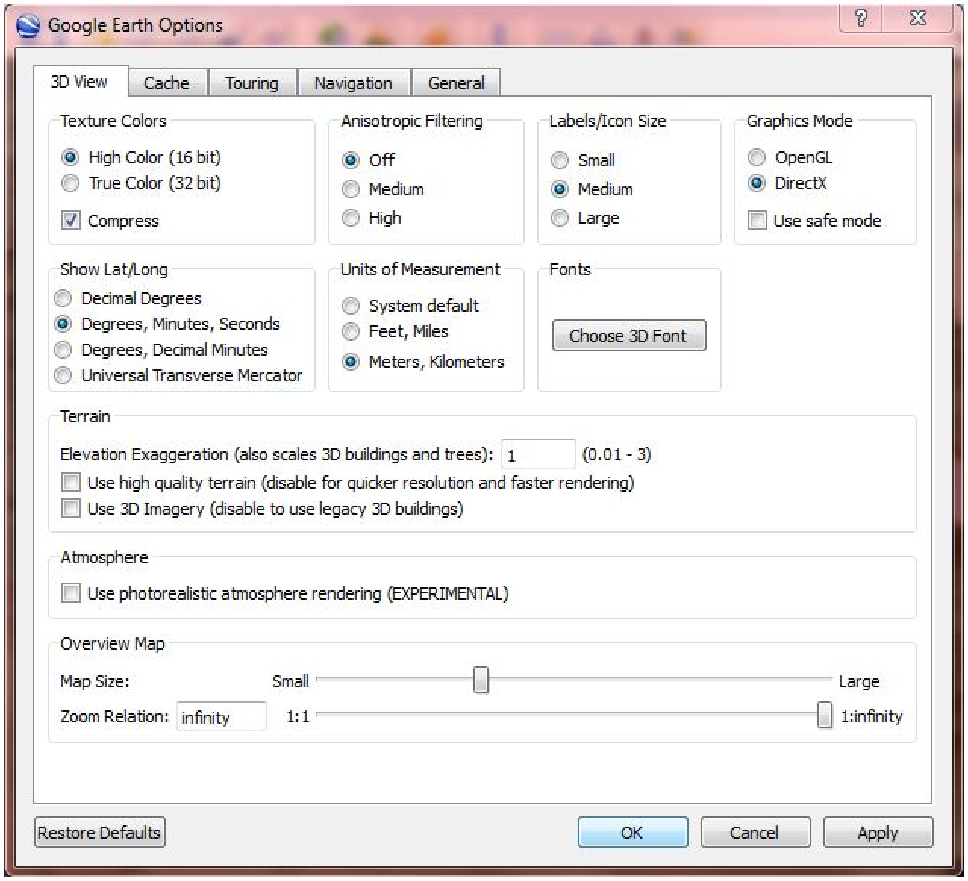
The class of the image in terms of scene or clutter level could be used to inform processing and tracking. Different image features were tested by using them to classify images. This section describes the test data, features, and algorithms used to investigate scene classification. Initial tests were completed to select parameters. Tests that followed compared different features in terms of their ability to facilitate image classification. The image sets, features, classification algorithms used, and test conditions, summarized in Table 1, are described in the following section.

Table 1: Summary of tests. Tests used either 9 image sets or 4 image sets. In the first test, images from 9 image sets were classified into 9 classes (“sea,” “rural,” “desert,” “forest,” “preserve,” “coast,” “suburban,” “urban,” or “metropolitan”). In the second test, images from 9 image sets were classified in to 2 classes (“low clutter” or “high clutter”). In the third test, images from 4 image sets were classified into 4 classes (“rural,” “forest,” “suburban,” or “metropolitan”). In the fourth test, images from 4 image sets were classified in to 2 classes (“low clutter” or “high clutter”). In each test the number of training images used was varied.

Feature	Classifier	Test Conditions		
μ	LDA	9 Image sets	9 Classes	10, 20, ...90 Training images per image set
σ	LDA			
c_v	LDA		2 Classes	
$\sigma_{\mathbf{P}}$	LDA	4 Image sets		
$\sigma_{\mathbf{P}}$ & μ	LDA		4 Classes	
BoK	SVM			

3.1 Data Collection

Accessible satellite data from Google Earth were used to investigate a methodology for implementing scene classification. The satellite data are low-noise visible band images captured from directly overhead. The satellite images were collected systematically using Google Earth software. Images were captured at a camera altitude of about 200-250m with respect to the ground elevation. The resolution of the Google Earth viewing window was set to 720×576 pixels. Images were cropped to the 512×512 pixels in the top lefthand corner in order to remove extraneous text inserted at the bottom of each image. The information present at the bottom of each image lists the image date, coordinates in latitude and longitude, elevation, and eye altitude. A consistent rendering configuration was also used as shown in Figure 6. A configuration was selected which minimizes the use of visual effects and post-processing.



Texture Colors
● High Color (16 bit)
○ True Color (32 bit)
☑ Compress

Anisotropic Filtering
● Off
○ Medium
○ High

Labels/Icon Size
○ Small
● Medium
○ Large

Graphics Mode
○ OpenGL
● DirectX
□ Use safe mode

Show Lat/Long
○ Decimal Degrees
● Degrees, Minutes, Seconds
○ Degrees, Decimal Minutes
○ Universal Transverse Mercator

Units of Measurement
○ System default
○ Feet, Miles
● Meters, Kilometers

Fonts

Terrain
Elevation Exaggeration (also scales 3D buildings and trees): 1 (0.01 - 3)
□ Use high quality terrain (disable for quicker resolution and faster rendering)
□ Use 3D Imagery (disable to use legacy 3D buildings)

Atmosphere
□ Use photorealistic atmosphere rendering (EXPERIMENTAL)

Overview Map
Map Size: Small — Large
Zoom Relation: infinity 1:1 — 1:infinity

Figure 6: Google Earth 3D View options used for data collection.

The 900 images collected spanned nine distinct scenarios (classes). Scenarios were selected by visually inspecting satellite imagery of the United States. Satellite imagery that appeared visually similar in terms of content was determined to belong to a single class. A class did not need to be representative of a scenario commonly found throughout the United States. Classes were selected to be visually different in a cognitive sense. Nine classes with 100 images each were identified. A human could easily identify the class of one of the 900 images given an example of each class. The nine different scene classes were labelled as "sea," “rural,” “desert,” “forest,” “preserve,” “coast,” “suburban,” “urban,” and “metropolitan”. The image sets are listed in order of increasing clutter content. Images of the “sea” image set were considered to be the least cluttered and “metropolitan” images are considered to be the most cluttered. Examples of each class are shown in Figures 21-29 located in the Appendix. These 9 image sets corresponding to 9 classes were also classified as either “high clutter” or “low clutter” as shown in Table 2. Images were classified in terms of both scene and clutter content separately.

Table 2: High/low clutter class assignment for the 9 image sets

Low clutter			High clutter	
Sea	Desert	Preserve	Suburban	Coast
Rural	Forest		Metropolitan	Urban

3.2 Image Features

Statistical image features used were the intensity mean, μ, intensity standard deviation, σ, intensity coefficient of variation, c_v, and P-cell average intensity standard deviation, $\sigma_{\mathbf{P}}$. For an M×N image, the intensity of a single pixel is given by $x_{m,n}$, where m and n denote conventional image indices. The RGB pixels of each image were converted to grayscale according to

$$x = 0.2989R + 0.5870G + 0.1140B \tag{10}$$

where R, G, and B are 16 bit High Color red, green, and blue values, respectively. The pixel intensity corresponding to the top left-hand corner of the image is given by $x_{0,0}$. The intensity mean of an image is given by

$$\mu = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x_{m,n}, \tag{11}$$

the intensity standard deviation is given by

$$\sigma = \sqrt{\frac{1}{(MN-1)} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x_{m,n} - \mu)^2}, \tag{12}$$

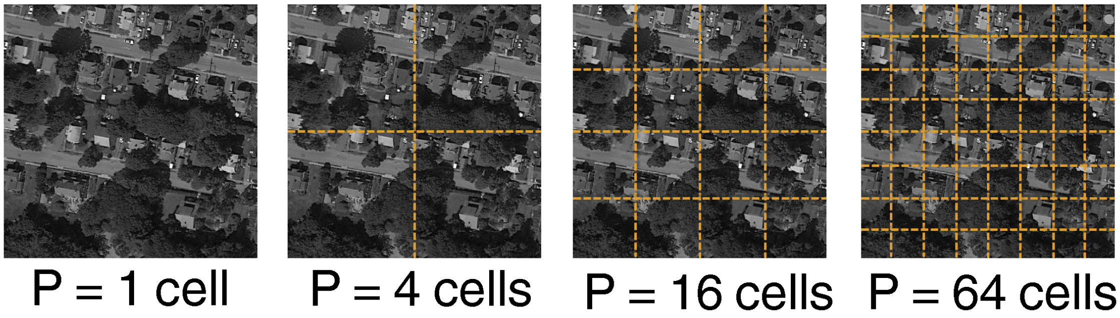
the intensity coefficient of variation is given by,

$$c_v = \frac{\sigma}{\mu} \tag{13}$$

and the P-cell average intensity standard deviation is given by

$$\bar{\sigma}_P = \frac{1}{P} \sum_{i=0}^{P-1} \sigma_i, \tag{14}$$

where the intensity standard deviation for the ith cell, σ_i is calculated as shown in Equation 12. The number of cells, P, is the only parameter that can be varied. An image is divided into P cells for different values of P is shown in Figure 7. The size of each cell should be twice the size of a target in order to best quantify the clutter present in the scene. Square cells are always used. A study has shown that basing cell size on the target allows the metric to provide the best measure for the detectability of the target in a given scene. In this case, no target is present in the imagery and the P which best facilitates classification is not related to the target size. The parameter P will be varied in testing. The ability of each feature to achieve linear separability was tested using LDA.



P = 1 cell

P = 4 cells

P = 16 cells

P = 64 cells

Figure 7: A “suburban” image divided into P cells. The standard deviation of each cell is computed. The mean of these standard deviations is taken to be $\bar{\sigma}_P$. The standard deviation of the first cell in any image is given by σ_1.

The more complex image feature used was the bag of keypoints (BoK) descriptor (Csurka et al., 2004). The bag of keypoints descriptor describes the image using a visual vocabulary of image features established by training. In order to describe an image using the visual vocabulary, image features must be extracted and matched to features within the visual vocabulary. The BoK descriptors can be regarded as a histogram indicating the number of times each feature from the visual vocabulary matched a feature extracted from the image. The features can be extracted from an image using a variety of approaches. The Scale Invariant Feature Transform (SIFT) was used to identify keypoints and extract image descriptors. The SIFT descriptor has been described as a position-dependent histogram of local gradient directions around an interest point (Lindeberg, 2012). The reader is encouraged to refer to Lindeberg (2012) for a thorough explanation of SIFT. A support vector machine was used to classify images based on their BoK descriptors. Parameters of the BoK technique were varied to select fixed parameters to use in final tests.

A BoK descriptor is constructed using SIFT keypoints; therefore, any frame not containing SIFT keypoints is not classifiable using this method without adding an exception. If no SIFT keypoints were identified in an image, the image was classifieed as either “sea” when using 9 classes, “rural” when using 4 classes, or “low clutter” when using 2 classes.

Initial tests were completed to identify which number of clusters and number of SIFT features per image most consistently rendered the greatest classifier accuracy. These tests used 10 training images per class and 90 validation images per class. A low number of training images were chosen in order to save computation time. Consistent SVM parameters, described in Section 3.4, were also used throughout this study. The maximum number of features extracted from each image was varied: 25, 50, 100, 150, 200, 250, 300, and 500 features per image. The number of clusters used to form a visual vocabulary was varied from 10 to 1000 clusters in increments of 10 clusters and from 1000 to 5000 clusters in increments of 50 clusters. Decay in performance was expected to occur as the number of clusters used to form a visual vocabulary approached the total number of features used to train the visual vocabulary.

3.3 Classification Methods

The primary metric used to evaluate the performance of a classification approach was the classification accuracy. The classification accuracy is defined here as the total number of correctly classified validation images divided by the total number of validation images. The feature(s) of classified images presented as a scatter plot is useful for inspecting the linear separability of the images based on those feature(s).

