CROP MONITORING AND RECOMMENDATION SYSTEM USING MACHINE LEARNING TECHNIQUES
A PROJECT REPORT

Submitted by

S KRISHNA PRASAD (2013503514)
B SIVA SREEDHARAN (2013503525)
S JAISHANTH (2013503565)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

MADRAS INSTITUTE OF TECHNOLOGY, CHENNAI
ANNA UNIVERSITY: CHENNAI 600 025

October 2016
Certified that this project report titled “CROP MONITORING AND RECOMMENDATION SYSTEM USING MACHINE LEARNING TECHNIQUES” is a bonafide work done by “S.KRISHNA PRASAD (2013503514), B.SIVASREEDHARAN (2013503525) and S.JAISHANTH (2013503565)” under my supervision, in partial fulfilment for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further, that to the best of my knowledge the work reported here in does not form part or full of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion to this or any other candidate.

Date: 

SIGNATURE

Dr. P.VARALAKSHMI
SUPERVISOR

Associate Professor,
Department of Computer Technology,
Madras Institute of Technology,
Anna University,
Chennai-600 044.
ACKNOWLEDGEMENT

We are highly indebted to our respectable Dean, Dr. A. RAJADURAI and to our reputable Head of the Department Dr. P. ANANDHAKUMAR, Department of Computer Technology, MIT Campus, Anna University for providing us with sufficient facilities that contributed to success in this endeavor.

We would like to express my sincere thanks and deep sense of gratitude to our Supervisor, Dr. P. VARALAKSHMI for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of this phase of project work.

We sincerely thank our project coordinators namely Dr. R. GUNASEKERAN, Dr. P. PABITHA and Ms. Y. NANCY JANE, Department of Computer Technology, MIT Campus, Anna University for their kind support and suggestions to carry out the project work.

We would be failing in our duty, if we forget to thank all the teaching and non-teaching staff of our department, for their constant support throughout the course of our project work.

S.KRISHNA PRASAD
B.SIVA SREEDHARAN
S. JAISHANTH
ABSTRACT

This document proposes a crop recommendation system using Convolutional Neural Network (CNN) and Support Vector Machine (SVM). The CNN takes in the image of the soil as the input and produces the soil class (type) as the output. The soil class together with geographic parameters like latitude and longitude is fed into SVM which produces the suitable crop as the output. Convolutional Neural Network is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. In simple words, it replicates the working of a human retina. Support Vector Machine is a typical classifier model. We have used SVM which predicts the best fit for the input with the help of Radial Basis Function (RBF) as the kernel. Further scope of the project would extend to predictive analytics on the commodity market of the goods grown in the agricultural fields to predict its waxing and waning. The remote sensing data can provide information of crop environment, crop distribution, and leaf area index (LAI), and crop phenology. This information is integrated in crop simulation models, in a number of ways such as use as direct forcing variable, use for recalibrating specific parameters, or use simulation-observation differences in a variable to correct yield prediction.
CHAPTER 1

INTRODUCTION

1.1 MACHINE LEARNING

Artificial Intelligence Machine Learning is a field of Computer Science, where new developments evolve at recent times, and also helps in automating the evaluation and processing done by the mankind, thus reducing the burden on the manual human power. Finding out the suitable crops based on the soil’s appearance becomes tedious for novice farmers. There also exists a need to prevent the agricultural decay. Effective utilization of agricultural land is crucial for ensuring food security of a country. In this document we proposed a crop recommendation system using Convolutional Neural Network (CNN) and Support Vector Machine (SVM).

1.1.1 Types of Problems and tasks

Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system. These are

- **Supervised learning**: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

- **Unsupervised learning**: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
• **Reinforcement learning**: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle), without a teacher explicitly telling it whether it has come close to its goal. Another example is learning to play a game by playing against an opponent.

1.2 **CONVOLUTIONAL NEURAL NETWORK**

CNN is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex, whose individual neurons are arranged in such a way that they respond to overlapping regions tiling the visual field. It consists of multiple layers of small neuron collections which process portions of the input image, called receptive fields. The outputs of these collections are then tiled so that their input regions overlap, to obtain a better representation of the original image; this is repeated for every such layer. A more detailed description of the layer of a Convolutional Neural Network can be found in the next section.

Figure 1.1. Structure of CNN
1.3 TYPICAL LAYER OF CNN

The typical layer of a Convolutional Neural Network are

- **Convolutional Layer**: it is the core building block of a Convolutional Network, and its output volume can be interpreted as holding neurons arranged in a 3D volume. The CONV layer’s parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. During the forward pass, we slide (more precisely, convolve) each filter across the width and height of the input volume, producing a 2-dimensional activation map of that filter. As we slide the filter, across the input, we are computing the dot product between the entries of the filter and the input. Intuitively, the network will learn filters that activate when they see some specific type of feature at some spatial position in the input. Stacking these activation maps for all filters along the depth dimension forms the full output volume. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at only a small region in the input and shares parameters with neurons in the same activation map (since these numbers all result from applying the same filter). Three hyper parameters control the size of the output volume: the depth, stride and zero-padding.

