Automatic marbling prediction of sliced dry-cured ham using image segmentation, texture analysis and regression

Eva Cernadas[a], Manuel Fernández-Delgado[a], Elena Fulladosa[b], Israel Muñoz[b]

[a]*Centro Singular de Investigación en Tecnoloxías Intelixentes da USC (CiTIUS), Universidade de Santiago de Compostela, Rúa Xenaro de la Fuente Domínguez, Santiago de Compostela, 15782, Spain*
[b]*IRTA-TA, Food processing and Engineering, Finca Camps i Armet, Monells, Girona, 17121, Spain*

Abstract

Dry-cured ham is a traditional Mediterranean meat product consumed throughout the world. This product is very variable in terms of composition and quality. Consumer's acceptability of this product is influenced by different factors, in particular, visual intramuscular fat and its distribution across the slice, also known as marbling. On-line marbling assessment is of great interest for the industry for classification purposes. However, until now this assessment has been traditionally carried out by panels of experts and this methodology cannot be implement in industry. We propose a complete automatic system to predict marbling degree of dry-cured ham slices, which combines: 1) the color texture features of regions of interest (ROIs) extracted automatically for each muscle; and 2) machine learning models to predict the marbling. For the ROIs extraction algorithm more than the 90% of pixels of the ROI fall into the true muscle. The proposed system achieves a correlation

Email addresses: eva.cernadas@usc.es (Eva Cernadas), manuel.fernandez.delgado@usc.es (Manuel Fernández-Delgado), elena.fulladosa@irta.cat (Elena Fulladosa), israel.munoz@irta.cat (Israel Muñoz)

of 0.92 using the support vector regression and a set of color texture features including statistics of each channel of RGB color image and Haralick's coeficients of its grey-level version. The mean absolute error was 0.46, which is lower than the standard desviation (0.5) of the marbling scores evaluated by experts. This high accuracy in the marbling prediction for sliced dry-cured ham would allow to deploy its application in the dry-cured ham industry.

Keywords: Dry-cured ham, intramuscular fat, marbling, support vector regression, texture analysis, image segmentation

1. Introduction

Dry-cured ham is a traditional meat product of many Mediterranean countries that is widely consumed throughout the world, being its flavour and texture characteristics highly appreciated by consumers. There are many factors affecting the final characteristics of dry-cured ham, such as processing conditions and raw material characteristics, i.e. fat content (Coll-Brasas et al., 2021). In sliced dry-cured ham, visual intramuscular fat, subcutaneous fat thickness and color are the parameters most used by the consumers for the product evaluation, therefore affecting consumer's acceptability and purchase decision (Lorido et al., 2019). Although several non-invasive technologies can be used to categorize entire hams according to its fat content (de Prados et al., 2015), these technologies cannot be used to predict intramuscular fat (IMF) in sliced products because of its variability between the muscles of a ham.

Eating quality in meat has been associated to the fat distribution rather than to the total IMF. Distribution of IMF is usually known as marbling, because of its appearance similar to marble (Cernadas et al., 2002). One of the most important challenges for producers is the heterogeneity of the marbling in slices, that can vary significantly among ham pieces and even within the same piece. Classification of slices of dry-cured ham according to the marbling degree is of special interest for the food industry. Producers would be able to segment the market, offering products tailored to consumer's needs and increasing the value of their production.

Marbling ranking in different meats and meat products has been performed by panels of trained experts or relying on standards consisting of pictures depicting a scale of marbling (from 0.0 to 10.0), as it is the case for the National Pork Producers standards (Moines, 1999). In the case of dry-cured ham, a marbling ranking has been developed but it is not still published. However, marbling evaluations by experts are costly and are not feasible for the ham industry. Computer image analysis might be a solution because it is a fast and non-destructive technology, and it is a replicable and repetitive method that has been successfully applied to the assessment of multiple food characteristics: fish (Dutta et al., 2016), cheese (Dias et al., 2021) or bread (Srivastava et al., 2015). The scientific literature includes several studies that apply computer image analysis to determine IMF and marbling in different meat products. Combining magnetic resonance imaging (MRI) and computer vision techniques, the works (Cernadas et al., 2005;

Ávila et al., 2019) predicted marbling in the biceps muscle of dry-cured hams and loins, but MRI is a technology that is expensive to install in meat industries.

For the segmentation of IMF in meat and meat products, several techniques have been applied: K-means clustering in beef Longissimus dorsi muscle (Jackman et al., 2009), automatic thresholding (Liu et al., 2018) and the Otsu method (Uttaro et al., 2021) in pork loin, multi-scale line detection (Cernadas et al., 2002), gradient-based techniques (Santos-Garcés et al., 2014) and convolutional neural networks (Muñoz et al., 2019) for dry cured ham. In general, these segmentation algorithms perform quite well, high correlation values or low classification errors are obtained, depending on the aim of the study. For classification of marbling, the following techniques have been applied: chemical pre-treatments and line detection algorithms (Faucitano et al., 2004), line detection algorithm (Liu et al., 2012) and (Huang et al., 2013) in pork meat, neural networks (Muñoz et al., 2015) and decision trees in hyperspectral images of dry cured ham (Velásquez et al., 2017). In general, the results of marbling classification are quite satisfactory with low prediction errors and at least 90% of the samples correctly classified. However, the evaluation of IMF and marbling in dry-cured ham slices is still a challenge. A wide range of variation in the color of lean and fat tissues can be observed in slices across hams, which poses a challenge for image segmentation and the evaluation of marbling. These differences are explained by the different levels of drying of the hams, and the presence of precipitates

such as phosphates of tyrosine crystals, with a white color similar to that of the fat.

This paper proposes a prototype to automatically predict marbling of the principal muscles from a ham slice using image segmentation, texture analysis and regression models. Specifically, we define an algorithm that automatically extracts squared regions inside the main muscles. Then, color texture features are computed for each region, which are the inputs to a regression model that predicts the marbling score for each muscle. The paper is organised as follows. Section 2 describes the materials used to obtain the ham slice, to develop the sensorial analysis on the ham muscles and to annotate the contour of ham muscles. Section 3 describes the algorithm used to extract automatically the square ROIs representing each ham muscle, and briefly explains the color texture features extraction techniques and the regression models. Section 4 describes the experimental setup and statistical evaluation measures used. Section 5 presents and discusses the results. Finally, section 6 summarises the main conclusions and proposals of future work.

2. Materials

This section describes the material used to obtain the data in this research: the system used to capture images of ham slice (subsection 2.1), the traditional procedures to determine the marbling of a ham muscle (subsection 2.2), and the process to draw the outline of each muscle on the ham slice

(subsection 2.3).

2.1. Image acquisition

High quality images were acquired with a calibrated digital camera Canon EOS 50D (15.1 megapixels) and an objective Canon EF-S 18–200 mm f/3,5–5,6 IS. The camera was mounted in a black closet ($1.06 \times 1.06 \times 2.50$ m^3) with 8 equidistant halogen lights Solux Q50MR16 CG/47/36°12 V/50 W/4700 K (Eiko Ltd., Shawnee, Kansas, U.S.A.) to ensure a correct lighting. White balance was carried out with a white card (Lastolite). The camera was connected to a computer to store the images. Slices were placed 30 cm below the camera on a uniform black surface. Photos of both sides of the dry-cured ham slices were taken. All the images were taken during the same session. The white balance of the images was carried out with Capture One PRO 5.0 software (Phase One A/S Inc., Frederiksberg, Denmark) and RGB images of 667×1000 pixels with 16 bits color were obtained, with one pixel corresponding to 0.3968 mm^2. For the evaluation of marbling, the computer screen was calibrated so that the colors of the images were as close as possible to the colors of the samples (NEC Multisync LCD 2690 WUXI2). Figure 1 shows a scheme of the image acquisition system used.

2.2. Marbling evaluation

Sensory analysis (marbling evaluation) of the samples was carried out by six trained panelists (ISO 8586-2: 2012) and consisted of a visual assessment of the marbling of the most representative muscles: *Biceps femoris*

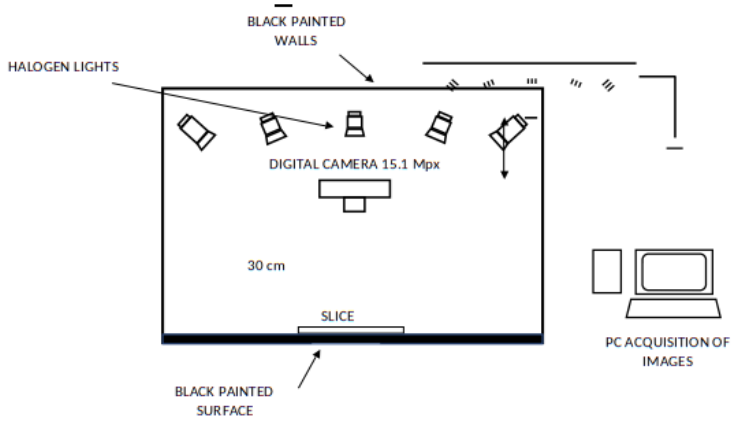


Figure 1: Scheme of the image acquisition system.

