Colour Characterization and Detection of Dry Chinese Sausage Casing Twist using Colour Image Analysis

CheeYuen Lam¹, Phaklen Ehkan¹, Suwimol Jungjit²

¹Embedded, Networks and Advanced Computing (ENAC) Research Cluster, School of Computer and Communication Engineering, Universiti Malaysia Perlis, Arau, Perlis, Malaysia.
²Department of Computer and Information Technology, Faculty of Science, Thaksin University, Phatthalung, Thailand.

Corresponding author’s: phaklen@unimap.edu.my

Abstract. Chinese sausage is traditional air-dry sausage that loved by the Chinese community around the globe. This sausage can be served alone after steam cook or cook with other ingredients to create other tasty Chinese delicacies. However, a manual hand cutting process is required to cut out the sausage linking twist during the sausage’s packaging phase. This hand cutting process is tedious, time consuming and danger to workers. This study has proposed two types of sausage casing twist detector using image processing technique based on HSL and RGB colour characteristic and a set of blob detection technique to detect sausage casing twist blob on the output image. Hue, green-blue and red-blue colour characteristics are found being significant to represent the sausage casing twist in the sample images and used in the proposed algorithms. The RGB based algorithm is capable to produce a low noise image with SNR of 3.73dB as compared to Hue-based detector at -8.89dB which unsuccessful to remove shadow and object outlining of the output image. The proposed blob detection algorithm is able to detect 73.02% of the blob in hue-based image while 94.18% in RGB based image. The false detection rate in hue-based image has reached 333.33% compared to 12.17 % in RGB based image.

1. Introduction

Chinese sausage has been favorite delicacies consumed by the Chinese community. Every year, tons of Chinese sausages have been produced to fulfill the market demand. During the manufacturing process of Chinese sausage, sausage meat mixture is stuffed into the collagen sausage casing and twist linked using the sausage linking machine. Each strand of sausage casing is length around 15 meters, which can produce up to 40 pairs of sausages. After the sausage gone through drying process for a fixed period, the sausage is then proceeding to cutting process. Each of the sausage pairs required to be cut and detach from its main sausage strand for packaging process. Example of individual sausage pair is denoted by the yellow frame as shown in Figure 1(a). Currently, the cutting process is fully conducted by human operators using hand and scissor. This cutting process becomes a heavy workload when the Chinese sausages were produced in mass volume. A large number of human operators are deployed into this repeating process causing consumption of human and time resources in the manufacturing pipeline. Therefore, computerize automation is essential to level up the production performance. To automate this human based process, the sausage linking twist required to be detected by the computer and automatic cutting process can then be implemented.

The main objective of this paper is to study the colour characteristic of the sausage linking twist in HSL and RGB spaces and their corresponding sausage linking twist detection rate using proposed image
process technique and detection algorithm. The sausage linking twist is the small tiny object pointed by the green arrow (Figure 1(b)). HSL and RGB spaces are commonly used in the food processing related computer vision system.

Several studies have been carried out in food processing by using Hue space [1-3] and also RGB colour models [4-10] as the primary feature extraction colour space. However, [11] applied both Hue and RGB colour spaces in its study of beef joint moisture. Hue space has two main advantages; it is very similar to the human eye perception of colour [12] and hue angle has represented all combination colours in a single dimension scale of 360°. Alternatively, RGB colour space has the advantages of simple processing. This is because RGB does not need to perform any colour space conversion before performing image feature extraction. Several researches have applied FPGA for developing a vision-based system [13-16].

2. Methods
2.1 Image Acquisition
The input digital image is produced from an analog closed-circuit TV (CCTV) with three times manual optical zoom. The image sensor which is used in the CCTV is 1/3-inch Sony Effio couple-charged device (CCD) imaging sensor has a frame rate of 50 frames per second and generate analog phase alternating line (PAL) video signal of 700 interlaced TV line. This sensor also equipped with auto white balance function on its sensed image. Two LED white light tubes with 6000K of luminance are used for the lighting scheme and aligned in parallel side by side to CCTV camera. The output signal of CCTV which is fed to the Altera DE2 Field Programmable Gate Array (FPGA) board is converted into digital colour video stream with 640 x 480 pixels resolution and displayed on computer monitor. The sample image is then captured and stored into SD card using a 24-bit Bitmap image file writer module [17]. The experimental setup can be seen in Figure 2.