- **Pooling Layer**: it is common to periodically insert a Pooling layer in between successive Conv layers in a ConvNet architecture. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and
computation in the network, and hence to also control overfitting. The Pooling Layer operates independently on every depth slice of the input and resizes it spatially, using the MAX operation. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples every depth slice in the input by 2 along both width and height, discarding 75% of the activations. Every MAX operation would in this case be taking a max over 4 numbers (little 2x2 region in some depth slice). The depth dimension remains unchanged

- **Fully-connected layer:** neurons in a fully connected layer have full connections to all activations in the previous layer. Their activations can hence be computed with a matrix multiplication followed by a bias offset. In other libraries, fully connected blocks or layers are linear functions where each output dimension depends on all the input dimensions. MatConvNet does not distinguish between fully connected layers and convolutional blocks. Can freely and dynamically self-organize into arbitrary and temporary network topologies.

- **ReLU:** the rectified linear unit will apply an elementwise activation function, such as the \( f(x) = \max(0, x) \) thresholding at zero. This leaves the size of the volume unchanged

- **SoftMax:** Soft Max is a loss function makes the score compete through the normalization factor. SoftMax can be seen as the combination of an activation function (exponential) and a normalization operator,
1.4 CONVOLUTION

For simplicity we assume a grayscale image to be defined by a function
I : \{1,\ldots,n_1\} \times \{1,\ldots,n_2\} \rightarrow W \subseteq \mathbb{R}, (i, j) \mapsto I_{i, j}
such that the image I can be represented by an array of size \(n_1 \times n_2\). Given
the filter \(K \in \mathbb{R}^{2h_1+1 \times 2h_2+1}\), the discrete convolution of the image I is with
filter.

1.5 CNN CONCEPTS

CNNs have an associated terminology and a set of concepts that is
unique to them, and that sets them apart from other types of neural network
architectures. The main ones are explained as follows:

1.5.1 Input/Output Volumes

CNNs are usually applied to image data. Every image is a matrix of pixel
values. The range of values that can be encoded in each pixel depends upon its
bit size. Most commonly, we have 8 bit or 1 Byte-sized pixels. Thus the possible
range of values a single pixel can represent is \([0, 255]\). However, with coloured
images, particularly RGB (Red, Green, Blue)-based images, the presence of
separate colour channels (3 in the case of RGB images) introduces an additional
‘depth’ field to the data, making the input 3-dimensional. Hence, for a given
RGB image of size, say \(255 \times 255\) (Width x Height) pixels, we’ll have 3 matrices
associated with each image, one for each of the colour channels. Thus the image
in its entirety, constitutes a 3-dimensional structure called the Input Volume
\((255 \times 255 \times 3)\).
1.5.2 Features

Just as its literal meaning implies, a feature is a distinct and useful observation or pattern obtained from the input data that aids in performing the desired image analysis. The CNN learns the features from the input images. Typically, they emerge repeatedly from the data to gain prominence. As an example, when performing Face Detection, the fact that every human face has a pair of eyes will be treated as a feature by the system, that will be detected and learned by the distinct layers. In generic object classification, the edge contours of the objects serve as the features.
1.5.3 Filters (Convolution Kernels)

A filter (or kernel) is an integral component of the layered architecture. Generally, it refers to an operator applied to the entirety of the image such that it transforms the information encoded in the pixels. In practice, however, a kernel is a smaller-sized matrix in comparison to the input dimensions of the image, that consists of real valued entries.

The kernels are then convolved with the input volume to obtain so-called ‘activation maps’. Activation maps indicate ‘activated’ regions, i.e. regions where features specific to the kernel have been detected in the input. The real values of the kernel matrix change with each learning iteration over the training set, indicating that the network is learning to identify which regions are of significance for extracting features from the data.

1.5.4 Kernel Operations Detailed

The exact procedure for convolving a Kernel (say, of size 16 x 16) with the input volume (a 256 x 256 x 3 sized RGB image in our case) involves taking patches from the input image of size equal to that of the kernel (16 x 16), and convolving (or calculating the dot product) between the values in the patch and those in the kernel matrix.

The convolved value obtained by summing the resultant terms from the dot product forms a single entry in the activation matrix. The patch selection is then slided (towards the right, or downwards when the boundary of the matrix is reached) by a certain amount called the ‘stride’ value, and the process is repeated till the entire input image has been processed. The process is carried
out for all colour channels. For normalization purposes, we divide the calculated value of the activation matrix by the sum of values in the kernel matrix.

