

Artificial vision system for object classification in real time using Raspberry Pi and a Web camera

Sistema de visión artificial para clasificación de objetos en tiempo real usando Raspberry Pi y una cámara web

SERRANO-RAMÍREZ, Tomás†\*, LOZANO-RINCÓN, Ninfa del Carmen, MANDUJANO-NAVA, Arturo and SÁMANO-FLORES, Yosafat Jetsemaní

Universidad Politécnica de Guanajuato, Ingeniería Automotriz

ID 1<sup>st</sup> Autor: Tomás, Serrano-Ramírez / ORC ID: 0000-0001-6118-3830, Researcher ID Thomson: G-6039-2018, CVU CONACYT ID: 493323

ID 1<sup>st</sup> Co-author: Ninfa del Carmen, Lozano-Rincón / ORC ID: 0000-0002-8347-1240, CVU CONACYT ID: 348911

ID 2<sup>nd</sup> Co-author: Arturo, Mandujano-Nava / ORC ID: 0000-0003-2022-4397, CVU CONACYT ID: 270254

ID 3<sup>rd</sup> Co-author: Yosafat Jetsemaní, Sámano-Flores / ORC ID: 0000-0003-4173-6236, CVU CONACYT ID: 444850

DOI: 10.35429/JITC.2021.13.5.20.25

Received March 10, 2021; Accepted June 30, 2021

Abstract

Computer vision systems are an essential part in industrial automation tasks such as: identification, selection, measurement, defect detection and quality control in parts and components. There are smart cameras used to perform tasks, however, their high acquisition and maintenance cost is restrictive. In this work, a novel low-cost artificial vision system is proposed for classifying objects in real time, using the Raspberry Pi 3B + embedded system, a Web camera and the Open CV artificial vision library. The suggested technique comprises the training of a supervised classification system of the Haar Cascade type, with image banks of the object to be recognized, subsequently generating a predictive model which is put to the test with real-time detection, as well as the calculation for the prediction error. This seeks to build a powerful vision system, affordable and also developed using free software.

Resumen

Los sistemas de visión por computadora son parte esencial en tareas de automatización industrial tales como: identificación, selección, medición, detección de defectos y el control de calidad en partes y componentes. Existen cámaras inteligentes empleadas para la realización de dichas tareas, sin embargo, su alto costo de adquisición y mantenimiento es privativo. En este trabajo, es propuesto un novedoso sistema de visión artificial de bajo costo, para la clasificación de objetos en tiempo real, mediante el sistema embebido Raspberry Pi 3B+, cámara Web y la librería de visión artificial Open CV. La técnica sugerida comprende el entrenamiento de un sistema de clasificación supervisado del tipo Haar Cascade, con bancos de imágenes propios del objeto a reconocer, generando posteriormente un modelo predictivo el cual se pone a prueba con la detección en tiempo real, así como el cálculo en el error en la predicción, Con esto se busca la construcción de un sistema de visión poderoso, de costo accesible y que además sea desarrollado mediante software libre.

Computer vision, Raspberry Pi, OpenCV

Visión artificial, Raspberry Pi, OpenCV

Citation: SERRANO-RAMÍREZ, Tomás, LOZANO-RINCÓN, Ninfa del Carmen, MANDUJANO-NAVA, Arturo and SÁMANO-FLORES, Yosafat Jetsemaní. Artificial vision system for object classification in real time using Raspberry Pi and a Web camera. Journal of Information Technologies and Communications. 2021. 5-13: 20-25

† Researcher contributing as first autor

## Introduction

One of the pillars of Industry 4.0 is computer vision, which is revolutionizing the way automation and quality control systems are carried out in manufacturing processes. Tasks such as the identification, selection, measurement and detection of parts or components are seen in a daily example of applications in this field.

In the industrial field, it is still common the use of expensive cameras, specifically designed for artificial vision tasks, these cameras have specialized hardware and software for processing part or the entire information on site, Compensating the modest performance of the computer equipment or embedded systems to which information is sent from the camera.

Today the roles have changed, with the availability of computers and embedded systems with higher performance, reduced size and competitive cost, allowing the use of ordinary cameras. It moves the video processing, from the camera to the embedded system or the computer. without compromising speed and real-time response. It represents a great advance in the field of artificial vision, allowing ordinary video cameras such as those found in smartphones or personal computers (webcams) to achieve results that were previously reached by cameras with special lenses, optic filters, depth sensors, proprietary processing hardware, among other additions.