2.2 Sausage Sample Preparation
The dried Chinese sausage strands are collected from the production line of the sausage manufacturer. All sausage strands are collected from the same batch. Only the meat type sausage is analyzed in this
study. Fifty (50) samples images containing the un-cut sausage are captured to be used for image analysis process.

2.3 Image Pre-Processing

The analog PAL video signal is fed to the DE2 FPGA board and a signal conversion process takes place to convert the analog signal into digital RGB value. This conversion process is done by the pre-designed conversion architecture provided in the Altera DE2 sample project named DE2_TV. This pre-defined architecture responsible to convert incoming frame pixel data in YCbCr colour space into RGB colour space and store the converted pixel data into a VGA frame buffer for screen display. The image samples are captured from the pixel stream of VGA screen, store in SD card and are analyzed by using C code.

3. Proposed Image Processing Technique

There are two set of image processing algorithms which are based on HSL and RGB space proposed in this study. Both algorithms are aimed to filter and produce a binary image that contains only the sausage linking twist blob.

3.1 HSL-Based Algorithm

HSL-based algorithm is utilizing hue angle value of a pixel as primary data. The main reason of using hue value is that it can directly represent colour in a simple scale of 0° to 360°. Since the image samples are taken at the conveyor belt, thus the colour exists within the image is less complex. The colour of the sausage linking twist has a consistent yellowish colour, sausage body is reddish, and a conveyor belt is white colour.

This hue-based algorithm involving five process steps which begins with RGB to Hue conversion, Hue stabilizing, smoothing, thresholding and ending with erosion process. The RGB to hue conversion formula is defined in Equations (1) and (2) [18].

\[
\text{hue}_{x,y} = \begin{cases} 
60 \left(\frac{g_{x,y} - b_{x,y}}{d}\right), & r_{\text{max}} \\
60 \left(2.0 + \frac{b_{x,y} - r_{x,y}}{d}\right), & g_{\text{max}} \\
60 \left(4.0 + \frac{r_{x,y} - g_{x,y}}{d}\right), & b_{\text{max}} 
\end{cases}
\]

\[d = \max(r, g, b) - \min(r, g, b)\]  

The Aguston’s equation is applied because of its digital implementation friendly than Gonzalez’s equation [19] without any involvement of trigonometry operation. Once the hue value is calculated, it is being stabilized by a comparison operation. All high hue values range between 330° to 360° degrees of red colour are forced into range 0° to 16° degrees. This is because the red colour of the sausage body hue value overlaps from 330° to 360° and also 0° to 30° of the hue scale causing the bright and dark spot in the image.

\[
\text{hue}_{\text{smooth(s,y)}} = \frac{\text{hue}_{x-1,y} + \text{hue}_{x,y} + \text{hue}_{x+1,y}}{3}
\]

\[
\text{bin}_{\text{thres}} = \begin{cases} 
0, & \text{if } 32 < \text{hue}_{\text{smooth(s,y)}} < 72 \\
1, & \text{if other wise}
\end{cases}
\]

\[
\text{bin}_{\text{final}} = \text{bin}_{\text{thres}} \oplus E
\]
Figure 3 illustrates the Hue-based processing stages. The stabilized hue value pass through three pixels averaging operation smoothing process to smooth the image by using Equation (3). The smoothed image is fed to a band pass thresholding operator (Equation (4)) to filter only yellow colour hue range remains in the image to produce a binary image. The band pass thresholding boundary value is defined from $32^*$ to $72^*$ which is yellow colour region. The final stage is to eliminate unwanted salt and pepper noise in the threshold image by using one dimension three-pixel length erosion kernel, $E$. All the salt and pepper noises with length less than 3 pixels long are eliminated after going through Equation (5) in the final output image.

