Comparative Analysis of Supervised Classification Algorithms for Urban Sprawl Detection

N. MINALLAH, W. KHAN, A. RASHID, M. N. KHAN, N. AZIZ, M. UZAIR, S. YOUSAF

National Center for Big Data and Cloud Computing (NCBC), University of Engineering and Technology Peshawar, Pakistan

Received 4th October 2018 and Revised 29th May 2019

Abstract: As human population is increasing, a number of small towns are turning into big cities. But human race has developed itself in technological terms smartly, which is helping human kind to act efficiently and consume the resources appropriately. Collection of urban sprawl statistics of an area has become efficient by using remote sensing. In comparison to the traditional methods, the new method of using remote sensing for the detection and classification of urban sprawl has substantially enhanced. Using this data management has become capable of taking suitable measures for its residents. This work compares six supervised classifiers, i.e. Maximum Likelihood, Minimum Distance, Support Vector Machines, Mahalanobis, Parallelepiped and Feed Forward Neural Network for urban classification. The data used is of SPOT 5 and criteria for comparison of classifiers is based on accuracy. Due to the absence of blue band in SPOT imagery the collected data samples for training tend to be complex and overlapping. Training data collected and divided into 10 different samples, show 82.74% accuracy for Artificial Neural Network. In comparison with Artificial Neural Networks, the lowest recorded results are of parallelepiped Classifier.

Keywords: Machine Learning, Maximum Likelihood, Minimum Distance, Support Vector Machines, Mahalanobis, Parallelepiped and Feed Forward Neural Network, Remote Sensing, Urban Sprawl detection, SPOT-5

1. INTRODUCTION

The world is changing rapidly and in the past few decades there has been a considerable increase in urban area. Due to the increasing population, urban Planning and Management has become appalling task. For observing and monitoring the urban sprawl, one of the most important and basic tool is remote sensing, so that further actions and planning can be carried out. This work presents a comparison of state of the art supervised classification techniques for Urban Sprawl Change detection. For experimentation the chosen study area is the region of Hayatabad, Peshawar, Pakistan. SPOT-5 Satellite Data for year 2015 has been acquired from SUPARCO, Pakistan.

The rest of the paper resides as: Section 2 discuss study area considered for research. Section 3 describes the data acquired from SUPARCO Pakistan. Section 4 is related to Experimental Setup, following by it Section 5 describes parameters setting for classifiers. While Section 6 discusses impact of classifiers. Section 7 is for Results and Discussions and last section (Section 8) which discusses conclusion of our work.

2. STUDY AREA

The test site for this work is Hayatabad, which is an urban settlement in the valley of Peshawar. Peshawar is the provincial capital of Khyber Pakhtunkhwa (KP), Pakistan, residing at 34.0151° N and 71.5249° E. The Surroundings of Peshawar is mountainous regions by North, South and West.

3. DATA DESCRIPTION

The satellite data used in this work is SPOT (Satellite Pour l’ Observation de la Terre), a series of 7 satellites launched by CNES (French Space Agency). SPOT 5 level 1A, August data of 2014 Hayatabad region has been acquired from SUPARCO (Pakistan Space and Upper Atmosphere Research Commission). (Fig. 1) shows the experimental area of Hayatabad. The details of SPOT 5 satellite Images listed in (Table 1).

4. EXPERIMENTAL SETUP

The basic methodology adopted for completion of this work is mentioned in this section of manuscript, which includes preprocessing of the acquired satellite data, collection of training data for classifiers, details of training data used in the study, classification algorithms and methods used for generating results.

<table>
<thead>
<tr>
<th>Band</th>
<th>Band Width Range (nm)</th>
<th>Code</th>
<th>Maximum Resolution (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>0.5-0.59</td>
<td>Green</td>
<td>10.0</td>
</tr>
<tr>
<td>B2</td>
<td>0.61-0.68</td>
<td>Red</td>
<td>10.0</td>
</tr>
<tr>
<td>B3</td>
<td>0.78-0.89</td>
<td>NIR</td>
<td>10.0</td>
</tr>
<tr>
<td>SWIR</td>
<td>1.58-1.75</td>
<td>SWIR</td>
<td>20.0</td>
</tr>
</tbody>
</table>

Table 1 SPOT Image Band Characteristics
4.1 Pre-Processing

The image that is acquired is contrast stretched by enhancing intensity values of the image. The data has also been georeferenced according to GCS WGS 1984.

