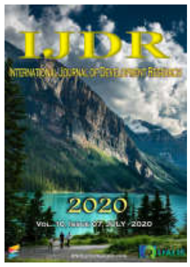
ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 10, Issue, 07, pp. 38650-38656, July, 2020
https://doi.org/10.37118/ijdr.19567.07.2020



RESEARCH ARTICLE OPEN ACCESS

TEXT RECOGNITION IN PCBS: AN OBJECT CHARACTER RECOGNITION (OCR) ALGORITHM

***Yuzo Iano, Daniel Katz Bonello and Umberto Bonello Neto**

Department of Communications, School of Electrical and Computer Engineering, University of Campinas

ARTICLE INFO

Article History:
Received 17th April, 2020
Received in revised form
14th May, 2020
Accepted 11th June, 2020
Published online 30th July, 2020

Key Words:
OCR Algorithm, AOI system,
PCB industry,
FMC assembling.

****Corresponding author: Yuzo Iano***

ABSTRACT

The Optical Character Recognition (OCR), together with another inspection techniques, is the mostly method utilized in the PCB industry. This work have as objective to identify text characters in five main PCBs template samples. Further, through a “Guide User Interface” (GUI) developed in MATLAB software, estimate the parameters as the quantity of detected texts, plot the graphic with the mean time average, and classifying the text character detected. Results showing the efficacy detection capacity of the OCR algorithm also are described in this work.

Citation: ***Yuzo Iano, Daniel Katz Bonello and Umberto Bonello Neto.*** “Text Recognition in PCBs: An Object Character Recognition (OCR) Algorithm*”, International Journal of Development Research*, 10, (07), 38650-38656.

INTRODUCTION

Printed circuit boards are the main support for all the electronic products utilized in the various industrial brenches [Anitha, 2019]. AOI systems in the present involving technology are responsible to the identification of various types of defects, they are the solution for the automated inspection, and however, its expansive hence industries with small production scale can not to invest in this kind of machinery [Moganti, 1996]. Taking into account those parameters, certainly an analysis in function of quantity of detected texts, mean time average for the PCBs samples reading, and classifying the text character detected in the bare PCBs, was proposed in this work. Hence, defects founded in a bare board, using the OCR technique for the detection and classification about the type of text founded, and, furthermore, a novel approach appoints the mean average of the time spent to detect those characteristics. Therefore, we can say that with this concept in mind, once that OCR technique for detection text characters in bare PCBs is completed, the results appoints satisfactory identification of texts capacity for this novel algorithm developed in MATLAB, within a detection precision average about 90% better than the previous algorithms proposed in the works [Stephan, 2015; We, 2018].

METHODOLOGY

Definition of Flexible Manufacturing Cell (FMC) for a low volume production PCBs assembly: Facing the currently technological scenario and the high development of solutions increasingly integrated to the conception and validation of mechatronics devices, the usage of toolings performed to the integration of flexible manufacturing cells (FMC) in a low production PCB assembly line in the industry become an indispensable option to streamline and enable the implementation of different architectures, thus allowing the reduction of project time finalization or scientific study, comply with the desired efficiency. The Fig. 1 shows the schematic of a FMC utilized in the PCB assembly industry:

Flexible Manufacturing Cells (FMCs), sometimes referred as Flexible Manufacturing Systems (FMS), are characterized by multiple machines (robots), with alleatory automatized movement of mechanical parts from and to the processing manufacturing stations, and the control made by a central computer with sofisticated command softwares. The Fig. 2 shows the most common FMCs configurations founed in the literature [We, 2018].

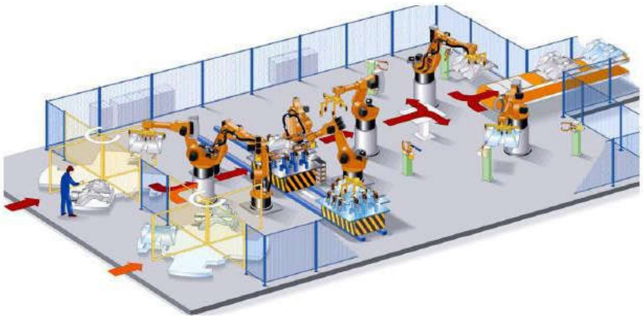


Fig.1. Schematic of a FMC utilized in the PCB industry.

Furthermore, FMCs can be operated with AOI robustness algorithms. These algorithms, such as OCR algoritms, are responsible to detect defects during the bare PCBs assembling process, making the production more reliable and free of reworks or process losts because inneficient inspection. Thus, FMCs is a technology which is in constant evolution, and new technicques has been developed to integrated the production inspection mesh in the PCBs industries, becoming an indeed and novel approach [We, 2018].