The scatter plot shown in Figure 8 shows the $\bar{\sigma}_{64}$ of each validation image as well as the decision boundaries established using LDA. In this example, images were classified using a single feature, so the features could have been plotted along a single line. Plotting the features in a two-dimensional space allows one to distinguish individual data points. A confusion matrix can be used to show the number of times an image of a particular class was determined to belong to each class. An example of a confusion matrix is shown in Table 3. The confusion matrix indicates that one “rural” image was misclassified as “suburban.” The scatter plot, consistent with the confusion matrix, shows a that single “rural” data point does actually fall within the decision boundaries for “suburban” images. LDA acheieved a classification accuracy of 0.90 in this situation. This would indicate that the 4 classes have a high linear separability based on $\bar{\sigma}_{64}$. In order to confirm this, one can inspect the scatter plot. The scatter plot shows that the “forest” image set occupied a small range of $\bar{\sigma}_{64}$ values, indicating that images of the “forest” class had consistent characteristics. The “metropolitan” image set spanned a range of $\bar{\sigma}_{64}$ values three times larger. The scatter plots and confusion matrices were inspected in this manner to obtain greater detail.

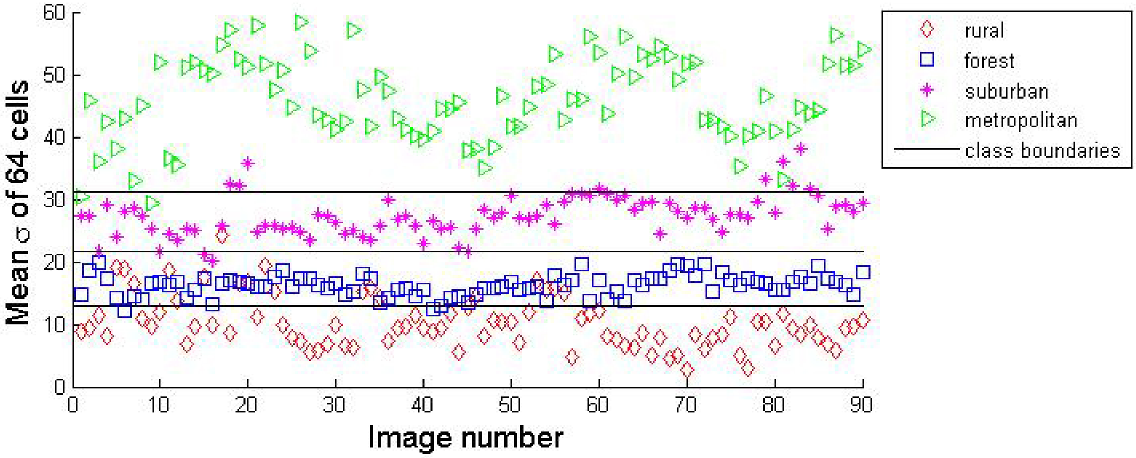


Figure 8: A scatterplot showing the $\bar{\sigma}_{64}$ for the 360 validation images of 4 different classes. The first 10 images of each of the 4 image sets were used as training images and the remaining 90 images from each image set were used as validation images. This scatterplot corresponds to the confusion matrix of Table 3.

Table 3: Example of a confusion matrix. Images from 4 image sets were classified into their 4 respective classes. Rows indicate image set and columns indicate class. Elements along the diagonal correspond to correct classifications. For example, 71 “rural” images were correctly classified as “rural,” 18 “rural” images were misclassified as “forest,” and 1 “rural” image was misclassified as “suburban.” The class labels are presented in order of increasing clutter content from left to right. In this case, “rural” is the class with the lowest clutter and “metropolitan” is the class with the highest clutter. These images were classified using LDA and the $\bar{\sigma}_{64}$ of each image. This confusion matrix corresponds to the scatter plot of Figure 8.

	Rural	Forest	Suburban	Metropolitan
Rural	71	18	1	0
Forest	3	87	0	0
Suburban	0	3	78	9
Metropolitan	0	0	2	88

The proportion of training images to validation images was varied because this is an important consideration in any classification problem. Different classification approaches often require differing degrees of training. Varying the proportion of training images allowed for a more comprehensive comparison of the different image features.

The test conditions sought to emphasize the hypothesized advantages presented by the different classification approaches. Increasing the variety and number of images, such as when classifing 9 image sets into 9 classes, was expected to favor the BoK/SVM approach. As the dimensionality of the classification problem is reduced, such as when classifying images into 2 classes, the approach using LDA with scalar statistical features was expected to rival or surpass the performance of the BoK/SVM approach in terms of overall accuracy.

Overfitting can occur in classification models with large numbers of adjustable parameters or high flexibility. Overfitting occurs when a model fits the noise as as well as the signal. Small data sets are especially prone to overfitting. In many cases, a data set represents only a small example of actual data. A model which fits the noise may work well for a small data set, but not for actual data. Cross validation was used to prevent overfitting.

The parameters of a classification model are established using training and validation data, in this case 900 images. Each image can either serve as a training image or validation image. It is common to use 80% of the data for training and 20% for validation. Initially training was completed using the first n images of each class. The remaining 100-n images of each class were used as validation images. Instead of using the same images for training each time, cross validation was used. In cross validation, the data used for training and validation is cycled in a round-robin fashion (Murphy, 2012). The results of classification are averaged across the different combinations. For each test, 10 different combinations were used such that each image was used for training an equal number of times. This technique helps to prevent overfitting. Cross validation was implemented by dividing each image set into 10 folds consisting of 10 images each. The first fold consisted of images 1-10, the second of images 11-20, etc. Cross validation cycled which fold(s) were used for training. For example when using 30 images (3 folds) from each set for training, folds 1-3 were used for training first. Subsequently, folds 2-4, 3-5, 4-6, 5-7, 6-8, 7-9, 8-10, 9-1, and 10-2 were used. The results of classification using these 10 combinations were averaged. Cross validation was used to reduce the impact of training image selection on results.

3.4 Implementation of Classifiers

MATLAB's statistical toolbox offers multi-class LDA. The `classify` command was used with the argument `"linear"` to do multi-class LDA. The command takes a matrix of features of validation images, a matrix of features of training images, and a vector of truth labels associated with the training images. The `confusionmat` command was used to produce confusion matrices.

An open source computer vision library, OpenCV, offers C++ libraries for implementing SIFT, BoK, and SVMs. The parameters of the support vector machine were kept constant. The OpenCV implementation of the support vector machine is intended for use with bag of keypoints method. The support vector machine used the following parameters:

- type: C-support
- kernel: Radial basis function $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$
- C value: 10 (number of tolerable errors during training)
- end criteria: algorithm-dependent accuracy reaches machine epsilon, ϵ $(2.2204 \cdot 10^{-16})$
- max number of iterations: 1000
- required accuracy: ϵ

The C++ classes `SiftFeatureDetector`, `SiftDescriptorExtractor` were used with the `detect` and `compute` methods to detect keypoints and compute SIFT descriptors for each image. The `BOWKMeansTrainer` class was used to construct the visual vocabulary with a fixed number of clusters. The `add` method was used to add SIFT descriptors to the trainer before clustering the descriptors using the `cluster` method.

The visual vocabulary was then associated with an instance of the `BOWImgDescriptorExtractor` class using the `setVocabulary` method. The `BOWImgDescriptorExtractor` was initialized to use an instance of the

`SiftDescriptorExtractor` class and a “FlannBased” `DescriptorMatcher`. The SIFT keypoints of all images were identified again using the `detect` method of the `SiftFeatureDetector`. The `compute` method of `BOWImgDescriptorExtractor` class was then used to compute each image’s SIFT descriptors, match those SIFT descriptors to the SIFT descriptors representing the cluster centers (visual words), and compute a bag of keypoints descriptor.

Instances of the `CvTermCriteria` and `CvSVMParams` classes were then initialized with the parameters described above using the calls

```
CvTermCriteria criteria = cvTermCriteria(CV_TERMCRIT_EPS, 1000, FLT_EPSILON);
CvSVMParams svm_param = CvSVMParams( CvSVM::C_SVC, CvSVM::RBF, 10.0, 8.0,
                        1.0, 10.0, 0.5, 0.1, NULL, criteria);.
```

An instance of the `SVM` class was then instantiated and trained with the bag of keypoints descriptors of the trainging images using the `train` method. Finally, the `predict` method was used to classify the validation images based on their bag of keypoints descriptors.

One should refer to (Burges, 1998) and (Chang & Lin, 2011) for a more thorough description of support vector machine implementations. OpenCV’s support vector machine libraries are based on LIBSVM. The LIBSVM implementation document (Chang & Lin, 2011) contains a wealth of infomation.

4 Results

4.1 Parameter Selection

The classification algorithm parameters were fixed as described in the previous section; however, tests were completed to determine parameters to be used for the $\bar{\sigma}_{\mathbf{P}}$ and the BoK descriptor. LDA was done using the $\bar{\sigma}_{\mathbf{P}}$ with 4 image sets and 4 classes. The training images consisted of 10 image from each image set and the validation images consisted of 90 images from each image set. The number of cells, P, was varied. For each value of P, LDA was completed 10 times with 10 different sets of training images. The average classification accuracies and standard deviations are shown in Figure 9. Note that using a P of 1 is the same as taking the standard deviation of the entire image. The classification accuracy peaked at 64 cells. As a result, $\bar{\sigma}_{\mathbf{P}}$ was used for the classification experiments that followed.

The BoK descriptor of an image is defined in terms of a visual vocabulary consisting of k clusters of SIFT keypoints identified using k-means clustering. A maximum number of SIFT keypoints are identified in each training image. For example, if 200 images are used for training and 100 keypoints are extracted from each, then the visual vocabulary is constructed by clustering 20000 SIFT descriptors (assuming the maximum number of keypoints is identified in each image). In order to cluster these 20000 SIFT descriptors, k must be selected such that related SIFT descriptors are associated with the same visual word. If k is set to 1, then all SIFT descriptors correspond to the same cluster or visual word. If k is set to 20000, then each visual word would correspond to a single SIFT keypoint and the BoK descriptor of each image would be unique, consisting of 100 unique visual words. The BoK descriptor would not provide a useful description at these limitations. Varying its parameters rendered the results shown in Figure 10. Classification with 25, 50, 100, and 500 features extracted per image produced the worst accuracy. For 50-1000 clusters, classification with 150, 200, 250, 300, and 500 clusters all produce comparable accuracies. At 1000 clusters, classification with 500 features extracted per image begins have a downward sloping accuracy.

Classification with 150 features per image most frequently achieves high accuracy for 1000-5000 clusters. This parameter selection test was completed with 10 training images from each image set. Using more training images increases the total number of features used when creating the visual vocabulary. Therefore, increasing the number of training images may cause decay in performance to occur if too many clusters are used. Decay in performance at a high number of clusters was observed for 500 features per image. Hence, the lowest number of features per image and clusters, 150 features per image and about 100 clusters, which produced a fairly high and stable accuracy were the selected parameters. In the context of Figure 10, high accuracy is considered to be an accuracy of around 0.7 and the accuracy is considered stable if increasing or decreasing the number of clusters by about 50 does not impact accuracy substantially. It was especially important that the accuracy be stable for a higher number of clusters because increasing the total number of features used appeared to cause decay to occur at a lower number of clusters. Tests varying the number of training images would need to be conducted in order to reach this conclusion; however, the parameters were selected to take into account that possibility.

