AUTOMATIC CLASSIFICATION OF BEAN SEEDS USING PHENOTYPE CHARACTERISTICS

MIGUEL GARCÍA, ENG.

A Final Report Submitted for the Degree of
“Maestría en Ingeniería con énfasis en Ingeniería de Sistemas
Modalidad de Profundización"

Escuela de Ingeniería de Sistemas y Computación
Facultad de Ingeniería
Universidad del Valle

SUPERVISORS:
María Trujillo, Ph.D
Deisy Chaves, M.Sc

ABSTRACT

The quality of beans is usually determined by visual inspection, which is subjective, laborious, and prone to error. In this final project, a three-step automatic bean classification system is proposed using supervised learning. Firstly, an image of beans is pre-processed in order to remove noise, then each bean seed is segmented using the Watershed Transform algorithm. Secondly, global and local features are extracted from each individual seed. Ten global features are defined based on the phenotype and morphology characteristics used by experts. Local features are calculated using the Scale-Invariant Transform Feature (SIFT), Local Binary Pattern (LBP) and Opponent-SIFT (OSIFT) descriptors, concatenated using three strategies (Early, Intermediate and Late fusion) and represented using the Bag-of-Feature (BoF) method. Codebooks with 500 and 1000 visual words are evaluated. Thirdly, global features and BoF representations are used to build a Support Vector Machine (SVM) and Random Forest (RF) models in order to classify six bean varieties provided by the International Center for Tropical Agriculture.

A total of 600 bean images containing 39,040 individual bean seeds are used for evaluating the performance of classification models. Results showed that global features are more discriminant than local features, because, in this, they are defined based on the application domain. The best accuracy of 98.5% using global features was obtained by RF, while the best performance using local features (accuracy of 95.2%) was obtained by SVM classifier, SIFT-LBP-OSIFT descriptors, intermediate fusion as concatenation strategy and a codebook with 1000 visual words. In addition, the proposed approach was compared against a strategy for bean classification of the-state-of-the-art which used multilayer perceptron neural network (MLP) and color features. Experiments showed that the proposed local and global features achieved a better accuracy than the MLP based strategy which yielded an accuracy of 87.3%.

ACKNOWLEDGEMENTS

. . . this section is written in Spanish.

A la *Dra. María Trujillo* por supervisar este trabajo, por la valiosa orientación y asesoría que me brindó, asi como también por las clases recibidas que me ayudaron a entender los temas relacionados al proyecto.

A la *M. Ing. Deisy Chaves* por su apoyo, dedicación, y paciencia para guiarme durante todo el desarrollo del proyecto.

A mi madre *María Luisa Ampudia* y hermanos *Dairo García* y *Manuel Ampudia* por el apoyo, la paciencia, el cuidado y la comprensión.

A la *M. Ing. Lida de la Hoz* de *HarvestPlus-CIAT*, por sus consejos y facilidades que me brindó para poder trabajar en el proyecto.

A *María Isabel* de *CIAT*, por facilitarme el equipo para la toma de las imágenes de semillas de frijol.

Al *Ing. Harold Díaz* del programa de frijol del *CIAT*, por su motivación y aportes.

A los investigadores de semilla del programa de frijol del *CIAT*, por brindarme información necesaria sobre la inspección de semillas de frijol.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

WT Watershed Transform

LBP Local Binary Pattern

SIFT Scale-Invariant Transform Feature

OSIFT Opponent-SIFT

SVM Support Vector Machine

RF Random Forest

BoF Bag-of-Feature

Acc Accuracy

Part I

FINAL MASTER PROJECT

1 INTRODUCTION

Common bean (Phaseolus vulgaris L.) is one of the most important grain legumes for human consumption since it is a cheap source of dietary proteins [1]. In Colombia, harvesting and exporting beans is one of the main activities of the economy in several regions of the country, of great importance as a generator of income, employment as well as a basic product in the diet of the population. Generally, beans produced in different regions of the country are prepared for both local and international markets [2].

The International Center for Tropical Agriculture (CIAT) localized in Cali (Colombia) has a program that investigates and develops varieties of beans with genetic resistance to the main pests and diseases that affect crops. The classic genetic improvement made at CIAT consists of hybridizations between bean breeding lines to select the best offspring, aiming to improve the qualities of the parent lines and thus generate new varieties of beans [2]. Seed inspection allows researchers to determine which varieties are discarded or continue to the next line of breeding by considering phenotypic characteristics, such as non-common shape, non-striking color, and among others.

The development of bean varieties adapted to drought stress conditions and resistant to different diseases through genetic improvement is a very useful strategy to guarantee food security in marginal areas [2]. For this reason, it is important to develop tools to accelerate and make breeding programs more efficient, so breeders may have tools to support seed inspection tasks that are currently done manually. Manual inspection involves labor intensive work and the decision made thereof can be very subjective depending on the mood and condition of the person involved.

Considering the importance of the classification of bean varieties in the process of bean breeding lines, in this project, an automatic bean classification system is proposed for classifying six (6) bean varieties (see Figure 1) by applying supervised machine learning algorithms. First, an image of beans is pre-processed in order to remove noise and each seed in the image is segmented. Second, global and local features are extracted from each individual seed. Local features are represented using the BoF method. Third, global features and BoF representations are used to build a SVM and a RF models. Local features are extracted by describing key points with different texture, shape and color descriptors, and represented using Bag of Features. Global features are defined by the problem domain.

For the development of the automatic bean classification system, RF is selected based on the speed performance and minimum storage requirements, in comparison to other classifiers [3]. In addition, RF has been widely used in computer vision tasks, such as tracking, object recognition and image classification, obtaining good results [3].

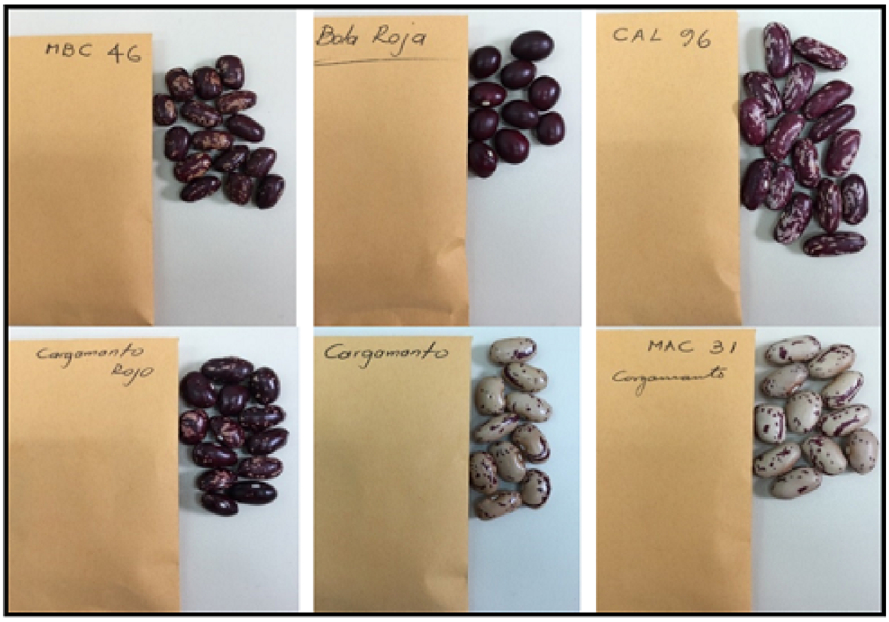


Figure 1: Bean varieties considered for classification in this project.

