

OpenCV Basics: A Mobile Application to Support the Teaching of Computer Vision Concepts

Jose Sigut¹, Miguel Castro¹, Rafael Arnay¹, and Marta Sigut¹

Abstract—Contribution: Open Source Computer Vision Library (OpenCV) Basics is an application designed with the purpose of facilitating the initiation of industrial engineering students in the field of Computer Vision, making the learning process easier, more dynamic and more direct. To this end, an application has been developed for the Android operating system with which users can make use of a wide variety of algorithms available in the OpenCV library.

Background: Teaching topics related to Computer Vision can rely on the use of new technologies such as mobile applications. With this type of support, students can learn concepts that might otherwise be difficult to understand.

Intended Outcomes: The objective is to facilitate the assimilation of concepts related to Computer Vision by taking advantage of the camera and the processing power of a mobile device to observe in real time the effects produced on an image by many of the image processing algorithms included in OpenCV. This application is currently available to be downloaded for free through the Google Play Store so that anyone interested in the field of Computer Vision can make use of it.

Application Design: The proposed approach introduces students to concepts related to Computer Vision by making use of the developed application, complementing the theoretical contents taught by the teacher with specific examples.

Findings: The degree of satisfaction of OpenCV Basics users has been evaluated within the framework of the course advanced robotized systems, taught in the industrial engineering degree at the University of La Laguna.

Index Terms—Computer Vision, mobile application, Open Source Computer Vision Library (OpenCV).

I. INTRODUCTION

COMPUTER Vision [1] is one of the branches of computing that has experienced more remarkable growth in recent years. Both the detection of faces and objects in scenes are two clear examples of applications of artificial vision with greater impact. It should not go unnoticed that there are tools that constantly perform functions of analysis, interpretation, and manipulation of images. To give an example, in some social networks, such as Facebook, faces are automatically detected when an image is uploaded. In other cases, there are applications that allow to alter the original colors of an

image by applying different filters. Within this vast world of Computer Vision, there can be found different libraries that include powerful image processing algorithms, in order to help develop applications. Possibly, the most popular and best documented library for general-purpose image processing is Open Source Computer Vision Library (OpenCV) [2], a free open-source library. Among other interesting features, OpenCV is available to be downloaded on a variety of platforms, such as Windows, Linux, Android, MacOS, FreeBSD, and others.

In recent decades, Computer Vision has broken into the field of education and especially into graduate programs. In particular, in degrees, such as computer engineering and some of industrial engineering there are courses in which the problem of artificial vision is addressed. Generally, it is the student's first contact with this world, so the contents are relatively basic. Despite the underlying complexity of Computer Vision algorithms, these topics tend to be quite attractive for students because it is easy to take them to a practical level, and check their effects when applied to an image. This fact should be considered by the teacher when proposing a teaching–learning strategy that maximizes the academic performance achieved by the students.

Another issue that is often key in achieving the desired motivation in students is to integrate as much as possible the use of the technology, which is so present in their lives. In fact, this can not only contribute to involving students in the teaching–learning process but, in the particular case of the application presented in this article, also allows the execution of the algorithms seen in class both inside and outside the classroom. *Bebis et al.* [3] carried out an exhaustive and very interesting review of the state of Computer Vision in education. Although this article was published in 2010, many of the issues presented there are still valid because the authors dedicate a full section to innovative Computer Vision courses and a section within it to Computer Vision courses using the interactive technology.

In the world of Computer Vision in the field of education, there are numerous contributions. For example, *Reimer et al.* [4] presented a tangible interface to teach concepts related to Computer Vision to students who have little experience in programming. Nevertheless, while this can be useful for undergraduate students, in a graduate course it is necessary to understand the algorithms at a deeper level and to be able to program them. In [5], it is presented a project of a Computer Vision course for the detection and recognition of traffic signs. *Cielniak et al.* [6] proposed the integration of mobile robotics and Computer Vision. In [7], the goal is to introduce students to the main approaches of visual

Manuscript received July 30, 2019; revised January 13, 2020; accepted May 2, 2020. Date of publication May 29, 2020; date of current version October 29, 2020. (Corresponding author: Rafael Arnay.)

Jose Sigut, Rafael Arnay, and Marta Sigut are with the Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna Avda, 38204 San Cristóbal de La Laguna, Spain (e-mail: rarnayde@ull.es).

Miguel Castro is with DISA Corporación Petrolífera S.A., Área de Sistemas, 38003 San Cristóbal de La Laguna, Spain.

Digital Object Identifier 10.1109/TE.2020.2993013

object recognition and human face recognition using Computer Vision. Other recent contributions in the field of robotics are those presented in [8] and [9].

From another point of view, but also of great interest, Cote and Albu [10] presented an exploratory study of the essays of senior year engineering students on the social and cultural impacts of Computer Vision technologies.