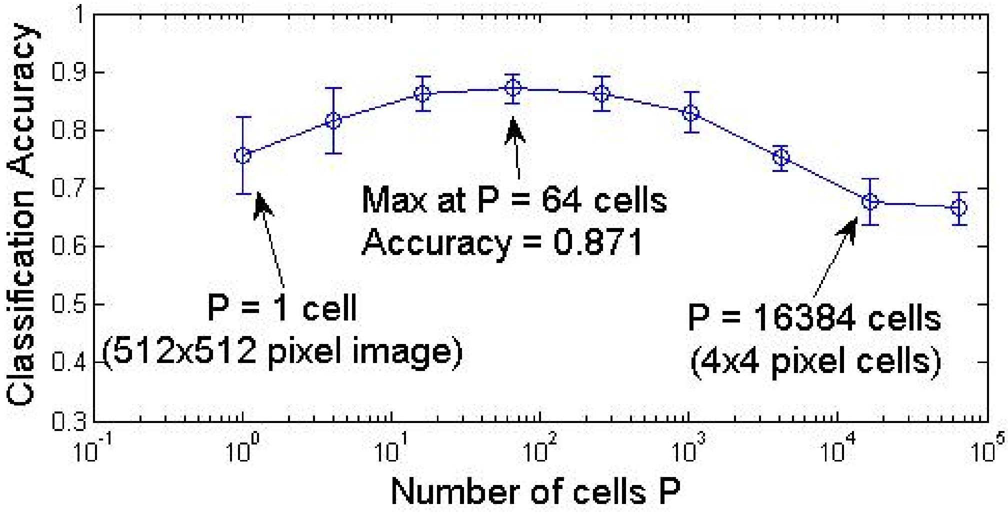


Figure 9: Classification accuracy using $\sigma_{\mathbf{P}}$ for various P values.

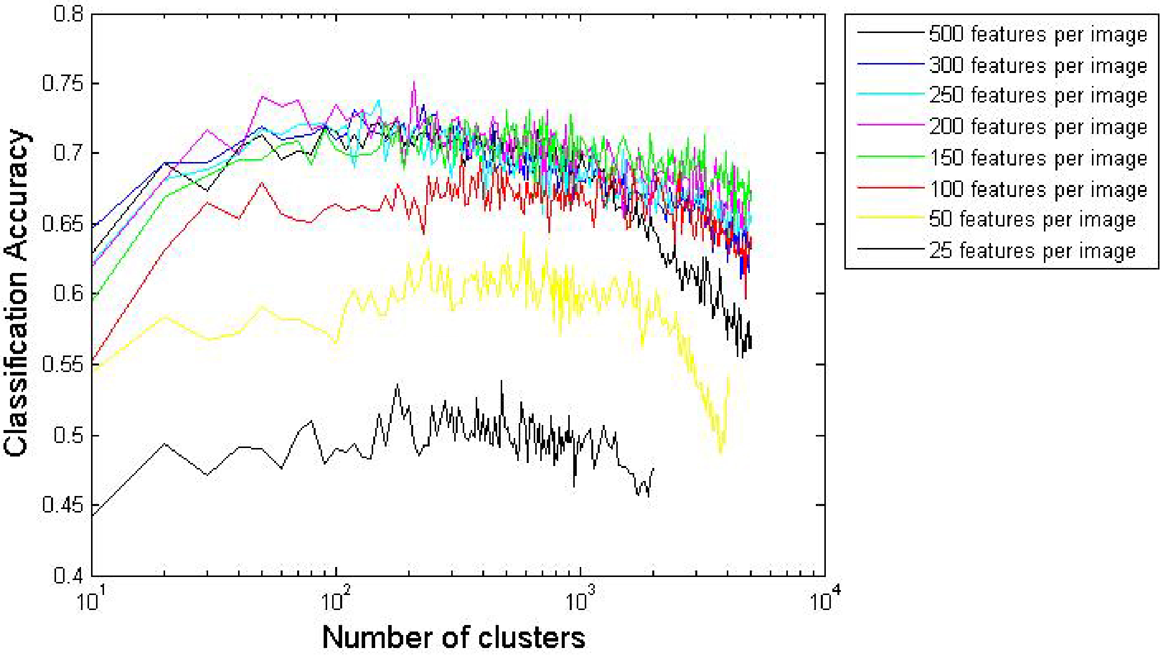


Figure 10: Semi-log plot showing the classification accuracy for SVM/BoK. The horizontal axis shows the number of clusters or size of the visual vocabulary used. The training set consisted of 10 images from each class (90 images). The validation set consisted of 90 images from each class (810 images). The bottom black curve corresponds to 25 features per image.

4.2 Classification Using Different Image Features

The six image features were tested under four classification scenarios and with a varied number of training images. The results of the experiments are summarized in Figure 11. The approach using the BoK descriptor usually achieved the highest accuracy. Even when classifying 9 image sets into 9 classes, the BoK descriptor outperformed $\bar{\sigma}_{64}$ & μ by about 10%. Its accuracy was usually comparable to and sometimes less than the accuracy attained when classifying images based on $\bar{\sigma}_{64}$ & μ. Scalar image features all rendered low accuracies below 0.6 when classifying 9 image sets into 9 classes. The image mean, μ, never reached an accuracy greater than 0.7. Additionally, μ was the only feature to experience a decrease in accuracy when classifying 4 image sets into 2 classes instead of 4 classes. Besides this exception, the accuracy always increased when classifying image sets into 2 classes instead. In the two class scenarios, many of the one-dimensional image features produced accurcacies on par with the higher dimensional features.

The number of training images used appeared to have the greatest impact on BoK-based classification; increasing the proportion of training images to validation images increased the average accuracy somewhat. In general, increasing the number training images and decreasing the number of validation images did not affect the average classification accuracy. For the other features, increasing the proportion of training images did not always increase accuracy.

Large error bars indicate that the classification performance varied greatly when using different images for training. If error bars for a particular feature are large, then the images chosen had varied properties with respect to that feature. The standard deviation feature, σ and mean feature μ often had the largest error bars, especially in scenarios with 4 image sets.

Inspecting the four features for all 900 images provides insight regarding the linear separability of classes. Figure 12 shows that none of the four features achieves good linear separabiliity when classifying 9 image sets into 9 classes. The top-left scatter plot showing μ for each image differs greatly from the three other scatter plots. The top-left plot is the only one in which the "metropolitan" does not lie at the top. Additionally, besides the "sea" and "forest" image sets, no other image set appears to fall into any distinctive layer. In fact, the "rural" images span a great range of μ values. This makes sense because the color of the rural images varies greatly. This makes μ a poor feature to use for classification independently.

In Figure 11, it is evident that μ renders a low average classification rate in all classification scenarios. The difference in the average accuracy between μ and all other features is emphasized as the complexity of the classification problem is reduced from classifying 9 image sets to classifying 4 image sets. Th 4 image sets used were "rural," "forest," "suburban," and " metropolitan." The top-left scatter plot of Figure 12 shows that the "rural" and "metropolitan" image sets are two of the image sets which span large ranges of μ values.

For the other three scatter plots, using only these 4 image sets allows for linear separability as shown in Figure 8. This results in a great accuracy difference between μ and the other features.

Since the scatter plot of μ differed greatly from the scatter plots of the three other features, it is a suitable candidate for being used with another feature in two-dimensional classification. In order to test this assertion, the scatter plot of μ and $\bar{\sigma}_{64}$ for all 900 images, shown in Figure 13, was inspected. Immediately, one notices that the 9 different classes are much more distinguishable. Although the image sets cannot be separated without error, the potential for good class boundaries is much greater. Interestingly, the “sea,” “forest,” “suburban,” “urban,” and “metropolitan image sets appear to fall along a line with a constant slope. These 5 classes are considered to be more regular scenes. The image sets were chosen because they were expected to differ in terms of σ and in terms of SIFT features. The “preserve,” “rural,” “desert,” and “coast” image sets were intended to be more irregular. It was not initially obvious how these 4 irregular image sets would compare to the other 5 image sets. The “coast” image set, for example, contains elements of both the “sea” and “metropolitan” image sets. The “rural” image set is known to contain low complexity features with varying colors, so it makes sense that σ is low and that μ spans a wide range of values. The average accuracy attained by jointly using μ and $\bar{\sigma}_{64}$ rivaled that attained by using the BoK descriptor.

The average accuracy achieved using c_v seemed to behave somewhat unpredictably. In the top-left plot of Figure 11, it gave the worst accuracy. In the top-right, it gave an accuracy between μ and σ. This result coud be expected since the c_v depends on both μ and σ. Strangely, the c_v rendered an average accuracy rivaling that given by $\bar{\sigma}_{64}$ in the 4 image set situation shown by the bottom plots. It exceeded the average accuracies of both μ and σ by a large margin in the bottom-left plot.

Legend	μ	σ	c_v	$\bar{\sigma}_{64}$	$\bar{\sigma}_{64}$ & μ	BoK
	×	o	*	□	▷	◁

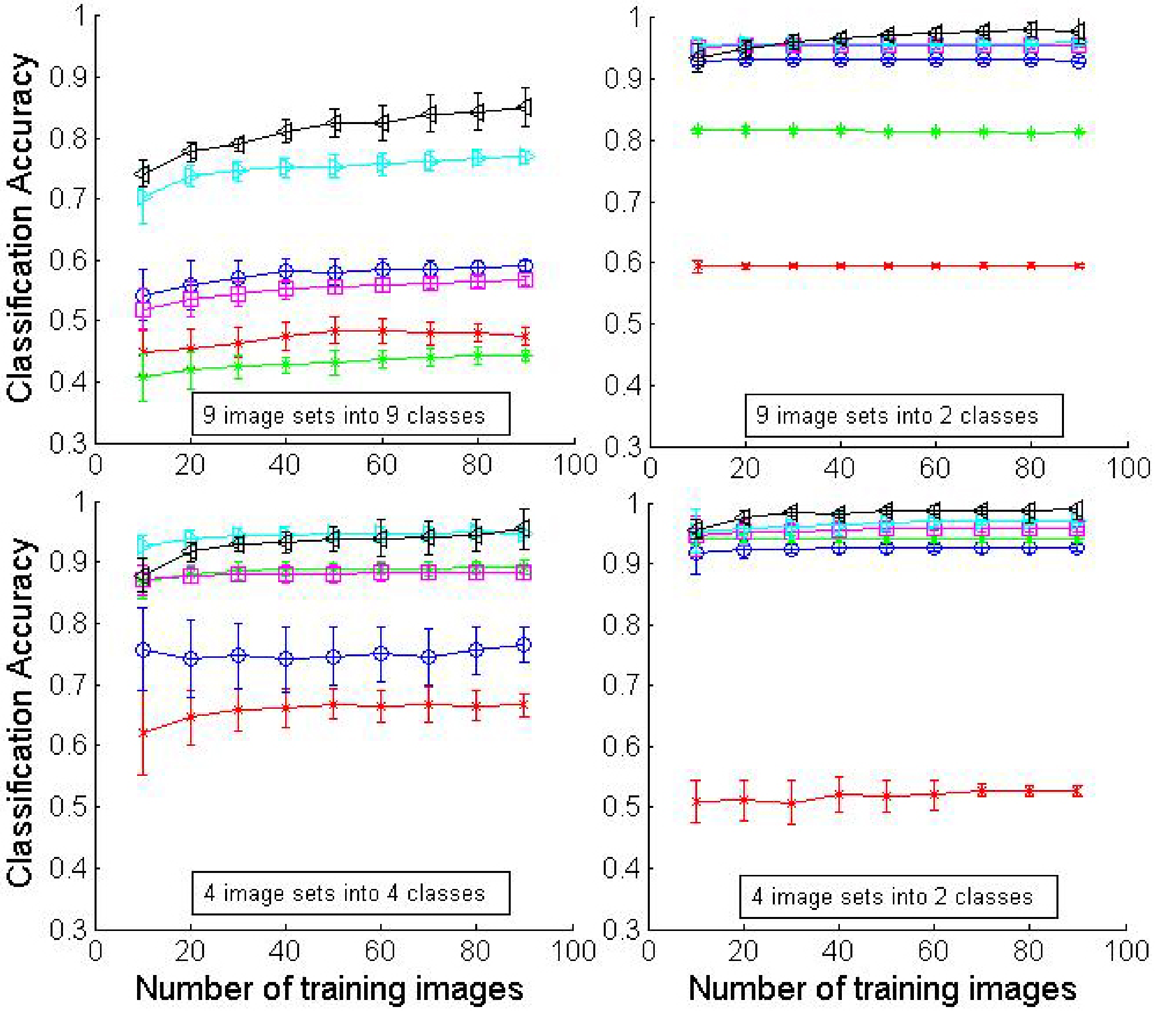


Figure 11: Plots show average accuracy for classification using different features and different numbers of training images. Each point is the average accuracy for 10 combinations of training images. Error bars indicate the standard deviation of the average accuracy for 10 combinations of training images. On the top-left, 9 image sets were classified into 9 classes. On the top-right, 9 image sets were classified into 2 classes. On the bottom-left, 4 image sets were classified into 4 classes. On the bottom-right, 4 image sets were classified into 2 classes.