SVM is selected based on the good generalization capabilities using local features [4]. Local and global features are used in order to represent the content of the bean images. Local features are commonly used because are more robust to occlusions and special variations compared to global features [5]. Regarding seed segmentation, the Watershed algorithm is used for its usefulness to extract overlapped objects in images.

The remaining of the document is organized as follows: Chapter 2 details the problem statement; Chapter 3 includes the state-of-the-art; Chapter 4 describes the proposed approach for bean automatic classification; Chapter 5 focuses on the experimental evaluation; Chapter 6 contains a brief description of the software prototype development; and Chapter 7 comprises the conclusions.

2 PROBLEM STATEMENT

The progress in breeding programs depends on the precise selection of genotypes that have new or improved features [2]. This means that an accurate phenotypic characterization (e.g color, shape and size) will remain one of the mainstays of bean breeding.

Bean seed has a wide variation of color (white, cream, red, yellow, brown, and purple), shape (oval, round, and elongated), size and brightness. An illustration of different bean varieties is presented in Figure 1. These characteristics of the seed are given as a consequence of the genetic diversity that exists within bean species, so, they are taken into account for the classification of bean varieties [2].

The bean classification problem has been addressed using manual and automatic approaches. Manual classification is performed by an expert. However, human perception could be easily fooled and labor costs are relatively high since a significant amount of seeds are inspected after harvesting. In addition, it is difficult to standardize the classification results obtained by experts [6]. On the other hand, in the automatic classification, a camera is used to take digital images of bean seed samples, in order to quantify characteristics as color, shape and texture using computer vision algorithms, and classify them into bean varieties using machine learning algorithms. This approach has great importance to researchers in the field of agriculture [6]. However, automatic classification of bean varieties is still an open problem due to difficulties in the process of the segmentation of images with "glued" seeds and the selection of the most appropriated features to distinguish bean varieties with a similar appearance.

2.1 OBJECTIVES

In this section, a general objective and specific objectives of the project are presented.

2.1.1 *General Objective*

Classify automatically bean seeds from different varieties based on the analysis of digital images.

2.1.2 *Specific Objectives*

- Define a protocol for the acquisition of bean seed images.
- Build a model using algorithms of machine learning for the classification of bean seeds.
- Develop a software prototype for the classification of bean seeds.

- Validate the classification model proposed versus one of the state-of-the-art strategies for classification of bean seeds.

2.2 RESEARCH RESULTS

Table 1 includes a summary of the results obtained during the development of each specific objective.

Table 1: Relation of the objectives and the corresponding result

#	Objective	Result
1	Define a protocol for the acquisition images of bean seeds.	An acquisition protocol was defined, in the Section 4.1.
2	Build a model using algorithms of machine learning for the classification of bean seeds.	Two classification models were built based on SVMs and RFs classifiers, in Chapter 5.
3	Develop a software prototype for the classification of bean seeds.	A user friendly interface was developed, in Chapter 6.
4	Validate the classification model proposed versus one of the state-of-the-art strategies for classification of bean seeds.	A strategy for classification of bean seeds was implemented, in Section 5.5.

3 STATE OF THE ART

3.1 MANUAL CLASSIFICATION OF BEAN SEEDS

Generally, bean classification is performed manually by human inspectors. The manual inspection process at CIAT is described:

1. After harvesting, the seeds go through a process of cleaning (venting), where any dirt such as dust or soil from the seed is removed.

2. A drying process is done on the seeds, where they are exposed to the sun.

3. A performance analysis is carried out on a sample of 100 seeds, this analysis required measure from the seeds characteristics such as weight and humidity.

4. The seeds are inspected by the breeders, who based on their experience, determine the phenotypic features such as:
   - Color, both dominant and secondary are determined visually. Inspectors use a color scale: (1) White. (2) Cream-beige. (3) Yellow. (4) Brown-marroon. (5) Pink. (6) Red. (7) Purple. (8) Black. (9) Others - gray, etc.
   - Size, taking into account the weight, where: Small - corresponds to seeds up to 25g/100, Medium - corresponds to seeds between 25 and 40g/100, Large - corresponds to seeds from 40g/100. However, there are inspectors that determine the size visually.
   - Shape, it is visually classified as oval, round, and elongated.
   - Brightness, it is visually determined whether the seed is bright or opaque.
   - Finally, the analyzed seed is classified taking into account the observed features, as the variety with the most similarity.

3.2 AUTOMATIC IMAGE CLASSIFICATION

Image classification is a challenge in computer vision. In general, image classification aims to find representations of images that can be used to categorize automatically images into different classes. Typically, the supervised approach used machine learning algorithms to train a classification model from a labeled training set of images, these algorithms that classify images require a pre-processing prior to classification. This process may involve extracting relevant features based on some prior knowledge about their context [7].

Perhaps one of the most significant developments in the last decade is the application of local features to image classification, including the introduction of bag-of-features representation which inspires a lot of research in the field [7]. The classification process may be summarized in four stages [8, 9]: 1) Feature extraction by partitioning images into patches and describe their content using local descriptors; 2) Image codification by assigning descriptors at each image patch to a predetermined vocabulary; 3) Calculation of aggregated statistics using coded descriptors (pooling) to generate a bag-of-features by image; 4) Classification by applying a classification model. The classifier is trained using the bag-of-features generated by images as feature vectors.

3.3 AUTOMATIC CLASSIFICATION OF BEAN SEEDS

In the literature, there are some works addressing the automatic classification of bean seeds. In [6] was developed a computer vision system for beans classification based on the skin colors of the grains. They implemented known techniques such as binarization, edge detection, mathematical morphology operators and color features quantification by statistical moment in Matlab software. They employed a Multi-layer Perceptron for the classification task, obtaining a success rate of 90.6%.

In [10] a computer vision system was developed using a multilayer perceptron neural network to classify ten bean varieties based on color characteristics. The experiments showed that the system was able to classify bean varieties with a total sensitivity of 96% and specificity of 97.1%.

In [11] was proposed a system for classification of six landraces of beans from Italy, using image analysis library KS-400. In the experiments, they used features like size, shape, color and texture of the grains and obtained success rate of 99.56%. The same authors, in the subsequent work [12] conducted new experiments taking into account fifteen Italian traditional landraces of beans, where they achieved a success rate of 98.49%.

In [13] was used color histograms and statistical analysis to evaluate if there is a relationship between changes in the skin color of beans and the phenomenon "hard-to-cook beans." The results showed that this relationship exists, in addition, the proposed model can be used to detect hard-to-cook beans.

Additionally, in [14] was proposed a robust correlation-based granulometry module for segmentation of grains and conducted several experiments considering three of the most consumed Brazilian beans. They used color features and achieved a success rate of 99.88%. Finally, in [15] was proposed a system that employs the Wateshed Transform (WT) with refinement heuristics for segmenting grains and a

Multilayer Perceptron neural network for classifying them. They presented a system for automatic classification of Brazilian beans (same varieties used in [14]) aiming to improve the processing time of the system proposed in [14]. They used color features and achieved a success rate of 99.14%.

All the seven works above [6, 10, 11, 12, 13, 14, 15] demnostrated the importance of computer vision systems for inspection of beans. It is observed that the characteristics most used and appropriated for bean characterization have been color, texture, shape and size. Neural networks are the most reported machine learning technique for classifying beans. Table 2 presents a grouping of the revised works and the techniques used in each case.

