Detection of Coffee Seeds in Images by the Color

Fernando Pujaico Rivera, *University of Lavras*

Here is presented a techniques to the detection of coffee seeds by the use of *RGB* pixel color.

1 Data set and model

The main problem, in this work, consist into detect the coffee seeds in a picture like the showed in Fig. 1.

Figure 1: *Source image.*

1.1 Model

Here a classification model is proposed through the function $f_{\mathbf{c}}: \mathbb{R}^3 \rightarrow \mathbb{R}$,

$$f_{\mathbf{c}}(\mathbf{x}) = \frac{1}{1 + e^{-h_{\mathbf{c}}(\mathbf{x})}}, \tag{1}$$

$$h_{\mathbf{c}}(\mathbf{x}) = c_1 + c_2 x_1 + c_3 x_2 + c_4 x_3, \tag{2}$$

where the column vector $\mathbf{x} \in \mathbb{R}^3$ represents a pixel, with the *RGB* additive color model, $\mathbf{x} = [x_1,\ x_2,\ x_3]^{\mathrm{T}} \equiv [red,\ green,\ blue]^{\mathrm{T}}$ in the Fig. 1; thus, the value $y = f_{\mathbf{c}}(\mathbf{x})$ indicates the state or label of sample $\mathbf{x}$. We consider that all color points in the Fig. 1 are separated principally in two sets (states); one set with color points that belong to a coffee seed, with y values close to 1, and another with everyone who doesn't belong, with y values close to 0. The use of kernel function $h_{\mathbf{c}}(\mathbf{x})$ indicates that we assume that we can separate (approximately) these two sets by the hyper-plane $h_{\mathbf{c}}(\mathbf{x}) = 0$.

The function $f_{\mathbf{c}}(\mathbf{x})$, that is a modification of a sigmoid function, has the characteristic that returns values between 0 and 1, being that take a value 0.5 when $h_{\mathbf{c}}(\mathbf{x}) = 0$; remembering that the function $f_{\mathbf{c}}(\mathbf{x})$ only reaches 0 and 1 at $-\infty$ and $+\infty$, respectively; thus, our objective in the next sections will be found the column vector $\mathbf{c} \in \mathbb{R}^4$, where $\mathbf{c} = [c_1,\ c_2,\ c_3,\ c_4]^{\mathrm{T}}$, that characterize the function $f_{\mathbf{c}}(\mathbf{x})$.

1.2 Training

To get the parameter vector $\mathbf{c} = [c_1,\ c_2,\ c_3,\ c_4]^{\mathrm{T}}$ of function $f_{\mathbf{c}}(\mathbf{x})$, we need a set of training data like shown in the Fig. 2, where the picture indicates with

Figure 2: *Training data.*

the white pixels, the approximate location[1] of coffee seeds, and the black pixels where not.

Thus, using the pictures of Fig 1 and 2, we can training the model proposed in the function $f_{\mathbf{c}}(\mathbf{x})$ to get the vector $\mathbf{c} = \hat{\mathbf{c}}$ that optimizes the fit of training information.

2 Training the model $f_{\mathbf{c}}(\mathbf{x})$

When we training the model $f_{\mathbf{c}}(\mathbf{x})$ our objective is found the vector $\mathbf{c} = \hat{\mathbf{c}}$, with the parameters of model, that minimise the error when we try fit the model with the training information. This information is located in

- $\mathbf{I}_c$: The colour image shown in Fig. 1.
- $\mathbf{I}_{bw}$: The black and white image shown in Fig. 2.

The Fig 3 represents this procedure.

The training block follow the procedure described in the Algorithm 1, that additionally need some variables like $\epsilon = 0.00005$, $N = 11023$ and $K = 10$ to work.

- ϵ: The value y to each pixel $\mathbf{x}$ chosen in image $\mathbf{I}_c$ with a "black" pixel in image $\mathbf{I}_{bw}$, where $0 < \epsilon \ll 0.5$.
- N: The number of randomly selected points (pixels) in each training repetition.

[1]Manually painted.

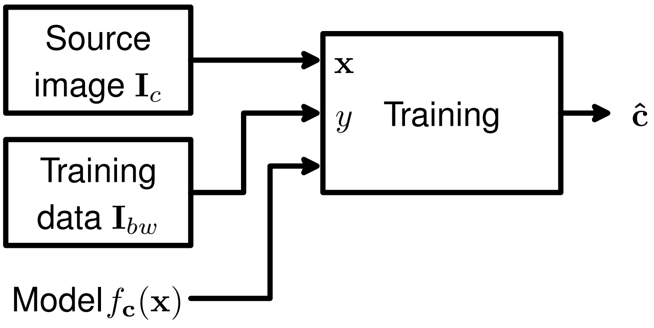

Figure 3: *Block diagram of model training.*

- K: The number of training repetitions.

