Supervised and unsupervised learning for plant and crop row detection in precision agriculture

by

Varun Varshney

B.Tech., Jaypee Institute of Information Technology, 2014

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2017

Approved by:

Major Professor
Dr. William H. Hsu

Copyright

Abstract

The goal of this research is to present a comparison between different clustering and segmentation techniques, both supervised and unsupervised, to detect plant and crop rows. Aerial images, taken by an Unmanned Aerial Vehicle (UAV), of a corn field at various stages of growth were acquired in RGB format through the Agronomy Department at the Kansas State University. Several segmentation and clustering approaches were applied to these images, namely K-Means clustering, Excessive Green (ExG) Index algorithm, Support Vector Machines (SVM), Gaussian Mixture Models (GMM), and a deep learning approach based on Fully Convolutional Networks (FCN), to detect the plants present in the images. A Hough Transform (HT) approach was used to detect the orientation of the crop rows and rotate the images so that the rows became parallel to the x-axis. The result of applying different segmentation methods to the images was then used in estimating the location of crop rows in the images by using a template creation method based on Green Pixel Accumulation (GPA) that calculates the intensity profile of green pixels present in the images. Connected component analysis was then applied to find the centroids of the detected plants. Each centroid was associated with a crop row, and centroids lying outside the row templates were discarded as being weeds. A comparison between the various segmentation algorithms based on the Dice similarity index and average run-times is presented at the end of the work.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank all the people who contributed in some way to the work described in this thesis. First and foremost, I thank my academic advisor, Professor William H. Hsu, for accepting me into his group. During my tenure, he contributed to a rewarding graduate school experience by giving me intellectual freedom in my work, engaging me in new ideas, and demanding a high quality of work in all my endeavors. Additionally, I would like to thank my committee members Professor Mitchell L. Neilsen, and Professor Torben Amtoft for their interest in my work.

I would also like to thank Dr. Ignacio A. Ciampitti, and Sebastian Varela of the Department of Agronomy at K-State for giving me the opportunity to work on this project. I would like to thank all the members of the KDD group with whom I had the opportunity to work. They provided a friendly and cooperative atmosphere, and high-quality seminars that helped me gain insights into topics beyond my research. I would be remiss if I did not thank Sheryl Cornell, who deserves credit for providing much needed assistance with administrative tasks, reminding us of impending deadlines, and keeping our work running smoothly. I am grateful for the funding sources that allowed me to pursue my graduate school studies: the opportunities I had under Professor Robby, Dr. Dennis J. Lang, and Professor Daniel A. Andresen to serve as a teaching assistant.

Finally, I would like to acknowledge friends and family who supported me during my time here. I would like to thank Mom, Dad, my sister, Parul, my cat, Simba, and my uncle and

aunt, Suren and Shabina, for their constant love and support. Swapnil Kumar, and Lauren Lynch made my time here at K-State a lot more fun. I am lucky to have friends like Pratik Mehra, and Rishabh Chaudhary back home in New Delhi, and I thank them for their advice, friendship, love, and unyielding support.

Chapter 1 - Introduction

The application of computer science and automation to the field of agriculture is a crucial part of precision agriculture. Processes like planting of crop seeds in a field through machinery, fertilization, spraying of pesticides to preserve the crops, and ultimately harvesting, all have the properties of being repetitive and precise in nature. These properties make these processes suitable for automation without much human intervention and labor.

To automate guidance systems for machinery like tractors, and harvesters, plant and crop row detection is an important step. The separation of crop plants from weeds is crucial for site-specific pesticide treatments and curbing excessive use of chemicals that could result in destroying the crop plants in addition to weeds.

The detection of plants and crop rows is affected by several environmental conditions. Shadows and poor lighting conditions affect image quality. Weeds with similar spectral signatures can also be present in the field, interspersed between the crop plants, making the process of separation a challenge, thereby producing erroneous crop rows. The terrain and topography of the field also affect the image quality and pose a challenge for machinery to navigate through. Different plant heights and volumes due to difference in growth stages can also create severe problems.

Under the above mentioned adverse environmental conditions and even with irregular inter-row spaces, supervised and unsupervised methods were compared for the segmentation of the crop field images to detect plants, thereby facilitating the demarcation of straight crop rows in the field. The proposed strategy exploits the performance of some partial procedures involved in the above referenced methods and includes new procedures integrated in the global strategy to achieve a valid procedure for the maximum number of situations in crop row detection. The main considerations were:

1. The images to be analyzed are free of perspective projections. In other words, the images are taken from a top-down view (for example, by a drone).
2. Monocular RGB camera is sufficient to capture the images. There is no need of stereo vision.
3. Number of crop rows in the image can be unknown at the time of analysis.
4. Weeds, straw, and hay can be present in the images along with the crop plants.
5. The crop rows to be detected must be parallel to each other and the orientation angle of the crop rows should lie between -45 degrees and +45 degrees with respect to the x-axis.

The following methods have been employed and compared for image segmentation:

1. K-Means clustering algorithm
2. Gaussian Mixture Models (GMM)
3. Support Vector Machines (SVM)
4. Excess Green Index (ExG) algorithm
5. Fully Convolutional Networks for image segmentation (deep learning)

The result of the above segmentation approaches in each case is then evaluated to find the crop rows. The detected plants and crop rows are validated against hand-labeled ground truth data to determine the accuracy and performance is evaluated on running time.

Chapter 2 - Background and Related Work

Generally speaking, the segmentation of the images to separate plants from other objects is based on the Excess Green Index algorithm. The basic workflow of this approach is shown in Figure 1.

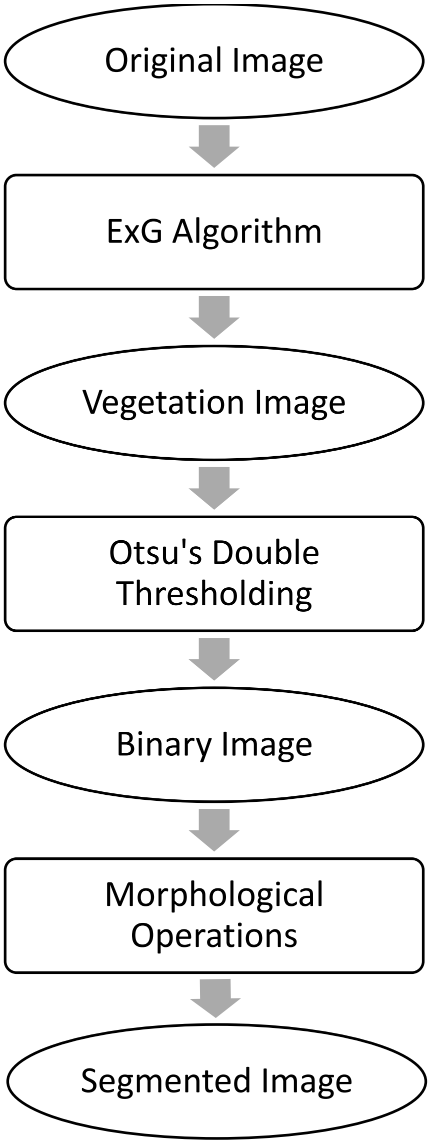


Figure 1. General workflow for vegetation segmentation in precision agriculture.

Crop row detection is an important problem precision agriculture. Many approaches have been proposed to solve this problem. Broadly, they are classified into the following groups: methods based on the Hough transform, methods based on linear regression, methods based on green pixel accumulation, filter based methods, and stereo vision based methods.

2.1 Hough Transform based methods

The most commonly used method for crop row detection is the Hough transform [1]. In [2], the authors present a method for crop row detection based on the Hough transform and prior knowledge about the number of crop rows and the row structure. Usage of prior knowledge makes the method tolerant of missing plants and the presence of weeds. The method tracks peaks in Hough space instead of searching for them. It is assumed that the number of crop rows is known in advance. Theoretical positions of crop rows are used for creating a reference in the Hough plane. Using this idea, all lines in the original image representing parallel lines in the scene vote for almost the same point in the Hough plane and thus the problem of finding peaks in the Hough plane is made easier. The reference line position is assumed to be known in advance. In [3], the authors also use the Hough transform for crop row detection. Three different transformations are tested for transformation of captured color images to grayscale images. On the obtained grayscale image, image segmentation is applied and the segmented image is divided into three vertical segments. These segments are merged into one image by normalization of summed pixel values. The Hough transform is applied to the merged image and only one crop row is obtained, which is transformed to three crop rows in the original image by knowing row spacing. The authors state that linear structures in the soil surface in a real field might create problems in crop row detection. Furthermore, the method can detect only three crop rows and the camera has to be set so as to capture only three crop rows. In [4], a method based on the Randomized Hough transform for the center line of a single crop row is proposed. The method is tested on sparse, general and intensive plant distribution, respectively. The obtained results show that the proposed method is faster and more accurate than the standard Hough transform. Since the proposed method has not been tested for more than one crop row, its application to real crop row detection is unknown. The Hough transform or adaptations of the Hough transform for crop row detection are used in several other papers [5], [6], [7] and [8].

2.2 Linear Regression based methods

In [9], a method for automatic crop row detection in maize fields in the presence of weeds is presented. The Excess Green index [10] is used and then double Otsu thresholding is applied for separating crops from weeds and soil. After image segmentation, crop rows are detected using least squares linear regression. Prior knowledge about the expected location of each crop row in the image and the area to be explored in the image are combined into templates used to select the area of the image linear regression is applied to. A new template must be created for each field arrangement. The current implementation of the method can detect only three crop rows. The method is compared favorably with the standard Hough transform and the authors state that double thresholding is the main contribution of this paper. The method presented in [11] is a combination of different color vegetation indices, Otsu thresholding [12] and linear regression. By using a combination of color vegetation indices, greenness of crops and weeds is emphasized in relation to soil, stones and other materials. Otsu thresholding is applied to the transformed image to create a binary image which is used in further processing. The expected crop rows are obtained by using the knowledge about the arrangement of crop rows and camera parameters. The expected crop rows are corrected by separately applying the Theil–Sen estimator to the data points representing crops and weeds around each expected crop row. Like the method presented in [9], this method has limited application to different crop row arrangements.

2.3 Green Pixel Accumulation based methods

A method based on an infrared image and summation of gray pixels for crop row detection is presented in [13]. An infrared filter is used to avoid image segmentation, because crops and weeds are brighter than soil for wavelengths between 700 and 1400 nm [13]. Summation of gray pixels is carried out for every column in a rectangular image processing area and after that, sinusoidal fitting or low-pass filtering is performed. The maximum of the fitted sinusoidal curve or the maximum of the filtered sum-curve represents the crop row positions. In the reported experiments, the camera was positioned orthogonally and the acquired images were free of perspective projection. Furthermore, a crop row angle greater than 15° and the presence of high weeds can cause error. In [14], the authors present a real-time method for crop row detection in maize field based on fuzzy clustering and accumulation of green pixels. Fuzzy clustering is used in an offline process to obtain the threshold value used for segmentation of crops and weeds from the ground in an online process. After image segmentation, crop rows are detected by searching for the maximum accumulation of green pixels along straight line segments. The authors used general knowledge about the maize field to achieve a short processing time. It is assumed that the number of crop rows is known in advance. Furthermore, the method can only detect straight line segments and crop rows which start from the bottom of the image. The method is compared with the standard Hough transform, according to effectiveness and processing time. Different image resolutions were tested and the proposed method outperformed the Hough transform in all cases. In [15], the authors present a method for crop row detection based on accumulation of green pixels and weighted linear regression. A captured image is transformed into a grayscale image using the Excess Green index and this transformed image is split into approximately 15 horizontal strips. For each strip, the center of gravity is calculated according to the sum of pixel values in columns. The center line of the guiding row is calculated by using the obtained centers and weighted linear regression. The approximate inter-row distance must be known and it is used in the calculation of the gravitation center. The method is tested on a barley field and the obtained results show that the central point of the detected guiding row can be estimated with the accuracy of about 6–12 mm.

