2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

Rice Leaf Detection with Genetic Programming

Minh Luan Nguyen
School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
Email: nmluan@ctu.edu.vn

Vic Ciesielski
School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
Email: vic.ciesielski@rmit.edu.au

Andy Song
School of Computer Science and Information Technology
RMIT University
Melbourne, Australia
Email: andy.song@rmit.edu.au

***Abstract*—This paper describes an approach to the detection rice plants in images of rice fields by using genetic programming. The method involves the evolution of a genetic programming classifier of 20×20 pixel windows to distinguish rice and non-rice windows, applies the evolved classifier to each pixel position in a test image in a scanning window fashion and determines the class of a pixel by majority voting. The individual pixel values in the window comprise the terminal set. The four arithmetic operators, augmented by square root, comprise the function set. Fitness is a weighted sum of true positive and true negative rates. The classifier achieves an accuracy of 90% on positive and negative windows and is highly accurate in localizing rice leaves in test images for micro-spraying of nutritional supplements. The evolutionary approach clearly outperforms a thresholding approach based on colour which is unable to distinguish between rice an leaves.**

I. Introduction

In countries with economies based on agriculture the usage of pesticides, fungicides and nutritional supplements for crop products can be very costly and wasteful. To minimize such costs plant leaves need to be accurately detected so that the substances can be sprayed or applied only where needed. There has been previous work on cereal classification tasks using a variety of approaches including color and shape analysis techniques, back propagation neural network models, spectral information and reflection measurements [1], [2], [3], [4]. A goal of some of this work is to find ways to differentiate cereal plants and weeds. There has been some successes in applying these techniques to dry-fields such as corn, tomato, wheat, soya and sugar beet fields. However, there has been no investigation of rice plants, which is surprising considering that rice is the major cereal crop of the world.

Detecting rice plants is a difficult problem because rice plants and weeds have very similar colours and shapes. Rice and weed plants can grow together and are often interleaved. The images we have used also contain areas of soil. These various factors make this real world problem a difficult one. Since the colours of weed and rice leaves are so similar, detection based on colour is unlikely to be successful. However, discriminating by texture is a possibility since texture is independent of colour. In previous work by Ciesielski et al. [5], one step classifiers based on pixel values were successful in distinguishing a wide range of Brodatz textures suggesting that the approach might be suited to this problem.

Genetic programming (GP) [6], [7] has proved to be a very powerful problem solving method, an automatic programming tool and a machine learning tool. One of the advantages of the GP approach is in exploring situations where very little domain knowledge is available or the nature of the problem is not fully understood. Also genetic programming can evolve new and uncommon strategies that human designers might not consider. Another benefit of using GP is that re-training is straight-forward and the evolved classifiers can be tolerant to noise or other variations which often occur in real world environment. For these reasons we propose GP as the method for the rice leaf detection task.

There are two approaches to the use of genetic programming for object detection in images: the window approach and the whole image approach. The window method involves the evolution of a classifier of windows that just cover the object(s) of interest and applying the classifier to the images to localize the objects of interest. This approach has been used successfully for many problems including coin detection [8], [9], moving object detection [10], [11], [12] and texture discrimination [5]. The whole image approach involves the evolution of a tree of image processing operations which takes a test image as input and delivers a binary image which shows the pixels corresponding to the object(s) as black and all other pixels as white, for example [13]. Generally this approach is only used for small images due to large computation times and difficulties in generating good programs. Our primary interest in this paper is to determine whether the window approach will suit the rice leaf detection problem.

The main questions which will be investigated in this research are:

1) How to configure genetic programming for the rice leaf detection problem?
2) How to use evolved GP programs to perform detection?
3) How does the GP classifier compare to conventional classifiers?
4) How does evolutionary approach compare to a thresholding approach?

To answer these questions the paper is organized as follows. Section II briefly describes related work done by previous researchers. Section III presents the methodology of our approach including the way to represent the classifiers and the parameters of the GP runs. Section IV presents our results and a discussion. The final section presents our conclusions and avenues for future research.

978-1-4799-0454-9/13/$31.00 ©2013 IEEE

II. Related Work

Yang et al. [2] investigated the problem of distinguishing young corn plants from four species of weeds. They used a number of features, including colour, with a neural network classifier. In the case of discriminating between corn and one weed species, the highest success recognition rate was 100% for corn and 92% for Abutilon theophrasti, a kind of weed. However, they obtained low recognition rates for the other three kinds of weeds 8%, 46% and 62% respectively. They also used a five output neural network to classify corn and the four weeds species. The best corn recognition rate was 58% and the overall results were deemed unsatisfactory.