4.2 Training Data for Classifiers

In our work we have selected 4 class training data to train our classifiers. These include Vegetation, Roads, Buildings and Barren land. Pixel based statistics of these classes are shown in Table 2. These datasets are created in a descending manner, showing percentages of the total ground truth data with the regular intervals of 10%. 3D scattered plot of 100% training dataset is illustrated in (Fig. 2). Furthermore collected training or ground truth data means of Barren Land, Buildings, Roads and Vegetation are represented in (Fig. 3).

4.3 Classification

Machine learning has enabled us to process complex data, and provide valuable information in terms of defined classes. We compared the following supervised classifiers in our study:

- Minimum Distance
- Support Vector Machines (SVM)
- Feed Forward Neural Networks (FFNN)
- Mahalanobis Distance Classification
- Parallelepiped Classification

Using the collected training data, we first train the above mentioned classifiers and then test these with validation data. Mentioned classifiers are tested on the basis of their accuracies discussed in results and discussion section.
Comparative Analysis of Supervised Classification...

Maximum Likelihood

Maximum likelihood classification (MLC) is one of the most widely used supervised classifier. MLC is a standout amongst the most famous strategies for image classification in remote sensing. Each pixel in MLC with the most extreme probability is characterized into the relating class. Every pixel is grouped to the class that has the maximum likelihood (that is, the greatest probability). In the event that maximum probability is smaller than a threshold value you indicate, the pixel stays unclassified. The below equation represents a discriminant function (Pushpendra et al., 2014) of MLC.

\[ G_i(X) = \ln P(w_i) - 0.5 \ln \left| \sum \frac{1}{\sigma_i} \right| + \frac{1}{2} \sum \frac{1}{\sigma_i} \left( X - \mu_i \right)^2 \]

Where:
- \( i \) = ith class
- \( X = n \)-dimensional data
- \( P(oi) = Probability of class \omega_i \) and is assumed same for all classes
- \( | Li | = determinant of the covariance matrix of the data in class \omega_i \)
- \( L^{-1} = inverse of | L | \)
- \( S(M_i) = X - M_i \)
- \( M_i = Mean vector \)

MLC has variety of applications. It is based on Bayes classification (John and Richards, 1999) and Mean vector, covariance metrics are the Key component of MLC that can be regained from training data (Pushpendra et al., 2014). The algorithm depends upon the statistical model of Gaussian probability density function (Fabio et al., 1999). The shortcoming of MLC process is that it cannot be used straightaway when the number of features is greater than the number of training samples. Thus, the applicability of nearest neighbor or MLC can be restricted when there is minimum number of training samples and high-dimensional feature space of hyperspectral dataset (Chenghai et al., 2009) But comparatively MLC produces nearly better or similar classification accuracy than others classifiers.

Minimum Distance

The Minimum Distance classification is one of the commonly used and computationally simple algorithm having equivalent classification accuracy to that of Maximum Likelihood approach. Maximum Likelihood approach is considered as one of the accurate classifier at the cost of intensive computations. The operation performed using the Minimum Distance classifier is very simple. The only thing it requires from the user is the training samples to be used in classification. The distance between the unknown vector and the mean vector for class \( i \) is calculated using Euclidean Distance formula given in equation below (Russell and Green, 2008).

\[ d(m_i, X) = \sqrt{\sum_{i=1}^{n} (m_i - X)^2} \]

Where \( m_i \) is dimensional mean vector for class \( i \) and \( X \) is \( n \)-dimensional unknown vector. In matrix form

\[ d(X, m_i) = \sqrt{(m_i - X)^T - (m_i - X)} \]

Where \( T \) is the transpose of the Matrix.

A threshold value has to be defined for the classifier so that it can exclude the pixels having lower value than the threshold.

Mahalanobis Distance

Generally, in multivariate analysis, variables (usually two in number) are represented through coordinate system in a Euclidean space (Li et al., 2018). But it is difficult to represent as well measure the variables when greater than two, along the planar coordinates. This is where the Mahalanobis distance (MD) comes into picture (Surekha et al., 2019). It provides a powerful method for measuring the similarity of one set of conditions to ideal set of conditions, and prove useful while comparing the similarity of a test data to training data samples being taken as true data.
The quantitative similarity of entire landscape to ideal conditions is described using Mahalanobis distance (Umair et al., 2015). Moreover, Mahalanobis distances take benefit of covariance among variables, as it depends on the mean and variance of the predicting variables, as well the covariance matrix of all other variables (Sahu, et al., 2015). An ellipse in 2D space (i.e. when only 2 variables are measured), or an ellipsoid or hyper ellipsoid when more variables are used is formed around the mean of the constant Mahalanobis Distance (Rahul et al., 2018).