2.4 Filter based methods

An initial trial of automatic hoe guidance based on computer vision is presented in [16]. The method is based on the extended Kalman filter and very simple image features obtained from near-infrared thresholded images. The prediction of the system state is calculated using a process model, the previous state and the inputs of the Kalman filter. This prediction is corrected by new observations. The images used for evaluation are without shadows and, according to the reported guidance performance, the authors state that the method is not commercially acceptable. In [17], the authors extended their system from their earlier work [16] by substituting image thresholding with bandpass filtering on near-infrared images. According to a periodic pattern in image rows due to parallel crop rows, the authors created a bandpass filter and used it to filter eight lines of each captured image. Maximal values of these filtered lines along with kinematic information about the tractor are used in the extended Kalman filter to estimate the desired hoe position. The obtained maximal filtered values are processed sequentially with the Kalman filter, which means that the previous results have an impact on the current result. The method is tested on a field of winter wheat and the authors reported a mean error of 15.6 mm. Unlike earlier attempts by these authors, this method is not sensitive to the presence of shadows. Adaptation of this method for mechanical weed control in sugar beet is presented in [18]. Because of the space between sugar beet plants in one crop row, the image is separated into eight bands and filtered lines from each band are merged. It is important to ensure that each band contains a part of a plant at each individual crop row. In [18], the authors proposed two different bandpass filters. One is the same filter used in [17], but in [18] it is used for large plants. The other is a truncated version of the same filter, which is used for small plants. Because of possible usage of two different filters, the operator must specify if plants are small or large, which makes the method less automatic. Experimental results of the proposed method show an average error of 16 mm

2.5 Stereo Vision based methods

A method for crop row detection and automatic machine guidance based on stereo vision is presented in [19]. Crops are represented by a set of 3D points computed from disparity images obtained by stereo vision. From the calculated 3D points, an elevation map is created which is filtered to fill the missing pixels. Navigation points are determined using cross-correlation between the cross-section of the elevation map and the designed crop model. The proposed method was tested on straight and curved soybeans crop rows and the authors reported that the root mean square error was below 0.05 m. In [20], the authors present lateral offset measurement for machine navigation in cultivated stubble fields based on stereo vision. The Harris corner detector is used for feature detection and the normalized cross-correlation method is used to find matching features in consecutive images. The random sample consensus (RANSAC) method is used to remove outliers and a lateral offset is calculated according to 3D coordinates of correctly paired features. The proposed algorithm is tested on straight and curved paths. Maximum absolute deviation for the straight paths is less than 50 mm, while the accuracy is lower for curved paths. The authors state that the method could be useful for guiding a machine on straight paths in cultivated stubble crop fields, while some improvements are needed for curved paths.

Chapter 3 - Methodology

The approach proposed in this paper compares several algorithms to segment the images for detection of plants. Moreover, with respect to crop row detection, a new strategy based on Green Pixel Accumulation and Template Creation is also discussed wherein the number of crop rows to be detected in the image, the expected location of each crop row, and the image region to be explored do not have to be specified in advance by the user.

3.1 Image Dataset

The images were acquired from a drone flying above a plot of corn field by the Department of Agronomy at the Kansas State University under the supervision of Dr. Ignacio Ciampitti. The images taken were in the form of RGBA. The resolution of the images was set at 1848 x 1063 taken from a height of 15 feet above the ground. Different plant populations were captured. In order to apply the various segmentation algorithms on the images to detect crop plants and thereby detect crop rows, the images had to be first stitched into an ortho-mosaic and then smaller, non-overlapping plots were cut. Several image enhancing and cleaning techniques were applied like histogram normalization, cropping, and removal of alpha-channel. A total of 50 images were taken with 4 varying plant populations

3.2 Image Segmentation

Image segmentation is employed so that separation between soil, weeds, and crop plants can be achieved. Various techniques used for segmentation are described below.

3.2.1 Excess Green (ExG) Index algorithm

Excess Green or ExG Index algorithm is a standard algorithm to segment an image into 'green' and 'non-green' areas. This vegetation index can be used to discriminate if a pixel corresponds to vegetation or soil. This index can be computed for each pixel as follows:

$$E = 2G' - R' - B'$$

where

$$R' = \frac{R}{R+G+B} \text{ ,}$$

$$G' = \frac{G}{R+G+B}\text{,}$$

$$B' = \frac{B}{R+G+B}\text{,}$$

E is the ExG output matrix, and G, R and B are the matrices associated to the green, red, and blue spectral channels, respectively. Figure 2 shows the effect of applying the ExG algorithm on an image containing some vegetation.

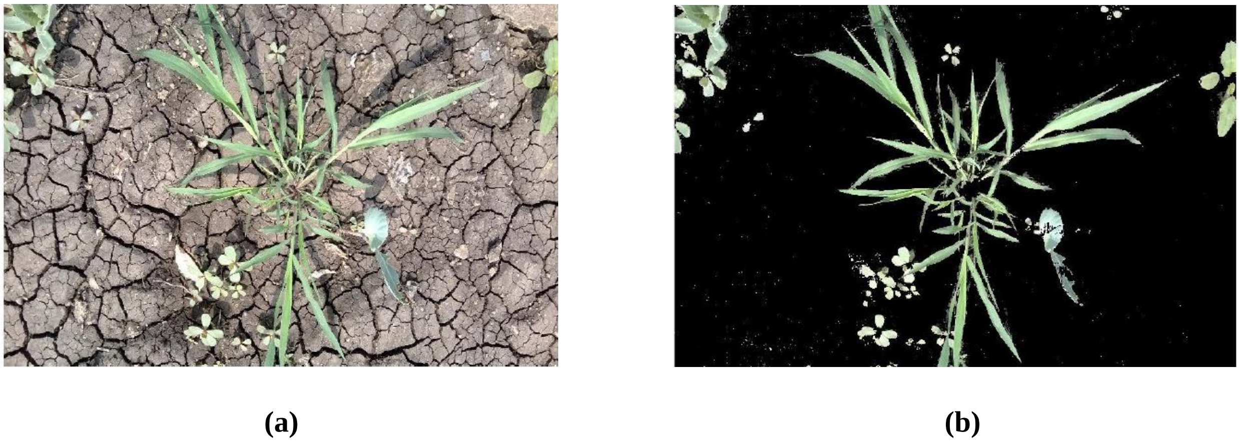


Figure 2. (b) shows the effect of applying the ExG algorithm on (a). Only the vegetation, i.e. green part, remains in (b).

3.2.2 K-Means Clustering

K-Means clustering algorithm [24] is an unsupervised clustering algorithm that classifies the input data points into multiple classes based on their inherent distance from each other. The algorithm assumes that the data features form a vector space and tries to find natural clustering in them. The points are clustered around centroids $\mu_i = 1,2, \ldots, k$which are obtained by minimizing the objective:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left(x_j - \mu_i\right)^2$$

where there are k clusters $S_i, i = 1,2, \ldots, k$ and μ_i is the centroid or mean point of all the points $x_j \in S_i$.

The algorithm takes an RGB image, and the number of clusters as input. Various steps in the algorithm are as follows:

1. Compute the intensity distribution (also called the histogram) of the intensities.
2. Initialize the centroids with k random intensities.
3. Repeat the following steps until the cluster labels of the image does not change anymore.
    a. Cluster the points based on distance of their intensities from the centroid intensities.

    $$c^i := argmin_j \left|x^i - \mu^j\right|^2$$

    b. Compute the new centroid for each of the clusters.

$$\mu_i := \frac{\sum_{i=1}^{m} 1\{c_i = j\} x^i}{\sum_{i=1}^{m} 1\,\{c_i = j\}}$$

where k is a parameter of the algorithm (the number of clusters to be found), i iterates over the all the intensities, j iterates over all the centroids and $\mu_i$are the centroid intensities

3.2.3 Gaussian Mixture Models

Since the images have components which are overlapping and contain a considerable amount of noise in the form of weeds, straw, and hay, a mixture of Gaussians (GMM) was employed to segment the image into different classes. The probability given in a mixture of K Gaussians is:

$$p(x) = \sum_{j=1}^{K} w_j \,.\, N(x|\mathrm{\mu}_j, \Sigma_j)$$

where w_j is the prior probability (weight) of the j^{th} Gaussian. To estimate the parameters of the GMM, the expectation-maximization algorithm is used.

Expectation Maximization (EM) is one of the most common algorithms used for density estimation of data points in an unsupervised setting. The algorithm relies on finding the maximum likelihood estimates of parameters when the data model depends on certain latent variables. In EM, alternating steps of Expectation (E) and Maximization (M) are performed iteratively till the results converge. The E step computes an expectation of the likelihood by including the latent variables as if they were observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the last E step. The parameters found on the M step are then used to begin another E step, and the process is repeated until convergence.

Mathematically for a given training dataset $\{x_1, x_2, \dots, x_m\}$ and model $p(x, z)$ where $z$is the latent variable, we have:

$$l(\theta) = \sum_{i=1}^{m} logp(x; \theta)$$

$$\sum_{i=1}^{m} log \sum_{z} p(x, z; \theta)$$

As can be seen from the above equation, the log likelihood is described in terms of x, z and θ. But since z, the latent variable is not known, we use approximations in its place. These approximations take the form of E and M steps mentioned above and formulated mathematically below.

E Step, for each i:

$$Q_i(z^i) \coloneqq p(z^i | x^i; \theta)$$

M Step, for all z:

$$\theta := argmax_{\theta} \sum_{i} \sum_{z^i} Q_i\,(z^i) log \frac{p(x^i, z^i; \theta)}{Q_i(z^i)}$$

where Q_i is the posterior distribution of z^i's given the x^i's.

Conceptually, The EM algorithm can be considered as a variant of the K-Means algorithm where the membership of any given point to the clusters is not complete and can be fractional.

3.2.4 Support Vector Machines

Support Vector Machine (SVM) is a supervised classification method derived from statistical learning theory that often yields good classification results from complex and noisy data. It separates the classes with a decision surface that maximizes the margin between the classes. The surface is often called the optimal hyperplane, and the data points closest to the hyperplane are called support vectors. The support vectors are the critical elements of the training set.

SVM can be adapted to become a nonlinear classifier through the use of nonlinear kernels. While SVM is a binary classifier in its simplest form, it can function as a multiclass classifier by combining several binary SVM classifiers (creating a binary classifier for each possible pair of classes), for example, the pairwise classification strategy for multiclass classification.

3.2.5 Deep Learning (Fully Convolutional Networks)

Classification using a machine learning algorithm has 2 phases:

1. Training phase: In this phase, a machine learning algorithm is trained using a dataset comprised of the images and their corresponding labels.
2. Prediction phase: In this phase, the trained model is utilized to predict labels of unseen images.

The training phase for an image classification problem has 2 main steps:

1. Feature Extraction: In this phase, domain knowledge is utilized to extract new features that will be used by the machine learning algorithm. HoG [23] and SIFT [24] are examples of features used in image classification.
2. Model Training: In this phase, a clean dataset composed of the images' features and the corresponding labels is utilized to train the machine learning model.

In the prediction phase, the same feature extraction process is applied to the new images and the features are passed to the trained machine learning algorithm to predict the label.

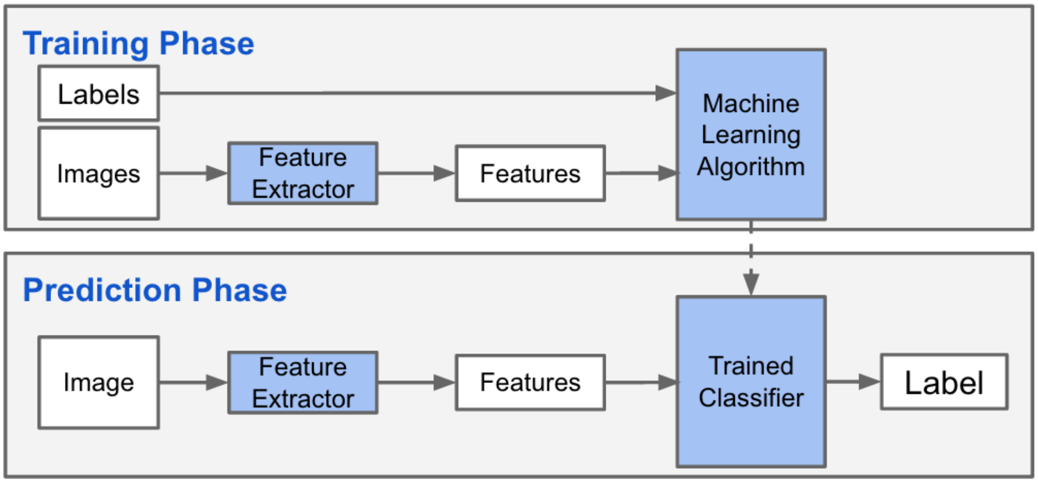


Figure 3. Phases in traditional Machine Learning.

The main difference between traditional machine learning and deep learning algorithms is in the feature engineering. In traditional machine learning algorithms, the features need to be hand-crafted. By contrast, in deep learning algorithms feature engineering is done automatically by the algorithm. Feature engineering is difficult, time-consuming and requires domain expertise. The promise of deep learning is more accurate machine learning algorithms compared to traditional machine learning with less or no feature engineering.