(BF), *Semimembranosus* (SM) and *Semitendinosus* (ST) of a dry-cured ham (Bermúdez et al., 2014). Marbling was scored by consensus (in our case three panelists) by means of scoring scale from 0.5 (minimum marbling) to 10 (maximum marbling) at intervals of 0.5. When scoring marbling, the panelists considered the total amount and the distribution of the fat streaks. Marbling evaluation was done in triplicate by the panelists. The standard deviation of the panelists among trials was determined at 0.5 points.

Muscle	#images	Min.	Max.	Avg.	Dev.
Biceps femoris	337	1	7	3.1	1.0
Semimembranosus	322	0.5	6	2.0	1.0
Semitendinosus	55	4	9	6.2	1.2

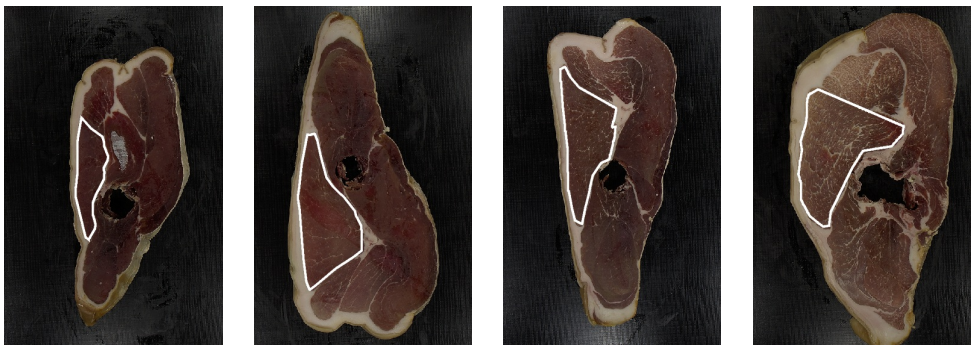
Table 1: Number of images, minimum, maximum, average and standard desviation of the marbling values for the different ham muscles used in this experimentation.

A collection of commercial dry-cured hams were obtained from different ham producers with crosses from different pig breeds (Large White, Lan-

drace, Duroc and Iberian) and having a wide range of marbling. A 2 cm thick slice containing muscles BF, SM and ST was obtained at 10 cm from the aitch bone in the distal direction (at the widest part of the ham) and packed into plastic bags of polyamide/polyethylene (oxygen permeability of 50 $cm^3/m^2/24$ h at 23°C and water permeability of 2.6 $g/m^2/24$ h at 23°C and 85% RH, Sacoliva© S.L., Spain). The image dataset is composed of 714 images obtained in the following way: photos were obtained from 180 commercial dry-cured hams, 2 slices/ham (at different points in the ham, obtaining slices quite different one from another) and 2 muscles for each slice, giving a total of 180 hams $\times$ 2 slices $\times$ 2 muscles = 720 photos. Six of these photos were not included in the evaluation due to defects on the surface such as cuts and phosphate crystals. For each image, it was only provided the measure of marbling for one muscle with values between 0.5 and 9 with the distribution shown in table 1. Figure 2 shows examples of different marbling scores for biceps muscle.

2.3. Muscle annotation

To develop the first experiment, the experts delineated the contour of the muscle for which the marbling was estimated by sensorial analysis. To draw the contours for all images, they used a home-made software, as it can be seen in figure 3.



a b c d

Figure 2: Examples of marbling scores for biceps muscle: a) slice 8372 with score 1; b) slice 8485 with score 1; c) slice 8274 with score 3; and d) slice 8424 with score 6.5.

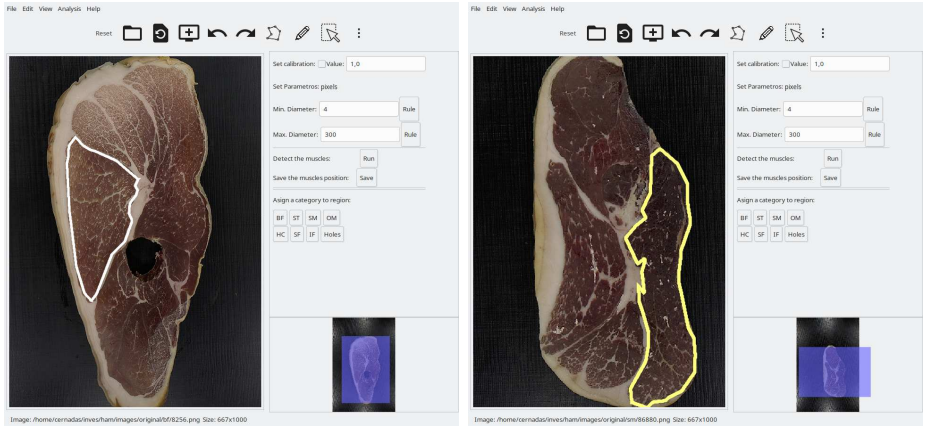


Figure 3: Examples of the contours of biceps (left panel) and semimembranosus (right panel) muscles overlapped to images of ham slice.

3. Methods

The system proposed to predict the marbling from dry-cured ham slices, shown in figure 4, encloses the following stages: 1) the image acquisition system already described in section 2.1; 2) the automatic extraction of the ROIs in the ham slice; 3) the computation of features from the ROI extracted;

158 and 4) the regression model to predict the marbling score of each muscle in
159 the ham slice. The subsections 3.1, 3.2 and 3.3 describe the stages 2, 3 and
160 4, respectively.

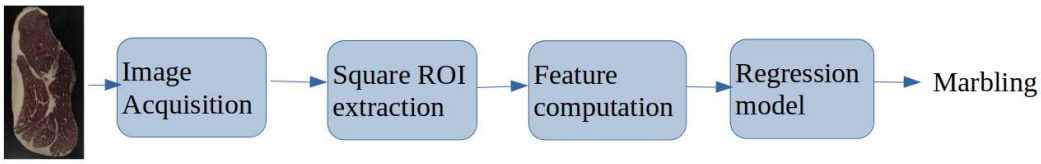


Figure 4: Stages of the method to predict the marbling from dry-cured ham slices.

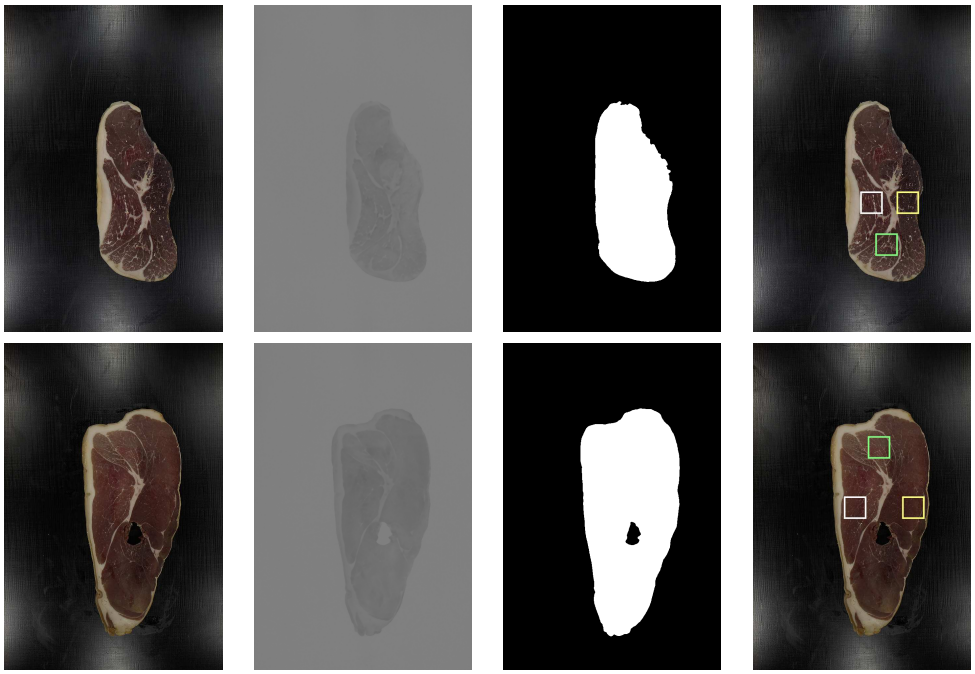


Figure 5: Examples of the extraction of square ROIs from the ham slice for all muscles: a) original images; b) channel b of Lab color space after smoothing; c) binary mask with the ham slice and bone hole and d) the extracted ROIs for each muscle, white for biceps, yellow for semimembranosus and green for semitendinosus.

