

An Embedded Computer Vision System for Beans Quality Inspection

Robson A.G. Macedo
Informatics and Knowledge
Management Graduate Program,
Nove de Julho University, São
Paulo, Brazil

Peterson A. Belan
Informatics and Knowledge
Management Graduate Program,
Nove de Julho University, São
Paulo, Brazil

Sidnei A. Araújo
Informatics and Knowledge
Management Graduate Program,
Nove de Julho University, São
Paulo, Brazil

ABSTRACT

Despite the importance of agricultural grains appearance for their choice by the consumers as well as for determining their selling price, the visual inspection of the quality of these products is usually conducted in a manual way and, therefore, susceptible to high operational costs, human errors and inaccurate results. Recently, a computer vision system for quality inspection of beans composed by a set of hardware and software, named the SIVQUAF, was proposed in the literature. However, the software of the SIVQUAF was designed for a personal computer, which makes its operation more complex, decreases its performance and raises the cost of the equipment. Thus, in this work we explored the customization and optimization of SIVQUAF aiming its running on a Raspberry Pi 3, keeping similar performance, generating the SIVQUAF-Compact. Besides redesigning and parallelizing algorithms, we proposed improvements in the classification and defect detection steps, and a new touch-sensitive interface. The experiments conducted with SIVQUAF-Compact embedded in a Raspberry Pi 3 demonstrated that in addition to reproducing high hit rates in the tasks of segmentation (97.50%), classification (97.06%) and detection of defects (74.78%), there was a significant gain in terms of cost, operation and compaction of the equipment, increasing its operational, technical and economic viability.

Keywords

Embedded System; Raspberry; Computer Vision; Visual Inspection; Bean

1. INTRODUCTION

The largest producers of beans in the world are Myanmar, India, Brazil, the United States of America (USA), Mexico, and Tanzania, respectively, which are responsible for approximately 60% of all global production of the grain [1]. Brazil is the world's largest consumer and producer of common beans (*Phaseolus vulgaris* L.). No wonder, bean and rice form the Brazilian basis diet.

One question of extreme importance in the commercialization of agricultural grains, both in the acquisition and the sale is the visual quality, which is evaluated from properties such as color, shape, and size. These characteristics generally impact on product's market price. Thus, the visual quality inspection is a task of great importance for most agricultural products, including beans [2–4]. However, it is very common that these tasks occur manually, demanding a lot of time, increasing operational costs and difficulting the standardization of results [5,6]. In this context, the use of computer vision systems (CVS) aiming at the automation of these tasks can bring a competitive advantage for companies. Actually, computer

vision can be considered as one of the keys for achieving industry 4.0 [7].

In the last decade, several works have been presented in the literature proposing the development of CVS for inspection the visual quality of agricultural grains, such as the works presented in Table 1. In such systems, the first, and most critical step is the segmentation of grains, since the classification and detection of defects depend on the success of this step.

Despite the reported high success rates, only the most recent works (since 2015) present robust approaches for segmentation of touching grains. This was a severe limitation that hampered the practical application of some proposed systems. In addition, one can see that there are only a few works addressing the detection of defects in grains. It is also rare works proposing specific equipment for visual inspection of grains. In our literature review we found the works of OuYang *et al.* (2010) [8] and Zareiforush *et al.* (2016) [9] proposing equipment for the inspection of rice quality.

Regarding beans, only Belan (2019) [10] proposed a CVS composed by set of hardware (equipment) and software, named SIVQUAF, to inspect the visual quality of the product. However, the software that composes the SIVQUAF must be run on a personal computer, which compromises its usability and performance, since the various tasks performed on a general use computer compete by the same hardware resources. Besides that, the need for a personal computer to control the equipment increases the cost of the proposed CVS, compromising its operational and economic viability.

It should also be noted that from all the investigated works, only Araújo, Pessota and Kim (2015); Belan, Araújo & Santana (2015); Belan *et al.* (2016); Belan *et al.* (2018) and Belan (2019) report concern with the processing time to make the use of the proposed systems in practical applications feasible and, possibly, their running on low cost embedded platforms.

It is in this context that the present work is inserted, with the proposal to customize, optimize and improve some steps of the SIVQUAF, making feasible its running on a Raspberry Pi 3, which consists of a versatile and low cost embedded platform with limited hardware resources.

Table 1: Works addressing CVS for quality inspection of agricultural grains found in the literature (from 2009 to 2019)

Reference	Ref.	Inspected grain(s)	Executed task(s)	Results	Segments touching grains?	Employs embedded technology?	Employs Equipment (E) / Apparatus (A) / only Camera (C) / Others (O)
Qing <i>et al.</i> (2009)	[11]	Rice	Defects detection	Precision: 99.5%	Yes	No	A
Venora <i>et al.</i> (2009)	[12]	Italian beans	Classification	Precision: 98.49%	No	No	A
Anmi & Savakar (2010)	[13]	Wheat, peas, peanuts and others	Classification	–	No	No	A
Aggarwal & Mohan (2010)	[2]	Rice	Classification	Precision: 90.00%	No	No	C
Laurent <i>et al.</i> (2010)	[14]	Beans	Classification	Not informed in terms of rates	No	No	A
OuYang <i>et al.</i> (2010)	[8]	Rice	Classification of 5 varieties	Success rates: 99.99%. 99.93%. 98.89%. 82.82% e 86.65%	No	No	E
Liu et al (2011)	[15]	Soja	Classification	Precision: 97.00%	No	No	A
Patil, Yadahalli & Pujuari (2011)	[5]	Corn, soy, lentils, sorghum, wheat and others	Classification	–	No	No	C
Pessota (2013)	[16]	Beans	Classification	Average success rate: 95.00%	Yes	No	A
Siddagangappa & Kulkarni (2014)	[17]	Rice, beans and others	Classification	Success rates: 98.00%	No	No	A
Kambo e Yerpude (2014)	[18]	Indian Rice	Classification	Accuracy: 80%. 75% e 80%	No	No	A
Dubosclard <i>et al.</i> (2014, 2015a, 2015b)	[19–21]	Rice	Classification	Time: 90s to classify an image	Yes	No	A
Potter <i>et al.</i> (2015)	[22]	Corn	Classification	Success rates: 81.00 a 89.00%	No	No	A
Belan, Araújo & Santana (2015)	[23]	Brazilian Beans	Classification	Success rates: 99.95%	No	No	A
Araújo, Pessota e Kim (2015)	[24]	Brazilian Beans	Classification	Success rate: 99.99%	Yes	No	A
Araújo <i>et al.</i> (2015)	[25]	Brazilian Beans	Classification	Success rate: 99.95%	Yes	No	A
Belan, Araújo & Alves(2016)	[26]	Brazilian Beans	Classification	Success rate: 99.14%	Yes	No	A
Belan <i>et al.</i> (2016)	[27]	Brazilian Beans	Classification	Success rate: 99.48%	Yes	No	A
Zareiforoush <i>et al.</i> (2016)	[9]	Rice	Defects detection	Precision: 98.72%	Yes	No	E
Ramos <i>et al.</i> (2017)	[29]	Coffee bean	Classification	Precision: 95.00%	Yes	Yes	O (Smartphone)
Bhat, Panat & Arunachalam (2017)	[30]	Rice	Classification	–	Yes	No	A
Belan <i>et al.</i> (2018)	[31]	Beans	Only segmentation	Accuracy: 86.20%	Yes	No	O (Database of images)
Belan (2019)	[10]	Brazilian Beans	Classification and defects detection	Average success rate: 99.00%	Yes	No	E

