

Probability Occurrence Of Land Slide Using Soil Features And Classification

By

Vishal Kumar Das (University Roll No:11700114095)
Soumya Pramanik (University Roll No:11700114069)
Sudipta Ray (University Roll No:11700114082)

UNDER THE GUIDANCE OF

Mr. Jaydip Mukhopadhyay

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY IN COMPUTER
SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Session 2018



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RCC INSTITUTE OF INFORMATION TECHNOLOGY
[Affiliated to West Bengal University of Technology]
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700015

**DEPARTMENT OF COMPUTERS SCIENCE AND ENGINEERING
RCC INSTITUTE OF INFORMATION TECHNOLOGY**



TO WHOM IT MAY CONCERN

I hereby recommend that the Project entitled <**Soil detection and classification**> prepared under my supervision by **Vishal Kumar Das** (Reg. No.141170110095 , Class Roll No.CSE/2014/073), **Soumya Pramanik** (Reg. No.141170110069 , Class Roll No.CSE/2014/090), **Sudipta Ray** (Reg. No.141170110085 , Class Roll No.CSE/2014/082), of B.Tech (8th Semester), may be accepted in partial fulfillment for the degree of **Bachelor of Technology in Computer Science & Engineering** under West Bengal University of Technology (WBUT)..

.....
Project Supervisor
Department of Computer
Science and Engineering RCC
Institute of Information
Technology

Countersigned:

.....
Head Department of Computer
Sc & Engg, RCC Institute of
Information Technology Kolkata – 700015.

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GRC
INSTITUTE OF INFORMATION TECHNOLOGY**



CERTIFICATE OF APPROVAL

The foregoing Project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR
EVALUATION OF PROJECT

1. _____

2. _____

(Signature of Examiners)

ACKNOWLEDGEMENT

It is our pleasant duty to offer our service acknowledgement to those honorable personalities of Rcc institute of information technology, who have been a constant help and encouragement in carrying out this project.

First of all we would like to thank the Principal of Rcc institute of information technology, for providing standard facilities and infrastructure to complete this project efficiently.

We would like to express our sincere gratitude and thanks to project guide Professor **Mr. Jaydip Mukhopadhyay** for his continuous support and suggestion to frame and complete this interesting project on time. The faculty members of then department of CSE,for their valuable help intellectually and morally.

We would like to thank our friends and family members for their brilliant ideas in preparing this project. Finally we are very grateful to those entire people who have helped us directly or indirectly in bringing up and developing this project.

Vishal Kumar Das, CSE/2014/073

Soumya Pramanik, CSE/2014/090

Sudipta Ray, CSE/2014/085

Table of Contents

CONTENTS	PAGE NO.
ABSTRACT	06
INTRODUCTION OF THE PROJECT	07-08
PROPOSED SYSTEM	09-20
METHODOLOGY OF IMPLEMENTATION	21-22
RESUKTS AND SAMPLE OUTPUT	23-24
CONCLUSION	25
APPENDIX	26-28
FUTURE WORK	29
REFERENCES	30

ABSTRACT

This paper describes the SVM based classification and grading of soil samples using different scientific features. Different algorithms and filters are developed to acquire and process the colored images of the soil samples. These developed algorithms are used to extract different features like color, texture, etc. Different soil types like red, black, clay, alluvial, etc are considered. The classification makes use of Support Vector Machine, machine learning technique. SVM seeks to fit an optimal hyper plane between the classes and uses only some of the training samples that lie at the edge of the class distributions in feature space (support vectors). This should allow the definition of the most informative training samples prior to the analysis. The accuracy of a supervised classification is dependent to a large extent on the training data used. Till now classification of soil and classification of crop for the appropriate soil is done separately. This project aims at combining both the techniques, where classification of soil for appropriate soil is a part of classification of soil.