Perez et al. [1] proposed methods using color information to distinguish between vegetation and background such as soil and residue, while using shape analysis to discriminate between crop and broad-leaf weeds. They captured near-ground images that included broad-leaf weed and two or three rows of crop in natural light conditions. After using color processing to eliminate soil and residue, they determined the position of crop rows using histograms. In the context of row crops, they first detected plants in between the rows to find weeds. To discriminate between weeds and items of crop not connected to the rows, they used shape analysis to extract a feature vector of broad-leaf weeds which included s major axis length, aspect ratio, area, roundness and other heuristic features based on the experiments of researchers. They used Bayes rule and K-nearest neighbour classification to evaluate the algorithms that generated the feature vectors. The results of study for both classification methods gave an accuracy percentage 89.7% for crop and 74.5% for weeds (Bayes rule), and 89.0% for crop and 79.2% for weeds (K-nearest neighbour). However, they mentioned that the shape analysis techniques require higher computation cost if the crops were not planted in rows or difficult to determine the position of rows.

Ukrit et al. [14] proposed a lawn weed detection method based on their observation of the clearly different colours of weed and lawn in the winter. In this season, the colour of weeds is green or yellow while the colour of lawn is yellow, brown or grey. To detect weed areas, they used a simple method based on the green band, red band and a threshold of dark and bright areas. The results have been used in chemical system with 91.48% of weeds destroyed and a non-chemical system with 70.21%of the weeds destroying by sparking.

In related work using GP for object detection to classify and localize objects in large image there are a number of successful projects that used the sweeping window approach. In [8], [9] means and standard deviations of sub-windows are used as features for detecting coins of two different sizes and explored a number of fitness function variations to reduce false alarms. In [8], the authors also investigated four types of training data: exact centre, close to centre, include centre and background based on coin position in the sweeping window. They found that the first two types of data could be used to obtain good detection results.

Genetic programming has also been used to evolve novel texture features [5], [15], [16]. In this work texture feature extraction programs were evolved from examples of 13 Brodatz textures. The evolved features were competitive human derived features on a number of classification problems, including malt used for brewing beer.

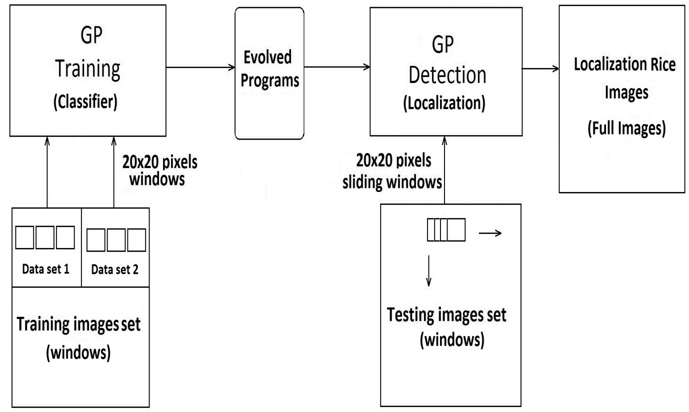


Fig. 1. The process of rice leaf detection using the GP sliding window approach.

III. Methodology

The methodology is a two step process. The first is evolving a classifier program to determine whether a particular a window is leaf or non leaf, the second is applying the evolved program to a test image to locate all of the leaf area. These steps are shown in the Figure 1.

The first step requires positive (rice leaf) and negative (other) windows to be cut from the available training images. The window size needs to be chosen carefully. Each window must be big enough to capture the object or area of interest, but not so big that there is too much extraneous data. For this work we have used a window size of 20×20.

In the second step, localization, the desired output is a binary image in which the leaf pixels are shown as black and all other pixels are white. This image is generated by applying the evolved program is applied, in sliding window fashion to test images. The evolved program can be applied with any step size that is less than the window size. If the step size is 20 then each pixel will receive one classification. If the step size is 1, then (ignoring edge effects) each pixel will receive 400 classifications and the final output will be determined by majority voting.

A. Image Preparation

The reason for detecting rice leaves is to drive a mechanical system that can target and spray a nutritional supplement on a rice leaf but not on weed leaf or soil. These systems would perform image collection with a camera mounted on a vehicle moving across a rice field. Thus real time classification is required. We have chosen an image size matched with a VGA camera which creates 640×480 image. In total, 600 images were captured by digital camera from a top view. We randomly chose 300 images for the training task and reserveded the remaining images for the localization task.

