isrjournals

International Journal of Advanced Research in Computer Science Engineering and Information Technology

Volume: 4, Issue: 3,Special Issue: 3 ,Apr,2016 ,ISSN_NO: 2321-3337

Identification of Weed in Agricultural Field By Using Video Matting and Image Segmentation

R.RAJBHARATH[1], R.HEERA[2]
Assistant Professor[1], PG Scholar[2]
Dept.of Computer Science, Manakula Vinayagar Institute Of Technology[1, 2]
Puducherry, India.

.

ABSTRACT *- The main objective of the paper is to evaluate a software solution for automatic identification of weed from original crops .The identification process is achieved by implementing image segmentation and Video Matting techniques. The aim of the project is to detect and classify weeds from different species of huge cultivated crops in our country like paddy, wheat, brinjal, tomato other cash crops and oil seeds etc. The plant identification species were taken for our approach is first taken as trained set of data by collecting its details such as its size, color, texture and shape. The trained data is processed till its complete parameters are identified clearly in order to classify it from other crops. The experimental results indicate the proposed approach can recognize and classify the crop identification with a little computational effort. The proposed method mainly focuses on analyzing and identification of weeds which is attained by identifying the non similar parameters to the trained set of data. The inputs for the system are feed as videos of the cultivated crops which is processed by video matting technique and converted into frames by frame diffusion algorithm .The related structure and segmenting parameters like size, color, texture and shape were collected as testing set of data from the converted frames . The segmented parameters where analyzed with the available training set of data and the result is generated as to identify the weed from the original crops.*

Keywords— Image Segmentation, Video Matting, Support Vector Machine, One Class Support Vector Machine, Foreground Segmentation.

1. INTRODUCTION

The main purpose of the paper is to analyze Foreground segmentation, through which we can extract objects of interest from input videos [1]. It is a fundamental problem in computer vision and often serves as a pre-processing step for other video analysis tasks such as surveillance, teleconferencing, action recognition and retrieval. Over the years a significant amount of related techniques have been proposed in both computer vision and graphics communities. However, some of them are limited to sequences captured by stationary cameras, while others require significant amount of training examples or cumbersome user interactions [3]. Furthermore, most existing algorithms are rather complicated and computationally too demanding to be operated in real time. As a result, there still lacks an efficient and powerful algorithm capable of processing challenging live video scenes with minimum user interactions.[4] Motivated by the above finding, we here present a novel integrated foreground segmentation and boundary matting approach, which is an extension to our preliminary work on foreground segmentation. The algorithm is able to propagate

Volume: 4, Issue: 3,Special Issue: 3 ,Apr,2016 ,ISSN_NO: 2321-3337

labeling information to neighboring pixels through a simple train-re label-matting procedure, resulting in a proper segmentation of the frame [4]. This same procedure is used for further propagation to label information across adjacent frames, regardless of the foreground or background motions. Several techniques are also proposed in order to reduce computational costs. Furthermore ,by exploiting the parallel structure of the proposed algorithm, real-time processing speed of 14 frames per second (FPS) is achieved for VGA-sized videos when matting is not applied, with the frame rate dropping to 8 FPS when matting over large fuzzy areas is needed[4]. The key insight of our approach is to maintain two competing one-class Support Vector Machines (C-1SVMs) at every pixel location. The two one-class Support Vector Machines (1SVMs) capture the local foreground and background color densities separately, but determine a proper label for the pixel jointly. By iterating between training local C-1SVMs and applying them to label the pixels, the algorithm effectively propagates initial user labeling to the entire image, as well as to consecutive frames. Finally, using two 1SVMs to model foreground and background color is attributions separately facilitates the matting calculation along object boundaries, making it possible to solve foreground segmentation and boundary matting problems in an integrated manner[2].

2. EXSISTING SYSTEM

The existing system involves in classification of Fuzzy objects [1]. This system not only classify Fuzzy objects but also gives the group index, the general sub-grade rating as well as possible material sample is made of. It also helps in reducing the tedious work of having to go through the chart over and over again. The system has indicated that MATLAB is not only for calculation and analysis but also can be used in analysis involving numeric's and word problems simultaneously.

2.1. Methodologies Used In Existing System

- Video Segmentation
- Foreground Segmentation
- Video Matting

2.2. Limitations

- Less accuracy in object detection and necessity of background model.
- Complexity in background updating and processing time.
- Handling novel foreground colors.
- Low Resolution in handling images.
- Handling Illumination changes.
- Handling large Semi-Transparent objects.