Table 2: Summary of techniques used in bean seed classification systems

Reference work	Features	Classification method	Acc (%)	Sensibility & Specificity (%)
[6]	Color & size	Neural Networks	90.60	—
[10]	Color	Neural Networks	—	96 & 97.10
[11]	Color, size, texture & shape	Linear Discriminant Analysis	99.56	—
[12]	Color, size & shape	Linear Discriminant Analysis	98.49	—
[14]	Color	k-Nearest Neighbors	99.88	—
[15]	Color	Neural Networks	99.14	—

To our best knowledge, local features have not been used for the classification of bean seeds. One advantage of using local features is that they may be used to recognize objects despite significant clutter and occlusion [5]. Therefore, in this project, local features are evaluated. In addition, the results are compared with the results obtained evaluating global features.

4

PROPOSED APPROACH FOR AUTOMATIC BEAN CLASSIFICATION

The proposed approach for automatic bean classification is illustrated in Figure 2. Given an RGB image: First, the image is pre-processed in order to remove noise and each seed in the image is segmented using the WT algorithm. Second, global and local features are extracted from each individual seed. Local features are represented using the BoF method. Third, global features and BoF representations are used to build a SVM and a RF models in order to classify the six bean varieties considered in this research project, illustrated in Figure 1.

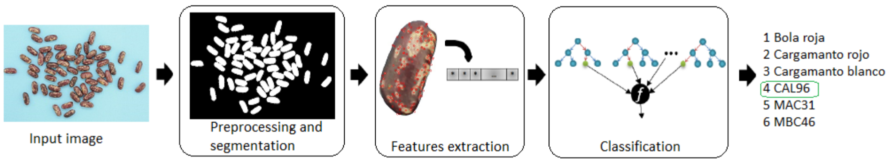


Figure 2: Flow diagram of the bean classification proposed approach.

4.1 IMAGE ACQUISITION

The performance of an image processing system depends on the image acquisition process. Thus, an acquisition protocol is defined in order to homogenize this process. In this project, six bean samples were provided from CIAT. A product photography studio Photo-eBox Plus (PeP) was used to obtain standard images from the samples(see Figure 3). The PeP has multiple switches that allow on-off control over the fluorescent lights around the box. A Nikon D3300 camera with a lens (AF-S Micro-Nikkor f/2.8G ED) was used to take images, this camera was located in the center of the box and 30 cm above the sample holder (10 x 13", where the user can load up a few beans -for example, 100 seeds). A blue color is selected as background because of its contrast with beans [10]. The acquisition protocol is described next:

1. Place the camera on the tripod at a height of 30 cm.
2. Place the blue sample holder inside the PeP.
3. Turn on the lights *Left Side Light* and *Right Side Light* in the PeP.
4. Start the Camera Control Pro program.
5. Select the *Exposure 1 tab* and set the following parameters: (a) Shutter Speed: 1/50 sec. (b) Aperture: f/16. (c) Exposure Mode: Manual.

6. Select the *Exposure 2 tab* and set the following parameters: (a) ISO Sensitivity: ISO 500. (b) White Balance: Auto.

7. Select the *Storage tab* and set the following parameters: (a) Data Format: JPEG (Fine). (b) Image Size: 4288x2848.

8. Select the *Download Options tab* and select the folder to which photographs will be downloaded as they are taken.

9. Load up the beans on the sample holder.

10. Take the image.

Figure 3: Photo e-Box Plus used to acquire the bean images.

4.2 PRE-PROCESSING AND SEGMENTATION

Image pre-processing and segmentation is the first step in image analysis and also an important part of any automatic image recognition system. The quality of the results of analysis depends highly on these stages [16]. Before processing the images of beans seeds, it was necessary to conduct pre-processing in order to remove noise due to lighting variability and segmentation to extract each seed. For that, the well-known region-based segmentation approach called Watershed Transform (WT) [17] is used. The Watershed algorithm starts with markers. These markers can be either manually defined using via point-and-click, or can be automatically or heuristically defined using methods such as thresholding and/or morphological operations [18].

Based on these markers, the watershed algorithm treats pixels in the input image as local elevation (called a topography) — the method "floods" valleys, starting from the markers and moving outwards, until the valleys of different markers meet each other [18]. In order to obtain an accurate watershed segmentation, the markers must be correctly placed.

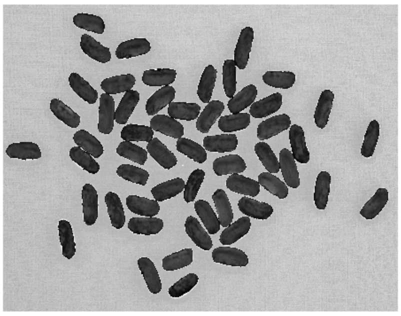
Image pre-processing and segmentation was conducted using the next steps:

1. All pictures were captured in the RGB color space.
2. The Median Filter with a window of 5x5, is used for removing noise.
3. Taking into account that the blue color has good contrast with the beans, a gray image was obtained subtracting B channel from R channel (see Figure 4b).
4. R-B gray image was converted into binary using the Otsu algorithm [19].
5. Morphological opening and closing operations are applied to remove small objects and complete beans by closing small holes.
6. The region corresponding to the background (*background_area*) was identified by applying dilation to the binary image. Dilation increases object boundaries allowing to identify background regions clearly.
7. Regions that contain beans (*foreground_area*) were extracted. Since the beans are touching each other, the distance transform is used to create a border as far as possible from the center of the overlapping beans by thresholding [20]. In this work, a threshold value of 60, set experimentally, was used (see Figure 4c and Figure 4d).
8. In the binary image are regions which may correspond to seeds or background (*unknown regions*). These areas are normally around the seed boundaries where foreground and background meet (or even two different seeds meet). The unknown regions were obtained by subtracting *foreground_area* from *background_area* (see Figure 4e).
9. A marker is created and the regions inside the image (foreground, blackground and unknown) are labelled. For this, the *connectedComponents* method was used.
10. The watershed method was applied with the markers obtained in the previous step. After that, a binary mask image was created for each segmented object (bean seeds) in order to extract them. Finally, the *findContours* method was applied to get the contour of each bean seed (see Figure 4f).

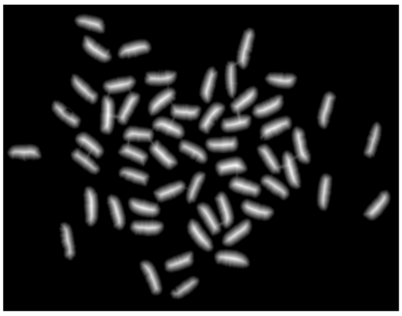
4.3 features extraction

A feature vector is obtained from a bean seed image to train classification models using global and local features. Global features are calculated over the whole image. Local features are calculated over image patches (small regions) and represented using a BoF method.

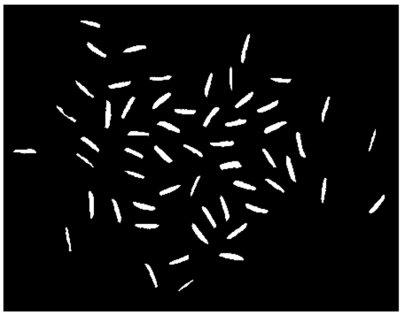
(a) RGB Original Image.



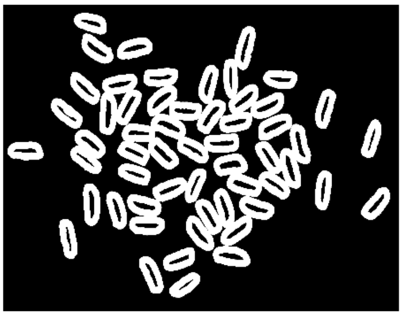
(b) Gray image.