As mentioned earlier, color information is unlikely to be useful in this classification task because of the similarity in the colours of leaves and weeds. This is evident in the sample images shown in Figure 2. Furthermore, to reduce the number

Fig. 2. A sample of real images from rice fields. Rice and weed plants are interleaved and their leaves are very similar in colour. Note that the width of rice leaves is larger than that of weed leaves.

of classifier inputs we have converted the colour images to grey scale using the following formula.

$$Grayscale(pixel) = 0.3 * R + 0.59 * G + 0.11 * B \quad (1)$$

To generate positive and negative examples for GP classifier input a training set of image windows or cut-outs is needed. As noted earlier the size of these windows needs to be chosen based on the characteristics of the object or area of interest. In this case we expect that a window size of 20×20 will large enough to capture the texture of a rice leaf. If the window is larger it will contain non leaf areas which will make classification more difficult. The size could possibly be decreased to save on computational cost. However, if the size is too small (equal to or smaller than the size of weed leaves) it may not be possible to discriminate between rice and weed leaf.

The train and test cut-outs were generated manually from the 300 images chosen for training. To generate positive examples, we manually set the position of each window on a rice leaf as shown in Figure 3. Then these windows were cut out and labeled as positive. The negative examples were generated by cutting out from non leaf areas (weeds, soil, other). In total, we generated 4,000 windows of 20×20 pixels which we divided into two sets for 2-fold cross validation. In each set, we have 1,000 windows of the positive class and 1,000 of the negative class.

B. Genetic Programming Representation

In a GP application it is necessary to specify the terminals, functions and fitness measure. It is also necessary to determine the run time parameters of population size, maximum generations, maximum depth of the trees and probabilities of crossover, elitism, and mutation. We used the RMIT-GP package [17].

1) Terminals: The individual pixel values of the windows were used as the inputs to the evolved programs. Random constants could also be terminals, however, after some experimentation we found that they did not improve the accuracy of the evolved classifiers. Details of the training data are shown in Table I. In this table, every Att is the gray value of a pixel and every window has a total of 400 Atts corresponding to 400 pixels.

2) Function Set: The function set includes the arithmetic operators: addition (+), subtraction (-), multiplication (*) and protected division (/) (producing a result of zero for a zero divisor). In addition to the basic arithmetic operators we use the square root operator which is beneficial for classification using dynamic range selection[18]. Dynamic range selection is a generalization of the usual method of classifying an example. Usually the example data is passed to an evolved program and if the output is positive then the example is labelled as positive, otherwise it is labelled as negative. In dynamic range selection the real line is divided into a number of alternating positive and negative segments. If the output of the program lands in positive segment then the example is labelled as positive, otherwise it is labelled negative. Dynamic range selection generally gives shorter training times and higher accuracy than the standard method. The function set is shown in Table II.

3) Fitness Function: This is a critical part of GP configuration. A good fitness function will provide better solutions as it gives individuals with higher performance more chance to reproduce. In our study the fitness of the evolved programs is based on performance achieved in the classification task.

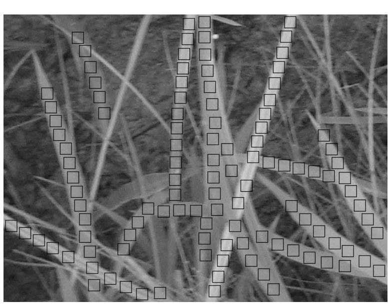


Fig. 3. Windows cut out from the original gray image. The square windows on the rice leaves are positive examples.

TABLE I. TRAINING DATA.

Window 1	Att1	Att2	Att3	...	Att400	Class
Window 2	132	134	131	...	140	Negative
Window 3	183	182	180	...	185	Positive
...	...	...				
Window 2000	173	160	165	...	120	Positive

TABLE II. FUNCTION SET

Function	Input Parameters Type	Return Type
+	Double, Double	Double
-	Double, Double	Double
*	Double, Double	Double
/	Double, Double	Double
sqrt	Double	Double

TABLE III. GENETIC PROGRAMMING CONFIGURATION FOR RICE LEAF DETECTION

Parameters	Value
Population Size	100
Maximum Depth of Program	15
Minimum Depth of Program	2
Maximum Generations	500
Mutation Rate	0.3
Crossover Rate	0.65
Elitism Rate	0.05
Fitness	Weighted TPR, TNR
Size of data fold 1	2,000
Size of data fold 2	2,000
Validation	2-fold cross validation
Number of Runs	10
Window Size (pixels)	20×20
Test Image Size (pixels)	640×480

