Computer image analysis for intramuscular fat segmentation in dry-cured ham slices using convolutional neural networks.

*I. Muñoz, P. Gou, E. Fulladosa,

IRTA. Food Technology. Finca Camps i Armet 17121 Monells, Girona, Spain

*Corresponding author: Israel Muñoz. israel.munoz@irta.es

ABSTRACT

Determination of intramuscular fat (IMF) content in dry cured meats is critical because it affects the sensory quality and consumer's acceptability. Recently, deep learning has become one of the most promising techniques in machine learning for image analysis. However, few applications in food products are found in the literature. This study presents the application of deep learning for the detection of intramuscular fat (IMF) in images of slices of dry cured ham. 8 convolutional neural networks (CNNs) have been studied and compared using segmented images (252 for training, 61 for validation and 62 for testing). The performance was compared to other simple CNNs. CNNs were able to segment IMF with an overall pixel accuracy of 0.99 and a recall and precision rates for fat near 0.82 and 0.84, respectively, using a limited number of training images. However, performance is affected by the quality of the ground truth due to the difficulty of labelling correctly pixels.

Keywords: Convolutional neural network, deep learning, intramuscular fat, image analysis, dry-cured ham

1. INTRODUCTION

The amount of visible fat in dry-cured ham and distribution of fat streaks, affects palatability and consumers acceptability. Marbling is used as a visual cue by consumers to judge dry-cured ham quality. Although high IMF content is closely related to positive emotional responses during consumption of dry-cured ham (Lorido, Pizarro, Estévez & Ventanas, 2019), consumers prefer to purchase ham with moderate amounts of IMF, linked to positive nutritional and flavour characteristics (Morales, Guerrero, Aguiar, Guàrdia & Gou, 2013). This is a challenge for the industry, since the amount of IMF, even within the same breed, can widely vary. For the industry, it is of interest to characterize online the IMF of slices of dry cured ham. This will allow the companies to segment the market, and offer products tailored to the consumers' needs.

Computer image analysis (CIA) is a reliable alternative for fast and non-destructive assessment of food characteristics such as colour, freshness, textural properties and other quality aspects. Some applications include the determination of marbling scores in pork meat (Liu, Ngadi, Prasher & Gariépy, 2012), the assessment of fish quality and freshness (Dutta, Issac, Minhas & Sarkar, 2016) and the quality assessment of pizza (Sun & Brosnan, 2003), cheese (Caccamo et al., 2004) and bread (Srivastava, Vaddadi & Sadistap, 2015). CIA has also been applied to grading of fruits and vegetables (Blasco, Munera, Aleixos, Cubero & Molto, 2017).

IMF detection using CIA is challenging because IMF cannot be easily characterized. For this reason, simple segmentation approaches are not useful and more sophisticated techniques are needed. For example, a segmentation-based approach was reported by Jackman, Sun and Allen (2009), which used K-means clustering to segment images of beef *Longissimus dorsi* muscle into background, lean muscle, and intramuscular fat areas. Results showed that IMF pixels were underestimated by 12.4% with respect to ground truth images. One of the most usual techniques for IMF detection is line detection algorithms. Faucitano, Huff, Teuscher, Gariepy and Wegner (2005) evaluated marbling by enhancing the colour contrast of pork meat samples using chemical pre-treatments and line detection algorithms. The authors did no check the accuracy of this approach. Liu, Milan, Shen and Reid (2012) and Huang, Liu, Ngadi and Gariépy (2013) used a line detection algorithm for determining a marbling score of pork loins and pork chops, respectively . Qiao, Ngadi, Wang, Gariépy and Prasher (2007) studied the potential of hyperspectral imaging techniques to assess pork quality and marbling levels using a hyperspectral imaging system and artificial neural networks. Both authors focused on the ability of these algorithms to predict marbling scores.

Recently, Lohumi et al. (2016) applied hyperspectral imaging for the characterization of intramuscular fat in beef. Several methods were evaluated and the accuracy ranged from 91% to 96%. Velázquez, Cruz-Tirado, Siche and Quevedo (2017) segmented fat and classified the degree of marbling in beef from hyperspectral images using decision trees. Decision trees were able to reach an accuracy of 99.92% for the classification of lean and fat pixels during the construction of the tree (training). Liu, Ngadi, Prasher and Gariépy

(2018) segmented fat by automatically estimating the threshold between the lean and fat tissues. No information on accuracy was given.

In dry-cured ham, segmentation of IMF is more complex. The variation of dryness and colour across the slice, the presence of phosphates and tyrosine crystals and, in some cases, of nitrification rings make image segmentation more difficult. Cernadas, Dur and Antequera (2002), by using a multi-scale line detection framework for the recognition of fat streaks in the image, correctly classified 90% of the fat streaks with an acceptable rate of false positives. Widiyanto et al. (2013) segmented correctly IMF and lean in slices of dry-cured ham using fuzzy c-means and bias field estimation, obtaining a dice similarity coefficient of 0.94 for lean and 0.88 for IMF. Muñoz, Rubio-Celorio, Garcia-Gil, Guardia and Fulladosa (2015) and Santos Garcés, Muñoz, Gou, Garcia-Gil and Fulladosa (2014) used gradient-based techniques, such as discrete Fourier transform (DFT), but not evaluated the accuracy of the IMF estimation. However, new approaches for image analyses have been developed in the previous decade, which allow researchers to develop powerful algorithms for complex tasks. One of these new tools is deep learning (Goodfellow, Bengio, Courville & Bengio, 2016), in particular, deep convolutional networks. A convolutional neural network (also known as CNN or ConvNet) is a type of neural network used for deep leaning in image applications. CNNs are used in a wide range of applications in image analysis. For example, object recognition (He, Zhang, Ren & Sun 2016), image classification (Krizhevsky, Sutskever & Hinton 2012) or image segmentation (Badrinarayanan, Kendall & Cipolla, 2017). The main advantage of this technique is that eliminates the need for hand-engineered filter design, as those are learned by the CNN itself. In the last few years, this technique has been applied to an increasing number of problems. In many cases, the performance of CNN has outperformed conventional CIA algorithms, becoming the state of the art solutions for many real applications. One of the applications is pixelwise classification, also known as semantic segmentation, which aims at assigning labels to pixels in an image (Long, Shelhamer & Darrell, 2015; Lin, Milan, Shen & Reid, 2017). This is one the approaches that can be used to segment IMF in images.

Deep learning is a promising and very powerful tool to solve computer image problems. However, there are still very few applications of deep learning in the food sector. Recently, deep learning techniques have been applied to evaluate automatically the quality of fresh-cut lettuce (Cavallo, Cefola, Pace, Logrieco & Attolico, 2018), to assess nutrient concentrations of commercially prepared pureed food (Pfisterer, Amelard, Chung & Wong, 2018), or to automate the segmentation of the skeleton of pigs using CT images (Kvam, Gangsei, Kongsro & Schistad-Solberg, 2018) or the detection of salmon muscle gaping (Xu & Sun, 2018). Other applications in food include food localization and recognition in images (Bolaños & Radeva, 2016).

This study aims at the segmentation of IMF in slices of dry-cured ham using deep learning. This problem has already been addressed using conventional image analysis techniques.

2. MATERIAL AND METHODS

2.1. Sampling

Ham slices were sampled from 190 dry-cured hams as it was described in Muñoz, Rubio-Celorio, Garcia-Gil, Guardia and Fulladosa (2015).