There exist numerous examples of libraries or software tools for teaching Computer Vision. Some of them are: CVIPtools is a software package for the exploration of Computer Vision and image processing [11]; ImageJ is a public domain digital image processing program programmed in Java and developed at the National Institutes of Health [12]; Titere is a set of Laboratory Tests Remote Training by Means of Image Transmission Via Internet [13]; NeatVision is a Java-based image analysis and software development environment [14]; VIGRA is an image processing and analysis library that puts its main emphasis on customizable algorithms and data structures [15]; JavaVis is a Computer Vision library designed to be easy to use, both for launching and testing existing algorithms and for developing new ones [16]; and ImageNets is a large visual database designed for use in object recognition research [17].

The approach taken in this article differs substantially from the contributions cited above. OpenCV Basics is a mobile application designed to cover a wide range of parameterizable Computer Vision algorithms. It has been used to facilitate the initiation of students of an industrial engineering degree to the field of Computer Vision, specifically, in the advanced robotized systems (ARS) course. It is, therefore, an initiative that seeks to take advantage of the fact that today any university student has a mobile phone and is familiar with different types of applications. Having the possibility of executing Computer Vision algorithms on the mobile phone instead of having to do it on a desktop or laptop computer, reports certain benefits from an academic point of view. The greater freedom of movement that allows the use of mobile devices, makes it possible to consider exercises that involve the detection and tracking of mobile objects. Also, interactions in work groups can be further exploited when using this type of technology. For example, by having a member of the group hold an object and move it freely while the others must detect it, each of them adjusting the parameters of the algorithms independently in real time. In short, using educational applications on mobile devices allows to convert a theory class into a practical one during the period of time the teacher decides, keeping all students involved thanks to the ubiquity of these devices. As far as the authors know, there are no other mobile applications aimed at teaching concepts related to Computer Vision.

Among the secondary objectives of the application developed are those that may be useful for teaching or improve the user experience when using this application. The most relevant are the following.

- 1) Execute the algorithms in the photo mode, where the user can take a photograph and apply the algorithms offered by the OpenCV library, and in the image mode, where a static image can be loaded from the external storage of the device or via the Web.
- 2) Maintain compatibility with some of the previous versions of Android [18].
- 3) Adapt the application conveniently to ensure stability and speed in mobile devices.
- 4) Enhance the usability of the application by creating an intuitive interface.

This article is organized as follows. Section II describes some similar applications, Section III describes the course in which the application is used, Section IV describes the application itself, and Section V describes the usage of the presented application in the classroom. Finally, in Sections VI and VII, an evaluation of the application via a questionnaire and the conclusions are shown, respectively.

II. RELATED WORK

There seems to be a shortage of applications of this type. While it is true that there are applications for mobile devices that run some Computer Vision algorithms, their features are far from the development carried out by the authors. In particular, these applications do not implement the most basic algorithms or do not allow the user to make variations in the parameters of the implemented algorithms that help them understand their influence on the resulting image. Both features are basic for the students that begin in the field of Computer Vision.

Specifically, the applications available in Google Play Store [19] that have more points in common with the application presented in this article are OpenCV Samples [20], OCVED [21], DNN [22], Canny: Open CV Camera & Filter Camera Photo Editor [23], and OpenCV for Unity Example [24]. Concerning OpenCV Samples, it is an application that is characterized by allowing the processing of data in real time, by having several advanced image manipulation algorithms and by implementing complex examples of the OpenCV tutorial. However, taking into account the above, it also has important shortcomings, such as not allowing basic algorithms to be executed, modifying the input parameters of the algorithms, using the front camera of the terminal, or using static images, photographs, or loaded images. OCVED allows to perform some functions related to edge detection only. DNN uses OpenCV for the recognition of objects as help for people with reduced visibility. OpenCV for Unity implements some of the most complex openCV algorithms but does not implement the most basic ones, necessary for teaching Computer Vision in an introductory course. Canny: Open CV Camera & Filter Camera Photo Editor, contains a cumbersome amount of filters but many of them have nothing to do with the OpenCV basic algorithms. It is important to remark that none of these applications implement a wide set of operations as the application presented in this article does, nor they are intended for educational purposes.

III. COURSE DESCRIPTION

The ARS course is a part of the syllabus of the four-years degree in industrial electronics and automation at the University of La Laguna. It is a six ECTS course given over one 15-week term with a weekly 2-h theory lecture and a 2-h

practical session. The course is taught in the second semester of the fourth academic year as a continuation of the third-year course robotized systems. Therefore, the students attending the ARS course (average age of 22) are expected to have some basic skills in robotic manipulation, kinematics, sensors and actuators, and mobile robotics. The content of ARS is divided into two main parts: the first part (first half of the term) about the principles of Computer Vision and a second part more focused on robotics itself. The skills achieved by the students at the completion of the course are the following.

- 1) Familiarity with hardware and software of a Computer Vision system.
- 2) Application of Computer Vision techniques in the field of robotics.
- 3) More advanced skills in robotics, including robot programming, and dynamics and control.