Note that:

- pixels are numbered from 1 in the example;
- the values in the activation map are normalized to ensure the same intensity range between the input volume and the output volume. Hence, for normalization, we divide the calculated value for the ‘red’ channel by 2 (the sum of values in the kernel matrix);
- we assume the same kernel matrix for all the three channels, but it is possible to have a separate kernel matrix for each colour channel;
- for a more detailed and intuitive explanation of the convolution operation

1.5.5 Receptive Field

It is impractical to connect all neurons with all possible regions of the input volume. It would lead to too many weights to train, and produce too high a computational complexity. Thus, instead of connecting each neuron to all possible pixels, we specify a 2 dimensional region called the ‘receptive field’ (say of size 5×5 units) extending to the entire depth of the input (5x5x3 for a 3 colour channel input), within which the encompassed pixels are fully connected to the neural network’s input layer. It’s over these small regions that the network layer cross-sections (each consisting of several neurons (called ‘depth columns’)) operate and produce the activation map.
1.5.6 Zero-Padding

Zero-padding refers to the process of symmetrically adding zeroes to the input matrix. It’s a commonly used modification that allows the size of the input to be adjusted to our requirement. It is mostly used in designing the CNN layers when the dimensions of the input volume need to be preserved in the output volume.

Figure 1.3: A zero-padded 4 x 4 matrix becomes a 6 x 6 matrix.

1.5.7 Hyperparameters

In CNNs, the properties pertaining to the structure of layers and neurons, such spatial arrangement and receptive field values, are called hyperparameters. Hyperparameters uniquely specify layers. The main CNN Hyperparameters are receptive field (R), zero-padding (P), the input volume dimensions (Width x Height x Depth, or W x H x D ) and stride length (S).

1.6 SUPPORT VECTOR MACHINE
The SVM module takes in the training soil class, longitude and latitude as the input and uses Radial Basis Function (RBF) as kernel, to find the best-fit curve for the given input data and produces Recommendation Model as output.

Support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of kernel function \( k(x,y) \) selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters of images of feature vectors that occur in the data base. With this choice of a hyperplane, the points in the feature space that are mapped into the hyperplane are defined by the relation: The equation of the output from a linear SVM is

\[
  u = w \cdot x - b;
\]

Where \( w \) is the normal vector of the hyperplane, and \( x \) is the input vector.
1.6.1 Advantages

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true of image segmentation systems, including those using a modified version SVM.
- Hand-written characters can be recognized using SVM.

1.7 REMOTE SENSING

Remote sensing is "the acquisition of information about an object, without being in physical contact with that object"

Types of remote sensing are as follows:

- **Passive**: The sensor records energy that is reflected or emitted from the source, such as light from the sun. This is also the most common type of system.

- **Active**: where the object is illuminated by radiation produced by the sensors, such as radar or microwaves.
1.8 NORMALIZED DIFFERENCE VEGETATION INDEX

- The normalized difference vegetation index (NDVI) is a simple graphical indicator that can be used to analyze remote sensing measurements, typically but not necessarily from a space platform, and assess whether the target being observed contains live green vegetation or not.

- Live green plants absorb solar radiation in the photosynthetically active radiation (PAR) spectral region, which they use as a source of energy in the process of photosynthesis. Leaf cells have also evolved to re-emit solar radiation in the near-infrared spectral region (which carries approximately half of the total incoming solar energy), because the photon energy at wavelengths longer than about 700 nanometers is not large enough to synthesize organic molecules. A strong absorption at these wavelengths would only result in overheating the plant and possibly damaging the tissues. Hence, live green plants appear relatively dark in the PAR and relatively bright in the near-infrared. By contrast, clouds and snow tend to be rather bright in the red (as well as other visible wavelengths) and quite dark in the near-infrared. The pigment in plant leaves, chlorophyll, strongly absorbs visible light (from 0.4 to 0.7 µm) for use in photosynthesis. The cell structure of the leaves, on the other hand, strongly reflects near-infrared light (from 0.7 to 1.1 µm). The NDVI is calculated from these individual measurements as follows:

\[
NDVI = \frac{(NIR - VIS)}{(NIR + VIS)}
\]

- where VIS and NIR stand for the spectral reflectance measurements acquired in the visible (red) and near-infrared regions, respectively
The NDVI has been widely used in applications for which it was not originally designed. Using the NDVI for quantitative assessments (as opposed to qualitative surveys as indicated above) raises a number of issues that may seriously limit the actual usefulness of this index if they are not properly addressed. The following subsections review some of these issues.

1.8.1 Limitation

- **Atmospheric effects:** The actual composition of the atmosphere (in particular with respect to water vapor and aerosols) can significantly affect the measurements made in space. Hence, the latter may be misinterpreted if these effects are not properly taken into account (as is the case when the NDVI is calculated directly on the basis of raw measurements).

