JME
Journal of Mechatronics Engineering

Embedded Linux System for Digital Image Recognition using Internet of Things

Davi S. Freire[1], Maiara J. R. Silva[1], Sandro C.S. Jucá[1], Renata I. S. Pereira[2]

[1]Federal Institute of Education, Science and Technology of Ceará (IFCE)
[2]Federal University of Ceará (UFC)

<davihd160@gmail.com>, <maiarajessicars@gmail.com>,
<sandrojuca@ifce.edu.br>, <renata@dee.ufc.br>

DOI: 10.21439/jme.v1i2.14

Received: 28 Feb. 2018. **Accepted:** 15 May 2018

Abstract. The present paper describes the use of Digital Image Processing and Internet of Things for gestures recognition using depth sensors available on Kinect® device. Using the open source libraries OpenCV and libfreenect, image data are translated and used for communicating Raspberry Pi embedded Linux system with PIC microcontroller board for peripheral devices controlling. LED is triggered according to the hand gesture representing the corresponding number. Data are stored in a PHP Apache server running locally on Raspberry Pi. The proposed system can be used as a multifunctional tool in areas such as learning process, post-traumatic rehabilitation and visual and motor cognition time. Using image binarization and Naive-Bayes classifier, the achieved results show error lower than 5%.

Keywords: Digital Image Processing, Embedded Systems, Gesture Recognition, Internet of Things, Raspberry Pi.

1 Introduction

This study consists of the elaboration of a tool that implements the recognition of hand gestures identified by a Linux embedded system for the control of loads, such as the activation of LEDs. For this, a protocol for identifying visual commands will also be developed. Therefore, the study of concepts of Digital Image Processing (DIP), embedded systems and Internet of Things (IoT) will be carried out, with the objective of creating an auxiliary tool in the most diverse applications .

Recognition of hand gestures is important for human-machine interaction, since there are numerous applications in virtual reality, sign language recognition and computer games (FREIRE; SILVA; JUCÁ, 2017). Despite many previous work, such methods of recognizing traditional hand gestures that are based on vision are still far from satisfactory for real-life applications. Due to the limitations of optical sensors, the quality of the captured images is sensitive to lighting and background conditions, so it is not able to detect and identify the hands in a robust way, which greatly affects the performance of gesture recognition.

With the development of low-cost depth cameras, such as Kinect®, new opportunities for gesture recognition arise. This work proposes to recognize hand gestures using Kinect®, verifying how many fingers are being shown in the captured scenes, making it possible to perform quantification, identification of forms and later assign the execution of actions by recognizing each established pattern. Thus, it is possible to generate indicators, for example, on the process of learning numbers for children who are at the appropriate level for this type of knowledge, as well as for children or adults who are relearning after traumatic processes of total or partial loss memory or other cognitive faculties. The generation of these indicators, under the coordination of a professional of the area, can be obtained with unsupervised tests to the end users, in order to assist them in the evaluation of the level of visual, auditory, motor or reaction time cognition, among other information that can be made available and that are important to specialists. They can be used in other applications in several different areas.

For the development of this research, the C/C++ programming language was used because it is multiplatform and free for development along with OpenCV computer vision libraries, which are also multiplatforms and open source. Imaging and video processing modules have been used, the libfreenect library is free and has drivers that allow the use of the Microsoft Kinect device in digital image processing in both 2D and 3D. For communication between the embedded system and the developed tool the Raspberry Pi will be used.

The sequence of this work is organized as follows: the state of the art is presented in section 2, followed by the technologies used in section 3, methodology in section 4, the results obtained in section 5 and finally section 6 with the conclusion.

2 Bibliographic review

This section describes concepts related to the processing of images for the recognition of gestures allied to the Internet of Things.

2.1 Internet of Things (IoT)

With the advancement of technology, the Internet shows us the possibility of connecting day-to-day devices to the Internet, thus giving birth to the Internet concept of Things, which represents another technological revolution, with the premise of being the future of computing and communication.

Systems with greater capacity for integration and extension are the most common requests that manufacturers have received and new intelligent products are born at all times, although these products often do not communicate with those of other manufacturers (IBM, 2014), being This question worked in several ways in Andersen, Fierro e Culler (2016), Desai, Sheth e Anantharam (2015), Pereira et al. (2016). With this motivation, Internet of Things is increasingly present in the development of systems that require greater interoperability, the market has been dealing with increasingly heterogeneous and interconnected environments for some time.