The proposed system follows this approach, using as sensor an inexpensive web camera, with a resolution of 1.3 Mega pixels and data transmission via USB 2.0. For video processing is used the Raspberry Pi 3B + embedded system, a low-cost bank card size single board computer with a 1.4 GHz 64 bit processor, 1GB ddr2 ram memory, four USB 2.0 ports, HDMI, WiFi, Bluetooth, 40 GPIO input and output pins, among other features. Another advantage of this work is the use of the artificial intelligence algorithm known as Haar Cascade, a supervised classifier that has proven an outstanding performance, in embedded systems for face recognition. For the development of the Artificial vision system for object classification in real time, Python3 programming language, OpenCV computer vision library and Raspberry OS operating system were used, giving priority to the use of free software tools.

The field of artificial intelligence is strongly linked to the development of computer vision applications. There are learning algorithms which can be applied to video captured in real time, which can detect and classify objects for which they were trained. This has led to a number of applications in manufacturing processes, safety and even in the automotive sector with the development of autonomous vehicles.

There are two types of learning algorithms: supervised and unsupervised. Supervised learning consists of the generation of a predictive model through the analysis of examples in which the answer is known. In the case of computer vision, the examples can be large image banks, in which the object to be detected is found in different positions, types of environments and lighting. The greater number of images under different conditions, the more robust the predictive model becomes, always trying to find a balance between computational cost and prediction error.

The Haar Cascade classifier proposed by Paul Viola and Michael Jones (Viola & Jones, 2001) is from a supervised type. Unlike other classifiers whose computational cost is considerable in order to achieve acceptable performance, Haar Cascade has shown good results at a lower computational cost. This has led to its implementation in smartphones, single board computers, and embedded systems which generally sacrifice performance in their portability feature.

The specific objective of this research is to implement a classifier with the aforementioned hardware and software characteristics, which distinguishes a specific type of immersed part among other types, to later label it, regardless of its size or position. In this case, the classification of hexagonal nuts was arbitrarily chosen, which will be found immersed in screws and washers of different sizes, simulating a classic classification problem in an industrial environment.

## Classification system development

The implementation methodology for the artificial vision system applied for object classification in real time is described below.

This involves three successive steps: the first corresponds to the capture of the image database with the object to be classified, the second implies the training and generation of the predictive algorithm through the database and the Haar Cascade classifier. Finally, the real-time detection test is performed, as well as the measurement of the classification error.

Obtaining the image database

The first step to carry out this work is to generate an image database for training the Haar Cascade classifier. The database must contain a great number of images and for this reason a program in Python and Open CV to carry out the compilation in a most efficient and simple way was implemented. A rectangular frame has been delimited and the object whose image must be captured must be placed inside it. If a key is pressed, the image inside the frame is saved and resized to 50x60 pixels. The reduction is carried out in order not to extend the training process too long.

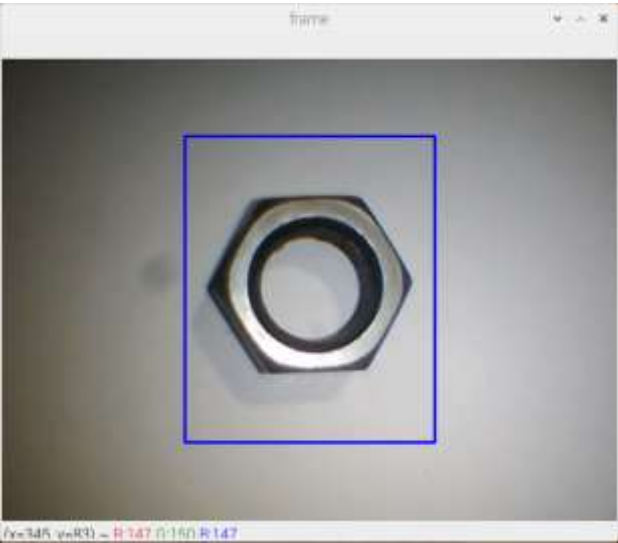


Figure 1 Capturing an image for the training database

The Haar Cascade classifier requires an image database separated into two large groups: the first corresponds to the positive images, in which the object to be recognized (class nut) is found, trying to vary the size of the object, the position and if possible, environments similar to those found in the classification scenarios.



Figure 2 Database sample with positive images.

The second group competes for negative images, which correspond to all objects and backgrounds that do not belong to the nut class and which may be found in future classification scenarios, including possible false positives such as washers and bolts with different sizes and types. A white background is considered for all images in the database, because it simulates a controlled environment for an industrial process, which in turn reduces the complexity of the classification process.