![Figure 3. Hue-based processing block diagram.](image)

### 3.2 RGB-Based Algorithm

RGB-based algorithm is simpler than Hue-based algorithm without colour space conversion process. Since the yellowish colour of sausage link is mainly generated by red and green colour components, therefore blue component has very low intensity in yellow colour. This algorithm uses green-blue (G-B) value difference of each pixel to identify the sausage linking twist region in the image as defined in Equation (6). After G-B difference image is produced, all pixels are going through an averaging process with 3-pixel kernel. It is aimed to smooth the G-B difference image. The larger of G-B difference is indicating the purer yellow colour of that particular pixel.

$$
G_B^{\text{diff}}(x, y) = \left| G_{(x, y)} - B_{(x, y)} \right|
$$

A high pass threshold operation (Equation (7)) is applied to segment the image into binary image. The threshold value is set at 46 out of 255 where this value begins to provide a significant yellow colour.

$$
G_B^{\text{thres}} = \begin{cases} 
0, & \text{if } 46 < G_B^{\text{diff}}(x, y) < 255 \\
1, & \text{otherwise}
\end{cases}
$$

$$
G_B^{\text{final}} = G_B^{\text{thres}} \circ D
$$

After segmented into binary image, one dimension 3-pixel dilation operator, $D$, is taking place along with x-direction of the image. This dilation process (Equation (8)) is intended to dilate the object blob with 1-pixel front and back respectively to preserve the blob for detection purpose as can be seen in Figure 4.

![Figure 4: G-B based processing block diagram.](image)

### 3.3 1-D Connected Pixel Detection

This study has proposed a technique to detect and locate the sausage linking twist blob in binary image. The initial step of this technique is to detect the connected blob pixel in 1-dimensional manner and
extract its starting and ending x coordinates for localization purpose. 1-D connected pixel detector has a 2-pixels length window denoted as C1 and C2. It is required to scan through row by row horizontally in the ROI. This detection technique has a \textit{length\_limit} variable. This variable determines the minimum length of the connected pixel to be qualified for extraction. Those connected pixels length shorter than the limit will be discarded. The operation mechanism of the connected pixel detector is described in a flow diagram as revealed in Figure 5. The extracted start, middle and end x location of the detected connected pixel is stored as a list in a text file. This file is then passed to the blob localization process for further summarizing the blob validity.

![Figure 5. Connected pixel detection mechanism.](image)

### 3.4 Blob Localization Technique

Blob localization process is to analyze blob coordinate based on the coordinate information from the detected connected pixel list and it serves two purposes. First is to group the detection of connected pixel based on their coordinate information, and second is to filter false blob and true object blob according to vote value. A generic blob descriptor need to be defined in order to conduct localization process. The blob descriptor contains of coordinate parameters to describe the current possible blob. The parameters and their descriptions are listed in Table 1.

The general localization process flow is loading the first detection of connected pixel train entry in the list and set as a new blob by occupying a blob descriptor. Then, the second pixel train entry from the list is read, and its middle point is used for comparing its fall within the rectangular boundary. If the middle point is falling within boundary, then this entry is belonging to this blob descriptor and vote value in this blob descriptor is incremented by 1. Then, next entry is read. If the next entry does not fall within boundary, then this entry will occupy a new blob descriptor and forming a new blob. Each entry from the list is passed through every blob descriptor for matching until they find a most suitable blob descriptor or they form a new blob at last. The parameters in the blob descriptor are updated for every successive entry. Once all the entries in the list are processed, a classification is performed to classify the blob descriptor into true and false blobs. A minimum limit is used to evaluate the vote value in the blob descriptor. The blob descriptor with the highest vote value and surpassing the limit will be taken as true blob; otherwise it will be discarded in the classification process.