2. THEORETICAL BACKGROUND

2.1 Visual inspection of Brazilian bean grains quality

The rules established by the Brazilian Ministry of Agriculture, Livestock and Supply (BMALS) determine the classification of Brazilian beans in Group, Class, and Type. The Group refers to the botanical species, being the beans classified in Group I when belonging to the species *Phaseolus vulgaris* L. (common bean), or in Group II when belonging to the species *Vigna unguiculata* L. Walp. The Class depends on the color of the skins of the grains contained in the inspected sample

(Black, White, Colored, or Mixed), regardless of the group. The Type, on the other hand, is related to the defects found, being the main ones: broken, burnt, moldy, bored, germinated, impurities, and foreign matter [32].

The quality inspection of beans is carried out manually and consists of extracting at least a sample of 1 Kg from a lot of beans, from which two subsamples of at least 250g are obtained for visual inspection and chemical inspection. From the subsample for visual inspection, foreign matter and impurities are separated using a sieve with holes of 5mm of diameter, to then determine the Group, Class and Type, according to the reference values established by BMALS [32].

The main problems associated with the current bean quality inspection process are the high probability of errors, the cost of the process and the difficulty in standardizing the results.

2.2 SIVQUAF

The SIVQUAF proposed by Belan (2019) consists of a set of software and hardware (equipment), which are briefly described in sections 2.2.1 and 2.2.2.

2.2.1 Software

The software, composed by an interface (Figure 1) and image processing algorithms, was developed in C/C++ language using the Microsoft® *Visual Studio*™ 2012 platform and the OpenCV 2.4 library [33]. Its working, including all processing steps briefly described in the subsections 2.2.1.1 to 2.2.1.5, is illustrated in the diagram of Figure 2. Details about computer vision and artificial intelligence techniques employed in the development of the SIVQUAF can be found in Belan (2019).

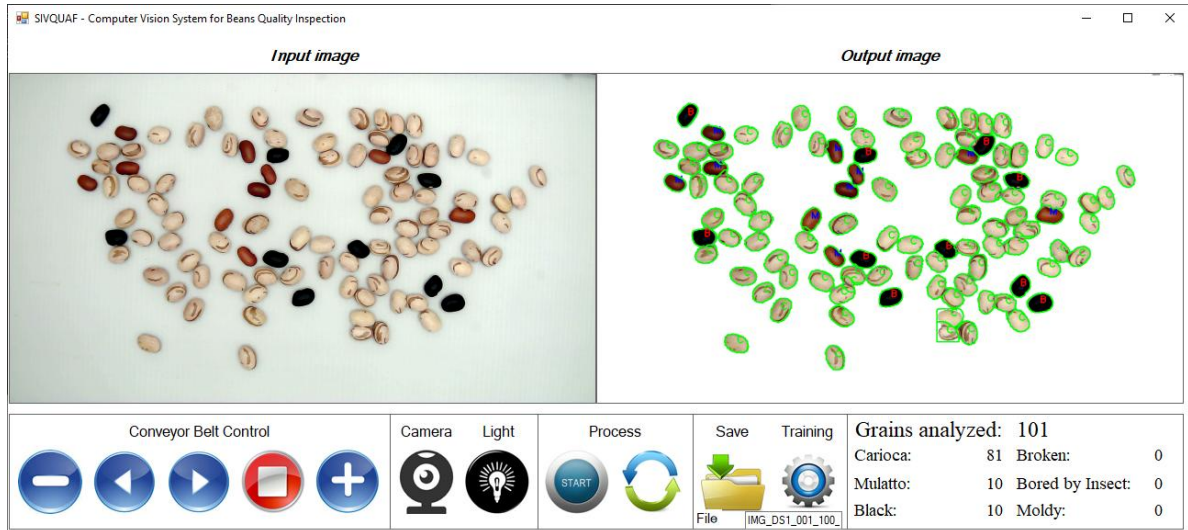


Fig 1: Interface for the SIVQUAF operation. Adapted from Belan (2019)

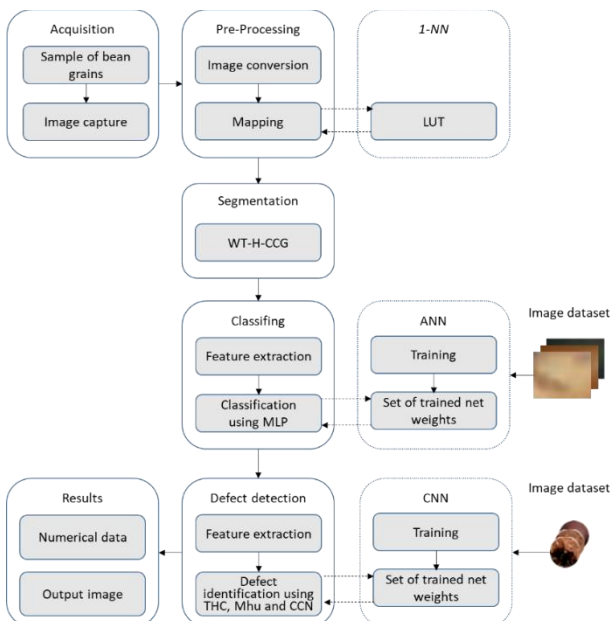


Fig 2: Diagram of working of the SIVQUAF software. Adapted from Belan (2019)

2.2.1.1 Acquisition

The input of the software is an RGB color image (Figure 4a), acquired in the image acquisition chamber, which is processed by the steps described below.

2.2.1.2 Pre-processing

The pre-processing step employs a method based on the k-NN algorithm (k-nearest neighbors), with k=1 (that is, 1-NN), which maps each pixel of the input image (Figure 4a) for a grayscale value, being the typical colors of beans mapped to darker tones and typical background colors mapped to clearer

tones. However, instead of conducting training for each image as proposed in Araújo, Pessota & Kim (2015), the mapping is done in an efficient way using a previously created *lookup table* (LUT) [10].

2.2.1.3 Segmentation

The segmentation process consists of isolating and each grain in the image, even in the case of touching grains, so it's possible to make an individualized analysis. In the SIVQUAF the algorithm for segmentation, called WT-H-CCG, uses the Watershed Transform (WT), heuristics (H) for joining (J) and/or separation (S) connected components and Cross-Correlation Granulometry (CCG).