According to Harbor (2014), the Internet of Things transforms the world into a digital nervous system based on sensors and actuators. The Figure 1 shows that IoT can be used in many different areas, such as the use of GPS for vehicle tracking (LEE; TEWOLDE; KWON, 2014), video-monitoring (CHEN; CHIEN, 2012), embedded systems (PEREIRA; JUCÁ; CARVALHO, 2016) home automation (DEY; ROY; DAS, 2016; KODALI et al., 2016a; KODALI et al., 2016b), sensors with the ability to measure temperature (MADDIKATLA; JANDHYALA, 2016) or precision agriculture (CAMBRA et al., 2017).

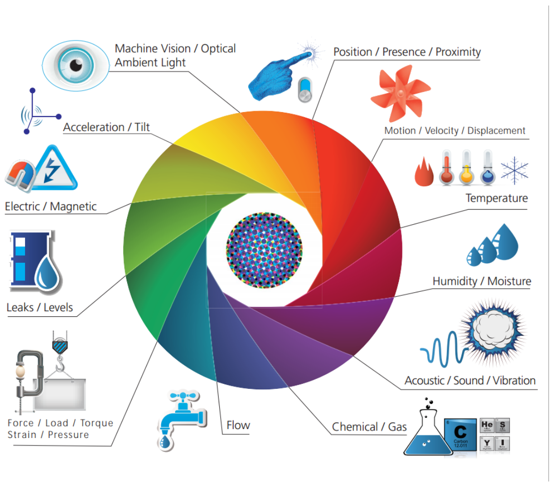

Figure 1: Map of sensors and their uses (HARBOR, 2014).

2.2 Digital Image Processing (DIP)

Digital Image Processing (DIP) is the manipulation of one image per computer so that the input and/or output are images. According to Camara et al. (1996), DIP can improve the visual aspect of certain structures for analysis and provide means for their interpretation, and may even generate products that can be submitted to other processing. Thus, it facilitates the extraction of information from images. Some concepts for extracting this information will be discussed below.

2.2.1 Image Segmentation

A segmentation process that correctly identifies the location, topology, and shape of objects is a critical requirement for information resulting from an image analysis system to be trusted (PEDRINI; SCHWARTZ, 2007). This step is of great importance, having a wide area of study about it, having its use in the most diverse situations, such as medical image analysis (AYDIN et al., 2017; RAZZAK; NAZ, 2017; HO et al., 2017; AJAM et al., 2017; TAN; KUMAR, 2012; LAHIRI et al., 2017) for the diagnosis of pathologies, in the field of computational vision and pattern classification (JÉGOU et al., 2017; DU; GAO, 2017), among many others.

2.2.2 Thresholding

Image segmentation technique consisting of pixel classification of an image according to the specification of one or more thresholds, that is, the threshold will be a value associated with a function that will allow the classification of the pixels in sets. When thresholding occurs by the use of only one threshold, it is called a binary threshold. That said, a simple way to extract objects from a scene is by means of a threshold T that separates the pixels into two sets: points of objects and background points. Thus, each coordinate pixel (x, y) such that $f(x, y) > T$ is called an object point, otherwise it is a background point, as seen in equation 1.

$$G(v) = \begin{cases} 0, & v \leq T \\ 1, & v > T \end{cases} \tag{1}$$

Pixels with a value of 1 correspond to objects, while pixels with a value of 0 correspond to the background. Since this threshold has only two levels of intensity values, 0 (black) and 1 (white) it is called binarization. As can be seen in Figure 2, the choice of a good threshold T is of paramount importance for thresholding to occur from the best way possible within a context, so various types of techniques can be used to choose this threshold, such as global thresholding (XIE et al., 2010; NOSRATI; ANDREWS; HAMARNEH, 2013), entropy-based thresholding (NAND; NOOPUR; NEOGI, 2014), adaptive thresholding (ROUSSON; DERICHE, 2002; MARZOUKI; DELIGNON; PIECZYNSKI, 1994), thresholding based on histogram processing (DU; BURR, 2012; QIANG et al., 2013).

2.2.3 Classification of Standards

A default is an array of descriptors (GONZALEZ; WOODS, 2011). The classification of standards aims at a mapping that relates the properties of the samples with labels, in which samples with similar characteristics must have the same label. This set of distinct samples to which the same label is assigned are called class and thus each receives one of the labels $\omega_1, \omega_2, \omega_3, ..., \omega_m$, which m denotes the number of classes of interest in an experiment.