(c) Distance transform image from (a).



(d) Thresholding image from (b).



(e) Unknown region (border).

(f) Segmented bean seeds, enclosed by a red color line.

Figure 4: Segmentation process of bean seeds.

4.3.1 *Global Features*

Ten features are selected to represent bean seeds based on the characteristics of color, shape and brightness used by experts during manual classification:

1. **Predominant and secondary colors:** are extracted from each seed using the RGB color space and the K-means algorithm, in Figure 5d and Figure 5e.

2. **Brightness (B):** is calculated by averaging the intensity value of the RGB channels in the bean area.

$$B(I) = \sqrt{0.241 \times r^2 + 0.691 \times g^2 + 0.068 \times b^2}, \tag{1}$$

where r, g and b corresponds to the Red, Green and Blue channel of the RGB image, I.

3. **Area (A):** of a bean seed is measured by counting the number of pixels inside the seed, in Figure 5b.
4. **Perimeter (P):** is computed by counting the number of pixels lying on the border of a bean seed, in Figure 5c.
5. **Maximum diameter and minimum diameter:** are respectively, the major and minor axis lengths of an object in a binary image.
6. **Equivalent diameter (Deq):** corresponds to the diameter of the circle with the same object area [20], in Figure 5f:

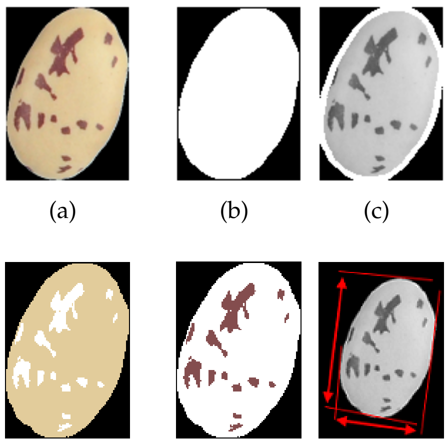
$$D_{eq} = \sqrt{\frac{4A}{\pi}} \tag{2}$$

7. **Circularity (C):** is calculated as a ratio between the perimeter and the area of a bean seed. A circularity value of 1 corresponds to a circle [20].

$$C = \frac{4\pi \times A}{P^2} \tag{3}$$

8. **Elongation (E):** is measured as a ratio between the maximum diameter ($D_{maxferet}$) and the area of a bean seed [20].

$$E = \frac{{D_{maxferet}}^2}{A} \tag{4}$$



(a) (b) (c)

(d) (e) (f)

Figure 5: Illustration of global features: (a) Original image of a bean seed; (b) Area of bean seed, in white color; (c) Perimeter of bean seed, line in white color; (d) Predominant color of bean; (e) Secondary color of bean (spots); (f) Feret diameters.

4.3.2 *Local Features*

There are several ways to extract local regions, as it is illustrated in Figure 6. *Random sampling* of a fixed number of patches from a given image. *Key point detectors* are used to choose the local regions which correspond to characteristic points in the image as edges. *Dense sampling grid* by partitioning the image into square patches spaced at a fixed number of pixels.

Figure 6: Strategies to extract local regions to represent images.

In this project, local regions are calculated with a dense regular grid of 16x16 pixels and sliding region with a step size of 8 pixels. Each region is described by SIFT descriptor which uses a histogram of normalized orientations to describe a feature point as a vector of 128 values [21] (see Figure 7). SIFT is widely used because is robust against changes in scale, rotation, and viewpoint.

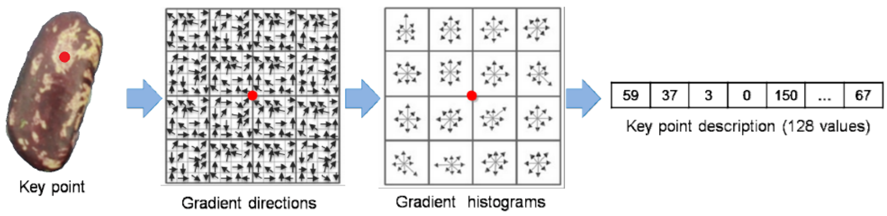


Figure 7: Sift description of a key point.

The local region also is described by Local Binary Pattern (LBP), which compute a local representation of texture. This local representation is obtained by comparing each pixel with its surrounding neighborhood of pixels [22]. A main benefit of the original LBP implementation is that it may capture extremely fine-grained details in the image. However, it is not able to capture details at varying scales, only at a fixed 3 x 3 scale. To handle this, an extension to the original LBP implementation was proposed by [23]. This LBP extension considers variable neighborhood sizes using two parameters: (i) The number of points, p in a circularly symmetric neighborhood to analyze (in order to remove the dependency on a square neighborhood) and (ii) The radius of the circle r, which allows us to account for different scales. In this research, LBP is computed using $p = 24$ and $r = 5$ (these parameters were set experimentally).

Additionally, the local region is described by Opponent-SIFT descriptor, which describes a channel in the opponent color space using each region. Since SIFT descriptors have high performance when they are applied to describe color content [24].

4.3.3 *BoF*

Local features obtained previously are combined using the BoF method in order to generate a unique image representation. This method represented an image using a visual vocabulary or codebook, where each element is a visual word. A codebook is constructed by clustering features extracted from a set of training images [25]. In this way, an image can be represented by the number of occurrences of each visual word in the image.

4.3.4 *Codebook Construction*

The codebook is obtained applying an unsupervised learning algorithm to a subset of the whole local descriptors extracted from the training images [9]. In this work, a subset of 50,000 and 100,000 local features is randomly chosen. The K-means algorithm with k=500 and k=1,000 is used to learn the codebook. Finally, the codebook corresponds to the centroids from the clustering, where each centroid represents a visual word. This process is illustrated in Figure 8.

Figure 8: Codebook construction.

4.3.5 *BoF Representation*

Given an image, local descriptors are extracted and each descriptor is assigned to the closest visual word in the codebook using the Euclidean Distance. Finally, a BoF histogram is generated as image representation. This histogram is normalized using L1 Norm. The whole process is illustrated in Figure 9.

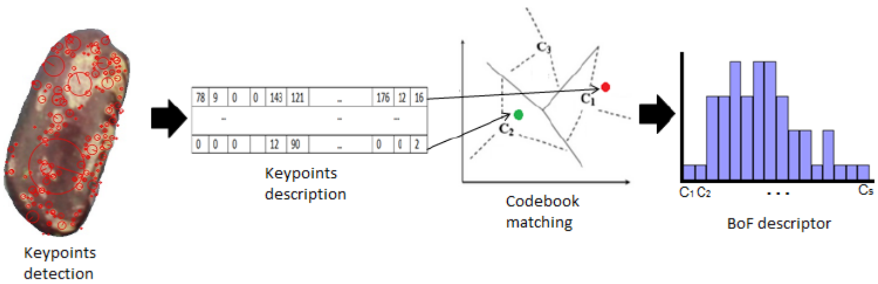


Figure 9: BoF image representation.

4.3.6 *Concatenation of Features*

A concatenation method is required for combining local features in order to get a better image representation. In this project, three types of local descriptors (SIFT, LBP and OSIFT) are combined using three concatenation methods (see Figure 10): **Early fusion:** SIFT-OSIFT, SIFT-LBP and SIFT-OSIFT-LBP features are concatenated before the codebook is generated. **Intermediate fusion:** SIFT-OSIFT, SIFT-LBP and SIFT-OSIFT-LBP features are concatenated after the codebook is obtained. **Late Fusion:** addresses the problem of combining the prediction scores of multiple classifiers, in which each score is predicted by a classifier trained with a specific feature [20]. In our case, SIFT, OSIFT, and LBP prediction scores are combined to obtain SIFT-OSIFT, SIFT-LBP and SIFT-OSIFT-LBP models.