The MD is a measure of the distance between a point P and a distribution D, introduced by P. C. Mahalanobis in 1936. It is a generalize idea in multi-dimension to measure a point P is how many standard deviations away from the mean of data distribution D. If the point P lies at the mean of Distribution D, the distance is zero, and the distance grows along each principle component axis as P moves away from the mean of D. If each of these principle component axes is re-scaled to have unit variance, then in transformed space the Mahalanobis distance correspond to standard Euclidean distance. Thus the Mahalanobis distance is scale-invariant and unit less, and consider the correlations of the data set (Ajay et al., 2017).

In order to classify a test pixel that belongs to one of N classes, the covariance matrix of each class is usually based on known samples belong to each class (Maryam 2019). Then, for a given test sample, Mahalanobis distance to each class is computed, and for a class for which it is minimum, it classifies the test point to that class (Najam et al., 2016).

Mahalanobis Distance for each pixel to each class in the image as calculated as follow:

\[ D^2 = (x - m)^T C^{-1} (x - m) \]

Where:
- \( D^2 \) = Mahalanobis distance
- \( x \) = Data Vector
- \( m \) = independent variables Vector of mean values
- \( C^{-1} \) = Independent variables Inverse covariance matrix
- \( T \) = Indicates vector should be transposed.

The following properties are exhibited by Mahalanobis distance Classifier:

- It takes into consideration the difference of variances in each direction.
- It also accounts for the covariance among variables.
- For uncorrelated variables with unit variance it is reduced to the familiar Euclidean distance (Ajay et al., 2019).

Mahalanobis distance classifier demonstrated to be useful in cases when the axes are correlated in feature space (Vignajeeth et al., 2018). It finds the center of the sample data and the standard deviation of the sample points from this central value of data. If a test pixel having one standard deviation distance from this center of the data than this test pixel is classified into that class. The further away a pixel is from center of data, higher the chances of pixel not belonging to a class and is placed unclassified.

The shortcoming of this classifier is that it uses statistics for each class. All pixels are characterized to the closest class unless a separation limit is determined.

**Support Vector Machine**

Proposed nearly 50 years ago, SVM, a binary classifier, has become a proprietory choice for researchers in data analytics. SVM is a kernel based classifier where the training data is first scaled, upon which a discriminant function is applied (Giorgos et al., 2011). A hyper-line is drawn dividing the designated classes according to the feature set provided (Shih-Wei Lin, et al., 2008). The multi classification in SVM is performed in one against one technique (David et al., 2015). Using a designated kernel function, decision boundary called hyper-plane, is created, using training data. (Fig.4) presents the concept of separating hyper-plane in SVM (Varma, et al., 2016). The bold line in is defined by:

\[ w \cdot x + b = 0 \]

Where \( b \) and \( w \) are the parameter of the hyper-plane. Vectors which are not in hyper plane are presented by:

\[ w \cdot x + b > 0 \]

Whereas the two margins or support vectors are presented by:

\[ w \cdot x + b = -1 \text{ and } w \cdot x + b = +1 \]

Where \( w \) is the vector product of two vectors. This kernel function is based on four characteristics.

1. Linear basis function

\[ K(x, y) = x^T y \]

2. Radial basis function

\[ k(x, y) = exp\{-\|x-y\|^2/\sigma^2\} \]

3. Polynomial degree \( k \)

\[ k(x, y) = (1 + x^T y)^k \]

4. Sigmoid

\[ \tilde{K}(x, y) = \left\{ -\sum_{i=1}^{N} \frac{(x-y)^2}{\sigma^2} \right\} \]
Neural Net
   The motivation behind neural networks is the brain neural system used to perform different decision-making tasks. In brain there are different sections for a task (SitiKhairunniza-Bejo, et al., 2014). Artificial neurons are the building blocks of artificial neural networks. At least three layers are required for construction of Artificial Neural Networks ANN’s. First layer is input layer, that distributes input to PE’s middle layer is hidden layer that consists of number of Processing Elements PE’s the number of PE’s is dependent on complexity of problem and the last layer is output layer (Marchant and Onyango. 2003).

   ANNs is usually helpful in the process of recognition (Konstantinos et al., 2019), classification, enhancement (Senthil and Kumar. 2018) analysis (Jean et al., 2008) estimation and prediction (Amin et al., 2016). Having no evident sets of parameters that require estimation, ANN is referred to as non-parametric.