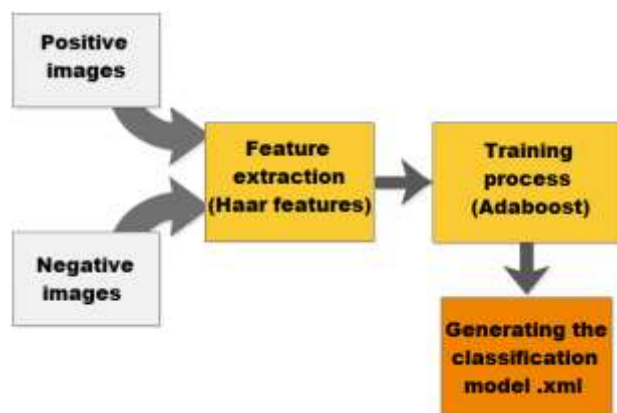


Figure 3 Database sample with negative images

Classifier training

The Haar Cascade classifier falls within a type of machine learning algorithms known as AdaBoost (adaptive enhancement). They use weak classifiers connected in succession to form a more efficient algorithm known as a strong classifier. Haar Cascade applies this principle, using as weak classifiers a type of binary windows known as Haar characteristics, which have different configurations that, when superimposed on a certain section of an image, serve to detect lines and contours of different attributes.

There are a large number of Haar characteristics, exceeding hundreds of thousands, which makes their direct use in the extraction of characteristics computationally expensive. As a consequence, the Adaboost algorithm is applied, selecting the most relevant Haar characteristics found in the positive images from the training database, and discarding the vast majority which are irrelevant. These characteristics will be used in the generation of weak classifiers, as part of a predictive model for classification using Haar Cascade.



**Figure 4** Block diagram for the training process

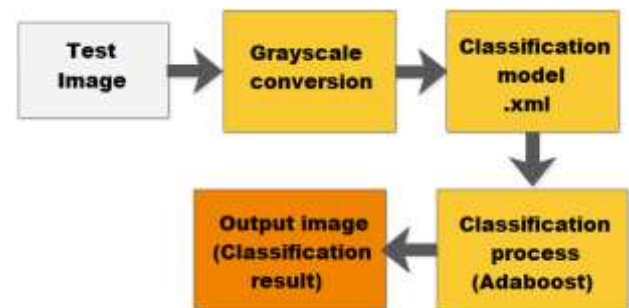
It should be mentioned that the training carried out offline and can generally last from tens of minutes or even days, depending on the complexity of the object to be detected and the number of sample images in the database.

### Real-time detection test

The next stage consists of the generation of a test algorithm, which receives input images with similar environmental and lighting conditions to those of the training database, although at the same time completely new. These images must be transformed to grayscale for further processing using the classification model generated in the previous stage.

The next step is the implementation of the Adaboost algorithm, applying the weak classifiers obtained in the previous stage, for the analysis of different subregions in the test image. Analyzing at such locations can give both positive and negative results. In the case of having a positive result, the region is assigned a higher weight and in the opposite case it is assigned a lower weight.

The results in the subregions with negative label being almost immediately discarded and the analysis is focused on the regions where higher weights are obtained or with positive results, implementing in them the cascade analysis of different weak classifiers. If the sub-region stands up to the scrutiny of all classifiers showing positive results, then the final result of the classification of that sub-region is positive.



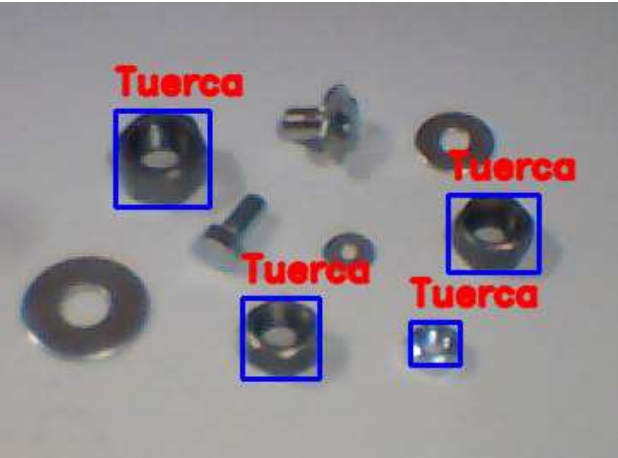
**Figure 5** Block diagram for the detection process.

It should be considered that the system was designed to analyze in real time, a video signal acquired from a Web Camera. This is possible due to the computing power of current embedded systems, achieving analysis and classification in tens of images per second.

