International Journal of Engineering & Technology, 7 (4.6) (2018) 498-504

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Aerial Scene Classification by Surf Feature Extarction Using Unsupervised Learning

S.G.Hymlin Rose [1],T.D.Subha [2]

ASSISTANT PROFESSOR, St.Joseph College of Engineering, Sriperumbudur, Chennai, INDIA

Abstract

The high-resolution satellite imagery which consisted of rich data allow us to directly model aerial scenes by understanding their spatial and structural patterns. Efficient representation and recognition of scenes from image data are challenging. For satellite image analysis, pixel and object based classification approaches are widely used but these approaches often exploit the high-fidelity image data only in a limited way. In this paper, we explore a supervised feature learning approach for aerial scene classification. This system follows some peculiar steps like Noise Removal, Feature Extraction (SURF), Feature Learning and Classification. SURF features are extracted from the input image and classification is done by Latent Dirichlet Allocation Algorithm. This technique can be applied to several challenging aerial scene data sets: ORNL-I data set consisting of 1-m spatial resolution satellite imagery, UCMERCED data set with sub-meter resolution, and ORNL-II data set for large-facility scene detection. The proposed method is highly effective in developing a detection system that can be used to automatically scan large-scale high-resolution satellite imagery for detecting large facilities such as a shopping mall.

Keywords: *aerial, SVM, unsupervised,etc.*

1. Introduction

Amateur aerial photography, the casual taking of photographs from a moving aircraft, involves challenges that differ from standard ground-based amateur photography. It also obviously provides some unique advantages.

The vast majority of aerial photos are taken by amateurs in private aviation small fixed-wing aircraft, or in airliners. To obtain high-quality pictures in what can be a more difficult but highly rewarding environment for photography, the emphasis is on candid snapshots from light single-engine aircraft (small planes), in which the camera is subject to wind, strong vibration and other factors not common in professional settings or airliners. Photographic subjects are often chosen spontaneously in amateur aerial photography and exact advance selection of the camera location is all but impossible; while in professional work the subject and location are usually known and planned in advance for fine adjustment and shooting - often from a helicopter.

Most of these pointers also apply to planned aerial photos from small planes. The complexity is in the position of the plane. Often photos are a combination of candid and planned. For example an interesting view is noticed during a flight, and a candid snapshot is taken. Sometimes a better photo is desired or spontaneously imagined, so the small plane is maneuvered to capture a planned perspective or different angle or exposure.

Amateur camera equipment is typically more affected by the drawbacks of using small planes as a photographic platform. An important consideration for amateur photography from small aircraft is the ability to open a window, since even new aircraft windows introduce significant blue haze to the photos that is not noticed by the eye during flight. Major advantages to amateur aerial photography from small planes vs. airliners include: lower altitude; slower flight; often the ability to open a window; having some control over camera location and angle by means of maneuvering the aircraft; and discovering subjects, events, lighting or perspectives that could have been impossible to plan in advance.

Figure 1.1: Aerial Scenes

2. Literature Review

Aaron K. Shackelford, et.al [7] deals with object-based approach for urban land cover classification from high-resolution multispectral image data that builds upon a pixel-based fuzzy classification

approach. This combined pixel/object approach is demonstrated using pan-sharpened multispectral IKONOS imagery from dense urban areas. The fuzzy pixel-based classifier utilizes both spectral and spatial information to discriminate between spectrally similar Road and Building urban land cover classes. After the pixel-based classification, a technique that utilizes both spectral and spatial heterogeneity is used to segment the image to facilitate further object-based classification. An object-based fuzzy logic classifier is then implemented to improve upon the pixel-based classification by identifying one additional class in dense urban areas: non-road, non-building impervious surface. With the fuzzy pixel-based classification as input, the object-based classifier then uses shape, spectral, and neighborhood features to determine the final classification of the segmented image.

The main advantage of this paper is using these techniques, the object- based classifier is able to identify Buildings, Impervious Surface, and Roads in dense urban areas with 76%, 81%, and 99% classification accuracies, respectively. The main disadvantage of this paper is that it does not require intermediate stages of segmentation.

Chih-Wei Hsu et.al[8]deals with Support vector machines (SVMs) were originally designed for binary classification. Several methods have been proposed where typically they construct a multiclass classifier by combining several binary classifiers. Some authors also proposed methods that consider all classes at once. As it is computationally more expensive to solve multiclass problems, comparisons of these methods using large-scale problems have not been seriously conducted. Especially for methods solving multiclass SVM in one step, a much larger optimization problem is required so up to now experiments are limited to small data sets. In this paper they give decomposition implementations for two such "all-together" methods. They then compare their performance with three methods based on binary classifications: "one-against-all," "one-against-one," and directed acyclic graph SVM (DAGSVM).