The working of this algorithm can be described as follows: first, the pre-processed image (Figure 4b) is segmented by applying WT. Since the segmentation by WT is susceptible to errors, heuristics based on information of grains such as average and the distance between their centers, are applied aiming at the junction (J) or separation (S) of two or more segmented connected components (CCs) erroneously grouped or divided. As the use of heuristics cannot fix all segmentation errors, the CCG algorithm is used only in the cases where the application of heuristics was not effective. For the application of the CCG, 72 templates were used, considering 4 scales and 18 rotation angles for each scale (4×18).

Figure 4b illustrates cases where joints (J) of small connected components and CCG were applied, while Figure 4c illustrates a part of the input image with the grains segmented.

2.2.1.4 Classification

The grains classification consists in labeling each segmented CC to one of the three classes considered in this work (Carioca – C, Mulatto – M or Black – B). To this end, from ten radial projections of each grain, average values from the three bands of the RGB color space and the three bands of the

CieLab space are extracted, that is $\{\bar{R}, \bar{G}, \bar{B}, \bar{L}, \bar{a}, \bar{b}\}$. Thus, as illustrated in Figure 3, a Multi-Layer Perceptron artificial neural network (MLP-ANN) maps each grain for a $Class_{bean} \in \{C, M, P\}$.

The employed MLP-ANN was configured with the following architecture: 6 neurons in the input layer, two hidden layers with 30 and 45 neurons, and three neurons in the output layer to indicate the grain class.

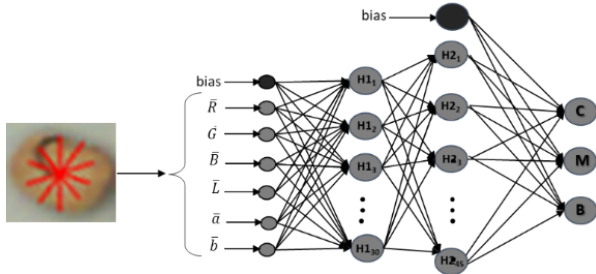


Fig 3: RNA-MLP employed in the SIVQUAF' software for grains classification. Belan (2019)

It should be noted that this method of extracting features can be susceptible to errors since there are many occurrences of dark brown streaks covering a considerable area of carioca

beans (grains of predominantly light brown skin) that can make the MLP-ANN confuse them with mulatto (grains of skin predominantly dark brown). Part of a classified image is shown in Figure 4d.

2.2.1.5 Defects detection

a) Broken grains

A whole bean has an almost elliptical shape. However, as a consequence of mechanical impact on the harvest or due to another factor in the inspection process, some grains may be broken. The algorithm for detection broken grain employs the seven invariant moments of Hu (MHu) that compose the signature of a grain, which is compared with the signature extracted from the image of an ellipse or from an image of a healthy grain considered as standard.

An example of identification of this type of defect is shown in Figure 4e. It is worth mentioning that Belan (2019) tested other algorithms for this task and, according to him, but the use of MHu presented the best cost-benefit.

b) Bored grains

The bored grains, such as those illustrated in Figure 4e, are grains having one or more holes caused by woodworm *Acantoscelides obtectus*, whose attack seriously compromises the appearance, conservation, and quality of the beans.

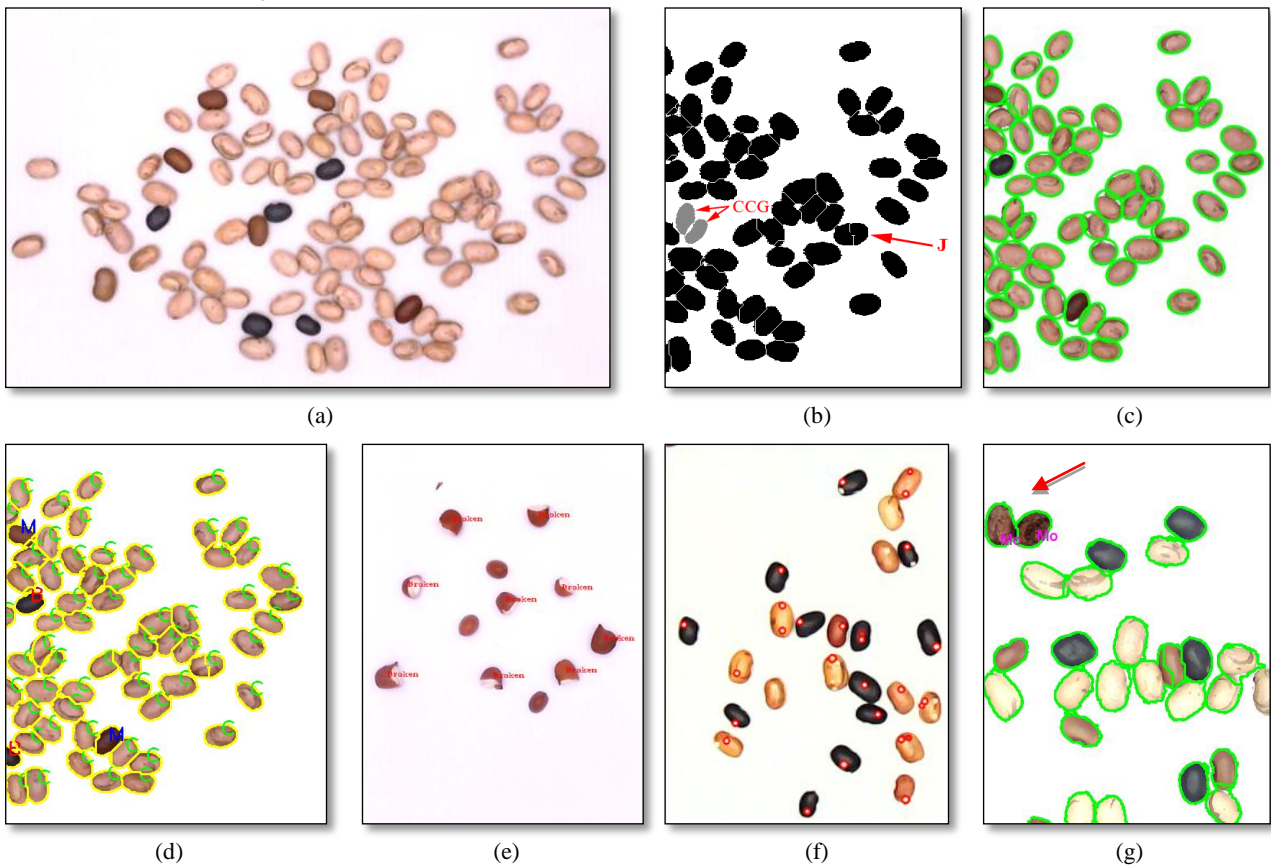


Fig 4: Results of the processing steps of the software that composes the SIVQUAF. (a) input image; (b) details of the application of CCG and the heuristic for joining (J) fragments of grains; (c) final segmentation result; (d) classification; (e) detection of broken grains; (f) detection of bored grains and (g) detection of moldy grains