Data: $\mathbf{I}_c$, $\mathbf{I}_{bw}$, ϵ, N, K.
Result: The optimised $\hat{\mathbf{c}}$ parameter to form the classifier $f_{\hat{\mathbf{c}}}(\mathbf{x})$.

$\mathbf{y} = [\underbrace{\epsilon \quad \dots \quad \epsilon}_{N\ elements} \quad \underbrace{1-\epsilon \quad \dots \quad 1-\epsilon}_{N\ elements}]^{\mathrm{T}}$;
for $k = 1$ **to** K **do**
  $\mathbf{P}_b = \mathit{get\_points}(\mathbf{I}_c, \mathbf{I}_{bw}, N, \text{“}\mathbf{black}\text{”})$;
  $\mathbf{P}_w = \mathit{get\_points}(\mathbf{I}_c, \mathbf{I}_{bw}, N, \text{“}\mathbf{white}\text{”})$;
  $\mathbf{P} = \begin{bmatrix} \mathbf{P}_b \\ \mathbf{P}_w \end{bmatrix}$;
  $[\hat{\mathbf{c}}_k,\ e(\hat{\mathbf{c}}_k)] = \mathit{get\_parameters}\,(\mathbf{P}, \mathbf{y})$;
end
$\hat{\mathbf{c}} = \frac{\sum_{k=1}^{K} \frac{\hat{\mathbf{c}}_k}{e(\hat{\mathbf{c}}_k)}}{\sum_{k=1}^{K} \frac{1}{e(\hat{\mathbf{c}}_k)}}$;

Algorithm 1: Method to get the vector $\hat{\mathbf{c}}$.

2.1 Function $\mathit{get\_points}(\mathbf{I}_c, \mathbf{I}_{bw}, N, TYPE)$

The function choose randomly[2] N points (pixels) $\mathbf{x}_n$, $1 \leq n \leq N$, from the colour image $\mathbf{I}_c$, with the condition that each point selected in $\mathbf{I}_c$ need be of type $TYPE$ in the same position of image $\mathbf{I}_{bw}$; where $TYPE \in \{\text{“}\mathbf{black}\text{”}, \text{“}\mathbf{white}\text{”}\}$. Thus, the function return the matrix $\mathbf{P} \in \mathbb{R}^{N \times 3}$,

$$\mathbf{P} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_n^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_N^{\mathrm{T}} \end{bmatrix}, \tag{3}$$

with the information of N randomly selected black or white points.

[2]Follow an uniform distribution.

2.2 Function $get\_parameters\,(\mathbf{P}, \mathbf{y})$

Given, a group of L points $\mathbf{x}_l \in \mathbb{R}^3$, labelled with the values $y_l \in \mathbb{R}$, $1 \leq l \leq L$; ordering in the matrices $\mathbf{P} \in \mathbb{R}^{L\times 3}$ and $\mathbf{y} \in \mathbb{R}^L$ respectively,

$$\mathbf{P} = \begin{bmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_l^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_L^{\mathrm{T}} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_l \\ \vdots \\ y_L \end{bmatrix}; \tag{4}$$

if we want to create a classifier using the function $f_{\mathbf{c}}(\mathbf{x})$, with domain $\mathbf{x} \in \mathbb{R}^3$, range $y \in \mathbb{R}$ and parameters grouped in the vector $\mathbf{c} \in \mathbb{R}^4$, as defined in Eq. (5),

$$y = f_{\mathbf{c}}(\mathbf{x}), \tag{5}$$

or your equivalent

$$logit(y) = h_{\mathbf{c}}(\mathbf{x}), \tag{6}$$

where $logit(y) \equiv ln\left(\frac{y}{1-y}\right)$. We can affirm that the vector $\mathbf{c} = \hat{\mathbf{c}}$ that fit the model $f_{\mathbf{c}}$ between the samples $\mathbf{x}_l$ and the labels y_l, and minimise the square error $e(\mathbf{c})$,

$$\begin{aligned} e(\mathbf{c}) &= \frac{1}{L}\sum_{l=1}^{L} ||h_{\mathbf{c}}(\mathbf{x}_l) - logit(y_l)||^2, \\ &= \frac{1}{L}\sum_{l=1}^{L} ||\begin{bmatrix}1 & \mathbf{x}_l^{\mathrm{T}}\end{bmatrix}\mathbf{c} - logit(y_l)||^2, \\ &= \frac{1}{L}||\mathbf{A}\mathbf{c} - \mathbf{z}||^2, \end{aligned} \tag{7}$$