Our fitness function is a weighted sum of the true positive and true negative rates. There are two reasons for this. First, in the final application there will be different costs associated with false positive and false negative errors. Second, the localization process could be more accurate. The fitness function is shown in Equation 2:

$$Fitness = w_P * TPR + w_N * TNR \tag{2}$$

where w_P and w_N are the weights of the true positive rate (TPR) and true negative rate (TNR) respectively. The fitness value ranges from 0% to 100%, and we apply the following condition to the weight pair:

$$w_P + w_N = 1 \tag{3}$$

The overall classification accuracy will be:

$$Accuracy = \frac{TPR + TNR}{2} \tag{4}$$

In this work, we need to give particular attention to the false negative rate (FNR), i.e. weeds or soil are classified as rice leaves. This is an important factor because there will be a cost associated with spraying nutritional supplements on weeds rather than on rice plants. The FNR is implicit in equation 2. If we increase weight w_N that means we will decrease the FNR. The evolution cannot reasonably achieve an expected FNR if the fitness function is simply classification accuracy.

4) Run-time Parameters: The run time parameter settings shown in Table III have been used for all runs. The termination condition is a maximum of 500 generations or achieving a perfect fitness value of 100%. Each training and test run was repeated 10 times.

IV. RESULTS AND DISCUSSION

In this section we first present the results for step 1, the training of the classifier and then the results for step 2, localization of the rice leaves.

A. Train/Test Results for the GP Classifier

The train and test accuracies for 3 different combinations of w_P and w_N are shown in Table IV. The table shows the true positive rate (TPR) the true negative rate (TNR) and the accuracy. Also shown are the average and best fitness. Note that these are in fact over 20 separate training and test runs (10 runs each for each of two folds). It is expected that both TPR and TNR will be higher as if their weight is higher, and this is what occurs. For example, when w_P is set to 0.3, 0.5 and 0.7 the TPR is 85.29%, 93.25% and 96.65%, respectively. This level of control is one of the benefits of fitness that is a weighted sum of TPR and TNR and is important when the costs of false positive and false negative errors are not the same.

TABLE V. DEGREE OF OVERFITTING

w_P	w_N	Variation in Accuracy	Variation in Average Fitness
0.3	0.7	**1.97**%	**2.3**%
0.5	0.5	6.32%	6.32%
0.7	0.3	6.42%	5.79%

An interesting aspect of the results is that the weight pair ($w_P = 0.3$, $w_N = 0.7$) gives the lowest values for accuracy and average of fitness function values for the training task, 89.44% and 91.10%, respectively. In the testing task, however, that pair gives the highest values which are 87.47% and 88.80%, respectively. This combination also has the lowest amount of overfitting as measured by the difference between training and test accuracy as shown in Table V. For example, in the second row of Table V, the accuracy of training and testing are 89.44% and 87.47%, respectively. As a result the variation in accuracy is 1.97%. By comparison on the training and testing results of each weighted pair, we can choose an evolved program with reasonable accuracy and with stable variation data.

B. Localization Results, No Voting

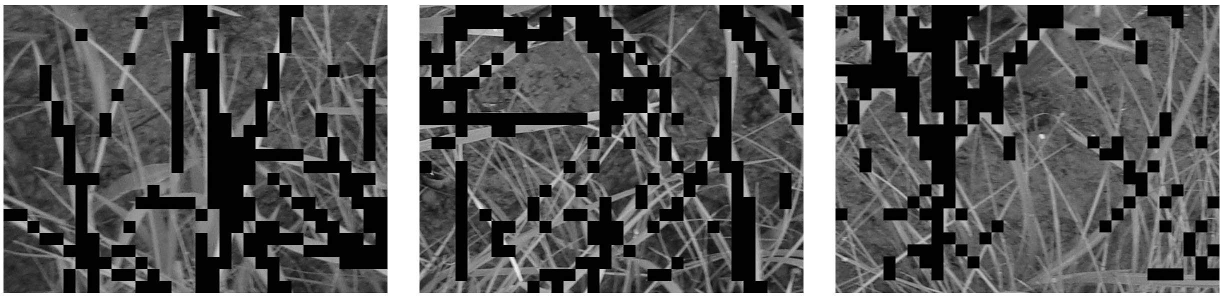
In this step one of the evolved programs is chosen as the classifier to apply to the full images. There are two factors which will affect the output binary image: (1) The values ofw_P and w_N and (2) the sliding window strategy. With respect to the sliding window strategy we investigate two alternatives, one simple and one more complex. The first alternative gives one classification for each pixel in the full image, the second gives a number of classifications which must be resolved by a voting procedure.