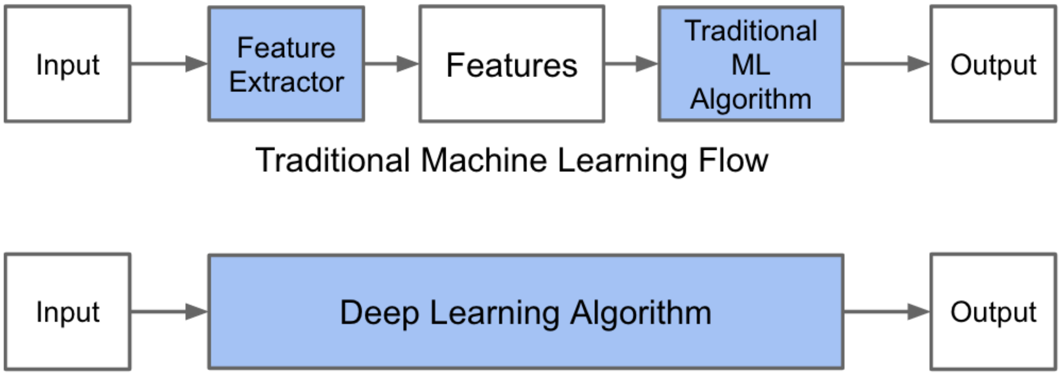


Figure 4. Machine Learning workflow versus Deep Learning. Feature Engineering is taken care of automatically by the deep learning algorithm.

3.2.5.1 Artificial Neural Networks

An artificial neuron has a finite number of inputs with weights associated to them, and an activation function (also called transfer function). The output of the neuron is the result of the activation function applied to the weighted sum of inputs. Artificial neurons are connected with each other to form artificial neural networks.

3.2.5.2 Feedforward Neural Networks

Feedforward Neural Networks are the simplest form of Artificial Neural Networks.

These networks have 3 types of layers: Input layer, hidden layer and output layer. In these networks, data moves from the input layer through the hidden nodes (if any) and to the output nodes.

Below is an example of a fully-connected feedforward neural network with 2 hidden layers. "Fully-connected" means that each node is connected to all the nodes in the next layer.

Note that, the number of hidden layers and their size are the only free parameters. The larger and deeper the hidden layers, the more complex patterns can be modeled in theory.

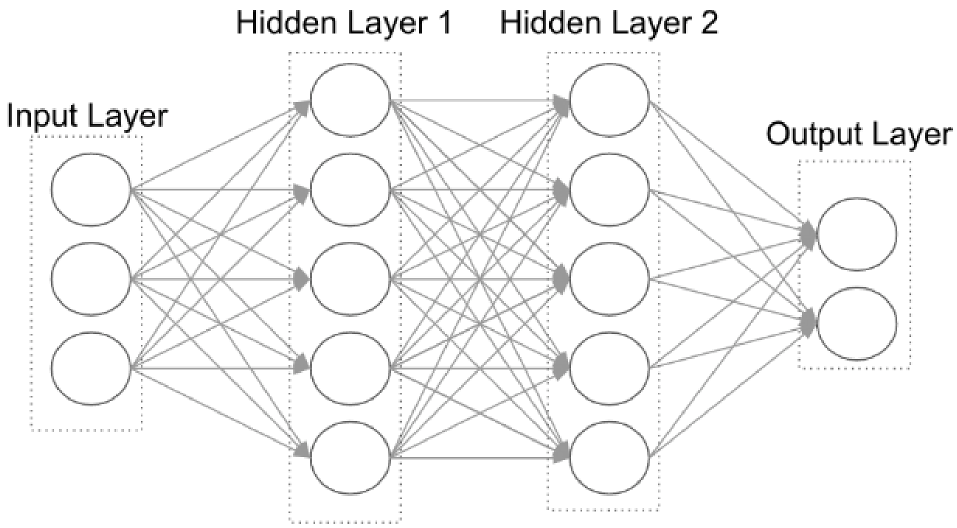


Figure 5. Example of a "Fully Connected" Feed-Forward Neural Network.

3.2.5.3 Activation Functions

Activation functions transform the weighted sum of inputs that goes into the artificial neurons. These functions should be non-linear to encode complex patterns of the data. The most popular activation functions are Sigmoid, Tanh and ReLU. ReLU is the most popular activation function in deep neural networks.

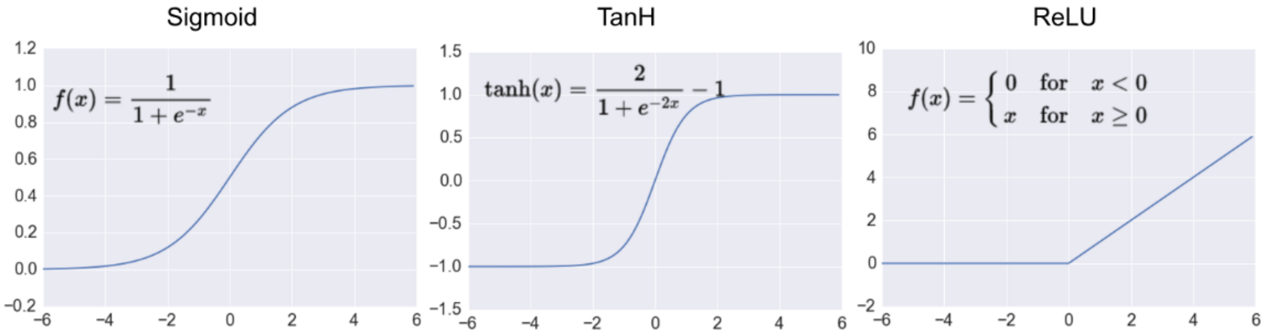


Figure 6. Common activation functions used to transform inputs going into neural networks.

3.2.5.4 Convolutional Neural Networks (CNNs or ConvNets)

Convolutional neural networks are a special type of feed-forward networks. These models are designed to emulate the behaviour of a visual cortex. CNNs perform very well on visual recognition tasks. CNNs have special layers called convolutional layers and pooling layers that allow the network to encode certain images properties.

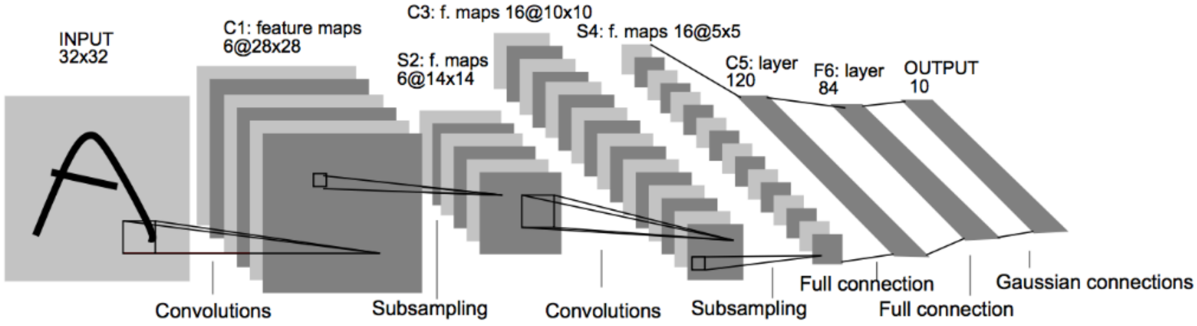


Figure 7. Convolutional Neural Network called LeNet created by Yann LeCunn

1. **Convolution Layer:** This layer consists of a set of learnable filters that we slide over the image spatially, computing dot products between the entries of the filter and the input image. The filters should extend to the full depth of the input image. For example, if we

want to apply a filter of size 5x5 to a colored image of size 32x32, then the filter should have depth 3 (5x5x3) to cover all 3 color channels (Red, Green, Blue) of the image. These filters will activate when they see same specific structure in the images.

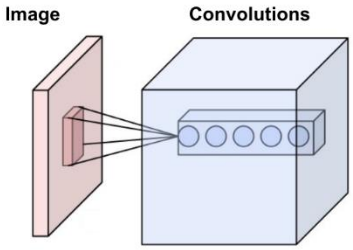
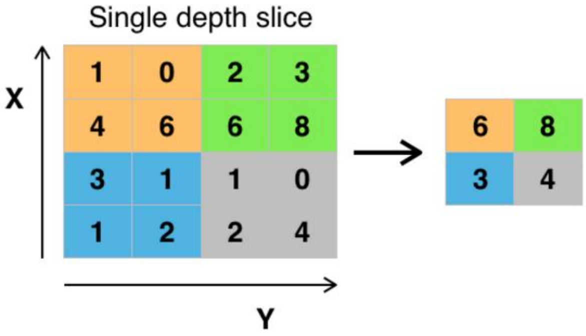


Figure 8. Neurons of a convolutional layer, connected to their receptive field.

2. **Pooling Layer:** Pooling is a form of non-linear down-sampling. The goal of the pooling layer is to progressively reduce the spatial size of the representation to reduce the number of parameters and computation in the network, and hence to also control overfitting. There are several functions to implement pooling among which max pooling is the most common one. Pooling is often applied with filters of size 2x2 applied with a stride of 2 at every depth slice. A pooling layer of size 2x2 with stride of 2 shrinks the input image to a 1/4 of its original size.



1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

6	8
3	4

Figure 9. A 2x2 Pooling layer with a stride of 2 shrinks the input image to 1/4th of its size.

The simplest architecture of a convolutional neural networks starts with an input layer (images) followed by a sequence of convolutional layers and pooling layers, and ends with fully-connected layers. The convolutional layers are usually followed by one layer of ReLU activation functions.

The convolutional, pooling and ReLU layers act as learnable features extractors, while the fully connected layers acts as a machine learning classifier. Furthermore, the early layers of the network encode generic patterns of the images, while later layers encode the details patterns of the images. Note that only the convolutional layers and fully-connected layers have weights. These weights are learned in the training phase.

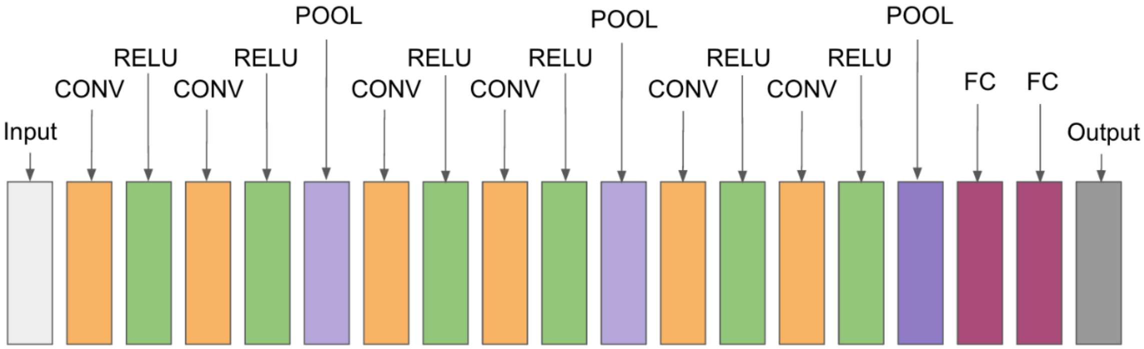


Figure 10. Architecture of a typical CNN.

3.2.5.5 Fully Convolutional Networks (FCN)

A Fully Convolutional neural network (FCN) is a normal CNN, where the last fully connected layer is substituted by another convolution layer with a large "receptive field". The idea is to capture the global context of the scene. The main difference between a CNN and an FCN is that the fully convolutional net is learning filters everywhere. Even the decision-making layers at the end of the network are filters.

An FCN tries to learn representations and make decisions based on local spatial input. Appending a fully connected layer enables the network to learn something using global information where the spatial arrangement of the input falls away and need not apply. The idea is that if the last convolutional layer is chosen to be big enough we will have this localization effect scaled up to our input image size.

In this work, a Fully Convolutional Network based on AlexNet and pretrained on the PASCAL VOC dataset is used as a starting point. PASCAL VOC has 20 prediction classes. In this work, that is also modified to only 2 classes : plant and non-plant (soil, straw, etc.).

3.3 Crop Row Detection

Crop Row detection is a crucial step after segmentation. This work addresses the detection of straight crop rows that are parallel to each other and are free from perspective projection. The main challenge faced in crop row detection is the presence of high weed pressure along with straw remaining from the previous crop cycle. In order to detect crop rows accurately, further processing of the image is required. The approaches employed in the cleanup are discussed below.

3.3.1 Morphological Operations

Morphological operations are performed to reduce the noise and clean up the image to provide a clean input to the next step. Some morphological operations are described below: [21]

1. **Dilation**: Dilation is one of the two basic operators in mathematical morphology, the other being erosion. It is typically applied to binary images, but there are versions that work on grayscale images. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus, areas of foreground pixels grow in size while holes within those regions become smaller. The dilation operator takes two pieces of data as inputs. The first is the image which is to be dilated. The second is a set of coordinate points known as a

structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the dilation on the input image. The mathematical definition of dilation for binary images is as follows: suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element. Let Kx denote the translation of K so that its origin is at x. Then the dilation of X by K is simply the set of all points x such that the intersection of Kx with X is non-empty. The mathematical definition of grayscale dilation is identical except for the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D. Dilation is the dual of erosion i.e. dilating foreground pixels is equivalent to eroding the background pixels.