Also report the training time, testing time, and the number of unique support vectors. Note that they are results when solving the optimal model. For small problems there is no testing time as they conduct cross validation. Here they say "unique" support vectors because a training data may correspond to different nonzero dual variables. For example, for the one-against-one and one-against-all approaches, one training data may be a support vector in different binary classifiers. For the all-together methods, there are variables so one data may associate with different nonzero dual variables. Here they report only the number of training data which corresponds to at least one nonzero dual variable. They will explain later that this is the main factor which affects the testing time. Note that the number of support vectors of the first six problems are not integers. This is because they are the average of the ten-fold cross-validation.

The main advantage of this paper is that it provide better classification results and easy to solve dual problems. The main disadvantage of this paper is some problems their training time is much longer. It is computationally more expensive to solve multiclass problems, comparisons of these methods using large-scale problems have not been seriously conducted.

S.Radha1 et.al[6]deals with retrieval of geographic images. For the retrieval of geographic images it uses local invariant features. SIFT algorithm is used. Multi SVM algorithm is used to perform image retrieval and using this algorithm the false image retrieval is considerably reduced. An SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. Local features allow a greater range of objects and spatial patterns to be observed. SIFT detector is used to identify the key-points in the image. Feature extraction is used to extract features from the image. Using the proposed algorithm the image retrieval is shown to give accurate results. It not only performs image retrieval but also detection and classification of geographic images.

The main advantage of this paper is performed image retrieval using Multi Support Vector Machine and it was seen that the false image retrieval has been considerably reduced. SVMs achieve significantly higher search accuracy than traditional query refinement schemes .The main disadvantage of this paper is manual image retrieval is time-consuming, laborious and expensive.

Paolo Gamba, et.al[4]deals with a mapping procedure exploiting object boundaries in very high-resolution (VHR) images is proposed. After discrimination between boundary and non boundary pixel sets, each of the two sets is separately classified. The former are labeled using a neural network (NN), and the shape of the pixel set is finely tuned by enforcing a few geometrical constraints, while the latter are classified using an adaptive Markov random field (MRF) model. The two mapping outputs are finally combined through a decision fusion process.

The main advantage of this paper is that it increases the performance. Classification is improved because neighborhood spatial patterns are chosen in order to follow the boundaries rather than crossing them, thus increasing the homogeneity of classification inside objects without "blurring" their boundaries. The main disadvantage of this paper is that it is difficult when the image is taken from long distance. When the input to the geometric refinement is not accurate, the output improves only partially.

Imdad Ali Rizvi et.al[5]deals with Object-based image analysis is quickly gaining acceptance among remote sensing community, and object-based image classification methods are increasingly being used for classification of land use/cover units from high-resolution satellite images with results closer to human interpretation compared to per-pixel classifiers. The problem of nonlinear separability of classes in a feature space consisting of spectral/spatial/textural features is addressed by kernel-based nonlinear mapping of the feature vectors. This facilitates use of linear discriminant functions for classification as used in artificial neural networks (ANNs). In this paper, performance of a recently introduced kernel called cloud basis function (CBF) is investigated with some modification for classification.A modified version of CBF was implemented as a kernel in neural network. In this paper, some modifications to the original algorithm proposed have been tested and implemented to accommodate the features of the multispectral images. Mahalanobis distance is preferred over Euclidean distance, as it captures the structure of the classes better. It takes into consideration the covariance matrix for each basis function; hence, a good estimation is desired for the covariance matrix in order to be appropriately used for the winner class selection.

The main advantage of this paper is that it provides better result for object detection. Higher classification accuracy compared to conventional radial basis function. The main disadvantage of this paper is that this approach provides low efficient detection. Large number of misclassifications occurs between these classes when only spectral information is taken into account.

Xin Huang et.al [3]deals with Classification and extraction of spatial features are investigated in urban areas from high spatial resolution multispectral imagery. The proposed approach consists of three steps. First, as an extension of the previous work [pixel shape index (PSI)], a structural feature set (SFS) is proposed to extract the statistical features of the direction-lines histogram. Second, some methods of dimension reduction, including independent component analysis, decision boundary feature extraction, and the similarity-index feature selection, are implemented for the proposed SFS to reduce information redundancy. Third, four classifiers, the maximum-likelihood classifier, backpropagation neural network, probability neural network based on expectation– maximization training, and support vector machine, are compared to assess SFS

and other spatial feature sets.The main advantage of this paper is that it provides better performance. This method does not need any searching and therefore is fast. It is used to achieve the spatial feature extraction for the HSRM data. The main disadvantage of this paper is that the result is not accurate.