### Table 1. Blob descriptor parameters.

| Parameters     | Description                                      |
|----------------|--------------------------------------------------|
| Current\_ref\_x | Current x location used to calculate rectangular window |
| Current\_ref\_y | Current y location used to calculate rectangular window |
| Avg\_start\_x  | Running average starting x location for successive grouped entry |
| Avg\_end\_x    | Running average end x location for successive grouped entry |
| Avg\_mid\_x    | Running average middle x location for successive grouped entry |
| Window\_length | Constant value of ½ total rectangle length         |
| Window\_width  | Constant value of ½ total rectangle width          |
| Vote           | Number of total successive entry grouped belongs this blob |
4. Results and Discussion

4.1 Colour Character Analysis

There are objects exist in the sample images such as conveyor surface, sausage body, sausage casing twist, shadow, and overlap shadow. Due to the illuminant configuration of the experimental setup, shadow has casted on the surface of the conveyor belt. The category of this shadow is belonging to penumbra. An overlapped penumbra shadow has casted when there are two sausage bodies are distancing apart in a certain distance. This analysis is to analyze the colour characteristic of those mentioned object in the sample images. Multiples patch of pixels from every object component in the image are analyzed in terms of individual RGB intensities, RGB component intensities difference and individual HSL intensities. The colour characteristics are helpful for determination of threshold limit in the proposed algorithm. The hue value is normalized to 255 as it required fitting into the 8-bits gray scale bitmap image.

The main object of interest in this study is the sausage casing twist. The colour characteristic of sausage casing twist is revealed in Table 2. The red and green components have higher intensities than blue component to form yellow colour. By comparing individual RGB and intensities difference of RGB, it is found that the intensities difference of RGB has more significant representation of sausage casing twist compared to another object in the image. The RGB intensities of sausage body are similar to the casing twist. However, R-B and G-B intensities difference are significant among other objects with the difference as high as 68. This value is selected to implement as threshold limit in the RGB algorithm.

Besides, hue value for sausage casing twist also lies within the yellow hue range with normalized mean hue of 42° or 59° in full scale. The saturation and luminance of the sausage casing twist do not show any significant representation as they overlap with other objects. Therefore, threshold limit for hue-based algorithm are set based on hue value only.

| Colour Component | Conveyor Mean | Sausage Body Mean | Sausage Casing Twist Mean | Penumbra Mean | Overlapped Penumbra Mean |
|------------------|---------------|-------------------|---------------------------|--------------|-------------------------|
|                  | Mean | σ  |   Mean | σ  |   Mean | σ  |   Mean | σ  |   Mean | σ  |
| R                | 251  | 2.9 | 152   | 22.9 | 185   | 22.4 | 218   | 10.6 | 107   | 11.0 |
| G                | 254  | 0.9 | 123   | 26.5 | 185   | 21.1 | 255   | 1.6  | 98    | 10.3 |
| B                | 255  | 1.2 | 110   | 23.3 | 117   | 23.0 | 222   | 13.6 | 86    | 10.7 |
| R-G              | 3    | 3.5 | 28    | 8.0  | 6     | 4.2  | 37    | 10.0 | 9     | 2.4  |
| R-B              | 4    | 2.4 | 42    | 5.2  | 69    | 11.4 | 10    | 6.5  | 21    | 2.6  |
| G-B              | 2    | 1.2 | 14    | 7.2  | 68    | 8.1  | 32    | 13.3 | 12    | 2.9  |
| Hue              | 148  | 18.1 | 13   | 7.7  | 42    | 5.4  | 91    | 12.7 | 24    | 4.8  |
| Saturation       | 100  | 0.0 | 19    | 3.4  | 37    | 10.5 | 99    | 4.8  | 10    | 1.8  |
| Luminance        | 98   | 4.5 | 51    | 8.8  | 59    | 8.5  | 91    | 4.8  | 37    | 4.3  |

4.2 Proposed Algorithm Output

The full colour sample images of sausage linking twist is processed by both algorithms. It is noticeable that many bright spots appear in the sausage body region of the hue image. These bright spots are caused by the high hue value of red colour in the range of 330° to 360°. The horizontal line appeared in the output image is artificial line created as boundary marker (Figure 6). Comparing both processed of output image, it is obvious that G-B based algorithm produces a much better-quality image than hue algorithm. The output image from hue algorithm is still containing a small fraction of shadow in between the sausage body and it also failed to filter out the sausage body outlining. This remaining noise will affect the blob detection rate during detection process.
Figure 6. Example of sample images.