- **Clouds:** Deep (optically thick) clouds may be quite noticeable in satellite imagery and yield characteristic NDVI values that ease their screening. However, thin clouds (such as the ubiquitous cirrus), or small clouds with typical linear dimensions smaller than the diameter of the area actually sampled by the sensors, can significantly contaminate the measurements. Composite NDVI images have led to a large number of new vegetation applications where the NDVI or photosynthetic capacity varies over time.

- **Soil effects:** Soils tend to darken when wet, so that their reflectance is a direct function of water content. If the spectral response to moistening is not exactly the same in the two spectral bands, the NDVI of an area can appear to change as a result of soil moisture changes (precipitation or evaporation) and not because of vegetation changes.
• **Anisotropic effects:** All surfaces (whether natural or man-made) reflect light differently in different directions, and this form of anisotropy is generally spectrally dependent, even if the general tendency may be similar in these two spectral bands. As a result, the value of NDVI may depend on the particular anisotropy of the target and on the angular geometry of illumination and observation at the time of the measurements, and hence on the position of the target of interest within the swath of the instrument or the time of passage of the satellite over the site.

• **Spectral effects:** Since each sensor has its own characteristics and performances, in particular with respect to the position, width and shape of the spectral bands, a single formula like NDVI yields different results when applied to the measurements acquired by different instruments.

### 1.9 LEAF AREA INDEX

Leaf area index (LAI) is a dimensionless quantity that characterizes plant canopies. It is defined as the one-sided green leaf area per unit ground surface area.

- Half of the total needle surface area per unit ground surface area
- Projected (or one-sided, in accordance the definition for broadleaf canopies) needle area per unit ground area
- Total needle surface area per unit ground area

LAI is used to predict photosynthetic primary production, evapotranspiration and as a reference tool for crop growth. As such, LAI plays an essential role in theoretical production ecology. An inverse exponential relation between
LAI and light interception, which is linearly proportional to the primary production rate, has been established

1.9.1 DETERMINING LAI

- **Direct methods:** Direct methods can be easily applied on deciduous species by collecting leaves during leaf fall in traps of certain area distributed below the canopy. The area of the collected leaves can be measured using a leaf area meter or an image scanner and image analysis software. The measured leaf area can then be divided by the area of the traps to obtain LAI. Alternatively, leaf area can be measured on a sub-sample of the collected leaves and linked to the leaf dry mass (e.g. via Specific Leaf Area, SLA cm²/g). That way it is not necessary to measure the area of all leaves one by one, but weigh the collected leaves after drying (at 60–80 °C for 48 h). Due to the difficulties and the limitations of the direct methods for estimating LAI, they are mostly used as reference for indirect methods that are easier and faster to apply.

- **Indirect methods:** Indirect methods of estimating LAI in situ can be divided in two categories:
  1. indirect contact LAI measurements such as plumb lines and inclined point quadrats.
  2. indirect non-contact measurements

1.9.2 Disadvantages of methods

The disadvantage of the direct method is that it is destructive, time consuming and expensive, especially if the study area is very large.
The disadvantage of the indirect method is that in some cases it can underestimate the value of LAI in very dense canopies, as it does not account for leaves that lie on each other, and essentially act as one leaf according to the theoretical LAI models. Ignorance of non-randomness within canopies may cause underestimation of LAI up to 25%, introducing path length distribution in the indirect method can improve the measuring accuracy of LAI.
CHAPTER 2

LITERATURE SURVEY

2.1 IMAGE DENOISING

Xiang Xu et.al. (2016) present a new technique based on multiple morphological component analysis (MMCA) that exploits multiple textural features for decomposition of remote sensing images. The proposed MMCA framework separates a given image into multiple pairs of morphological components (MCs) based on different textural features, with the ultimate goal of improving the signal-to-noise level and the data separability. A distinguishing feature of our proposed approach is the possibility to retrieve detailed image texture information, rather than using a single spatial characteristic of the texture. In this paper, four textural features: content, coarseness, contrast, and directionality (including horizontal and vertical), are considered for generating the MCs.

Hemant Kumar et. al. (2016) state how Hyperspectral unmixing is the process of estimating constituent endmembers and their fractional abundances present at each pixel in a hyperspectral image. A hyperspectral image is often corrupted by several kinds of noise. This work addresses the hyperspectral unmixing problem in a general scenario that considers the presence of mixed noise. The unmixing model explicitly takes into account both Gaussian noise and sparse noise. The unmixing problem has been formulated to exploit joint-sparsity of abundance maps. A total-variation-based regularization has also been utilized for modeling smoothness of abundance maps. The split-Bregman technique has been utilized to derive an algorithm for solving resulting
optimization problem. Detailed experimental results on both synthetic and real hyperspectral images demonstrate the advantages of proposed technique.