INTRODUCTION OF THE PROJECT

In past years, automation and intelligent sensing technologies have revolutionized each and every part of our world like agriculture, building, aeronautics etc. These initiatives have been accredited to the rising concerns about product quality and safety. Also, rising labour costs, shortage of skilled workers and the need to improve production processes have all put pressure on producers and processors. Automated solutions are the answer for the problems that are being faced today by the agriculture world. Agriculture planning plays a significant role in economic growth and food security of agro-based country. Selection of soil is an important issue for agriculture planning. Crop selector could be applicable for minimize losses when unfavourable conditions may occur and this selector could be used to maximize crop yield rate when potential exists for favourable growing conditions. Machine Vision Systems (MVS) provide an alternative to manual inspection of soil samples for their characteristic properties and the amount of nutrient material. Computers have been successfully employed for classification of plants, diagnosis of plant diseases, recognition of leaves and soil, etc. Till now classification of soil were done separately. But in this methodology two techniques are implemented together for providing a great usage for farmers and agriculture industry.

In practice a manual segmentation and classification procedure is followed. This procedure requires expertise, and is expensive, time consuming, subjective and not completely reproducible. The challenge is to automate this procedure. In order to achieve this a new algorithm called CONCC (CONstraint Clustering and Classification) was developed. It can be used in automatic classification with the constraint of contiguity and includes the following two steps:

- Segmentation: to find J segments of data from a single series data (e.g. CPT);
- Classification: to build a classifier to assign classes to these segments; it is built using measured data and extracted features from segments.

PROPOSED SYSTEM

A. Image Acquisition

Different images of soil samples which are to be classified are captured using color camera and are provided as an input to the system. The features of each type of soil are collected and are stored in a separate database. This database is later used in the final stage for soil and crop detection.

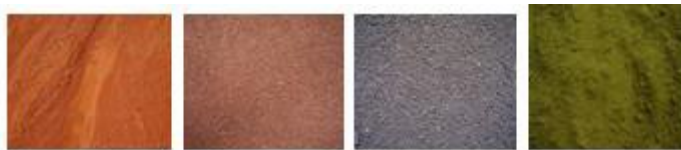


Fig. 1: Images of Soil Samples

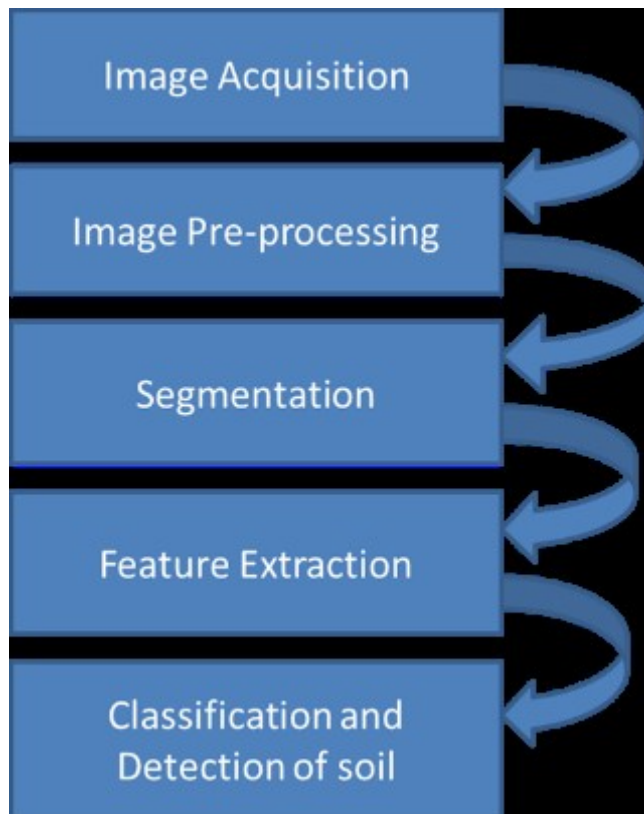


Fig. 2: Flowchart of the proposed system

B. Image Pre-processing

The image acquired from our previous stage is not error free. The quality of the image is decisive for results of analysis as it affects both the ability to detect features under analysis and precision of subsequent measurement. In order to get an error free image pre-processing techniques are applied. This phase is also known as the enhancement of the image since the image is enhanced by improving its contrast and removing errors to obtain a better quality image for our future processes. The image contains errors like noise or artifacts like scratches, lapping tracks, comet tails, etc which needs to be eliminated before the further processes. Hence a filter called the Smoothing filter is made use to remove the noise and artifacts from the image. There are two types of filters: low pass filter and high pass filters. Smoothing filter is a low pass filter. It is used to remove high spatial frequency noise from a digital image. Smoothing filter employs a moving window operator which affects one pixel of the image at a time, changing its value by some function of a local region of pixels. The operator moves over the image to affect all the pixels in the image. Thus with many iterations the smoothing filter gradually enhances the image by removing the errors. For more accuracy and clarity in the image its edges and contrast needs to be enhanced. Colour map of the image are used for enhancing edges and contrast.