4.3 Image Signal-Noise-Ratio Analysis

The performance for both proposed algorithm is evaluated based on the image SNR as shown in Figure 7. The sausage linking twist region in the image is first identified by human operator thru drawing a blue rectangle bounding the region. These selected regions are considered as sausage linking twist region and used as reference to classify each pixel on the output image from both algorithms. The pixel in the image is classified according to the criteria in Table 3. The white pixel is ignored in classification process because it was treated as background pixel.

Figure 7. Image filtering result: (a) hue image (b) stabilized hue image (c) threshold and eroded hue image (d) G-B difference image of input image (e) threshold G-B difference image and (f) dilated difference image.
Table 3. SNR evaluation pixel classification.

| Pixel Classes     | Denoted Pixel colour | Description                                      |
|-------------------|----------------------|--------------------------------------------------|
| Input pixel       | Black                | Appear in input image                            |
| True valid pixel  | Green                | Locate within human define region                |
| Noise pixel       | Red                  | Locate beyond human define region                |

When an input pixel is located within the rectangle region which defined by human operator, it is classed as true positive pixel; otherwise the pixel is classed as noise pixel. This classification process is conducted on 50 samples of image. The number of pixel of each class is summed up. The SNR is calculated to benchmark their filtering performance for both architectures. The SNR is calculated based on Equation (9).

\[
SNR(db) = 10 \log_{10} \frac{V}{N},
\]

where, \(V\) is total number of true valid pixel, and \(N\) is total number noise pixel in the particular image. Based on these classification conditions, the numbers of valid data pixel and noise pixel are obtained, and SNR of the output image is determined as shown in Figure 8.

Based on the results revealed in Figure 9, hue-based filtering architecture produces noisy image while G-B based filtering produces a clearer image for the sausage linking twist detection. The average SNR of the hue-based filtering and G-B based filtering is -8.89 dB and 3.73 dB, respectively (as shown in Table 4).

Table 4. Averaged SNR.

| Method     | Average SNR |
|------------|-------------|
| Hue-based  | -8.89 dB    |
| G-B based  | 3.73 dB     |
Figure 9. SNR comparisons between sample images of hue and G-B based algorithm.

4.4 Sausage Casing Twist Blob Detection Rate
Since the output image quality from hue-based and G-B based filtering method are evaluated, these sample images are used to perform blob detection rate analysis. This is to determine the blob detection rate among the sample images from both proposed algorithms. The relationship of connected pixels limit and vote value limit from connected pixel detector and blob descriptor also are investigated in this analysis.

Hypothetically, if the minimum connected pixel limit is set to low; most of the information in the image is preserved but also preserving noise pixel in the image. If this value is set higher, noise can be filtered including some useful information in the image may also filtered and ignored. The same hypothesis relation is also applied to minimum vote value. Therefore, this test is to prove this hypothesis and to observe the blob detection rate under the influence of these two parameters and the types of output image.

The minimum connected pixels limit is beginning from 5 pixels to 21 pixels and minimum vote value is beginning from 9 to 21 with increment of 2 for each step. A total of 189 sausage linking twist existed in the 50 sample images. At starting of test, the processed output images from both algorithms from section 4.2 are processed by the connected pixel detection and blob localization algorithms. The center point of the detected blob is estimated in the algorithm and is then used to match with human define sausage casing twist mask region (Figure 8(b)) to verify blob detection accuracy. Four criteria are analyzed in this test; true blob detection rate, false blob detection rate, undetected blob rate and duplicate blob detection rate.

i) True Blob Detection Rate
As compare the results from Figure 10, the best achieved detection rate for hue-based image is only 73.02 % when vote limit is 9, with minimum 7 connected pixels. The detection rate achieved for G-B based image is 94.18% with minimum 9 vote and 5 connected pixels limit. The detection rate for both types of image has declined when these two parameters incremented. The detection process is totally failed when these parameters reach a value of 21. The minimum vote value and minimum connected pixel are having a concurrent relation to each other. Both values required to be adjusted at the same rate. If both parameter values are at reversed order and vice versa, the blob detection will have failed.
ii) False Blob Detection Rate
The hue-based image has a severe false blob detection rate which up to 333.33 % on minimum vote of 9 and minimum 5 connected pixels (Figure 11). At the same parameter values, false detection rate has 5.3 times higher than true blob detection. This shown that the noise blob in hue image is far more than the sausage linking twist blob. On the other hand, the false detection rate for G-B based image is only 12.17 %. When both parameters are incremented accordingly, the false detection rate is getting less for both types of image. At the condition when both parameters are at a value of 21, the false rate is dropping to 0.53 % and 0.00 % for hue and G-B based image, respectively.