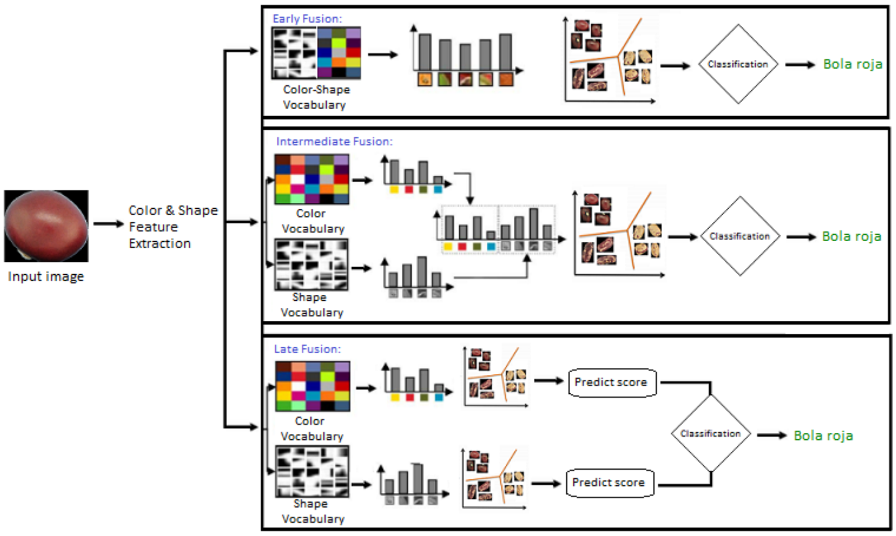


Figure 10: Illustration of the three concatenation strategies.

4.3.7 *Classification*

In this step, a classification model is built by a machine learning algorithm using the BoF representation obtained for the training images. In this work, SVM and RF are used.

SVM is a supervised training algorithm which learns a classification model from data by choosing the best hyperplane that categorizes new examples [26]. The best hyperplane is defined as the one which maximizes the margin or distance of separation between the two classes and the separating hyperplane [27]. The trade-off between maximizing the separation distance and minimizing the training error term is controlled by a regularization parameter, C.

On the other hand, RF is an ensemble of T decision trees. Each tree is trained using a subset of samples selected randomly from a labeled training set. A parameter D is used to restrict the maximum depth to grow the decision trees. In particular, given an input feature vector, each decision tree of the forest classifies it, and the ensemble output is the most popular class [3].

5 BUILDING AND EVALUATING CLASSIFICATION MODELS

Global and local features described in Section 4.3 are used to train classifiers. Global features are normalized in order to avoid the effect of different scales. 600 images (100 images per variety) containing 39,040 individual bean seeds were used, 420 images (70%) with 27,332 bean seeds for training and 150 images (30%) with 11,708 bean seeds for testing.

SVM classification models are trained using a parameter $C = \{1, 5, 10\}$. RF classifiers are trained using $T = \{50, 100, 200\}$ trees and a maximum depth tree, $D = \{6, 10\}$. Each model was built with a codebook of 500 and 1000 visual words. Optimal parameters for each classifier were chosen experimentally using a 5-fold cross-validation. In addition, the best classifier learned in each case is used to classify the testing set for evaluating model generalization.

Accuracy (Acc) and F1 score were used as evaluation measures. Accuracy corresponds to the fraction of correctly classified instances in the test set. On the other hand, F1 score is a weighted average of the precision and recall metrics Table 3 shows the learned models with local features.

Table 3: Learned RF models by using local features.

Model	Set of features	Concat.
1	SIFT-OSIFT	Early
2	SIFT-OSIFT	Interm.
3	SIFT-OSIFT	Late
4	SIFT-LBP	Early
5	SIFT-LBP	Interm.
6	SIFT-LBP	Late
7	SIFT-OSIFT-LBP	Early
8	SIFT-OSIFT-LBP	Interm.
9	SIFT-OSIFT-LBP	Late

5.1 FEATURE SELECTION FOR SVM

The criterion for selecting the optimal parameters for each model was to maximize the accuracy rate. Table 4, Table 5 and Table 6 show the optimal parameters found for each model using SVM, global features and local features with codebooks of 500 and 1000 visual respectively.

In the case of the ten global features, the accuracy rate obtained was 98.3% for all the parameters.

Table 4: Optimal parameters for SVM using global features

Set of Features	Acc ± I. C (%) for Parameter C		
	1	5	10
GLOBAL	98.3 ± 0.004	98.3 ± 0.005	98.3 ± 0.004

Table 5: Optimal parameters for SVM using local features with codebook size of 500

	Acc ± I. C (%) for Parameter C		
Model	1	5	10
1	93.2 ± 0.015	93.2 ± 0.011	93.2 ± 0.011
2	94.4 ± 0.008	94.4 ± 0.008	94.4 ± 0.008
3	93.6 ± 0.009	93.6 ± 0.009	93.6 ± 0.009
4	94.0 ± 0.013	94.0 ± 0.013	94.0 ± 0.013
5	94.5 ± 0.018	94.5 ± 0.018	94.5 ± 0.018
6	93.5 ± 0.012	93.5 ± 0.012	93.5 ± 0.012
7	93.6 ± 0.004	93.5 ± 0.004	93.5 ± 0.004
8	**95.2 ± 0.014**	**95.2 ± 0.014**	**95.2 ± 0.014**
9	94.4 ± 0.012	94.2 ± 0.014	94.2 ± 0.014

Table 6: Optimal parameters for SVM using local features with codebook size of 1000

	Acc ± I. C (%) for Parameter C		
Model	1	5	10
1	92.2 ± 0.022	92.2 ± 0.022	92.2 ± 0.022
2	94.1 ± 0.011	94.1 ± 0.011	94.1 ± 0.011
3	91.2 ± 0.014	91.2 ± 0.014	91.2 ± 0.014
4	94.3 ± 0.004	94.3 ± 0.004	94.3 ± 0.004
5	95.5 ± 0.010	95.5 ± 0.010	95.5 ± 0.010
6	93.2 ± 0.011	93.2 ± 0.010	93.2 ± 0.010
7	92.4 ± 0.015	92.4 ± 0.015	92.4 ± 0.015
8	**96.0 ± 0.012**	**96.0 ± 0.012**	**96.0 ± 0.012**
9	92.5 ± 0.016	92.5 ± 0.016	92.5 ± 0.016

Results shown in Table 5 and Table 6 are consistent with the performance observed for global features. Accuracy values did not have significant changes for the different values of the C parameter, in some cases the values were constant. The same behavior it is observed for the codebook, for both sizes, 500 and 1000, the accuracy presented similar values. The better accuracy performance (96.0 ± 0.012) was obtained using a dictionary size of 1000, SIFT-OSIFT-LBP features and intermediate fusion as concatenation strategy.

5.2 FEATURE SELECTION FOR RF

Table 7, Table 8 and Table 9 show the optimal parameters found for each model by using RF, global features and local features with codebook of 500 and 1000 visual words respectively.