   Generally, all of the NN’s are trained first, and then compares incoming image pixel values with the available region of interest classes by checking on which side of the linear separating surface they lie (Amin et al., 2011). The necessary processing element model is based on input from the previous layer and weight. An individual neuron or processing element takes a vector of inputs \( x = x_1, x_2, \ldots, x_N \). w is connection weight is a bias.

Feed Forward neural Network
   The feed forward neural network was the simplest type of artificial neural network devised (Edwin and Zagajewski. 2017). A feed forward neural network is also known as multilayer Perceptron’s (MLPs) is an artificial neural network (Olivier et al., 1993). The goal and function of feed forward neural network with n perceptron that we’d like to use to train to solve some problem is to approximate for a classifier. The process of learning take place by the adjustment of node weights so that to minimize the difference between output and node activation output. Approximate mapping of an input pixel in to given set of categories based on the calculated sum of the products of the weights \( L, w_i, x \) and the inputs that is calculated at each node. Each pixel is processed to make a class selection for each pixel.

   The network is to learn weights and biases so that the output from the network correctly classifies the digit. Feed-forward networks have the following characteristics:

   - Processing elements are arranged in layers, input data travels via first input layer and the output layer producing outputs. The intermediate layers have no connection with the external world, and hence are called hidden layers
   - In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes
   - FFNN is acyclic means that within the network there is no feedback connections among neurons \( w_x \) to send back the information (Giles 2017).

   There are connections, where PE h1 is connected to input \( x_1, x_2 \) and \( x_3 \) and \( h_2 \) is connected to input \( x_1, x_2 \) and \( x_3 \) and so here all the weights are given in matrix equation below. The PE computes matrix product of the hidden with those weights plus its own bias then passes to the activation function as well, matrix computation is given.

\[
\begin{bmatrix}
    w_{11} & w_{12} & w_{13} \\
    w_{21} & w_{22} & w_{23}
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2
\end{bmatrix}
= \begin{bmatrix}
    w_{11} \cdot x_1 + w_{12} \cdot x_2 + w_{13} \cdot x_3 \\
    w_{21} \cdot x_1 + w_{22} \cdot x_2 + w_{23} \cdot x_3
\end{bmatrix}
\]

   The math that is needed to process here in the hidden node is the weighted sum of each input multiply by the column of input given together as H. The mathematical computation model for neuron is given in equations below, that is the computation that will take place on each artificial neuron. We do all this computation to figure out how to tune all this system to match the output to what we think it should be. Small alteration in weight cause insignificant corresponding change in the output from neural network, this property will make learning possible. But as it occurs a small alteration in weights leads to a big change in output.

\[
H_i = \sigma (w_i \cdot I_i + B_i)
\]
\[
O = \sigma (w \cdot H + B)
\]
Parallelepiped

Parallelepiped is a supervised statistical classifier that classify multispectral data using simple decision rule (Ana et al., 2018). It categorize a pixel to a class while using the class limit and signature stored in it. A pixel is classified and assigned to a class, if its value lies between the upper and lower threshold value for all n bands being classified, for that particular class. While pixels that do not fall within any of the parallelepiped classes are designated as unclassified (Mohammed et al., 2018). It takes into account the variance in training classes, so is computationally simple but due to correlation amongst classes problems may arise from parallelepiped overlap. Which results in large number of classified pixels as tied and result in poor accuracy. Being a fast classifier having simple decision rules parallelepiped classifier is usually used when there is a need for speed (John and Richards 1999).

In this classifier, a pixel membership is tested through multidimensional boxes which are constructed for each class using class mean and standard deviation. A pixel X vector is allocated to class j if the following equation is satisfied:

\[ \mu_{ij} - K \sigma_{ij} X_{ij} + K \sigma_{ij} \]

Where \( \mu_{ij} \) is mean value for class j in band i, K is a constant, and \( \sigma_{ij} \) is standard deviation for class j in band i, and this hold true for all i=1 to n, where n is number of spectral bands,

5. RESULTS AND DISCUSSION

This section discusses performance of classifiers, based on its accuracies.

Impact of NN

In this case an accuracy of 82.74% was achieved by using 100% data as training. While an accuracy of 63.68% was attained using 20% training data.

Kappa coefficient a quantitative measure is used to assess the classification accuracy and get the reliable accuracy estimation. The analysis of an individual region of interest ROI class or several classes for classified images will contain an amount of errors. This helps us to detect the sources of errors and find a way to improve the classification accuracy.