In the detection of the holes, the Hough transform for circle detection (HTC), which is implemented in the OpenCV library, was used. To avoid problems in the detection of holes, maximum and minimum radius are stipulated and used as parameters of the HTC algorithm. They are easily obtained because the holes do not have large differences in terms of

diameter. According to Belan (2019), images with resolutions lower than 1920×1080 can make the detection of this defect unfeasible.

c) Moldy grains

According to Belan (2019), it is extremely difficult to discern whether a grain is moldy or burnt, observing only its external

appearance. According to literature, for determining if the grain is burnt it is necessary to open it and check the internal part, since the coloring of its skin is very similar to the moldy grain [32]. Thus, both defects are treated as unique defect (moldy/burnt). The moldy defect (Figure 4g) is identified in the SIVQUAF by a convolutional neural network (CNN) implemented in Python language (and not attached to the software), which was trained from a database of 5920 images of defective and healthy grains.

2.2.2 Hardware (equipment)

The equipment (Figure 5) that composes the SIVQUAF was developed with low-cost electromechanical materials, such as a table made of structural aluminum, which includes an image acquisition chamber (opaque white acrylic box, with a built-in camera and a lighting system), servo motor, input box with grain separator mechanism and a grain output box.



Fig 5: The SIVQUAF (set of hardware and software). Belan (2019)

The cost of the materials to make the equipment, including a full-HD camera, was approximately US\$1,200.00. Also considering a personal computer for running the software, the total cost of the SIVQUAF is approximately US\$ 2,400.00.

2.3 Embedded systems

According to Ondřej (2016), an embedded system is a

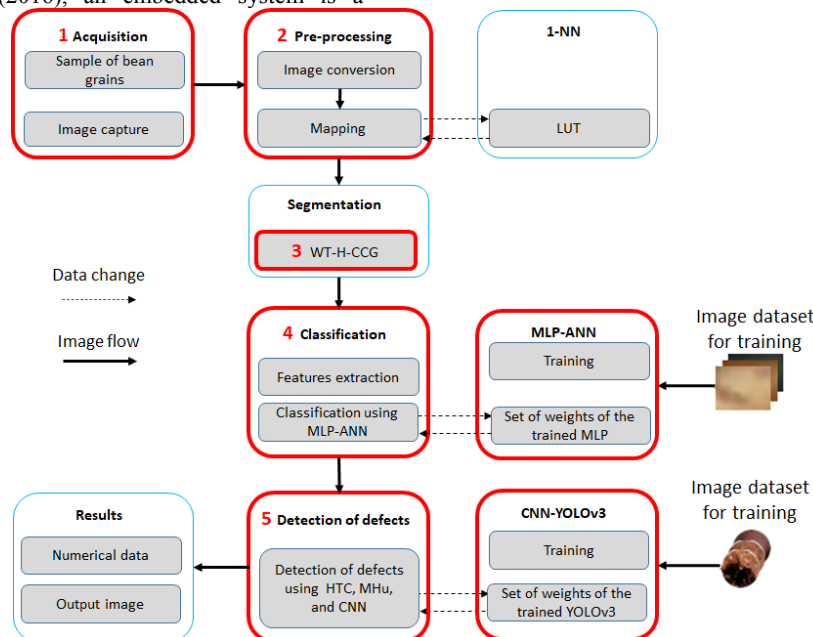


Fig 6: Diagram of working of the SIVQUAF-Compact software

software embedded in an equipment such as in an electronic car injection system, in a pen that is capable of translating texts from one language to another, in a refrigerator, televisions, etc. An embedded system can also be defined as a specialized computer system that is part of a larger machine or system.

We can see an increasing use of embedded systems or embedded computing (CE), in practically all objects constructed by man, although most of them go unnoticed by us. Virtually all electronic equipment already has or will soon have an embedded computer system.

The main feature of embedded systems is the unique functionality, that is, running only one program repeatedly. In other words, an embedded system performs a set of predefined tasks, according to requirements. In addition, some issues such as ease of operation, cost, and size of the hardware for embedded systems must be taken into account [34].

There are numerous hardware platforms (Single Board Computers – SBC) for embedded systems, such as Raspberry Pi, Orange Pi, Odroid, Banana Pi, Tinker Board, and Inforce, among others [35].

3. SIVQUAF-Compact

3.1 Software

The software that composes the SIVQUAF-Compact includes the same processing steps of the SIVQUAF software and was also developed in C/C++ using the OpenCV library. Figures 6 and 7 illustrates, respectively a schematic diagram of its functioning and the SIVQUAF-Compact interface.

The customized, optimized, and/or improved steps are highlighted in red and enumerated from 1 to 5 in the diagram, and described in Table 2. It is worth mentioning that other customizations related to the change of operating system (Windows to Linux) were also necessary. However, they are not described in this work.