In this section we fix the sliding window strategy to no voting and examine the effects of different weights. If the sliding window, which is 20 pixels is moved across the full image in steps of 20 pixels, each pixel will receive only one classification and there is no need for voting..

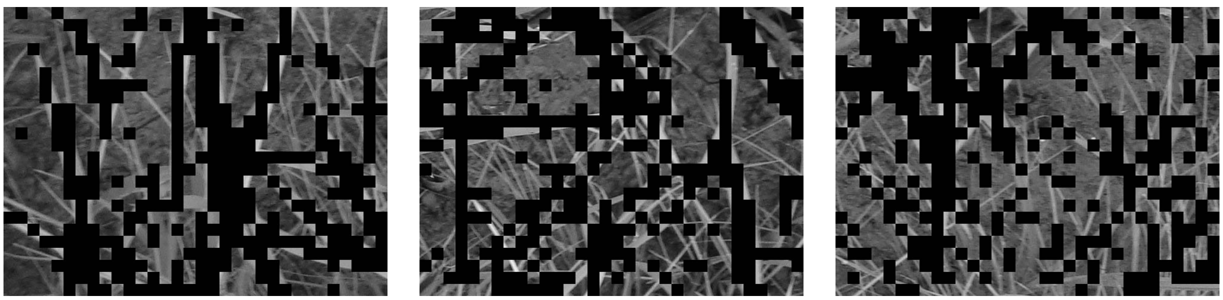
Firstly, for the pair ($w_P = 0.3$, $w_N = 0.7$), we take the evolved program of run 7 which had the following testing results: Fitness = 89.88%, Accuracy = 88.20%, TPR = 84.00% and TNR = 92.40% and apply it to three test images, those of Figure 2. The results are shown in Figure 4(a) where the black windows indicate rice leaves. As can be seen in the results images some areas of rice leaves have been missed, i.e. there is no black window over a rice leaf. Also, the black windows also appear in some areas of weed leaves or soil which indicate the number of false positive classifications. Note that for visualisation comparison purposes the binary images have been overlaid on the original images. Secondly, we repeat the previous experiment using the pair ($w_P = 0.7$, $w_N = 0.3$) and choosing the best evolved program of run 3 (Fitness = 90.28 %, Accuracy = 86.60%, TPR = 95.80% and

TABLE IV. TRAINING AND TESTING RESULTS FOR THE CLASSIFIER, 2-FOLD CROSS VALIDATION, AVERAGES OF 10 RUNS.

w_P	w_N	Training					Testing				
		TPR (%)	TNR (%)	Accuracy (%)	Fitness Function (%) Best	Fitness Function (%) Average	TPR (%)	TNR (%)	Accuracy (%)	Fitness Function (%) Best	Fitness Function (%) Average
0.3	0.7	85.29	93.59	**89.44**	92.58	**91.10**	84.14	90.80	**87.47**	90.56	**88.80**
0.5	0.5	93.25	89.07	91.16	92.80	91.16	88.53	81.14	84.84	90.09	84.84
0.7	0.3	96.65	83.37	90.01	94.64	92.67	91.83	75.34	83.59	90.28	86.88



(a)



(b)

Fig. 4. The results of localization of rice leaves on test set images with no voting. The first row is for ($w_P = 0.3$, $w_N = 0.7$). The second row is for ($w_P = 0.7$, $w_N = 0.3$).

TNR = 77.40%) to appy to the test set images. The output images are shown in Figure 4(b). Note that, in Figure 4, the same original images are shown the same columns, but they have different binary image overlays.