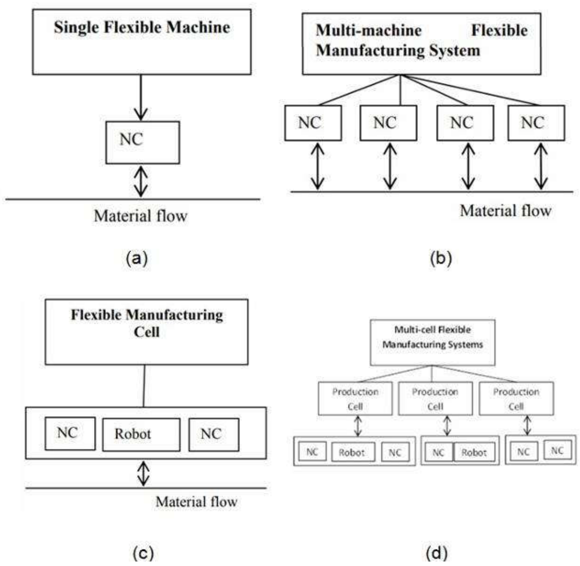


Fig. 2. Main FMCs characteristics

Now, the Fig. 3 shows the conventional process of PCBs manufacturing [Clyde, 2008]:

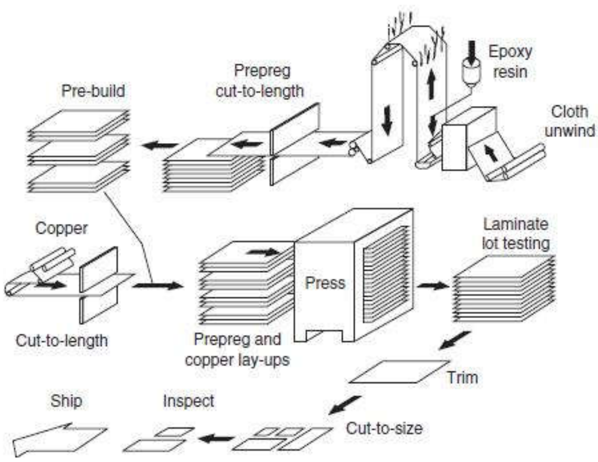


Fig. 3. The conventional manufacturing process

The major process flow of the low volume PCB assembly production is explained in the brief shown in the Fig. 4, according to the SMT methods [Will, 2020; Surface Mount Process, 2020]:

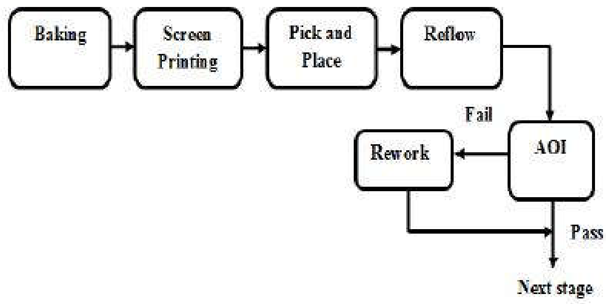


Fig.4. The PCB assembling process flow

Some relevant aspects to investing in OCR algorithms development for the PCBs industries: The most usual errors founded in the PCBs manufacturing assembly process in the industry can be represented by the Fig.5:

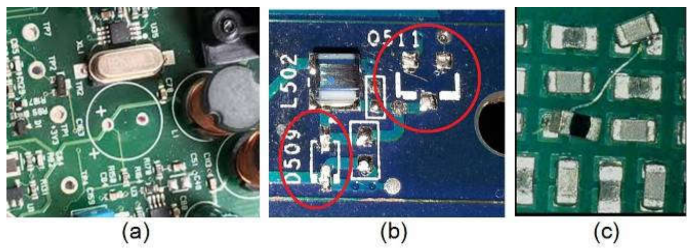


Fig. 5. Most usual errors founded in the PCBs manufacturing process: a – wrong polarity; b – missing components; c – misaligned components

Now its suitable a briefly description about these most usual errors founded in the PCBs manufacturing process [Surface Mount Process, 2020]:

- **Wrong polarity:** the assembling orientation of PCBs components is made through placement marks printed on the board (bare PCBs), hence in this process stage the pick and place operation is responsible to the electronic components correct polarity reading, placing them on the correct board position.
- **Missing components:** missing components also are the root cause of many rework and scraps inside the industry. Thus, with an OCR technique for the correct detection symbology in concern to the components placement on the board, will avoid these components will be missing during the productive process.
- **Misaligned components:** misaligned components also are the root cause of reworks during the PCBs assembling process, once the OCR techniques may detect whether electronic components addressing were printed in a correct pattern.