Legend	sea	rural	desert	preserve	forest	coast	suburban	urban	metropolitan
	◇	□	*	▷	o	×	◁	□	*

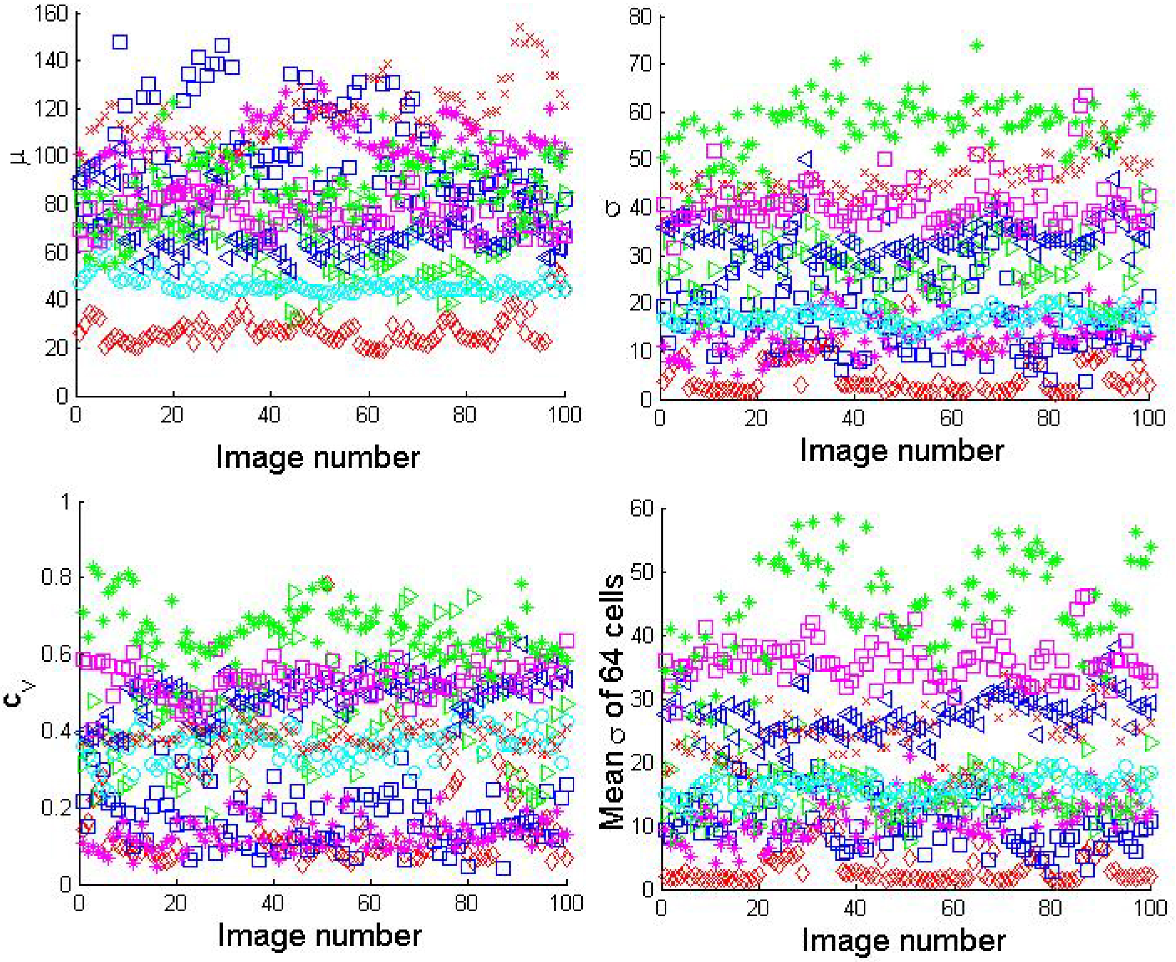


Figure 12: Four scatterplots show the four different features of all 900 iamges: μ on the top-left, σ on the top-right, c_v on the bottom-left, and $\bar{\sigma}_{64}$ on the bottom-right. Note that there is poor linear separability for all four features. Many of the classes overlap frequently.

Legend	sea	rural	desert	preserve	forest	coast	suburban	urban	metropolitan
	◇	□	*	▷	o	×	◁	□	*

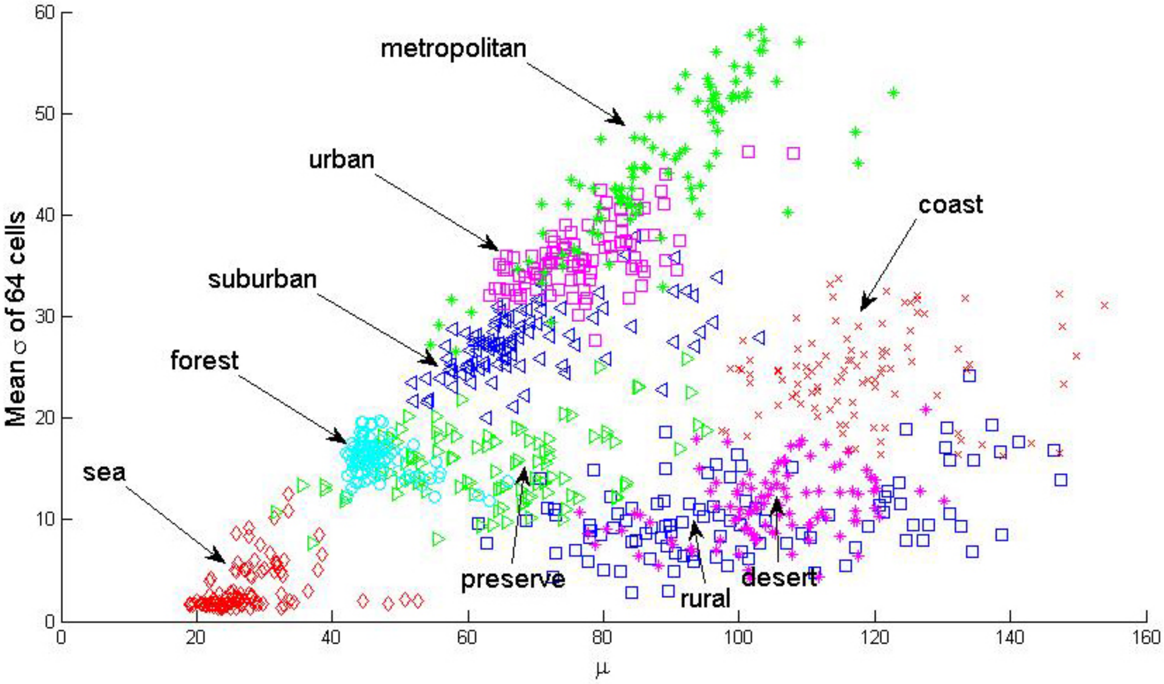


Figure 13: Scatterplot showing μ plotted against $\bar{\sigma}_{64}$ for all 900 iamges. Note that the linear separability is increased drastically from the one-dimensional cases shown in the top-left and bottom-right plots of Figure 12. Many of the classes overlap frequently.

5 Discussion

Classification was completed with 9 image sets into 9 classes and 4 image sets into 4 classes in order to highlight differences between classification based on Bok and classification based on statistical features. The BoK approach was hypothesized to attain a much higher accuracy when classifying 9 image sets into 9 classes. For a large number of classes, it was hypothesized that classes would be more likely to have similar statistical properties. The classification accuracy and linear separability of similar classes would be low. Introducing even more image classes may be able to provide a better answer to this hypothesis. The BoK approach did in fact achieve the greatest accuracy margin when classifying 9 image sets into 9 classes. Similarly, when classification was reduced to 4 distinct classes, statistical features rendered relatively high accuracies. Additionally, reducing the problem to 2-class classification improved accuracy as expected.

Theoretically, if a set of images consisted of the same shuffled pixels, then the statistical properties of such a set of images would be identical. As a result, it would be impossible to classify such a set of images into different classes based on statistical properties. Pixels could be shuffled to take different forms; the same pixels could be used to show different objects or scenes. Although the images would be indistinguishable in terms of their mean and standard deviation, the BoK descriptor should hypothetically be able to distinguish between the forms or objects present in each image. Shuffling the pixels of an image could also change the $\bar{\sigma}_{\mathbf{P}}$ of the image.

For the purposes of facilitating adaptive target tracking, the classifier need only separate images into 2 classes: "low clutter" or "high clutter." Using more classes with plug-in rules could provide a better solution. As the number of classes is increased, one might want to consider classifying based on higher-dimensional features. Classifying images based on only two statistical features demonstrated a consistent increase in accuracy. Since one must compute the mean of an image in order to compute its standard deviation, classifying images using both features is certainly worthwhile.

The accuracy of the BoK approach in the 9 class and 4 class scenarios may have been impacted by the exception implemented to account for images not containing any SIFT keypoints. Any image not containing SIFT keypoints was classified as “sea” with 9 classes or “rural” with 4 classes. Inspecting confusion matrices produced by the BoK approach actually showed that not many images were misclassified as “sea.” Actually, “sea” images were often misclassified as urban images. If “sea” images being used for training contain no SIFT features, then their BoK descriptor cannot be computed either. In such a situation, any “sea” image containing at least a single keypoint is guaranteed to be misclassified.

Inspection of the confusion matrices for 9 class classification showed that misclassification occurred between “coast” and “metropolitan” images. Up to 26 “coast” images were misclassified as “metropolitan.” Both image sets were captured in Florida, as shown in Figure 18 of the Appendix. The primary qualitative difference between the two image sets is that a “coast” image contains buildings and water separated by a beach exactly as is shown in Figure 26 of the Appendix. The buildings contained in these images are similar, but the “metropolitan” image set contains many skyscrapers and roads while the “coast” image set contains small hotels and much more greenery. The examples in the Appendix, Figures 26 and 29, are good examples of ”coast“ and ”metropolitan“ images. In ”coast“ scenes, SIFT was expected to identify keypoints corresponding to sea, beach, and urban features in appropriate proportions; however, Figure 14 shows that the actual keypoints only correspond to the beach and urban features.

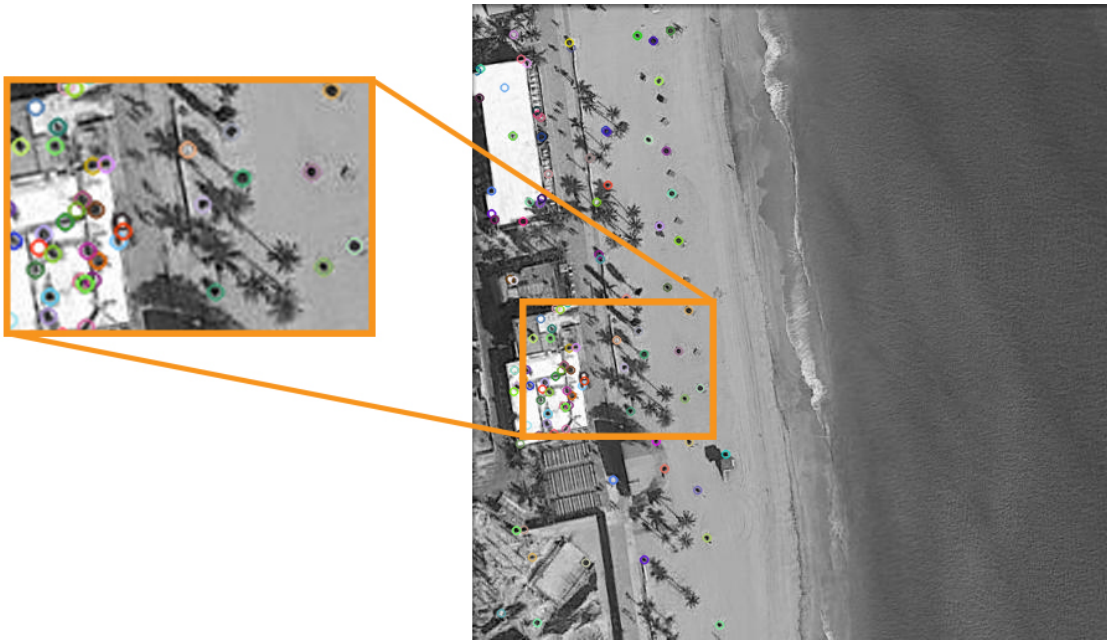


Figure 14: Keypoints identified in a typical “coast” image are represented by small colored circles. The panel on the left provides an enlarged view of a portion of the image in order to emphasize the colored keypoints.