Gabriela Ghimpe teanu et.al. (2016) explain the image decomposition model that provides a novel framework for image denoising. The model computes the components of the image to be processed in a moving frame that encodes its local geometry (directions of gradients and level lines). Then, the strategy we develop is to denoise the components of the image in the moving frame in order to preserve its local geometry, which would have been more affected if processing the image directly. Experiments on a whole image database tested with several denoising methods show that this framework can provide better results than denoising the image directly, both in terms of Peak signal-to-noise ratio and Structural similarity index metrics.

2.2 IMAGE CLASSIFICATION

Xia Zhang et.al.(2016) developed a new crop classification method involving the construction and optimization of the vegetation feature band set (FBS) and combination of FBS and object-oriented classification (OOC) approach. In addition to the spectral and textural features of the original image, 20 spectral indices sensitive to the vegetation’s biological parameters are added to the FBS to distinguish specific vegetation. A spectral dimension optimization algorithm of FBS based on class-pair separability (CPS) is also proposed to improve the separability between class pairs while reducing data redundancy. OOC approach is conducted on the optimized FBS based on CPS to reduce the salt-and-pepper noise. The proposed classification method was validated by two airborne hyperspectral images. The first image acquired in an agricultural area
of Japan includes seven crop types, and the second image acquired in a rice breeding area consists of six varieties of rice. Results demonstrate that the proposed method can significantly improve crop classification accuracy and reduce edge effects, and that textural features combined with spectral indices sensitive to the chlorophyll, carotenoid, and Anthocyanin indicators contribute significantly to crop classification. Therefore, it is an effective approach for classifying crop species, monitoring invasive species, as well as precision agriculture related applications.

Xin Huang et. al. (2016) lists the differential morphological profiles (DMPs) that are widely used for the spatial/structural feature extraction and classification of remote sensing images. They can be regarded as the shape spectrum, depicting the response of the image structures related to different scales and sizes of the structural elements (SEs). DMPs are defined as the difference of morphological profiles (MPs) between consecutive scales. However, traditional DMPs can ignore discriminative information for features that are across the scales in the profiles. To solve this problem, we propose scalespan differential profiles, i.e., generalized DMPs (GDMPs), to obtain the entire differential profiles. GDMPs can describe the complete shape spectrum and measure the difference between arbitrary scales, which is more appropriate for representing the multiscale characteristics and complex landscapes of remote sensing image scenes. Subsequently, the random forest (RF) classifier is applied to interpret GDMPs considering its robustness for high-dimensional data and ability of evaluating the importance of variables. Meanwhile, the RF “out-of-bag” error can be used to quantify the importance of each channel of GDMPs and select the most discriminative information in the entire profiles.
Kunshan Huang et.al.(2016) proposed a novel spectral–spatial hyperspectral image classification method based on K nearest neighbour (KNN) which consists of the following steps. First, the support vector machine is adopted to obtain the initial classification probability maps which reflect the probability that each hyperspectral pixel belongs to different classes. Then, the obtained pixel-wise probability maps are refined with the proposed KNN filtering algorithm that is based on matching and averaging nonlocal neighbourhoods. The proposed method does not need sophisticated segmentation and optimization strategies while still being able to make full use of the nonlocal principle of real images by using KNN, and thus, providing competitive classification with fast computation. Experiments performed on two real hyperspectral data sets show that the classification results obtained by the proposed method are comparable to several recently proposed hyperspectral image classification methods.

2.3 CROP CLUSTERING

Wei Yao et.al. applies evaluates a modified Gaussian-test-based hierarchical clustering method for high resolution satellite images. The purpose is to obtain homogeneous clusters within each hierarchy level which later allow the classification and annotation of image data ranging from single scenes up to large satellite data archives. After cutting a given image into small patches and feature extraction from each patch, k-means are used to split sets of extracted image feature vectors to create a hierarchical structure. As image feature vectors usually fall into a high-dimensional feature space, we test different distance metrics, to tackle the “curse of dimensionality” problem. By using three
different synthetic aperture radar (SAR) and optical image datasets, Gabor texture and Bag-of-Words (BoW) features are extracted, and the clustering results are analyzed via visual and quantitative evaluations.

Michael D. Johnson et.al elicits Crop yield forecast models for barley, canola and spring wheat grown on the Canadian Prairies were developed using vegetation indices derived from satellite data and machine learning methods. Hierarchical clustering was used to group the crop yield data from 40 Census Agricultural Regions (CARs) into several larger regions for building the forecast models. The Normalized Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI) derived from the Moderate-resolution Imaging Spectroradiometer (MODIS), and NDVI derived from the Advanced Very High Resolution Radiometer (AVHRR) were considered as predictors for crop yields. Multiple linear regression(MLR) and two online machine learning models – Bayesian neural networks (BNN) and model-based recursive partitioning (MOB) – were used to forecast crop yields, with various combinations of MODIS-NDVI, MODIS-EVI and NOAA-NDVI as predictors.