The Fig.6 shows which stage of SMT PCBs components assembly process would be profitable to utilize the OCR algorithm:

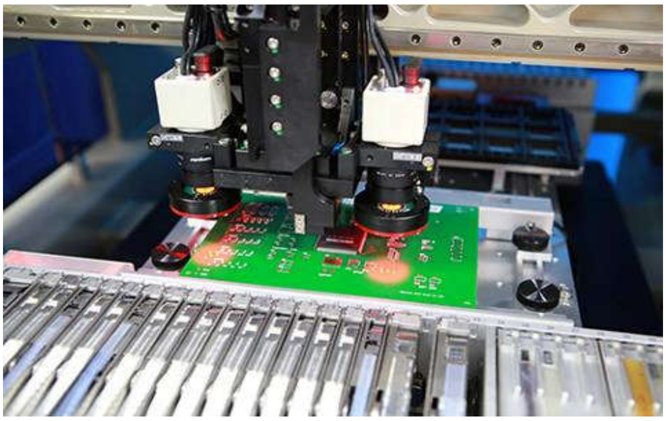


Fig. 6. Pick and place SMT machine

The automated robotic machines are responsible to the pick and place operation in a FMC, placing the SMT components in specific defined locations on the PCBs pads. Hence, the OCR algorithm proposed in this work would work as a position reference for these SMT components [Surface Mount Process, 2020].

Development of OCR Algorithm in MATLAB: The following algorithm shows the script of the code developed in MATLAB for bare pre-assembled PCBs text recognition operation also capable to quantify the image processing time (seconds) for each five samples loaded into the code. In this algorithm, at the left side, are loaded the five main samples of bare PCBs containing the texts elements to be detected by the code. The column results indicate the number of the sample with detected texts in the images. There is the magnified sample image, which texts were detected, in the bottom of the GUI window. Now, upside window, there is time execution graphic per image, which accounts the image execution time while algorithm is running. Finally, the table at the right down side of the window shows the detected texts, image and classification for each five samples tested in algorithm as shown in the Fig. 12.

Bench of Bare PCBs Samples with Character Texts: A bench of 5 images was composed to test the OCR algorithm of texts recognition developed in MATLAB, analyzing the text quantity into the PCBs. There are images related to various printed texts together with components located on the board, indicating values of components reference resistance, addressing of components, voltage and components polarity [Zenodo, 2020].

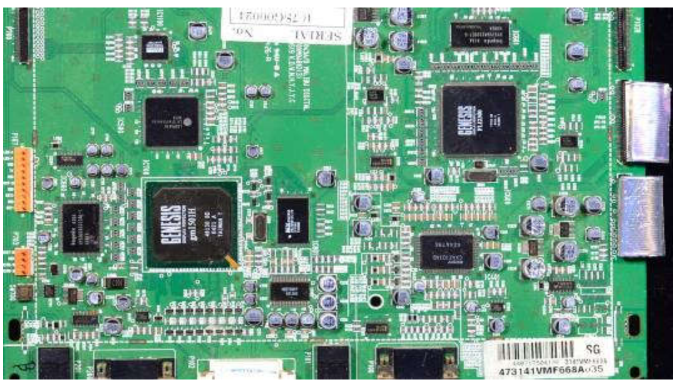


Fig. 7. Bare printed text PCB sample 1.

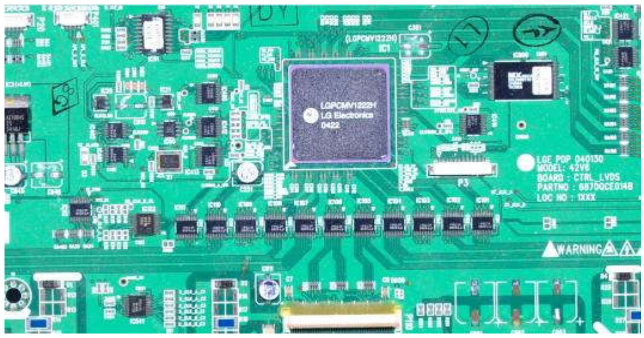


Fig.8. Bare printed text PCB sample 2.