Table 7: Optimal parameters for RF using global features

Set of Features	Acc ± I. C (%) for Parameter T and D					
	50 and 6	100 and 6	200 and 6	50 and 10	100,and 10	200 and 10
GLOBAL	97.6 ± 0.005	97.7 ± 0.005	97.7 ± 0.005	98.3 ± 0.005	**98.4 ± 0.006**	**98.4 ± 0.005**

In the case of the global features, it was observed that a higher value of D increases the accuracy.

Table 8: Optimal parameters for RF using local features with codebook size of 500

	Acc ± I. C (%) for Parameter T and D					
Model	50 and 6	100 and 6	200 and 6	50 and 10	100 and 10	200 and 10
1	84.8 ± 0.023	85.4 ± 0.024	86.0 ± 0.021	89.3 ± 0.033	89.7 ± 0.026	90.1 ± 0.021
2	88.1 ± 0.035	88.4 ± 0.033	88.7 ± 0.024	91.3 ± 0.021	91.8 ± 0.031	92.6 ± 0.023
3	98.3 ± 0.008	98.4 ± 0.005	98.4 ± 0.006	98.9 ± 0.003	98.9 ± 0.001	99.0 ± 0.001
4	82.2 ± 0.033	83.4 ± 0.024	83.5 ± 0.031	86.5 ± 0.025	87.4 ± 0.026	87.7 ± 0.032
5	89.3 ± 0.015	90.3 ± 0.015	90.5 ± 0.019	92.0 ± 0.008	92.7 ± 0.008	92.4 ± 0.011
6	99.3 ± 0.004	99.2 ± 0.003	99.2 ± 0.003	99.4 ± 0.004	99.4 ± 0.003	99.4 ± 0.003
7	83.9 ± 0.016	84.3 ± 0.034	84.6 ± 0.025	88.3 ± 0.026	89.2 ± 0.020	89.5 ± 0.017
8	91.8 ± 0.022	92.9 ± 0.011	93.2 ± 0.014	93.9 ± 0.009	94.4 ± 0.009	94.9 ± 0.015
9	**99.4 ± 0.002**	**99.3 ± 0.002**	**99.3 ± 0.003**	**99.5 ± 0.002**	**99.6 ± 0.002**	**99.5 ± 0.002**

Table 9: Optimal parameters for RF using local features with codebook size of 1000

	Acc ± I. C (%) for Parameter T and D					
Model	50 and 6	100 and 6	200 and 6	50 and 10	100 and 10	200 and 10
1	83.1 ± 0.025	83.8 ± 0.016	85.2 ± 0.023	88.1 ± 0.016	88.7 ± 0.015	89.4 ± 0.017
2	86.0 ± 0.034	87.6 ± 0.014	88.3 ± 0.037	89.9 ± 0.035	90.9 ± 0.030	91.5 ± 0.028
3	98.4 ± 0.008	98.4 ± 0.007	98.4 ± 0.008	99.0 ± 0.008	99.1 ± 0.006	99.1 ± 0.006
4	81.9 ± 0.036	82.5 ± 0.026	83.0 ± 0.030	85.5 ± 0.021	86.9 ± 0.020	87.0 ± 0.024
5	89.1 ± 0.023	89.9 ± 0.017	90.5 ± 0.025	91.5 ± 0.010	92.0 ± 0.016	91.9 ± 0.025
6	**99.5 ± 0.002**	**99.5 ± 0.004**	**99.5 ± 0.003**	99.6 ± 0.002	**99.6 ± 0.002**	99.6 ± 0.001
7	83.7 ± 0.024	84.2 ± 0.021	84.6 ± 0.021	87.5 ± 0.027	88.9 ± 0.014	88.9 ± 0.021
8	91.1 ± 0.026	92.0 ± 0.011	92.3 ± 0.021	93.3 ± 0.014	93.8 ± 0.013	94.5 ± 0.012
9	**99.5 ± 0.003**	**99.5 ± 0.002**	**99.5 ± 0.003**	**99.7 ± 0.002**	**99.6 ± 0.003**	**99.7 ± 0.002**

For local features, results are consistent with the performance observed for global features, where a higher value of D increases the accuracy. The best accuracy (99.7 ± 0.002) was obtained using a dictionary size of 1000, SIFT-OSIFT-LBP features, late fusion as concatenation strategy, T={50,200} and D=10. This indicates that 50 decision trees are enough to perform the highest accuracy.

5.3 SVM EVALUATION

Table 10 shows the classification results of the generalization dataset using the best SVM model learnt for global and local features.

Table 10: Accuracy of bean classification using SVMs

Model	Codebook size	Performance SVM for the best C		Generalization Acc (%)	F1 Score (%)
		Parameter C	Acc ± I. C (%)		
GLOBAL		**10**	**98.3 ± 0.004**	**98.3**	**98.2**
1	500	1	93.2 ± 0.015	94.6	94.1
2	500	1	94.4 ± 0.008	94.4	93.9
3	500	1	93.6 ± 0.009	93.8	93.3
4	1000	1	94.3 ± 0.004	94.5	94.0
5	1000	1	95.5 ± 0.010	94.6	94.2
6	500	1	93.5 ± 0.012	94.2	93.7
7	500	1	93.6 ± 0.004	94.3	93.9
8	**1000**	**1**	**96.0 ± 0.012**	**95.2**	**94.9**
9	1000	1	94.4 ± 0.012	94.3	93.8

Models obtained with SVM shown that both, global and local features, classified correctly bean varieties, with accuracies over 93.8% for the generalization test. In particular, models built using global features have a higher accuracy of 98.3% and F1 score of 98.2% than models built with SIFT-OSIFT-LBP local features (accuracy of 95.2% and F1 score of 94.9%). The concatenation strategy for local features affected the accuracy, in particular, the intermediate fusion exhibits better accuracy values in comparison to early and late fusion strategies, because of intermediate fusion includes more features, and it uses three codebooks for texture, shape and color features independently. In addition, due to codebooks are separated, the final descriptor always has the same number of features for describing texture, shape and color, in contrast with early fusion where the unique codebook is built with the concatenation of three features and late fusion where the codebook is built using each feature independently and the concatenation is performed at the classifier level. The combined descriptors also affected the accuracy, taking into account the similarity in features between bean seeds analyzed. Combining more descriptors allows to discriminate seeds better, in this case, the higher accuracy and F1 score was achieved by combining the descriptors SIFT, OSIFT and LBP for shape, color and texture features.

5.4 RF EVALUATION

Table 11 shows the results using RF for global and local features. Results are consistent with the performance observed for SVM classifiers. RF models showed that global and local features allowed to classify correctly bean varieties, with accuracies over 87% for the generalization test. In particular, models built using global features have a higher accuracy of 98.5% and F1 score of 98.4% than models built with SIFT-OSIFT-LBP local features and the late fusion strategy (accuracy of 95.1% and F1 score of 94.7%).

Table 11: Accuracy of bean classification using RF

Model	Codebook size	Performance RF for the best T and D			Generalization Acc (%)	F1 Score (%)
		Parameter		Acc ± I. C (%)		
		T	D			
GLOBAL		**200**	**10**	**98.4 ± 0.005**	**98.5**	**98.4**
1	500	200	10	90.1 ± 0.021	90.8	90.0
2	500	200	10	92.6 ± 0.023	93.6	93.0
3	1000	100	10	99.1 ± 0.006	90.3	89.5
4	500	200	10	87.7 ± 0.032	87.9	86.9
5	500	200	10	92.7 ± 0.008	93.1	92.5
6	1000	200	10	99.6 ± 0.002	92.0	91.3
7	500	200	10	89.5 ± 0.017	90.6	90.0
8	**500**	**200**	**10**	**94.9 ± 0.015**	**95.1**	**94.7**
9	1000	50	10	99.7 ± 0.002	93.0	92.5

Classification accuracy values showed that the higher values were achieved using global features for both RF and SVM classifiers. The generalization results showed that RF had a better accuracy for global features, while SVM presented a better performs for local features. However, classification accuracy values for both RF and SVM are similar.