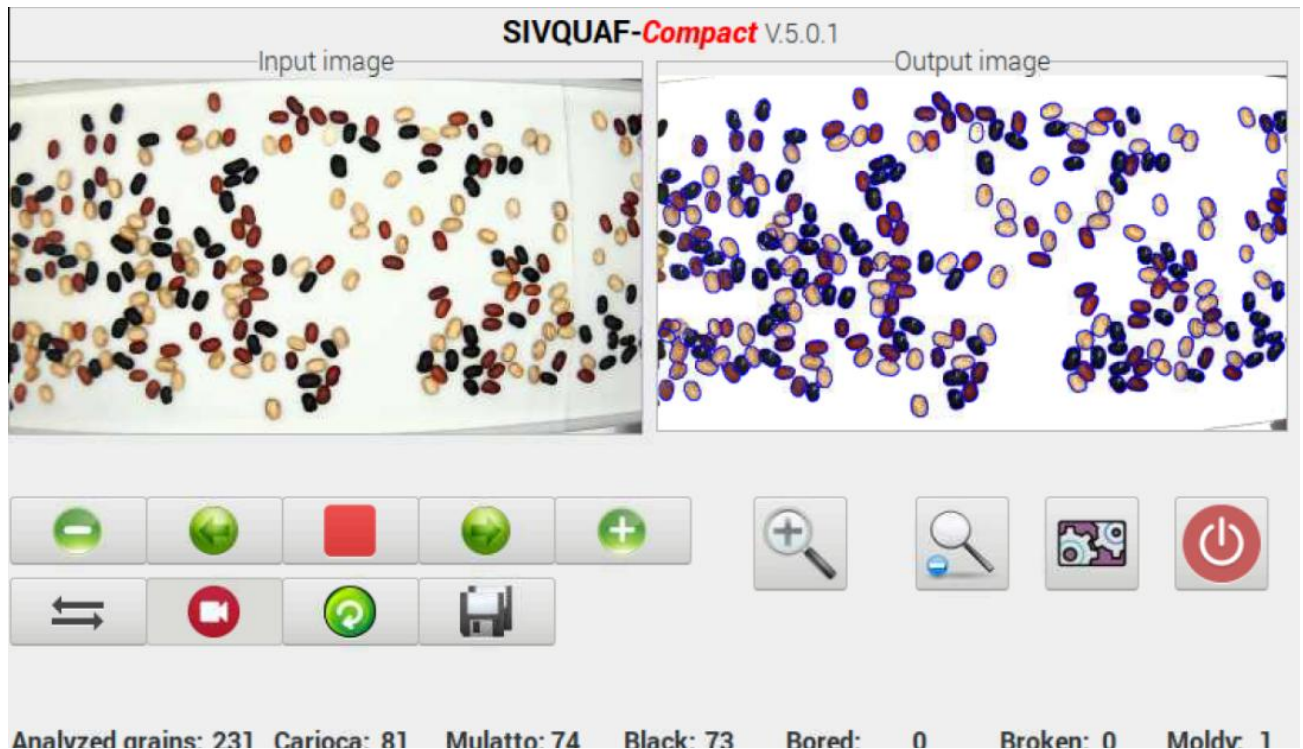


Fig 7: Interface for the SIVQUAF-Compact operation

Table 2: Optimizations, customizations, and improvements proposed in this work for embedding the SIVQUAF software

Processing step	SIVQUAF	SIVQUAF-Compact	Advantages
1. Acquisition	<ul style="list-style-type: none"> * Interface designed for a general purpose computer, not allowing adjustments in lighting and camera parameters. * Capture images with resolution of 1920×1080. 	<ul style="list-style-type: none"> * Interface designed for a Display TFT Touchscreen 7", allowing adjustments in lighting, camera parameters and file recording. * Capture images with resolution of 1280×720. 	<ul style="list-style-type: none"> * Customization of the screen. * Improvements in the system operation. * Optimization of the average time for processing an image, through the adoption of lower image resolution.
2. Pixel mapping	<ul style="list-style-type: none"> * Performed for each pixel in a serial manner. 	<ul style="list-style-type: none"> * Use of parallelism in the call of LUT to map the pixels. 	<ul style="list-style-type: none"> * Optimization of the time to remove the background from the analyzed image.
3. WT-H-CCG	<ul style="list-style-type: none"> * Use of WT-H-GCC (serial programming) 	<ul style="list-style-type: none"> * Use of WT-H-GCC (parallel programming) 	<ul style="list-style-type: none"> * Optimization of time when calling the procedure of CCG in the segmentation algorithm.
4. Classification	<ul style="list-style-type: none"> * Extraction of features from radial projections considering all channels from Lab and RGB color spaces. * Use of MLP-ANN with 4 layers of 6, 30, 45 and 3 neurons. 	<ul style="list-style-type: none"> * Extraction of features from circular projections considering the following channels from Lab and RGB color spaces: a, b, R, G, B. * Use of MLP-ANN with 3 layers of 20, 45 and 3 neurons. 	<ul style="list-style-type: none"> * Improvements in the invariance regarding the aging of the grain and gain in the classification success rate. * Time optimization in calling the classification algorithm.
5. Detection of defects	<ul style="list-style-type: none"> * Use of CNN with LeNet architecture. * CNN to identify the moldy beans was not incorporated to the SIVQUAF software. 	<ul style="list-style-type: none"> * Use of CNN with YOLOv3 architecture. * Incorporation of the CNN to identify moldy beans in the SIVQUAF-Compact software. 	<ul style="list-style-type: none"> * Improvement of the software allowing the execution of experiments with the detection of the three defects in online mode.

3.1.1 User interface

The interface designed and customized for the SIVQUAF-Compact (Figure 6) allows the operator to control the movement of the conveyor belt, process a sample, activate online (continuous) and offline inspection modes, and configure the system. The lighting control allows the brightness adjustment of LEDs (Light-emitting diode) inside the image acquisition chamber between 0 and 1500 lumens. It should be noted that this functionality did not exist on the SIVQUAF.

3.1.2 Pre-processing (mapping)

As can be seen in subsection 2.2.1.2, SIVQUAF's pre-processing step uses a lookup table (LUT). However, as this

task requires a lot of processing time to map the entire image, a special OpenCV method called *forEach* was employed allowing parallel access to pixels optimizing this step of the software.

3.1.3 Segmentation

Just like on the SIVQUAF, segmentation is done using WT-H-CCG. However, this step was optimized with the use of parallelism to make possible the use of the CCG algorithm. The optimized algorithm allows us to employ all available hardware resources (processing cores and memory) so that operations are processed faster.

3.1.4 Classification

The task of grain classification, as in the SIVQUAF, is performed by a Multi-Layer Perceptron artificial neural network (MLP-ANN). However, unlike Belan (2019) which opted for the extraction of color features from radial projections as illustrated in Figure 3, here the features are extracted over circular projections. Another difference in this work were the color channels considered from RGB and CIE Lab spaces.

In short, for the extraction of the color attributes it was decided to use 4 circular projections from the center of the grain (Figure 8) on which are extracted the average values of RGB bands and the average values of the bands a and b from the CieLab color space, that is, $(\bar{R}, \bar{G}, \bar{B}, \bar{a}, \bar{b})$, composing a vector of 20 attributes that serves as input of the MLP-ANN employed to classify and label each grain, which was defined with the following architecture: 20 neurons in the input layer, a hidden layer with 45 neurons, and an output layer with 3 neurons to indicate the grain class (C, M, or B).

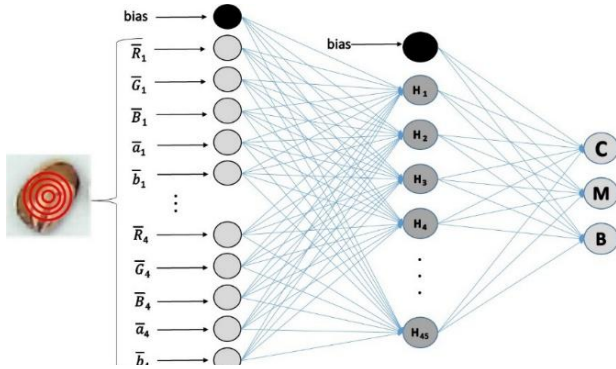


Fig 8: RNA-MLP employed in the SIVQUAF-Compact software for grains classification.

These improvements in the classification step allowed us to overcome a limitation of the SIVQUAF related to imprecision in the classification of Carioca grains with darkened skins due to the aging time. Nevertheless, the modification in the MLP-ANN architecture allowed a reduction in the processing time, essential to increase the viability of the embedded software.