2.2. Image acquisition

Images were acquired with the photographic system depicted in Fig. 1. The exact methodology was described in Muñoz, Rubio-Celorio, Garcia-Gil, Guardia and Fulladosa (2015).

2.3 Ground Truth

Two regions of interest (ROIs) (both sides of a 1 cm thick slices) corresponding to the Biceps femoris (BF) muscles were manually selected from each image (Fig. 2). BF muscle was chosen because it is the biggest and the most representative muscle in dry cured ham slices. Besides, together with Semitendinosus (ST) muscle, it may have an considerable amount of intramuscular fat, which is also correlated to the ST muscle (the fattiest muscle). 375 ROIs were evaluated and 5 ROIs were discarded from the study due to defects on the surface (such as cuts and phosphate crystals) which made them unsuitable for the CIA. Patches of 64x64 pixels (one patch per image) were automatically extracted from the ROIs with three channels of information corresponding to R,G,B colour channels. All patches were treated as independent samples, as IMF distribution and colour of IMF and lean showed big differences in patches obtained from both sides of the same ham slice.

Next, reference images of correctly segmented IMF (Ground Truth) were obtained from these patches similarly to the methodology described in Muñoz, Rubio-Celorio, Garcia-

Gil, Guardia & Fulladosa, 2015) and (Santos Garcés, Muñoz, Gou, Garcia-Gil & Fulladosa, 2014). For each ROI, IMF was segmented using edge detection based on the discrete Fourier transform (DFT) (Rangayyan, 2004). DFT followed by a gaussian high pass filter with a cut-off frequency of 250 was applied to each image. After filtering, the images were transformed back using the Inverse Discrete Fourier Transform (IDFT). The real component of the transformed matrix was used for further processing. Pixels with values equal or below a threshold value were labelled as IMF. This threshold value was set manually. An expert in the field of food technology, trained for the sensory evaluation of foods and specially for dry-cured ham visual evaluation was responsible for adjusting the threshold values following the guidelines established for dry-cured ham by Claret, Guerrero, Guàrdia, Garcia-Gil and Arnau (2009). After this, most of the IMF was correctly segmented. However, several thresholding operations in combination with different logical operators (AND, OR, NOT) were applied to the image (combining the IDFT transform image and the RGB image) for the segmentation of still incorrectly segmented pixels. This work was also carried out by the trained food technologist and the threshold values adjusted accordingly. In some cases, even for a trained expert, it was difficult to decide whether a pixel should be labelled as fat or lean, in particular for small fat streaks and contour pixels due to the wide range of RGB values for fat and lean, structure of fat, etc

Small size patches (3x64x64 pixels) were used in order to have same size samples for training (BF muscles are different in shape and in the number of pixels) and speed up learning.

2.4 Convolutional neural network architecture

The convolutional neural network (CNN) was trained to classify pixels into two different classes (class 0: lean, class 1: fat) using pixelwise classification (semantic segmentation). Ground Truths for images were determined as described in section 2.3 and were used as labelled images during training of the CNNs. In the CNN architecture used in this work, four of the most common types of operation in a CNN were used: convolution, non-linear, pooling and upsampling layers.

In Convolution layers, a filter (also known as kernel) performs the convolution operation over a matrix (images). Convolution can be thought as a sliding window function applied to a matrix. The number of parameters to be learned in these filters is equal to the number of elements of these filters (depth x height x width) (Fig. 3a). In this work (as in others studies in the field), when referring to filters only the height and the width is given, whereas the depth can be obtained from the depth of the input matrix (image).

Non-linear layers are usually placed right after convolution layers. Non-linear layers perform a non-linear operation on the matrices resulting from the convolution operation, similar to the sigmoid function. The most common function in CNN is the rectifier linear function (ReLU) (Fig. 3a).

Pooling layer reduces the size of the image, also known as downsampling. Among the existing pooling layers, average and max pooling are the most common ones. Pooling layer consists of a sliding window function that moves over the matrix and takes the largest value in the window. The matrix is partitioned into several non-overlapping regions where the operation associated with pooling is applied. Pooling reduces the size of the matrix. In Fig. 3b, the max pooling is applied.

Upsampling layers can be considered as a kind of reverse convolution (Fig. 3b), sometimes denoted as deconvolution. Upsampling resizes an input matrix to the desired size by upsampling and interpolation (e.g. bilinear interpolation). In CNN, it can be used to resize the output of a CNN to the original size after convolutional and pooling operations have reduced the size of the original image.

Fig. 4 depicts the basic architecture used in this work, in the case of using 512 filters in the first layer. This architecture is based on the work by Long, Shelhamer and Darrell (2015), in which information from different layers of the CNN are combined to make predictions, and the VGG net (Simonyan & Zisserman, 2015), in which the number of filters increases with the depth of the network. Prior to the final selection of the CNN architecture used in this study, several parameters were evaluated, namely number of convolutional layers (1-4), kernel size (3x3,5x5,7x7) and number of filters.

In this architecture, a RGB patch (3x64x64 pixels) is convolved by 512 3x3 convolution filters (and depth 3, as the image has three channels: R, G and B) and zero padding is applied to ensure that after convolution the height and width of the image remains the same. Zero padding consists in adding “0” around the border of the matrix of data. For 3x3 convolution filters, a zero padding of size 1 must be applied to ensure that the size does not change after convolving. Therefore, convolution, including padding, transforms the input image into 512 64x64 matrices. After convolution, the rectifier function (ReLU) is applied to each element of the obtained matrices and next, the 2x2 max pooling is applied. A 2x2 pooling reduces the size of the matrix by a factor of 2 (i.e. from 64x64 to 32x32), but it does not change the number of matrices (512). The whole structure (network layer) (Conv-ReLU-pool) is repeated 3 more times. At the end of the process, there are 4096 8x8 matrices. After each max pooling the number of matrices is increased by a factor 2 at the next convolution operation in order to keep the complexity of the network (Simonyan & Zisserman, 2015). After each pooling operation, an upsampling operation is applied, using bilinear interpolation, to obtain two matrices (two classes) with the original size (64x64 pixels). Additionally, an upsampling operation is applied to the matrices obtained at ReLU1. All 64x64 pixels obtained by upsampling at different layers are finally added together (2x64x64 pixels). According to Long, Shelhamer & Darrell (2015) the combination of information from different layers is equivalent to combine coarse, high level information with fine low layer information. This integration of information allows the network to predict finer details. Next, the output of the network (2x64x64 matrices) is passed through a softmax classifier. The output after the softmax classifier is a probability map having the same size as the input image (64x64) with each pixel having two values, the probability of belonging to class 0 (lean) and class 1 (fat). The class with the highest probability value is selected as the segmented class. The performance of this CNNs architecture is compared to other more simple CNNs architectures in which all upsampling operations are removed from the architecture with the exception of the last upsampling operation previous to the softmax classifier (Upsample 5 in Fig. 4).

In this study, different parameters of this architecture were studied, namely, the depth of the network (from 1 to 4 Conv-ReLU-pool layers) and the number of filters at Conv1 (128 and 512). In the results and discussion section, network architectures will be denoted as 2_128, first figure indicates the number of Conv-ReLU-pool structures (network layers) and the second figure indicates the number of filters at the first convolutional layer (Table 1). In total, 8 different combinations of depth of the network (1-4) and number of filters were studied (128, 256). According to this notation, Fig. 4 depicts a 4_512 architecture (4 Conv-ReLU-pool layers and 512 filters in layer 1). The same notation is used for the simple networks with the difference that only the last upsampling is included in the network (Upsample5 in Fig. 4). In this case, the number of filters at Conv 1 is 128 and 512 and the depth of the network from 1 to 3. A subscript (s) has been added to denote a simple network (Table 1).