Inspection of the confusion matrices for 9 class classification also showed that misclassification occurred between “rural” and “preserve” images. Approximately, 20 “rural” or “preserve“ images were consistently misclassified as “preserve” or “rural.” Besides the three strange misclassification pairs discussed, most other misclassifications occurred less frequently and adjacent to the diagonal of the confusion matrix. Since the classes were sorted according to their perceived clutter level, most misclassification occured between similar classes.

It is possible that decreasing the number of validation images had a greater impact on accuracy than increasing the number of training images. Instead of always using the full 900 image data set, tests can be completed by fixing the number of training images or validation images to 50 and varying only the number of validation or training images. Such tests would allow one to determine the effects of the number of training images and number of validation images.

Inconsistent classification accuracies when using different training images cause the average accuracy to have a high standard deviations. Inconsistency may have been more pervasive in certain classes. For example, the coefficient of variation, c_v for certain sea images reached values typical of "metropolitan," "urban" or "suburban" images. Using outlier images in a small training set could cause negative bias. Cycling training sets by means of cross validation helped to eliminate the impact of outliers on the results. The misclassification rate for "sea" images when using BoK, for example, was identified to vary depending on whether the training images contained any keypoints. Misclassification increased significantly if the training images did not contain keypoints.

The imagery in this project was captured at a fixed range and in the visible waveband. Satellite images of the Earth's surface are only a small subset of possible scenes. A dataset consisting of images of the same scenes at a nearer range would be completely different. Changing the viewing angle or the eye altitude might drastically change the imagery. In general, a lens or imaging device is limited by the diffraction of light and image resolution. The details present in any given image depend on the distance between the image and the sensor. A forest viewed from above has a much different texture than a single tree viewed from above with the same resolution. From far enough away, all groups of trees do not appear much different in a cognitive sense. A scene appears differently in different wavebands such as infrared or ultraviolet. Depending on the waveband of the sensor collecting imagery, certain issues must be considered. Infrared images, for example, are often more susceptible to noise than visible band images. For example, the satellite images used in this study contained no noise that would be obvious to a human observer. In the visible spectrum, certain features such as water, buildings, trees, land, and roads all have consistent grayscale pixel intensities and are fairly easy to resolve. There must be a difference in temperature in order to be able to distinguish between two objects in infrared scenes. As a result, not all objects in a scene can be resolved and radiant intensities of objects depend on a number of factors. Water vapor, pressure, temperature, and gaseous constituents in the atmosphere all affect the transmission of infrared radiation through the atmosphere. Although the classes of the images used in this study were defined in terms of content and perceived clutter level, it may be conducive to define classes based on other characteristics. This classification scheme presented would only be useful for tracking when there are a finite number of distinct background scenarios. In this project, each scene image set has been selected to be distinct.

6 Conclusions

The BoK descriptor and certain statistical image features were shown to be useful in classification. For visible band images, LDA can be used to classify images based on their statistical properties with an accuracy of about 0.95 with 2 classes, 0.92 with 4 classes, and 0.75 with 9 classes. The BoK descriptor is able to reach an accuracy of 1 with 2 classes, about 0.92 with 4 classes and 0.83 with 9 classes. The complexity may not be worth the computational cost, but it may prove to be more worthwhile for a larger number of classes.

This methodology can be applied for imagery of any waveband. No parameters were chosen directly based upon the kind of images being classified. Target trackers modified to leverage this classification scheme may be allowed to track more successfully in a variety of background scenarios. Future work to expand this project could involve implementing this classification scheme alongside an actual target tracker. Additionally, many more image features exist which may prove to be more effective for classifying images in other contexts.

References

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition.

Chang, C. C., & Lin, C.-J. (2011). Libsvm: a libarary for support vector machines. *ACM Transactions on Intelligent Systems Technology*.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., & Bray, C. A. (2004). Visual categorization with bags of keypoints. In *In workshop on statistical learning in computer vision, eccv* (pp. 1–22). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.604

Dougherty, G. (2013). Pattern recognition and classification. , 14. Retrieved from http://dx.doi.org/10.1007/978-1-4614-5323-9 doi: 10.1007/978-1-4614-5323-9

Grauman, K., & Leibe, B. (2011). Visual object recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *5*(2), 1-181. Retrieved from http://dx.doi.org/10.2200/S00332ED1V01Y201103AIM011 doi: 10.2200/S00332ED1V01Y201103AIM011

Krzanowski, W. J. (1988). *Principles of multivariate analysis: A user's perspective.* Oxford University Press.

Li, T., Zhu, S., & Ogihara, M. (2006). Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge and Information Systems*, *10*(4), 453-472. Retrieved from http://dx.doi.org/10.1007/s10115-006-0013-y doi: 10.1007/s10115-006-0013-y

Lindeberg, T. (2012). Scale invariant feature transform. *Knowledge and Information Systems*, *7*(5), 10491-.

Marsland, S. (2009). *Machine learning: An algorithmic perspective.* Chapman & Hall/CRC.

Meitzler, T., Gerhart, G., & Singh, H. (1998, Jul). A relative clutter metric. *Aerospace and Electronic Systems, IEEE Transactions on*, *34*(3), 968-976. doi: 10.1109/7.705903

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective.* The MIT Press.

Rahman, Z.-u., & Jobson, D. J. (2003). Information theoretic analysis of noise sources in image formation. *Proc. SPIE 5108*, *Visual Information Processing XII*(39).

Reynolds, W. R. (1990). *Toward quantifying infrared clutter* (Vol. 1311). Retrieved from http://dx.doi.org/10.1117/12.21836 doi: 10.1117/12.21836

Salem, S., Halford, C., Moyer, S., & Gundy, M. (2009). Rotational clutter metric. *Optical Engineering*, *48*(8), 086401-086401-11. Retrieved from http://dx.doi.org/10.1117/1.3204234 doi: 10.1117/1.3204234

Schmieder, D., & Weathersby, M. (1983, July). Detection performance in clutter with variable resolution. *Aerospace and Electronic Systems, IEEE Transactions on*, *AES-19*(4), 622-630. doi: 10.1109/TAES.1983.309351

Sutherland, R. A., Montoya, J. R., & Thompson, J. C. (2002). *Atmospheric effects on visible and infrared scene clutter characerization* (Vol. 4718). Retrieved from http://dx.doi.org/10.1117/12.478810 doi: 10.1117/12.478810

Szeliski, R. (2011). Computer vision algorithms and applications. , 3. Retrieved from http://dx.doi.org/10.1007/978-1-84882-935-0 doi: 10.1007/978-1-84882-935-0

Wang, X., & Zhang, T. (2011). Clutter-adaptive infrared small target detection in infrared maritime scenarios. *Optical Engineering*, *50*(6), 067001-067001-12. Retrieved from http://dx.doi.org/10.1117/1.3582855 doi: 10.1117/1.3582855

Yoon, S., Song, T., & Kim, T. (2013). Automatic target recognition and tracking in forward-looking infrared image sequences with a complex background. *International Journal of Control, Automation and Systems*, *11*(1), 21-32. Retrieved from `http://dx.doi.org/10.1007/s12555-011-0226-z` doi: 10.1007/s12555-011-0226-z

Young, I. T., Gerbrands, J. J., & van Viliet, L. J. (2007). *Fundamentals of image processing.*

7 Appendix

The following section describes the data set used in this study in greater detail. Figures 15-20 show geographical maps where the coordinates of all 900 images are marked. Figures 21-29 show examples of images from each class. Tables 4-12 list the coordinates of all 900 images by class. This information allows for the exact replication of methodology, but the exact same satellite images may not necessarily be accessible in the future.

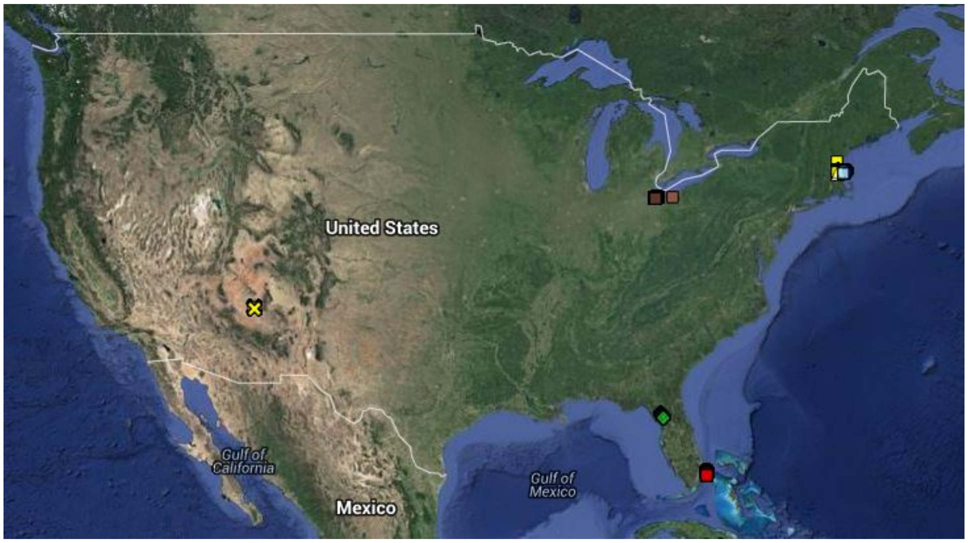


Figure 15: Map of the United States of America where markers correspond to the coordinates of the 900 satellite images used in this study.

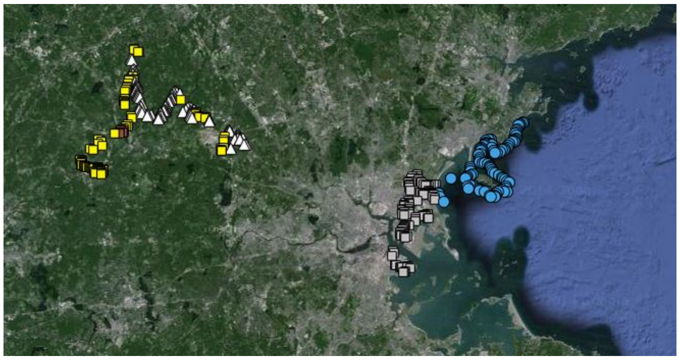


Figure 16: Geographical map of the Boston, Massachusetts area where markers correspond to the coordinates of satellite images of the "urban," (white squares) "suburban," (white triangles) "forest," (yellow squares) and "sea" (blue circles) classes. Six brown square markers also belong to the "rural" class.

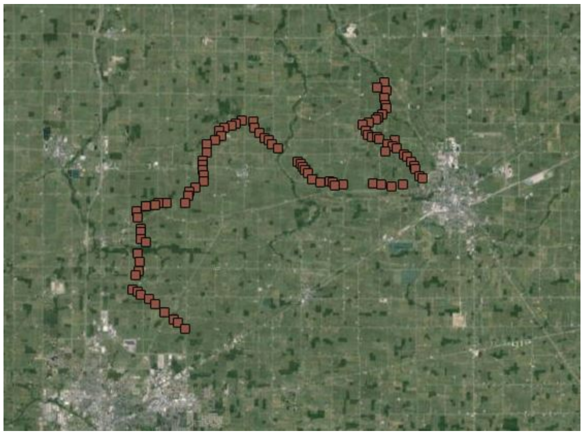


Figure 17: Geographical map of Findlay, Ohio area where markers correspond to the coordinates of satellite images of the “rural” class.