3. Proposed System

The system use supervised feature learning algorithm. It follows the following steps like preprocessing, Noise Removal, Feature extraction and Classification. Feature extraction is done using Speeded-Up Robust Features (SURF) algorithm and classification is done by Latent Dirichlet Allocation (LDA). This system is very accurate than the previous methods and time complexity is also reduced. The block diagram is shown in figure 3.1

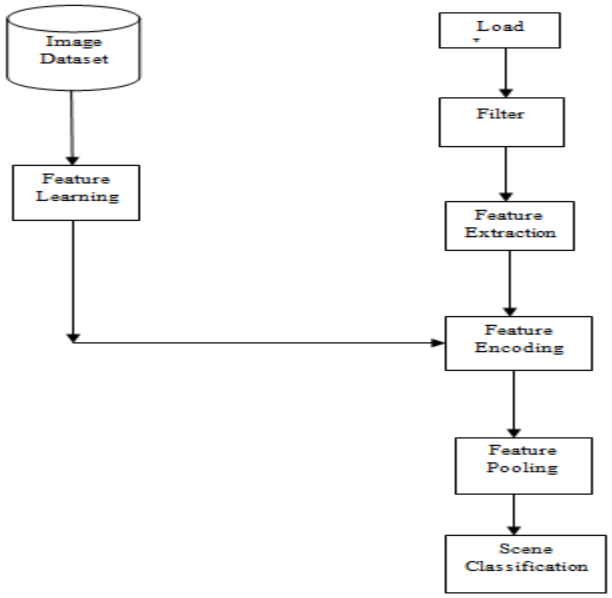

Figure 3.1: Block diagram

The goal is to accurately classify the given image patch into one of the predefined scene categories. Our approach consists of five broad steps:

i) Feature Extraction
ii) Feature Learning/Analysis
iii) Feature Encoding
iv) Feature Pooling
v) Classification.

In the first step, low-level feature descriptors from the image patch. As part of the feature learning process, a set of normalized basis functions are computed from the extracted features in an unsupervised manner. A variant of sparse coding called Orthogonal Matching Pursuit (OMP-k) to compute the basis function set. During feature encoding, we project the features onto the learned basis function set and apply soft threshold activation function to generate a set of sparse features. Then pool the sparse features to generate the final feature representation for the image patch. The final features are then fed to a linear support vector machine (SVM) classifier.

3.1 Feature Extraction

Evaluate the scene classification framework with three different feature extraction strategies. First, simply use the raw pixel intensity values as features, next measure the oriented filter responses at each pixel to construct the feature vector based on filter energy and finally, experiment with dense SIFT descriptors. Then perform feature extraction on the gray image generated from the RGB color channels. Our system computes low-level feature descriptor for each overlapping pixel blocks. Pixel blocks consist of local and contiguous groups of pixels. Then compute descriptors representing low-level feature measurements. At this stage, the input image is represented as set of vectors representing low-level feature measurements .When extracting features based on raw pixel intensity values, simply represent the pixel block as column vector $\mathbf{x}^i \in \mathrm{R}^b$where b is the product of the block dimensions and I represents the block index. Note that throughout this project, matrices are denoted with bold capital letters, vectors with bold small letters, scalars in italicized letters, superscripted and subscripted indices to denote the column and row positions of the vector respectively, and indices enclosed in brackets denote the element position.

This filter bank consists of first and second derivatives of Gaussian functions at 6 orientations and 3 scales, 8 Laplacian-of-Gaussian, and 4 Gaussian at different scales. For each scale set the Gaussian width correspondingly to $\{1, \sqrt{2}, 2, 2\sqrt{2}\}$. For each pixel block, compute the average filter energy at every scale and orientation to generate feature vector $\mathbf{x}^i \in \mathrm{R}^b$where b = 48. Finally, compute SIFT-based descriptors for each pixel block. This is in contrast to the approaches in where feature descriptors are computed only at certain "interest points." Previous work showed that dense SIFT descriptors produced higher classification accuracy than the sparse "interest points"-based descriptors. For computing SIFT descriptors for each pixel block, the pixel block is further divided into 4 × 4 non-overlapping sub-blocks. For each sub block a magnitude weighted orientation histogram is computed. The orientations are divided into 8 intervals. The magnitudes are further weighted by a Gaussian function with σ equal to one-half the width of the descriptor window. Local histograms are stacked to form the feature vector $\mathbf{x}^i \in \mathrm{R}^b$where b = 128. Then use the dense SIFT implementation provided by for feature computation.

3.2 Feature Learning

Feature learning or representation learning is a set of techniques in machine learning that learn a transformation of "raw" inputs to a representation that can be effectively exploited in a supervised learning task such as classification. Feature learning algorithms themselves may be either unsupervised or supervised, and include auto encoders, dictionary learning, matrix factorization, restricted Boltzmann machines and various forms of clustering.

Multilayer neural networks can also be considered to perform feature learning, since they learn a representation of their input at the hidden layer which is subsequently used for classification or regression at the output layer, and feature learning is an integral part of deep learning, to the point that the two are sometimes considered synonyms. (By contrast, kernel methods such as the support vector machine compute a fixed transformation of their inputs by means of a kernel function, and do not perform feature learning.)

When the feature learning can be performed in an unsupervised way, it enables a form of semi supervised learning where first, features are learned from an unlabeled dataset, which are then employed to improve performance in a supervised setting with labeled data.

K-means clustering can be used for feature learning, by clustering an unlabeled set to produce k centroids, then using these centroids to produce k additional features for a subsequent supervised learning task. These features can be derived in several ways; the simplest way is to add k binary features to each sample, where each feature j has value one if and only if the j^{th} centroid learned by k-means is the closest to the sample under consideration. It is also possible to use

the distances to the clusters as features, perhaps after transforming them through a radial basis function (a technique that has used to train RBF networks).