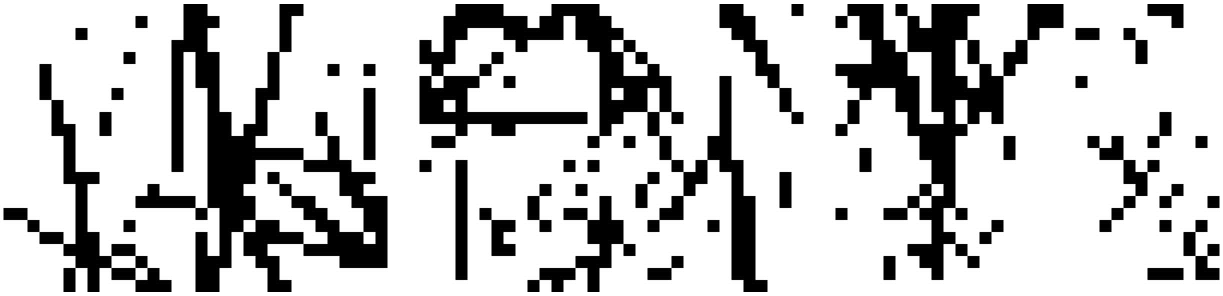
It is evident that if we increase w_P, the evolution will produce best programs with higher TPR and lower TNR, and a consequent high FNR. For example, the two FNRs of ($w_P = 0.3$) and ($w_P = 0.7$) above are 7.60% and 22.60% respectively. Therefore, the errors on the negative class will increase with higher w_P. By looking at the results of the localization images in Figures 4(a) and 4(b), we can see more clearly the that the higher FNR of an evolved program results in the more errors. These errors are the black windows that appear on the weed leaves instead of on the rice leaves. This corresponds to wasting nutritional supplements on weeds. In this application, in terms of the trade off between TPR and TNR, the TNR is more significant than TPR.

C. Localization Results, Majority Voting

The previous section gave the results for moving the sliding window 20 pixels at a time and having only one classification for each pixel. This strategy could result in errors in situations where the window only partially covers a rice leaf. Moreover, the boundaries between rice leaves and background are rough. These problems can be addressed by the moving the scanning a smaller number of pixels and having several classifications for each pixel and then determining the class by majority voting.

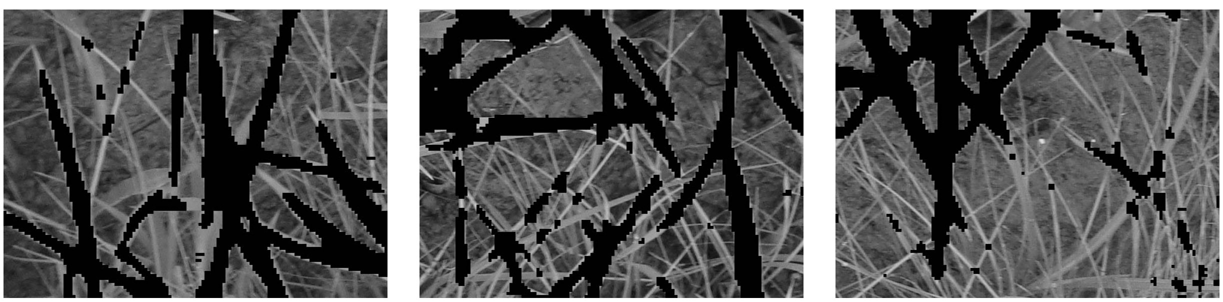
There 19 choices for the step size to use for the sliding window. We show the results for a step size of 5. This means an overlap of 15 between successive windows. Each pixel receives $15 \times 15 = 225$ votes. The final classification is the larger of the positive and negative votes. This step size is a point on the tradeoff between lower computation with a larger step size and increased accuracy with a smaller step size. In our experiments there was little difference in accuracy between a step size of 1 and a step size of 5. Figure 5 shows the results for ($w_P = 0.3$, $w_N = 0.7$), the same as the combination for Figure 4(b). The differences between before applying the majority voting and after can be seen in Figure 5(a) and Figure 5(b). Even though the same evolved program was used (Fitness = 89.88%, Accuracy = 88.20%, TPR = 84.00% and TNR = 92.40%) for Figure 4(a) and Figure 5(a), the results in Figure 5(b) and Figure 5(c) are clearly better in term of removing some black error windows located on weed leaves, achieving clearer rice leaf edges and recovering some missing black windows of rice leaves.

For visual comparison purposes we have manually marked up a perfect output image for one of the test images, Figure 6(a). This is the best possible output of the localization process. On the marked up image and output images from GP localization the black pixels and white pixels show rice leaves and non-rice objects respectively. Errors occur when



(a)

(b)



(c)

Fig. 5. Using majority voting to enhance rice areas detection and remove errors detection. Fig. (a) is the localized binary images with window size=20 pixels and step=20 pixels. Fig. (b) is the localized binary images with window size=20 pixels and step=5 pixels. Fig. (c) is the Fig. (b) overlaid on original images.