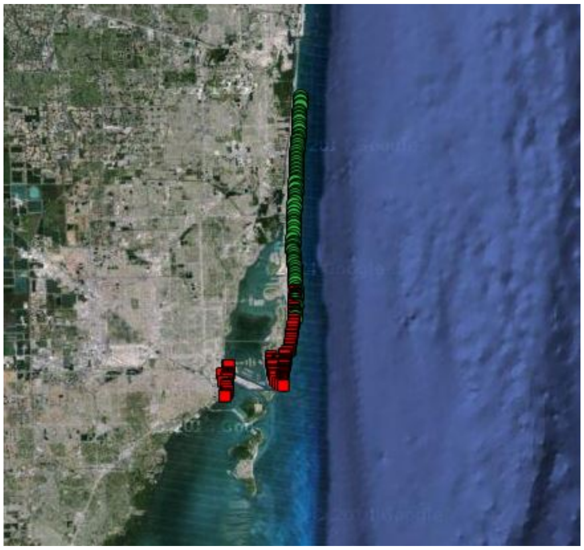


Figure 18: Geographical map of Miami, Florida area where markers correspond to the coordinates of satellite images of the “coast” (green circles) and “metropolitan” (red squares) classes.

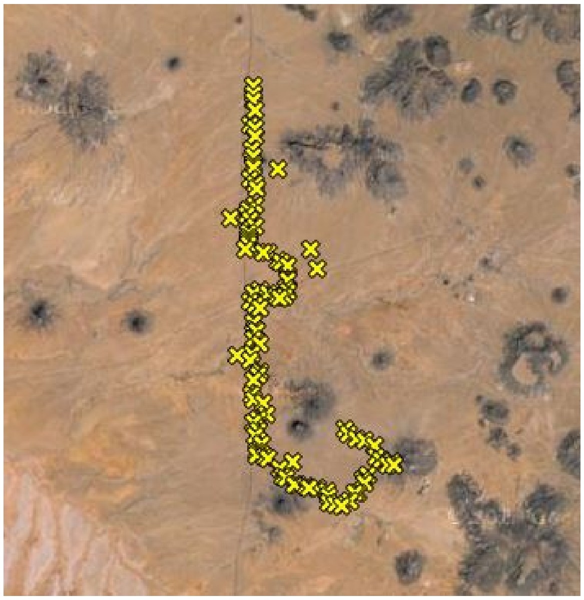


Figure 19: Geographical map of a patch of desert in Arizona where markers correspond to the coordinates of satellite images of the “desert” class.

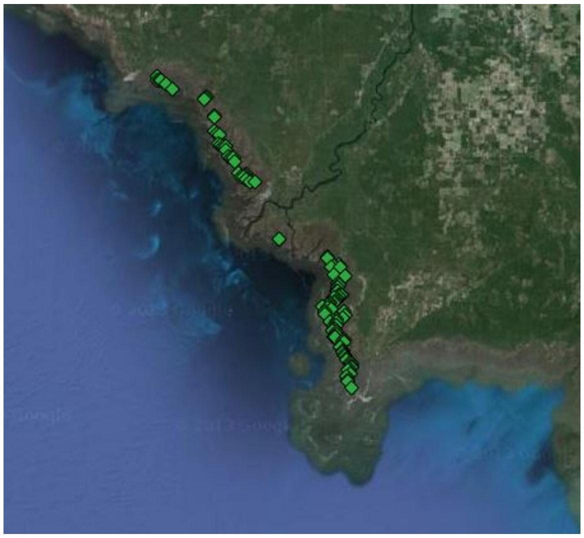


Figure 20: Geographical map of the Big Bend Seagrasses Aquatic Preserve located in Florida where markers correspond to the coordinates of satellite images of the “preserve” class.

Figure 21: Example of image classified as “sea” and “low clutter” from New England.

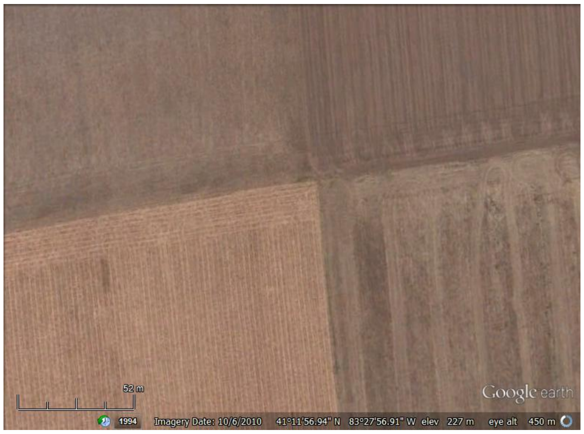


Figure 22: Example of image classified as “rural” and “low clutter” from Illinois.

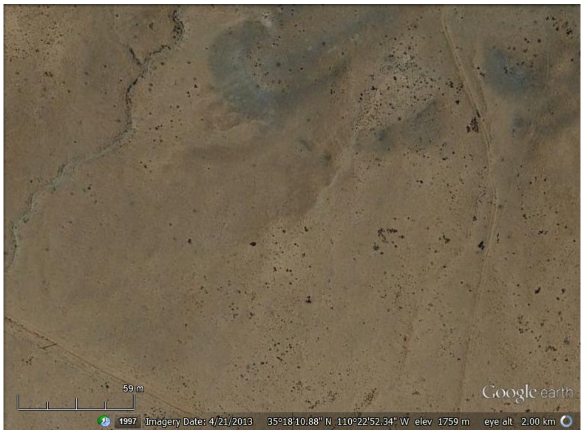


Figure 23: Example of image classified as “desert” and “low clutter” from Arizona.

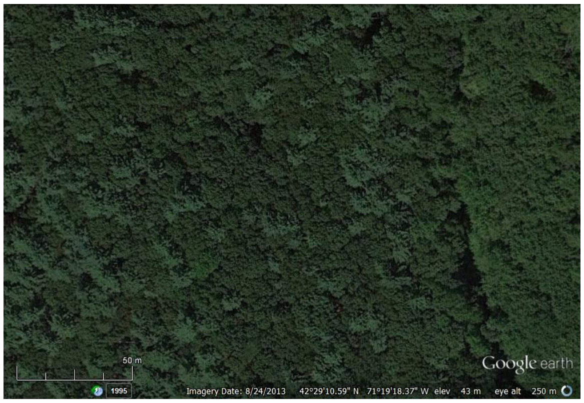

50 m
1995
Imagery Date: 8/24/2013 42°29'10.59" N 71°19'18.37" W elev 43 m eye alt 250 m
Google earth

Figure 24: Example of image classified as “forest” and “low clutter” from Massachusetts.

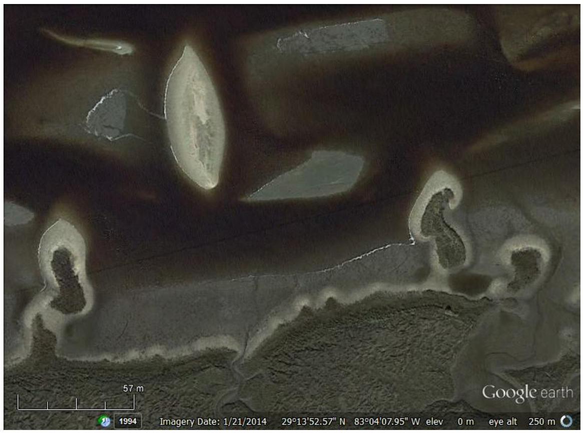

57 m
1994
Imagery Date: 1/21/2014 29°13'52.57" N 83°04'07.95" W elev 0 m eye alt 250 m
Google earth

Figure 25: Example of image classified as “preserve” and “low clutter” from the Big Bend Seagrasses Aquatic Preserve located in Florida.

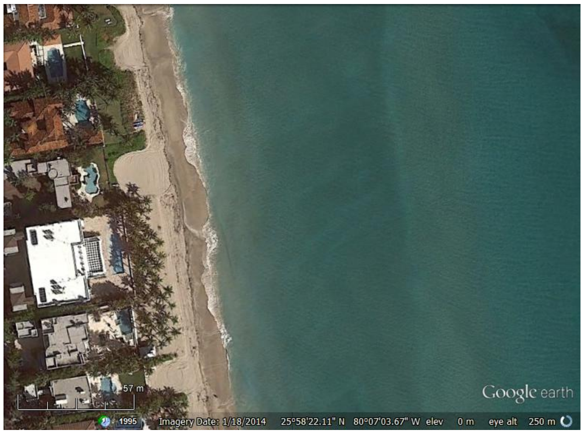

57 m
1995
Imagery Date: 1/18/2014 25°58'22.11" N 80°07'03.67" W elev 0 m eye alt 250 m
Google earth

Figure 26: Example of image classified as “coast” and “high clutter” from Miami Beach, Florida.

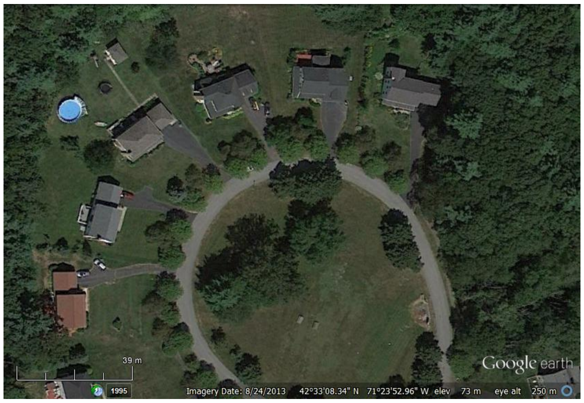

39 m
1995
Imagery Date: 8/24/2013 42°33'08.34" N 71°23'52.96" W elev 73 m eye alt 250 m
Google earth

Figure 27: Example of image classified as "suburban" and "high clutter" from Massachusetts.

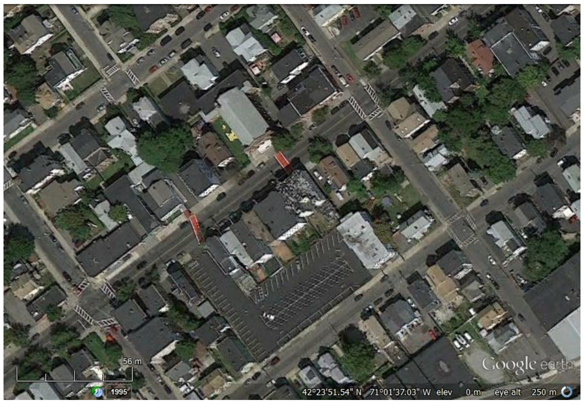


Figure 28: Example of image classified as "urban" and "high clutter" from Boston, Massachusetts.

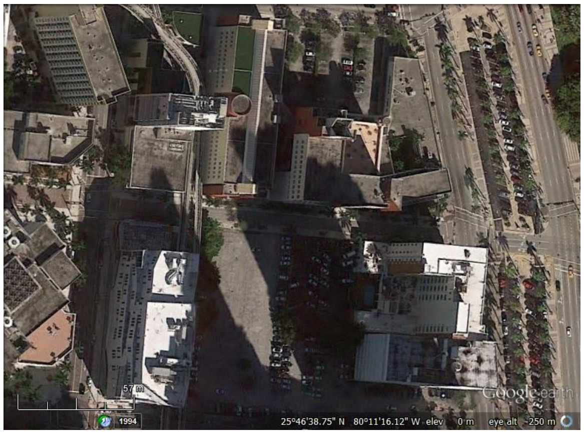


Figure 29: Example of image classified as "metropolitana" and "high clutter" from Miami, Florida.

Table 4: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as "Sea" in decimal.