In a comparative evaluation of unsupervised feature learning methods, Coates, Lee and Ng found that *k*-means clustering with an appropriate transformation outperforms the more recently invented auto-encoders and RBMs on an image classification task. *K*-means has also been shown to improve performance in the domain of NLP, specifically for named-entity recognition; there, it competes with Brown clustering, as well as with distributed word representations (also known as neural word embedding).

Feature learning consists of learning a set of basis functions **D** from the feature vectors extracted above. Note that the basis function set is also referred as dictionary, codebook, and visual words. First, randomly sample low-level features from the entire data set to generate matrix $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M]$ where M is the number of samples. We set $M = 100000$ for all the experiments described in the later section. The matrix is normalized by subtracting the mean and dividing by the standard deviation. Next, to whiten the data apply a Zero Component Analysis (ZCA) transform. The main idea here is that the feature elements representing spatially adjacent pixels might exhibit high correlation, and by removing these correlations can force the model to learn the high-order structure in the data. Then compute the whitened feature matrix as $\mathbf{X_{white}} = \mathbf{TX}$, where $\mathbf{T} = \mathbf{UP}^{-(1/2)}\mathbf{U^T}$ where **U** and **P** are the eigenvectors and eigenvalues of the covariance matrix of **X**. Next, given the whitened feature matrix $\mathbf{X_{white}}$, learn the basis functions by finding best solution for a minimization problem which is similar to the sparse coding framework. The basis function **D** is learned using alternate minimization of (1)

$$\min_{\mathbf{D},\mathbf{s}^i} \sum_i \|\mathrm{D}\mathrm{s}^i - \mathrm{x}^i\|_2^2$$

$$\text{subject to } \|\mathbf{D}^j\|_2 = 1, \forall j \qquad (1)$$

$$\text{and } \|\mathrm{s}^i\|_0 \leq k, \forall i$$

Where $\|\mathbf{s}^i\|_0$ is the number of nonzero elements in column vector $\mathbf{s}^i$. Iteration begins by randomly initializing **D** and **s**, and proceeds to minimize (1) by alternatively fixing the variables. To initialize **D** randomly pick feature vectors from $\mathbf{X_{white}}$ and normalize each column to be unit vector ($\|\mathbf{D}^j\|_2 = 1$). In this paper, set $k = 1$ so, given **D** we set $\mathbf{s}^i(j) = \mathbf{D}^{jT} x^i$ where $\arg\max_j \mathbf{D}^{jT} \mathbf{x}^i$ and all other elements of $\mathbf{s}^i$to 0. Now with sparse codes $\mathbf{s}^i$ fixed can compute $\mathbf{D} = \mathbf{X_{white}} * \mathbf{S}^T$ where $\mathbf{S} = [\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^M]$ run a fixed number of iterations (set to 100 for all the experiments) to generate $\mathbf{D} \in \mathrm{R}^{b \times d}$ where b is the feature length and d is the length of the dictionary set. The main idea behind the minimization framework is to find a set of basis functions and corresponding sparse weights that can be used to reproduce the original feature matrix ($\mathbf{X_{white}}$) with least reconstruction error. The set of normalized basis functions **D** generated at this step can be seen as a codebook based on which low-level feature descriptors are encoded during the feature encoding phase. As the size of the dictionary d increases, the number of basis vectors that will used to encode the low-level feature descriptor also increases resulting in a high-dimensional vector. In this project, carefully set the dictionary size d based on cross validation.

3.3 Feature Encoding

From the basis function set **D**, proceed to encode the low-level feature descriptors in terms of the basis functions. The main objective here is to generate a robust representation that effectively and efficiently encodes the local patterns in the scene. To highlight the importance of the feature encoding step analyzed other simple alternatives. Simply concatenating the original feature descriptors $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$, where N is the number of feature descriptors extracted from the image patch, could result in prohibitively high-dimensional feature vector which might be almost impractical to deal with considering the huge storage and computation cost.

Another alternative would be to simply average the feature descriptors to form a vector representing the scene category. Our experiments show that this naive strategy results in poor characterization of the scenes. Here, a strategy is explored in which the feature descriptors are encoded in terms of the basis function set **D**. Previous work shows that the basis function generation step can be paired with any suitable encoding method that yields the best performance for the problem at hand. Following this can employ a simple and efficient sparse feature generation strategy. The basis function set **D** represents normalized local spatial patterns that can be linearly combined to reconstruct the low level feature descriptors. To represent the scene in terms of the basis functions, project the feature descriptor $\mathbf{x}^i$onto the basis vectors represented in the set **D** to compute the linear weights.

Next apply a soft threshold activation function to generate sparse features. The main idea is that would like to retain information about the most important basis functions associated with the low level feature descriptor. In our encoding scheme, positive and negative weights above and below certain thresholds defined by the threshold parameter α are retained and remaining elements are forced to zero resulting in a sparse representation of the low level feature descriptor. Previously, a similar soft threshold activation function was used to estimate sparse codes. Our experiments confirm that the sparse features generated by encoding the positive and negative linear weights produces state-of-the-art classification results.