3.1. Automatic extraction of ROIs

The ham slice images are processed to automatically extract square ROIs from the biceps femoris, semimembranosus and semitendinosus muscles. These extracted ROIs will be used in the third experiment. In this process, we take into account the anatomical information about the distribution of the muscles and subcutaneal/intermuscular fat within the ham. As it can be seen in figure 3, some slices present a hole in the slice (left panel), due to the slice is cut by the ham bone, and others not (right panel). As well, the biceps muscle can be in the right or left side of the image. The algorithm to extract the square ROIs encloses the following steps: 1) extract the ham slice from the image; 2) check if the slice has the bone hole; 3) if there is not a hole in the slice, find the biggest intermuscular fat region in the slice; 4) in both previous cases (step 2 or 3), a reference position is calculated to know the slice orientation, which allows to know if the muscles are upper/bottom or if the BF muscle is on left/right side of the slice; and 5) extract a square region for each muscle. In our case, we use a ROI with length $s = 64$ pixels.

To extract the ham slice, the original RGB image $I(x, y)$, with $x = 1, \ldots, N$, and $y = 1, \ldots, M$, of dimensions $N \times M$, is transformed to the Lab color space, because it is more robust to illuminance variance than the RGB space (Cernadas et al., 2017). Let $I_b(x, y)$ be the b channel of the ham slice after smoothing with a mask (we use a mask of 5 pixels) in order to attenuate the small fat features and noise (column b in figure 5). Let h_b be the histogram of image $I_b(x, y)$. The maximum value H_b of h_b is chosen to

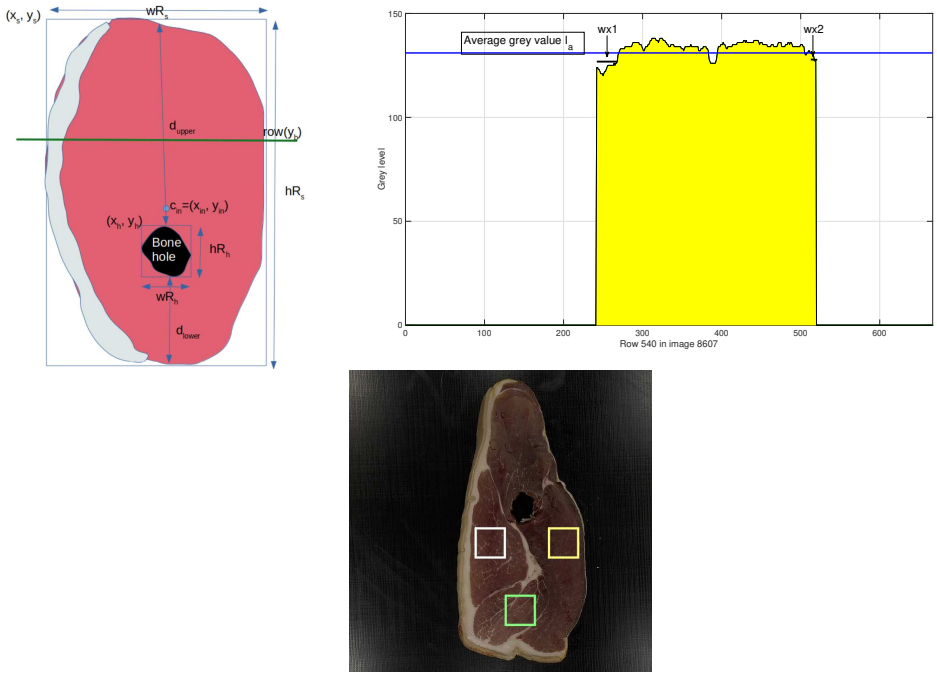


Figure 6: Extraction of the square ROIs of each muscle. Upper left panel: scheme of a ham slice. Upper right panel: row 540 of the transformed version I_a of the ham slice no. 8607. Lower panel: the ham slice with the extrated ROIs overlapped: *biceps* (white), *semimembranosus* (yellow) and *semitendinosus* (green).

calculate the area of ham slice. The $I_b(x, y)$ image is thresholded to calculate the binary image $B(x, y)$ using the following expresion:

$$B(x,y) = \begin{cases} 0 & |I_b(x,y) - H_b| \leq 5 \\ 255 & \text{otherwise} \end{cases} \quad (1)$$

After thresholding, we apply morphological operators to the binary image B in order to fill small holes. First, the biggest region of B is extracted and it

Algorithm 1: Automatic extraction of square ROI images for each muscle from a ham slice.

Algorithm: $[I_{BF}, I_{SM}, I_{ST}]$=ExtractSquaredROI(I, s)

Data: I: original RGB image of ham slice; s: size of ROI
Result: I_{BF}, I_{SM}, I_{ST}: square ROI images for *biceps*, *semimembranosus* and *semitendinosus* muscles

$I_b \leftarrow b$ channel of Lab image smoothed by box filter
$h_b \leftarrow$ histogram of I_b
$H_b \leftarrow$ maximum of h_b
$B \leftarrow$ binary image using eq. 1 and morphological processing
$B_m \leftarrow$ image mask with ham slice outline and hole if exist
$R_s \leftarrow (x_s, y_s, wR_s, hR_s)$ rectangle enclosing ham slice
$I_a \leftarrow a$ channel of Lab image smoothed and masked by B_m
$\mu_a \leftarrow$ average value of I_a inside B_m; `offset`$\leftarrow$10
if `existsHole(`I_b`)` **then**
    $R_h \leftarrow (x_h, y_h, wR_h, hR_h)$ rectangle enclosing hole
    $d_{upper} \leftarrow y_s - y_h$; $d_{lower} \leftarrow y_s + hR_s - (y_h + hR_h)$
    if $d_{upper} > d_{lower}$ **then**
        $y_c \leftarrow y_s - (y_s - y_h)/4$
        $y_e \leftarrow y_h - s/2 -$ offset
    else
        $y_c \leftarrow y_s + hR_s - s - (y_s + hR_s - y_h - hR_h)/4$
        $y_e \leftarrow y_h + hR_h + s/2 +$ offset
    end
    $x_c \leftarrow$ `middlePointX(row(`$y_c + s/2$`))`
    $I_{ST} \leftarrow$ `extST(`$x_c - s/2$`,`y_c`,`s`)`
    $[I_{BF}, I_{SM}] \leftarrow$ `extBSM(`y_e`)`
else
    $T_2 \leftarrow$ second Otsu's threshold of I_a
    $(x_{in}, y_{in}) \leftarrow$ centroid of the largest inner region inside I_a after thresholding using T_2
    $I_{ST} \leftarrow$ `extSTC(`x_{in}, y_{in}, s`)`
    $[I_{BF}, I_{SM}] \leftarrow$ `extBSM(`y_{in}`)`
end

is associated with the contour of ham slice. Next, the algorithm searches for a large black region inside this contour. If this region is found, it is associated to the bone hole. The contours of ham slice and bone hole are used to create a binary mask image $B_m(x, y)$=0 (black color in column c of figure 5) when (x, y) is inside the contour and outside the bone, and $B_m(x, y)$=1 (white color) when (x, y) is outside the contour or inside the bone. This process is shown in the columns a, b and c of the figure 5. The process to extract the ROI for each muscle, denoted as I_{BF}, I_{SM} and I_{ST} for *biceps femoris*, *semimembranosus* and *semitendinosus*, respectively, is drawn in figure 6 and summarized by algorithm 1. Let R_s be the rectangle enclosing the ham slice, which is defined by the top left vertex (x_s, y_s) and by its width (wR_s) and height (hR_s). Let I_a be the a channel of Lab image of the ham slice masked (multiplied) by B_m. Two cases can be considered:

a) When the ham slice has visible bone hole, the algorithm uses as reference its enclosing rectangle R_h, with initial coordinates (x_h, y_h) and width and height wR_h and hR_h, respectively.

b) When the ham slice has not a visible bone hole, the algorithm searches for the largest intermuscular fat region and calculates its centroid. In order to discard the background from the image I_a, we selected a threshold applying the Otsu's method (Otsu, 1979) with three thresholds $\{T_i\}_{i=1}^{3}$, that correspond to different types of materials in the image (background, subcutaneal fat and muscle). Experimentally, we checked

that threshold T_2 ensures an accurate segmentation of background from the remaining materials. In the binary image generated by thresholding I_a with T_2, the inner biggest region is selected as representing the intermuscular fat region, whose centroid is (x_{in}, y_{in}).

These reference points (bone hole or centroid of intermuscular fat) allow to select the height where the different muscles should be extracted. In order to extract the BF and SM muscles, the algorithm finds out whether the biceps is on the left or right side of the ham slice. The position in the horizontal axis for extracting the BF and SM muscles is determined analysing a specific row in the image I_a. This row is smoothed in order to attenuate the random noise and it is denoted as $\texttt{row}(y_b) = I_a(x, y_b)$, $x = 1, \ldots, M$ (see upper left panel of figure 6). In order to locate the BF muscle, we estimate the pixels representing the subcutaneal fat (close to the *biceps* muscle) along the $\texttt{row}(y_b)$ counting the number of values. Specifically, the procedure is as follows:

1. Let k_1 be the first value of x where $I_a(x, y_b) > 0$ coming from left to right (see the upper right panel of figure 6). Let $wx1$ be the number of values of x where $I_a(x, y_b) < \mu_a$, for $x = k_1, k_1 + 1, \ldots, M$ (i.e. going from left to right side), where μ_a is the average value of I_a inside B_m.