Impact of SVM

SVM is a stat of the art classifier, used in many domains throughout research and development. The basic functionality of SVM depends upon the kernel function used in the computation of boundaries. This kernel function is setup after observation of ground truth or training data of the classifier. Different kernels wiz linear, sigmoid, polynomial degree and radial basis function has been applied during our testing of SVM. Due to the complexity of our training data illustrated in (Fig 3), we chose polynomial degree 5, based kernel with gamma being 0.4 and the probability of error to be 0.9. Testing upon these setting gave us good results in terms of accuracy for 4 different classes.

The accuracy 77.201 was achieved by using 40% of data as training and 81.8586% was attained for 100% training data as shown in (Table 3).

Impact of Minimum Distance

Minimum distance provides an accuracy of 62.7692 for 60% training data while 65.1332% for 10% of training data. Using minimum distance classifier the results we achieve are ambiguous and unreliable.

Impact of Mahalanobas

For Mahalanobis Distance multiple threshold values are used for each class for correct classification of all pixels to their respective classes. These threshold values are selected from the statistics of each class of all bands. The Maximum standard deviation values in all bands for each class is picked as threshold for that class. Pixel that lie at a distance greater than this threshold value are labeled as unclassified. The Classifier has highest probability of detection when threshold value is 0, as the threshold increases, so as the number of unclassified pixels and the probability of Detection decreases.

This behavior of Mahalanobis classifier indicates that our classification on the retrieved ground truth data is not up to the mark. An accuracy of 70.051 was attained after training the classifier with 70% data and 66.74% accuracy was recorded for 10% training data depicted in (Table 3).

Impact of Parallelepiped

For Parallelepiped Classification different threshold for each class are taken into account based on the maximum standard deviation of each class for all the bands.

As parallelepiped classifier work best in the areas where there is minimum overlap. And as the classes in ground truth imagery overlap in the underlying problem so the results of the said classifier is poor.

Accuracy and Kappa Coefficient

(Table 3 -4) shows results of Mahalanobis, SVM, ANN, Maximum Likelihood, Minimum Distance and Parallelepiped in terms of their kappa and classifier accuracy. Judging from information plotted in (Fig.5). Artificial Neural Network has a higher accuracy of 82.74% and kappa coefficient of 0.7688 than other five classifiers. Second standing in terms of good results is SVM, a binary classifier. SVM attained an accuracy of 81.85% while a kappa value of 0.757 has been noted accordingly. It has been noted that with the decrease of
training data in ANN both accuracy and kappa decreases. The lowest recorded accuracy and kappa coefficient is recorded by training 20% of samples and testing the remaining 80% as testing for the classifier. Poor results have been found by using parallelepiped classifier tends to be 51.23% recorded accuracy for using all the samples as training data.

Table 3 Accuracy Comparisons of Mahalanobis, SVM and ANN in terms of Kappa and Accuracy

<table>
<thead>
<tr>
<th>Training Set Data (%)</th>
<th>Mahalanobis Distance</th>
<th>SVM (Polynomial Degree)</th>
<th>ANN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall Accuracy</td>
<td>Kappa Coefficient</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>69.2735</td>
<td>0.5894</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>69.4541</td>
<td>0.5919</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>70.051</td>
<td>0.5999</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>69.2597</td>
<td>0.596</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>69.2041</td>
<td>0.5885</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>69.3152</td>
<td>0.5899</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>69.7597</td>
<td>0.5959</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>68.9298</td>
<td>0.5743</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>67.7872</td>
<td>0.5703</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>66.7454</td>
<td>0.5565</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training Set Data (%)</th>
<th>SVM (Polynomial Degree)</th>
<th>Overall Accuracy</th>
<th>Kappa Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Overall Accuracy</td>
<td>Kappa Coefficient</td>
</tr>
<tr>
<td>100</td>
<td>81.8586</td>
<td>0.757</td>
<td>82.74</td>
</tr>
<tr>
<td>90</td>
<td>81.7058</td>
<td>0.755</td>
<td>81.2</td>
</tr>
<tr>
<td>80</td>
<td>81.5947</td>
<td>0.7535</td>
<td>82.08</td>
</tr>
<tr>
<td>70</td>
<td>81.6363</td>
<td>0.754</td>
<td>79.22</td>
</tr>
<tr>
<td>60</td>
<td>81.5891</td>
<td>0.7494</td>
<td>81.09</td>
</tr>
<tr>
<td>50</td>
<td>80.9557</td>
<td>0.7449</td>
<td>74.28</td>
</tr>
<tr>
<td>40</td>
<td>77.201</td>
<td>0.7354</td>
<td>74.34</td>
</tr>
<tr>
<td>30</td>
<td>78.2436</td>
<td>0.7235</td>
<td>78.39</td>
</tr>
<tr>
<td>20</td>
<td>77.2436</td>
<td>0.6983</td>
<td>63.68</td>
</tr>
<tr>
<td>10</td>
<td>78.65</td>
<td>0.6935</td>
<td>71.81</td>
</tr>
</tbody>
</table>