C. Segmentation

Once the enhancement of the image is completed in the previous stage using image pre-processing techniques segmentation of the image is performed. Popular known algorithm K-means Clustering algorithm is employed for segmentation of the image. It is used as a partition clustering which aims at partitioning a given data set into disjoint subsets so that specific clustering criteria are optimized. The most widely used criterion is the clustering error criterion which for every point computes its squared distance from the corresponding cluster centre and then takes the sum of these distances for all points in the data set. The centre of the cluster is picked and each pixel of the image is assigned to the cluster. Re-computation of cluster centre is done by using the average of all the pixels. This iteration continues and the next step is to take each point belonging to a given data set and associate it to the nearest centroid.

K-means Algorithm is as follows-

- 1) Input: K, set of points x_1, \dots, x_n .
- 2) Place the centroids C_1, \dots, C_k at random locations.
- 3) Repeat until convergence:

For each point x_i :

Find nearest centroid C_j . $\arg \min D(x_i, C_j)$

Assign the point x_i to cluster j

For each cluster $j=1, \dots, K$: $C_j(a) = \frac{1}{n_j} \sum_{x_i \rightarrow C_j} x_i(a)$ for each $a=1, \dots, d$

For each cluster $j=1, \dots, K$:

New centroid $C_j =$ mean of all points x_i assign to cluster j in previous step.

- 4) Stop when none of the cluster assignments change.

A segment g_j to be identified is defined as:

$g_j = \{ \{X_{i1}, \dots, X_{iK}\}, \dots, \{X_{i1}, \dots, X_{iK}\} \}_{i=1}^{n_j}$
where $x_{ik} \in T^k$ and represents the measured data;

$i = 1, 2, \dots, n_j$, where n_j is the number of instances in segment g_j ;

$k = 1, 2, \dots, K$, where K is the number of dimensions (signals);

$j = 1, 2, \dots, J$, where J is the number of segments.

After segmentation the classification problem is solved when segments are attributed to classes:

$g_j \rightarrow C$

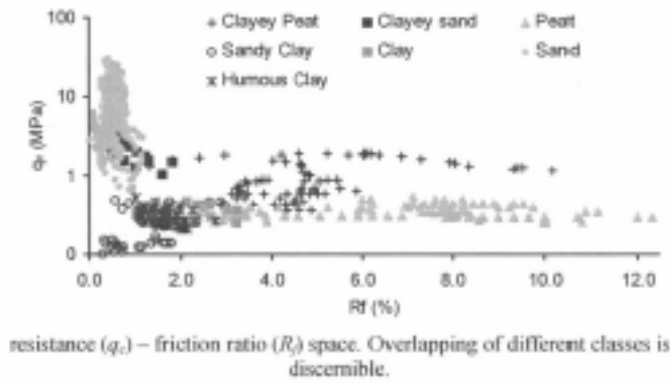
The number of possible partitions in this case is considerably smaller than the unconstrained clustering case. Accordingly, most of the available algorithms employ the exhaustive or semi-exhaustive search within this reduced search space.

D. Feature Extraction

After the segmentation of the image is performed by k-means in the above phase our next step is the feature extraction stage. This is the foremost step in this methodology. All the features that are required for us to classify the soil type and crop detection are done in this phase. A number of features like the texture, colour, intensity, saturation, hue, etc are extracted for detection of soil type. a filter known as Gabor Filter is implemented for feature extraction. Gabor Filter is a linear filter used for edge detection. Frequency and orientation representations of Gabor filter are similar to those of human visual system and they have been found to be particularly appropriate for texture representation and discrimination. A set of frequencies and orientation representations may be helpful for extracting useful features from the image. Also other features like entropy, standard deviance, mean, etc can be extracted using Gabor filter. The main and important feature of soil that is colour is needed to be extracted. Hence a measure called colour moments are used to differentiate images based on their features of colour. These provide a colour similarity between images which can be compared to the values of images indexed in the data base for tasks like image retrieval.

1)FEATURE EXTRACTION USING BOUNDARY ENERGY

After the segments are found the following task is to assign classes to the segments. However, for different locations the mapping (2) could be different due to the spatial variability. When data from several test locations are combined an overlap of instances of different classes is usually observed (Fig. 2). The spatial variability has several reasons such as site-specific conditions, location and depth, measuring instruments, etc. To solve the problem it needs to be brought to a higher dimension by bringing in additional features so that a partition of the input space is possible by a classifier for assigning a class to that subset of inputs.



Shapes of signals can be represented using multi-scale transforms such as Fourier transform, w-representation using Marr wavelet and Morlet wavelet, Gabor transform etc. [3]. These representations can be used to derive shape measures using multi-scale energy methods such as boundary energy, multi-scale wavelet energy, etc. In this research the boundary energy has been used in parameterising the shape effects.

2) Boundary energy

Boundary energy is defined as the amount of energy required to modify the shape of a contour to its lowest energy level (a circle), with the same perimeter as the original object. The concept of boundary energy originated from the theory of elasticity and was first applied in biological shape characterisation [19]. Since then it has been widely used as a global shape measure for classification of a variety of shapes. Boundary energy is defined as follows:

$$B_{\alpha,k} = \frac{1}{N} \sum_{n=0}^{N-1} c(\alpha, n)^2 \quad (3)$$

where $B_{\alpha,k}$ denotes the boundary energy of a signal at scale α along the dimension k , c is the curvature at point n , $n = 1, 2, \dots, N$; and N is the number of discrete observations. A detailed description on boundary energy can be found in [3].

It is evident that computation of boundary energy depends upon an accurate estimation of multi-scale curvature of a series data, which for a discrete signal is not an easy task. Commonplace computation techniques of curvature based on finite difference methods can lead to high errors, effectively thwarting the possibility of using boundary energy as a shape measure. In this regard a Fourier based

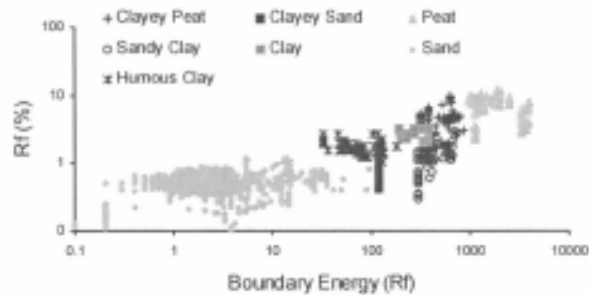


Fig. 3. Scatters of friction ratio (R_f) and the boundary energy of R_f using data from several Cone Penetration Tests. Several clusters of instances corresponding to classes are now more or less disjoint.

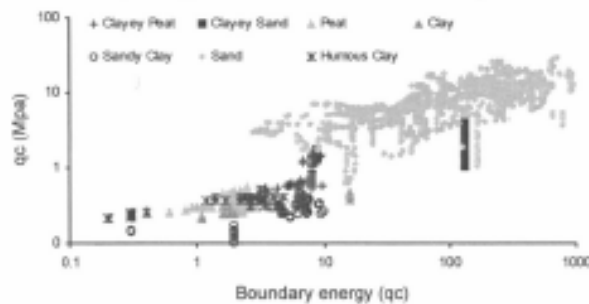


Fig. 4. Scatters of cone tip resistance (q_c) and the boundary energy of q_c using data from several Cone Penetration Tests. Several clusters of instances corresponding to classes are now more or less disjoint.

classification which is used to compared between different types of soil like clay,peat,sand etc.

E. SVM Classification

Support Vector Machine (SVM) algorithm is used for soil classification. It has successful applications in many fields like bioinformatics, text, image recognition, etc. SVM is a universally accepted algorithm due to its simple nature. It is considered as an alternative to neural networks algorithm. The working nature is explained as follows: Given a set of training examples, each marked for 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. SVM is effective to analyzing the separating planes and to identify the largest margin so that the support to the data points will be identified. So we want to learn the mapping: $X \rightarrow Y$, where 'x' belongs to X is some object and 'y' belongs to Y is a class label.

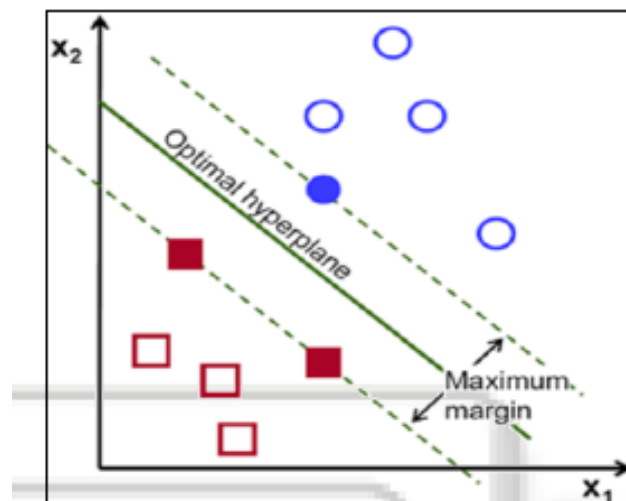


Fig. 3: classification by SVM

This algorithmic approach is based on distinctive characteristics analysis by analyzing the expected error minimization. This approach considered the empirical risk to improve the training procedure. The risk estimation is here based on the structural analysis so that the generalization error will be reduced. The error margin is analyzed under class deviation and based on it nearest training patterns are obtained. . This model is also based on the polynomial kernel representation so that the effective learning to the elements will be done and more accuracy will be obtained.

F. classification of object:

Classifiers are trained using DT, ANN and SVM to learn the following mapping from the labelled training data:

$$\{x/ 1, B,, 1-, -2, BI., Xl, K, BI, K\} C_i$$

Where

I = index of instalnces, $I= 1, n2, \dots, s.$, and n is the number of instances in g_j ;

K = number of signals (dimension);

$B_{ca k}$ = Boundary energy at point I in dimension k for input

The classifier (Eq. 4) lea asto classify each instance. Finally the mapping (2) is ensured by compacting the classes of instances within a segment. This is required due to t, deatothat frequently within a series data there could be instances (measurements) that obviously belong to the class C_1 , but are within the segment corresponding to another class C_2 and should be attributed to the class C_2 . Such points (vectors) could be the result of noise or instrumentation errors.

The weights (w_i) are computed as points on a Gaussian curve with zero mean and standard deviation = 2, and the compact class (C_p) is determined by

$$\bar{C}_p = w_i C_p / \sum_{i=1}^{n_i} w_i$$

where n_i = number of instances in a segment and C_p is the class for the i th instance..

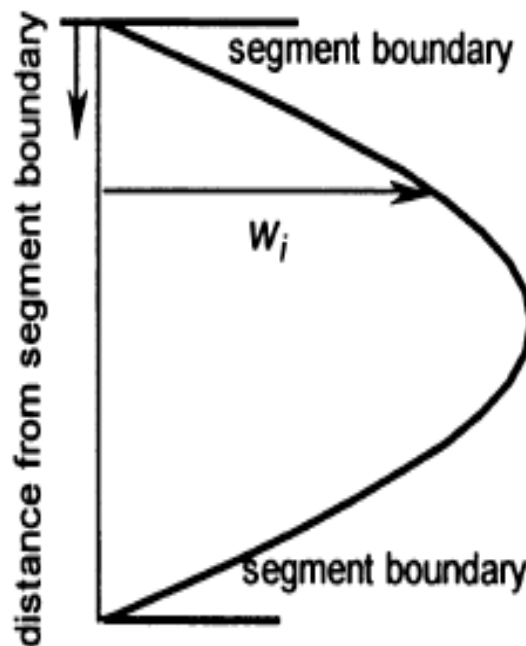


Fig. 6. The variation of weight w_i of an instance in relation to its position within a segment. The compaction algorithm of the classifier multiplies the class of each instance by w_i in order to determine a single class of a segment.

METHODOLOGY OF IMPLEMENTATION

In this project we have used MATLAB platform to analyze and classify various image of soil elements. The images of soil are passed through various techniques like image acquisition, image-pre processing, segmentation, feature extraction, classification and detection of soil.

Some of the important MATLAB functions that will be used are –

function colorAutoCorrelogram: This function creates the auto-correlogram vector for an input image of any size. The different distances which is assumed a priori can be user-defined in a vector. It implements the algorithm as defined in Huang et al. paper 'Image Indexing using color autocorrelogram'.

function colorMoments = colorMoments(image)
input: image to be analyzed and extract 2 first moments from each R,G,B
output: 1x6 vector containing the 2 first color momentst from each R,G,B channel

function colorMoments
: to calculate color moment from an color image

input: image to be analyzed and extract 2 first moments from each R,G,B
output: 1x6 vector containing the 2 first color momentst from each R,G,B channel

extract color channels

function [Feature_Vector]: Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as soil image matching and retrieval.

```
queryImage = imresize(queryImage, [256 256]);  
hsvHist = hsvHistogram(queryImage);  
autoCorrelogram = colorAutoCorrelogram(queryImage);  
color_moments = colorMoments(queryImage);  
%for gabor filters we need gray scale image  
img = double(rgb2gray(queryImage))/255;  
[meanAmplitude, msEnergy] = gaborWavelet(img, 4, 6); % 4 = number of scales,  
6 = number of orientations  
wavelet_moments = waveletTransform(queryImage);  
construct the queryImage feature vector  
Feature_Vector = [hsvHist autoCorrelogram color_moments meanAmplitude  
msEnergy wavelet_moments];
```

function hsvColorHistogram: The provided Matlab functions demonstrate a simple method for training and testing a color detection system. In general, a color detection algorithm searches an image for pixels that have a specific color. In this demo the HSV color space has been used, instead of the RGB space.

input: image to be quantized in hsv color space into 8x2x2 equal bins

output: 1x32 vector indicating the features extracted from hsv color

function f = lowpassfilter: Constructs a low-pass butterworth filter.

usage: f = lowpassfilter(size, cutoff, n)

where: size is a two element vector specifying the size of filter to construct [rows cols].

cutoff is the cutoff frequency of the filter 0 - 0.5

n is the order of the filter, the higher n is the sharper the transition is. (n must be an integer >= 1).

Note that n is doubled so that it is always an even integer.

$$f = \frac{1}{1.0 + (w/cutoff)^{2n}}$$

The frequency origin of the returned filter is at the corners.

function [itrfin] = multisvm(T,C,test):MULTISVM classifies the class of given training vector according to the given group and gives us result that which class it belongs.

We have also to input the testing matrix

Inputs: T=Training Matrix, C=Group, test=Testing matrix

Outputs: itrfin=Resultant class(Group,USE ROW VECTOR MATRIX) to which tst set belong

function varargout = SoilDetect_GUI(varargin):

SOILDETECT_GUI, by itself, creates a new SOILDETECT_GUI or raises the existing singleton*.

H = SOILDETECT_GUI returns the handle to a new SOILDETECT_GUI or the handle to the existing singleton*

SOILDETECT_GUI('CALLBACK',hObject,eventData,handles,...) calls the local function named CALLBACK in SOILDETECT_GUI.M with the given input arguments.

function waveletMoments = waveletTransform(image)

input: image to process and extract wavelet coefficients from

output: 1x20 feature vector containing the first 2 moments of wavelet coefficients

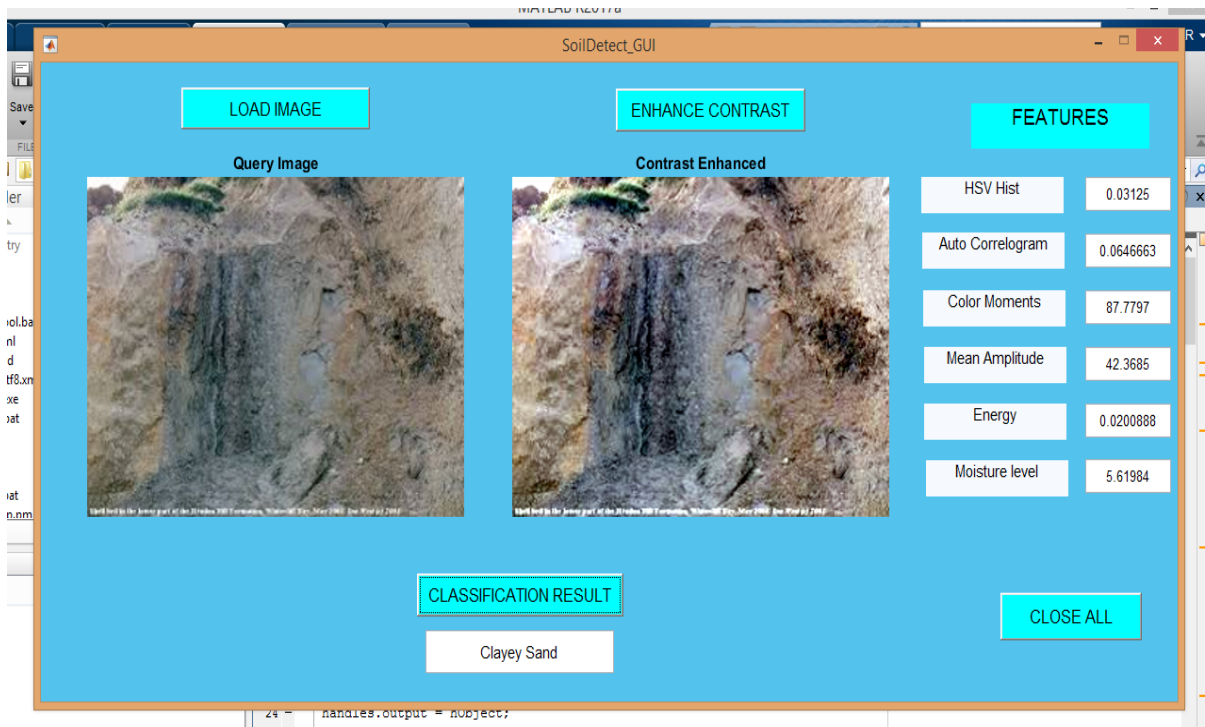
RESULTS AND SAMPLE OUTPUT:

The screenshot displays the 'SoilDetect_GUI' application window. The interface is divided into several sections:

- LOAD IMAGE**: A button to load the original image.
- ENHANCE CONTRAST**: A button to process the image with enhanced contrast.
- Query Image**: A preview of the original soil image.
- Contrast Enhanced**: A preview of the image after contrast enhancement.
- FEATURES**: A table of extracted features and their values.
- CLASSIFICATION RESULT**: A button to view the final classification.
- CLOSE ALL**: A button to close the application.

Feature	Value
HSV Hist	0.03125
Auto Correlogram	0.0622638
Color Moments	57.8808
Mean Amplitude	29.6013
Energy	0.0140935
Moisture level	3.50849

The classification result is **Silty Sand**.



SOIL DETECTION AND CLASSIFICATION RESULT OUTPUT WITH DIFFERENT SOIL LIKE CLAY,SAND,SITTY SAND ETC

CONCLUSION

The results from this study can be used for rapid identification of soil types when they arrive in railcars at the terminal soil elevators. Since this classification technique does not require time consuming image processing routines such as Fourier descriptors, it can readily be implemented using commercial imaging libraries with Digital Signal Processing (DSP) boards for real time operations. The work carried out has relevance to real world classification of soils and it involves both image processing and pattern recognition techniques. SVM involves the sound theory first and then implementation and experiments. A significant advantage of using SVM that the solution to an SVM is global and unique, they have simple geometric interpretations and gives sparse solution. The computational complexity of SVMs does not depend on the dimensionality of the input space and they are less prone to overfitting.

APPENDIX:

Program source code:

```
function varargout = SoilDetect_GUI(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @SoilDetect_GUI_OpeningFcn, ...
    'gui_OutputFcn', @SoilDetect_GUI_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function SoilDetect_GUI_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
handles.output = hObject;
ss = ones(300,400);
axes(handles.axes1);
imshow(ss);
axes(handles.axes2);
imshow(ss);
guidata(hObject, handles);

function varargout = SoilDetect_GUI_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)

clc
[filename, pathname] = uigetfile({'*.*'; '*.bmp'; '*.jpg'; '*.gif'}, 'Pick a Soil Image');
I = imread([pathname,filename]);
I = imresize(I,[256,256]);
I2 = imresize(I,[300,400]);
axes(handles.axes1);
imshow(I2);title('Query Image');
ss = ones(300,400);
axes(handles.axes2);
imshow(ss);
handles.ImgData1 = I;
guidata(hObject,handles);
```

```
function pushbutton2_Callback(hObject, eventdata, handles)
```

```
I3 = handles.ImgData1;  
I4 = imadjust(I3,stretchlim(I3));  
I5 = imresize(I4,[300,400]);  
axes(handles.axes2);  
imshow(I5);title(' Contrast Enhanced ');  
handles.ImgData2 = I4;
```

```
hsvHist = hsvHistogram(I4);  
autoCorrelogram = colorAutoCorrelogram(I4);  
color_moments = colorMoments(I4);
```

```
img = double(rgb2gray(I4))/255;  
[meanAmplitude, msEnergy] = gaborWavelet(img, 4, 6);  
wavelet_moments = waveletTransform(I4);
```

```
Feature_Vector = [hsvHist autoCorrelogram color_moments meanAmplitude  
msEnergy wavelet_moments];
```

```
whos Feature_Vector
```

```
F1 = mean2(hsvHist(:));  
F2 = mean2(autoCorrelogram(:));  
F3 = mean2(color_moments(:));  
F4 = mean2(meanAmplitude(:));  
F5 = mean2(msEnergy(:));  
F6 = mean2(wavelet_moments(:));
```

```
set(handles.edit3,'string',F1);  
set(handles.edit4,'string',F2);  
set(handles.edit5,'string',F3);  
set(handles.edit6,'string',F4);  
set(handles.edit7,'string',F5);  
set(handles.edit8,'string',F6);
```

```
handles.ImgData3 = Feature_Vector;  
guidata(hObject,handles);
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

```
load('TrainFeat_Soil.mat')
```

```
test = handles.ImgData3;  
result = multisvm(TrainFeat,Train_Label,test);  
disp(result);
```

```
if result == 1
```

```
    A1 = 'Clay';  
    set(handles.edit1,'string',A1);  
    helpdlg(' Clay ');  
    disp(' Clay ');
```

```
elseif result == 2
```



```
A2 = 'Clayey Peat';
set(handles.edit1,'string',A2);
helpdlg(' Clayey Peat ');
disp('Clayey Peat');
elseif result == 3
A3 = 'Clayey Sand';
set(handles.edit1,'string',A3);
helpdlg(' Clayey Sand ');
disp(' Clayey Sand ');
elseif result == 4
A4 = 'Humus Clay';
set(handles.edit1,'string',A4);
helpdlg(' Humus Clay ');
disp(' Humus Clay ');
elseif result == 5
A5 = 'Peat';
set(handles.edit1,'string',A5);
helpdlg(' Peat ');
disp(' Peat ');
elseif result == 6
A6 = 'Sandy Clay';
set(handles.edit1,'string',A6);
helpdlg(' Sandy Clay ');
disp('Sandy Clay');
elseif result == 7
A7 = 'Silty Sand';
set(handles.edit1,'string',A7);
helpdlg(' Silty Sand ');
disp(' Silty Sand ');
end
guidata(hObject,handles);

function pushbutton5_Callback(hObject, eventdata, handles)

close all
```

FUTURE WORK

Apart from soil detection and classification we can also analyze edge gap between different set of peat, soil pH level, moisture level, does the specific soil could be used in by digital image processing. Also we can add hardware support to ensure the best results with arduino and

develop an automatic system for a domestic environment that monitors and regulates agricultural factors.

REFERENCES

- ✚ Xiang Gao, Tancheng Lu, Peng Liu, Qiyong Lu,(2014), “A Soil Moisture Classification Model Based on SVM

- ✚ [Kumar . R, Singh.M.P, Kumar.P, Singh.J.P, “Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique”, (2015), Conf: on ICSTM, pp.138-145.

- ✚ Bhattacharya, and D.P. Solomatine, "An algorithm for clustering and classification of series data with constraint of contiguity", Proc. 3Trd nt. Conf: on Hybrid and Intelligent Systems, Melbourne, Australia, 2003, pp. 489-498.

- ✚ Gordon, A.D. "A survey of constrained classification", Computational Statistics & Data Analysis, vol. 21, pp. 17-29, 1996.

- S. Haykin, Neural Networks: A Comprehensive Foundation

- Neural Machine, <http://ivwww.data-machine.com/>, 28.1.2005.

- Soumya sree, Asst. Prof. A Rafega Beham, (2015), “Automated Diagnosis of Retina Images for Diabetic Patients based on BP and SVM”, proceedings of the International Journal of Computer Science and Mobile Computing, NHCE, Bangalore, pp.299-306.