The first part of the course follows the classical approach to Computer Vision as a sequence of stages in which the image is processed at different levels. At the pixel level, the image is enhanced for better quality by applying different types of filters and increased contrast through an equalization of the histogram. Once enhanced, low-level features, such as color or edges, are extracted. These low-level features are the basis for segmentation algorithms, such as thresholding, Hough transform, watershed, and others, to generate more semantically significant regions. Morphological operations can be useful for postprocessing the segmented image which is used as the basis for high-level tasks such as object recognition. According to this description, this part is structured as follows.

- 1) *Introduction to Computer Vision*: 2-h theory.
- 2) *Image Enhancement (Filtering and Histogram Equalization)*: 2-h theory + 2-h practical work.
- 3) *Feature Extraction (Color and Edge Detection)*: 3-h theory + 3-h practical work.
- 4) *Image Segmentation (Thresholding, Hough Transform, and Watershed)*: 4-h theory + 4-h practical work.
- 5) *Morphological Processing*: 2-h theory + 2-h practical work.
- 6) *Introduction to Object Recognition*: 3-h theory + 4-h practical work.

The theoretical lectures are intended for the explanation of the basic concepts and this is where the presented application plays a prominent role as it will be detailed in the next sections. The practical work consists of several programming sessions with the Octave language that allows to implement the different techniques explained easily and quickly. The evaluation of this part of the course consists of a final test which includes a questionnaire about basic concepts of Computer Vision and a series of programming exercises in Octave that address the different topics seen in class.

In the second part of the course, an introduction to robot programming is made and concepts related to the kinematics and dynamics of manipulative robots are explained. Specifically, the theoretical sessions are dedicated to explaining different types of robot control, such as kinematic, dynamic, and visual control. According to this description, this part is structured as follows.

TABLE I
ALGORITHMS IMPLEMENTED FOR THE PREPROCESSING OF
THE IMAGE AND CONFIGURABLE PARAMETERS

Algorithm	Parameters
Blur	Kernel Size
Gaussian Blur	Kernel Size
Median Filter	Kernel Size
Bilateral Filter	Kernel Size
Histogram	Channels (R,G,B)
Remapping	-
Histogram Equalization	-

TABLE II
ALGORITHMS IMPLEMENTED FOR THE MORPHOLOGICAL
OPERATIONS AND CONFIGURABLE PARAMETERS

Algorithm	Parameters
Erosion	Kernel Size Kernel Shape
Dilation	Kernel Size Kernel Shape
Extended Morphological Operations: Opening, Closing, TopHat, BlackHat	Kernel Size Kernel Shape

- 1) *Introduction to Robot Programming*: 2-h theory.
- 2) *Cinematic Control*: 2-h theory.
- 3) *Introduction to Robot Dynamics*: 1-h theory.
- 4) *Dynamic Control*: 5-h theory.
- 5) *Visual Control*: 4-h theory.

In the practical sessions, where the students put into practice the concepts related to Computer Vision learned in the first part of the course, robots are programmed using the ACL language, and the OpenCV library to carry out tasks related to visual control such as the manipulation of objects depending on their shape or color. This project is developed over seven weeks (14 h) as related theoretical concepts are explained in the theory sessions.

IV. APPLICATION

The OpenCV Basics application implements more than 20 Computer Vision algorithms. The execution of some of them is practically immediate through a call to the OpenCV library. In other cases, it is necessary to perform a series of operations and/or configurations. The Computer Vision algorithms implemented in the application have been grouped into four major categories, which are as follows:

- 1) image preprocessing;
- 2) morphological operations;
- 3) edge detection;
- 4) image segmentation.

Tables I–IV show a summary of the algorithms implemented within each of the aforementioned categories. In some cases, the algorithms included in the OpenCV library can have a considerable number of parameters, something that can be counterproductive from a didactic point of view in an introductory course to Computer Vision. In order to simplify the interaction by users, it was decided to limit the modifiable