iii) Undetected Blob Rate
The undetected rate is showing an inversed pattern as compared to true detection rate (Figure 12). The lowest undetected rate is at 39.68 % and 7.41 % when both parameters at minima value for hue and GB based image. When both values are adjusted into 21, most of the sausage linking twist is missed out from detection with rate 97.35 % and 99.47 % for hue and GB based image accordingly. This shown that once the vote value and connected pixels limit are getting too large, the blob has slip through the oversized detection window.

![Figure 10. True blob detection rate.](image)

![Figure 11. False blob detection rate.](image)

![Figure 12. Undetected blob rate.](image)
iv) Duplicate blob detection rate
Figure 13 shows a detection rate of duplicate phenomenon occurs during this analysis. This occurs when the vote value at 9 and minimum 5 connected pixels. Some of the sausage linking twist blob with larger blob area has been detected twice when using narrow window parameters. As comparing both duplicate rates, it is found that duplicate detection occurs until range with 7 connected pixels limit and also minimum 13 votes limit. This cover larger condition than G-B based image.

![Duplicate Detection Rate](image1)

![Duplicate Detection Rate](image2)

(a) Hue-based image
(b) G-B based image

**Figure 13.** Duplicate blob detection rate.

The parameters of minimum vote limit and minimum connected pixels limit has justified being suitable at 15 votes and 5 connected pixels. There are two considerations when selecting these values. First consideration is low value zone must have avoided to prevent duplicate detection occurs. Second consideration is these values must not set too large to prevent miss detection occurs. Therefore, the connected pixels limit is justified at 5 pixels in the connected pixel detector so that most of the information in the image is preserved. Then, a minimum vote value is justified at 15 votes in the blob localization process to avoid abundance of noise blob to be classified as object blob since most of the noise blob is detected in lower vote limit value range of 9 to 13.

5. Conclusion and Future Work
The colour characterization of the sausage casing twist is successfully being carried out. The G-B and R-B difference values carried the significant representation attributes to sausage casing twist colour in the image among other colour characteristics. The proposed RGB algorithm is suitable to be used to detect sausage casing twist in digital image with the output image SNR merit of 3.73dB while hue base algorithm has failed to filter noise causing output image SNR of -8.89dB. Hue-based algorithm is not suitable used to process the sausage casing twist due to too many noises remaining in the image. The proposed connected pixel detector and blob localization algorithm is effectively developed and tested. These algorithms are successfully detecting the blob in hue-based image at the rate of 73.02% while 94.18% for G-B based images. The minimum connected pixels length and vote value are justified at 5 pixels and 15 votes being suitable to be used for blob detection of the image.

The performance of the study can be further improved in detection rate such as a concurrent image processing algorithm can be explored for better image filtering accuracy, and high level statistical classification algorithm can be applied to improve detection of sausage casing twist.

References
[1] S. Nashat, A. Abdullah, S. Arumvith, M.Z. Abdullah, “Support Vector Machine Approach to Real-time Inspection of Biscuits on Moving Conveyor Belt”, Computers and Electronics in Agriculture, 2011, 75 (1), pp. 147-158.
[2] Domingo Mery, Jorge J. Chanona-Pérez, Alvaro Soto, José Miguel Aguilera, Aldo Cipriano, Nayeli Veléz-Rivera, Israel Arzate-Vázquez, Gustavo F. Gutiérrez-López, “Quality Classification of Corn Tortillas Using Computer Vision”, Journal of Food Engineering, 2010, 101(4), pp. 357-364.
[3] Xin Sun, Jennifer Young, Jeng Hung Liu, Laura Bachmeier, Rose Marie Somers, Kun Jie Chen, David Newman, “Prediction of Pork Colour Attributes using Computer Vision System”, Meat Science, 2016, 113, pp. 62-64.