Volume: 4, Issue: 3,Special Issue: 3 ,Apr,2016 ,ISSN_NO: 2321-3337

3. PROPOSED SYSTEM

This paper mainly focuses on identifying the weeds in agricultural environment through which we can analyze the fertility of the crops and also categorizes the nature of crops that can be cultivable from that particular landmass. The data with respect to the crop required for cultivation in accordance to its fertility where collected as training set of data. The data such as videos of landmass were the crops are grown are taken and necessary parameters like size, color, texture and shape etc are analyzed and stored. The segmented inputs where analyzed with the available training set of data and deliver the result on identification of weeds. Here the weeds can be identified by comparing the parameters with the input video files and stored data sets. If the stored data sets and input video files are matched, then it is said to be the same variety of crops and will display the output as “Not a weed ", else the input video file is not a part of the stored data set and it will be displayed as “It’s a weed ".

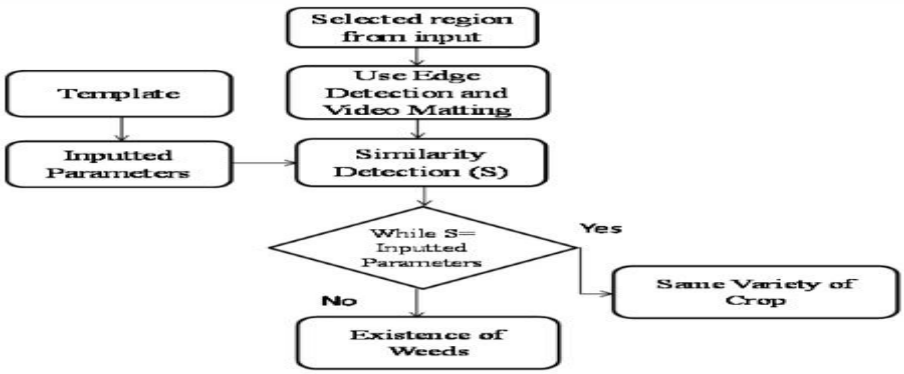

Fig. 1. Block Diagram of Proposed System

3.1. Methodologies Used In Proposed System

Object identification based on

- Gaussian kernel
- Probability Density Estimation
- Probability Density Function
- Edge Detection

3.2. Related Algorithms

Pseudo code for Frame Diffusion Method

Step 1. Select the Video and to extract frames and get frame means from an avi movie format

Step 2. Save individual frames to separate image files.

Step 3. Also computes the mean gray value of the color channels.

Step 4. Clear the command window and Close all figures

Step 5. Change the current folder to the folder of this m- file.

Step 6. Prepare a figure to show the images in the upper half of the screen.

Step 7. Comparing the String and retrieving file.

Step 8. Loop through the movie, writing all frames out and each frame will be in a separate file with unique name.
Step 9. Write the image array to the output file, if requested.
Step 10. Construct an output image file.

3.3. SVM Classification

As with any supervised learning model, you first train a support vector machine, and then cross validate the classifier [1]. Use the trained machine to classify (predict) new data and satisfactory predictive accuracy can use various SVM kernel functions, and tune the parameters of the kernel functions.

- **Training an SVM Classifier**

Train, and optionally cross validate, an SVM classifier using fitcsvm [1]. The syntax is:
SVMModel=fitcsvm(X,Y,'KernelFunction','rbf','Standardize',true,'ClassNames',{'negClass','posClass'});

The inputs are:

- X — Matrix of predictor data, where each row is one observation, and each column is one predictor.
- Y — Array of class labels with each row corresponding to the value of the corresponding row in X. Y can be a character array, categorical, logical or numeric vector and vector cell array of strings. Column vector with each row corresponding to the value of the corresponding row in X. Y can be a categorical or character array, logical or numeric vector and cell array of strings.
- Kernel Function — the default value is 'linear' for two-class learning, which separates the data by a hyper plane. The value 'rbf' is the default for one-class learning, and uses a Gaussian radial basis function. An important step to successfully train an SVM classifier is to choose an appropriate kernel function.
- Standardize — Flag indicating whether the software should standardize the predictors before training the classifier.
- Class Names — Distinguishes between the negative and positive classes, or specifies which classes to include in the data. The negative class is the first element (or row of a character array), e.g., 'negClass', and the positive class is the second element (or row of a character array), e.g., 'posClass'. Class Names must be the same data type as Y. It is good as to practice to specify the class names, especially if you are comparing the performance of different classifiers. The resulting, trained model (SVMModel) contains the optimized parameters from the SVM algorithm, enabling to classify the new data. For more name-value pairs to control the training, see the fitcsvm reference page[7].