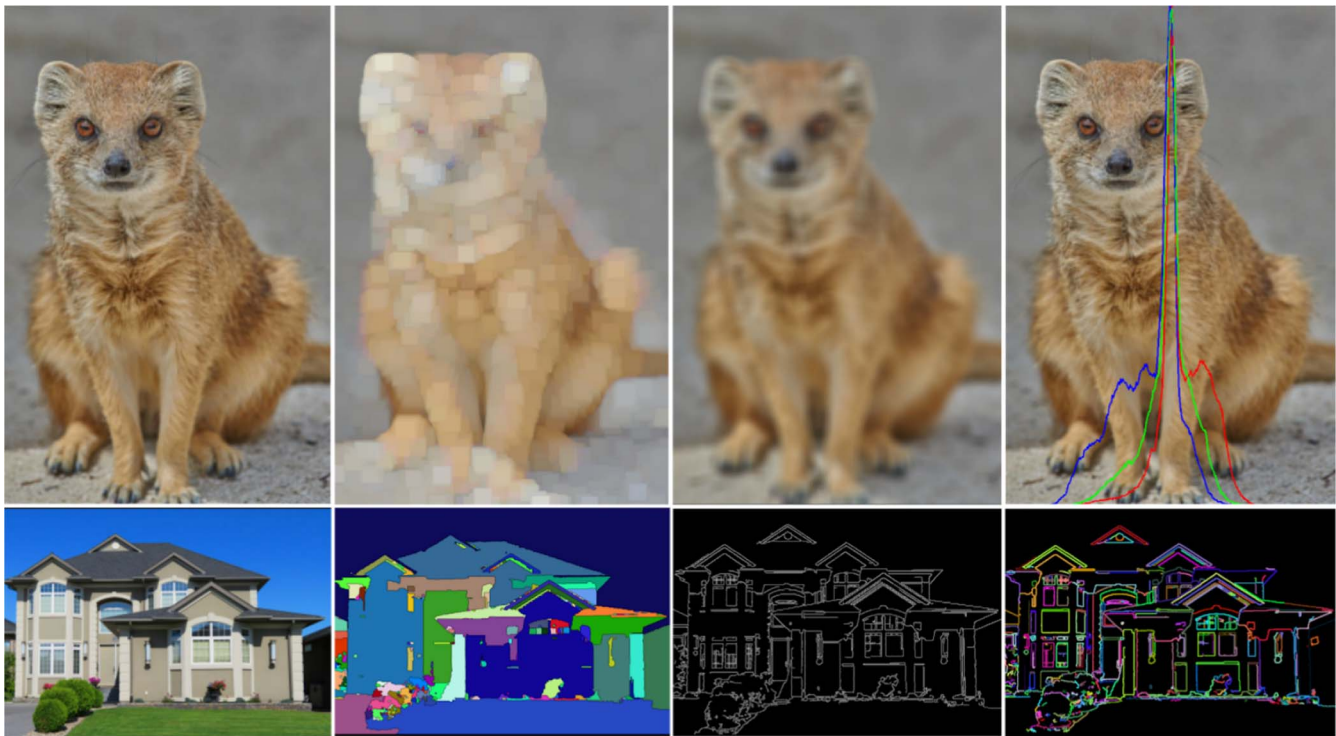


Fig. 1. Execution examples of some algorithms included in the application. The images on the left of both the upper and lower rows correspond to the original images. The upper row shows the results of applying, from left to right, a dilation operation, a Gaussian smoothing, and a histogram analysis. In the lower row, a watershed algorithm, a Canny edge detector, and a contour detection have been applied from left to right.

TABLE III
ALGORITHMS IMPLEMENTED FOR THE EDGE DETECTION AND CONFIGURABLE PARAMETERS

Algorithm	Parameters
Sobel	Kernel Size Edge Type Derivative Order
Canny	Kernel Size Edge Type Lower Threshold
Laplace	Kernel Size

TABLE IV
ALGORITHMS IMPLEMENTED FOR THE IMAGE SEGMENTATION AND CONFIGURABLE PARAMETERS

Algorithm	Parameters
Thesholding	Threshold limit Threshold Type
Contours	Threshold limit
Convex Hull	Threshold limit
Box and Ellipse Fitting	Threshold limit
Hough Transform (line)	Threshold limit
Hough Transform (circle)	Threshold limit Min. distance between circles Min.and Max circle radii
Watershed	Binary Image Threshold limit Peaks Threshold limit

parameters to those that have a more direct influence on the execution of each algorithm. In order for the user to modify the value of these parameters in a fast and intuitive way, a sliding bar widget has been used. In some of the algorithms included in the application, there are parameters that can trigger errors if they are not configured correctly. In order to manage this type of situation, the following criteria were adopted.

- 1) If the change in a parameter results in an exception thrown by the library, then the last valid value with which everything worked correctly is used.
- 2) In case of error, the current parameter is marked in red, indicating to the user that the change made is not adequate. Once a correct value is indicated, the red color disappears.

During the development of the OpenCV Basics application, it was considered that it would be interesting to provide it with some functionalities not contemplated in the default behavior

of the OpenCV library for mobile devices. One of them is the resolution change of the preview, which will be further commented in the next section, devoted to the interface. The other refers to the change in the orientation of the preview.

By default, the OpenCV library is limited to displaying the frames in the horizontal format. The modification introduced by the authors allows to detect the orientation of the device to determine how the frame should be drawn, whether in a vertical or horizontal mode. Although it may seem unimportant, the orientation of the device plays an important role in the way in which the elements are presented on the screen and in the ease of the user to manipulate the terminal. Taking into account that the settings menu can show more elements, if the

device is vertical, it is more likely that in this way the user does not have to scroll on the screen or that the scene itself is partially hidden by other elements of the screen.

Fig. 1 shows the results of the execution of some algorithms using the preloaded images available in the application.

A. Interface of the Application

To facilitate user navigation through the different menus and options of the application, a simple and intuitive interface has been implemented, which allows a fluid interaction.