Sea					
X	Y	X	Y	X	Y
42.47959	-70.8733	42.46113	-70.923	42.41031	-70.8982
42.47858	-70.8739	42.4591	-70.9234	42.4122	-70.9016
42.47739	-70.8764	42.45776	-70.9238	42.41101	-70.9046
42.47687	-70.878	42.45648	-70.9247	42.41062	-70.9067
42.47586	-70.8799	42.45604	-70.9257	42.40871	-70.9093
42.47195	-70.8805	42.45389	-70.9296	42.40618	-70.9106
42.47079	-70.8821	42.45243	-70.9304	42.40472	-70.9106
42.46923	-70.883	42.45144	-70.9297	42.40529	-70.9168
42.46835	-70.8832	42.45034	-70.9299	42.40612	-70.9178
42.4657	-70.8828	42.44776	-70.931	42.40985	-70.9225
42.46306	-70.8829	42.44696	-70.9329	42.41064	-70.9256
42.4617	-70.8842	42.42353	-70.9706	42.4114	-70.9285
42.46061	-70.8846	42.44357	-70.9322	42.41173	-70.9318
42.45947	-70.8856	42.44143	-70.9321	42.41226	-70.9324
42.45945	-70.8884	42.43977	-70.9314	42.41378	-70.9339
42.45768	-70.8923	42.43887	-70.9281	42.41644	-70.9396
42.4557	-70.8933	42.43877	-70.9254	42.41659	-70.9417
42.45395	-70.8947	42.43799	-70.9216	42.41686	-70.9441
42.45299	-70.8972	42.43584	-70.9182	42.41529	-70.9481
42.45204	-70.8995	42.43509	-70.9182	42.41335	-70.9468
42.45162	-70.901	42.43383	-70.9173	42.42351	-70.9506
42.45026	-70.9038	42.43122	-70.915	42.42428	-70.9497
42.44898	-70.9081	42.42923	-70.9134	42.42523	-70.949
42.44838	-70.9108	42.42822	-70.9116	42.42652	-70.9448
42.45087	-70.9126	42.42656	-70.9061	42.42771	-70.9413
42.45112	-70.9132	42.42604	-70.9056	42.43049	-70.9447
42.45424	-70.9148	42.42406	-70.901	42.43284	-70.9448
42.45373	-70.915	42.42356	-70.9	42.43488	-70.9443
42.4565	-70.9164	42.42211	-70.8971	42.40876	-70.9871
42.45698	-70.9156	42.42063	-70.8952	42.40811	-70.9871
42.45809	-70.9163	42.41892	-70.8926	42.40481	-70.9858
42.46243	-70.9168	42.41511	-70.8956	42.39993	-70.9806
42.46217	-70.917	42.41316	-70.8967		
42.46246	-70.9195	42.4107	-70.895		

Table 5: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as "Rural" in decimal.

Rural					
X	Y	X	Y	X	Y
42.47097	-71.4118	41.16857	-83.4409	41.17409	-83.5856
42.47203	-71.4139	41.16732	-83.4388	41.17124	-83.5855
42.47122	-71.4154	41.1642	-83.4523	41.16896	-83.5847
42.46914	-71.4118	41.16339	-83.4599	41.16594	-83.5844
42.46909	-71.4129	41.16404	-83.4665	41.16239	-83.5899
42.4688	-71.4161	41.1647	-83.472	41.16077	-83.594
42.46804	-71.4169	41.16424	-83.4917	41.15871	-83.5951
41.21296	-83.4695	41.16391	-83.4974	41.15479	-83.597
41.21544	-83.4644	41.16576	-83.501	41.15512	-83.6091
41.21096	-83.4635	41.16511	-83.4991	41.15495	-83.6148
41.20771	-83.4646	41.16542	-83.5068	41.154	-83.6171
41.20395	-83.4636	41.16694	-83.5127	41.15357	-83.623
41.20248	-83.4632	41.16994	-83.515	41.15334	-83.6229
41.19915	-83.4658	41.17163	-83.5174	41.15088	-83.6292
41.19785	-83.4684	41.17288	-83.5187	41.14742	-83.6285
41.1969	-83.4701	41.17573	-83.5223	41.14218	-83.6265
41.19786	-83.469	41.17471	-83.5204	41.14045	-83.6264
41.19493	-83.4751	41.18239	-83.535	41.13662	-83.6267
41.19407	-83.4789	41.18432	-83.5394	41.13527	-83.6232
41.19108	-83.4769	41.18627	-83.5417	41.1294	-83.6285
41.18899	-83.4752	41.18718	-83.5442	41.12532	-83.6271
41.18802	-83.4725	41.19016	-83.5478	41.12094	-83.628
41.18671	-83.4678	41.19279	-83.5511	41.11851	-83.63
41.18771	-83.4692	41.19571	-83.5515	41.11162	-83.632
41.18437	-83.4625	41.19599	-83.5592	41.11043	-83.6285
41.18628	-83.4579	41.19475	-83.5636	41.10906	-83.6267
41.18308	-83.4592	41.19385	-83.566	41.10675	-83.6208
41.18209	-83.4563	41.19235	-83.5707	41.10444	-83.6166
41.18052	-83.4639	41.19093	-83.5743	41.10022	-83.6104
41.17937	-83.4513	41.18851	-83.5731	41.0966	-83.6044
41.17833	-83.4496	41.18625	-83.5769	41.09478	-83.6018
41.17569	-83.447	41.18416	-83.5827	41.09159	-83.5972
41.17354	-83.4446	41.18007	-83.5829		
41.17169	-83.4436	41.17629	-83.5852		

Table 6: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as “Desert” in decimal.

Desert					
X	Y	X	Y	X	Y
35.324728	-110.38623	35.313347	-110.40821	35.381533	-110.4131
35.322786	-110.38531	35.322439	-110.42456	35.382186	-110.41488
35.321442	-110.38164	35.324053	-110.42316	35.379369	-110.39804
35.320506	-110.37805	35.327094	-110.42488	35.385278	-110.41876
35.319044	-110.37393	35.327833	-110.42279	35.385308	-110.421
35.315064	-110.37343	35.329542	-110.41968	35.386194	-110.40119
35.313906	-110.37164	35.333717	-110.42111	35.386194	-110.42815
35.312353	-110.36871	35.334914	-110.42488	35.387847	-110.42722
35.313633	-110.36682	35.336592	-110.42552	35.390219	-110.42668
35.31215	-110.36594	35.341206	-110.42526	35.392081	-110.42678
35.309153	-110.37715	35.343175	-110.42314	35.394236	-110.4245
35.307364	-110.37879	35.344364	-110.42231	35.396422	-110.42663
35.306117	-110.37829	35.348325	-110.4261	35.397858	-110.428
35.303022	-110.38121	35.350244	-110.42537	35.399564	-110.42704
35.302742	-110.38227	35.353539	-110.42233	35.401372	-110.42423
35.299772	-110.38443	35.355619	-110.42463	35.388875	-110.42703
35.298489	-110.38798	35.358253	-110.42361	35.406789	-110.42373
35.298867	-110.3907	35.358969	-110.42395	35.408903	-110.42412
35.299619	-110.39312	35.349711	-110.43183	35.410172	-110.42619
35.301403	-110.39351	35.360825	-110.42499	35.396781	-110.43488
35.3036	-110.39381	35.367442	-110.42456	35.414486	-110.42525
35.304569	-110.39792	35.365981	-110.422	35.415611	-110.42468
35.30465	-110.40173	35.368311	-110.42631	35.418294	-110.42467
35.304925	-110.40328	35.371028	-110.4249	35.420428	-110.42531
35.305322	-110.40649	35.372128	-110.42218	35.413697	-110.41435
35.3058	-110.40759	35.371033	-110.41886	35.42495	-110.42438
35.307744	-110.41102	35.369864	-110.41561	35.427831	-110.42432
35.308639	-110.41247	35.369708	-110.41414	35.428914	-110.42599
35.310936	-110.4143	35.370067	-110.41205	35.434053	-110.42419
35.314408	-110.41889	35.371492	-110.41022	35.437392	-110.42518
35.314889	-110.42042	35.372858	-110.4109	35.439511	-110.42493
35.315783	-110.4234	35.375569	-110.41072	35.442275	-110.42497
35.316681	-110.42327	35.376389	-110.4103		
35.319508	-110.42135	35.380414	-110.41035		

Table 7: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as “Forest” in decimal.

Forest					
X	Y	X	Y	X	Y
42.47159	-71.2711	42.49286	-71.3207	42.48644	-71.3743
42.47104	-71.2702	42.49411	-71.3219	42.48635	-71.3779
42.47064	-71.2696	42.49347	-71.3249	42.48556	-71.3794
42.47051	-71.2689	42.49148	-71.3269	42.48853	-71.3862
42.47037	-71.2683	42.48953	-71.331	42.48931	-71.3879
42.47012	-71.2679	42.50782	-71.3352	42.49111	-71.389
42.46812	-71.258	42.50927	-71.3392	42.49162	-71.3899
42.46634	-71.2581	42.50863	-71.3403	42.49409	-71.3878
42.46553	-71.2579	42.50995	-71.3427	42.49477	-71.388
42.46485	-71.2577	42.50907	-71.3428	42.49541	-71.3885
42.46291	-71.2574	42.50841	-71.3445	42.49759	-71.386
42.46298	-71.2579	42.50673	-71.3457	42.50374	-71.3887
42.46157	-71.2612	42.50534	-71.346	42.50405	-71.3904
42.46184	-71.255	42.50397	-71.3461	42.50506	-71.3918
42.46194	-71.2545	42.50252	-71.3468	42.50621	-71.393
42.46271	-71.2535	42.50144	-71.3487	42.50691	-71.3924
42.46291	-71.2519	42.50136	-71.3497	42.50903	-71.3934
42.46306	-71.2508	42.50028	-71.3504	42.51014	-71.3941
42.45693	-71.2471	42.49901	-71.3506	42.51175	-71.3942
42.45661	-71.2467	42.49688	-71.3529	42.51306	-71.3931
42.45673	-71.2484	42.49571	-71.3542	42.51374	-71.3926
42.4526	-71.2675	42.49451	-71.3543	42.51486	-71.393
42.47973	-71.2958	42.49417	-71.3542	42.51528	-71.3929
42.48074	-71.2956	42.49156	-71.3548	42.5175	-71.3933
42.48151	-71.2952	42.49021	-71.3555	42.51843	-71.3958
42.49164	-71.3158	42.49027	-71.3567	42.52293	-71.3986
42.48329	-71.3191	42.49031	-71.3576	42.52719	-71.3984
42.48497	-71.3207	42.48929	-71.3613	42.52755	-71.3992
42.48628	-71.3218	42.48823	-71.3612	42.53042	-71.3983
42.48679	-71.3219	42.48744	-71.3607	42.53189	-71.398
42.48711	-71.3211	42.48562	-71.3604	42.54119	-71.4
42.48959	-71.3214	42.48425	-71.3637	42.5415	-71.4017
42.48934	-71.3211	42.48301	-71.3639		
42.49034	-71.3203	42.48264	-71.3646		

Table 8: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as "Preserve" in decimal.

Preserve					
X	Y	X	Y	X	Y
29.14474	-83.0555	29.22524	-83.0746	29.34696	-83.1781
29.15073	-83.0591	29.22674	-83.0727	29.34434	-83.1746
29.15225	-83.0597	29.23054	-83.0713	29.34348	-83.1725
29.15556	-83.0607	29.23127	-83.0689	29.34516	-83.1705
29.15819	-83.0567	29.23222	-83.0661	29.34671	-83.1693
29.16173	-83.0598	29.24249	-83.0696	29.34128	-83.1687
29.16326	-83.0576	29.252	-83.0739	29.33937	-83.1653
29.16425	-83.0548	29.25858	-83.077	29.33934	-83.16
29.16493	-83.0547	29.26017	-83.0746	29.28493	-83.1332
29.16757	-83.0556	29.26743	-83.082	29.26101	-83.0695
29.16919	-83.0566	29.26873	-83.0818	29.25908	-83.0675
29.17088	-83.0604	29.43854	-83.2658	29.25041	-83.0624
29.17198	-83.0614	29.43503	-83.2652	29.24276	-83.0701
29.17491	-83.0627	29.43532	-83.2628	29.23963	-83.0707
29.17602	-83.0636	29.4345	-83.2615	29.23636	-83.0713
29.17806	-83.0651	29.43161	-83.2559	29.23664	-83.07
29.1792	-83.0651	29.42684	-83.2484	29.23241	-83.0685
29.1837	-83.0678	29.41668	-83.2147	29.23231	-83.0665
29.18757	-83.0638	29.41812	-83.2136	29.22968	-83.0743
29.18937	-83.065	29.4196	-83.2115	29.22493	-83.0787
29.19058	-83.0637	29.40125	-83.2033	29.22268	-83.0808
29.19266	-83.0728	29.40063	-83.2035	29.22058	-83.0807
29.19279	-83.0739	29.38666	-83.2029	29.21971	-83.0777
29.19582	-83.0721	29.38415	-83.2008	29.21717	-83.0636
29.19773	-83.0712	29.38307	-83.1978	29.21296	-83.0623
29.20029	-83.0749	29.37492	-83.1988	29.21038	-83.0652
29.20329	-83.0748	29.37391	-83.1963	29.20966	-83.0718
29.20713	-83.0788	29.37296	-83.1937	29.20745	-83.0712
29.2073	-83.078	29.37234	-83.1913	29.20319	-83.072
29.20799	-83.0755	29.36862	-83.1907	29.20051	-83.0732
29.21477	-83.0846	29.36331	-83.1886	29.1984	-83.0755
29.21724	-83.0844	29.36194	-83.1858	29.19343	-83.0732
29.22166	-83.0868	29.36181	-83.1825		
29.22429	-83.0797	29.35878	-83.1822		

Table 9: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as "Coast" in decimal.