2.5 Software and Hardware

Matlab 2008b and its image processing toolbox (The MathWorks, Inc., United States) were used to select the ROI and segment IMF in images using the procedure described in the previous section.

Caffe was used to create, train, validate and test the CNN architecture. Caffe is a deep learnig framework and stands for Convolutional Architecture for Fast Feature Embedding (Jia et al., 2014). Results were processed using Python 2.7.13. The following parameters were used in this study in Caffe: Batch size 32, the base learning rate 1e-4, the momentum 0.9, the weight decay 0.05. The learning rate policy was “inv” (learning rate decay over time) and the parameters for this policy were gamma 0.01 and power 0.5.

Caffe tries to minimize the multinominal logistic loss (also known as cross-entropy classification loss) and it was used to compute the error classification during training and optimization. Stochastic gradient descent was used for the optimization of the network. Each network was trained for 50,000 iterations. In Caffe the term iteration is used instead of epoch, for this reason the term iteration is used across the text. The equivalency between epoch and iteration is as follows:

Epoch_index=floor(iteration_index x batch_size)/(number of training data samples)

Convolution layers: Weights were initialized using “xavier” method. Bias were of type “constant” which initialises biases to zero. A learning rate multiplier of 1 was selected for the weights and a multiplier of 2 for the biases. Kernel sizes of 3x3 were used in this study and zero padding was of size 1. The stride was 1.

Upsampling layers: Upsampling layers used the “bilinear” method. For upsampling from 64x64, 32x32, 16x16 and 8x8 to 64x64 a kernel size of 1,4,8,16, a stride of 1,4,8,16 and a zero padding of size 0,1,2,4 were used, respectively.

Prior to the tests several parameters of the network were tested and adjusted: base learning rate, batch number, momentum, weight decay, gamma and power. Once, these parameters were determined, all networks structures were trained using the same values

Training, validation and testing of CNNs was performed on a Z820 workstation with 512 GB of RAM and 16 cores Intel Xeon ES-2687W at 3.10 GHz

2.6 Testing

2/3 of patches were randomly selected for training (252), 1/6 for validation (61) and 1/6 for testing (62) by assigning a random number to each image patch. Patches were assigned to each group based on the value of the random number. This means that for the training set a total of 1,032,192 pixels (252 images x 64 rows x 64 columns) were available for training.

The following metrics were used to evaluate the performance on the test set:

tp, *tn*, *fp*, *fn* denote true positive, true negative, false positive and false negative respectively. Positive class denotes fat, negative class denotes lean.

1) Overall pixel accuracy: percentage of pixels correctly predicted (fat and lean pixels)

$$Accuracy = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

2) Fat recall rate: rate of pixels correctly predicted as fat into the total number of pixels labelled as fat.

$$Fat\ recall\ rate = \frac{t_p}{t_p + f_n}$$

3) Fat precision rate: rate of pixels correctly predicted as fat into the total number of pixels predicted as fat.

$$Fat\ precision\ rate = \frac{t_p}{t_p + f_p}$$

4) Rate of false negatives near the areas predicted as fat (FnPFc rate): rate of false negatives that are correctly predicted as fat (t'_p) after applying a 3x3 dilation operation on the pixels predicted as fat by the CNN.

$$FnPFc\ rate = \frac{t'_p}{f_n}$$

5) Rate of false positives near the areaslabelled as fat (FpLFc rate): rate of false positives that are correctly predicted as non-fat (t'_p) after applying a 3x3 dilation operation on the pixels labelled as fat (ground truth).

$$FpLFc\ rate = \frac{t'_p}{f_p}$$

At evaluation, special attention was given to segmentation of fat in the images. FnPFc and FpLFc rates attempt to incorporate the uncertainty of manual classification (classification of contour pixels) during the preparation of the ground truth images.

Dilation operations are used in computer vision for expanding the shapes contained in an image. The size of the expansion depends on the size of the operation (3x3 in this case) or the number of times the operation is applied (1 in this case).The application of a dilate operation on the pixels predicted or labelled as fat may incorporate this uncertainty into the evaluation of performance. The results presented for the different network architectures correspond to the iteration with the best overall pixel accuracy of the test set for the last 5.000 training iterations. Then, the learned parameters of the network were used to evaluate the test set. Training data was recorded every 500 iterations.

Moreover, the average time needed for the segmentation of images of the test set was also recorded.

After training, validation and testing, four representative images from the test set were segmented and analysed using the worst and the best performing (overall pixel accuracy) architectures and the segmentation was compared for the two architectures.

Results presented in the result and discussion section lack any statistical significance as the training, validation and testing was done only once, because of the long training times of the architectures studied in this investigation (4 months). This is quite common in many works in the field of CNN (i.e. Long, Shelhamer & Darrell, 2015; Ronneberger, Fischer & Brox, 2016; Lin, Milan, Shen & Reid, 2017) and many times comparisons are based on one single training (on training, validation and test sets) due to this limitation.

3. RESULTS AND DISCUSSION

Fig. 5 shows the change in the multinomial logistic loss with the number of iterations for the training and test sets of the CNNs 3_128 and 3_512. These two CNNs were chosen as an example to study the learning of the network. Logistic Loss decreased more rapidly for CNN 3_128 than for CNN 3_512 due to the lower number of learnable parameters (1,478,914 vs. 23,610,368) of the network. After approximately 4000 and 6000 iterations for CNNs 3_128 and 3_512 respectively, the loss for the training and test set tended to decrease very slowly, even though the loss for the training set decreased more rapidly than for the test set. After around 25.000 iterations, the logistic losses barely changed. One of the reasons for this result is the learning rate decay and the convergence of the learning of the network. Overfitting was not observed, as the logistic loss for the test set did not increase with the number of iterations. However, training the networks for a larger number of iterations might have increased the logistic loss for the test set (not tested).

Logistic loss of the test set was lower for CNN 3_512 than for CNN 3_128. The larger number of learnable parameters of CNN 3_512 may have captured better the complexity of the segmentation for this task. However, a 16 fold increase in the number of learnable parameters only brought about a small improvement in the performance of the network. For CNN 3_128 logistic loss for the test set was only slightly lower than that of the training set. This result can be surprising, but it is not uncommon for small size sets of test data (62 images) as chance during random selection of training, validation and test sets may produce this result. The difference between the loss for the training and test set decreased with the number of iterations.

Table 2 shows the results for the simple CNNs and the CNN architectures developed for this work. Simple CNN architectures performed worse (performance, lower recall and precision rates for fat segmentation) than those architectures specifically conceived for this work with the same number of filters in the first layer. Results also showed that performance in simple CNN increased with the number of filters in the first layer, but decreased with the number of layers. This latter result is not clearly observed for the complex CNNs presented in Table 2. However, it seems that 1_128 and 1_512, performed worse than other architectures with more layers. This result can be specially observed for 1_512 vs 3_512 and 4_512, even though it cannot be considered conclusive due to the lack of statistical significance. As expected processing time was much lower for the simple architectures.