3.4. Classifying New Data with An SVM Classifier

Classify new data using predict. The syntax for classifying new data using a trained SVM classifier (SVMModel) is:

[Label, score] = predict (SVMModel, newX);

Volume: 4, Issue: 3,Special Issue: 3 ,Apr,2016 ,ISSN_NO: 2321-3337

The resulting vector and label represents the classification of each row in X, score is an n-by-2 matrix of soft scores. Each row corresponds to a row in X, which is a new observation. The first column contains the scores for the observations being classified in the negative class, and the second column contains the scores observations being classified in the positive class. To estimate posterior probabilities rather than scores, first pass the trained SVM classifier (SVMModel) to fit Posterior, which fits a score-to-posterior-probability transformation function to the scores. The syntax is

ScoreSVMModel = fitPosterior (SVMModel, X, Y)

3.5. Tuning an SVM Classifier

Try tuning parameters of your classifier according to this scheme:

Pass the data to fitcsvm, and set the name-value pair arguments 'KernelScale','auto'. The software uses a heuristic procedure to select the kernel scale. The heuristic procedure uses sub-sampling. Therefore, to reproduce the results, set a random number seed using rng before training the classifier. Cross validate the classifier by passing it to crossval. By default, the software conducts 10-fold cross validation. Pass the cross-validated SVM model to kFoldLoss to estimate and retain the classification error. Retrain the SVM classifier, but adjust the 'KernelScale' and 'BoxConstraint' name-value pair arguments.

BoxConstraint: One strategy is to try a geometric sequence of the box constraint parameter. For example, take 11 values, from 1e-5 to 1e5 by a factor of 10. IncreasingBoxConstraint might decrease the number of support vectors, but also might increase training time.

KernelScale: One strategy is to try a geometric sequence of the RBF sigma parameter scaled at the original kernel scale. Do this by:

i. Retrieving the original kernel scale, e.g., ks, using dot notation: ks = SVMModel.KernelParameters.Scale.

3.6. Read the Training Data

A set of classification patterns including all inputs and perhaps the associated ideal outputs, using individual data to update the behavior of a classification system and by using collective data to assess the performance of the system.

3.7. SVM Test the Data

A set of classification patterns including inputs and ideal outputs uses only to measure the performance of a classification system; individual input patterns are never used to change or update the behavior of the classifier.

P = graycomatrix (im, levels, distances, angles)

Calculates the gray-level co-occurrence matrix P of a gray-level image im. P is a 4-dimensional matrix (histogram). The value P(i,j,d,theta) is the number of times that gray-level j occurs at a distance d and at an angle theta from gray-level i. im is the input image which should contain integers in [0, levels-1], where levels indicate the number of gray-levels counted (typically 256 for an 8-bit image). Distances and angles are the vectors of the different distances and angles to use.

3.8. GLCM Algorithm

The Gray Level Co-occurrence Matrix (GLCM) and associated texture feature calculations are the image analysis techniques. Given an image composed of pixels each with an intensity (a specific gray level), the GLCM is a tabulation of how often different combinations of gray levels co-occur in an image or an image section. Texture feature calculations use the contents of the GLCM to give a measure of the variation in intensity (a.k.a. image texture) at the pixel of interest.

The virtual variable is created in the following way:

Quantize the image data: Each sample on the echogram is treated as a single image pixel and the value of the sample is the intensity of that pixel. These intensities are then further quantized into a specified number of discrete gray levels as specified under Quantization.

Create the GLCM: It will be a square matrix N x N in size where N is the Number of levels specified under Quantization. The matrix is created as follows:

i. Let 'S' be the sample under consideration for the calculation.

ii. Let 'W' be the set of samples surrounding sample 'S' which fall within a window centered upon sample s of the size specified under Window Size.

Normalize the GLCM: Divide each element by the sum of all elements The elements of the GLCM may now be considered probabilities of finding the relationship i, j (or j, i) in W. Calculate the selected feature. This calculation uses only the values in the GLCM.

4. MODULES DESCRIPTION

4.1. Estimation of Foreground or Background Objects

Browse the video file in order to upload four different types of outputs. They are adaptive in background which shows the slow motion of the video. The next one is capturing the video in the form of frames. Thus the uploaded video is converted into frames. Finally binaries the moving object id is obtained. This is done with the help of edge detection.

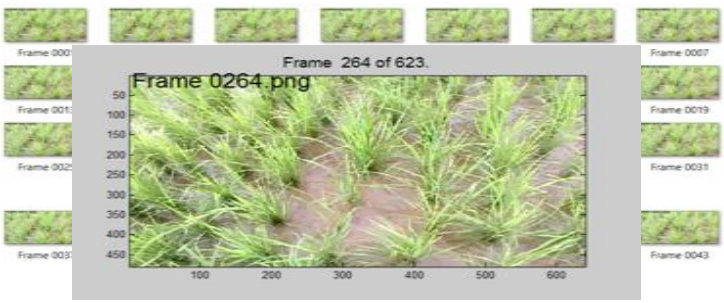

Fig. 2. Video browsing

Fig. 3. Frame Conversion

4.2. Performing Levels of Accuracy

The next module is to perform the level of accuracy. This in turn will save the file in the particular path location so that it can be retrieved back easily whenever we are in need of it.