3.1.5 Detection of defects

The same algorithms proposed by Belan (2019) for detecting broken and bored grains were used in the SIVQUAF-Compact. However, to identify moldy beans we developed a new CNN based on the YOLOv3 architecture, which is entirely incorporated into the software of the SIVQUAF-Compact. Just to remember, for this task the SIVQUAF employs a LeNet architecture CNN, written in Python, which was not incorporated in its software.

The training of the CNN YOLOv3 was carried out using the following parameters: 3,000 training epochs, training set containing 4,000 images, being 2,000 of defective and 2,000 of healthy grains, considering carioca, mulatto, and black beans.

3.2 Hardware (equipment)

The equipment is the same proposed by Belan (2019). However, it can be seen in Figure 9 that SIVQUAF-Compact includes an embedded module attached to the equipment, without the need for a computer, mouse, and keyboard for its control, since there is a touch screen for the operation of the developed software.



Fig 9: SIVQUAF-Compact (set of hardware and software)

The embedded module is indicated by the red arrow on the image acquisition chamber. Its arrangement in the equipment provides a comfortable operation, and without the need for extra connections for the working of the set of hardware and software.

The *Raspberry Pi 3* was chosen as embedded platform due to the fact that there is ample documentation and libraries available on it, and also because it is a relatively compact, robust and a low cost platform (US\$ 35.00), compared to many of its competitors.

As the operating system (OS), Linux Debian version 4.9 was chosen due to its compatibility with *Raspberry Pi 3*. The camera used was the OV2710 CMOS Full HD (1080p) model with a USB-2.0 connection. It was configured to acquire images with a resolution of 1280x720 pixels. To measure illumination inside the chamber, the Gy30 light sensor was used. It was positioned inside the acquisition chamber and connected to the *Raspberry* module.

Regarding the cost, there was a significant reduction when compared to the cost of the SIVQUAF, due to the non-need a personal computer (with an approximate cost of US\$ 1,200), which is being supplied by the embedded module with a total cost of US\$ 125 (*Raspberry Pi 3* = US\$35 + touchscreen 7" = US\$70 + class 10 micro-SD = US\$10 + box to embed the devices and attach the touchscreen = US\$10). Adding to this the customization of the interface and the compacting of the equipment, it can be said that there was an increase in its operational, technical, and economic viability.

4. RESULTS

4.1 Experimental setup

To evaluate the performance of the SIVQUAF-Compact,

experiments were carried out in online (or continuous) mode. As described in Belan (2019), in this mode, the grains contained in a batch are spilled continuously on the conveyor belt for the system to capture and process the images (frames). As explained in Table 2, SIVQUAF-Compact acquires images with resolution of 1280×720.

Belan (2019) also mention that using a limit speed of the conveyor belt to maintain the quality of the images, the time between two consecutive acquisitions is 8 seconds. So, we considered this time as a limit for the SIVQUAF-Compact to conclude all the processing steps to analyze an image.

As in the work of Belan (2019), we considered beans from Group I named Carioca, Black and Mulato, being the first two most consumed in Brazil. Batches containing 1000 grains were used in the experiments, with 400 Carioca (new and aged), 300 Mulatto, and 300 Black. The reason for having new and aged beans was to test the improvement made in the classification step.

Samples containing 50 grains of each defect mixed to another 150 healthy grains were used to evaluate the algorithms for detection of broken and bored grains. In the case of the moldy defect, samples containing only 20 defective grains mixed with the other 150 grains were used, due to the difficulty of obtaining and maintaining grains with this defect.

For each experiment, considering segmentation, classification and detection of defects, the success rates and average processing times were computed and compared with those obtained by SIVQUAF.

4.2 Experiments conducted with the SIVQUAF-Compact

4.2.1 Segmentation and classification

The results of the experiments regarding segmentation and

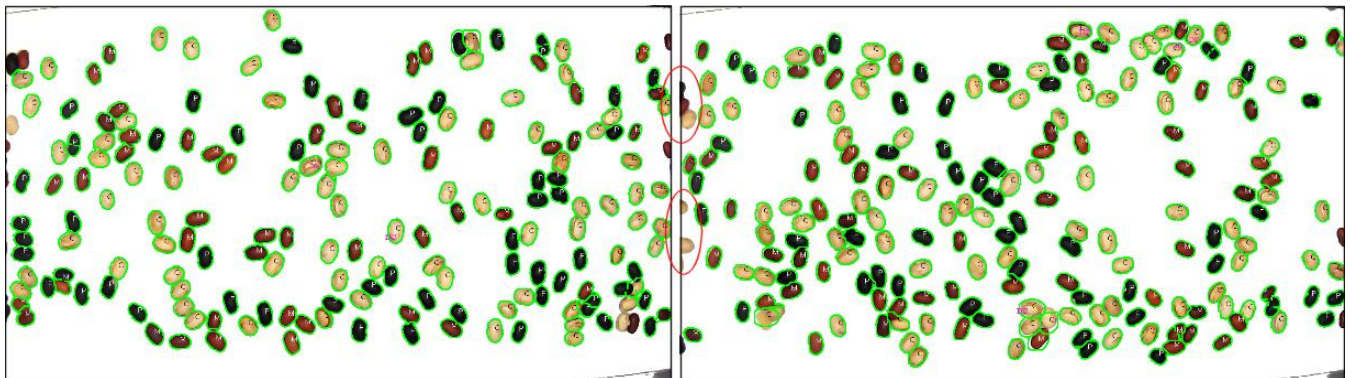


Fig 10: Grains at the edges of two consecutive acquired images that were not processed

4.2.2 Detection of defects

In the experiments with broken grains, the success rates of the SIVQUAF-Compact (57.33%) was lower than those obtained by the SIVQUAF (82.00%). This can be explained, in part, by the fact that during the experiments conducted some grains had the broken face down at the time of image acquisition, leading the algorithm to identify them as healthy (entire) grains. Another problem is when two broken grains are glued together, making the algorithm recognize them as a single grain. The solution to these problems requires further investigation, which is not a goal of this work.

Regarding bored grains, the SIVQUAF-Compact achieved 85.33% of success rate, that is, slightly less than that obtained by the SIVQUAF (88.00%). In these experiments, it can be

observed that the errors are mainly related to the quality of the image (lack of detail), a fact that was already indicated by Belan (2019) and was proven in this work.

classification are shown in tables 3 and 4. As was done in Belan (2019), repetitions of each experiment were carried out to obtain average success rates.

Table 3 - Results obtained in the segmentation of grains

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Avg	Standard deviation σ
Number of grains correctly segmented in each batch	975	969	967	975	989	975	8.6
Success rates (%)	97.50	96.90	96.70	97.50	98.90	97.50	-

Table 4 – Results obtained in the classification of grains

Class	Number of grains correctly classified in each batch					Avg success rate (%)
	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	
Carioca	398	393	395	396	396	98.90
Mulatto	279	276	280	283	281	93.27
Black	297	292	295	296	296	98.40
Success rates (%)	97.40	96.10	97.00	97.30	97.06	97.06

As can be seen in Tables 3 and 4, in experiments for both segmentation and classification tasks, success rates above 96% were obtained. Such rates, despite being slightly lower than those obtained by the SIVQUAF (98.47% and 98.10%), demonstrate the technical viability of the SIVQUAF-Compact.