2. **Erosion**: Erosion is one of the two basic operators in the area of mathematical morphology, the other being dilation. It is typically applied to binary images, but there are versions that work on grayscale images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus, areas of foreground pixels shrink in size, and holes within those areas become larger. The erosion operator takes two pieces of data as inputs. The first is the image which is to be eroded. The second is a set of coordinate points known as a structuring element (also known as a kernel). It is this structuring element that determines the precise effect of the erosion on the input image. The mathematical definition of erosion for binary images is as follows: suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element. Let Kx denote the translation of K so that its origin is at x. Then the erosion of X by K is simply the set of all points x such that Kx is a subset of X. The mathematical definition for grayscale erosion is identical except in the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D. Erosion is the dual of dilation, i.e. eroding foreground pixels is equivalent to dilating the background pixels.

3. **Opening**: Opening and closing are two important operators from mathematical morphology. They are both derived from the fundamental operations of erosion and

dilation. Like those operators, they are normally applied to binary images, although there are also gray level versions. The basic effect of an opening is somewhat like erosion in that it tends to remove some of the foreground (bright) pixels from the edges of regions of foreground pixels. However, it is less destructive than erosion in general. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve foreground regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of foreground pixels. An opening is defined as an erosion followed by a dilation using the same structuring element for both operations. The opening operator therefore requires two inputs: an image to be opened, and a structuring element. Gray level opening consists simply of a gray level erosion followed by a gray level dilation. Opening is the dual of closing, i.e. opening the foreground pixels with a particular structuring element is equivalent to closing the background pixels with the same element. While erosion can be used to eliminate small clumps of undesirable foreground pixels, e.g. `salt noise', quite effectively, it has the big disadvantage that it will affect all regions of foreground pixels indiscriminately. Opening gets around this by performing both an erosion and a dilation on the image.

4. **Closing**: Closing is an important operator from the field of mathematical morphology. Like its dual operator opening, it can be derived from the fundamental operations of erosion and dilation. Like those operators, it is normally applied to binary images, although there are gray level versions. Closing is similar in some ways to dilation in that it tends to enlarge the boundaries of foreground (bright) regions in an image (and shrink background color holes in such regions), but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve background regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels. Closing is opening performed in reverse. It is defined as a dilation followed by an erosion using the same structuring element for both operations. The closing operator therefore requires two inputs: an image to be closed and a structuring element. Gray level closing consists

straightforwardly of a gray level dilation followed by a gray level erosion. Closing is the dual of opening, i.e. closing the foreground pixels with a particular structuring element, is equivalent to closing the background with the same element. One of the uses of dilation is to fill in small background color holes in images, e.g. `pepper noise'. One of the problems with doing this, however, is that the dilation will also distort all regions of pixels indiscriminately. By performing an erosion on the image after the dilation, i.e. a closing, we reduce some of this effect.

3.3.2 Row Orientation

While some images of the plots of the corn field in the image dataset had the crop rows running parallel to the x-axis, others had rows that were at an angle. In order to accumulate all the ‘green-ness’ of the rows to detect the number of rows, the crop rows had to be more or less horizontal or in other words parallel to the x-axis. To achieve this and correct the orientation of these images, the Hough transform technique was used to detect the longest contiguous crop row. Then the whole image was rotated by the angle of this line so as to make the crop rows horizontal. The outputs of this step are the original rotated image and rotated binary image from the morphological step above.

3.3.2.1 Hough Transform

[22] The Standard Hough Transform (SHT) uses the parametric representation of a line:

$$\rho = x * cos(\theta) + y * sin(\theta)$$

The variable ρ is the distance from the origin to the line along a vector perpendicular to the line. θ is the angle of the perpendicular projection from the origin to the line measured in degrees clockwise from the positive x-axis. The range of theta is $-90° \leq \theta < 90°$. The angle of the line itself is $\theta + 90°$, also measured clockwise with respect to the positive x-axis.

The SHT is a parameter space matrix whose rows and columns correspond to ρ and θ values respectively. The elements in the SHT represent accumulator cells. Initially, the value in each cell is zero. Then, for every non-background point in the image, ρ is calculated for every θ. ρ is rounded off to the nearest allowed row in SHT. That accumulator cell is incremented. At the end of this procedure, a value of Q in $SHT(\rho\ ,\theta)$ means that Q points in the xy-plane lie on the line specified by θ and ρ. Peak values in the SHT represent potential lines in the input image.

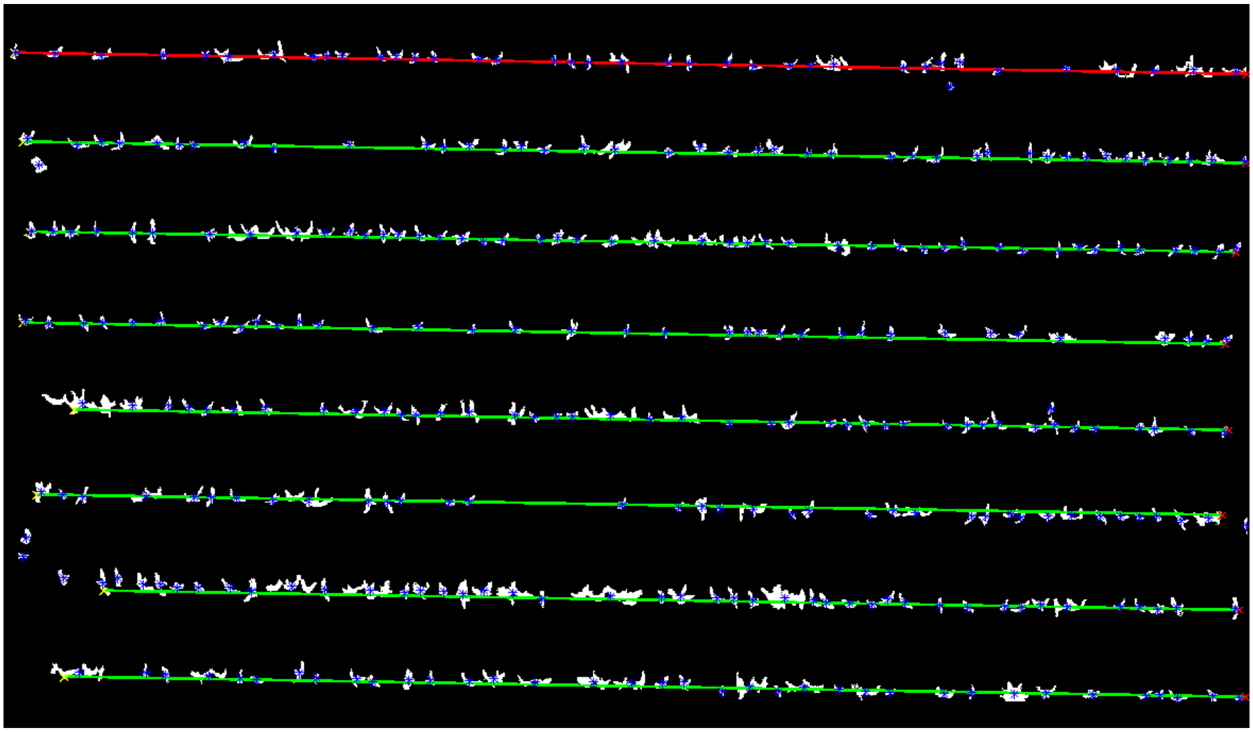


Figure 11. The red line indicates the longest contiguous line segment detected in the image. The angle of this line is taken to be the angle by which the image is rotated to make the crop rows parallel to the x-axis.

3.3.3 Green Pixel Accumulation and Row Template Creation

To detect the number of straight, horizontal crop rows in the images, a template creation technique was used. The images had already been preprocessed by first applying one of the segmentation approaches described above so that only the green components of the images, i.e.

the plants in the crop rows, remained and then thresholding or binarizing the image. Accumulating the remaining (the white portions) components by proceeding in a column-by-column fashion, gave peaks where there were crop rows in the image. Then, the obtained peak curve was smoothed and peaks below a certain threshold were discarded. The threshold, as of now, is taken to be the median of all the peaks minus 2. The number of peaks remaining were taken to be the number of rows in the image. A rectangular template was superimposed on each of these detected peaks and the regions outside of these templates were discarded.

3.3.4 Centroid Detection and Plant-Row Membership

After the crop rows have been detected and superimposed on the original rotated image, centroids are detected for each connected component (candidate plant) in the rotated binary image from steps 3 and 4.

To count the number of centroids/plants in each row, the following steps are taken:

a) The detected centroids are first sorted in ascending order in terms of their y – coordinates.

b) Starting from the first row in the row list and the first centroids in the centroids list, while there are still centroids/plants to count:

c) If the y – coordinate of the centroid lies inside the current row, the count of the current row is increased by 1.

d) Else if the y – coordinate of the centroid does not lie inside the current row, the centroid is tested for the next row.

    i. If the above test fails, the centroid is discarded as being outside of any row and the next centroid in the centroid list is tested.

ii. If the test succeeds, this means that all the centroids which could have lied inside the current row have been found. So, the next row becomes the current row

e) The process is repeated from step 2 until all centroid candidates and rows are exhausted.

Chapter 4 - Experiment Design

The algorithms listed above were tested on the dataset containing 50 images with varying number of plants and noise (straw). The resulting segmentation from each of the algorithms was evaluated against hand-labeled ground truth. The Dice Index was used as a measure of similarity between the segmentation and the ground truth. Number of plants per row were also compared between the ground truth and the segmentation algorithms.

4.1 Test Bed

The experiments were done on a machine equipped with an Intel Core i5 – 6600k processor running at 3.5 Ghz and 16 GB of DDR4 RAM, and an nVidia GTX 1070 GPU with 8 GB of video memory. Initial experiments were carried out using the OpenCV computer vision framework and Python before switching to MATLAB 2015a edition. For fully convolutional networks, two deep learning frameworks were considered, namely, Caffe framework developed by the University of Berkeley, and TensorFlow developed by Google, Inc. Ultimately, the Caffe library was chosen over Tensorflow as the authors of the original FCN segmentation paper [25] utilized the Caffe library.

4.2 Evaluation Measures

The detection accuracy was evaluated by comparing the segmentation results with a ground truth segmentation, following the methodology of pattern comparison and precision analysis. Therefore, True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) are counted. TP are correctly detected pixels, i.e., those assigned to areas containing only a plant. TN are pixels correctly assigned to the rest of the image. Consequently, FP and FN are pixels which are not correctly assigned (FP = false plant detection, FN = false non-plant detection).

Precision is given by:

$$\mathrm{p} \ = \frac{TP}{(TP + FP)},$$

And, Recall is given by:

$$\mathrm{r} \ = \frac{TP}{(TP + FN)}$$

The Dice Similarity Index combines these parameters into the following equation:

$$DI \ = \frac{2(p.r)}{(p + r)}$$

The Dice Index characterizes the overall performance of the detection. Average run-time of the various segmentation algorithms is also compared.

Chapter 5 - Results

The results of running the various segmentation algorithms are presented in this section. After segementation, the crop row detection algorithm is applied to the images to produce straight crop rows parallel to the x-axis. The original image shown in Figure 12 has 8 crop rows and a total plant population of 240. Per row plant population is given in Table XX. Figure 13 shows the hand-annotated image taken as ground truth.