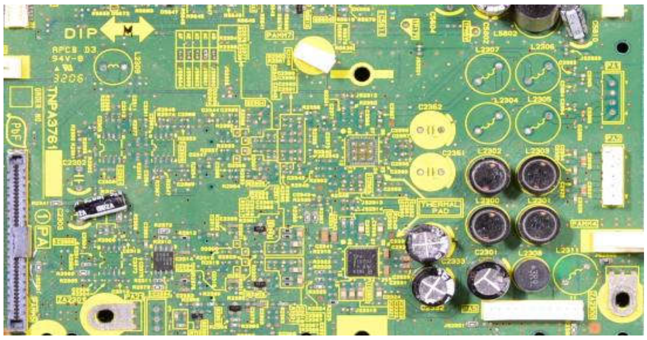


Fig. 9. Bare printed text PCB sample 3

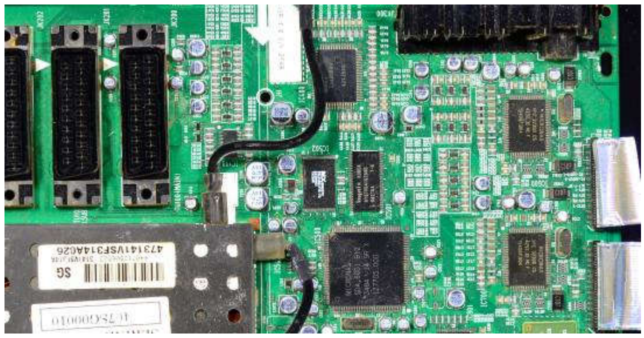


Fig. 10. Bare printed text PCB sample 4

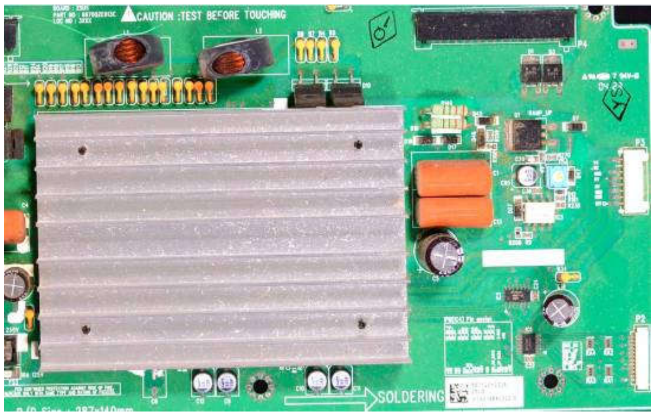


Fig.11. Bare printed text PCB sample 5.

Comparison with Previous Studies and Improved GUI Interface Developed in MATLAB to Detect Text Characters on PCBs Images: Compared to the previous works, related in the references of this article, we can see the new OCR algorithm proposed to detect text characters in pre-assembling bare PCBs samples has an optimized interface with loaded sample images at the left side of the algorithm screen, detect text results columns, magnified bare PCB sample image at the bottom, execution time (seconds) per image at the upper side of the screen, and a table at the right-down side indicating the type of detected text, where is in the image and the classification of the detected text [Stephan, 2015].

Fig. 12. GUI interface of the OCR algorithm

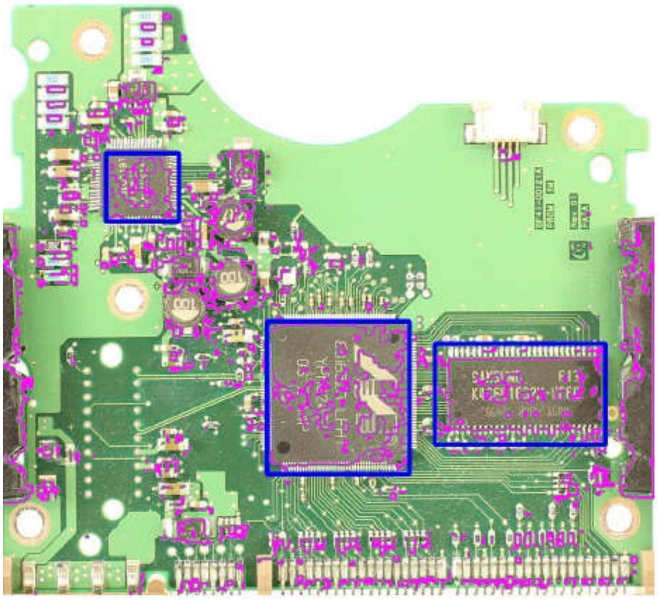


Fig. 13. Previous algorithm without GUI interface

Mathematic modeling: PCBs background estimation: At this stage, for each perfect bare PCB which will be processed in the OCR algorithm to detect text elements on the board such as punctuation, digits, letter, emoji and other text character elements, there are captured N continuous frames for studies purpose. Then, is observed a level pixel vector denominated as I, hence at this pixel level is stocked up a number in concern to n samples: the summary of vector "a" observed as well as the crossed product summary in concern to vector "b" observed.

$$n = \sum_{i=1}^{N} 1 \tag{1}$$

$$a = \sum_{i=1}^{N} Ii \tag{2}$$

$$b = \sum_{i=1}^{N} Ii \times Ii^{T} \tag{3}$$

Now is necessary to initialize the OCR algorithm, and this is equal from 3s to 4 seconds of algorithm initialization, in a general average. With this perspective, is believed that a simple (low computational cost) but a reasonable and accurate final model (low false-warning) is required to clustering the high throughput volume expected in the bare PCBs inspection using the OCR algorithm. From the sensors attached on the camera in the bare PCBs inspection system, the output vector at a pixel level is defined as the follow:

$$I = (R, G, B) \tag{4}$$

Observing the Eq.(4) is possible distinguish the RGB, hence these separated channels are codified in a (8:8:8) ratio. Thus, we have 8 bits colour information for each tree channels, and this will became our vector which we will learn through observation. The variations in the observed vectors at the pixel *I*, is modelled in concern to *K* Gaussians. Therefore, *P* (probability) of a pixel I belong to the background is given by, where *f* is a function:

$$P(I) = \sum_{i=1}^{K} \omega i f(I, \mu i, ci) \tag{5}$$

This model will be utilized to study the input vector on the camera sensors during the OCR algorithm running-time. Choosing a suitable mathematic modeling, we can to proceed for the description of algorithm results in concern to the inspection character time average for the samples 1, 2, 3, 4 and 5 during the execution running-time, as well as the frequency percentage sampling when the characters such as punctuation, digits, emoji, letter and other appears on the GUI interface of developed OCR algorithm. Results regarding to the text characters distribution are also described in the next section of this paper.