4.3. Load the Training Set Parameters

The next module focuses on loading the training set. The training set consists of the parameters like size, shape, color, texture etc. This can be done with the help of using sobel filter mechanism.

4.4. Load the Testing Set Parameters

Load the testing folder that is saved in a video file. This is done by using the method of SVM. Repeat the process for several times until the result is obtained.

4.5. Perform Comparison between Tested Set and Training Set

Between the tested and training set of parameters a comparison is made. So that the identification of weed can be obtained within a short period of time.

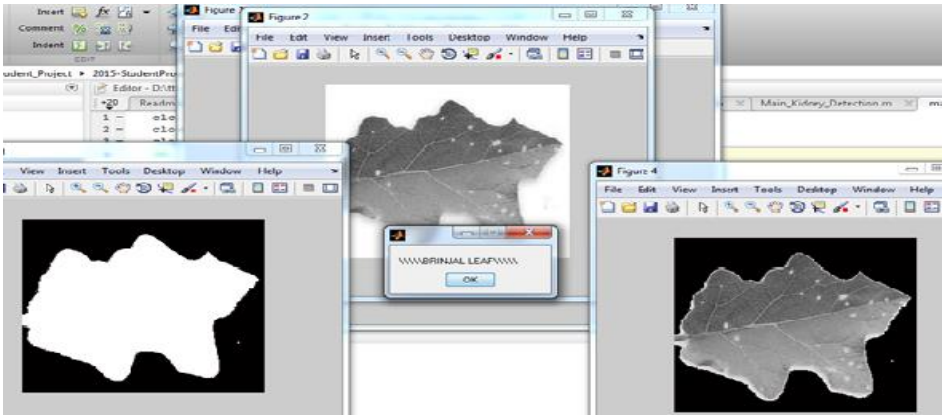

Fig 6. Similar Input

5. CONCLUSION

The main idea is to identify and classify the objects were done successfully. In the future work, will compare the identified object with the sample object. If that comparison is similar to each other, then it is said to be as sample object type, if not it is said to be as weeds. A new efficient algorithm is used for object identification and its parameters like size, shape, color. Background is evaluated using PDF and Gradient Vectors in efficient way with minimum noise. Using Feature Match, object detection plays a major role in various fields. Thus the proposed algorithm provides significantly to improve the performance and objects are to be identified and detected in effective way for Surveillance and Agricultural field.

6. REFERENCES

[1] J. L. Bentley, "Integrated Foreground Segmentation and Boundary Matting for live Videos," Commun. ACM, vol. 18, no. 9, pp. 509–517, 1975.

[2] R. F. Sproull, "Geodesic matting:A framewrok for last interactive image and video segmentation and matting.," Algorithmic a, vol. 6, no. 4, pp. 579–589, 1991.

[3] S.Korman and S.Avidan ," Realtime background subtraction from dynamic scenes", in proc.ICCV 2011,pp.1607-1614.

[4] C.Barnes , E.Shectman, A.Finkelstein , and D.B .Goldman, "Patch match: A randomized correspondence algorithm for structural image editing", ACM Trans. Graph.,vol.28, no. 3,pp.1-8,2009.

[5] S.A.Ramakanth and R.V.Babu, "Defocus video matting.,"Comput.Med.Imag.Graph" vol.38,no.1,pp.49-56,2014.

[6] Y.Hacohen, E.Shechtman, D.B.Goldman, and D.Lischinski,"Non-rigid dense correspondence with applications for image enhancement", ACM Trans. Graph.,vol.30, no.4,pp.1-70,2011.

[7] A. Yilmaz, O. Javed, and M. Shah, "background subtraction for freely moving cameras," ACM Comput. Surv., vol. 38, Dec. 2012

[8] S. Hinz, R. Bamler, and U. Stilla, "Scribble tracker: A matting based approach for robust tracking," ISPRS J. Photogramm. Remote Sens., vol. 61, no. 3–4, pp. 135–280, 2011

[9] I. Szottka and M. Butenuth, "Automatic real time video matting using time-of-flight camera and multichannel poisson equations," in Proc. 2011 Joint Urban Remote Sensing Event (JURSE), Apr. 2011, pp. 13–16.

[10] K. Palaniappan, F. Bunyak, P. Kumar, I. Ersoy, S. Jaeger, K. Ganguli, A. Haridas, J. Fraser, R. Rao, and G. Seetharaman, "Video sanpcut:robust video object cutout using localized classifiers," in Proc. 13th Conf. Information Fusion (FUSION),Jul. 2010, pp.