Coast					
X	Y	X	Y	X	Y
26.036194	-80.113642	25.966689	-80.117781	25.894122	-80.121261
26.033925	-80.113606	25.965614	-80.118453	25.892011	-80.12135
26.031569	-80.11365	25.962986	-80.117978	25.890619	-80.121097
26.02825	-80.113858	25.962344	-80.118467	25.887506	-80.120603
26.027061	-80.114136	25.959039	-80.117972	25.885772	-80.120564
26.022367	-80.1144	25.958639	-80.118367	25.883703	-80.120433
26.020953	-80.114503	25.956564	-80.118533	25.883108	-80.120619
25.827856	-80.11935	25.953136	-80.118397	25.879719	-80.119894
26.01865	-80.114675	25.951294	-80.118419	25.877594	-80.119647
26.017075	-80.114775	25.950633	-80.118956	25.875369	-80.119533
26.015258	-80.114922	25.947278	-80.118481	25.873203	-80.119408
26.013389	-80.115075	25.945408	-80.1185	25.871069	-80.119117
26.010483	-80.115464	25.944778	-80.119044	25.869206	-80.118922
26.007997	-80.115336	25.941647	-80.118858	25.866981	-80.118714
26.006942	-80.115428	25.939472	-80.118992	25.864858	-80.118694
26.004856	-80.115589	25.938897	-80.119683	25.862653	-80.118611
26.001344	-80.115261	25.935233	-80.120214	25.877219	-80.118508
26.000536	-80.115697	25.931664	-80.119822	25.859156	-80.118806
25.997661	-80.115517	25.929719	-80.120306	25.85615	-80.118386
25.996069	-80.115933	25.926122	-80.120256	25.854014	-80.118481
25.994847	-80.116503	25.925431	-80.120428	25.852211	-80.118339
25.992417	-80.116761	25.922367	-80.12045	25.8503	-80.118314
25.991536	-80.116622	25.920264	-80.120503	25.848611	-80.118469
25.989333	-80.116942	25.918347	-80.120578	25.84725	-80.118489
25.986817	-80.116769	25.916508	-80.120772	25.844689	-80.118647
25.985817	-80.117508	25.915767	-80.120989	25.842228	-80.118525
25.982742	-80.117169	25.91245	-80.120647	25.840111	-80.118406
25.980589	-80.117358	25.909686	-80.120603	25.839467	-80.118808
25.978553	-80.117431	25.905961	-80.120775	25.835919	-80.118761
25.976583	-80.117697	25.904119	-80.121344	25.834314	-80.118767
25.97475	-80.117769	25.900494	-80.121192	25.833797	-80.119303
25.972808	-80.117686	25.898497	-80.120714	25.83055	-80.11905
25.970833	-80.117764	25.897244	-80.121614		
25.968631	-80.117886	25.894983	-80.121311		

Table 10: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as "Suburban" in decimal.

Suburban					
X	Y	X	Y	X	Y
42.45068	-71.2798	42.50228	-71.4092	42.42985	-71.4505
42.45105	-71.2794	42.50199	-71.4094	42.43093	-71.4496
42.4519	-71.2794	42.5007	-71.4108	42.43343	-71.4551
42.45188	-71.2803	42.49966	-71.411	42.43454	-71.46
42.45359	-71.2794	42.4986	-71.4103	42.435	-71.4606
42.45396	-71.2769	42.49714	-71.4094	42.43567	-71.4629
42.46032	-71.2769	42.4963	-71.4098	42.43594	-71.4652
42.46189	-71.2772	42.4855	-71.4016	42.43641	-71.466
42.46512	-71.2753	42.48374	-71.4002	42.43667	-71.4689
42.48282	-71.3016	42.47756	-71.4051	42.43823	-71.4687
42.48388	-71.3004	42.47563	-71.406	42.43847	-71.4682
42.48452	-71.3025	42.47531	-71.4079	42.43933	-71.4671
42.48398	-71.307	42.47334	-71.4109	42.44003	-71.4669
42.48434	-71.3094	42.46786	-71.4236	42.4407	-71.4678
42.48492	-71.3114	42.46013	-71.4408	42.44169	-71.468
42.48737	-71.315	42.46022	-71.4411	42.44195	-71.4695
42.4884	-71.3145	42.46155	-71.4458	42.43631	-71.4673
42.48954	-71.3141	42.4578	-71.4454	42.4358	-71.4647
42.50178	-71.3341	42.45637	-71.4397	42.43561	-71.4638
42.52041	-71.3967	42.45329	-71.456	42.43516	-71.4617
42.52142	-71.3969	42.45189	-71.4529	42.43478	-71.4599
42.52474	-71.3985	42.43462	-71.4451	42.43439	-71.458
42.54861	-71.3902	42.43382	-71.4453	42.43416	-71.4564
42.54855	-71.3913	42.43303	-71.4452	42.43383	-71.4539
42.55103	-71.3967	42.43123	-71.4453	42.43348	-71.452
42.55232	-71.398	42.43054	-71.4444	42.43392	-71.4505
42.52189	-71.4046	42.43255	-71.439	42.43333	-71.4471
42.52032	-71.4049	42.43112	-71.4387	42.43331	-71.4449
42.51796	-71.4054	42.42976	-71.4388	42.43236	-71.4439
42.51481	-71.4075	42.42849	-71.4397	42.43125	-71.4422
42.51349	-71.4064	42.42764	-71.4417	42.43008	-71.4423
42.51188	-71.4068	42.42814	-71.4448	42.42872	-71.4411
42.51126	-71.4087	42.42835	-71.4472		
42.50965	-71.4097	42.42843	-71.4494		

Table 11: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as “Urban” in decimal.

Urban					
X	Y	X	Y	X	Y
42.40905	-70.9919	42.40839	-71.0089	42.39765	-71.027
42.40889	-70.9944	42.40716	-71.0079	42.39494	-71.0271
42.40917	-71.0007	42.40849	-71.0062	42.39295	-71.0274
42.40732	-70.9999	42.41083	-71.0008	42.38449	-71.0187
42.40723	-70.9981	42.41018	-70.9993	42.38326	-71.0182
42.40564	-70.9947	42.40857	-70.9962	42.38098	-71.0179
42.40423	-70.9927	42.40813	-70.9954	42.37717	-71.0269
42.40229	-70.9924	42.40718	-70.9934	42.37684	-71.0316
42.40128	-71.0154	42.38566	-71.0053	42.37529	-71.0335
42.41776	-70.9908	42.38599	-71.0086	42.37381	-71.0351
42.41759	-70.993	42.38678	-71.0077	42.373	-71.036
42.41719	-70.9976	42.38711	-71.0052	42.37086	-71.0379
42.41652	-71.001	42.38765	-71.002	42.36921	-71.0384
42.41677	-71.0039	42.38816	-71.0013	42.36759	-71.0387
42.41667	-71.0049	42.38403	-71.002	42.36671	-71.0369
42.41778	-71.0068	42.38816	-71.0013	42.36567	-71.0334
42.41899	-71.0096	42.38403	-71.0004	42.36417	-71.0327
42.41995	-71.0111	42.37446	-71.0331	42.36577	-71.0291
42.4223	-71.0138	42.39291	-71.0305	42.35123	-71.0484
42.42332	-71.0166	42.39103	-71.0276	42.34979	-71.0472
42.42538	-71.0161	42.38992	-71.0272	42.3461	-71.0502
42.42708	-71.0156	42.3879	-71.0271	42.33713	-71.0454
42.4251	-71.0225	42.38706	-71.0337	42.33681	-71.0444
42.42299	-71.0233	42.3882	-71.0359	42.33491	-71.0388
42.42082	-71.0239	42.38997	-71.0348	42.33469	-71.0371
42.42039	-71.0269	42.39041	-71.0342	42.33525	-71.0364
42.41912	-71.0275	42.39212	-71.036	42.33569	-71.0299
42.41748	-71.0271	42.39299	-71.0356	42.33637	-71.0284
42.41594	-71.0267	42.39618	-71.034	42.33433	-71.0255
42.41597	-71.0266	42.39673	-71.0332	42.33123	-71.0316
42.41461	-71.0262	42.39743	-71.0332	42.33062	-71.0339
42.41393	-71.0265	42.39915	-71.0325	42.32966	-71.0353
42.41169	-71.0258	42.4008	-71.0286		
42.40926	-71.0114	42.39872	-71.0267		

Table 12: Table listing the latitude (X), longitude (Y) coordinates of each image labeled as "Metropolitan" in decimal.

Metropolitan					
X	Y	X	Y	X	Y
25.75666	-80.1917	25.78995	-80.1296	25.79635	-80.1306
25.75824	-80.1917	25.79125	-80.1296	25.79472	-80.1305
25.7597	-80.1903	25.794653	-80.1282	25.7908	-80.1298
25.76118	-80.1897	25.795936	-80.1288	25.7885	-80.1303
25.76404	-80.1901	25.798872	-80.1267	25.78478	-80.1315
25.7659	-80.1901	25.800375	-80.1242	25.78306	-80.131
25.76657	-80.1903	25.802417	-80.125	25.78065	-80.1318
25.76781	-80.1891	25.803889	-80.1242	25.77928	-80.1332
25.7696	-80.1907	25.80665	-80.124	25.77854	-80.1335
25.7701	-80.1934	25.808492	-80.1235	25.77593	-80.1334
25.77216	-80.1898	25.810547	-80.1233	25.77442	-80.1332
25.77386	-80.1903	25.812419	-80.123	25.77283	-80.1329
25.77469	-80.1896	25.814492	-80.1227	25.77123	-80.1331
25.77687	-80.1887	25.816447	-80.1224	25.76921	-80.134
25.77743	-80.1878	25.818286	-80.1222	25.76952	-80.1375
25.77824	-80.1921	25.822086	-80.1215	25.77203	-80.1396
25.77785	-80.1952	25.824097	-80.1207	25.7749	-80.14
25.77944	-80.1887	25.8261	-80.1205	25.77696	-80.1414
25.78491	-80.1895	25.827803	-80.1204	25.77832	-80.1417
25.78562	-80.1874	25.831956	-80.1206	25.78015	-80.1424
25.76732	-80.1343	25.833794	-80.1202	25.78161	-80.142
25.76709	-80.132	25.835619	-80.1201	25.78517	-80.1424
25.77032	-80.1323	25.836783	-80.1202	25.78626	-80.1426
25.77221	-80.1328	25.839969	-80.1206	25.78814	-80.1436
25.77449	-80.1328	25.84205	-80.1206	25.78973	-80.1439
25.77497	-80.1324	25.843478	-80.12	25.79071	-80.1438
25.77789	-80.1321	25.845194	-80.1196	25.79357	-80.144
25.7797	-80.1321	25.848083	-80.1198	25.79007	-80.1371
25.78119	-80.1315	25.849211	-80.1199	25.7901	-80.1358
25.78299	-80.131	25.851033	-80.1198	25.78971	-80.1343
25.78471	-80.1306	25.852294	-80.1194	25.78779	-80.1309
25.78674	-80.1304	25.854531	-80.1195	25.78624	-80.1298
25.78799	-80.1298	25.855839	-80.1196		
25.78944	-80.1292	25.799017	-80.1281		