2. Let k_2 be the first value of x where $I_a(x, y_b) > 0$ coming from right to left, and $wx2$ the number of values that $I_a(x, y_b) < \mu_a$, for $x = k_2, k_2 - 1, \ldots, 2, 1$ (i.e. going from right to left side).

If $wx1 > wx2$ the BF muscle is on the left side of ham slice. Otherwise, it is on the right side. This process is performed by the `extBSM(y)` function in the algorithm 1, where y represents the row to be analysed in the image. So the square ROIs for muscles BF and SM, of size s, are extracted at positions $(k_1 + wx1 + s/2 + \text{offset}, y)$ and $(k_2 - wx2 - s/2 - \text{offset}, y)$ (we use an offset of 10 pixels in order to avoid defects in the contour of ham slice). The regions extracted for each muscle are shown overlapped to the ham slice in the column d of the figure 5. When there is bone hole, the y coordinate for extracting muscles are determined calculating the distances $d_{upper} = y_s - y_h$ and $d_{lower} = y_s + hR_s - (y_h + hR_h)$. If $d_{upper} > d_{lower}$, the muscles are above the bone hole, otherwise the muscles are below the bone hole. The `middlePointX(y)` function returns the middle point of the ham slice for the image row y. The `extST(x,y,s)` function extracts a square ROI of size s for the ST muscle from the original RGB image in the point (x, y) for ham slices with visible hole. The `extSTC(x,y,s)` function extracts a square ROI of the ST muscle for ham slices without visible hole. Let x_{c1}=`middlePointX(row(`$y_{in} + s/2$`))` and x_{c2}=`middlePointX(row(`$y_{in} - s/2$`))` be two middle points in the ham slice in the horizontal axis near the centroid of intermulcular fat. Let r_1 and r_2 be two ROIs of size s, extracted from the image I_a in the points $(x_{c1} - s/2, y_s - (y_s - y_{in})/6)$ and $(x_{c2} - s/2, y_s + wR_s - s - (y_s + wR_s - y_{in})/3)$ respectively. To select which ROI corresponds to the ST muscle, the mean value of both ROIs, μ_{r1} and μ_{r2}, are calculated and the ST muscle is the ROI with the highest mean value, which corresponds with the ROI containing

more fat.

3.2. Color texture features

Texture analysis has been employed in previous works to predict different qualities or attibutes of meat products from MRI images (Cernadas et al., 2005; Ávila et al., 2019), which are grey level images. Nevertheless, it is known that the color is also a very important characteristic in other computer vision applications (González-Rufino et al., 2013; Cernadas et al., 2017). Color texture analysis can be tackled from different paradigms: simple color features, grey level texture features and integrative color texture analysis. A recent work (Cernadas et al., 2017) concluded that parallel approaches, that concatenate the two former, are superior than analysing directly the color texture with integrative approaches.

There are many methods to extract only the color in a strict sense (Cernadas et al., 2017). In this work we use two color spaces, RGB and Lab, where the chromatic channels are a and b. We use first-order features of three types, each with 2 feature vectors:

1. Only mean value for each chromatic channel, denoted as CM, that stands for "color mean": 1) **CMRGB**, 3 features: mean color of the channels R, G, and B of the muscle; 2) **CMab**, 2 features: mean color of the channels a and b of the Lab image. In both cases, the mean values are only calculated inside the muscle regions.

2. Mean and variance for each chromatic channel, CMV, or color mean

and variance: 1) **CMVRGB**, 6 features: mean and variance of the channels R, G and B of the RGB image. 2) **CMVab**, 4 features: mean and variance of the channels a and b of the Lab image.

3. First order statistics, denoted as FOS, for each chromatic channel, that include mean, variance, skewness, kurtosis and entropy: 1) **FOSRGB**, 15 features: 5 features $\times$ 3 color channels of the RGB image. 2) **FOSab**, 10 features: 5 features $\times$ 2 color channels of the Lab image.

The most popular grey level texture features belong to the families of statistical, such as second-order features and local binary patterns (LBP), and frequencial, including wavelet and Gabor features. The grey level version of the original image is obtained following two alternative approaches: 1) converting a RGB image to grey level image; and 2) using the L channel of a Lab image.

Among the second-order statistics we selected the Haralick coeficients (Haralick et al., 1973), derived from the grey level coocurrence matrix (GLCM). These coeficients describe the probability of finding two pixels with the same value at different scales, or distances, and orientations, or angles. The parameters used normally are: 1) the orientations 0^o, 45^o, 90^o and 135^o; and 2) three scales, with pixel distances of 1, 2, and 3. For each scale, the GLCM matrix is averaged over all orientations, and the contrast, homogeneity, correlation and energy of the matrix is computed. Two feature vectors were considered: 1) **HarRGB**, that includes the four previous features for scales

$\{1, 2, 3\}$ calculated on the grey version of the RGB image, with 4×3=12 features; and 2) **HarLab**, which is similar to **HarRGB** but calculated on the L channel of Lab image. When the input is an irregular region, the features are computed only on pixels included in the region (González-Rufino et al., 2013). The Haralick's coeficients were computed using the `graycomatrix()` function of the Matlab Image Processing Toolbox[1].

The LBP operator is a state-of-art texture analysis approach proposed by Ojala et al. (2002), which describes each pixel comparing its value with the neighboring pixels. For each neighboring pixel, the result will be set to one if its value is higher than the value of central pixel, otherwise the result will be set to zero, developing a binary code for each pixel. We use the uniform LBP, which considers the binary paterns with only two transitions (from 0 to 1 and vice versa). In a circularly symmetric neighbor set of P pixels can occur $P + 1$ uniform binary patterns. The number of "1's" in the binary pattern is the label of the pattern, while the nonuniform patterns are labelled by $P + 1$. This process can be applied to different scales, as in the case of coocurrence image. The histogram of the pattern labels accumulated over the intensity image is employed as texture feature vector. The most common values for these parameters are $\{(P, R) \in (8, 1), (12, 2), (16, 3)\}$, where P is the number of neighbors and R is the distance between the central pixel and the neighbors. We construct the texture feature vectors **LBPRGB** and

[1]https://www.mathworks.com

LBPLab to be applied on the grey version on RGB image and on the L channel of the Lab image, respectively. Both vectors have $42 = (8+2) + (12+2) + (16+2)$ features, because the uniform LBP are $P+2$ features. We use the LBPMatlab[2] implementation provided by the LBP creators.

Discrete wavelet transform (DWT) representation is a theory for multi-dimensional signal decomposition (Walker, 2008; Laine and Fan, 1993) which recursively apply filters to decompose the image into low-pass and high-pass frequency bands. A compact representation for texture analysis can be computed taking the mean and variance of the energy distribution for the transformed coefficients in each sub-band and decomposition level. We compute texture feature vectors calculating the mean and variance of the energy over 3 levels of decomposition and create the feature vectors $\mathbf{WT}_{ijk}$, where: 1) $i = \{Haar, Daub\}$ is the type of filters to use, namelly Haar (2 coeficients) or Daubechies filtering with four coeficients; 2) $j = \{RGB, Lab\}$ is the grey level version of the RGB image or the L channel of the Lab image; and 3) $k = \{LL, All\}$ represents if only the low-low decomposition sub-bands are considered ($6 = 2 \times 3$ features) or all the sub-bands ($24 = 2 \times 3 \times 4$ features). We used the functions `wfilters()` and `dwt2()` of the Matlab Wavelet Toolbox.

Gabor filters are sinusoidal waves modulated by a Gaussian envelope that can be used for texture classification (Randen and Husoy, 1999). The

[2]http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

filters are applied to the images varying their frequency and orientation. Bianconi and Fernández (2007) analysed the influence of these parameters for texture classification. After applying a set of digital Gabor filters $G_{ij}(x, y)$ with $i \in \{1, \ldots, n_F\}$ and $j \in \{1, \ldots, n_O\}$, where n_F and n_O are respectively the number of frequency and orientations, some statistical features are computed over each filtered image. We used the `gabor()` and `imgaborfilt()` functions of the Matlab Image Processing Toolbox considering the wavelengths $[3, 6, 9, 12]$ and orientations $[0, 30, 60, 90, 120, 150]$, recommended by (Bianconi and Fernández, 2007). We compute the feature vectors **GaborRGB** and **GaborLab**, both with 48 features, that include the mean and standard desviation for each filter ($n_F \times n_O = 6 \times 4 = 24$ filters) applied on the grey level version of the RBG image or the L channel of the Lab image, respectively.