The main screen of the application, whose appearance is shown in Fig. 2, consists of a menu with five elements.

- 1) *Select Algorithm With a Camera*: This option asks the user to choose an algorithm and then prepares the application to run in a real-time mode. The frames are obtained directly from the camera. This is the preferred option and is shown with a larger button than the rest of the options.
- 2) *Load Image*: It requests the user to choose an algorithm and then opens a file browser to choose the image file to load from the external storage of the device.
- 3) *Load Image From Web*: It requests the user to choose an algorithm and then a dialog window appears where the URL to the image must be entered.
- 4) *Example Images*: They request the user to choose an algorithm and then offer a series of sample images to choose from.
- 5) *Help*: It offers help on the use of the application and shows the credits of the development.

When selecting the main menu option “Use algorithm with a camera,” the user will be directed to the algorithm type selection menu to be used [Fig. 3(left)]. Once a type has been selected, a submenu will be displayed that will allow selecting the specific algorithm to execute [Fig. 3(right)].

Once a specific algorithm is selected, it runs. Depending on the previously chosen option, the algorithm will process an image captured in real time with the front camera of the device (default behavior), or it will process a static image previously chosen through the main menu. At any time during the execution, the user can open a settings menu by touching the icon in the upper left corner of the screen (Fig. 4). This menu allows the user, on the one hand, to change the general settings of the execution and, on the other hand, to change some specific parameters of the algorithm that is being executed.

Within the general execution settings, the following options can be selected.

- 1) *Photo Mode*: It allows the user to take a photograph with the camera of the device for processing.
- 2) *Real-Time Mode*: The algorithm processes captured images through any of the cameras of the device.
- 3) *Change Camera*: It allows the user to select through which camera (front or rear) images are going to be captured.
- 4) *Change Resolution*: It allows to vary the resolution of the image to be processed. This is useful, above all, in devices with a limited computing power, since it allows to process images at a lower resolution than that captured

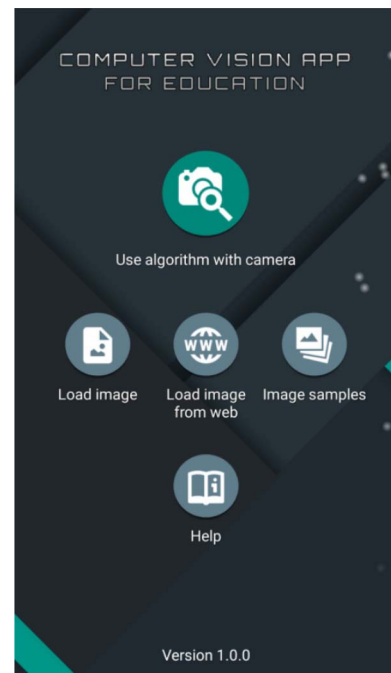


Fig. 2. Main menu of the application.



Fig. 3. Menus for the selection of the type of the vision algorithm to execute.

through the camera, thus maintaining the operation in real time.

- 5) *Change the Color Space*: It switches between a color space (RGB) and a grayscale space.

The flow diagram in Fig. 5 shows the general operation of the application from the moment the user selects the source of the image to be processed (camera of the mobile device, from the Web, and from an external storage device or images of example) until the chosen algorithm is executed.

V. OPENCV BASICS IN THE CLASSROOM

As it was pointed out in Section III, the main aim of OpenCV Basics in the context of the ARS course is to provide

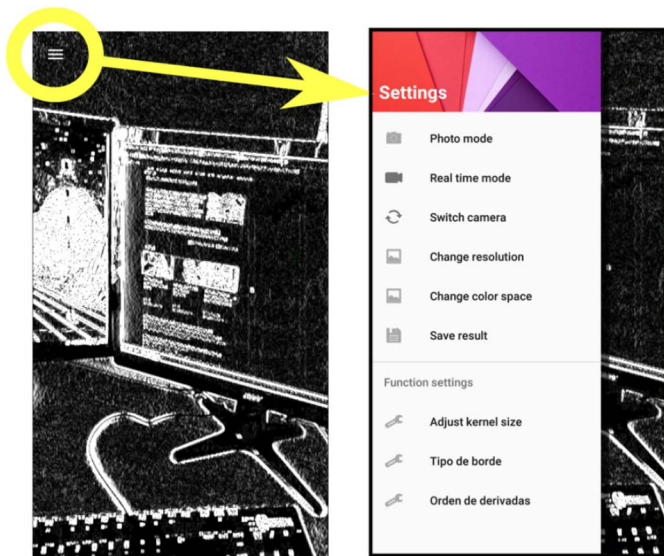


Fig. 4. Context menu to define the settings of the real-time vision algorithm.