3.4 Feature Pooling

With the sparse features $\mathbf{z}^i$ computed for an image patch, can estimate the final feature representation based on simple statistics of the sparse features. One popular choice is to pool the sparse features using simple averaging as follows:

$$\mathrm{p} = (1/\mathrm{N}) \sum_{i=1}^{\mathrm{N}} \mathrm{z}^i \qquad (2)$$

Previous researchers have explored various other methods to pool the sparse features. The sparse features are pooled by computing local histograms at different spatial scales and bins, and histograms are concatenated to form the final feature representation. Instead of computing local histograms the maximum values for the sparse code at different scales and spatial bins are retained as features. The spatial co-occurrence statistics of sparse features are computed instead of direct pooling. However, most of these feature pooling strategies result in costly training and storage requirements. The advanced feature pooling strategies require nonlinear SVM kernels such as histogram of intersection (HIK) or Chi-Square kernels to be used for feature to class label mapping. This would result in SVM training costs on the order or $O(n3)$ and storage costs on the order of $O(n^2)$ for $n \times n$ kernel matrix. In this project, a simple averaging-based feature pooling is used to generate final feature representation for the image patch.

3.5 Scene Classification

Contextual image classification, a topic of pattern recognition in computer vision, is an approach of classification based on contextual information in images.

"Contextual" means this approach is focusing on the relationship of the nearby pixels which is also called neighbourhood. The goal of this approach is to classify the images by using the contextual information.

The intent of the classification process is to categorize all pixels in a digital image into one of several land cover classes, or "themes". This categorized data may then be used to produce thematic maps of the land cover present in an image. Normally, multispectral data are used to perform the classification and, indeed, the spectral pattern present within the data for each pixel is used as the numerical basis for categorization. The objective of image classification is to identify and portray, as a unique gray level (or color), the features occurring in an image in terms of the object or type of land cover these features actually represent on the ground.

Image classification is perhaps the most important part of digital image analysis. It is very nice to have a "pretty picture" or an image, showing a magnitude of colors illustrating various features of the underlying terrain, but it is quite useless unless to know what the colors mean. Two main classification methods are Supervised Classification and Unsupervised Classification.

With supervised classification, identify examples of the Information classes (i.e., land cover type) of interest in the image. These are called "training sites". The image processing software system is then used to develop a statistical characterization of the reflectance for each information class. This stage is often called "signature analysis" and may involve developing a characterization as simple as the mean or the rage of reflectance on each bands, or as complex as detailed analyses of the mean, variances and covariance over all bands. Once a statistical characterization has been achieved for each information class, the image is then classified by examining the reflectance for each pixel and making a decision about which of the signatures it resembles most.

This is a two-stage classification process:

1. For each pixel, label the pixel and form a new feature vector for it.
2. Use the new feature vector and combine the contextual information to assign the final label.

It includes the following steps:

(i) **Merging the pixels in earlier stages**

Instead of using single pixels, the neighbour pixels can be merged into homogeneous regions benefiting from contextual information. And provide these regions to classifier.

(ii) **Acquiring pixel feature from neighbourhood**

The original spectral data can be enriched by adding the contextual information carried by the neighbour pixels, or even replaced in some occasions. This kind of pre-processing method is widely used in textured image recognition. The typical approaches include mean values, variances, texture description, etc.

(iii) **Combining spectral and spatial information**

The classifier uses the grey level and pixel neighbourhood (contextual information) to assign labels to pixels. In such case the information is a combination of spectral and spatial information

(iv) Training and testing an Image Classifier

The data provided in the directory data consists of images and pre-computed feature vectors for each image. The JPEG images are contained in data/images. The training images will be used as the positives, and the background images as the negatives.

Support Vector Machine Classification

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. Margin means the maximal width of the slab parallel to the hyperplane that has no interior data points.

4. Results and Discussion

4.1 Loading Input

This option allows us to select the required input image from the test database from a pop-up window that appears on the user interface screen as in figure 4.1