R.B. Arango et.al. (2016) focuses on a methodology for the automatic delimitation of cultivable land by means of machine learning algorithms and satellite data. The method uses a partition clustering algorithm called Partitioning Around Medoids and considers the quality of the clusters obtained for each satellite band in order to evaluate which one better identifies cultivable land. The proposed method was tested with vineyards using as input the spectral and thermal bands of the Landsat 8 satellite. The experimental results show the
great potential of this method for cultivable land monitoring from remote-sensed multispectral imagery.

2.4 WAVELET TRANSFORMS

M. Krishna Satya Varma et.al.(2016) state and create thematic maps of the land wrap present in an image, the classified data thus obtained may then be used. Classification includes influential an appropriate classification system, selecting, training sample data, image pre-processing, extracting features, selecting appropriate categorization techniques, progression after categorization and precision validation. Aim of this study is to assess Support Vector Machine for efficiency and prediction for pixel-based image categorization as a contemporary reckoning intellectual technique. Support Vector Machine is a classification procedure estimated on core approaches that was demonstrated on very effectual in solving intricate classification issues in lots of dissimilar appropriated fields. The latest generation of remote Sensing data analyzes by the Support Vector Machines exposed to efficient classifiers which are having amid the most ample patterns.

Olivier Regniers et.al.(2016) explore the potentialities of using wavelet-based multivariate models for the classification of very high resolution optical images. A strategy is proposed to apply these models in a supervised classification framework. This strategy includes a content-based image retrieval analysis applied on a texture database prior to the classification in order to identify which multivariate model performs the best in the context of application. Once identified, the best models are further applied in a supervised classification procedure by extracting texture features from a learning database.
and from regions obtained by a presegmentation of the image to classify. The classification is then operated according to the decision rules of the chosen classifier. The use of the proposed strategy is illustrated in two real case applications using Pléiades panchromatic images: the detection of vineyards and the detection of cultivated oyster fields. In both cases, at least one of the tested multivariate models displays higher classification accuracies than gray-level co-occurrence matrix descriptors.

N. Prabhu et.al. (2016 have presented a series of experiments to investigate the effectiveness of some wavelet based feature extraction of hyperspectral data. Three types of wavelets have been used which are Haar, Daubechies and Coiflets wavelets and the quality of reduced hyperspectral data has been assessed by determining the accuracy of classification of reduced data using Support Vector Machines classifier. The hyperspectral data has been reduced upto four decomposition levels. Among the wavelets used for feature extraction Daubechies wavelet gives consistently better accuracy than that produced from Coiflets wavelet. Also, 2-level decomposition is capable of preserving more useful information from the hyperspectral data. Furthermore, 2-level decomposition takes less time to extract features from the hyperspectral data than 1-level decomposition.

2.5 PREDICTION ALGORITHM

Monali Paul et.al.(2016) analysis the Soil Behaviour and Prediction of Crop Yield using Data Mining Approach. Yield prediction is very popular among farmers these days, which particularly contributes to the proper selection of crops for sowing. This makes the problem of predicting the yielding of crops an interesting challenge. Earlier yield prediction was performed by
considering the farmer's experience on a particular field and crop. This work presents a system, which uses data mining techniques in order to predict the category of the analyzed soil datasets. The category, thus predicted will indicate the yielding of crops. The problem of predicting the crop yield is formalized as a classification rule, where Naive Bayes and K-Nearest Neighbor methods are used.

Yvette Everingham et.al.(2016) predict the accurate prediction of sugarcane yield using a random forest algorithm. A data mining method like random forests can cope with generating a prediction model when the search space of predictor variables is large. Research that has investigated the accuracy of random forests to explain annual variation in sugarcane productivity and the suitability of predictor variables generated from crop models coupled with observed climate and seasonal climate prediction indices is limited. Simulated biomass from the APSIM (Agricultural Production Systems sIMulator) sugarcane crop model, seasonal climate prediction indices and observed rainfall, maximum and minimum temperature, and radiation were supplied as inputs to a random forest classifier and a random forest regression model to explain annual variation in regional sugarcane yields.