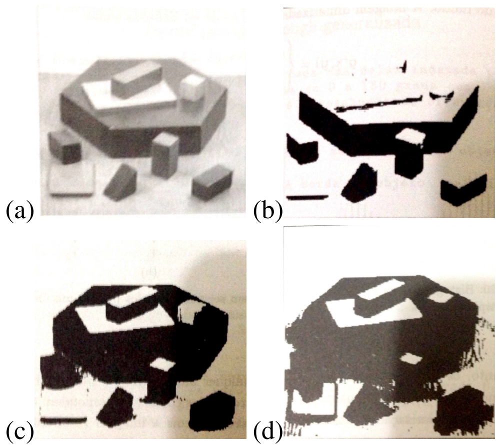

Figure 2: Segmentation by means of different threshold values T. (a)original image; (b)T=108; (c)T=179; (d)T=213 (PEDRINI; SCHWARTZ, 2007).

2.2.4 Bayes' Theorem

The Bayes' theorem (BAYES, 1763), also known as Bayesian reasoning or Bayesian inference, describes the probability of an event, based on a *priori* knowledge that may be related to the event.

$$P(A|\mathbf{B}) = P(A)\frac{P(\mathbf{B}|A)}{P(\mathbf{B})} \tag{2}$$

2.2.5 Naive Bayes

It is a family of classifiers based on the Bayes theorem (2) with an assumption of independence between the predictors, having its use in several types of classifications, such as face recognition (PUTRANTO; SITUMORANG; GIRSANG, 2016; CHEN et al., 2016) and objects(ZHANG; LIANG, 2016), reading emotions in songs (AN; SUN; WANG, 2017), texts(OGUL; OZCAN; HAKDAGLı, 2017). In simple terms, Naive Bayes are probabilistic classifiers in which it is assumed that the presence of a particular characteristic in a class is not related to the presence of any other resource (VIDHYA, 2015). For example, a fruit can be considered as an apple if it is red, round, and is about 3 inches in diameter. Even though these features depend on each other or on the existence of other characteristics, all these properties contribute independently to the probability that this fruit is an apple and that is why it is known as Naive.

3 Matherials and Methods

This topic describes the technologies used to perform this project.

3.1 SanUSB Tool (PIC18F Microcontroller)

The SanUSB development system, see Figure 3, is a tool composed of basic software and hardware of the PIC18Fxx5x family with USB interface. According to SANUSB (2009), this free tool demonstrates to be a useful tool in the development of real projects (JUCÁ; CARVALHO; BRITO, 2009; PEREIRA; JUCÁ, 2013; JUCÁ; PEREIRA; VASCONCELOS, 2012), for the ease firmware updating, that does not need its removal for such task, besides to have a multiplatform USB microcontroller recording software available, running on the main systems of today: Windows®, MacOS X® and Linux. It works in plug-and-play, not needing to install a drive to work.

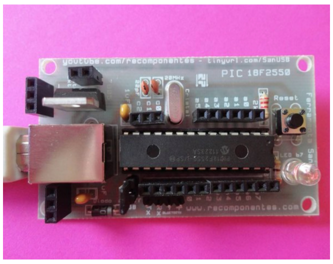

Figure 3: SanUSB PIC18F2550 Card. (SANUSB, 2006).

3.2 Microsoft Kinect®

Kinect ® is a motion sensor developed for the Xbox 360 ® and Xbox One ®, along with the company Prime Sense ©. In Simha et al. (2013), El-laithy, Huang e Yeh (2012) the ability of its use for image processing is demonstrated. The Kinect® module, see Figure 4, is about 23cm long and has 4 main features:

- RGB Camera (Red, Green, Blue) with 640x480 pixels resolution.
- Depth Sensor (InfraRed) with 320x240 pixel resolution, which allows the accessory to scan the environment around you in three dimensions.
- Built-in microphone, which in addition to capturing voices closer, can differentiate external noises.
- Own processor and software.

Figure 4: Device Microsoft Kinect® utilized on the project.

3.3 Libfreenect

Powered by OpenKinect, which is a community of people interested in using the Xbox Kinect ® device on PCs and other devices. It is a library that has drivers that allow you to use the device for development.

3.4 Raspberry Pi

The Raspberry Pi platform, considered the launch year in 2012, is one of the smallest computers in the world, the size of a credit card, USB connections to connect the keyboard and mouse used on desktop computers. You can connect it to TVs with HDMI output, as well as having a low hardware cost and zero cost in embedded system software. As can be seen in Figure 5, all hardware is integrated into a single card. They have the ability to develop everything a conventional computer does, such as browsing, creating spreadsheets, playing videos, processing texts, games, among other more complex tasks, such as online monitoring. That way it is used by people all over the world to learn how computers work, how to manipulate the electronic world around them, and how to program (SANUSB, 2006). An overview with prototyping of solutions with Raspberry Pi can be seen in Saari, Baharudin e Hyrynsalmi (2017), demonstrating an affinity in its use with the Internet of Things (PARDESHI et al., 2017; RUPANI et al., 2017) and in image processing (RUPANI et al., 2017).