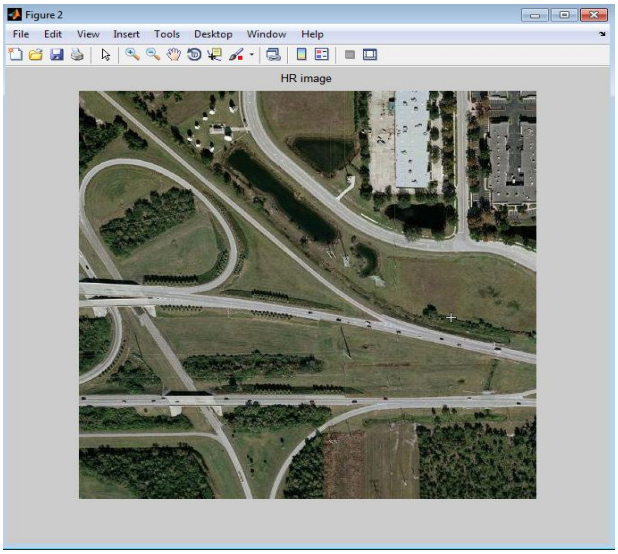

Figure 4.1: Input Image

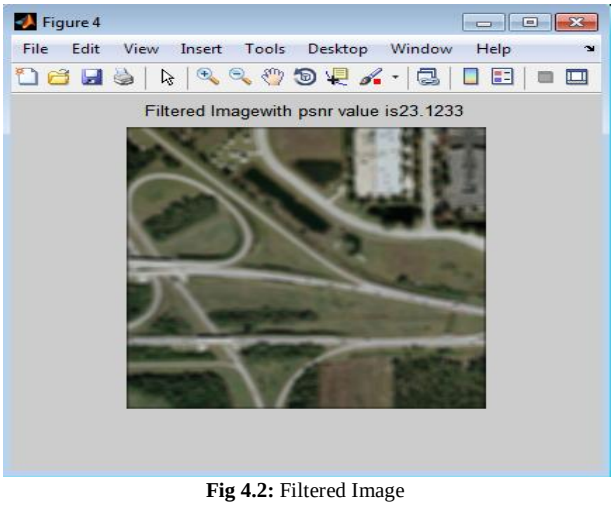

Fig 4.2: Filtered Image

4.2 Filtering

The input image is filtered using Kalman filter to improve the quality of input image and reduce the noise. PSNR can be calculated to find the quality improvement in the filtered image as shown in figure 4.2

4.3 Feature Extraction

The features are extracted using SURF algorithm as in figure 4.3

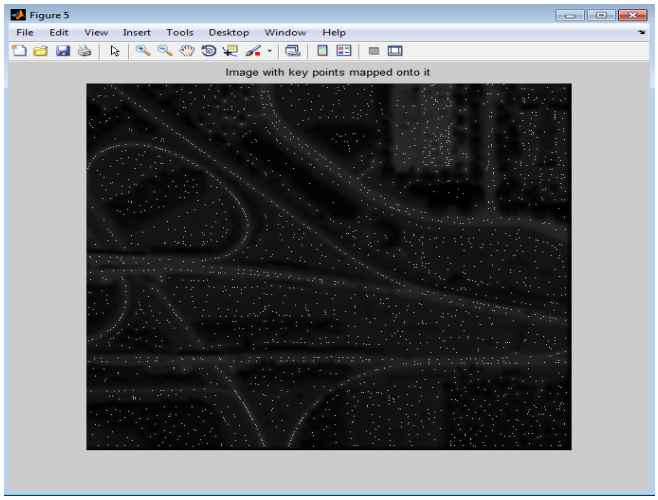

Figure 4.3: SIFT Feature Extraction

4.4 Feature Analysis

A set of normalized basis functions can be computed in an unsupervised manner from the features extracted in SIFT feature extraction. For this, Zero Component Analysis (ZCA) method is used to whiten the data. The correlation between spatially adjacent pixels is removed as in figure 4.4.

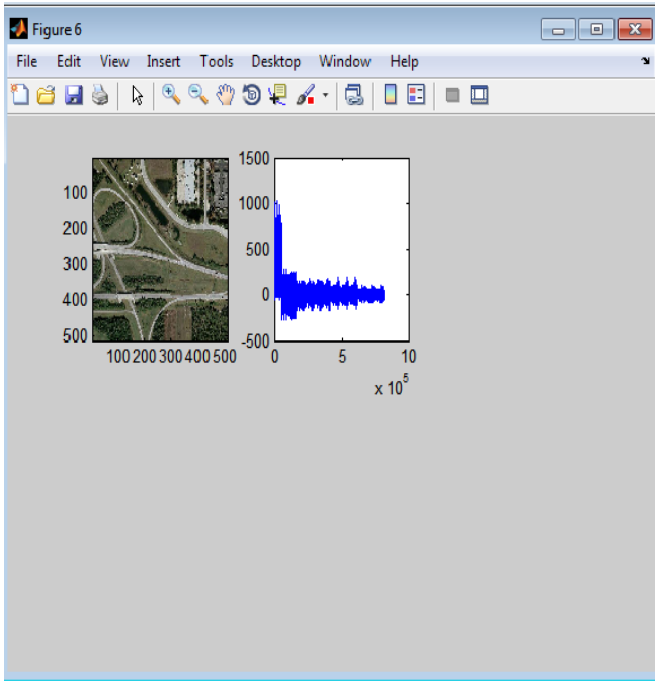

Figure 4.4: Feature Analysis

4.5feature Encoding

The images in the training database are encoded into four scene categories- Developed Suburban, Developed Urban, Emerging Suburban and Emerging Urban. The training images in the Developed suburban are shown below in figure 4.5.