For the architectures conceived for this study, as the number of filters and/or the number of layers increase, the number of parameters to be adjusted increases (Table 2) and thus, the CNN is expected to fit more accurately the training set, improving overall pixel accuracy of the training set. However, in our study, the overall pixel accuracy was very high (0.988) in the most simple CNN architecture (1_128) and increased to 0.991 in the most complex CNN architectures (3_512 and 4_512). These values are similar to those obtained in other works. During training, Velázquez, Cruz-Tirado, Siche & Quevedo (2017) obtained an accuracy of 0.9992 using decision trees for the segmentation of IMF in beef.

In general, increasing the number of filters and layers allows capturing better the complexity of the problem, but the overall pixel accuracy of the test set can decrease due to the well-known problem of overfitting (Hawkins, 2004), which is originated in models with more terms or more complicated approaches than necessary. In our study, the overall pixel accuracy of the test set hardly increased with the number of filters, from 0.988 for CNNs with 1_128 filters to 0.989 for CNNs with 3_512 filters and 4_512 and it did not change with the number of layers. The CNN with the highest overall pixel accuracy was 3_512. The overall accuracy tended to increase slightly with the number of learnable parameter. No drop in performance was observed with the increase of learnable parameters. Therefore, overfitting was not observed for this data and the studied architectures.

The overall pixel accuracy was highly influenced by the lean tissue segmentation of the CNNs, due to the large ratio of pixels corresponding to lean tissue in the images. The precision and recall rates of fat were also studied, as they give more accurate information than overall pixel accuracy on the performance of the CNNs for the segmentation of fat. For a similar overall pixel accuracy and for a given CNN, the fat recall and precision rates are related to each other, as an increase in one of them usually results in a decrease in the other one. In general, the fat recall rates were higher for CNN x_512 than for CNN x_128, whereas the precision rate was similar in both cases (around 0.84). These results were similar to other works found in the literature, even though metrics were not fully comparable. Jackman, Sun and Allen (2009) underestimated the number of marbling pixels (12.4% not classified as IMF) for beef. No information was given on misclassified lean pixels. For dry-cured ham, Cernadas, Dur and Antequera (2002) classified correctly 90% of the fat streaks with an acceptable rate of false positives, whereas Widiyanto et al. (2013) using a slightly different metric for accuracy (dice similarity coefficient) obtained 0.94 and 0.88 for the ham and IMF regions, respectively. For CNN 3_512 the dice similarity coefficient was calculated and similar values were obtained, 0.99 and 0.83 for lean and IMF regions, respectively.

FnPFc and FpLFc rates showed that for x_128 and x_512, around 35-40% of the misclassified pixels were found near the contours of the fat patches in the images. Similar rates were observed for the simple CNNs 1_128_s and 1_512_s. One of the reasons for these results are the difficulties faced by the trained expert during the preparation of the ground truth images. This amounts to using noisy data during training. Another possible reason was the lack of enough samples for training, due to the wide range of possible RGB values for the fat and lean, structures of the fat, etc. For 2_128_s, 3_128_s, 2_512_s and 3_512_s, the FpLFc rate was much higher than the FnPFc rate. This may indicate that these CNNs was overestimating the contours of the fat patches.

Processing time increased with the number of filters in the first layer and the depth of the CNN. In general, an increase in performance resulted in an increase in the processing time. High processing times (i.e 410 ms for CNN 4_512) might be a problem for the segmentation of images in real-time applications. As expected, processing time increased with the number of filters and the depth of the network. For example, for CNN 2_128 average processing time was 20 ms, whereas for CNN 2_512 was 58 ms. This represents an increase of the processing time by a factor of almost 3, for an increase by a factor of 2 in the number of filters in the first layer. CNN 1_512_s and CNN 1_128 had a similar performance (fat precision and recall rates) on the test set. However, processing time was lower for CNN 1_128 (19 ms vs. 9 ms). This fact should be further investigated.

The CNN 3_512 and CNN 1_128 was selected (the best and worst performing architectures) to evaluate the segmentation of images from the test set. In general, the best performing CNN (3_512) was able to segment correctly raw images (Fig. 6a). Some small divergences can be observed in the contours of the fat regions between the segmented image using the CNN and the ground truth. In some particular images, some areas were not correctly segmented (Figs. 6b, 6c and 7). The reason for these divergences have already been discussed above. In some other cases, the convolutional network segmented fat patches that were not correctly selected during the preparation of the ground truth images (Fig. 6c).

Results for CNN 3_512 and CNN 1_128 showed that in 49 images out of 62 images of the evaluation set, the CNN 3_512 had equal or higher overall pixel accuracy than CNN 1_128. In those images where CNN 1_128 performed better, the differences in pixel accuracy were very small. However, in some cases CNN 3_512 was able to segment fat much better than CNN 1_128. For example, in Fig. 7, both cases did not segment correctly some of the fat pixels. However, CNN 3_512 was able to segment IMF better than CNN 1_128. Probably, due to the larger number of filters and layers, the CNN 3_512 was able to store more information on fat detection from the training samples. However, the small number of samples in the training set would rather explain the poor performance of the CNN in this case for both architectures.

This study was applied to 3x64x64 patches obtained from images. However, using different strategies, the algorithm could be applied to segment larger images. For example, using an overlap-tile strategy (Ronneberger, Fischer & Brox, 2015).

The good results obtained for the detection of intramuscular fat in sliced dry-cured ham suggests that this methodology can be of interest for the dry-cured ham industry and might be used to develop systems for food quality analysis in other food products. One of the advantages of this machine learning technique is that no specialized knowledge and skills in computer vision are required. However, some challenges must be addressed. Image processing with CNNs might be too slow for real-time image segmentation in industrial processes, especially for CNNs with many filters and layers. Moreover, training samples must be collected and labelled before training the CNN. Although, in food elaboration processes (i.e dry cured ham), training examples are available in large quantities, preparation of ground truth images can be time consuming and may require the expertise of food technologists. Although CNNs provides state-of-the-art performance in many computer vision applications, other algorithms should be also evaluated (Support Vector Machines, Decision Trees, etc) as long image processing times might be a problem for real-time applications in industry.

Detection of intramuscular fat is the first step to efficiently quantify intramuscular fat content. Deep learning algorithms in combination with information obtained using other non-destructive technologies (Fulladosa, Gou & Muñoz, 2016; Fulladosa, Rubio-Celorio, Skytte, Muñoz & Picouet 2017; Fulladosa et al., 2018; Garrido-Novell, Garrido-Varo, Perez-Marin, Guerrero-Ginel & Kim, 2015; Gou et al., 2013) might help to find a nutritional label specific for each sliced ham pack and thus encourage consumers to adopt healthier eating habits and/or buy products according to their needs and/or preferences.

Detection of colour defects, for example, due to oxidation, could be performed (i.e. using deep learning) simultaneously with the IMF segmentation. With these systems, companies could discard and/or redirect to other process the defective products. Besides, a good detection of IMF in images may also provide a tool to improve prediction precision in other technologies. Prediction error of salt and water contents using computed tomography can be reduced with a good detection and quantification of fat content (Santos Garcés, Muñoz, Gou, Garcia-Gil & Fulladosa, 2014), leading to models that improve the optimization of the dry-cured ham elaboration process.

CONCLUSIONS

Results show that deep learning is able to segment correctly IMF in dry cured ham by just using training samples in combination with CNN. CNN attained a similar performance to that of conventional image analysis algorithms, reducing development time, at the cost of requiring greater computing resources.