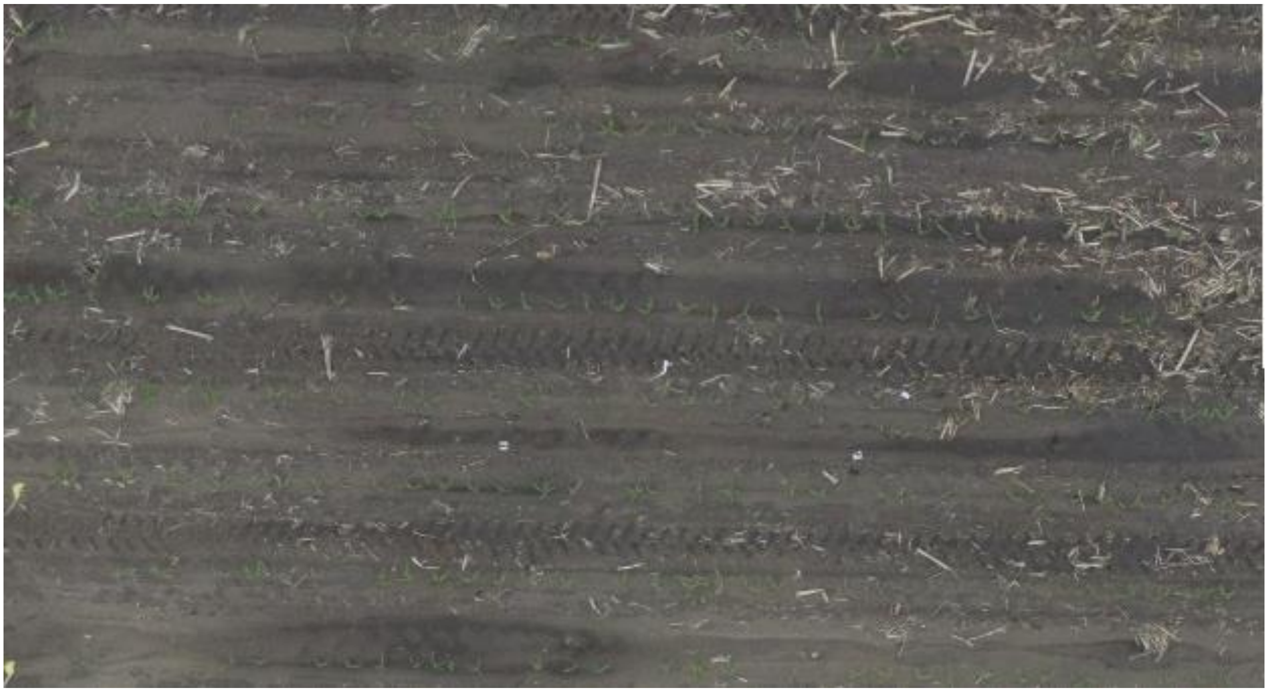


Figure 12. Original RGB image with 8 crop rows and a plant count of 240.

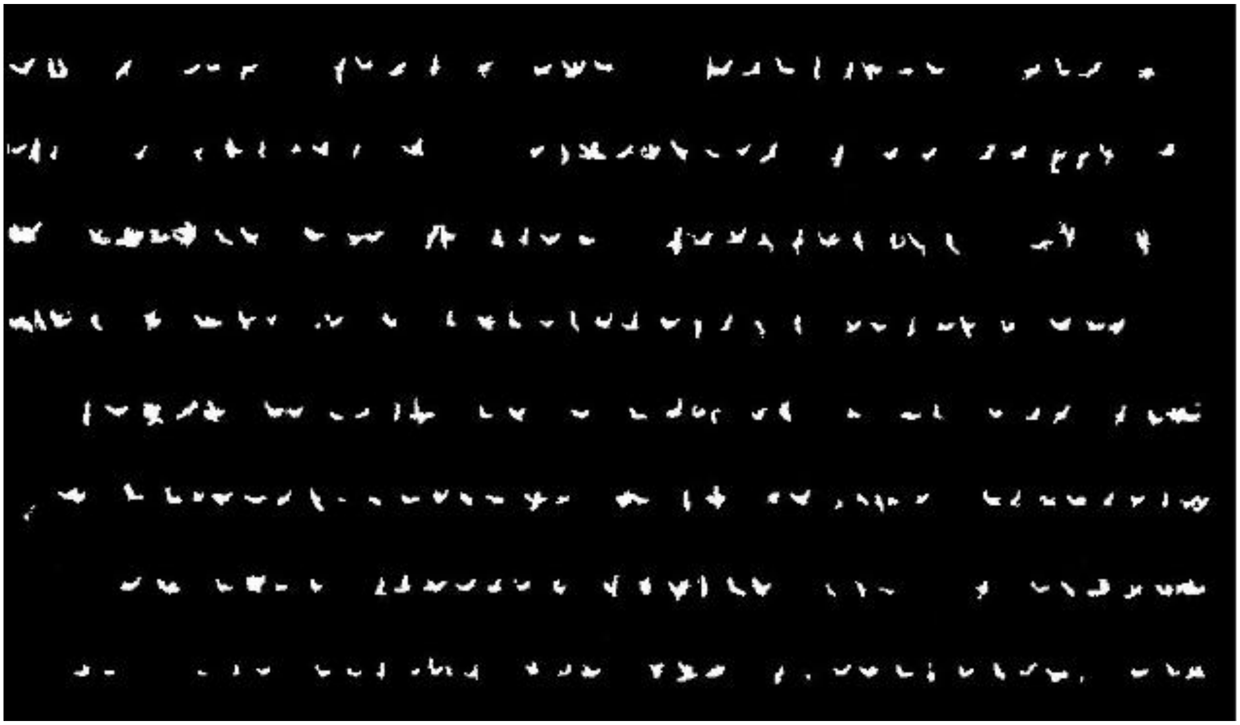


Figure 13. Ground truth for the image presented in Figure 12. The white pixels denote plants and the black pixels represent everything else (soil, straw, etc.).

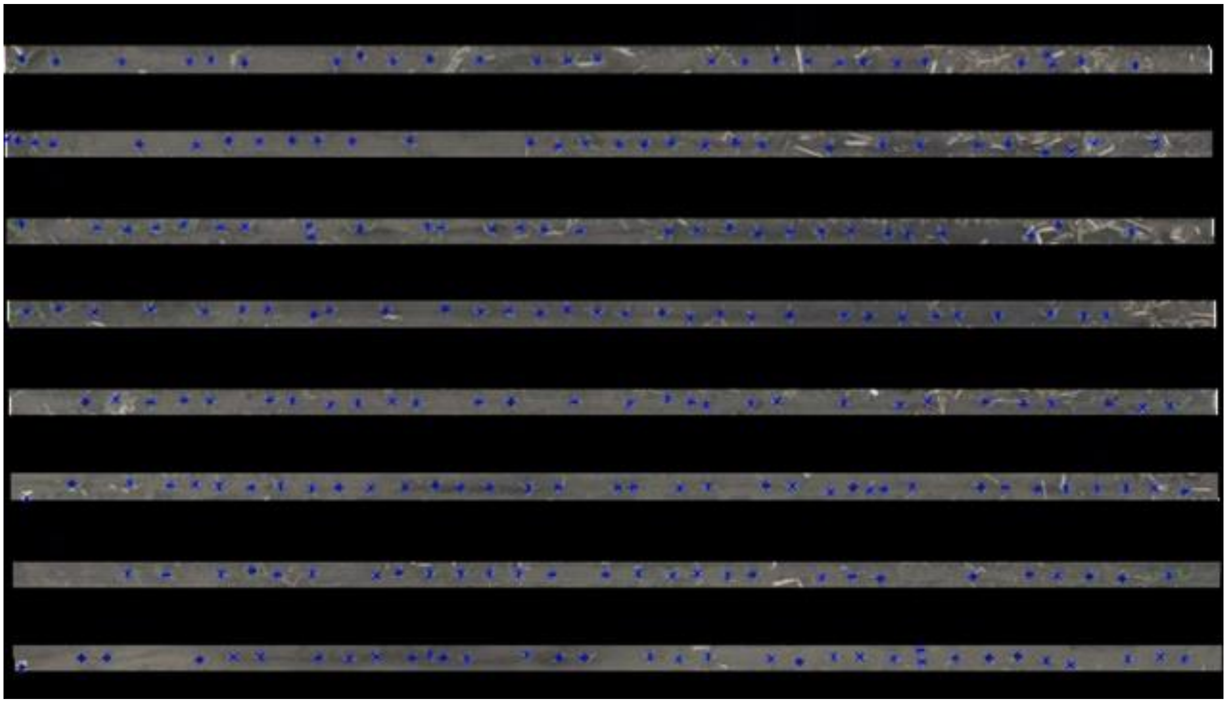


Figure 14. Ground truth image showing the eight crop rows along with the centroids (blue) of the plants in each row.

5.1 Excess Green (ExG) Index

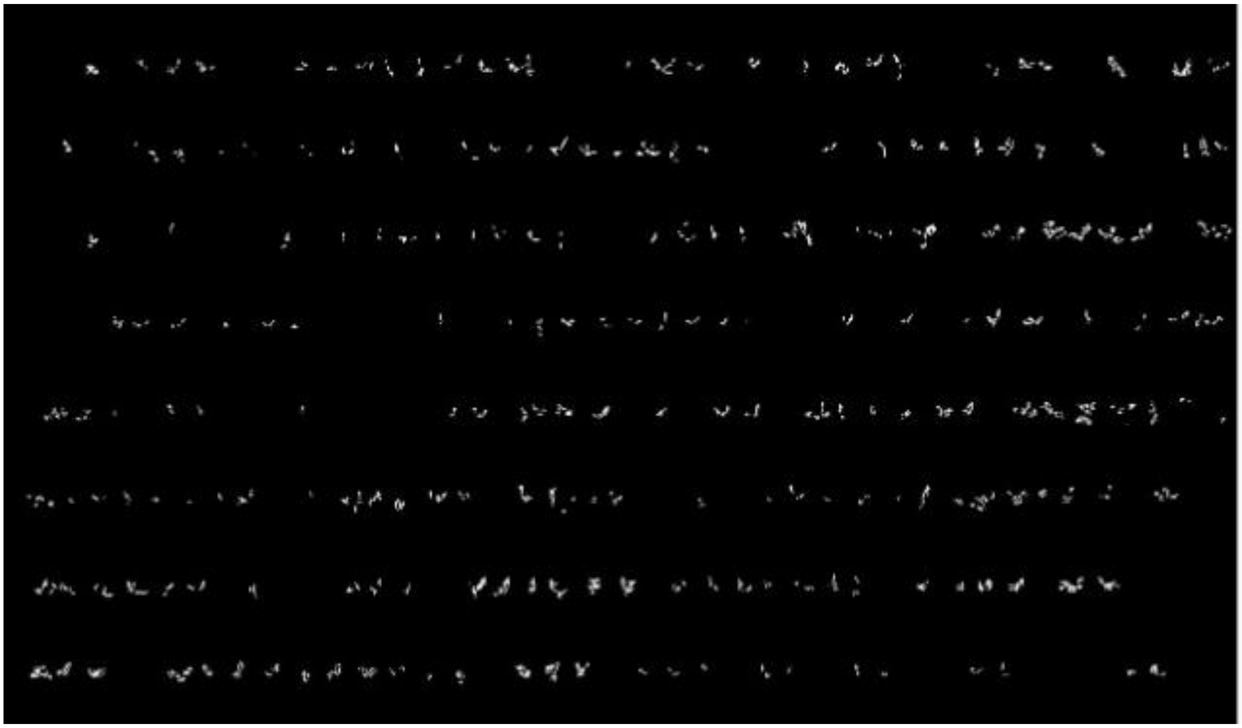


Figure 15. Segmentation results obtained through the ExG algorithm.

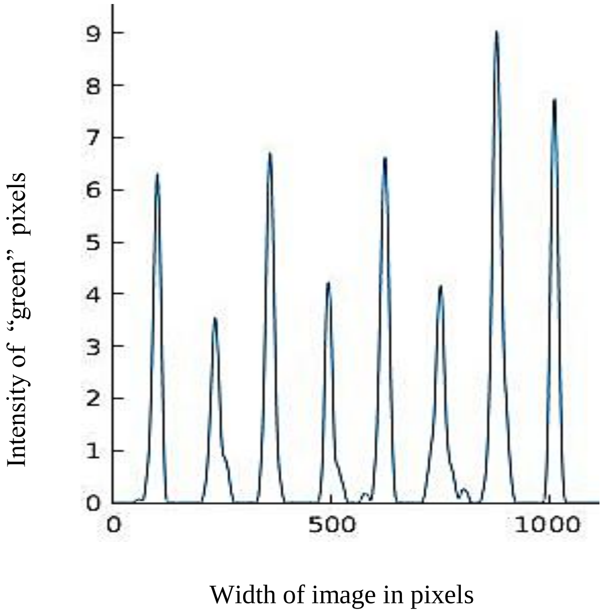


Figure 16. Green Pixel Accumulation (GPA) shows maxima where there is a concentration of plants and minima where there is a gap between crop rows. Since there are 8 local maxima above a set threshold of 2, number of rows in the image is taken to be 8.