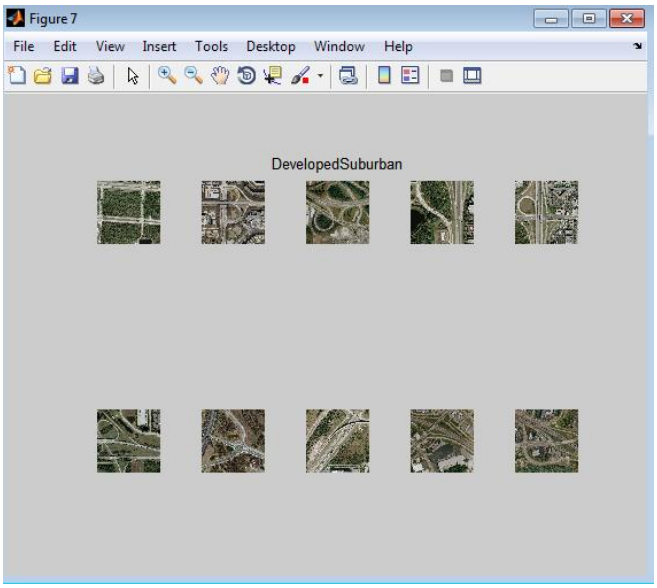

Figure 4.5: Feature Encoding of Developed Suburban

Suburban The training images in the Developed Urban are shown below in figure 4.6.