3.5 OpenCV

OpenCV is a free machine-readable computer-readable code library. Provides computational efficiency focused on real-time applications (BRADSKI; KAEHLER, 2008). It has more than 2500 optimized algorithms that

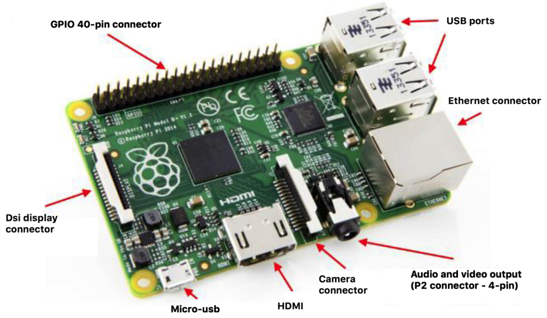

Figure 5: Raspberry Pi (SANUSB, 2006).

can be used in several different applications related to DIP and pattern recognition.

4 Methodology

The construction of this project is divided into two parts: In its initial stage, the recognition of gestures will occur through the processing of images, in which the data will be processed and sent by Raspberry Pi to the Internet for the storage of the information. The steps are described in the following subtopics.

4.1 Gesture processing

4.1.1 Gesture Recognition Method

The captured images were processed in the Linux embedded system using the Raspberry Pi platform. Following is the steps for gesture recognition, according to the flowchart shown in Figure 6:

I Kinect® Image Acquisition: The depth sensor of Kinect ® assigns values according to the distance of the objects in scene, these values will be used to determine the existence or not of objects in the scene.

II Separation of objects in the image by window: To capture only objects that can fit the definition of hands of the algorithm, control was used on the scene view, where the viewing window is in the range of 500 to 700 pixels of depth captured by the sensor of Kinect ®. Taking into account that a distance in pixels when being converted to a distance in centimeters can be altered by factors such as: the angle at which the sensor detects objects as well as the very distance of the sensor that the objects of the scene meet.

III Isolation of the pattern through binarization: As can be seen in Figure 7, this step consists of binarizing the image captured by the sensor inside the viewing window. If there is an object in the window in a certain pixel of the image it is assigned a white color (value 255), if there is no, it is assigned a black color (value 0).

IV Delimitation of the evaluation space: After binarization, the resulting image is delimited to an area of 200x250 pixels in the center of the image, with the sole aim of reducing the cost of processing in the analysis.

V Identification of the gesture: With the information resulting from the binarization it is possible to verify the area occupied by the user's hand when performing each gesture. The algorithm was tested with several users, in the learning period using the Bayes Theorem it was possible to define classifiers for the pattern of the area occupied by each gesture that are represented in the table 1.

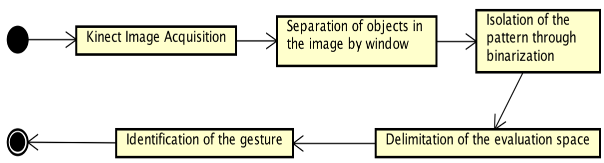

Figure 6: Flowchart with steps for the recognition of gestures.

Figure 7: Image of the scene and the binarization of the analyzed area.

The algorithm was submitted to tests with several users, obtaining with the analysis of the results a stan-

Table 1: Minimum and maximum values of pixels reached by the hand area for each gesture.

	Open Hand	Hand Showing Four Fingers	Hand Showing three Fingers	Hand Showing two Fingers	Hand Showing One Fingers	Closed Hand
Minimum value	5500	4800	4000	3500	3000	2000
Maximum value	6000	5499	4799	3999	3499	2999

dard for the area occupied by each gesture, with a margin of error of 200 pixels. The table 1 displays the areas occupied by each gesture.

Upon acquiring this information, the first part of the project is completed, in which the information is transferred via URL by the GET method to the PHP server allocated in Raspberry Pi, and the second part of the project is started.

4.2 Sending Raspberry Pi data to PIC

This topic will address the process of sending data from the computer to a PHP page hosted on an Apache server allocated in Raspberry Pi and it passes this data to the PIC. The process starts with the configuration of the server in Raspberry Pi and the installation of the PHP module, which will make the server able to receive the data.