TABLE VI. COMPARISON WITH CONVENTIONAL CLASSIFIERS

	J48	Naïve Bayes	SVM	NN (K=5)	**GP**	
					Average	Best
Accuracy (%)	90.10	90.25	88.60	86.25	87.47	90.30
True Positive Rates (%)	90.80	87.70	89.60	96.30	84.14	90.40
True Negative Rates (%)	89.40	92.80	87.60	76.20	90.80	90.20

the pixels of perfect image and output GP image are different on the same pixel position. Therefore, we use the exclusive Or (XOR) logical operator these images. Note that, the XOR binary operation takes two binary inputs and outputs 0 only when both the inputs are same. The result of this process is shown in Figure 6 in which (c) is the result output image of XORing (a) and (b) to show the errors. The errors could be false positives or false negatives. It is clear that, the errors mainly occur at the edges of rice leaves. The number of these errors would probably be lower with a smaller step size. In terms of selecting areas for micro spraying of nutrient on rice leaves the results are quite satisfactory.

D. Comparison with Four Machine Learning Algorithms

In this section we compare the classification accuracy of the evolved classifiers for step 1 with four conventional classifiers: Decision trees (J48), Naïve Bayes, support vector machines (SVM) and K-nearest neighbour with $K = 5$. We have used two fold cross validation in the same way as for the GP classifier. The results, which are shown in Table VI, are for the averages of the two test folds. The GP results are the average and the best for ($w_P = 0.3$, $w_N = 0.7$).

The results for J48, Naïve Bayes, SVM and GP are comparable. K-NN is the exception, with a higher TPR but lower TNR than the other classifiers. The GP approach has the advantage that it is relatively straight forward to manipulate the tradeoff between false positive and false negatives, although, of

(a) (b) (c)

Fig. 6. Using logical operator XOR of two binary images to consider the errors pixels of positive and negative false detection. Figure (a) is the perfect binary images. Figure (b) is the output image of GP localization. Figure (c) is the result output image of XORing of Figure (a) and (b) to show the errors.

course, cost sensitive implementations of the other classifiers could be used.

E. Comparison with Threshold Approach

In this section we compare the GP localizations at step 2 with a thresholding approach based on colour. We spent considerable time trying different combinations of thresholds for R, G and B. Figure 7(b) where black shows the pixels that passed the threshold test for rice leaf. The method performs well in terms of removing soil and residue. However, comparison with Figure 6(b), which is the GP localization image, it can be seen that this method cannot differentiate between weed and rice leaves. The errors can be observed in Figure 7(c) where the perfect image is XORed with the image from the thresholding.

V. Conclusion

Our goal in this work was to determine whether a scanning window approach using classifiers evolved by genetic programming could be used for detection of rice plant leaves in images consisting of rice, weeds and soil. The results of the investigation are very promising, as the approach can differentiate the rice leaves from weed leaves which have very similar color and locate rice leaves for spraying nutrient with an acceptable accuracy.

On the question "How to configure genetic programming for the rice detection problem?" we found that a GP classifier for 20×20 windows with the four arithmetic functions augmented by square root could be used. Fitness could be a weighted sum of true positive rate and true negative rate. Use of dynamic range selection results in more accurate classifiers. On the question "How to utilize evolved GP programs to perform detection?", we found that the classifiers could be applied to test images in scanning window fashion. Majority voting gave better localizations than scanning without voting. The false positive and false negative errors could be controlled by weights in the fitness function. On the question "How does the GP classifier compare to conventional classifiers?" we found that the GP classifier was comparable in accuracy with decision trees, Naïve Bayes, and support vector machines. The GP classifier had the advantage that the tradeoff between false positive and false negative errors could be relatively easily manipulated with the weights in the fitness function. On the question "How does the evolutionary approach compare to a thresholding approach?" we found the evolutionary approach far superior. It proved to be impossible to find R, G and B thresholds to distinguish between rice and weeds.

Although we have used a GP classifier for this problem our results suggest that a number of conventional classifiers trained with a suitable cost matrix could give comparable localization performance.

While the current investigation has shown promising results, further work is needed to improve the TNR with an acceptable TPR. This could perhaps be achieved by using additional features, for example keeping the colour, or computing explicit texture features, or possibly by a secondary evolutionary process for evolving specialised discriminating features for rice and weeds. It would also be interesting to see whether a multi-objective approach with TPR and TNR as the objectives could give better classifiers.