The increase in the complexity of the network helps to improve the performance, but up to a certain level, as the network may end up overfitting and processing time of images may increase considerably. One of the challenges is the need to obtain good training data for training the CNN, due to the difficulty in classifying pixels correctly and objectively even by trained experts. CNN opens new possibilities to solve complex detection problems in the food industry without the need of developing complex algorithms, facilitating the deployment of these technologies in the food industry.

ACKNOWLEDGMENTS

This work was partially supported by INIA (grant number RTA2013-00030-CO3-01) and CERCA programme from Generalitat de Catalunya.

BIBLIOGRAPHY

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481-2495.

Blasco, J., Munera, S., Aleixos, N., Cubero, S., Molto, E. (2017). Machine vision-based measurement systems for fruit and vegetable quality control in postharvest. In advances in Biochemical Engineering/Biotechnology book series, 161, 71-91.

Bolaños, M., & Radeva, P. (2016). Simultaneous Food Localization and Recognition. In 23rd International Conference on Pattern Recognition (ICPR) (pp. 3140–3145). Cancun, Mexico.

Caccamo, M., Melilli, C., Barbano, D.M., Portelli, G., Marino, G., & Licitra, G. (2004). Measurement of gas holes and mechanical opennesss in cheese by image analysis. Journal of Dairy Science, 87(3), 739-748.

Cavallo, D.P., Cefola, M., Pace, B., Logrieco, A.F., & Attolico, G. (2018). Non-destructive automatic quality evaluation of fresh-cut iceberg lettuce through packaging material, Journal of Food Engineering, 223, 46-52.

Cernadas, E., Dur, M. L., & Antequera, T. (2002). Recognizing marbling in dry-cured Iberian ham by multiscale analysis. Pattern Recognition Letters, 23, 1311–1321.

Claret, A., Guerrero, L., Guàrdia, M.D., Garcia-Gil, N. & Arnau, J. (2009). Desarrollo de escalas de referencia para determinados atributos sensoriales del jamón curado de cerdo blanco. V Congreso Mundial del jamón, Aracena, Huelva (Spain)

Dutta, M.K., Issac, A., Minhas, N. & Sarkar, B. (2016). Image processing based method to assess fish quality and freshness. Journal of Food Engineering, 177, 50-58.

Faucitano, L., Huff, P., Teuscher, F., Gariepy, C. & Wegner, J. (2005). Application of computer image analysis to measure pork marbling characteristics. Meat Science, 69, 537-543.

Fulladosa, E., Austrich, A., Muñoz, I., Guerrero, L., Benedito, J., Lorenzo, J. M. & Gou, P. (2018). Texture characterization of dry-cured ham using multi energy X-ray analysis. Food Control, 89 46-53.

Fulladosa, E., Gou, P. & Muñoz, I. (2016). Effect of dry-cured ham composition on X-ray multi energy spectra. Food Control, 70 41-47.

Fulladosa, E., Rubio-Celorio, M., Skytte, J. L., Muñoz, I. & Picouet, P. (2017). Laser-light backscattering response to water content and proteolysis in dry-cured ham. Food Control, 77 235-242.

Garrido-Novell, C., Garrido-Varo, A., Perez-Marin, D., Guerrero-Ginel, J. E. & Kim, M. (2015). Quantification and spatial characterization of moisture and NaCl content of Iberian dry-cured ham slices using NIR hyperspectral imaging. Journal of Food Engineering, 153 117-123.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT press.

Gou, P., Santos-Garcés, E., Hoy, M., Wold, J. P., Liland, K. H. & Fulladosa, E. (2013). Feasibility of NIR interactance hyperspectral imaging for on-line measurement of crude composition in vacuum packed dry-cured ham slices. Meat Science, 95 (2), 250-255.

Hawkins, D.M. (2004). The problem of overfitting. Journal of Chemical Information and Computer Sciences, 44(1), 1-12.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Huang, H., Liu, L., Ngadi, M.O., & Gariépy, C. (2013). Prediction of pork marbling scores using pattern analysis techniques. Food Control, 31, 224-229.

Jackman, P., Sun, D.-W., & Allen, P. (2009). Automatic segmentation of beef longissimus dorsi muscle and marbling by an adaptable algorithm. Meat Science, 83(2), 187-194.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International conference on multimedia (pp. 675-678). Orlando, USA.

Kvam, J., Gangsei, L.E., Kongsro, J., & Schistad-Solberg, A.H. (2018). The use of deep learning to automate the segmentarion of the skeleton from CT volume pigs. Translational Animal Science, 2(3), 324-335.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in neural information processing systems (pp. 1097-1105), Lake Tahoe, USA.

Lin, G., Milan A., Shen, C., & Reid, I. (2017). RefineNet Multi-path refinement networks for high-resolution semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Milan, Italy.

Liu, L., Ngadi, M.O., Prasher, S.O., & Gariépy, C. (2012). Objective determination of pork marbling scores using the wide line detector. Journal of Food Engineering, 110(3), 497-504.

Liu J.-H., Sun, X., Young, J.M., Bachmeier, L.A., & Newman, D.J. (2018). Predicting pork loin intramuscular fat using computer vision system. Meat Science, 143, 18-23.

Lohumi, S., Lee, S., Lee, H., Kim, M.S., Lee, W.H., & Cho, B.-K. (2016). Application of hyperspectral imaging for characterization of intramuscular fat distribution in beef. Infrared Physics & Technology, 74, 1-10.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp 3431-3440), Boston, USA.

Lorido, L., Pizarro, E., Estévez, M., & Ventanas, S. (2019). Emotional responses to the consumption of dry-cured hams by Spanish consumers: A temporal approach. Meat Science, 149, 126-133.

Morales, R., Guerrero, L., Aguiar, A.P.S, Guàrdia, M.D., & Gou, P. (2013). Factors affecting dry-cured ham acceptability. Meat Science, 95(3), 652-657.

Muñoz, I., Rubio-Celorio, M., Garcia-Gil, N., Guardia, M.D., & Fulladosa, E. (2015). Computer image analysis as a tool for classifying marbling: A case study in dry-cured ham. Journal of Food Engineering, 166, 148-155.

Pfisterer, K.J., Amelard, R., Chung, A.G., & Wong, A. (2018). A new take on measuring relative nutritional density: The feasibility of using a deep neural network to assess commercially-prepared puréed food concentrations. Journal of Food Engineering, 223, 220-235.

Qiao, Jun , Ngadi, M.O., Wang, N., Gariépy, C., & Prasher, S.O. (2007). Pork quality and marbling level assessment using a hyperspectral imaging system, Journal of Food Engineering, 83(1), 10-16.

Rangayyan, M.R. (2014). Biomedical image analysis. CRC Press.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI), 9351, 234-241.

Santos-Garcés, E., Muñoz, I., Gou, P., Garcia-Gil, N., & Fulladosa, E. (2014) Including estimated intramuscular fat content from computed tomography images improves prediction accuracy of dry-cured ham composition. Meat Science, 96(1), 943-947.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3th International Conference on Learning Representations (ICLR) (pp. 1-14). San Diego, USA.

Srivastava, S., Vaddadi, S. & Sadistap, S. (2015). Quality assessment of commercial bread samples based on breadcrumb features and freshness analysis using and ultrasonic machine vision (UVS) system. Journal of Food Measurement and Characterization, 9(4), 525-540.