3.3. Regression models

We selected for this experimentation several state-of-art regression models of different families that provided good performances in the comparative analysis (Fernández-Delgado et al., 2019). One of them is implemented in the Octave[3] scientific programming language, and the remaining ones in the R[4] statistical computing language. Most regressors in our collection have tunable hyper-parameters, i.e., parameters whose values must be specified

[3] http://www.octave.org
[4] http://www.r-project.org

previously to training, that often have a strong influence on the regressor performance. In these cases, it is a good practice to try several values for each hyper-parameter in a trial-and-error procedure, and to select the value that provides the best performance on a separate data collection. This method is called "grid search" tuning. The following is a list of these regressors, with its tunable hyper-parameters and the list of values tried for each one in the grid search. For the regressors programmed in R, these values were provided by the `getModelInfo()` function of the `caret` R package (Kuhn, 2016).

1. **lm** is the linear regression provided by the `stats` R package, which performs multivariate linear regression and has no tunable hyper-parameter (Bates and Chambers, 1992).

2. **svr**: epsilon-support vector regression with radial basis function kernel, using the `LibSVM` library (Chang and Lin, 2011) through its Octave interface. The regularization hyper-parameter C and $\gamma = 1/2\sigma^2$, where σ is the kernel spread, are tuned with values $\{2^i\}_{-5}^{15}$ and $\{2^i\}_{-10}^{10}$, respectively.

3. **M5**: regression tree (Quinlan, 1992) implemented by the Weka Data Mining Software[5] and accesed from a R program through the `RWeka` package. It has no tunable hyper-parameter.

4. **cubist**: M5 rule-based regressor with corrections based on nearest

[5]http://www.cs.waikato.ac.nz/ml/weka

neighbors in the training set (Quinlan, 1993), implemented by the `Cubist` R package. Its tunable hyper-parameters are the number of neighbors [0,5,9] and the number of committees [1,10,20].

5. **gbm**: generalized boosting regression model (`gbm` R package) with Gaussian distribution. The tunable hyper-parameters are the maximum depth of input interactions, with values [1,2,3,4,5] and the number of trees for prediction, with values from 50 to 250 step 50.

6. **rf**: random forest (Breiman, 2001) ensemble of averaged random regression trees (`randomForest` R package). The number of inputs selected at each tree (`mtry`) is tuned with 10 values between 2 and the number of features.

4. Experimental setup

In order to achieve a fully automatic system, which can operate on-line in the meat industries, we will develop three experiments to compare the computer predictions with the sensorial procedures:

1. **Experiment 1**: prediction of the marbling from irregular regions of the ham muscles that are delineated and annotated by the food technology experts. In this case, the prediction is done using the same information (irregular region) as the experts.

2. **Experiment 2**: prediction of the marbling using square regions extracted from the irregular regions of the experiment 1. This experiment

tests the reduction of performance when a region of the slice smaller than the whole irregular region is used to do the prediction.

3. **Experiment 3**: fully automatic extraction of square regions of interest (ROI) for each ham muscle, and prediction of the marbling using these ROIs. This experiment will test the performance loss when the ROI extraction may be suboptimal, but the method is fully automatic and does not require any food technology expert.

In order to test the performance of the ROI extraction algorithm, we define the overlapping percentage (OP) as:

$$OP_m = 100\frac{NO_m}{R_t} \tag{2}$$

where m may be biceps, semimembranosus or semitendinosus, R_t is the number of pixels of the ROI (in our case s^2=64×64=4,096 pixels) and NO_m is the number of pixels overlapped to the true region annotated by the expert for the muscle m.

To test the performance of regression models in the prediction of marbling, we used the classical K-fold cross-validation methodology, which uses training and test sets. The most popular performance measures are the Pearson's correlation coefficient (R) between the true and predicted marbling, the mean absolute error (average absolute difference between the predicted and true marbling, MAE) and the root mean square error (square root of the mean

squared difference between them, RMSE):

$$R = \frac{\sum_{i=1}^{N}(y_i - \bar{y})(o_i - \bar{o})}{\sqrt{\left(\sum_{i=1}^{N}(y_i - \bar{y})^2\right)\left(\sum_{i=1}^{N}(o_i - \bar{o})^2\right)}} \tag{3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - o_i| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - o_i)^2} \tag{5}$$

where y_i and o_i are the predicted and true values of marbling respectively for ham slice i, $\bar{y}$ and $\bar{o}$ are the mean values of $\{y_i\}_{i=1}^{N}$ and $\{o_i\}_{i=1}^{N}$, respectively, and N is the number of ham slices. The $|R|$ values can be interpreted according to Colton (1974) as: true and predicted values are not correlated at all (0-0.25), bad to moderate correlation (0.25-0.5), moderate to good (0.5-0.75), very good to excellent (0.75-1). In our study, we use $K = 4$ folds or trials, devoting $K - 2 = 2$ folds for training, one for validation and one for test. Since all the folds have the same size, 50%, 25% and 25% of the data are devoted to training, validation and test sets, respectively. In our case, each data corresponds to the image of a ham slice, and it is composed by the texture feature vector extracted from the image (input) and the marbling value (output) corresponding to that slice image. In order to guarantee that training, validation and test sets contain output values distributed across the

whole range of marbling values, the data are sorted by increasing marbling. After sorting, data 1, 2, 3 and 4 are included in folds 1, 2, 3 and 4, respectively. Data 5, 6, 7 and 8 are added to folds 1, 2, 3 and 4, respectively, and so on. Let $\{T_k, V_k, S_k\}_{k=1}^{K}$ be the training, validation and test folds on k-th trial. Trial 1 uses folds 1 and 2 for training, fold 3 for validation and fold 4 for test (see table 2). Trial 2 uses folds 2 and 3 for training, fold 4 for validation and fold 1 for test, and analogously for trials 3 and 4. Therefore, training, validation and test sets in all the trials include data with marbling values in the whole range of values, composing training sets of higher quality that are expected to allow regressors learn better.

K=4	Trial 1	Trial 2	Trial 3	Trial 4
Train	T_1={1,2}	T_1={2,3}	T_1={3,4}	T_1={4,1}
Validation	V_1={3}	V_1={4}	V_1={1}	V_1={2}
Test	S_1={4}	S_1={1}	S_1={2}	S_1={3}

Table 2: Distributions of folds in training, validation and test for each trial.

Table 2 reports the experimental methodolody. For each combination C_i of hyper-parameter values of the model, with $i = 1, \ldots, N$, and for each trial k, with $k = 1, \ldots, K$, the set T_k is used to train the model using the combination C_i of hyper-parameter values, while the set V_k is used as validation set to evaluate the performance P_{ik} of the trained model using C_i on V_k. The average P_i of $\{P_{ik}\}_{k=1}^{K}$ is thus the performance associated to the combination C_i of hyper-parameter values. The process is repeated for all the combinations $\{C_i\}_{i=1}^{N}$, and the combination C_I with the highest performance $I = \text{argmax}\{P_i\}_{i=1}^{N}$ is selected. Then, for $k = 1, \ldots, K$ the model with this

best combination C_I is trained on the set $\{T_k, V_k\}$, that includes $K-1$ folds, and tested on the set S_k (1 fold), achieving a performance P_k. The average of $\{P_k\}_{k=1}^{K}$ is the final test performance P of the model.

Algorithm 2: Experimental methodology, combining K-fold cross-validation and hyper-parameter tuning.

```
for i = 1 : N do
    for k = 1 : K do
        Train the model with C_i on dataset T_k
        Validate the model with C_i on dataset V_k
        P_ik=performance with C_i on V_k
    P_i=average of {P_ik}_{k=1}^K
I = argmax{P_i}_{i=1}^N
for k = 1 : K do
    Train the model with C_I on {T_k, V_k}
    Test the model with C_I on S_k
    P_k=performance with C_I on S_k
P=average of {P_k}_{k=1}^K
```

5. Results and discussion

We present the results obtained from the different points of view: the automatic extraction of ROIs to represent each muscle (subsection 5.1), the performance of marbling prediction model (subsection 5.2), comparison among the performance of different prediction models (subsection 5.3), the method stability with the muscles (subsection 5.4) and the computational time of the different stages of the procedure (subsection 5.5).