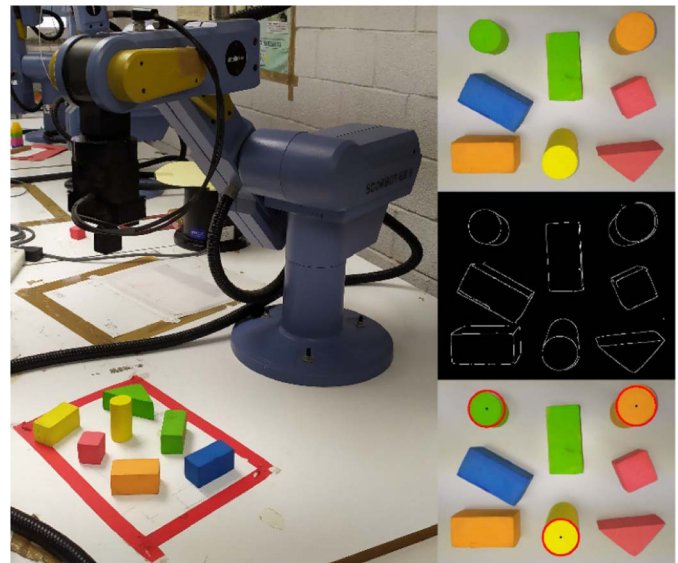


Fig. 6. Sample of the configuration that students will find in the laboratory and the type of forms that they will have to recognize through Computer Vision algorithms. In this example, circular forms are detected.

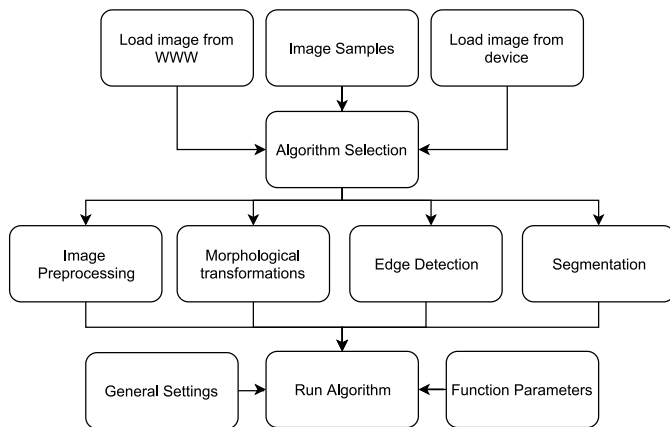


Fig. 5. Flowchart of the internal operation of the application.

support for the teaching of Computer Vision concepts to complement the theoretical lectures. In this respect, it is important to remark that although the use of the Octave language allows to implement the techniques of Computer Vision on different images and vary their respective parameters, the possibility of experimenting in real time, on the fly, with OpenCV Basics, offers the students a totally different and enriching experience due to its immediacy and versatility. This allows students to develop the required skills, working in a nonstructured learning environment where no other computational resources are available [25].

Even though the use of a mobile application in the classroom can be seen at first as something attractive for the students, it is very important to carefully design the activities to be carried out so that it does not turn into a distraction. With that in mind, after finishing the explanation of each of the theoretical modules mentioned in Section III (with the exception of the introduction and object recognition modules), 15–20 min are devoted to let the students interact with their mobile phones. During that time, two activities are proposed to the students but, first, they are asked to form groups of three to

four people to encourage the collaborative work and to involve those students who do not have an Android terminal.

The first activity consists of testing the corresponding Computer Vision algorithms in real time on any object in the classroom, including the students themselves using the front or rear camera of the mobile phone. The students are allowed to freely move around in the classroom and vary the parameters of the methods to obtain an immediate feedback. This first activity leads to a gentle approach to the concepts just explained, in a practical and intuitive way. The second activity is more closely related to the laboratory work of the second part of the course, in which the students work in a project involving the detection and manipulation of objects with a robotic arm using a webcam. As it was mentioned in Section III, the Computer Vision background acquired by the students in the first part of the course is essential for the practical aspects of the second part. For that purpose, some pieces with different geometric shapes and colors, as shown in Fig. 6, are delivered to each group of students. Under the supervision of the professor, the same algorithms which were tested in the first activity are now used with these pieces. The objective of this second activity is the correct adjustment of the parameters for the problem at hand as opposed to the first activity, where the students were more autonomous to set them freely with no specific objective in mind. The versatility of using a mobile phone in this context allows to simulate scenarios similar to what the students will find in the laboratory such as a piece grabbed by a member of the group as if it was a robotic arm or a piece placed on different backgrounds to check the robustness of the methods under consideration.