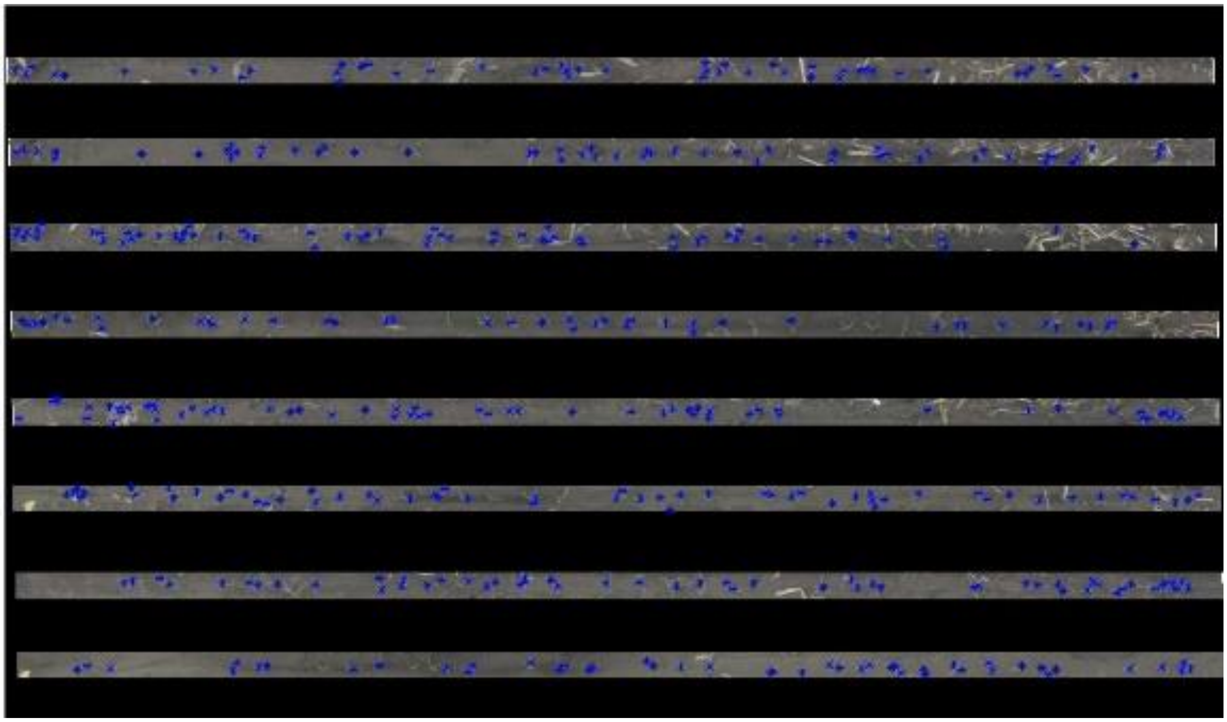


Figure 17. Result of running the crop row detection algorithm on the image segmented through the ExG algorithm in Figure 15.

5.2 K-Means Clustering

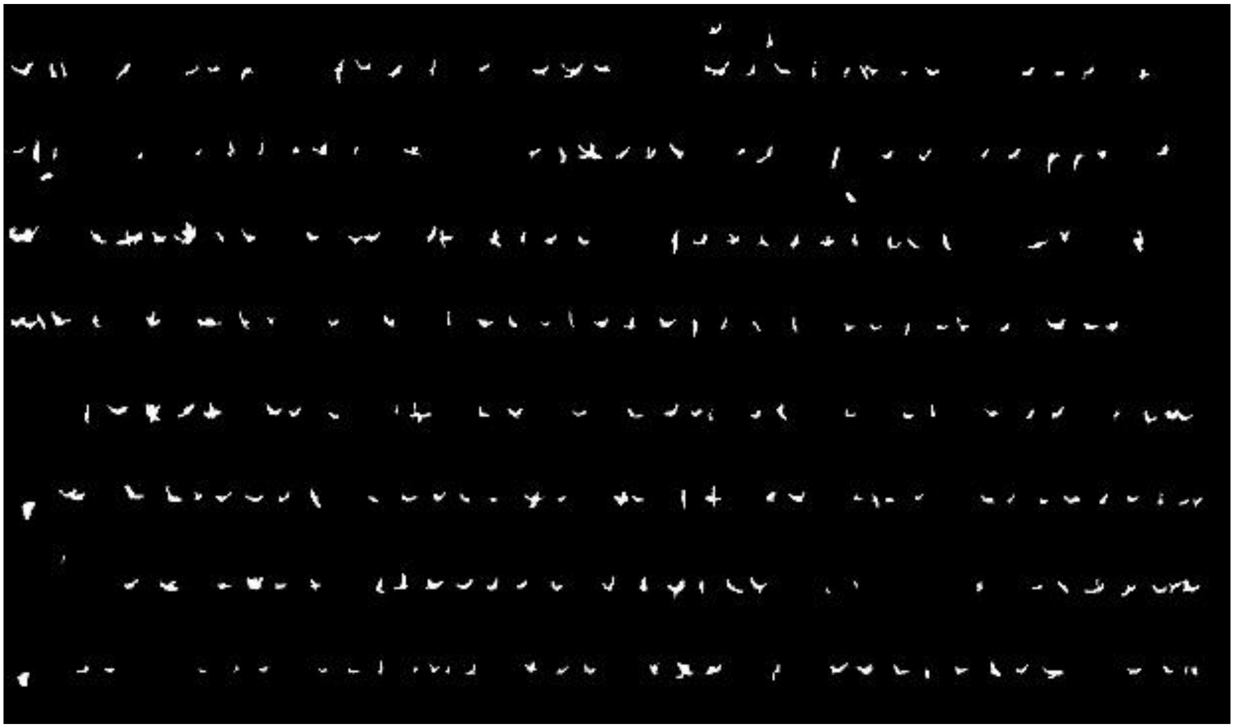


Figure 18. Segmentation results obtained through K-Means clustering with K = 3.

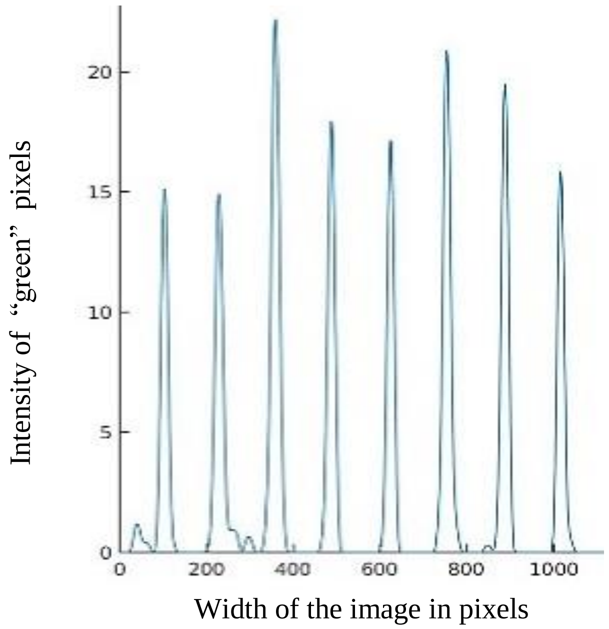


Figure 19. Result of running the GPA algorithm on the image in Figure 18. The eight local maxima above a set threshold of 2 indicate number of rows in the image.

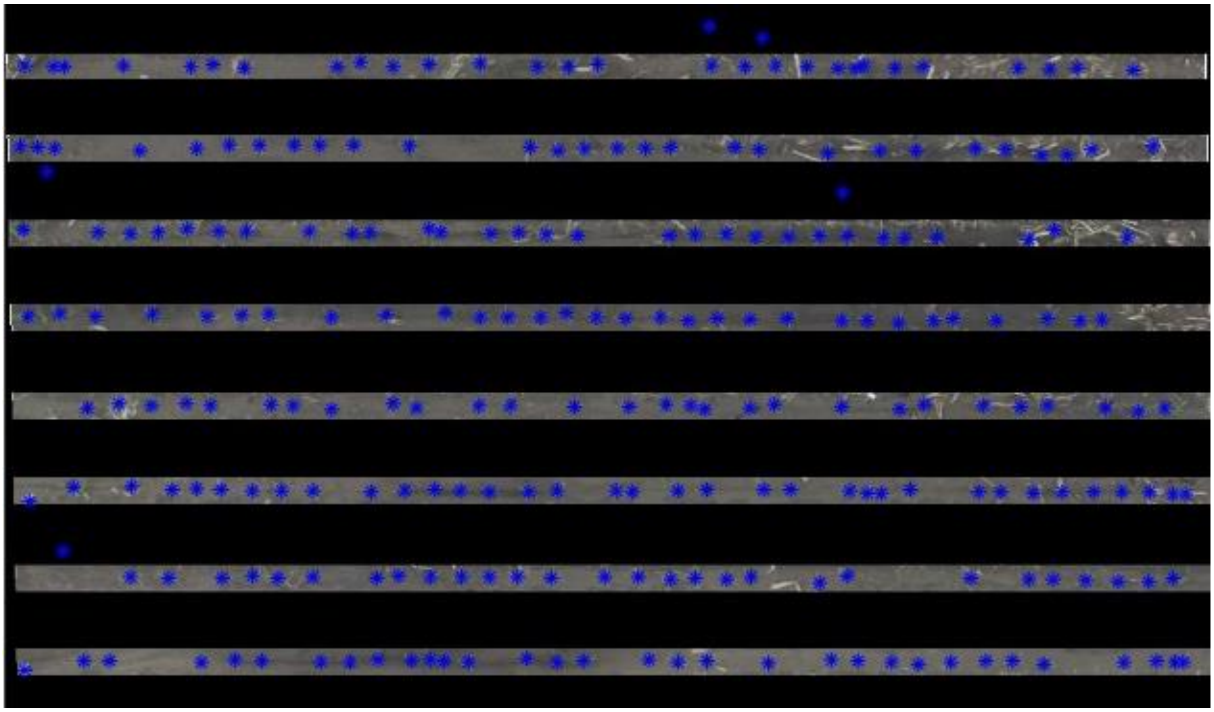


Figure 20. Result of running the crop row detection algorithm on the image segmented through the K-Means algorithm in Figure 18.

5.3 Gaussian Mixture Models (GMM)

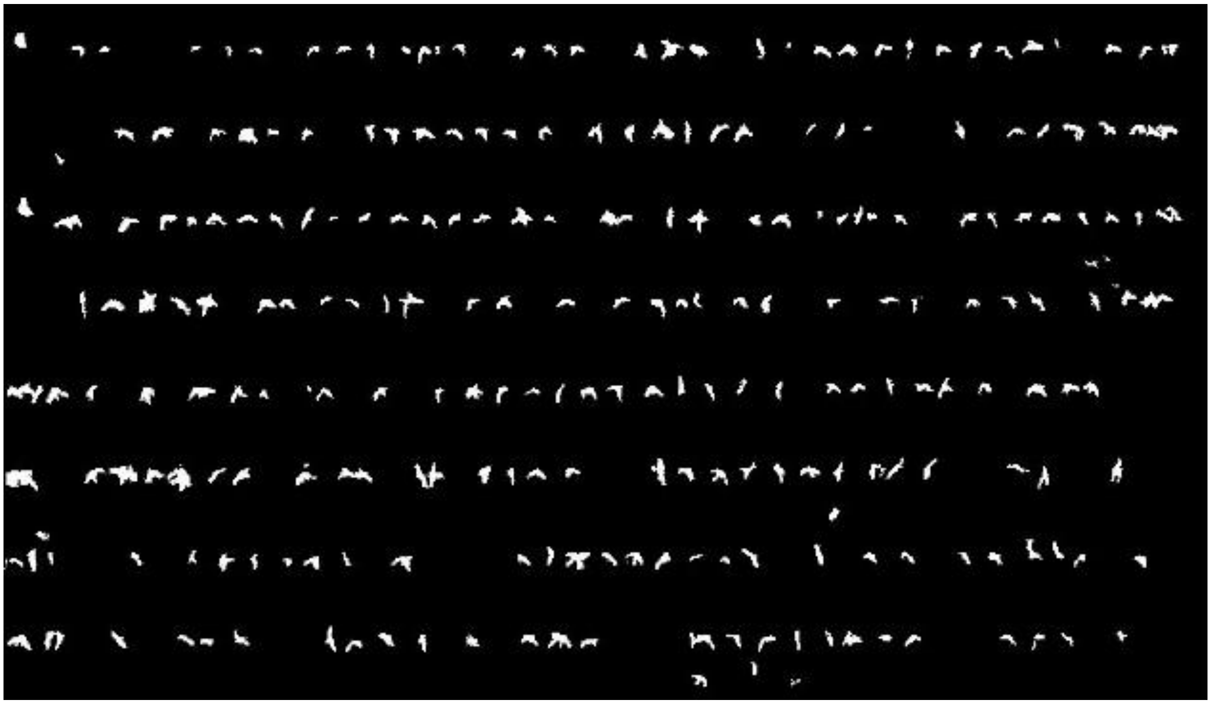


Figure 21. Segmentation results obtained through GMM where the number of classes is 3.