5.1. Extraction of ROIs

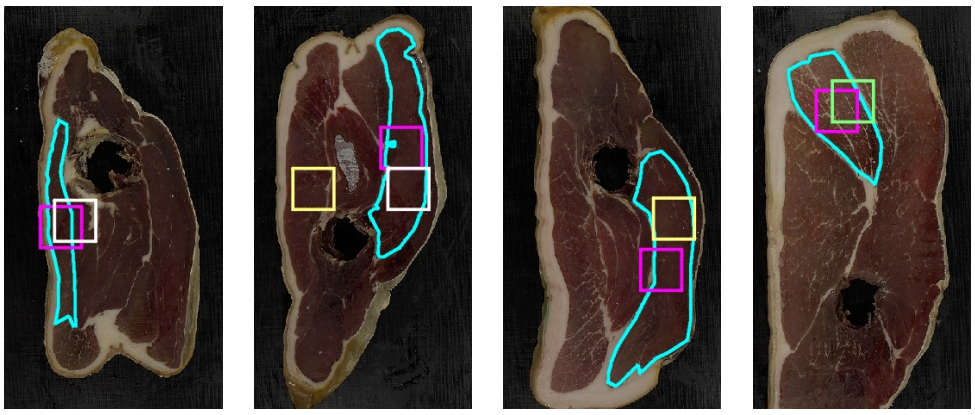
We tested the performance of the automatic algorithm to extract square ROIs, of size 64×64 pixels, for each muscle described in section 3.1, by using the measures described in eq. 2 of section 4 between the true muscle and the extracted ROI for each muscle m. We compared the results with the extraction done for experiment 2, in which the ROI is extracted from the centroid of the irregular region (muscle) annotated by the expert (see section 2.3). This extraction is normally satisfactory: in experiment 2, an overlapping of 100% is achieved for the 47% of images, and the overlapping is below 80% for only the 8% of the images. In experiment 3, the extraction is perfect for the 52% of images and the overlapping is lower than 80% for only the 12% of the images. Figure 7 shows some examples of suboptimal ROI extraction: for the ROI extraction using centroid (experiment 2), the causes of suboptimal extraction are muscles smaller than the ROI size (figure 7a) and muscles not rounded (figure 7c). For the automatic ROI extraction (experiment 3), the causes of failure are: i) suboptimal extraction (figure 7d) and ii) exchange SM and ST muscles due to artefacts in the ham slice (figure 7b). Table 3 shows the OP_m achieved by the ROI extraction algorithm for each muscle m in both experiments.

5.2. Marbling prediction

The texture features described in section 3.2 are computed for all the ham slices. In the experiment 1, the texture features are computed on the irregular

Muscle (m)	#images	OP_m Experiment 3	OP_m Experiment 2
Biceps femoris	351	93.79	98.54
Semimembranosus	335	89.92	89.20
Semitendinosus	55	94.76	99.84

Table 3: Average percentage of overlapping pixels of the automatic extracted ROI (column **Experiment 3**) and the extraction from the centroid (column **Experiment 2**) with the true muscle annotated by the experts used in the experiment 1 for every muscle.



a b c d

Figure 7: Examples of square ROI extraction, in cyan the muscle contour, in pink the square ROI extracted in the centroid, in white, yellow and green the square ROI automatically extracted for BF, SM and ST muscles respectively: a) ROI bigger than the BF muscle; b) the automatic ROI extractor fails in SM muscle; c) and d) suboptimal ROIs for SM and ST, respectively.

region, R_i, defined by each muscle and annotated by the food technology experts. So, Gabor and wavelet features were not computed because they must be applied on a square image. In experiment 2, squared regions RS_i of size 64×64 pixels are extracted from the centroid of R_i and all the texture features were computed. In experiment 3, the texture features were computed on the square regions automatically extracted from the ham slice using the

algorithm described in the section 3.1.

	Experiment 1		Experiment 2		Experiment 3	
Feature vector	R	MAE	R	MAE	R	MAE
Pure color features						
CMRGB	0.85	0.62	0.79	0.70	0.72	0.79
CMab	0.69	0.82	0.63	0.86	0.58	0.93
CMVRGB	0.90	0.50	0.86	0.57	0.81	0.67
CMVab	0.74	0.75	0.66	0.83	0.68	0.86
FOSRGB	**0.91**	0.47	**0.87**	0.54	**0.84**	0.63
FOSab	0.81	0.68	0.74	0.76	0.71	0.78
Grey-level texture features						
HarRGB	**0.91**	0.49	**0.88**	0.56	**0.80**	0.72
HarLab	0.90	0.50	0.85	0.61	0.77	0.76
mlbpRGB	**0.93**	0.43	**0.83**	0.62	**0.73**	0.79
mlbpLab	0.92	0.45	0.81	0.66	0.71	0.80
WTrgbHaarLL	–	–	0.85	0.57	0.80	0.67
WTrgbHaarAll	–	–	0.88	0.54	0.80	0.66
WTlabHaarLL	–	–	0.85	0.57	0.80	0.67
WTlabHaarAll	–	–	**0.88**	0.54	**0.82**	0.65
WTrgbDb4LL	–	–	0.86	0.58	0.79	0.70
WTrgbDb4All	–	–	0.88	0.53	0.79	0.70
WTlabDb4LL	–	–	0.87	0.57	0.80	0.70
WTlabDb4All	–	–	0.89	0.53	0.79	0.70
GaborRGB	–	–	**0.89**	0.54	**0.80**	0.70
GaborLab	–	–	0.88	0.55	0.80	0.71
Color texture features						
CMVHarRGB	**0.95**	0.38	0.91	0.47	0.84	0.64
FOSHarRGB	0.95	0.39	**0.92**	0.46	0.83	0.63
CMVmlbpRGB	0.93	0.41	0.89	0.51	0.79	0.68
FOSmlbpRGB	0.94	0.40	0.90	0.48	0.80	0.67
CMVWTLabHaarAll	–	–	0.89	0.53	0.82	0.65
FOSWTLabHaarAll	–	–	0.89	0.51	0.83	0.62
CMVGaborRGB	–	–	0.91	0.48	0.82	0.65
FOSGaborRGB	–	–	0.91	0.48	0.82	0.64

Table 4: Correlation (column R) and mean absolute error (column MAE) for the marbling prediction using pure color features (upper part of the table), grey level texture features (middle part of the table) and color texture features (lower part of the table) using the support vector regression (svr) for the three experiments.

Table 4 shows the correlation (R) and the mean absolute error (MAE) for marbling prediction using the support vector regression (svr) for the three

experiments. The feature vector FOSRGB achieved the highest R and lowest MAE among all pure color features (upper part of the table), with R=0.91, 0.87 and 0.84 in experiments 1, 2 and 3, respectively. As expected, when the performance of the automatic detection degrades, the regression accuracy decreases. But, it is still quite high for the third experiment, very good to excellent following the Colton's criteria. In relation with grey-level texture features (middle part of table), the use of irregular regions of the muscle (experiment 1) achieves the highest performance (R=0.93 and MAE=0.43 using the texture vector mlbpRGB). In general, the performance achieved in experiment 2 is higher than the experiment 3 for all the features vectors used, noting the loss of information in the automatic selection of the ROIs. Comparing the different families of grey-level texture features for square ROIs, the wavelet features achieve the highest performance for experiment 3 (R=0.82 and MAE=0.65) followed very close by the Haralick's coeficients and Gabor features (R=0.80), which means that the model is also from good to excellent. From the color space point of view, although the best results with wavelet features are achieved using the Lab color space, the difference with the use of RGB color space is not significant and the highest performance with the remaining feature vectors is better using RGB color space.

We developed experiments combining the best pure color features (vectors FOSRGB and CMVRGB) with the best grey-level texture features of each texture features families. Specifically, we chose the grey-level texture vectors HarRGB for Haralick's coeficients, mlbpRGB for local binary patterns, WT-

LabHaarAll for wavelet features and GaborRGB for Gabor features. The results are shown in the lower part of table 4. The performance increases 0.2 for the first experiment (from R=0.93 for mlbpRGB to R=0.95 for CMVHar-RGB) and 0.3 for the experiment 2 (from R=0.89 for GaborRGB to R=0.92 for FosHarRGB), but R does not increase in experiment 3. In all the cases, the best results were achived combining the color information (CMVRGB or FOSRGB vectors) and the grey-level texture information provided by the Haralick's or LBP coeficients (vector HarRGB and mlbpRGB). The MAE is a performance measure easier to interpret than R from the point of view of food technology experts. Figure 8 shows the reliability of the prediction using the **svr** regressor for experiments 1 (left panel) and `rf` regressor for experiment 3 (right panel) for the best feature vectors. The blue line represents the true marbling for each ham slice provided by the experts. The red points represent the predicted marbling by **svr** or **rf** for each ham slice. The average difference between the blue and red values for each ham slice is the MAE (0.38 and 0.60 in the left and right panels, respectively). This means that the prediction of the computer is the true value $\pm$ MAE in average. Taking in mind that the expert's tolerance in the sensorial analysis to establish the marbling is 0.5, the prediction of the computer is comparable with the precision of the experts.