5.5 STRATEGY FOR BEAN CLASSIFICATION OF THE STATE OF THE ART

In order to compare the results of this project with a state-of-the-art method for bean classification, the strategy described in [10] was chosen. In [10] a classifier was proposed using a multilayer perceptron neural network (MLP) and color features for classifying ten bean varieties. The MLP network applied had four layers: an input layer, two hidden layers and an output layer. The input layer had a maximum of 12 neurons, one for each mean color parameter in the RGB and HSI color space for bean and spot extracted from the bean. The two hidden layers have different numbers of neurons, being 20 and 10, respectively. The output layer was equal to the number of classes, 10 neurons corresponding to the ten bean varieties.

This approach was implemented from scratch because the source code was not available. The K-means algorithm was used to obtain the predominant and secondary color in a bean seed which correspond to the color of the bean and its spots. Then, the color features described in [10] was obtained for the bean and spot areas. The MLP network was implemented using the same architecture described in [10], unlike the last layer which has 6 neurons corresponding to the six varieties analyzed in this research (see Figure 1). Table 12 shows the result of the experiments.

The higher accuracy was achieved by RF and SVM using the global and local features proposed in Section 4.3. The implemented strategy

Table 12: Results of classification using strategy of state-of-the-art

Strategy for classification	Acc (%)
MLP network implemented using the strategy [10]	87.3
RF with global features, $T = 200$ and $D = 10$	98.5
SVM with local features, $C = 1$ and codebook size of 1000	95.2

using the neural network described in [10] exhibits an accuracy of 87.3%. It indicates that both proposed approaches show significant improvement in the accuracy for classifying automatically the considered been varieties in comparison to the MLP based approach.

The results can be explained based on the features evaluated, in this project, color, shape and brightness were considered for global features and color, shape and texture for local features, whereas in [10] only color features were taken into account. Therefore, selecting the appropriate features to distinguish bean varieties with a similar appearance is an important factor.

6 SOFTWARE PROTOTYPE

The classification of a bean sample requires the execution of different stages such as segmentation, morphological operation, extraction of features and methods of classification which make it difficult to achieve by an unfamiliar user (bean inspector). In particular, the proposed approach for bean classification was implemented as a software prototype by developing the next requirements using an agile software development methodology:

1. As an inspector, I want to send a bean sample image to classify.
2. As an inspector, I want to consult the classification of the sample.
3. As an inspector, I want to consult the total number of seeds in the image.
4. As an inspector, I want to consult the phenotypic features of the seeds.
5. As an inspector, I want to download the report.

A high level architecture was proposed in order to develop a system that support the requirements above, see Figure 11. The technologies used in the development are described next:

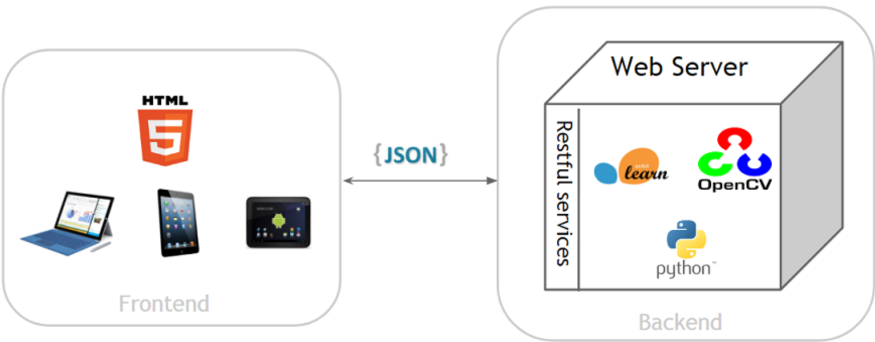


Figure 11: High level architecture diagram for web application.

1. **Frontend:** Web Client
   - *HTML5:* is used to design the web interface.
2. **Backend:** Web Server with the code for the different stages (segmentation, feature extraction, classification and report generation)
   - *Flask:* is a microframework for Python. It is used to create the web services that connect the user interface with the code in the web server.

- *Python:* is the programming language used to implement the different stages of the processing.
- *Sklearn:* is used for machine learning tasks using Python.
- *OpenCV:* is the library used for Image Processing and Computer Vision.

In particular, the user-friendly interface developed with HTML allows to a user to carry out all the stages of the bean sample analysis and generate a report in JSON format (see Figure 12). For this, the user selects, in the web interface, an image of beans to analyze and the classification method. After this information is sent, the classification process is performed, as follows: 1) The image is pre-processed in order to remove noise and each seed in the image is segmented. 2) Based on the set of features chosen by the user, global or local features are extracted from each individual seed. 3) The obtained feature vector is used to predict the bean variety applying the model obtained in the training stage, see Chapter 5. 4) A report is designed using JSON format to present the results to the user (see Figure 13).

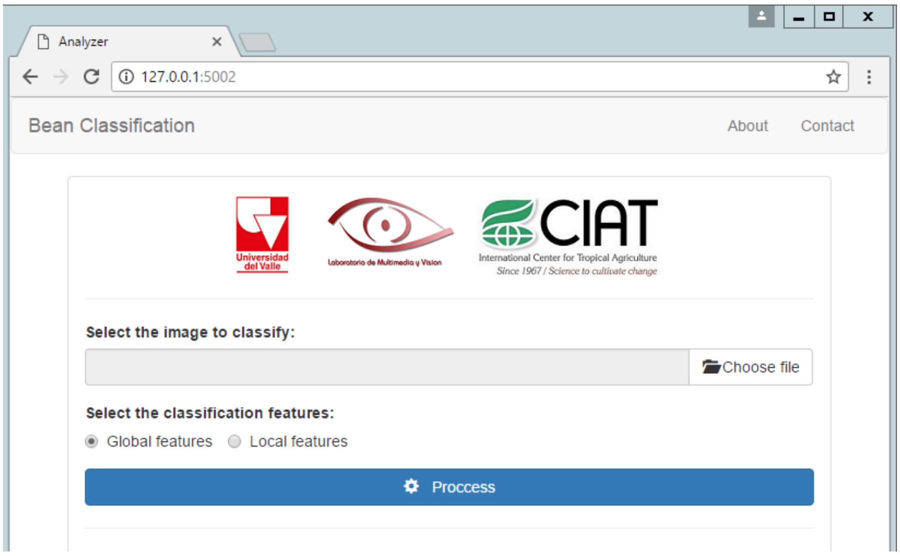


Figure 12: User interface of the bean classification system.

Download JSON


```
{
    global_features: {
        brightness: 77.43821200156724,
        color: {
            predominant: "RGB(87.07782692372946,58.43984385426634,54.69021086347781)",
            secundary: "RGB(119.71167121529339,94.56639065503599,82.95745861462913)"
        },
        shape: {
            area: 39830,
            circularity: 0.8302038166181402,
            elongation: 1.5813771670560983,
            majoraxis_length: 252.7462158203125,
            minoraxis_length: 202.06832885742188,
            perimeter: 776.8670920133591
        }
    },
    total_beans: 69,
    variety: "_bolaroja"
}
```


Figure 13: Illustration of the result of a bean classification.