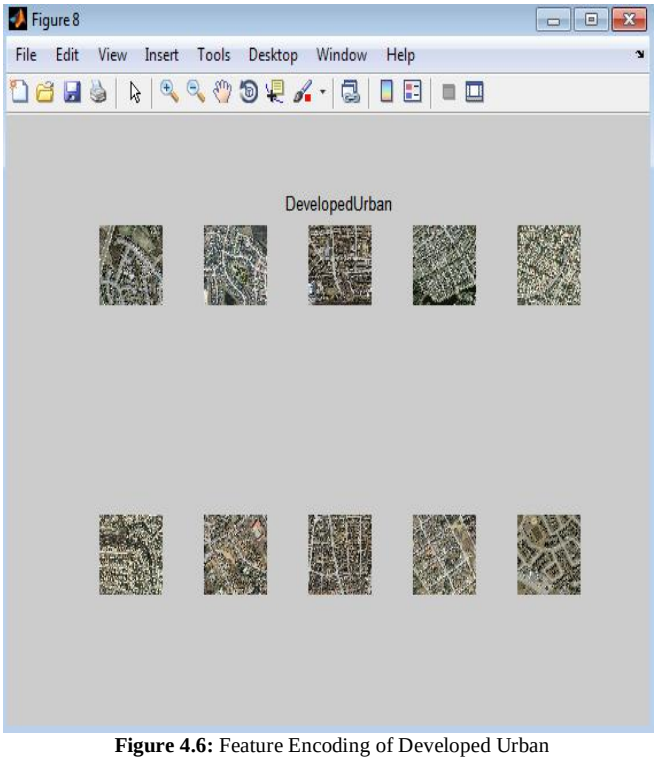

Figure 4.6: Feature Encoding of Developed Urban

The training images in the Emerging Suburban are shown in figure 4.7

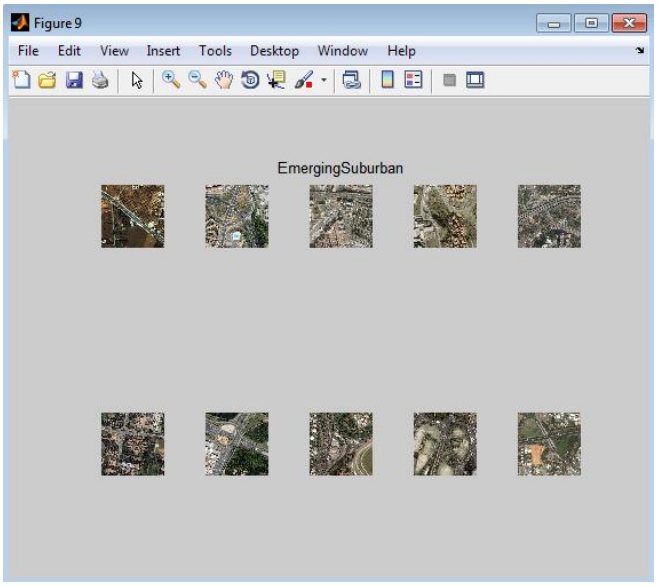

Figure4.7: Feature Encoding of Emerging Suburban

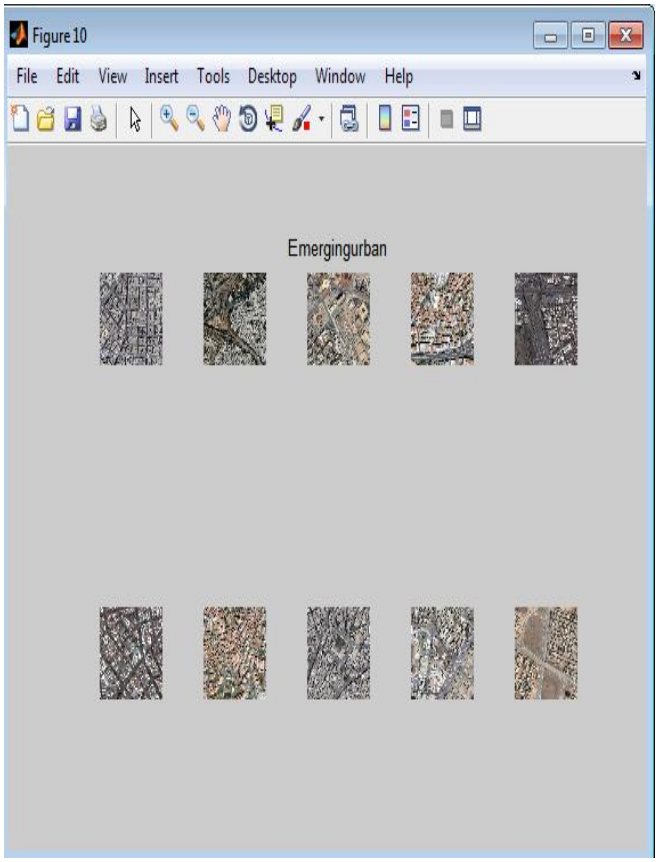

Figure 4.8: Feature Encoding of Emerging Urban

The training images in the Emerging Urban are hown above in figure 4.8

4.6 Scene Classification

Pooling of the features extracted from the input image is done to generate the final representation for the image. This stage maps the input image to its corresponding scene label as in figure 4.9.

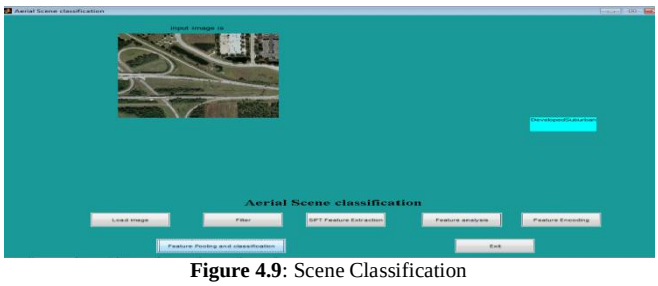

Figure 4.9: Scene Classification

5. Conclusion

In contrast to previous work on satellite image classification where the focus was on pixel or object-level thematic classification, here we explore a method to directly model aerial scene by exploiting the local spatial and structural patterns in the scene. Our approach model scenes by passing the complicated steps of segmentation and individual segment classification. The proposed classification framework involves feature extraction, learning, encoding and pooling. The feature extraction is done using SURF technique which is faster compared to SIFT based feature extraction. The time complexity is also greatly reduced. SURF feature extraction along with Latent Dirichlet Allocation helps in efficient classification of aerial images to their corresponding classes.

References

[1] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land use classification," in Proc. ACM Int. Conf. Adv.Geogr.Inf. Syst., 2010, pp. 270– 279.

[2] M. Pesaresi and A. Gerhardinger, "Improved textural built-up presence index for automatic recognition of human settlements in arid regions with scattered vegetation," IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 1, pp. 16–26, Mar.2011.

[3] I. A. Rizvi and B. K. Mohan, "Object-based image analysis of high resolution satellite images using modified cloud basis function neural network and probabilistic relaxation labeling process, "IEEE Trans. Geosci. Remote Sens., vol. 49, no.12, pp. 4815–4820, Dec 2011.

[4] R. Bellens, S. Gautama, L. Martinez-Fonte, W. Philips, J. C.-W. Chan and F. Canters, "Improved classification of VHR images of urban areas using directional morphological profiles," IEEE Trans. Geosci. Remote Sens., vol. 46, no. 10, pp. 2803–2813, Oct. 2008.

[5] A. K. Shackelford and C. H. Davis, "A combined fuzzy pixel-based and object- based approach for classification of high-resolution multispectral data over urban areas," IEEE Trans. Geosci. Remote Sens., vol. 41, no. 10, pp. 2354–2363, Oct. 2003.

[6] P. Gamba, F. Dell'Acqua, G. Lisini and G. Trianni, " Improved VHR urban area mapping exploiting object boundaries," IEEE Trans. Geosci.Remote Sens.,vol. 45, no. 8, pp. 2676–2682, Aug. 2007.

[7] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in Proc. IEEE Int. Conf. Comput. Vis., 2003, pp. 1470–1477.

[8] Y.-L. Boureau, F. Bach, Y. LeCun and J. Ponce, "Learning mid-level features for recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2010, pp. 2559–2566.

[9] S. Lazebnik, C. Schmid and J. Ponce, "Beyond bags of features: Spatial pyramid matching f or recognizing natural scene categories," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2006, vol. 2, pp. 2169–2178.

[10] D. G Lowe, "Object recognition from local scale-invariant features," in Proc. IEEE Int. Conf. Comput.Vis.,Kerkyra,Greece, 1999, pp. 1150–1157.