In order for the communication between the Raspberry and the PIC to work smoothly, a serial connection was made between the TX pin (data transmission) of the Raspberry Pi and the RX pin (data reception) of the microcontroller. As can be seen in Figure 8, the computer will send the command to which the gesture was checked for the server, which stores that variable. Afterwards, the Linux embedded system reads the stored data and sends it to the PIC, which will use this command to perform load control, decide which LED will be triggered, following the pattern in which the LED identifier is equal to the number of the recognized gesture.

5 Results and discussion

The development and testing environment of this work was performed on a Linux operating system, Debian Wheezy distribution. The algorithms in the research were written in the C/C++ languages, using the libfreenect and OpenCV libraries, in order to manipulate, process and process the images. Several users performed hand gestures in front of Kinect® and the generated images were used by the algorithms of this research in order to recognize and obtain important information

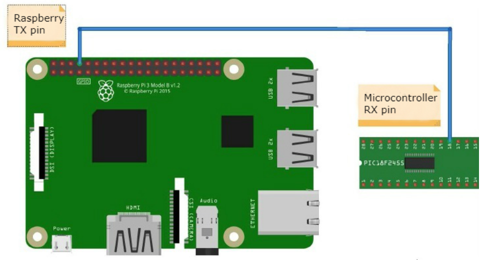

Figure 8: Schematic of the connection between the Linux embedded system and the microcontroller.

about the hands to perform the tests. The prototype hit rate was measured based on tests in which a user made a gesture showing a number of fingers in front of the depth sensor, see Figure 9 (a) and (b), hoping that the system to return the value corresponding to the acknowledged gesture and transfer to the microcontroller, which in turn will activate the LED corresponding to the gesture, according to Figure 9(c). Are presented on Table 2 and 3, respectively the rate of success in the recognition of each gesture by the system and the comparison between the methodology implemented with other views in the literature. Use of the application should be used on computers connected with Kinect®.

Table 2: Percentage accuracy of each gesture by the proposed system.

	Open Hand	Hand Showing Four Fingers	Hand Showing three Fingers	Hand Showing two Fingers	Hand Showing One Finger	Closed Hand
Minimum value	5500	4800	4000	3500	3000	2000
Maximum value	6000	5499	4799	3999	3499	2999

Table 3: Comparative between the methodology implemented and other views in the literature.

Methodology	Hit rate (%)
LECUN; HUANG; BOTTOU, 2004	87.00
MIAN;BENNAMOUN; OWENS, 2006	95.00
B.LEIBE; A.LEONARDIS; SCHIELE, 2006	91.00-97.00
SHOTTON et al., 2006	72.20
HARTL; ARTH; SCHMALSTIEG, 2010	89.00
YI et al., 2014	94.50
Proposal	95.50

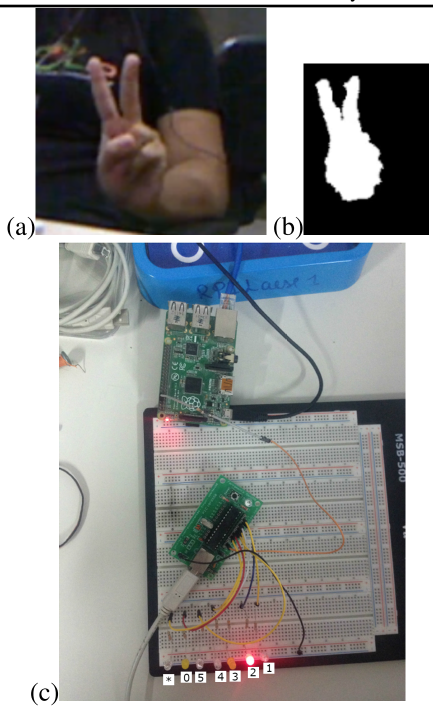

Figure 9: Result obtained through the recognition of the gesture. (a) Image captured by Kinect®; (b) Binarization for gesture recognition; (c) Activation of led number 2, according to the recognition of the gesture of showing two fingers in front of Kinect®.

6 Conclusion

The use of the librefreenect library for use of the Kinect® depth sensor has proved to be a viable alternative for in-depth image capture.