Benjamin Dumont et.al.(2016) predict the Assessing the potential of an algorithm based on mean climatic data to predict wheat yield. This paper presents a methodology that addresses the problem of unknown future weather by using a daily mean climatic database, based exclusively on available past measurements. It involves building climate matrix ensembles, combining different time ranges of projected mean climate data and real measured weather
data originating from the historical database or from real-time measurements performed in the field. Used as an input for the STICS crop model, the datasets thus computed were used to perform statistical within-season biomass and yield prediction. This work demonstrated that a reliable predictive delay of 3–4 weeks could be obtained. In combination with a local micrometeorological station that monitors climate data in real-time, the approach also enabled us to (i) predict potential yield at the local level, (ii) detect stress occurrence and (iii) quantify yield loss (or gain) drawing on real monitored climatic conditions of the previous few days.
CHAPTER 3

PROPOSED WORK

3.1 SYSTEM ARCHITECTURE

The overall block diagram of the system is shown in the figure 3.1. The image processor is implemented using OpenCV package in python. The image pre-processor and the image segmentation does the same task of segmenting the image to get a vivid soil portion from the image, as it may contain unwanted portions which may make the system to work with decreased efficiency. The CNN module takes the pre-processed image as input and finds the type of the soil and returns the soil class, which is then given to SVM along with latitude and longitude values. The system eventually aims predicting and thereby recommending the crops suitable based on input parameters specified.

3.2 USER INTERFACE

A simple and easy to use User Interface is to be designed for the system using the HTML and Bootstrap. The UI contains a ’choose file’ field which is used to choose the image file and two text areas for latitude and longitude inputs. On clicking the ’Submit’ button, the soil class and suitable crops are displayed in the second window.
3.3 MODULE DESIGN

3.3.1 IMAGE PRE-PROCESSOR (or) IMAGE SEGMENTATION

The pre-processor uses the package OpenCV to segment the images by forming contours. This works by finding the possible contours with
the specification to identify the soil areas. Then the image is segmented by keeping track of the formed contour areas.

Figure 3.2  Flow Diagram

3.3.2 CNN

CNN would take place in the pre-processed training images as input and tries to replicate the human-retinal functionality, building a Image Classifier Model. As this involves complex learning and classification process, this is planned to be run with the help of a GPU, with the support of Cuda, for faster execution. This consists of a number of neurons arranged in layers, which
contribute to the efficiency. Increase in depth of the layers significantly improves the efficiency. Efficiency can also be improved by altering the parameters of the input image.

3.3.3 SVM

The SVM module would take in the training soil class, longitude and latitude as the input and uses Radial Basis Function (RBF) as kernel, to find the best-fit curve for the given input data and produces Recommendation Model as output.

3.3.4 IMAGE CLASSIFIER EVALUATOR

This module takes in the pre-processed input image as the and uses the pickled Image Classifier Model to predict the soil class.

3.3.5 RECOMMENDATION MODEL EVALUATOR

This module takes in the soil class and GPS parameters as input and also uses the Recommendation Model to recommend the crop suitable for the data provide

3.3.6 CALCULATING NDVI

The normalized difference vegetation index (NDVI) is a simple graphical indicator that can be used to analyse remote sensing measurements and assess whether the target being observed contains live green vegetation or not.
NDVI VS LAI RELATIONSHIP CALCULATION

Accurate estimation of LAI is important for monitoring vegetation dynamics, and LAI information is essentially required for the prediction of microclimate and various biophysical processes within and below canopy.

\[(\text{LAI} = 0.128 \times \exp(\text{NDVI}/0.311))\]

3.3.7 YIELD CALCULATION BASED ON NDVI

The relationship between NDVI and yield of the data analysed, indicates the possibility of considering agrometeorological conditions to obtain accuracy in yield estimation.
CHAPTER 4
IMPLEMENTATION

4.1 SYSTEM DEVELOPMENT

This system has been developed using WinPython 2.7 which includes the packages such as:

- NumPy
- SciPy
- Pandas
- Theano
- Sklearn
- Matplotlib
- OpenCV

The overview of the algorithm for the system is as given below.

Initially Img is assigned the input image, Lon is assigned with longitude and Lat is assigned with latitude. This is fed to SVM classifier, which in turn invokes the CNN module to predict the soil class. Then the soil class together with Lat and Lon are used to predict the suitable crop.

| Img ←Input Soil Image |
| Lon ←Longitude |
| Lat ←Latitude |

PREDICT(Img, Lon, Lat):
1. soil class ←IMAGE CLASSIFIER EVALUATOR(Img)
2. suitable crop ←RECOMMENDATION MODEL EVALUATOR(soil class, Lon, Lat)
4.2 ALGORITHM AND WORKING OF EACH MODULE

4.2.1 CONVOLUTIONAL NEURAL NETWORKS

Description: Training the image classifier with training set

Input: Training set containing images various of soil classes and appropriate labels

Output: Image Classifier Model

- **CNN()**

1. DS ← dataset of soil images with soil classes as labels
2. CNN ← column of CNN for nnet in CNN
3. Weights of CNN are initialized randomly
4. for row in DS.rows
5. for nnet in CNN
6. Nnet.forward propagate(row)
7. nnet.back propagate ()