Sun, D.W., & Brosnan, T. (2003). Pizza quality evaluation using computer visión- part 1. Pizza base and sauce spread. Journal of Food Engineering, 57(1), 81-89.

Velazquez, L., Cruz-Tirado, J.P., Siche, R., & Quevedo, R. (2017). An application based on the decision tree to classify the marbling of beef by hyperspectral imaging. Meat Science, 133, 43-50.

Widiyanto, S., Cufí, X., Rubio, M., Muñoz, I., Fulladosa, E., & Martí, R. (2013). Automatic intra muscular fat analysis on dry-cured ham slices. In Proceedings of ibPRIA, 873-880.

Xu, J.-L., Sun, D.W. (2018). Computer vision detection of salmon muscle gaping using convolutional neural network features. Food Analytical Methods, 11(1), 34-47.

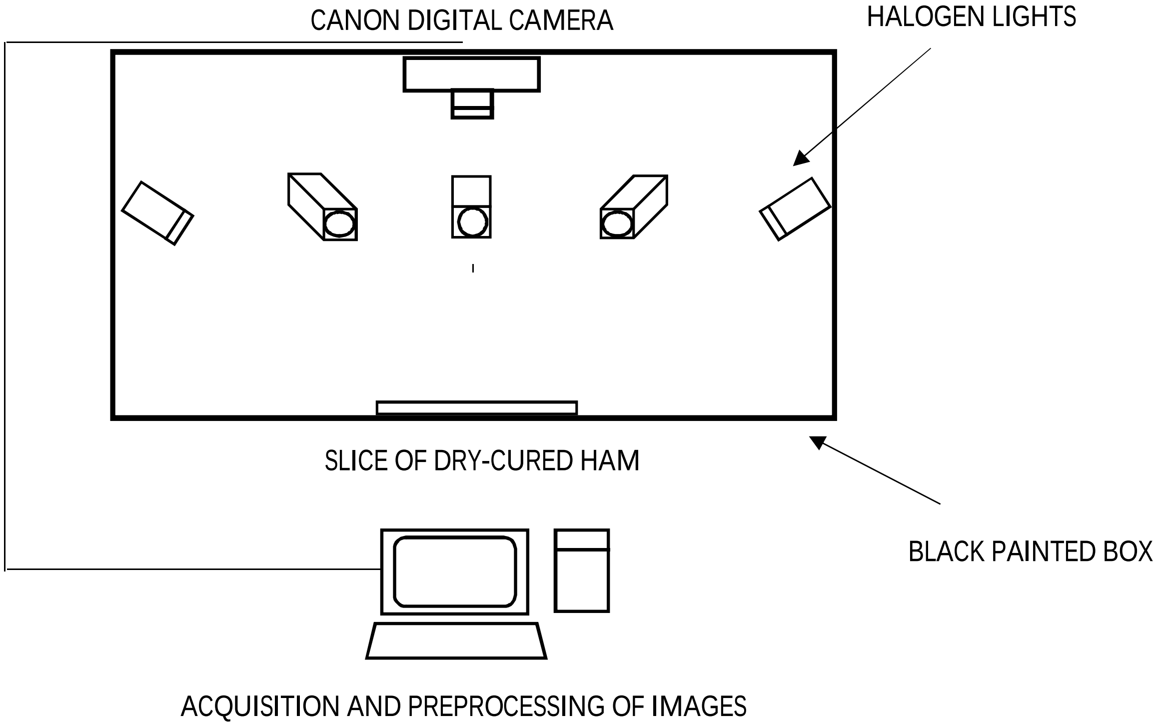


Figure 1. Overview of the image acquisition system.

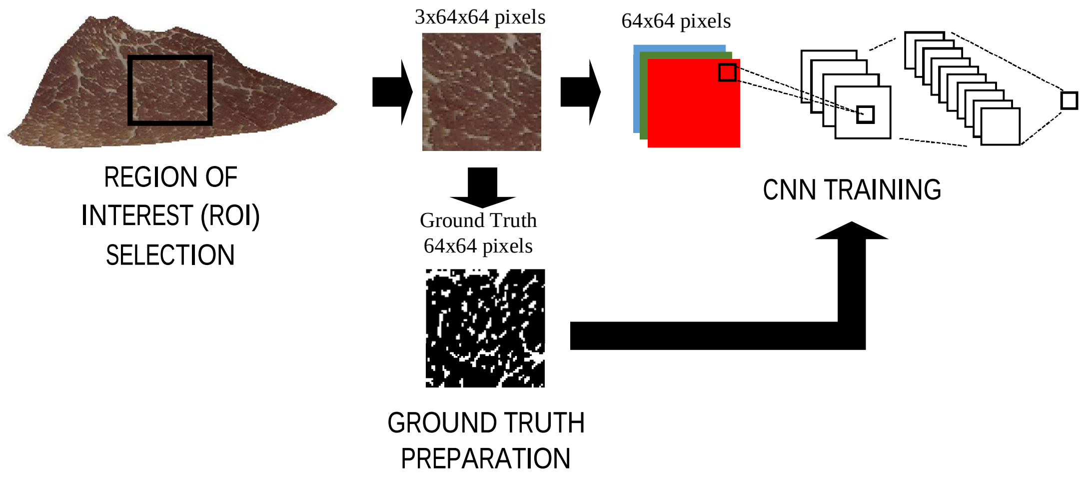
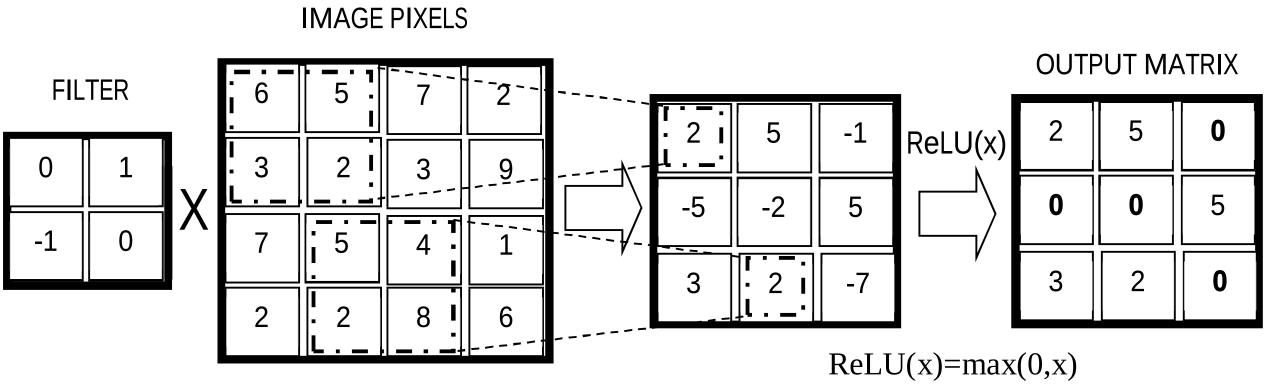


Figure 2. Overview of the learning scheme for fat segmentation using a convolutional neural network (CNN)

a)



b)

