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.

```matlab
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('======================================================
');
fprintf('Processing images to estimate of weed coverage
');
fprintf('======================================================
');

%%#####################################################################
%####### 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);
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(cenmasses1,:));
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;
\texttt{ind\_cover = [\ ];}
\texttt{ind\_cover(img\_i,1)=Iin;}

\texttt{end}

\texttt{%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%}
\texttt{The Adaptative Neuro-Fuzzy Classifier was written by Dr. Bayram Cetibili -}
\texttt{Suleyman Demirel University Computer Engineering Isparta Turkey}

\texttt{disp ('Starting classification')};
\texttt{load(sprintf('%stargets.mat', mat\_path), 'targets');}
\texttt{target\_tr = [];}
\texttt{target\_te = [];}
\texttt{target\_tr = targets(1:30,1);}
\texttt{target\_te(1:20,1) = targets(31:50,1);}

\texttt{data\_tr = ind\_cover(1:30,1);}
\texttt{data\_te(1:20,1) = ind\_cover(31:50,1);}

\texttt{[fismat,outputs,recog\_tr,recog\_te,labels,performance]=scg\_nfc(data\_tr,target\_tr,}
\texttt{data\_te,target\_te,100,3,1);}

\texttt{end}