For the classification system test, a stage was created with the same lighting and background as that used in the training images, placing different distributions of nuts, bolts and washers with different dimensions. The distance between the objects to be classified and the Web camera is not completely fixed, allowing a certain degree of movement, as could happen in some industrial processes including analysis on a moving conveyor belt.

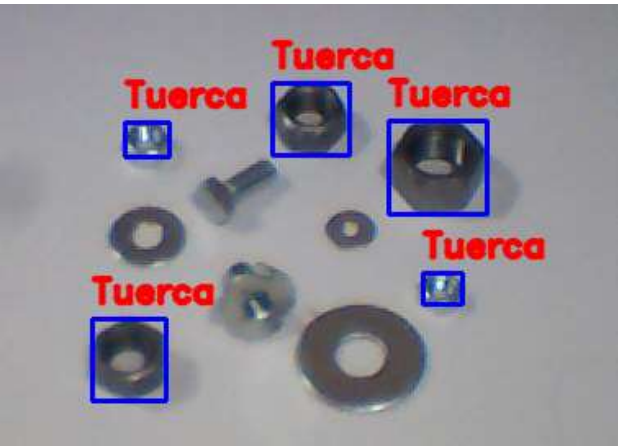
The system was trained to generalize and recognize the nut class, regardless of whether they vary in size, color, or texture. The algorithm was configured to enclose the objects detected as positive, in a blue box with the label corresponding to the nut class (tuerca). Objects that are not detected within that class are simply ignored.

Figure 6 shows a screenshot of the proposed machine vision system in operation. The image corresponds to a 1.3 Mega pixel video signal at 30fps, processed in real time. It is observed that the classification of the nuts was correct, without false positives or negatives.



**Figure 6** The proposed artificial vision system performing the classification of nuts (tuercas) in real-time using Har Cascade, Raspberry Pi and Web camera.

Changes were made in the distribution, number and type of pieces, obtaining satisfactory results in the classification. Figure 7 shows a classification case in which a change was made in the distribution of the pieces.



**Figure 7** Change in the parts distribution with a correct classification.

**Algorithm performance**

A database with 100 images was implemented for testing, in which scenarios similar to those shown in figures 6 and 7 can be observed, although with the difference that each image has its ground truth. The objective is to use this database to test the performance of the vision system, implementing the confusion matrix with the classification percentages in the parameters: true positive TP, true negative TN, false positive FP and false negative FN. This is possible because the Haar Cascade is a supervised classification algorithm.

The results obtained in the calculation of the confusion matrix are shown below.

|                   | Predicted Positive | Predicted Negative |
|-------------------|--------------------|--------------------|
| Actually Positive | TP =0.85           | FP =0.20           |
| Actually Negative | FN =0.15           | TN =0.80           |

**Table 1** Confusion Matrix

To calculate the accuracy, the following formula is applied:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Accuracy = 0.825$$
$$Error\ rate = 0.175$$

**Conclusions**

According to the objectives established at the beginning of this work, an artificial vision system for classifying objects in real time using Raspberry Pi and Web camera was successfully developed.

The advantages to consider are described below:

- Low cost
- Good performance
- Implementation with free software
- Real-time operation
- Embedded system
- Conventional webcam
- Portable

The system was implemented with free software in specific Python and OpenCV, with the advantages that this entails in portability, costs and debugging capacity. In addition to this, the use of the Raspberry Pi 3B + card and a conventional webcam as hardware, provide the system with an affordable cost (no more than 100 dollars) although with good computing power, achieving fluid and real-time operation. In addition, it has the advantage of being portable and small in size, becoming a serious competitor to expensive industrial cameras for computer vision applications.



## References

Priadana, A and Habibi, M. (2019) Face detection using haar cascades to filter selfie face image on instagram", 2019 International Conference of Artificial Intelligence and Information Technology ICAIIT, pp. 6-9.

Setjo, C H, Achmad B and Faridah. (2017) Thermal image human detection using Haar-cascade classifier", Proc. – 2017 7th Int. Annu. Eng. Semin. Ina. 2017, no. January 2020, 2017.

Tian X, Feng H and J Chen. (2020) An Industrial Production Line Dynamic Target Tracking System Based on HAAR and CAMSHIFT, International Journal of Pattern Recognition and Artificial Intelligence Vol. 34, No. 11, 2059037.

Viola, P. and Jones, M. (2001) Rapid object detection using a boosted cascade of simple features. In Proceedings, IEEE Conference on Computer Vision and Pattern Recognition.

Viola, P and Jones, M (2004) Robust real-time face detection. International Journal of Computer Visio, vol. 57, no. 2, pp. 137-154.