VI. EVALUATION

Once the course was finished, the students were asked to fill out a questionnaire voluntarily, the results of which are shown in Table V. Of the total of 35 students in the class,

TABLE V
RESULTS (MEAN, STANDARD DEVIATION, AND 95% CONFIDENCE INTERVAL) OF THE QUESTIONNAIRE AIMED AT KNOWING TO WHAT EXTENT THE APPLICATION PRESENTED IN THIS ARTICLE HAS BEEN USEFUL TO THE STUDENTS

Question	Mean	Std	95% CI
OpenCVBasics has helped you to better understand the concept of a histogram of an image	4.45	0.72	[4.15-4.76]
OpenCVBasics has helped you to better understand the effect on the image of a Gaussian smoothing filter	4.62	0.71	[4.32-4.92]
OpenCVBasics has helped you to better understand the effect on the image of an edge detector (Sobel, Laplace and Canny)	4.50	0.78	[4.17-4.82]
OpenCVBasics has helped you to better understand the segmentation method based on thresholding	4.37	0.76	[4.04-4.70]
OpenCVBasics has helped you to better understand the segmentation method based on the Hough transform (lines and circles)	4.08	0.93	[3.70-4.47]
OpenCVBasics has helped you to better understand the effect on the image of morphological operators (erosion and dilation)	4.04	1.04	[3.60-4.48]
The openCVBasics application is easy to use	4.37	0.82	[4.02-4.72]
The openCVBasics application complements the theoretical explanations of the teacher	4.70	0.69	[4.41-5.00]
The application openCVBasics has helped, in general, a better understanding of the different processing operations of the image seen in class	4.62	0.71	[4.32-4.92]
I would recommend the extension of the openCVBasics application for the teaching of other concepts and operations related to image processing	4.62	0.57	[4.38-4.86]
I would recommend continuing with the use of the openCVBasics application in future editions of the course	4.62	0.77	[4.30-4.95]

24 students answered the questionnaire (all of them passed the course). The 11 questions were aimed at knowing to what extent the application presented in this article has been useful and has helped them to acquire a greater mastery of the concepts studied in the course. The results are shown on a five-point Likert scale [26].

Based on the global average score (4.46) given by the students participating in the questionnaire to the 11 questions raised, the degree of general satisfaction with the OpenCV Basics application is quite high. The students consider that the application has helped them to better understand the effects of the included algorithms, although the worst evaluated were the morphological operators (erosion and dilation) and the segmentation method based on the Hough transform (lines and circles). Even so, these two issues have obtained a score above four: 4.04 and 4.08, respectively. On the other hand, when the students were asked to what extent the application complements the theoretical explanations of the teacher, the score awarded by these is the highest: 4.70.

VII. CONCLUSION

One of the biggest challenges faced by university teachers every day is to find the most appropriate teaching methodology to achieve, not only the best possible understanding of the concepts and techniques they teach but also to motivate and involve students in the process of learning. Undoubtedly, the use of new technologies can be a great ally in reaching these goals. OpenCV Basics was born, precisely, as the result of this reflection. For students of industrial engineering, it is important to know the operation of basic Computer Vision algorithms and the effect they have when applied to a specific image. However, the teaching methodology consisting of master classes complemented by computer practices has not yielded the desirable academic results. With the purpose of

trying to improve them, an application for mobile phones was devised with which the students could at any time run some algorithms ranging from basic to more advanced ones. In view of the results shown in the "Evaluation" section, the authors consider that OpenCV Basics complies quite well with the initially proposed objectives, since the evaluation of the users is quite good. In particular, the students who answered the satisfaction survey consider that the application complements the theoretical explanations of the teacher, giving this question a score of 4.70 out of 5, which makes it the best valued.

The questionnaire should not only serve to measure the degree of student satisfaction with the developed application but also to try to improve the aspects of it that have obtained a lower rating. In this sense, the authors believe that it would be convenient, for some of the techniques implemented, to prepare a scenario in which the effect of the artificial vision algorithms implemented, especially the morphological operators and the segmentation method based on the Hough transform, can be more clearly appreciated. In addition, OpenCV Basics is likely to undergo certain improvements, such as include more advanced image processing algorithms, such as recognition of faces or people; add to the application sections of information where the operation of the algorithms is summarized; give the possibility to record videos or execute algorithms on the frames of a video saved in the device, etc.

REFERENCES

- [1] R. Szeliski, *Computer Vision: Algorithms and Applications*. London, U.K.: Springer, 2010.
- [2] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision With the OpenCV Library*. Farnham, U.K.: O'Reilly Media, 2008.
- [3] G. Bebis, D. Egbert, and M. Shah, "Review of computer vision education," *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 2-21, Feb. 2003.
- [4] P. Reimer, A. B. Albu, and G. Tzanetakis, "Raydiance: A tangible interface for teaching computer vision," in *Proc. Int. Symp. Vis. Comput.*, 2011, pp. 259-269.