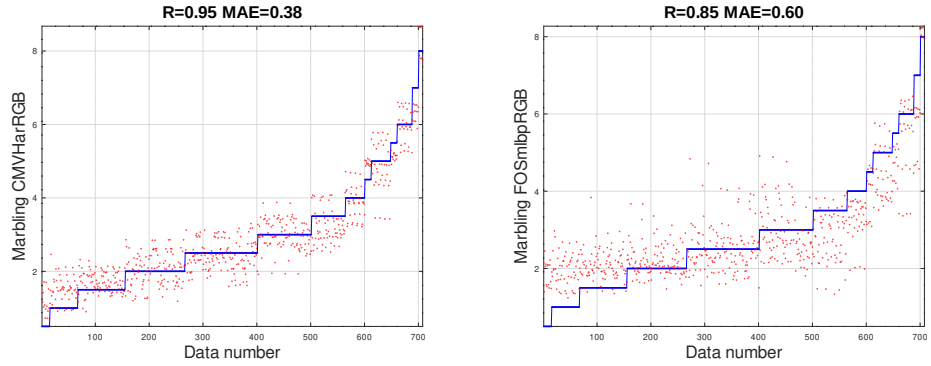


Figure 8: Graphical representation of the true (blue line) and predicted (red points) ham marbling (vertical axis) for all ham slices (horizontal axis) using the svr regressor: (left panel) using the CMVHarRGB vector in the experiment 1 and (right panel) using the FOSRGB plus LBPRGB vector in the experiment 3.

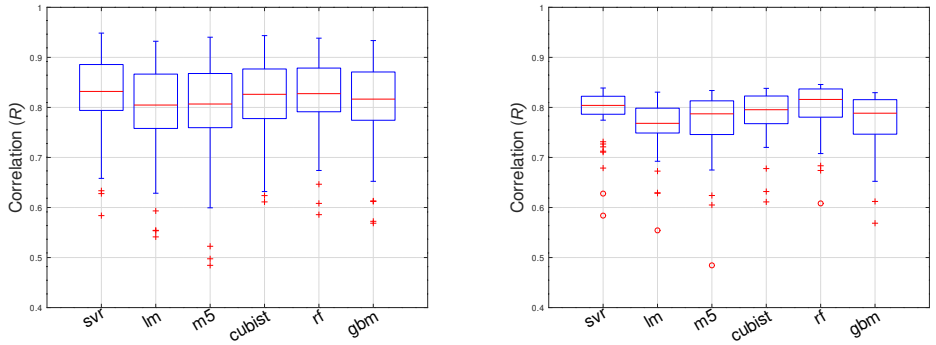


Figure 9: Box plots showing the correlation R for the different regressors (horizontal axis) considering all the experiments together (left panel) and only the experiment 3 (right panel).

5.3. Comparing different regressors for marbling prediction

In order to find the best prediction of marbling score, we applied the regressors described in section 3.3 to the color texture features described in section 3.2 for the three experiments. Figure 9 shows the box plots com-

paring the correlation for all regressors considering all feature vectors in the experiments 1, 2 and 3 (left panel) and only in experiment 3 (right panel). The upper and lower edges of each box indicate the 25th and 75th percentiles, respectively, the upper and lower blue segments enclose the remaining data and the red crosses are the data considered outliers. The red line inside the box is the median of the data. In both panels, there are not large difference among regressors. Considering the tree experiments (left panel), svr, rf and cubist have the highest medians, and svr has the smaller box, similar to rf but with higher median. In experiment 3 (right panel), the rf box shows the best median, followed by svr with a shorter box. Table 5 shows the color texture vector that achieved the highest correlation R, with the standard deviation over the K folds in the cross validation, for each regressor on the first (irregular regions for each muscle) and third experiments (square regions for each muscle). In experiment 3, the highest values of R are achieved by the rf regressor using the feature vector FOSmlbpRGB ($R = 0.846 \pm 0.008$). Considering that the MAE values (0.38 and 0.60) reflect the dispersion of the computer marbling prediction from the true marbling values, the computer predictions fall near the experts' tolerance, which is 0.5. Hence, for the first experiment, the computer predicts correctly the 90.4% (67.8% for the third experiment) of samples within a tolerance of ± 0.5 (the expert's tolerance) and within a tolerance of ± 1, the 99.3% (88.8% for the third experiment).

	Experiment 1		Experiment 3	
Regressor	Feature	R	Feature	R
lm	CMVmlbpRGB	0.932 ± 0.003	FOSHarRGB	0.831 ± 0.014
svr	CMVHarRGB	**0.948 ± 0.004**	CMVHarRGB	**0.839 ± 0.005**
m5	FOSmlbpRGB	0.940 ± 0.003	FOSHarRGB	0.834 ± 0.017
cubist	FOSmlbpRGB	0.943 ± 0.007	FOSHarRGB	0.838 ± 0.012
gbm	FOSmlbpRGB	0.934 ± 0.001	FOSRGB	0.829 ± 0.013
rf	FOSmlbpRGB	0.938 ± 0.004	FOSmlbpRGB	**0.846 ± 0.008**

Table 5: Correlation (R) for the marbling prediction using all the regressors and the best feature vector (column Feature) for the first and third experiments.

5.4. Comparing performance for different muscles

In order to test if the regressor behaviour is stable over the different muscles, we create two datasets with images belonging to the third experiment: 1) BFData for images of *Biceps femoris* (BF) muscle; and 2) SMData for images of *Semimembranosus* (SM) muscle. *Semitendinosus* muscle is not analysed because there are few images. For BF muscle, the best performance is achieved by the feature vector FOSGaborRGB using the `cubist` regressor (R=0.80 and MAE=0.50). For the SM muscle, the best performance is achieved by the feature vector FOSmlbpRGB using also the `cubist` regressor (R=0.74 and MAE=0.55). Although the correlation loss is 0.12 (0.92−0.80) and 0.18 (0.92−0.74) for the BF and SM muscles respectively, the loss in MAE is only 0.04 (0.50−0.46) and 0.09 (0.55−0.46) respectively. Thus, the computer predictions with a tolerance of ±1 are correct for the 93.5% of BF and 94.1% of SM muscles. This facts lead to conclude that our system is stable with the different muscles.

5.5. Computation time

In order to design a computer system to predict the marbling from a ham slice, which should operate in real time in the meat industry, it is very important the computational time needed by the different stages of the process. The experiments were performed on a desktop computer with Intel® Core^{TM} i7-9700 processor at 3.6GHz and 64GB RAM memory under Ubuntu 20.04. The algorithm for the extraction of a square ROIs for the different muscles was done in the C++ programming language using the computer vision library OpenCV[6] and the remaining processing was done using Matlab 2021a[7]. The average computational time to extract the ROIs was 18.7, 18.4 and 19.9 miliseconds for *biceps*, *semimembranosus* and *semitendinosus* muscles, respectively. The time required to compute the color texture features depends on the method used and the type of experiment (irregular regions in experiment 1 and square regions in experiments 2 and 3). For square regions of 64×64 pixels, the average computational time for each family of features was: 1) for pure color features the time ranges from 0.48 ms. for CMRGB to 2.21 ms. for FOSLab; 2) for Haralick's features: 7.59 and 9.20 ms. for HarRGB and HarLab, respectively; 3) for LBP texture features, the time is 170.03 ms. for mlbpRGB and 172.20 ms. for mlbpLab; 4) for wavelet features (vectors WT_{ijk}, where $i = \{Haar, Daub\}$, $j = \{RGB, Lab\}$ and $k = \{LL, All\}$) the time ranges from 1.17 ms. for WTrgbHaarLL to 2.73

[6]https://opencv.org
[7]https://mathworks.com

ms for WTLabDb4All; and 5) for Gabor texture features: 19.17 and 20.99 ms. for GaborRGB and GaborLab, respectively. The time spent by the svr regressor to predict the marbling using the texture feature vector is less than 1 milisecond per image. Overall, the computational time required by the whole process depicted in figure 4, discarding the acquisition of the ham slice image, can be estimated as 19 ms. for the automatic ROI extraction, plus less than 10 ms. for color texture feature computation, plus 1 ms. for regression model, summarized approximately 30 ms.

6. Conclusions and future work

This paper proposes a system to predict the marbling of dry-cured ham from a ham slice image. After the acquisition of ham image, a square ROI of the *semimembranosus* (SM), *semitendinosus* (ST) and *biceps femoris* (BF) muscles is automatically extracted using the procedure described by algorithm 1. The overlapping of the ROIs extracted by this method with the true muscle area is, in average, higher than 90% for all the muscles. The prediction of ham marbling using the support vector regression is: 1) a correlation R of 0.95 using the true ham muscles areas annotated by experts (experiment 1) and the feature vector CMVHarRGB, composed by the mean and variance of each channel of RGB color image combined with the Haralick's coeficients of the grey-level image; and 2) R=0.85 for square ROIs automatically extracted, in the experiment 3, using the feature vector FOSmlpbRGB (statistics of each channel of RGB color image combined with the

MLBP texture features of the grey-level image). The MAE achieved is 0.38 in the first case and 0.60 in the second one. These values are comparable to 0.5, which is the estimated standard deviation of the panelists. This leads to think that the computer system can perform the prediction similarly to a human expert. The computational time to do the prediction (without the image acquisition time) is approximatly 30 miliseconds to extract the square ROI, compute the color texture features and predict the marbling in a general purpose personal computer.