The fact that the performance of the SIVQUAF-Compact was lower than the performance of the SIVQUAF in these two steps of processing, can be justified mainly by the segmentation failures in the cases of grains that were divided at the edges of images (see Figure 10) and, consequently, were not processed by the classification step.

The results of experiments involving the detection of broken, bored, and moldy grains are shown in Table 5.

Table 5: Results obtained in the defects detection

Defect		Sample 1	Sample 2	Sample 3	Avg success rate (%)
Broken	Correct detections	24	26	36	-
	Success rate (%)	48.00	52.00	72.00	57.33
Bored	Correct detections	42	43	43	

	Success rate (%)	84.00	86.00	86.00	85.33
Moldy	Correct detections	16	18	15	
	Success rate (%)	80.00	90.00	75.00	81.67

Finally, the detection of the moldy defect in the average success rate obtained by the SIVQUAF–Compact (81.67%) was also lower than the rate obtained by the SIVQUAF (85.00%).

It is important to highlight that the experiments conducted by Belan (2019) did not consider images containing healthy grains mixed with defective grains. In addition, his experiments with moldy grains were not carried out in continuous mode. This explains, in part, the high success rates obtained by SIVQUAF in the detection of defects.

4.2.3 Processing time

The average times spent by SIVQUAF–Compact to process an image from each batch of 1000 grains considered in our experiments are described in Table 6. In such times, all steps of software processing were considered, that is, pre-processing, segmentation, classification and defects detection.

Table 6: Average processing times (in s)

	Batch 1	Batch 2	Batch 3	Batch 4	Batch 5	Overall avg time
Number of images processed in each batch	7	6	6	6	7	-
Average processing time per image	6,3	7,0	8,2	5,6	4,2	6,3

The average time spent by SIVQUAF–Compact to process an image was 6.3 s, that is, below the time of 8 s defined by Belan (2019) as the maximum time for processing an image. A comparison between the processing times of the SIVQUAF–Compact and the SIVQUAF is shown in Table 7, in which the time spent by each processing step is described.

Table 7 – Comparison of processing times (in s): SIVQUAF–Compact × SIVQUAF

	Segmentation	Classification	Detection of defects			Total time spent
			Broken	Bored	Moldy	
SIVQUAF–Compact	2.38	1.25	0.67	0.70	1.32	6.3
SIVQUAF	0.80	0.10	0.10	0.10	1.00	2.1

It should be emphasized that before performing the customizations, improvements, and optimizations of the algorithms, the average time to process an image on the Raspberry Pi 3 was more than 15 s, which would make the embedded version of the SIVQUAF unfeasible.

5. CONCLUSIONS

The experiments conducted with the SIVQUAF–Compact showed that it is capable of processing images of 1280×720 pixels with average success rates of 97.50% in segmentation, 97.06% in classification and 74.78% in defects detection. As expected, due to adjustments made in the algorithms because of the hardware of Raspberry Pi 3 limitations, the success rates obtained by the SIVQUAF–Compact were lower than those obtained by the SIVQUAF. However, the results obtained in segmentation and classification are still superior to the results of many works found in the literature.

The customizations, optimizations, and improvements made have led to a good reduction in the time spent for processing an image on *Raspberry Pi 3*, reaching 6.3 s on average. Based on the developments and experiments described in this work, one can see that there was a significant gain in both the cost and operation/compacting of the equipment, increasing its operational, technical, and economic viability.

The SIVQUAF–Compact represents a contribution to technological innovation, given the new functionalities implemented. As scientific contributions, we can cite improvements made in the SIVQUAF's algorithms, which allowed to overcome some of its limitations and disadvantages. Besides, this work can be used as a “roadmap” for researchers who intend to embed CVS on compact hardware such as Raspberry Pi3.

In future works we intended to carry out new experiments aiming to overcome the segmentation problems mentioned in section 4.2.1, as well as considering the optimization of the parameters used in the algorithms for detection of defects aiming to improve their performances.

6. ACKNOWLEDGMENTS

The authors would like to thank FAPESP–São Paulo Research Foundation (Proc. 2017/05188-9) and UNINOVE by financial support and CNPq–Brazilian National Research Council for the research scholarship granted to S. A. Araújo (Proc. 313765/2019-7).

7. REFERENCES

- [1] J. Coêlho, Grain production - beans, corn and soybeans, Cad. Setorial ETENE, Fortaleza. (2017) 1–14.
- [2] A.K. Aggarwal, R. Mohan, Aspect Ratio Analysis Using Image Processing for Rice Grain Quality, Int. J. Food Eng. 6 (2010) 1–14. <https://doi.org/10.2202/1556-3758.1788>.
- [3] MAPA, Ministério Da Agricultura Pecuária e Abastecimento - Projections of agribusiness Brazil 2014/2015 to 2024/2025 Long term projections, Ministério da Agricultura, Pecuária e Abastecimento, Brasília, Brasil, 2015. http://www.sapc.embrapa.br/arquivos/consorcio/informe_estatistico/Projecoes_Agronegocio_CAFE_Mapas_2015_2025.pdf (accessed April 1, 2018).
- [4] G. Stegmayer, D.H. Milone, S. Garran, L. Burdyn, Automatic recognition of quarantine citrus diseases, Expert Syst. Appl. 40 (2013) 3512–3517. <https://doi.org/10.1016/j.eswa.2012.12.059>.
- [5] N.K. Patil, R.M. Yadahalli, J. Pujari, Comparison between HSV and YCbCr Color Model Color-Texture based Classification of the Food Grains, Int. J. Comput. Appl. 34 (2011) 51–57.
- [6] J.D.D. Cabral, S.A. de Araújo, An intelligent vision system for detecting defects in glass products for packaging and domestic use, Int. J. Adv. Manuf. Technol. 77 (2015) 485–494.
- [7] J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E.B. Pinto, P. Eisert, J. Döllner, I. Vallarino, Visual computing as a key enabling technology for industrie 4.0 and industrial internet, IEEE Comput. Graph. Appl. 35 (2015) 26–40.
- [8] A. OuYang, R. Gao, Y. Liu, X. Dong, An Automatic Method for Identifying Different Variety of Rice Seeds Using Machine Vision Technology, in: 2010 Sixth Int.