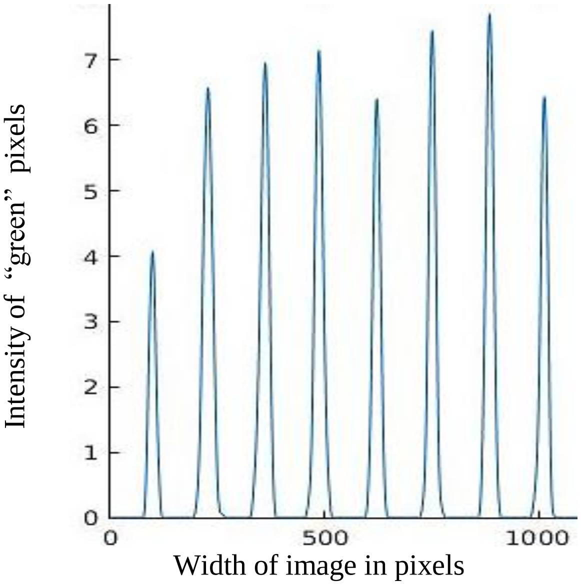


Figure 22. Result of running the GPA algorithm on the image in Figure 21. The eight local maxima above a set threshold of 2 indicate number of rows in the image.

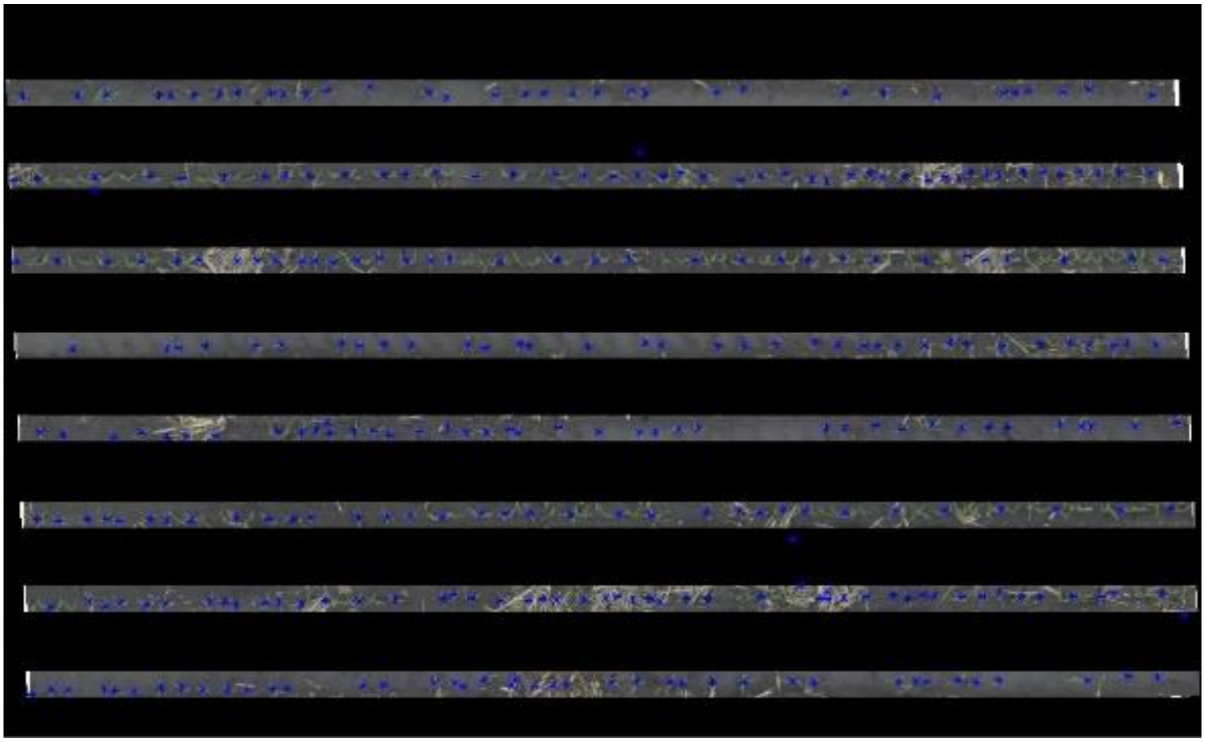


Figure 23. Result of running the crop row detection algorithm on the image segmented through the GMM algorithm in Figure 21.

5.4 Support Vector Machines

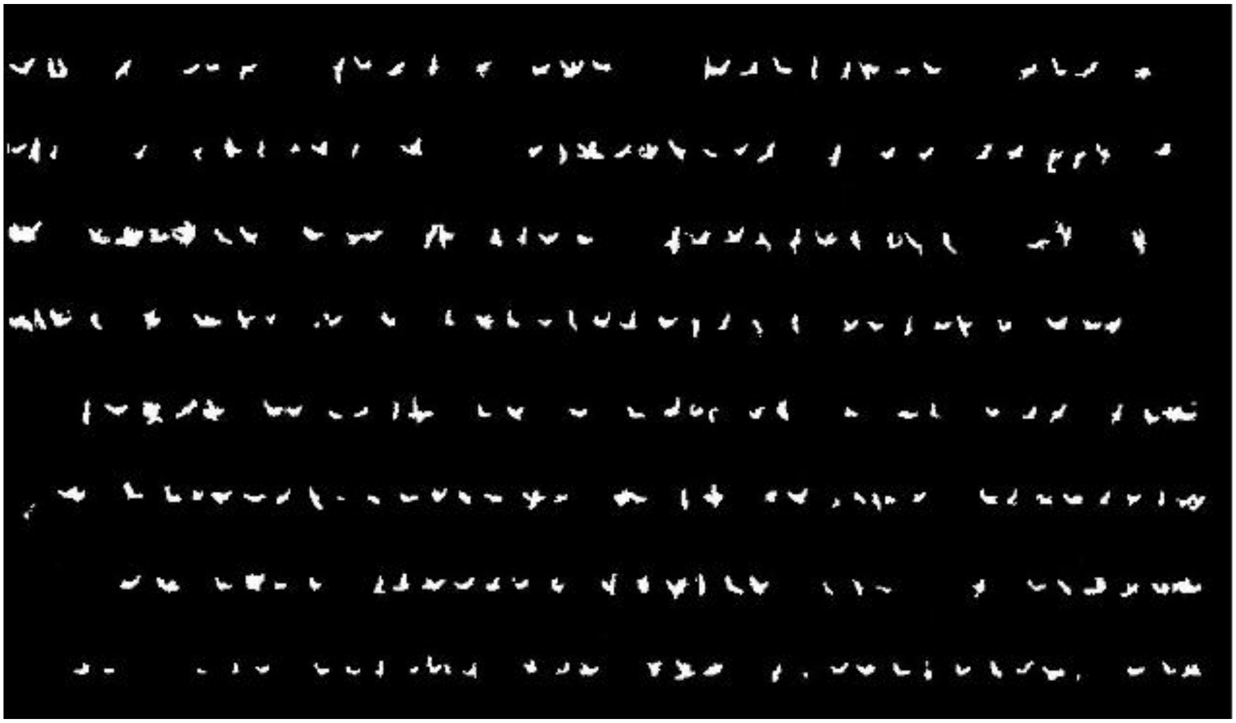


Figure 24. Segmentation results obtained through SVM where the number of classes is 3.

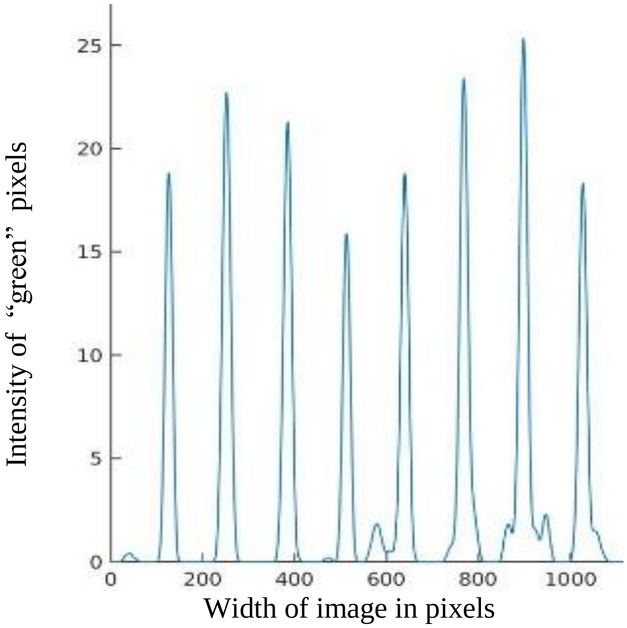


Figure 25. Result of running the GPA algorithm on the image in Figure 24. The eight local maxima above a set threshold of 2 indicate number of rows in the image.

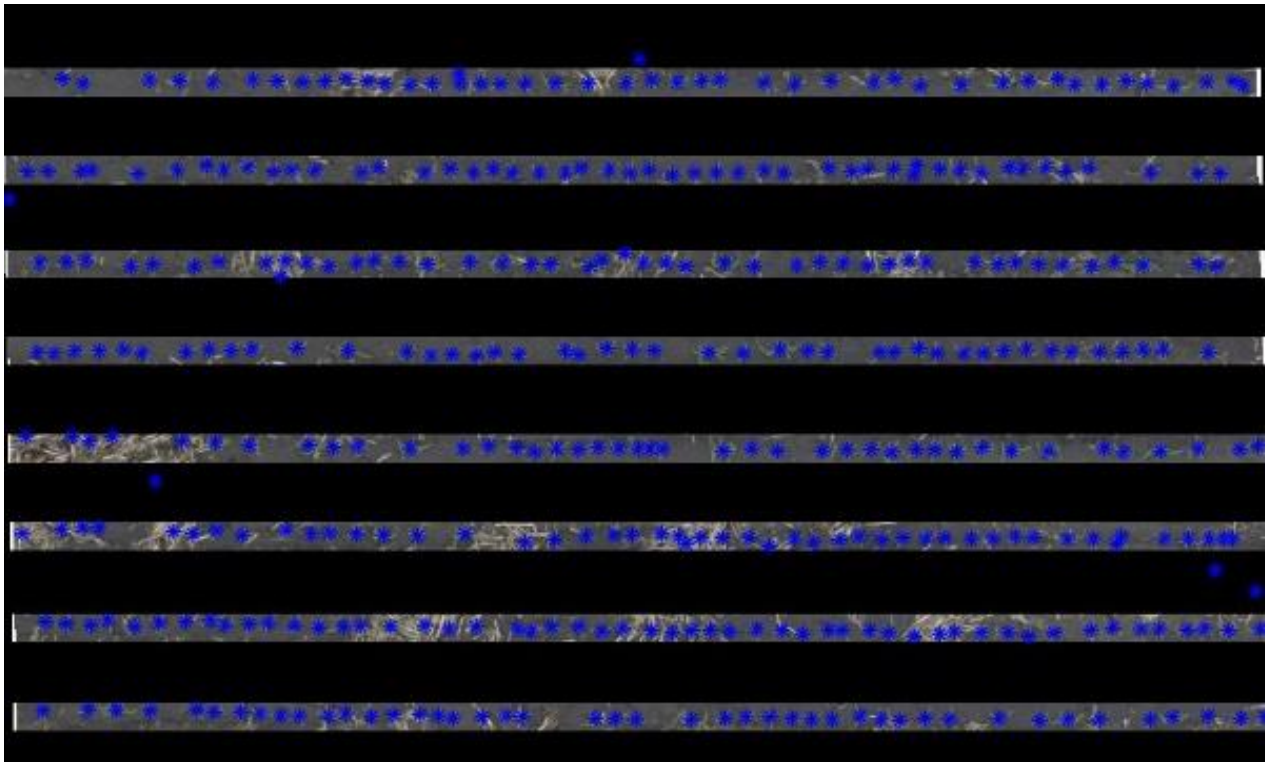


Figure 26. Result of running the crop row detection algorithm on the image segmented through the SVM algorithm in Figure 24.