RESULTS AND DISCUSSION

The results appoint a minimum running-time for the PCBs text recognition in the OCR algorithm about 285ms during the experiment of text recognition. This can be represented by the Graphic 1, in the Fig. 14. This time summary is made automatically by the OCR algorithm, while frequencies results are summarized manually. The results about the efficacy detection precision capacity of the proposed OCR algorithm also can be exposed, in the Table 1 and Graphic 2, respectively, showing several lines of image classification, as well as the indication in the sample of the bare PCB image. Hence, it appoints an average about 90% better than the previous algorithms proposed in the works of [Stephan, 2015; We, 2018]. The text quantity sum per sample is shown in the Graphic 3, as well as the text character percentage frequency distribution from sample 1 to sample 5 is shown in the Graphic 4, where the sum is, respectively: 165 characters (sample 1), 75 characters (sample 2), 151 characters (sample 3), 17 characters (sample 4) and 30 characters (sample 5).

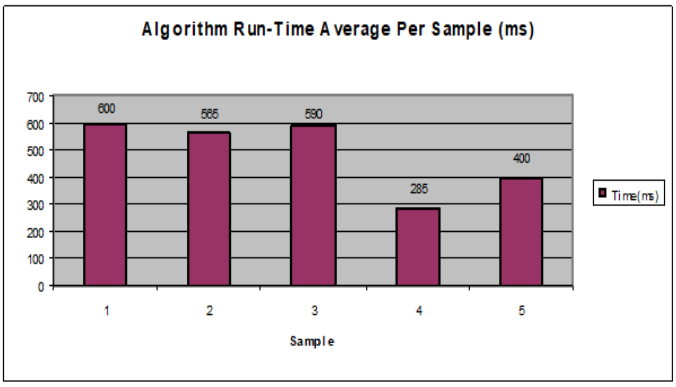


Fig.14. Graphic of the time average for the samples

Table 1. Sample classification frequency in the OCR algorithm

Sample	punctuation	digits	letters	emoji	other
1	45	36	29	27	28
2	28	15	12	11	9
3	45	36	29	27	14
4	10	3	2	1	1
5	18	8	2	1	1
Total	146	98	74	67	53

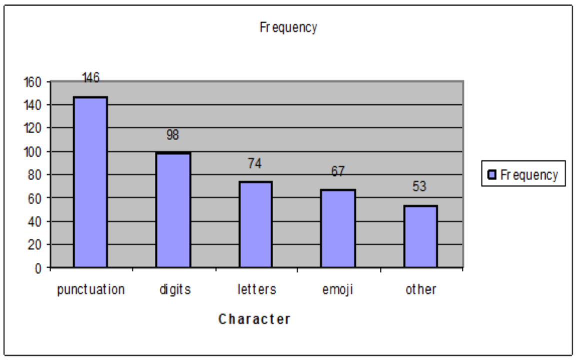


Fig.15. Graphic of the OCR distributions for the samples

Graphic 4 show the character frequency distribution per sample. The Graphic 4 represents the character frequency distribution for sample 1 to sample 5: text characters components such as punctuation, digits, letters, emoji and others (expressed in percentage) compounds the total text quantity detected by the OCR algorithm. Hence, we can account these elements. Also can be represented the percentage in concern to the text distribution along the bare PCBs boards for the samples 1, 2, 3, 4 and 5 through Graphic 5. Where the major correspondences of filled text distribution along the PCB board belongs to: sample 1 (41.67%), sample 3 (35.71%), sample 2 (32.26%), sample 5 (22.22%) and sample 4 (19.61%). The Graphic 5 in the Fig. 18 shows this correlation:

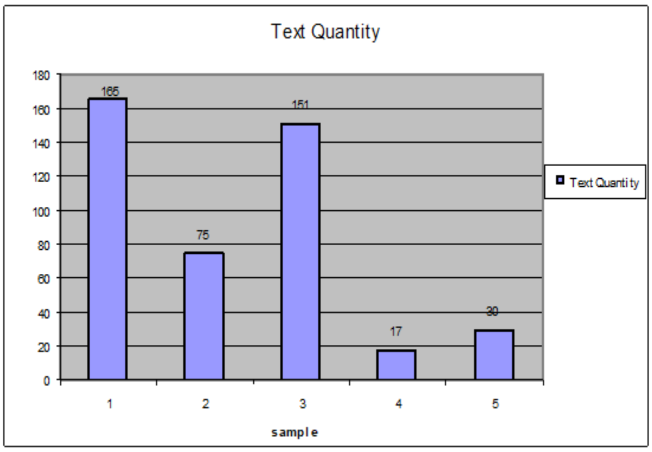


Fig.16. Graphic of the text quantity sum for the samples

Table 2. Total character frequency percentage in the samples

Sample	Frequency	Total	Frequency percentage
1	146	438	33.33
2	98		22.37
3	74		16.89
4	67		15.30
5	53		12.10

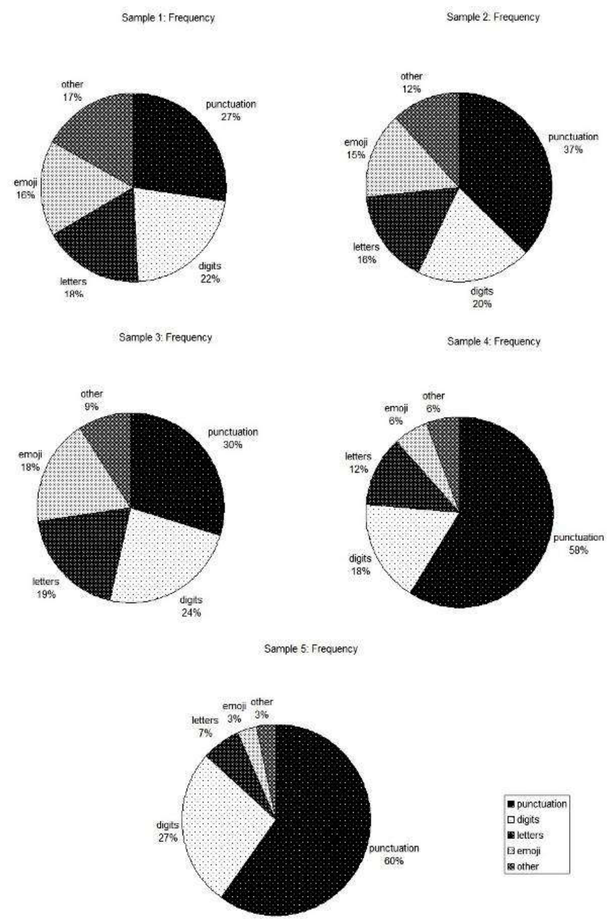


Fig.17. Text character sum frequency per sample

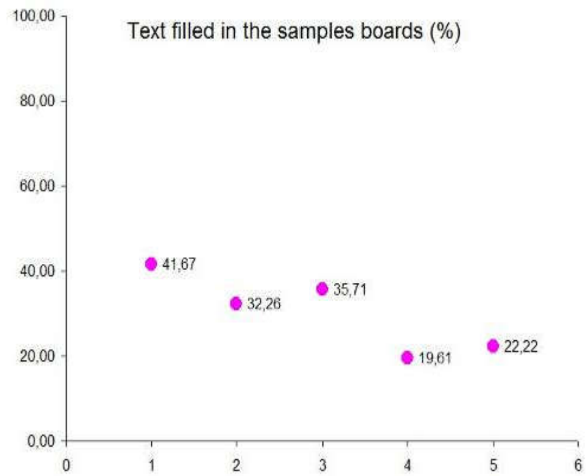


Fig.18. Correlation in concern to the text distribution along the PCBs

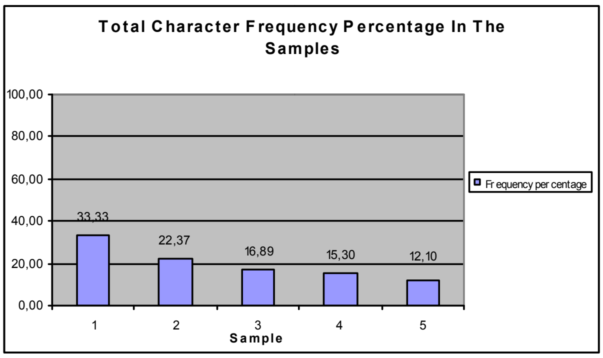


Fig.19. Graphic of the total character frequency percentage distribution in the samples

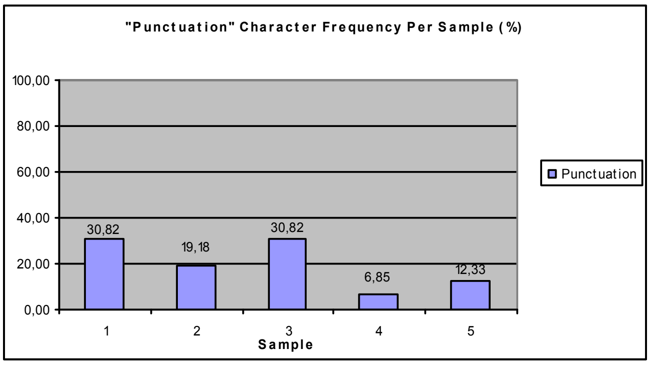


Fig. 20. Graphic of "punctuation" character frequency percentage distribution in the samples