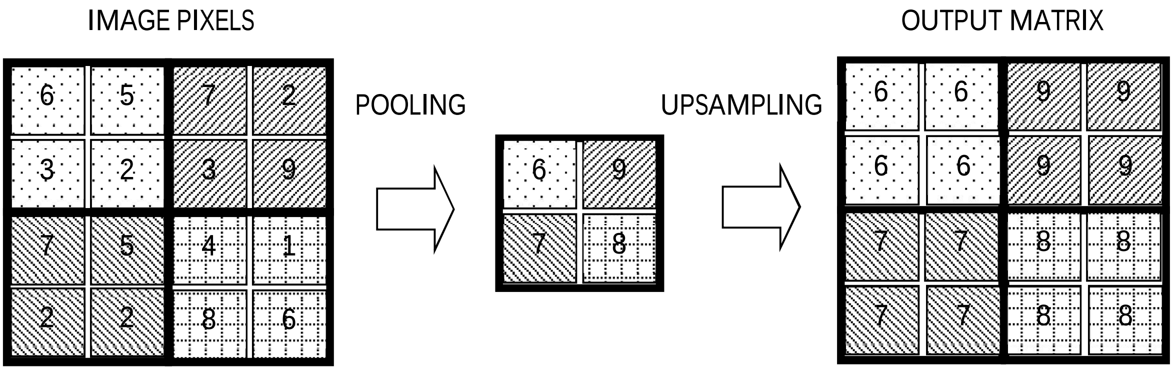


Figure 3. Main types of layers in CNNs: a) a convolution operation with a filter of 2x2 pixels and depth 1 followed by a Rectifier Linear Unit (ReLU) activation function; b) A 2x2 pixels max pooling layer followed by a nearest neighbour upsampling layer from a 2x2 to a 4x4 pixels matrix.

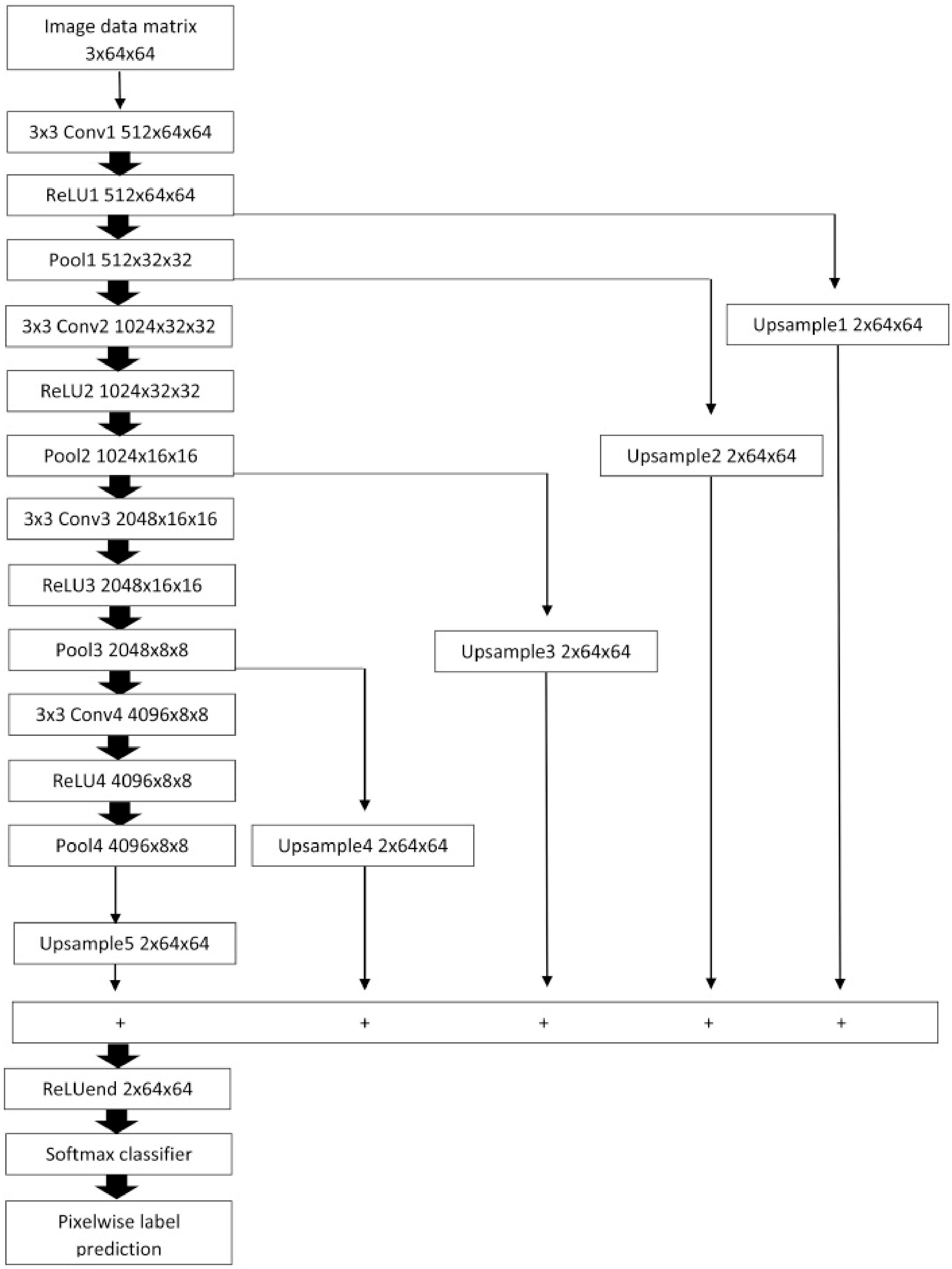


Figure 4. Architecture of the convolutional neural network architecture used in this work with 512 filters in the first layer and four layers. Convx, ReLUx, Poolx, Upsamplex indicate convolutional, rectified linear unit, pooling and upsampling operations respectively.