- [5] D. Gerónimo, J. Serrat, A. M. López, and R. Baldrich, "Traffic sign recognition for computer vision project-based learning," *IEEE Trans. Educ.*, vol. 56, no. 3, pp. 364–371, Aug. 2013.
- [6] G. Cielniak, N. Bellotto, and T. Duckett, "Integrating mobile robotics and vision with undergraduate computer science," *IEEE Trans. Educ.*, vol. 56, no. 1, pp. 48–53, Feb. 2013.
- [7] J. P. Vital, N. F. Ferreira, A. Valente, V. Filipe, and S. F. Soares, "Learning computer vision using a humanoid robot," in *Proc. IEEE Global Eng. Educ. Conf. (EDUCON)*, Dubai, UAE, 2019, pp. 639–645.
- [8] L. Ma and M. Alborati, "Enhancement of a VEX robot with an onboard vision system," in *Proc. IEEE 10th Int. Conf. Eng. Educ. (ICEED)*, Kuala Lumpur, Malaysia, 2018, pp. 117–121.
- [9] H. Cao, "Robot-based motion detection method and its application in PE practice teaching," *Int. J. Contin. Eng. Educ. Life Long Learn.*, vol. 29, nos. 1–2, pp. 87–100, 2019.
- [10] M. Cote and A. B. Albu, "Teaching socio-cultural impacts of technology in advanced technical courses: A case study," *Eur. J. Eng. Educ.*, vol. 44, no. 5, pp. 688–701, 2019.
- [11] *CVIptools*. Accessed: Dec. 29, 2019. [Online]. Available: <https://cviptools.ece.siu.edu>
- [12] *ImageJ*. Accessed: Dec. 29, 2019. [Online]. Available: <https://imagej.nih.gov/ij/index.html>
- [13] *Titere*. Accessed: Dec. 29, 2019. [Online]. Available: <http://www.disam.upm.es/vision/Vision2/projects/titere/index.html>
- [14] *NeatVision*. Accessed: Dec. 29, 2019. [Online]. Available: <http://neatvision.eeng.dcu.ie./index.html>
- [15] *VIGRA*. Accessed: Dec. 29, 2019. [Online]. Available: <http://ukoethe.github.io/vigra/>
- [16] M. Cazorla and D. Viejo, "JavaVis: An integrated computer vision library for teaching computer vision," *Comput. Appl. Eng. Educ.*, vol. 23, no. 2, pp. 258–267, 2015.
- [17] *ImageNets*. Accessed: Dec. 29, 2019. [Online]. Available: <http://imagenets.sourceforge.net/>
- [18] R. Meier, *Professional Android 4 Application Development*. Indianapolis, IN, USA: Wiley, 2012.
- [19] *Google Play Store*. Accessed: Jul. 17, 2019. [Online]. Available: <https://play.google.com/store>
- [20] *OpenCV Samples*. Accessed: Jul. 17, 2019. [Online]. Available: https://play.google.com/store/apps/details?id=com.jnardari.opencv_androidsamples
- [21] *OCVED*. Accessed: Jul. 17, 2019. [Online]. Available: <https://play.google.com/store/apps/details?id=com.andreyaleev.opencvfilters>
- [22] *DNN*. Accessed: Jul. 17, 2019. [Online]. Available: https://play.google.com/store/apps/details?id=org.opencv.sample.opencv_mobilenet
- [23] *Canny: Open CV Camera & Filter Camera Photo Editor*. Accessed: Dec. 31, 2019. [Online]. Available: <https://play.google.com/store/apps/details?id=app.bhupesh.armorking.canny&hl=en>
- [24] *OpenCV for Unity Example*. Accessed: Dec. 31, 2019. [Online]. Available: <https://play.google.com/store/apps/details?id=com.enoxsoftw.are.opencvforunityexample&hl=en>
- [25] M. Ben-Ari, "Constructivism in computer science education," *ACM SIGCSE Bull.*, vol. 30, no. 1, pp. 257–261, 1998.
- [26] R. Likert, "A technique for the measurement of attitudes," in *Archives Psychology*. New York, NY, USA: Sci. Press, 1932.

Jose Sigut was born in Santa Cruz de Tenerife, Spain, in 1970. He received the Bachelor of Science degree in physics and the Ph.D. degree from the University of La Laguna, San Cristóbal de La Laguna, Spain, in 1993 and 2002, respectively.

Since 2009, he has been an Associate Professor with the Department of Engineering Informatics and Systems, University of La Laguna and his main line of research is computer vision with biomedical applications. In this area, he has participated in different projects and has numerous publications in international conferences and journals.

Miguel Castro was born in La Oliva, Fuerteventura, in 1991. He received the graduation degree in computer science from the University of La Laguna, San Cristóbal de La Laguna, Spain, in 2018.

He is currently working as an Application Technician with DISA Corporación Petrolífera S.A, San Cristóbal de La Laguna.

Rafael Arnay was born in 1982. He received the M.Sc. and Ph.D. degrees (Hons.) in computer science from the University of La Laguna, San Cristóbal de La Laguna, Spain, in 2007 and 2014, respectively.

He joined the Department of Computer Science and Systems, University of La Laguna in 2007, where he is currently an Associate Professor. His current research interests include robotics, computer vision, and deep learning.

Marta Sigut was born in Santa Cruz de Tenerife, Spain, in 1973. She received the M.S. degree in applied physics and the Ph.D. degree from the University of La Laguna, Tenerife, Spain, in 1996 and 2002, respectively.

She currently works as a Professor with the University of La Laguna. Her areas of interest include control systems, robotics, and pattern recognition techniques.