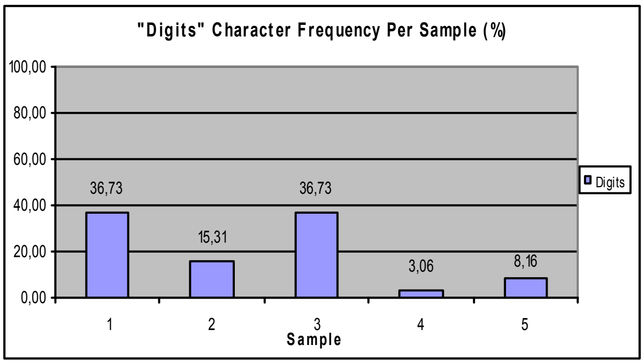


Fig. 21. Graphic of "digits" character frequency percentage distribution in the samples

Now, analyzing the Graphic 5, we can clearly notice that sample 1 has the major number of text elements filled along his board (more than 40%) and sample 4 has the minor number of text elements filled (less than 20%). It leads to make a suitable comparison with the time average for the samples: text elements filled on the PCBs boards are directly proportional to the inspection time values during the OCR running-time algorithm. This happens because as much as text elements are printed along the PCB boards, the bare board presents a computational time increased during the running-time algorithm to detect these elements, increasing in this way the inspection time processing of the whole system.

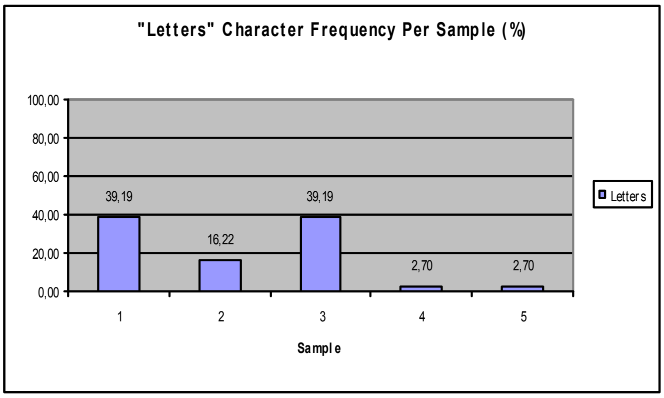


Fig. 22. Graphic of "letters" character frequency percentage distribution in the samples

Viewing the Graphic 1, this correspondence makes cense, because there is a percentage difference about 110.5% in relation to the inspection time of sample 1 (600ms) and sample 4 (285ms), respectively.

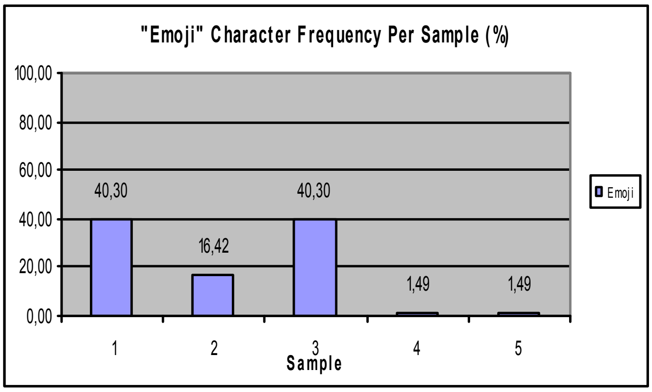


Fig. 23. Graphic of "emoji" character frequency percentage distribution in the samples

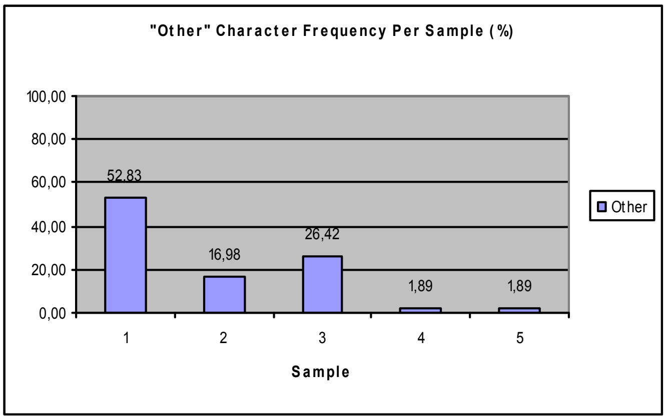


Fig.24. Graphic of "other" character frequency percentage distribution in the samples

Other results are in concern to the total character frequency percentage in the samples: viewing the Table 2 and Graphic 6, we can notice a major frequency percentage that text elements is detect on the sample 1 (33.33%) during the OCR algorithm running-time. In sequence, we can notice the minor frequency percentage (12.10%) detected on the sample 5, because there is less text elements printed along his board. The sum of total frequencies which text characters appear in the OCR algorithm is equal to 438 text elements detected. Thus, we can also represent graphically the character frequencies such as punctuation, digits, letters, emoji and others on the analyzed samples.