From that moment on, OpenCV libraries assist in the processing and analysis of images for the recognition of hand gestures, taking advantage of their real-time use and the large number of functions implemented in computer vision and machine learning, aiding in the creation of a window that will decrease the area to be analyzed of the image, reducing the processing of the embedded Linux system, since the area outside this window will not be analyzed. Finally the platform Raspberry Pi, which presents as advantages its low cost and total dedication to the device to which it controls, accomplishing the data storage and sending them to the microcontroller, for the control of loads, that in this project is the activation of LEDs of according to the recognized gesture, but which can be used in other situations, such as the activation of electronic devices.

The main advantage of the system is the high level of accuracy. Compared to other proposals for the recognition of objects in images presents a value greater than the average of the others. As far as we know, this is one of the first works on using the Kinect® depth sensor for gesture recognition combined with load control by an embedded system.

The described system works as expected, with an error rate below 5%, and has the potential to become a multifunctional tool to aid in the recognition of hand gestures combined with load control.

For future work, it is intended to create the option of registering new gestures by the user, for this, will be carried out the study of artificial intelligence technologies in order to enable the system to learn to synthesize area patterns for new gestures, in addition to using a database of images of hand gestures with greater comprehensiveness to carry out the quantification of the tests automatically and to work in the evolution of the system, testing other techniques and conducting more experiments.

It is important to emphasize that the concepts of DIP and IoT are not new, but that in recent years there is a greater investment of companies in projects that use these concepts, as well as their use in teaching environments.

References

AJAM, A.; AZIZ, A. A.; ASIRVADAM, V. S.; MUDA, A. S.; FAYE, I.; GARDEZI, S. J. S. A review on segmentation and modeling of cerebral vasculature for surgical planning. *IEEE Access*, v. 5, p. 15222–15240, 2017.

AN, Y.; SUN, S.; WANG, S. Naive bayes classifiers for music emotion classification based on lyrics. In: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. [S.l.: s.n.], 2017. p. 635–638.

ANDERSEN, M. P.; FIERRO, G.; CULLER, D. E. Enabling synergy in iot: Platform to service and beyond. p. 1–12, April 2016.

AYDIN, A. S.; DUBEY, A.; DOVRAT, D.; AHARONI, A.; SHILKROT, R. Cnn based yeast cell segmentation in multi-modal fluorescent microscopy data. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 753–759.

BAYES, T. An essay towards solving a problem in the doctrine of chances. In: *Philosophical Transactions of the Royal Society of London*. [S.l.: s.n.], 1763. p. 370–418.

BRADSKI, D. G. R.; KAEHLER, A. *Learning Opencv, 1st Edition*. First. [S.l.]: O'Reilly Media, Inc., 2008. ISBN 9780596516130.

CAMARA, G.; SOUZA, R.; FREITAS, U.; GARRIDO, J. *Integrating remote sensing and GIS by object-oriented data modelling.* [S.l.]: Computers & Graphics, Volume 20., 1996. 395–403 p.

CAMBRA, C.; SENDRA, S.; LLORET, J.; GARCIA, L. An iot service-oriented system for agriculture monitoring. In: *2017 IEEE International Conference on Communications (ICC)*. [S.l.: s.n.], 2017. p. 1–6.

CHEN, Y.-K.; CHIEN, S.-Y. Perpetual video camera for internet-of-things. In: *2012 Visual Communications and Image Processing*. [S.l.: s.n.], 2012. p. 1–7.

CHEN, Y. P.; LIU, C. H.; CHOU, K. Y.; WANG, S. Y. Real-time and low-memory multi-face detection system design based on naive bayes classifier using fpga. In: *2016 International Automatic Control Conference (CACS)*. [S.l.: s.n.], 2016. p. 7–12.

DESAI, P.; SHETH, A.; ANANTHARAM, P. Semantic gateway as a service architecture for iot interoperability. In: *2015 IEEE International Conference on Mobile Services*. [S.l.: s.n.], 2015. p. 313–319. ISSN 2329-6429.

DEY, S.; ROY, A.; DAS, S. Home automation using internet of thing. In: *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*. [S.l.: s.n.], 2016. p. 1–6.

DU, C.; GAO, S. Image segmentation-based multi-focus image fusion through multi-scale convolutional neural network. *IEEE Access*, v. 5, p. 15750–15761, 2017.

DU, H.; BURR, K. An algorithm for automatic flood histogram segmentation for a pet detector. In: *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*. [S.l.: s.n.], 2012. p. 3488–3492. ISSN 1082-3654.

EL-LAITHY, R. A.; HUANG, J.; YEH, M. Study on the use of microsoft kinect for robotics applications. In: *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*. [S.l.: s.n.], 2012. p. 1280–1288. ISSN 2153-358X.

