Meat Quality Control

Computer Vision, UPC

Xavier Martín Ballesteros and Adrià Cabeza Sant'Anna

March 19, 2019

Abstract

Our goal is to detect the percentage of fat in chops using images. We discuss about the different binarization methods we have used, such as *basic binarization*, *p-tile thresholding*, *optimal thresholding* and *kapur method*. Finally, we have analysed our results and decided which method is the best one.

Contents

1 Introduction

The objective of this assignment is to detect the percentage of fat in chops using images. To do it, we have used several different threshold techniques. Those methods consists in setting a constant value called *threshold* (T) and separe the pixels depending its value ($f(i,j)$):

$$g(i,j) = 1 \; if \; f(i,j) \geq T$$

$$g(i,j) = 0 \; if \; f(i,j) \leq T$$

In the following sections we will introduce different methods to find the threshold value and compared its results.

2 Binarization

2.1 Basic Binarization

This is the first method we tried and also the fastest and easiest one.
Our approach was to print the histogram of the picture and check where we could set the best value for the *threshold* in order to separe the fat. The histograms that we got were bimodal so we coud set an acceptable value just by looking at it. We believe that this distribution of pixel values is formed because we are working with grayscale pictures of chops where we can appreciate clearly a lighter tone for the fat and darker tones for the rest.

2.2 P-tile Method

This method uses knowledge about the area size of the desired object. It assumes the desired part of the image are brighter that the background and occupy a fixed percentage of the picture area. The *threshold* is defined as the grey level that mostly corresponds to mapping at least that fixed percentage into the object.

2.3 Otsu Method

Otsu method is one of the most successful methods for image thresholding. It is very effective for images which are bimodal. However, it may not be accurate for non-bimodal images. In this method we search among all possible thresholds to find the one that minimizes the weighted within-class variance, which is the same than maximizing inter-class variance.

The probability of a pixel to have the gray level i is

$$p_i = \frac{n_i}{N}, \qquad p_i \geqslant 0, \sum_{i=0}^{L-1} p_i = 1$$

where L is the number of different gray levels of a given picture, n_i is the number of pixels of gray level i and N is the total number of pixels. Then, the probability of being in class *Background* or *Foregound* is

$$\omega_B = \sum_{i=0}^{t} p_i \qquad \omega_F = \sum_{i=t+1}^{L} p_i$$

where t is the threshold value. The mean values of the classes are

$$\mu_B = \sum_{i=0}^{t} \frac{i * p_i}{n_B} \qquad \mu_F = \sum_{i=t+1}^{L} \frac{i * p_i}{n_F}$$

and the class variances are given by

$$\sigma_B^2 = \sum_{i=0}^{t} \frac{(i-\mu_B)^2 * n_i}{n_B} \qquad \sigma_F^2 = \sum_{i=t+1}^{L} \frac{(i-\mu_F)^2 * n_i}{n_F}$$

Hence, the within-class variance is computed as

$$\sigma_W^2 = \omega_B * \sigma_B^2 + \omega_F * \sigma_F^2$$

Nevertheless, we can transform this minimization problem into a maximization problem by computing the inter-class variance, which is faster to compute

$$\sigma_I^2 = \omega_B * \omega_F * (\mu_F - \mu_B)^2$$

2.4 Optimal thresholding by Clustering Method

Optimal thresholding methods select the threshold based on the minimization of a criterion function. Otsu, for example, tries to minimize the intra-class variance. This method tries to minimize the probability between the maxima of 2 distributions.

We used Ridler Calvard Method which segments the image into two clusters (*Background* and *Foreground*) using the initial theshold value. The algorithm starts assuming that the four corners are the only pixels in the *Background* and the rest is the *Foregound*. It also selects an initial estimate for the threshold t_1. At each step we compute the mean gray-level of the two clusters

$$\mu_B^{t_n} = \frac{\sum_{i=0}^{t_n} i * p_i}{n_B} \qquad \mu_F^{t_n} = \frac{\sum_{i=t_n+1}^{L} i * p_i}{n_F}$$

Then, the new threshold is compute as

$$t_{n+1} = \frac{\mu_B + \mu_F}{2}$$

The algorithm repeats these steps until t stabilizes, which means that $|t_n - t_{n+1}| < \varepsilon$.

2.5 Kapur, Sahoo and Wong Method

In this method two probability distributions are derived from the original gray level distribution of the image(i.e. object distribution and background distribution):

$$\frac{p_0}{P_t}, \frac{p_1}{P_t}, ..., \frac{p_t}{P_t}$$

and

$$\frac{p_{t+1}}{1-P_t}, \frac{p_{t+2}}{1-P_t}, ..., \frac{p_{l-1}}{1-P_t}$$

where t is the value of the threshold and $P_t = \sum_{i=0}^{t} p_i$. Define

$$H_b(t) = -\sum_{i=0}^{t} \frac{p_i}{P_t} log_e \left(\frac{p_i}{P_t} \right)$$

$$H_b(t) = -\sum_{i=t+1}^{l-1} \frac{p_i}{1-p_i} log_e \left(\frac{p_i}{1-P_t} \right)$$

Then the optimal threshold t^* is defined as the grey level which maximizes $H_b(t) + H_w(t)$, that is,

$$t^* = \ ArgMax \left(H_b(t) + H_w(t) \right)$$

3 Results

In the following section we will see the results of our methods being applied to the dataset given. A total of 14 chops images will be analysed to check how many fat do they have. Finally we will discuss which method was to best one and we will take our conclusions

Picture of the chop	**Otsu**	**Kapur**	**Optimal Thresholding**	**P-tile**
F1011flb.bmp	29.1682	31.3817	29.2252	39.5698
F1019flb.bmp	33.2429	36.2029	FALLA A SACO	39.5983
F1031flb.bmp	38.1320	44.5639	38.4935	39.2967
F1051flb.bmp	33.8235	36.8900	35.6833	39.5664
F1053flb.bmp	35.6418	FALLA A SACO	35.6833	37.8899
F1059flb.bmp	28.4953	FALLA	FALLA	FALLA
F1064flb.bmp	26.4423	26.0063	27.1537	FALLA A SACO
F1079f1b.bmp	31.2544	FALLA	FALLA	FALLA
F1083f1b.bmp	27.5718	FALLA	27.7740	FALLA
F1096f1b.bmp	28.8945	FALLA	FALLA	FALLA
F1097flb.bmp	29.2960	FALLA	FALLA	FALLA
F1101flb.bmp	33.2675	FALLA	34.7553	FALLA
F1102f1b.bmp	27.6687	FALLA	FALLA	FALLA
F1103f1b.bmp	34.0777	FALLA	37.7081	39.6781

Table 1: Results obtained using different methods of binarization

References

[1] C. Henden (2004). *Exercise in Computer Vision. A Comparison of Thresholding Methods.* [online] NTNU. Available at: http://www.pvv.org/ perchrh/papers/datasyn/paper2/report.pdf [Accessed 14 Mar. 2019].

[2] D. Martín Carabias (2012). *Analysis of image Thresholding Methods for application to augmented reality enviornments.* [online] UCM. Available at: https://eprints.ucm.es/16932/1/Tesis_Master_Daniel_Martin_Carabias.pdf [Accessed 14 Mar. 2019].

[3] P. K. Sahoo, S. Soltani, K.C. Wong and Y.C. Chen (1988). *A survey of Thresholding Techniques*. University of Waterloo, Waterloo, Canada [Accessed 16 Mar. 2019].

[4] N. Otsu, *A Threshold Selection Method from Gray-Level Histograms*, in IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, Jan. 1979. [online] Available at http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4310076&is number=4310064 [Acessed 17 Mar. 2019].

[5] Dr. Andrew Greensted (2010), *Otsu Thresholding*. [online]. Available at http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html [Accessed 17 Mar. 2019].

[6] Senthilkumaran, N. & Sivapriya, M. (2017), *Riddler's Thresholding Algorithm for DNA Image Using ISODATA Modified Algorithm* Journal

of Information Technology, Vol.3, No.2, pp.41-48. [online] Available at: http://www.ijitjournal.org/volume-3/issue-2/IJIT-V3I2P9.pdf [Accessed 17 Mar. 2019].