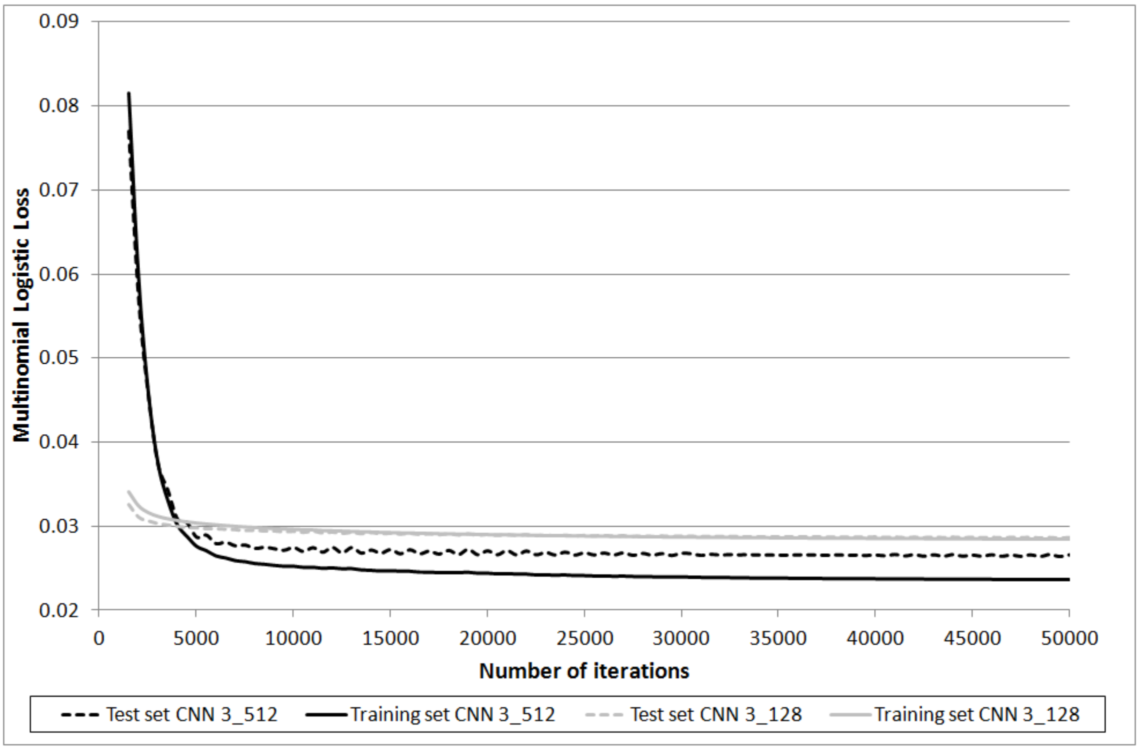
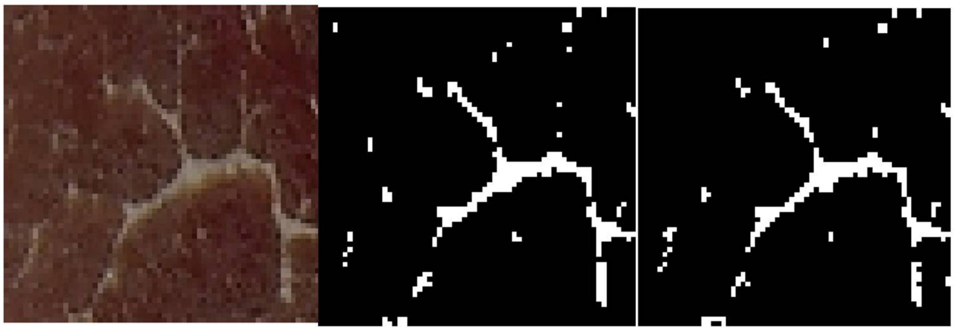
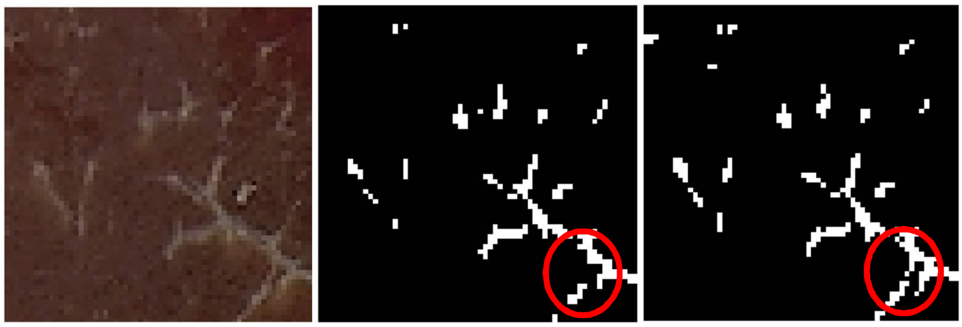


Figure 5. Multinomial Logistic Loss vs number of iterations (from 1000 to 50000 iterations) for the training and test sets of CNN 3_128 and CNN 3_512.

a)



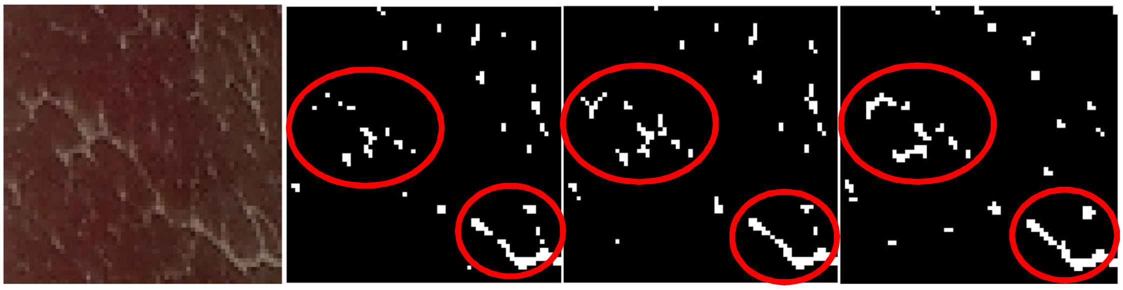
b)



c)

I II III

Figure 6. Images of slices of dry cured ham segmented with CNN 3_512. Raw image (I), segmented image with CNN 3_512 (II) and ground truth image (III). Red circle denotes areas with poor segmentation (b) and possible misclassified ground truth pixels (c)



(a) (b) (c) (d)

Figure 7. Segmentation of an image of a slice of dry cured ham. Raw image (a), segmented image by CNN 1_128 (b), segmented image by CNN 3_512 (c) and ground truth image (d). Red circles denotes areas with poor segmentation.

Architecture name	Kernel size	Number of layers	Number of filters in layers 1/2/3/4	Upsampling included	Number of learnable parameters
1_128_s	3x3	1	128	2	3,586
2_128_s	3x3	2	128	3	298,754
3_128_s	3x3	3	128	4	1,478,914
1_512_s	3x3	1	512	2	7,170
2_512_s	3x3	2	512	3	4,733,954
3_512_s	3x3	3	512	4	23,610,368
1_128	3x3	1	128	1,2	3,586
2_128	3x3	2	128/256	1,2,3	298,754
3_128	3x3	3	128/256/512	1,2,3,4	1,478,914
4_128	3x3	4	128/256/512/1024	1,2,3,4,5	6,198,530
1_512	3x3	1	512	1,2	7,170
2_512	3x3	2	512/1024	1,2,3	4,733,954
3_512	3x3	3	512/1024/2048	1,2,3,4	23,610,368
4_512	3x3	4	512/1024/2048/4096	1,2,3,4,5	99,111,938

Table 1. Description of the parameters of several CNN architectures.

Architecture	Overall pixel accuracy (training set)	Overall pixel accuracy (test set)	Fat recall rate (test set)	Fat precision rate (test set)	Rate of false negatives in the predicted fat contour (test set)	Rate of false positives in the labelled fat contour (test set)	Processing time per image (ms)
1_128_s	0.986	0.987	0.741	0.834	0.360	0.409	10
2_128_s	0.981	0.982	0.562	0.807	0.234	0.613	15
3_128_s	0.97	0.971	0.180	0.683	0.066	0.572	20
1_512_s	0.988	0.988	0.770	0.834	0.388	0.455	19
2_512_s	0.985	0.985	0.668	0.820	0.305	0.551	55
3_512_s	0.975	0.975	0.312	0.742	0.127	0.623	101
1_128	0.988	0.988	0.778	0.840	0.376	0.347	9
2_128	0.988	0.988	0.776	0.846	0.393	0.360	16
3_128	0.989	0.989	0.785	0.842	0.377	0.395	22
4_128	0.989	0.989	0.790	0.843	0.405	0.371	42
1_512	0.989	0.989	0.793	0.847	0.396	0.377	22
2_512	0.99	0.989	0.81	0.841	0.415	0.391	58
3_512	0.991	0.989	0.816	0.84	0.412	0.399	114
4_512	0.991	0.989	0.803	0.846	0.395	0.394	410

Table 2. Performance results of the studied CNN architectures