References

[1] A. Perez, F. Lopez, J. Benlloch, and S. Christensen, "Colour and shape analysis techniques for weed detection in cereal fields," in *First European Conference for Information Technology in Agriculture*, Copenhagen, 1997, pp. 15–18.

[2] C. Yang, S. Prasher, and J.A.Landry, "Weed recognition in corn fields using back-propagation neural network models," *Canadian BioSystems Engineering*, vol. 44, pp. 715–722, 2002.

[3] E. Vrindts and J. D. Baerdemaeker, "Using spectral information for weed detection in field circumstances," *EurAgEng Paper No. 00-PA-010 in AgEng*, 2000.

[4] ——, "Optical weed detection and evaluation using reflection measurements," in *In Proceedings SPIE, Precision Agriculture and Biological Quality*, G. E. Meyer and J. A. DeShazer, Eds., vol. 3543, Bellingham, Washington, USA, 1999, pp. 279–289.

[5] V. Ciesielski, A. Song, and B. Lam, "Visual texture classification and segmentation by genetic programming," in *Genetic and Evolutionary Computation for Image Processing and Analysis*, S. Cagnoni, E. Lutton, and G. Olague, Eds. New York: Hindawi Publishing Company, 2007, pp. 195–213.

[6] J. R. Koza, *Genetic Programming: On the Programming of Computers by the Means of Natural Selection*. Boston: MIT press, 1992.

[7] R. Poli, W. B. Langdon, and F. Nicholas, *A Field Guide to Genetic Programming*. Lulu.com, 2008. [Online]. Available: http://www.google.com.vn/books?id=3PBrqNK5fFQC

[8] M. Zhang and M. Lett, "Genetic programming for object detection: Improving fitness functions and optimising training data," *IEEE Intelligent Informatics Bulletin*, vol. 7, no. 1, pp. 12–21, 2006.

[9] V. A. Patil and S. Jagtap, "Genetic programming for object detection," *International Journal of Engineering Science*, vol. 4, 2012.

(a) (b) (c)

Fig. 7. Using a threshold approach to detect rice leaves and comparison with the perfect image. Figure (a) is the perfect binary images. Figure (b) is the output image of thresholding approach. Figure (c) is the result output image of XORing of Figure (a) and (b) to show the errors.

[10] B. Pinto. and A. Song, "Detecting motion from noisy scenes using genetic programming," in *Image and Vision Computing New Zealand, 2009. IVCNZ'09. 24th International Conference*. New Zealand: IEEE Press, Nov. 2009, pp. 322–327.

[11] Q. Shi and A. Song, "Selective motion detection by genetic programming," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*. IEEE Press, Jun. 2011, pp. 496–503.

[12] Q. Shi, Y. Wei, and A. Song, "Analysis of motion detectors evolved by genetic programming," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. Brisbane Australia: IEEE Press, Jun. 2012, pp. 1–8.

[13] T. Kowaliw, W. Banzhaf, N. Kharma, and S. Harding, "Evolving novel image features using genetic programming-based image transforms," in *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, A. Tyrrell, Ed., IEEE Computational Intelligence Society. Trondheim, Norway: IEEE Press, May 2009, pp. 2502–2507.

[14] U. Watchareeruetai, Y. Takeuchi, T. Matsumoto, H. Kudo, and N. Ohnishi, "A lawn weed detection in winter season based on color information," in *Proc. of IAPR Conf. MVA*, Tokyo, Japan, 2007, pp. 524–527.

[15] V. Ciesielski, B. Lam, and M. L. Nguyen, "Comparison of evolutionary and conventional feature extraction methods for malt classification," in *Proceedings of the 2012 Congress on Evolutionary Computation (CEC2012)*. Brisbane, Australia: IEEE Press, Jun. 2012, pp. 1–7.

[16] B. Lam and V. Ciesielski, "Discovery of human competitive image texture feature extraction programs using genetic programming," in *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO 2004)*, K. D. et al., Ed., vol. 2. Springer, Jun. 2004, pp. 1114–1125.

[17] V. Ciesielski. (2006) RMIT-GP: The RMIT University Genetic Programming System. [Online]. Available: http://www.cs.rmit.edu.au/\~vc/rmitgp

[18] T. Loveard, "Genetic programming methods for classification problems," Ph.D. dissertation, Univ. of RMIT, Australia, May 2003.