4.2.2 SUPPORT VECTOR MACHINE

Description: Training the Recommendation system with training set

Input: Training set containing suitable crops for given soil class and Latitude and longitude parameters

Output: Recommendation Model Evaluator
SVM()

1. DS ←dataset of soil images with soil classes as labels
2. svm[i] ←one vs rest Support Vector Machine for class i
3. for row in D
4. SVM[row.class].train(row)

4.2.3 IMAGE PRE-PROCESSOR

Description: Pre-process the input image so as to make it suitable for classification using image classifier

Input: Soil Image
Output: Processed soil image

IMAGE PREPROCESSOR()

1. I ←input image
2. Proc image ←selective search(I)
3. return Proc image
4.2.4 IMAGE CLASSIFIER EVALUATOR

**Description:** Evaluate the processed soil image to identify the soil class using image classifier

**Input:** Processed Soil Image

**Output:** Soil class of the image

- **IMAGE CLASSIFIER EVALUATOR** (Img)

  1. proc image ← preprocessed image
  2. CNN ← column of Convolutional Neural Networks
  3. N ← number of CNNs in the column
  4. $Y_i^{\text{CNN}}$ ← Prediction of a CNN for the class i
  5. $Y_i$ ← Prediction of MCDNN for the class i
  6. $Y_i ← 1/N(\sum_i Y_i^{\text{CNN}})$
  7. return $\text{argmax}(Y_i)$

4.2.5 RECOMMENDATION MODEL EVALUATOR

**Description:** Evaluate the soil class obtained from the image classifier and data obtained from GPS information using the Recommendation model

**Input:** Feature vector containing features like soil class, terrain, slope, elevation, etc
Output: a list of recommended crops

RECOMMENDATION MODEL EVALUATOR( soilclass, Lon, Lat )

1. X ←input feature vector
2. SVMi ←one vs rest SVM for class i
3. return argmaxi(SVMi .predict(X))

4.2.6 DEPLOYMENT DETAILS

The deployment of the system requires Graphical Processing Unit (GPU) and Windows 10 (or) 8.1 operating system. Graphical driver Cuda should also be present. The system must also be installed with Python 2.7 (or) 3.4. Any IDE like Pycharm can be used to deploy the system successfully.
CHAPTER 5

RESULTS AND ANALYSIS

5.1 DATASET FOR TESTING

5.1.1 FOR IMAGE CLASSIFIER

For the image classifier, the test data consists of hundred images belonging to different soil class collected manually from the internet.

5.1.2 FOR SVM

For the SVM classifier, non-exhaustive ten-fold cross validation method is used.

5.1.3 IMAGE CLASSIFIER

The input to the image classifier is a soil image. It predicts the soil class for a given image which is fed into the SVM along with latitude and longitude values

5.1.4 SVM

The input to the SVM is a tuple containing soil class, latitude and longitude values using which it recommends a crop
5.1.5 DENOISING

Figure 5.1 Denoising using PSNR Algorithm
Figure 5.2 Sample I/P and O/P

Band-2

Band-3
Figure 5.3 NDVI Bands
6.1 SUMMARY

This machine recommends the suitable crop given an image of the soil and the parameters like latitude and longitude, with classification of the soil class intermediate. The system builds up an Image Classifier Model, using CNN, which acts as an image classifier builder. The Image Pre-processor tries to get a maximum contour area out of the given soil image. This could then ease the work of Image Classifier Evaluator, to predict the soil class with improved accuracy. The accuracy of the system increases with increase in the number of neuron layers. But, there also exists a trade off between increase in the number of neuron layers and the time taken to train the system.

6.2 CRITICISMS

The predicted lack of accuracy in the pre-processing module propagates down to the CNN and the Image Classifier Evaluator modules, leading to inaccurate predictions of the soil class. This in turn may produce a downfall for the Recommendation module, since it takes in the soil class into account for the prediction of the suitable crop effectively. The lack of the readily available dataset, containing the latitude, longitude, soil class and suitable crop also significantly accounts for its inaccuracy, since the SVM could not find a best-fit curve with limited number of data available.
CHAPTER 7
FUTURE WORKS

The efficiency of the pre-processing is limited by the amount of unwanted information (like leaves, grass and other stuffs) present in it. Due to this undesirable information present in the input image, both during training and classification, the pre-processor fails to identify the exact contours, thus failing to perform with improved efficiency. Collection of more valid details of soil class, latitude, longitude and suitable crop can greatly accelerate the efficiency of work. The pre-processing unit could hence be improved and a lot more features can be extended, thus significantly contributing towards the agricultural welfare worldwide.
APPENDIX-1

IMPLEMENTATION TOOL

➢ PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

➢ MATLAB

MATLAB (matrix laboratory) is multi-paradigm numerical computing environment and fourth-generation programming language. A proprietary programming language developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python. Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.
REFERENCES