$$\mathbf{A} = \begin{bmatrix} 1 & \mathbf{x}_1^{\mathrm{T}} \\ \vdots & \vdots \\ 1 & \mathbf{x}_l^{\mathrm{T}} \\ \vdots & \vdots \\ 1 & \mathbf{x}_L^{\mathrm{T}} \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} logit(y_1) \\ \vdots \\ logit(y_l) \\ \vdots \\ logit(y_L) \end{bmatrix}, \tag{8}$$

can be found by the Eq. (9),

$$\hat{\mathbf{c}} = \left[\mathbf{A}^{\mathrm{T}}\mathbf{A}\right]^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{z}, \tag{9}$$

and affirm that $[\hat{\mathbf{c}},\ e(\hat{\mathbf{c}})] = get\_parameters\,(\mathbf{P}, \mathbf{y})$.

3 Results

Following the procedure described in the Algorithm 1, we can obtain the vector $\hat{\mathbf{c}}$ that optimize the model fitting of $f_{\hat{\mathbf{c}}}(\mathbf{x})$ between the training information; so that

$$\hat{\mathbf{c}} = \begin{bmatrix}0.13510 & -0.24261 & 0.10244 & -1.95836\end{bmatrix}, \tag{10}$$

and consequently

$$f_{\hat{\mathbf{c}}}(\mathbf{x}) = \frac{1}{1 + e^{-0.13510+0.24261x_1-0.10244x_2+1.95836x_3}}. \tag{11}$$

A Fig. 4 shows two sets, of N points each one, selected randomly in the Fig. 2, the points $\mathbf{x}$ drawn with the color red represent pixels in a coffee seeds, and the points $\mathbf{x}$ drawn with the color blue represent pixels that don't belong to a coffee seed.

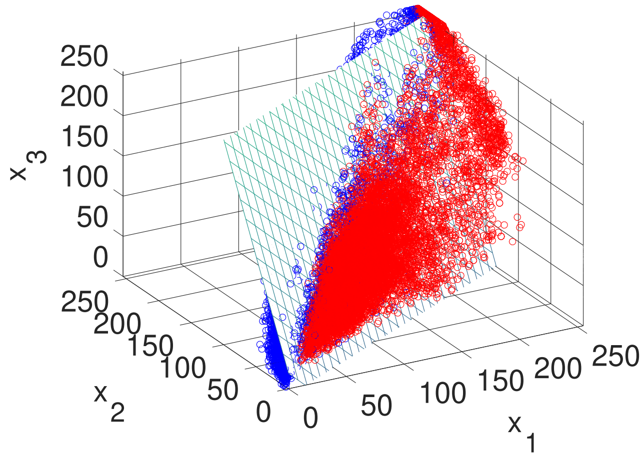

Figure 4: *Points* $\mathbf{x} \in \mathbb{R}^3$ *and the hyperplane* $h_{\hat{\mathbf{c}}}(\mathbf{x}) = 0$.

The Fig. 5 shows the result of evaluate each pixel $\mathbf{x}$ of Fig. 1 with the function $f_{\hat{\mathbf{c}}}(\mathbf{x})$. Thus, the resulting image have values between 0 and 1, and was colorized with a color map with colors between blue and red to show results $0 < y < 1$.

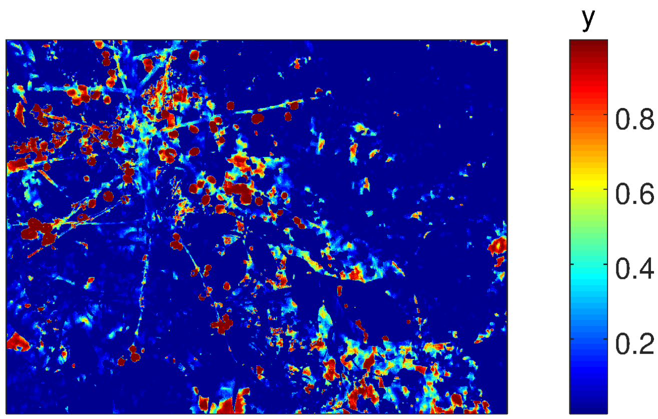

Figure 5: *Result of function* $f_{\hat{\mathbf{c}}}(\mathbf{x})$.