FREIRE, D. S.; SILVA, M. J. R.; JUCÁ, S. C. S. Sistema embarcado linux para reconhecimento de padrões da mão utilizando conceitos de pdi e iot. In: *III Escola Regional de Informática do Piauí*. [S.l.: s.n.], 2017. v. 1, p. 200–205.

GONZALEZ, R. C.; WOODS, R. C. *Processamento Digital de Imagens.* [S.l.]: Pearson, 2011. 569 p.

HARBOR, R. *Harbor Research's Infographic on the Internet of Things and Smart Services*. 2014. Disponível em: <https://goo.gl/Wf2K9Z>. Acesso em: 18 may 2016.

HO, D. J.; FU, C.; SALAMA, P.; DUNN, K. W.; DELP, E. J. Nuclei segmentation of fluorescence microscopy images using three dimensional convolutional neural networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 834–842.

IBM. *A Interoperabilidade da Internet das Coisas*. 2014. Disponível em: <https://www.ibm.com/developerworks/community/\blogs/tlcbr/entry/mp207?lang=en>. Acesso em: 12 jul. 2017.

JÉGOU, S.; DROZDZAL, M.; VAZQUEZ, D.; ROMERO, A.; BENGIO, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 1175–1183.

JUCÁ, S.; CARVALHO, P.; BRITO, F. Sanusb: software educacional para o ensino da tecnologia de microcontroladores. In: *Ciências Cognição (UFRJ)*. [S.l.: s.n.], 2009. v. 14, p. 134–144.

JUCÁ, S.; PEREIRA, R.; VASCONCELOS, M. Sanusbee: Ferramenta para gravaÇÃo wireless de microcontroladores via bluetooth e zigbee. In: *Connepi - Congresso Norte Nordeste de Pesquisa e Inovação, Palmas-TO*. [S.l.: s.n.], 2012.

KODALI, R. K.; JAIN, V.; BOSE, S.; BOPPANA, L. Iot based smart security and home automation system. In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*. [S.l.: s.n.], 2016. p. 1286–1289.

____. Iot based smart security and home automation system. In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*. [S.l.: s.n.], 2016. p. 1286–1289.

LAHIRI, A.; AYUSH, K.; BISWAS, P. K.; MITRA, P. Generative adversarial learning for reducing manual annotation in semantic segmentation on large scale miscroscopy images: Automated vessel segmentation in retinal fundus image as test case. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 794–800.

LECUN, Y.; HUANG, F. J.; BOTTOU, L. Learning methods for generic object recognition with invariance to pose and lighting. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. [S.l.: s.n.], 2004. v. 2, p. II–97–104 Vol.2. ISSN 1063-6919.

LEE, S.; TEWOLDE, G.; KWON, J. Design and implementation of vehicle tracking system using gps/gsm/gprs technology and smartphone application. In: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. [S.l.: s.n.], 2014. p. 353–358.

LEIBE, B.; LEONARDIS, A.; SCHIELE, B. Combined object categorization and segmentation with an implicit shape model. In: *Toward Category-Level Object Recognition, Part 4*. [S.l.: s.n.], 2006. p. 508–524.

MADDIKATLA, S. K.; JANDHYALA, S. An accurate all cmos temperature sensor for iot applications. In: *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. [S.l.: s.n.], 2016. p. 349–354.

MARZOUKI, A.; DELIGNON, Y.; PIECZYNSKI, W. Adaptative segmentation of sar images. In: *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings*. [S.l.: s.n.], 1994. v. 2, p. II/449–II/454 vol.2.

MIAN, A. S.; BENNAMOUN, M.; OWENS, R. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 28, n. 10, p. 1584–1601, Oct 2006. ISSN 0162-8828.

NAND, G. K.; NOOPUR; NEOGI, N. Defect detection of steel surface using entropy segmentation. In: *2014 Annual IEEE India Conference (INDICON)*. [S.l.: s.n.], 2014. p. 1–6. ISSN 2325-940X.

NOSRATI, M. S.; ANDREWS, S.; HAMARNEH, G. Bounded labeling function for global segmentation of multi-part objects with geometric constraints. In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 2032–2039. ISSN 1550-5499.

OGUL, I. U.; OZCAN, C.; HAKDAGLı, O. Fast text classification with naive bayes method on apache spark. In: *2017 25th Signal Processing and Communications Applications Conference (SIU)*. [S.l.: s.n.], 2017. p. 1–4.

PARDESHI, V.; SAGAR, S.; MURMURWAR, S.; HAGE, P. Health monitoring systems using iot and raspberry pi x2014; a review. In: *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. [S.l.: s.n.], 2017. p. 134–137.

PEDRINI, H.; SCHWARTZ, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. [S.l.]: Editora Thomson Learning, 2007. 528 p. ISBN 978-85-221-0595-3.

PEREIRA, C.; PINTO, A.; AGUIAR, A.; ROCHA, P.; SANTIAGO, F.; SOUSA, J. Iot interoperability for actuating applications through standardised m2m communications. In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. [S.l.: s.n.], 2016. p. 1–6.

PEREIRA, R.; JUCÁ, S. Sanusbot: Robô móvel educacional controlado por comando de voz via dispositivo android. In: *Connepi - Congresso Norte Nordeste de Pesquisa e Inovação*. [S.l.: s.n.], 2013.

PEREIRA, R. I. S.; JUCÁ, S. C. S.; CARVALHO, P. C. M. de. Online monitoring system for electrical microgeneration via embedded wifi modem. *IEEE Latin America Transactions*, v. 14, n. 7, p. 3124–3129, July 2016. ISSN 1548-0992.

PUTRANTO, E. B.; SITUMORANG, P. A.; GIRSANG, A. S. Face recognition using eigenface with naive bayes. In: *2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*. [S.l.: s.n.], 2016. p. 1–4.

QIANG, W.; ZHANG, Q.; MIAO, W.; LI, G.; WANG, H.; FENG, S. A power-constrained contrast enhancement algorithm for amoled display using histogram segmentation. In: *2013 IEEE 10th International Conference on ASIC*. [S.l.: s.n.], 2013. p. 1–4. ISSN 2162-7541.

RAZZAK, M. I.; NAZ, S. Microscopic blood smear segmentation and classification using deep contour aware cnn and extreme machine learning. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. [S.l.: s.n.], 2017. p. 801–807.

ROUSSON, M.; DERICHE, R. A variational framework for active and adaptative segmentation of vector valued images. In: *Workshop on Motion and Video Computing, 2002. Proceedings*. [S.l.: s.n.], 2002. p. 56–61.

RUPANI, A.; WHIG, P.; SUJEDIYA, G.; VYAS, P. A robust technique for image processing based on interfacing of raspberry-pi and fpga using iot. In: *2017 International Conference on Computer, Communications and Electronics (Comptelix)*. [S.l.: s.n.], 2017. p. 350–353.

SAARI, M.; BAHARUDIN, A. M. bin; HYRYNSALMI, S. Survey of prototyping solutions utilizing raspberry pi. In: *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. [S.l.: s.n.], 2017. p. 991–994.

SANUSB. Aplicações práticas de raspberry pi com microcontroladores pic. 1 ed, Ceará. In: . [S.l.: s.n.], 2006. Grupo SanUSB.

____. Aplicações práticas de eletrônica e microcontroladores em sistemas computacionais. 1 ed, Ceará. In: . [S.l.: s.n.], 2009. Grupo SanUSB.

SHOTTON, J.; WINN, J.; ROTHER, C.; CRIMINISI, A. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ____. *Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. p. 1–15. ISBN 978-3-540-33833-8. Disponível em: <https://doi.org/10.1007/11744023_1>.

SIMHA, S. N.; KUMAR, M. M.; NAGARAJA, V.; PRABHAKAR, T. V.; JAMADAGNI, H. S. Infinite coffee cup. In: *2013 Texas Instruments India Educators' Conference*. [S.l.: s.n.], 2013. p. 84–90.

TAN, C. W.; KUMAR, A. Efficient iris segmentation using grow-cut algorithm for remotely acquired iris images. In: *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. [S.l.: s.n.], 2012. p. 99–104.

VIDHYA, A. *6 Easy Steps to Learn Naive Bayes Algorithm*. 2015. Disponível em: <https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/>. Acesso em: 22 may 2017.

XIE, Q. J.; JIN, W. B.; MA, L.; HOU, D. B. Fast global segmentation based on the dual formulation of tv-norm. In: *2010 3rd International Congress on Image and Signal Processing*. [S.l.: s.n.], 2010. v. 3, p. 1382–1385.

YI, Y.; GUANG, Y.; HAO, Z.; MENG-YIN, F.; MEI-LING, W. Object segmentation and recognition in 3d point cloud with language model. In: *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*. [S.l.: s.n.], 2014. p. 1–6.

ZHANG, Y.; LIANG, J. A vision based method for object recognition. In: *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*. [S.l.: s.n.], 2016. p. 139–142.