Conclusion

In this work was deduced a model commonly utilized in the field of image processing to represent a pattern detection (Object Character Recognition) of a bench composed by five main images (bare PCB text samples) of PCBs with several types of text printed in the board. Once defined the OCR algorithm parameters, also capable to identify the character symbol of those bare PCBs in the table of identification into a GUI window in MATLAB, the images running-time also can be processed. Thus, were also defined optimized implementation strategies to account the detected text elements in the OCR algorithm.

Acknowledgment

The research was realized in the Laboratory of Communications at the University of Campinas. We are thankful also for Zenodo.org by providing the bare text PCBs scanners utilized in this research.

REFERENCES

Andrew H. 2002. Hacking the Xbox: An Introduction to Reverse Engineering. No Starch Press, Inc, pp. 3-6.

Anitha D. B. and Rao M. 2019. SMT Component Inspection in PCBA's using Image Processing Techniques. International Journal of Innovative Technology and Exploring Engineering (IJITEE)., 8(12), 541-547.

Aruna P., Mishra S.N., and Das P. 2016. An Advanced LabVIEW based OCR system for Hand Written Scripts. International Journal of Multidisciplinary Research and Development., 3(2), 01-05.

Bellamkonda, D., Kandimalla, B., Koppula, A.K. and Bharath B.V.G.S. 2020. Image Classification Using Deep Learning Methodologies. International Journal of Research in Electronics and Computer Engineering., 8(1), 57-61.

Ben J. 2014. Reverse Engineering of Printed Circuit Boards. Tech. rep. Stanford – Digital Image Processing, pp. 7-10.

Clyde F. Coombs Jr. 2008. Printed Circuits Handbook, 6th Edition, McGraw-Hill Handbooks, United States, pp 133-135.

Daniel K.B. OCR Algorithm. 2019a. [Software] Mathworks Corporation. 2019.

Dilpreet K.D., Preethi B., and Karmjit S. 2013. An Interactive Speech Synthesis System based on OCR using LabVIEW. International Journal of Electronics Communication and Computer Engineering., 4(5), 06-10.

Jasvinder K. and Monika A. 2013. A LabVIEW Approach to Detect the Theft of Indian Currency Notes. International Journal of Advanced Research in Electronics Communication Engineering (IJARECE)., 2(1), 15-21.

Joe G. 2014. Printed Circuit Board Deconstruction Techniques. 8th USENIX Workshop on Offensive Technologies (WOOT 14). Texas, United States, 8th-9th Aug. pp 24-32.

Luís Gustavo de Mello Paracêncio 2009. Proposta de Metodologias para Integração de Células de Manufatura. Universidade Estadual de Campinas, Brazil. pp 3-6.

Maiboroda, V.S., Belajev, O.O., Dzhulii, D.Yu. and Slobodianiuk, I.V. 2020. AISI 1045 Flat Surfaces Machining Using the Magneto-Abrasive Method. Journal of Engineering Sciences., 7(1), A1-A7.

Moganti M., Ercal F., Dagli C. H., and Tsunekawa S. 1996. Automatic PCB inspection algorithms: A survey. Computer Vision and Image Understanding., 63(2), 287–313.

Raja, M., Agarwal, A. and Gautam, A. 2020. Integrated Inertial Navigation System with camera-based system using Lab VIEW Software. International Journal of Scientific Research in Physics and Applied Sciences., 8(1), 59-64.

Rui W., Wei L., Runnan Q., and JinZhong W. 2017. Blur Image Classification based on Deep Learning. IEEE., 6(4), 1-6.

Stephan K., Henrik F.N., and Frank K. 2015. Automated PCB Reverse Engineering. Institute of Distributed Systems, Ulm University., 8(2), 1-11.

Surface Mount Process. Introduction to Surface Mount Technology. http://www.surfacemountprocess.com. Access: March/24/2020.

We Li et al. 2018. Text Recognition for Information Retrieval in Images of Printed Circuit Boards. RWTH Aachen University., 4(1), 7-14.

Will B. Through-Hole Soldering Defects And The Solutions. http:// www.smtnet.com. Access: April/06/2020.

Ye T., Ming Z., and Mark P. 2017. Deep Learning in Photovoltaic Penetration Classification. IEEE Power & Energy Society General Meeting., 4(1), 1-5.

Zenodo.org. PCB DSLR Dataset. https://www.zenodo.org. Access: July/01/2020.

****