[4] A. Manickavasagan, N.K. Al-Meziene, H.N. Al-Shekaili, “RGB Colour Imaging Technique For Grading of Dates”, Scientia Horticulture, 2014, 175, pp. 87-94.

[5] J. Blasco, N. Aleixos, S. Cubero, J. Gómez-Sanchis, E. Moltó, “Automatic Sorting of Satsuma (Citrus Unshiu) Segments Using Computer Vision and Morphological Features”, Computers and Electronics in Agriculture, 2009, 66 (1), pp. 1-8

[6] J. Blasco, S. Cubero, J. Gómez-Sanchis, P. Mira, E. Moltó, “Development of a Machine for the Automatic Sorting of Pomegranate (Punica Granatum) Arils Based on Computer Vision”, Journal of Food Engineering, 2009, 90(1), pp. 27-34.

[7] Muhammad Makky, Peeyush Soni, “Development of An Automatic Grading Machine for Oil Palm Fresh Fruits Bunches (FFBs) Based on Machine Vision”, Computers and Electronics in Agriculture, 2013, 93, pp.129-139.

[8] Nasser Behroozi Khazaei, Teymour Tavakoli, Hassan Ghassemian, Mohammad Hadi Khoshtaghaza, Ahmad Banakar, “Applied Machine Vision and Artificial Neural Network for Modeling and Controlling of The Grape Drying Process”, Computers and Electronics in Agriculture, 2013, 98, pp. 205-213.

[9] Mehrdad Baigvand, Ahmad Banakar, Saeed Minaei, Jalal Khodaei, Nasser Behroozi-Khazaei, “Machine Vision System for Grading of Dried Figures”, Computers and Electronics in Agriculture, 2015, 119, pp. 158-165.

[10] Israel Muñoz, Marc Rubio-Celorio, Núria Garcia-Gil, Maria Dolors Guàrdia, Elena Fulladosa, “Computer Image Analysis as a Tool for Classifying Marbling: A Case Study in Dry-cured Ham”, Journal of Food Engineering, 2015, 166, pp.148-155.

[11] Chaoxin Zheng, Da-Wen Sun, Liyun Zheng, “Correlating Colour To Moisture Content of Large Cooked Beef Joints by Computer Vision”, Journal of Food Engineering, 2006, 77(4), pp. 858-863

[12] Camstra, F., Vinciarelli, A. (Eds.), 2008. “Machine Learning for Audio Image and Video: Analysis Theory and Applications”, Springer–Verlag, London.

[13] Scharf, D., Le Viet, B., Le, T.B.H., Vandone, A., Giardini, M., “Hardware Accelerated Image Processing on an FPGA-SoC Based Vision System for Closed Loop Monitoring and Additive Manufacturing Process Control”, 2019, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11754 LNCS, pp. 3-12.

[14] Yu, X., Li, B., Dong, M., Yin, W., ”Target Detection and Tracking System Based on FPGA” 2020, IOP Conference Series: Materials Science and Engineering 793(1),012008.

[15] Srinivasa Rao, P., Yedukondalu, K., “Hardware Implementation of Digital Image Skeletonization Algorithm using FPGA for Computer Vision Applications”, 2019, Journal of Visual Communication and Image Representation, 59, pp. 140-149.

[16] Lentaris, G., Stratakos, I., Stamoulias, I., Lourakis, M., Zabulis, X., “High-Performance Vision-Based Navigation on SoC FPGA for Spacecraft Proximity Operations”, 2020, IEEE Transactions on Circuits and Systems for Video Technology, 30(4), pp. 1188-1202.

[17] Chee Yuen Lam, Voon Siew Soon, Phaklen Ehkan, “Reconfigurable Tri-Colour RGB SD Card Bitmap Image File Writer”, 2015, Lecture Notes in Electrical Engineering, Springer International Publishing, Switzerland, 362, pp. 1229-1236.

[18] Max K.Agoston, “Computer Graphic and Geometric – Implementation and Algorithms”. Springer, London 2005.

[19] Gonzalez, R.C., Woods, R.E. (Eds.), “Digital Image Processing”, Pearson Prentice Hall, Reading, New Jersey, USA, 2008.