Australian Journal of
Crop Science
AJCS

AJCS 10(9):1232-1237 (2016) ISSN:1835-2707
DOI: 10.21475/ajcs.2016.10.09.p6661

Development and testing of image processing algorithm to estimate weed infestation level in corn fields

Wesley E Santiago*, Neucimar J Leite, Bárbara J Teruel, Manoj Karkee, Carlos AM Azania, Renan Vitorino

The main parts of the MATLAB code designed for detecting weeds in the images has been attached here.

```
function class_by_size()

% This application was made to estimate of weed coverage and identify the level
of infestation in a single image area
%
% The image dataset has 50 samples, 1 attribute (estimate of weed coverage) and
3 classes (1- Low, 2- Intermediate and 3- High).
%
% @author: Santiago, Wesley Esdras.
% @email:  wesley.santiago@ufvjm.edu.br
% @release-date: 25 Jun. 2015
%
% +Paths of program/
%  -code/          % functions .m of program
%  +data/          % folder where the dataset is stored
%  +results/       % folder containing the files .mat (training)
%
%
clear all;
warning off;
close all;
clc
%
fprintf('======================================================\n');
fprintf('Processing images to estimate of weed coverage\n');
fprintf('======================================================\n');

%%#####################################################################
%####### Recovering the way where the images are stored

data_path = '../data/';
mat_path = [results_path 'mat/'];

images_names = dir(fullfile(data_path, '*.jpg'));
num_images = size(images_names,1);
image_paths = cell(num_images,1);
for i = 1:num_images
    image_paths{i} = fullfile(data_path, images_names(i).name);
end
images_names = {images_names.name}';

%%##################################################################
%####### Reading and processing each image
```

```
for img_i=1:size(images_names,1)
tic;
I2 = imread(image_paths{img_i});

disp (['Reading image img_' num2str(img_i) ' .JPG']);

r = I2(:,:,1);    g =I2(:,:,2);    b = I2(:,:,3);

%%#####################################################################
%####### Vegetation segmentation using Euclidean distance

img_d = cast(I2,'single')./255; % convert to 'single float' and normalize
pcd = (sqrt( img_d(:,:,1).^2 + abs(img_d(:,:,2)-1).^2 ));
tsh = graythresh(pcd);
pcd(pcd < tsh) =0;
pcd(pcd >= tsh) = 1 ;
msk1=~dif;

%%#####################################################################
%####### Morphological operation and remotion of small objects as noise

se = strel('disk',3);
msk1 = imopen(msk1,se);
L = bwlabel(msk1);
data = regionprops(L,'Area');
area = [data.Area];
brus = find (area < 60);
tabort = ismember (L, brus);
segbin = imsubtract(msk1,tabort);
segbin = im2bw(segbin);
msk1=segbin;

%%#####################################################################
%##### Identifying weeds at a single image

[kmeans centers] = kmeansPlus([area],2);
threshold = min(centers(1,:));
weed = find (area > threshold);
tabort = ismember (L, weed);
segbin = imsubtract(msk1,tabort);
segbin = im2bw(segbin);
msk2 = segbin;

%%###################################################################
%######## Creating a new colored image with just weeds

WEED(:,:,1) = (single(r)) .* (single(msk2)) ;
WEED(:,:,2) = (single(g)) .* (single(msk2)) ;
WEED(:,:,3) = (single(b)) .* (single(msk2)) ;

%%#####################################################################
%### Getting the level of weed infestation or estimate of weed coverage

ppd = sum(sum(msk2));
[Lim Cim]= size(WEED);
area_img = Lim*Cim;
Iin = ((ppd/area_img)*100);
time = toc;
```

```
ind_cover = [];
ind_cover(img_i,1)=Iin;

end

%%##############################################################################
###########
% The Adaptative Neuro-Fuzzy Classifier was written by Dr. Bayram Cetiþli -
Suleyman Demirel University Computer Engineeering Isparta Turkey
%
%Ceti?li B (2010) Development of an adaptive neuro-fuzzy classifier using
linguistic hedges: Part 1. Expert Sys. Appl. 37(8):6093-6101.

disp (['Starting classification']);
load(sprintf('%stargets.mat', mat_path),'targets');
target_tr = [];
target_te = [];
target_tr = targets(1:30,1);
target_te(1:20,1) = targets(31:50,1);

data_tr = ind_cover(1:30,1);
data_te(1:20,1) = ind_cover(31:50,1);

[fismat,outputs,recog_tr,recog_te,labels,performance]=scg_nfc(data_tr,target_tr,
data_te,target_te,100,3,1);

end
```