5.5 Fully Convolutional Network (FCN)

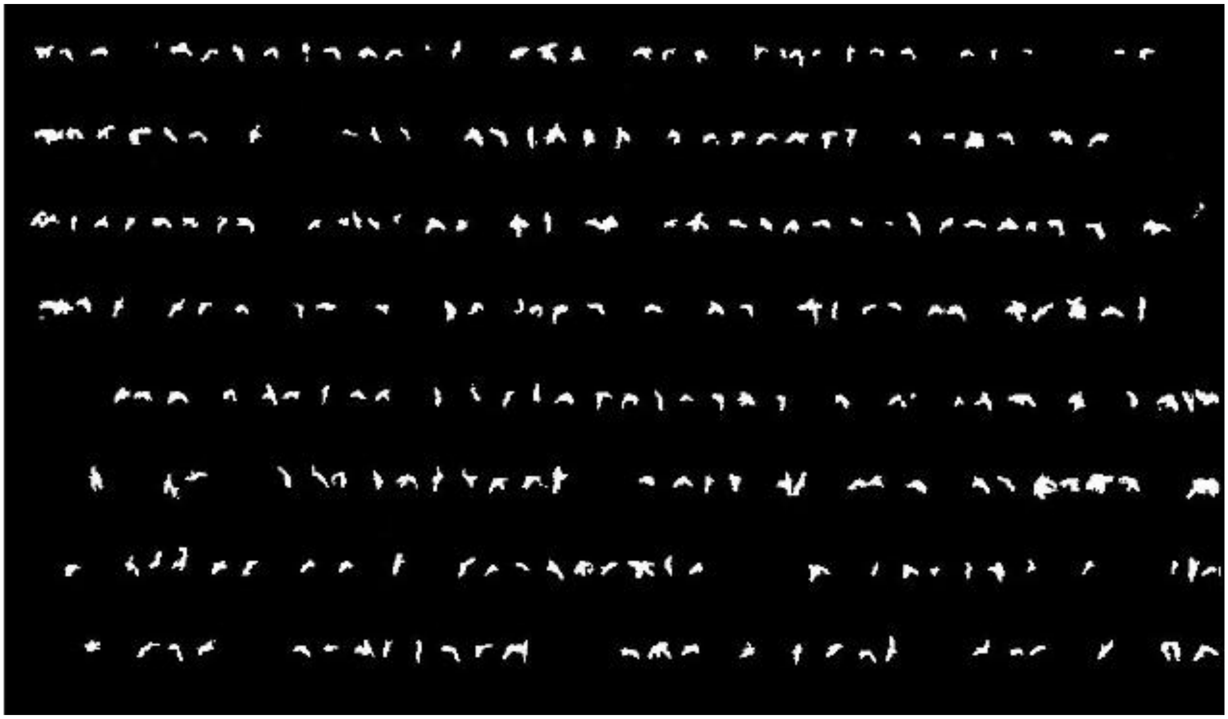


Figure 27. Segmentation results obtained through FCN.

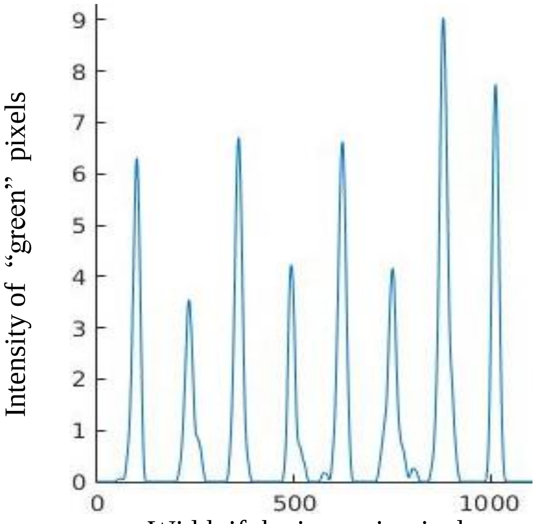


Figure 28. Result of running the GPA algorithm on the image in Figure 27. The eight local maxima above a set threshold of 2 indicate number of rows in the image.

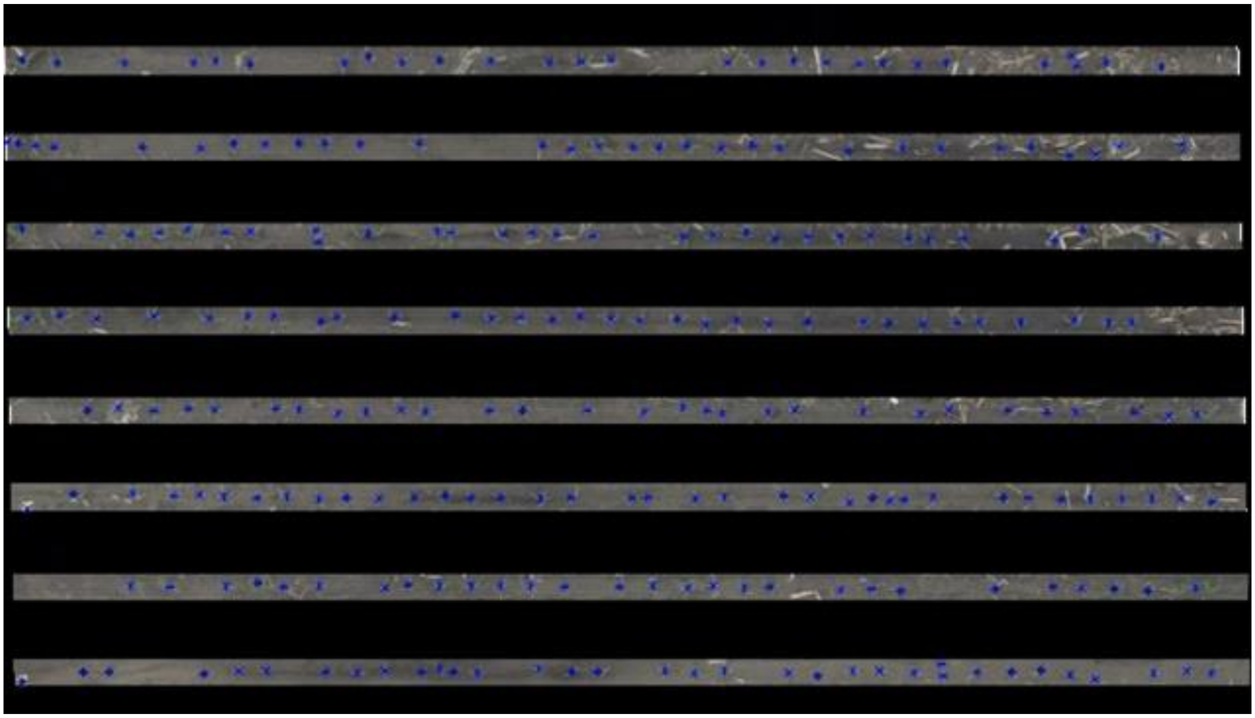


Figure 29. Result of running the crop row detection algorithm on the image segmented through the FCN algorithm in Figure 27.

5.6 Metrics

	ExG	K-Means	GMM	SVM	FCN
Dice index (%)	68.45	84.8	86.2	89.3	91.2

Table 1. Dice Similarity Index shows how similar the segmented image is with the ground truth.

	ExG	K-Means	GMM	SVM	FCN
Runtime (s)	0.54	6.2	48.3	259.5	1.73

Table 2. Runtime in seconds for the various segmentation algorithms.

Row Number	**Ground Truth**	**ExG**	**K-Means**	**GMM**	**SVM**	**FCN**
1	27	49	28	27	36	27
2	29	52	28	30	30	30
3	30	62	29	29	32	27
4	31	42	30	31	35	33
5	29	60	28	29	33	29
6	34	58	35	36	35	35
7	29	54	28	28	28	28
8	31	46	32	34	31	31
Total	240	423	238	244	260	240
Error %	**-**	**76.25**	**- 0.84**	**1.67**	**8.3**	**0**

Table 3. The number of plants by row obtained according to the segmentation algorithms tested. Error % indicates the overall error in detecting the number of plants.

From the above tables, we can observe that although the ExG algorithm is the fastest in segmenting plants from non-plants in the images, it has a high error rate and a low Dice Similarity Index which means that the similarity between the segmentation obtained through the ExG algorithm and the ground truth is poor. On the other hand, the FCN produces a high Dice index with low error rate while managing to run faster than every algorithm except the ExG algorithm.

Chapter 6 - Conclusion

In almost all the works done in the field of precision agriculture, the Excessive Green Index algorithm has been used extensively in separating vegetation pixels from other components like soil. However, it overestimates the number of vegetation pixels in the presence of noise like straw and weeds. In this work, other segmentation approaches have been explored which perform better under the environmental conditions listed above. After comparing various segmentation algorithms, it was found that Fully Convolutional Networks do indeed provide a good result based on the run-time and segmentation similarity determined by the Dice Index. Transfer learning is used repurpose an AlexNet based Fully Convolutional Network pretrained on ImageNet / Pascal VOC by modifying penultimate and last layers for 2-class segmentation and detection task on custom dataset used in this work. A new strategy that combines the use of Hough Transform for finding row orientation, thereby facilitating Green Pixel Accumulation that in turn is used for template creation is also discussed and implemented in this work.

Chapter 7 - Future Work

Transfer learning has been employed in the deep learning approach to segment the images. Designing a deep learning model specifically for the dataset used in this work instead of using a pretrained model could potentially provide improved results. In order to train a model from scratch, the dataset would also have to be expanded greatly. Even though Fully Convolutional Networks for semantic segmentation were introduced in 2016, other deep learning networks with some changes have emerged that could potentially provide improvements.

The strategy of Green Pixel Accumulation with Template Creation employed in this work can only be used to detect straight crop rows. This approach can be potentially extended to the task of detecting curved crop rows by evaluating the image row by row instead of column-by-column.

Chapter 8 - References

[1] P.V.C.Hough,"Method and means for recognizing complex patterns". United States Patent 3069654, 18 December 1962.

[2] J.A.Marchant,Tracking of row structure in three crops using image analysis, Comput. Electron. Agric. 15 (1996) 161–179.

[3] T. Bakker, H. Wouters, K. van Asselt, J. Bontsema, L. Tang, J. Müller, G. van Straten, A vision based row detection system for sugar beet, Comput. Electron. Agric. 60 (2008) 87–95.

[4] R. Ji, L. Qi, Crop-row detection algorithm based on Random Hough Transformation, Math. Comput. Model. 54 (2011) 1016–1020.

[5] J. A. Marchant, R. Brivot, Real-time tracking of plant rows using a Hough transform, Real-time Imaging 1 (1995) 363–371.

[6] F. Rovira-Más, Q. Zhang, J. F. Reid, J. D. Will, Hough-transform-based vision algorithm for crop row detection of an automated agricultural vehicle, Proc. Inst. Mech. Eng. Part D: J. Automob. Eng. 219 (2005) 999–1010.

[7] B. Åstrand, A. J. Baerveldt, A vision based row-following system for agricultural field machinery, Mechatronics 15 (2005) 251–269.

[8] S. Ericson B. Åstrand, A vision-guided mobile robot for precision agriculture, Proceedings of the 7th European conference on precision agriculture 2009.

[9] M. Montalvo, G. Pajares, J. Guerrero, J. Romeo, M. Guijarro, A. Ribeiro, J. Ruz, J. Cruz, Automatic detection of crop rows in maize fields with high weeds pressure, Expert Syst. Appl. 39 (2012) 11889–11897.

[10] D.M. Woebbecke, G. E. Meyer, K. vBargen, D. A. Mortensen, Color indices for weed identification under various soil, residue and lightening conditions, Trans. Am. Soc. Agric. Eng. 38 (1995) 259–269.

[11] J. Guerrero, M. Guijarro, M. Montalvo, J. Romeo, L. Emmi, A. Ribeiro, G. Pajares, Automatic expert system based on images for accuracy crop row detection in maize fields, Expert Syst. Appl. 40 (2013) 656–664.

[12] H. J. Olsen, Determination of row position in small-grain crops by analysis of video images, Comput. Electron. Agric. 12 (1995) 147–162.

[13] D. W. Deering, Field measurements of bidirectional reflectance, Theor. Appl. Opt. Remote Sens. (1989) 14–65.

[14] J. Romeo, G. Pajares, M. Montalvo, J. M. Guerrero, M. Guijarro, A. Ribeiro, Crop row detection in maize fields inspired on the human visual perception, The Scientific World Journal, vol. 2012, Article ID 484390, 10 pp, 2012, http://dx.doi.org/10.1100/2012/484390.

[15] H. T. Søgaard, H. J. Olsen, Determination of crop rows by image analysis without segmentation, Comput. Electron. Agric. 38 (2003) 141–158.

[16] N. D. Tillett, T. Hague, Computer-vision-based Hoe Guidance for Cereals - an Initial Trial, J. Agric. Eng. Res. 74 (1999) 225–236.

[17] T. Hague, N. D. Tillet, A band pass filter-based approach to crop row location and tracking, Mechatronics 11 (2001) 1–12.

[18] N. D. Tillett, T. Hague, S. J. Miles, Inter-row vision guidance for mechanical weed control in sugar beet, Comput. Electron. Agric. 33 (2002) 163–177.

[19] M. Kise, Q. Zhang, F. Rovira Más, A Stereo vision-based crop row detection method for tractor-automated guidance, Biosyst. Eng. 90 (2005) 357–367.

[20] Q. Wang, Q. Zhang, F. Rovira-Más, L. Tian, Stereovision-based lateral offset measurement for vehicle navigation in cultivated stubble fields, Biosyst. Eng. 109 (2011) 258–265.

[21] Morphological Operations. http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm

[22] Standard Hough Transform. https://www.mathworks.com/help/images/ref/hough.html

[23] Histogram of Oriented Gradients. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

[24] S. Tatiraju, A. Mehta, Image Segmentation using k-means clustering, EM and Normalized Cuts. Department of EECS University Of California – Irvine. https://www.ics.uci.edu/~dramanan/teaching/ics273a_winter08/projects/avim_report.pdf

[25] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015 (pp. 3431-3440).