- Conf. Nat. Comput., 2010: pp. 84–88.
- [9] H. Zareiforush, S. Minaei, M.R. Alizadeh, A. Banakar, B.H. Samani, Design, development and performance evaluation of an automatic control system for rice whitening machine based on computer vision and fuzzy logic, *Comput. Electron. Agric.* 124 (2016) 14–22. <https://doi.org/10.1016/j.compag.2016.01.024>.
- [10] P.A. Belan, Sistema de visão computacional para inspeção da qualidade de grãos de feijão, Universidade Nove de Julho, 2019.
- [11] Q. Yao, Y. Zhou, J. Wang, An automatic segmentation algorithm for touching rice grains images, in: *Audio Lang. Image Process. (ICALIP)*, 2010 Int. Conf., Shanghai, China, 2010: pp. 802–805. <https://doi.org/10.1109/ICALIP.2010.5685114>.
- [12] G. Venora, O. Grillo, C. Ravalli, R. Cremonini, Identification of Italian landraces of bean (*Phaseolus vulgaris* L.) using an image analysis system, *Sci. Hortic. (Amsterdam)*. 121 (2009) 410–418. <https://doi.org/10.1016/j.scienta.2009.03.014>.
- [13] B.S. Anami, D.G. Savakar, Influence of light, distance and size on recognition and classification of food grains' images, *Int. J. Food Eng.* 6 (2010). <https://doi.org/10.2202/1556-3758.1698>.
- [14] B. Laurent, B. Ousman, T. Dzudie, M.F.M. Carl, T. Emmanuel, Digital camera images processing of hard-to-cook beans, *J. Eng. Technol. Res.* 2 (2010) 177–188.
- [15] J. Liu, W.W. Yang, Y. Wang, T.M. Rababah, L.T. Walker, Optimizing Machine Vision Based Applications in Agricultural Products by Artificial Neural Network, *Int. J. Food Eng.* 7 (2011) 1–25. <https://doi.org/10.2202/1556-3758.1745>.
- [16] J.H. Pessota, Sistema especialista aplicado à inspeção da qualidade visual de grãos de feijão., Universidade Nove de Julho, 2013. <http://bibliotecatede.uninove.br/handle/tede/206> (accessed August 26, 2020).
- [17] M.R. Siddagangappa, A.H. Kulkarni, Classification and Quality Analysis of Food Grains, *J. Comput. Eng.* 16 (2014) 01–10.
- [18] R. Kambo, A. Yerpude, Classification of Basmati Rice Grain Variety using Image Processing and Principal Component Analysis, *Int. J. Comput. Trends Technol.* 11 (2014) 80–85. <https://doi.org/10.14445/22312803/ijctt-v11p117>.
- [19] P. Dubosclard, S. Larnier, H. Konik, A. Herbulot, M. Devy, Automatic method for visual grading of seed food products, *Lect. Notes Comput. Sci. (ICIAr 2014)*. 8814 (2014) 485–495. https://doi.org/10.1007/978-3-319-11758-4_53.
- [20] P. Dubosclard, S. Larnier, H. Konik, A. Herbulot, M. Devy, Deterministic Method for Automatic Visual Grading of Seed Food Products To cite this version?., in: *4th Int. Conf. Pattern Recognit. Appl. Methods*, Lisboa, Portugal, 2015: pp. 1–6.
- [21] P. Dubosclard, S. Larnier, H. Konik, A. Herbulot, M. Devy, Automated visual grading of grain kernels by machine vision, in: *Twelfth Int. Conf. Qual. Control by Artif. Vis.*, 2015: pp. 1–8. <https://doi.org/10.1117/12.2182793>.
- [22] P. Potter, J.M. Valiente, G. Andreu-García, Automatic Visual Inspection of Corn Kernels Using Principal Component Analysis, (2015).
- [23] P.A. Belan, S.A. de Araújo, J.C.C. Santana, UM SISTEMA DE ANÁLISE DE IMAGENS PARA CLASSIFICAÇÃO AUTOMÁTICA DE GRÃOS DE FEIJÃO BRASILEIRO, in: *Proc. {XXXVI} Iber. Lat. Am. Congr. Comput. Methods Eng., ABMEC - Brazilian Association of Computational Methods in Engineering*, 2015: pp. 1–7. <https://doi.org/10.20906/CPS/CILAMCE2015-0772>.
- [24] S.A. Araújo, J.H. Pessota, H.Y. Kim, Beans quality inspection using correlation-based granulometry, *Eng. Appl. Artif. Intell.* 40 (2015) 84–94. <https://doi.org/10.1016/j.engappai.2015.01.004>.
- [25] S.A. Araújo, W.A.L. Alves, P.A. Belan, K.P. Anselmo, A Computer Vision System for Automatic Classification of Most Consumed Brazilian Beans, *Lect. Notes Comput. Sci.* 9475 (2015) 45–53. <https://doi.org/10.1007/978-3-319-27863-6>.
- [26] P.A. Belan, S.A. Araújo, W.A.L. Alves, Image Analysis and Recognition, Springer International Publishing, Cham, 2016. <https://doi.org/10.1007/978-3-319-41501-7>.
- [27] P.A. Belan, M.M.A. Pereira, S.A. Araújo, W.A.L. Alves, Abordagem Computacional para Classificação Automática de Grãos de Feijão em Tempo Real, in: *SeTII 2016*, São Paulo, Brasil, 2016: pp. 1–4.
- [28] H. Zareiforush, S. Minaei, M.R. Alizadeh, A. Banakar, Qualitative classification of milled rice grains using computer vision and metaheuristic techniques, *J. Food Sci. Technol.* 53 (2016) 118–131. <https://doi.org/10.1007/s13197-015-1947-4>.
- [29] P.J. Ramos, F.A. Prieto, E.C. Montoya, C.E. Oliveros, Automatic fruit count on coffee branches using computer vision, *Comput. Electron. Agric.* 137 (2017) 9–22. <https://doi.org/10.1016/j.compag.2017.03.010>.
- [30] S. Bhat, S. Panat, N. Arunachalam, Classification of rice grain varieties arranged in scattered and heap fashion using image processing, in: *Ninth Int. Conf. Mach. Vis. (ICMV 2016)*, Nice, França, 2017: pp. 1–6. <https://doi.org/10.1117/12.2268802>.
- [31] P.A. Belan, R.A.G. De Macedo, M.A. Pereira, W.A.L. Alves, S.A. Araújo, A Fast and Robust Approach for Touching Grains Segmentation, *Lect. Notes Comput. Sci. (ICIAr 2018)*. 10882 (2018) 482–489. <https://doi.org/10.1007/978-3-319-93000-8>.
- [32] Embrapa, Bean Classification Manual, (2012) 1–25. <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/101039/1/manualilustrado-06.pdf>.
- [33] OpenCV - Open Source Computer Vision Library, OpenCV - Open Source Computer Vision Library, (2019). <https://opencv.org/opencv-4-1/>.
- [34] H. Ali, No Title, Polytechnic Institute of Porto, 2017.
- [35] ARMBIAN, ARMBIAN DOCUMENTATION, ARMBIAN Doc. (2019). <https://www.armbian.com/download/> (accessed June 14, 2019).