The good results and high speed of the marbling prediction for sliced dry-cured ham suggest that this application could be deployed in the dry-cured ham industry. Future work will be the development of a software to predict the marbling and other dry-cured sensorial measures.

Acknowledments

This work has received financial support from the Xunta de Galicia (Centro singular de investigación de Galicia, accreditation 2020–2023) and the European Union (European Regional Development Fund—ERDF), Project ED431G-2019/04. IRTA's contribution was also funded by the CCLabel project (RTI-2018- 096883-R-C41) and the CERCA programme from Generalitat de Catalunya.

References

E. Coll-Brasas, P. Gou, J. Arnau, A. Olmos, E. Fulladosa, Processing parameters involved in the development of texture and tyrosine precipitates in dry-cured ham: Modelisation of texture development, Meat Sci 172 (2021) 108362. doi:https://doi.org/10.1016/j.meatsci.2020.108362.

L. Lorido, E. Pizarro, M. Estévez, S. Ventanas, Emotional responses to the consumption of dry-cured hams by spanish consumers: A temporal approach, Meat Sci 149 (2019) 126–133. doi:https://doi.org/10.1016/j.meatsci.2018.11.015.

M. de Prados, E. Fulladosa, P. Gou, I. Muñoz, J. Garcia-Perez, J. Benedito, Non-destructive determination of fat content in green hams using ultrasound and x-rays, Meat Sci 104 (2015) 37–43. doi:https://doi.org/10.1016/j.meatsci.2015.01.015.

E. Cernadas, M. Durán, T. Antequera, Recognizing marbling in dry-cured Iberian ham by multiscale analysis, Patt Recog Lett 23 (2002) 1311–1321.

D. Moines, Marbling standards, National Pork Producers Council, 1999.

M. K. Dutta, A. Issac, N. Minhas, B. Sarkar, Image processing based method to assess fish quality and freshness, J Food Engin 177 (2016) 50–58. doi:https://doi.org/10.1016/j.jfoodeng.2015.12.018.

J. Dias, P. Lage, A. Garrido, E. Machado, C. Conceição, S. Gomes, A. Martins, A. Paulino, M. Duarte, N. Alvarenga, Evaluation of gas holes in

660 "Queijo de Nisa" PDO cheese using computer vision, J Food Sci Technol 58
661 (2021) 1072–1080. doi:https://doi.org/10.1007/s13197-020-04621-0.

662 S. Srivastava, S. Vaddadi, S. Sadistap, Quality assessment of commercial
663 bread samples based on breadcrumb features and freshness analysis using
664 an ultrasonic machine vision (uvs) system, J Food Meas Charact 9 (2015)
665 525–540.

666 E. Cernadas, P. Carrión, P. G. Rodríguez, E. Muriel, T. Antequera, Analyz-
667 ing magnetic resonance images of Iberian pork loin to predict its sensorial
668 characteristics, Comput. Vis. Image Und. 98 (2005) 345–361.

669 M. Ávila, M. Durán, T. Antequera, D. Caballero, T.Palacios-Pérez, E. Cer-
670 nadas, M. Fernández-Delgado, Magnetic resonance imaging, texture anal-
671 ysis and regression techniques to non-destructively predict the quality
672 characteristics of meat pieces, Engin Appl Artif Intel 82 (2019) 110–125.
673 doi:10.1016/j.engappai.2019.03.026.

674 P. Jackman, D.-W. Sun, P. Allen, Automatic segmenta-
675 tion of beef longissimus dorsi muscle and marbling by
676 an adaptable algorithm, Meat Sci 83 (2009) 187–194.
677 doi:https://doi.org/10.1016/j.meatsci.2009.03.010.

678 J.-H. Liu, X. Sun, J. Young, L. Bachmeier, D. Newman, Predicting pork
679 loin intramuscular fat using computer vision system, Meat Sci 143 (2018)
680 18–23. doi:https://doi.org/10.1016/j.meatsci.2018.03.020.

B. Uttaro, S. Zawadski, I. Larsen, M. Juárez, An image analysis approach to identification and measurement of marbling in the intact pork loin, Meat Sci 179 (2021) 108549. doi:https://doi.org/10.1016/j.meatsci.2021.108549.

E. Santos-Garcés, I. Muñoz, P. Gou, N. Garcia-Gil, E. Fulladosa, Including estimated intramuscular fat content from computed tomography images improves prediction accuracy of dry-cured ham composition, Meat Sci 96 (2014) 943–947. doi:https://doi.org/10.1016/j.meatsci.2013.09.018.

I. Muñoz, P. Gou, E. Fulladosa, Computer image analysis for intramuscular fat segmentation in dry-cured ham slices using convolutional neural networks, Food Control 106 (2019) 10.

L. Faucitano, J. Rivest, J. P. Daigle, J. Lévesque, C. Gariepy, Distribution of intramuscular fat content and marbling within the longissimus muscle of pigs, Canadian Journal of Animal Science 84 (2004) 57–61. doi:10.4141/A03-064.

L. Liu, M. Ngadi, S. Prasher, C. Gariépy, Objective determination of pork marbling scores using the wide line detector, J Food Engin 110 (2012) 497–504. doi:https://doi.org/10.1016/j.jfoodeng.2011.11.008.

H. Huang, L. Liu, M. Ngadi, C. Gariépy, Prediction of pork marbling

scores using pattern analysis techniques, Food Control 31 (2013) 224–229. doi:https://doi.org/10.1016/j.foodcont.2012.09.034.

I. Muñoz, M. Rubio-Celorio, N. Garcia-Gil, M. D. Guàrdia, E. Fulladosa, Computer image analysis as a tool for classifying marbling: A case study in dry-cured ham, J Food Engin 166 (2015) 148–155. doi:https://doi.org/10.1016/j.jfoodeng.2015.06.004.

L. Velásquez, J. Cruz-Tirado, R. Siche, R. Quevedo, An application based on the decision tree to classify the marbling of beef by hyperspectral imaging, Meat Sci 133 (2017) 43–50. doi:https://doi.org/10.1016/j.meatsci.2017.06.002.

R. Bermúdez, D. Franco, J. Carballo, J. Lorenzo, Physicochemical changes during manufacture and final sensory characteristics of dry-cured Celta ham. Effect of muscle type, Food Control 43 (2014) 263–269.

E. Cernadas, M. Fernández-Delgado, E. González-Rufino, P. Carrión, Influence of normalization and color space to color texture classification, Pattern Recogn. 61 (2017) 120 – 138.

N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans Syst Man Cyb 9 (1979) 62–66. doi:10.1109/TSMC.1979.4310076.

E. González-Rufino, P. Carrión, E. Cernadas, M. Fernández-Delgado, R. Domínguez-Petit, Exhaustive comparison of colour texture features

and classification methods to discriminate cells categories in histological images of fish ovary, Pattern Recogn. 46 (2013) 2391–2407.

R. M. Haralick, K. Shanmugan, I. Dinstein, Textural features for image classification, IEEE T Man Cyb. 3 (6) (1973) 610 – 621.

T. Ojala, M. Piatikäinen, T. Mäenpää, Multiresolution grey-scale and rotation invariant texture classification with Local Binary Pattern, IEEE T Pattern Anal. 24 (7) (2002) 971–987.

J. S. Walker, A Primer on wavelets and Their Scientific Applications, Chapman Hall, 2008.

A. Laine, J. Fan, Texture classification by Wavelet Packet signatures, IEEE T Pattern Anal. 15 (11) (1993) 1186–1191.

T. Randen, J. H. Husoy, Filtering for Texture Classification: A Comparative Study, IEEE T Pattern Anal. 21 (4) (1999) 291–310.

F. Bianconi, A. Fernández, Evaluation of the effects of Gabor filter parameters on texture classification, Pattern Recogn. 40 (2007) 3325–3335.

M. Fernández-Delgado, M. Sirsat, E. Cernadas, S. Alawadi, S. Barro, M. Febrero-Bande, An extensive experimental survey of regression methods, Neural Netw 111 (2019) 11–34. doi:10.1016/j.neunet.2018.12.010.

M. Kuhn, Caret: Classification and regression training, 2016. R package version 6.0-70.

D. Bates, J. Chambers, Statistical models in S, Wadsworth & Brooks/Cole, Pacific Grove, CA, 1992.

C. Chang, C. Lin, LIBSVM: a library for support vector machines, ACM T Intel. Syst. Tec. 2 (2011) 27:1–27:27.

R. Quinlan, Learning with continuous classes, in: Proc. Australian J. Conf. on Artif. Intel., 1992, pp. 343–348.

R. Quinlan, Combining instance-based and model-based learning, in: Proc. Intl. Conf. on Mach. Learn., 1993, pp. 236–243.

L. Breiman, Random forests, Mach Learn 45 (2001) 5–32.

T. Colton, Statistical in medicine, Little Brown and Co., NJ, 1974.