Table 4 Accuracy Comparisons of MLH, Minimum Distance and Parallelepiped in terms of Kappa and Accuracy

<table>
<thead>
<tr>
<th>Training Set Data (%)</th>
<th>Maximum Likelihood</th>
<th>Minimum Distance</th>
<th>Parallelepiped</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall Accuracy</td>
<td>Kappa Coefficient</td>
<td>Overall Accuracy</td>
</tr>
<tr>
<td>100</td>
<td>63.1199</td>
<td>0.5096</td>
<td>51.236</td>
</tr>
<tr>
<td>90</td>
<td>63.3431</td>
<td>0.5126</td>
<td>51.236</td>
</tr>
<tr>
<td>80</td>
<td>62.945</td>
<td>0.5074</td>
<td>51.2402</td>
</tr>
<tr>
<td>70</td>
<td>63.1997</td>
<td>0.5107</td>
<td>50.8127</td>
</tr>
<tr>
<td>60</td>
<td>62.7692</td>
<td>0.505</td>
<td>51.0546</td>
</tr>
<tr>
<td>50</td>
<td>63.7878</td>
<td>0.5186</td>
<td>51.0531</td>
</tr>
<tr>
<td>40</td>
<td>63.529</td>
<td>0.5147</td>
<td>49.9103</td>
</tr>
<tr>
<td>30</td>
<td>63.1894</td>
<td>0.5188</td>
<td>51.3025</td>
</tr>
<tr>
<td>20</td>
<td>65.1323</td>
<td>0.5346</td>
<td>51.6294</td>
</tr>
<tr>
<td>10</td>
<td>65.1323</td>
<td>0.5346</td>
<td>51.6294</td>
</tr>
</tbody>
</table>

CONCLUSION

Urbanization is not simply a new method, however a speedy transformation of human social roots on a worldwide scale, wherever village culture is being replaced by the urban culture. In Peshawar, the complexes of urban development is thus speedy that it demands fast response and perspective planning and management of it. Therefore, it is necessary and basic for policy manufacturers to include remote sensing into urban management. As Remote Sensing (RS) is a field that can be used to extract many type of information of an object, using Satellite imagery, without in physical contact of that object.

This paper, combined remote sensing concepts with Machine Learning Algorithms (MLAs) to give most feasible way of doing the task of urban settlement calculations. The output of our work can have substantial effect in organization and planning of Peshawar. For the stated goal, SPOT 5 imagery of August 2014 OF Hayatabad region has been acquired from SUPARCO (Pakistan Space and Upper Atmosphere Research Commission).The imagery is classified individually by six supervised classifiers. Maximum Likelihood, Minimum Distance, Support Vector Machines, Mahalanobis, Parallelepiped and Feed Forward Neural Networks (FFNN) for urban classification in Hayatabad Peshawar. For the assessments of classifiers, accuracy and kappa coefficients are used. Ground Truth data, for training of classifiers, has been collected and divided into 10 different samples as shown in (Table 2). Statistics of classifiers accuracy and kappa coefficients on 10% to 100% are tabulated in (Table 3 and Table 4). These statistics are plotted in Statistics which shows the behavior of classifier on validation. k-fold cross validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In our work k = 10-90. (Table 3)clearly highlights that Artificial Neural Network has a higher accuracy of 82.74% and kappa coefficient of 0.7688 than other five classifiers.
Second standing in terms of good results is SVM, a binary classifier. SVM attained an accuracy of 81.85% while a kappa value of 0.757 has been noted accordingly. It has been noted that with the decrease of training data in ANN both accuracy and kappa decreases. The lowest recorded accuracy and kappa coefficient is recorded by training 20% of samples and testing the remaining 80% as testing for the classifier. Poor results have been found by using parallelepiped classifier tends to be 51.23% recorded accuracy for using all the samples as training data.

REFERENCES:


