

UNIVERSITAT POLITÈCNICA DE
CATALUNYA

Computer Vision

Meat Quality Control

Adrià Cabeza Sant'Anna and Xavier Martin Ballesteros



March 21, 2019

Contents

1	Introduction	1
2	Binarization	1
2.1	Basic Binarization	1
2.2	P-tile Method	2
2.3	Otsu Method	2
2.4	Optimal thresholding Method	3
2.5	Kapur, Sahoo and Wong Method	3
3	Results	4
3.1	Table of results	4
4	Annex 1: Images	5

1 Introduction

The aim of this assignment is to detect the percentage of fat in the images with chops using binarization.

We have first computed a mask that only selects the chop pixels. To do it, we have used Otsu thresholding method. Once we have got the part of the image we wanted, we have used several techniques to calculate the threshold. Binarize the image with this threshold leaves in blank the fat pixels, which is what we wanted. Then, dividing the pixels of fat by the chop pixels, we got the percentage of fat in a specific chop. Moreover, we have also computed manually both thresholds (chop and fat) using the histogram obtained.

The methods consists in setting a constant value called *threshold* (T) and split up the pixels depending on its value ($f(i, j)$):

$$g(i, j) = 1 \text{ if } f(i, j) \geq T$$

$$g(i, j) = 0 \text{ if } f(i, j) \leq T$$

In the following sections we will introduce different methods to find the threshold value and compare 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 split up the fat. The histograms that we got were bimodal so we could 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 that the desired part of the image is brighter than the rest and occupy a fixed percentage of the picture area. The *threshold* is defined as the grey level where the percentage of pixels in the background is approximately the fixed percentage we wanted.

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

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

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 *Foreground* is

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

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

$$\mu_B = \sum_{i=0}^t \frac{i \times p_i}{n_B} \quad \mu_F = \sum_{i=t+1}^{L-1} \frac{i \times p_i}{n_F}$$

and the class variances are given by

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

Hence, the within-class variance is computed as

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

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

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

2.4 Optimal thresholding 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 threshold value. The algorithm starts assuming that the four corners are the only pixels in the *Background* and the rest is the *Foreground*. It also selects an initial estimate for the threshold t_1 . At each step we compute the mean gray-level of the two clusters

$$\mu_B^{t_n} = \frac{\sum_{i=0}^{t_n} i}{n_B} \quad \mu_F^{t_n} = \frac{\sum_{i=t_n+1}^{L-1} i}{n_F}$$

Then, the new threshold is computed as

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

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

2.5 Kapur, Sahoo and Wong Method

Kapur, Sahoo and Wong Method is the method we have implemented from the entropic family methods. In this method we work with the gray level histogram to obtain the optimal threshold applying information theory. The way it works is the following: two probability distributions are derived from the original gray level distribution of the image (i.e. object distribution and background distribution):

$$\begin{aligned} & \frac{p_0}{P_t}, \frac{p_1}{P_t}, \dots, \frac{p_t}{P_t} \\ & \text{and} \\ & \frac{p_{t+1}}{1 - P_t}, \frac{p_{t+2}}{1 - P_t}, \dots, \frac{p_{l-1}}{1 - P_t} \end{aligned}$$

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

$$\begin{aligned} H_b(t) &= - \sum_{i=0}^t \frac{p_i}{P_t} \log_e \left(\frac{p_i}{P_t} \right) \\ H_w(t) &= - \sum_{i=t+1}^{l-1} \frac{p_i}{1 - P_t} \log_e \left(\frac{p_i}{1 - P_t} \right) \end{aligned}$$

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

$$t^* = \text{ArgMax} (H_b(t) + H_w(t))$$

3 Results

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

3.1 Table of results

Picture of the chop	Manual	Otsu	Optimal	Kapur	P-tile
'F1011flb.bmp'	27.9423	29.1830	29.4718	32.0885	33.5235
'F1019flb.bmp'	22.7007	33.2420	33.7937	35.5768	33.7937
'F1031flb.bmp'	35.7952	38.2012	38.5656	43.6927	34.3695
'F1051flb.bmp'	22.2951	33.7487	35.5875	28.6444	33.7487
'F1053flb.bmp'	31.4224	35.5413	35.9936	39.9009	34.3633
'F1059flb.bmp'	21.7666	28.4536	28.7125	30.1826	34.8540
'F1064flb.bmp'	17.6764	26.3488	27.0577	22.7853	34.2446
'F1079flb.bmp'	13.6092	31.2078	33.6154	25.0521	33.8857
'F1083flb.bmp'	18.2551	27.6612	27.8640	22.2453	33.9433
'F1096flb.bmp'	22.9326	28.8762	28.8762	25.2340	34.9686
'F1097flb.bmp'	23.7057	29.2290	29.7544	29.7544	34.9986
'F1101flb.bmp'	24.4286	33.4608	34.9602	40.0134	34.9602
'F1102flb.bmp'	25.9011	27.8115	28.6156	30.8599	34.4794
'F1103flb.bmp'	19.2648	34.0299	37.7080	27.8738	34.1712

Table 1: Results of the fat percentage obtained using different methods of binarization

Even though we already know that *P-tile* does not make sense for this assignment we have implemented it because we wanted to expand our knowledge about different algorithms related to find the optimal threshold automatically.

After experimenting with several values for the percentage we have set it to 70% which gave us the best results.

Although we have tried different thresholding methods, we know that the percentages are not correct due to different levels of brightness, which creates false positives and/or false negatives, and placement of chops, which causes that the ruler, that is white, is being detected as fat.

4 Annex 1: Images

In this annex we can see an example of the binarized images outputted using the different thresholding methods we have implemented.

Binarització per Entropia (Kapur, Sahoo, and Wong) del greix (F1053flb.bmp)



Figure 1: Kapur

Binarització per Optimal Thresholding (Ridler Calvard) del greix (F1053flb.bmp)



Figure 2: Optimal

Binarització del greix manual (F1053flb.bmp)



Figure 3: Manual

Binarització per Otsu del greix (F1053flb.bmp)



Figure 4: Otsu

Binarització per P-Tile del greix (F1053flb.bmp)



Figure 5: Ptile

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 environments*. [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&isnumber=4310064> [Accessed 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].