7 CONCLUSIONS

An automatic bean classification system was proposed using supervised learning to classify six bean varieties. Local and global features were evaluated for building classification models. Ten global features were calculated based on characteristics of color, shape and brightness used during manual bean inspection. Local features were calculated by a BoF method. Each local feature was extracted using the SIFT, LBP and OSIFT descriptors, in addition, those descriptors were concatenated using three types of strategies: Early, Intermediate and Late fusion. Classification models were built based on SVMs and RFs classifiers.

In general, models using global features achieved a higher accuracy for the generalization test, over 98.5% with maximum F1 score of 98.4%, than classification models using local features (maximum accuracy of 95.2% and F1 score of 94.9%) and a codebook with 1000 visual words. In bean classification, global features are meaningful since they are defined taking into account the application domain. Additionally, classification accuracy using local features is affected by several factors such as the concatenation strategy and the number of combined features. In particular, RF showed better accuracy for the generalization test.

In the segmentation module, the WT algorithm was used for segmentation of images with “glued” seeds. This method works well in most of the cases.

In addition, a strategy for bean classification from the-state-of-the-art based on color features was implemented in order to classify the six bean varieties considered in this project and compare the results. Experiments showed that the proposed local and global characteristics achieved a better accuracy than the implemented strategy which exhibited an accuracy of 87.3%. It was observed that the use of appropriate features to represent the image content when the seeds have a similar appearance is crucial.

As a future work, a deep learning approach for classification will be evaluated.

BIBLIOGRAPHY

[1] O. Voysest, *Mejoramiento genético del frijol (Phaseolus vulgaris L.): legado de variedades de América Latina 1930-1999*. No. 321, Ciat, 2000.

[2] J. Arias, T. Rengifo, and M. Jaramillo, *Manual técnico: buenas prácticas agrícolas (BPA) en la producción de frijol voluble*. La selva, Medellín, Colombia: Print Ltda, 1st ed., 2007.

[3] A. Criminisi and J. Shotton, *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[4] C. Wallraven, B. Caputo, and A. B. Graf, "Recognition with local features: the kernel recipe.," in *ICCV*, vol. 3, pp. 257–264, 2003.

[5] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 524–531, IEEE, 2005.

[6] K. Kılıç, İ. H. Boyacı, H. Köksel, and İ. Küsmenoğlu, "A classification system for beans using computer vision system and artificial neural networks," *Journal of Food Engineering*, vol. 78, no. 3, pp. 897–904, 2007.

[7] X. Zhou, K. Yu, T. Zhang, and T. Huang, "Image classification using super-vector coding of local image descriptors," *Computer Vision–ECCV 2010*, pp. 141–154, 2010.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, pp. 1–2, Prague, 2004.

[9] S. Avila, N. Thome, M. Cord, E. Valle, and A. d. A. Araújo, "Bossa: Extended bow formalism for image classification," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2909–2912, IEEE, 2011.

[10] A. Nasirahmadi and N. Behroozi-Khazaei, "Identification of bean varieties according to color features using artificial neural network," *Spanish Journal of Agricultural Research*, vol. 11, no. 3, pp. 670–677, 2013.

[11] G. Venora, O. Grillo, C. Ravalli, and R. Cremonini, "Tuscany beans landraces, on-line identification from seeds inspection by image analysis and linear discriminant analysis," *Agrochimica*, vol. 51, no. 4-5, pp. 254–268, 2007.

[12] G. Venora, O. Grillo, C. Ravalli, and R. Cremonini, "Identification of italian landraces of bean (phaseolus vulgaris l.) using

an image analysis system," *Scientia horticulturae*, vol. 121, no. 4, pp. 410–418, 2009.

[13] B. Laurent, B. Ousman, T. Dzudie, M. M. Carl, and T. Emmanuel, "Digital camera images processing of hard-to-cook beans," *Journal of Engineering and Technology Research*, vol. 2, no. 9, pp. 177–188, 2010.

[14] S. A. De Araújo, J. H. Pessota, and H. Y. Kim, "Beans quality inspection using correlation-based granulometry," *Engineering Applications of Artificial Intelligence*, vol. 40, pp. 84–94, 2015.

[15] P. A. Belan, S. Araújo, and W. A. L. Alves, "An intelligent vision-based system applied to visual quality inspection of beans," in *International Conference Image Analysis and Recognition*, pp. 801–809, Springer, 2016.

[16] E. Ebrahimi, K. Mollazade, and A. Arefi, "An expert system for classification of potato tubers using image processing and artificial neural networks," *Int. J. of Food Eng*, vol. 8, no. 4, p. 9, 2012.

[17] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," 1979.

[18] V. Grau, A. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE transactions on medical imaging*, vol. 23, no. 4, pp. 447–458, 2004.

[19] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[20] E. Liu, K. Cashman, and A. Rust, "Optimising shape analysis to quantify volcanic ash morphology," *GeoResJ*, vol. 8, pp. 14–30, 2015.

[21] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2, pp. 1150–1157, Ieee, 1999.

[22] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[23] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[24] Z. Wang, B. Lu, Z. Chi, and D. Feng, "Leaf image classification with shape context and sift descriptors," in *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pp. 650–654, IEEE, 2011.

[25] S. O'Hara and B. A. Draper, "Introduction to the bag of features paradigm for image classification and retrieval," *arXiv preprint arXiv:1101.3354*, 2011.

[26] R. Burbidge and B. Buxton, "An introduction to support vector machines for data mining," *Keynote papers, young OR12*, pp. 3–15, 2001.

[27] N. Cristianini and B. Scholkopf, "Support vector machines and kernel methods: the new generation of learning machines," *Ai Magazine*, vol. 23, no. 3, p. 31, 2002.

Part II

APPENDIX

8 USER HISTORIES

	USER STORY
ID: 1	**User Story Name:** Classify bean seed
User: Inspector	**Iteration:** 1
Priority: High	
Description: As an inspector, I want to send a bean sample image to classify. The idea is to use an interface that allows sending a JPEG image to be classified.	
Status: Implemented	

Figure 14: User Story 1.

	USER STORY
ID: 2	**User Story Name:** Consult classification report
User: Inspector	**Iteration:** 2
Priority: High	
Description: As an inspector, I want to consult the classification of the sample. The idea is to show in the interface the result of the classification perform in *user story 1*. The name of the classified variety must be shown.	
Status: Implemented	

Figure 15: User Story 2.

USER STORY	
ID: 3	**User Story Name:** Consult total of seeds analyzed.
User: Inspector	**Iteration:** 2
Priority: High	
Description: As an inspector, I want to consult the total number of seeds in the image. The idea is to show in the interface the total seeds found in the image.	
Status: Implemented	

Figure 16: User Story 3.

USER STORY	
ID: 4	**User Story Name:** Consult phenotypic features
User: Inspector	**Iteration:** 2
Priority: High	
Description: As an inspector, I want to consult the phenotypic features of the seeds The idea is to show in the interface the extracted features of the seeds such as the color (primary and secondary), area, perimeter and brightness.	
Status: Implemented	

Figure 17: User Story 4.

	USER STORY
ID: 5	**User Story Name:** Download the results report
User: Inspector	**Iteration:** 2
Priority: Medium	
Description: As an inspector, I want to download the report. The idea is to download in JSON format the results of classification.	
Status: Implemented	

Figure 18: User Story 5.

9 PUBLICATIONS

As a result of this project the paper "Global and Local Features for Bean Image Classification" has been submitted to the 22nd Iberoamerican Congress on Pattern Recognition (CIARP 2017).