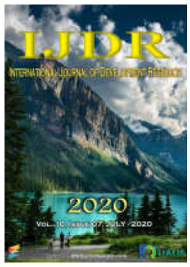
ISSN: 2230-9926

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

IJDR

International Journal of Development Research
Vol. 10, Issue, 07, pp. 37935-37942, July, 2020
https://doi.org/10.37118/ijdr.19231.07.2020



RESEARCH ARTICLE | OPEN ACCESS

EMBEDDED ARTIFICIAL INTELLIGENCE APPLICATION DEVELOPMENT PROPOSED FOR A PERFUME ENGINEERING DATABASE TREATMENT

***[1]Antônio Ricardo D'Araujo Amâncio Oliveira, [2]Mariana Lima Acioli Murari and [1]Paulo Caetano da Silva**

[1]Programa de Pós Graduação em Sistemas e Computação, UNIFACS, Stiep, Salvador - BA, 41770-235
[2]Escola de Arquitetura, Engenharia e Tecnologia da Informação, UNIFACS, Stiep, Salvador - BA, 41770-235

ARTICLE INFO

Article History:
Received 14th April, 2020
Received in revised form
10th May, 2020
Accepted 26th June, 2020
Published online 30th July, 2020

Key Words:

Cloud Artificial Intelligence,
FRAGRANTICA database,
Structured to Unstructured data conversion,
Image processing and recognition.

Corresponding author:
Antônio Ricardo D'Araujo Amâncio Oliveira,

ABSTRACT

Nowadays, databases are very useful for saving or registering data, information or events. Forecasting studies, autonomous decision-making control system became more accurate and easy to implement since databases had been invented. Artificial Intelligence applications show more and more effective when it comes to solving problems conventional computational algorithms cannot handle. They also show to be useful when the problem's exceptions are unknown or cannot be easily determined. This paper also summarizes the programming and hardware platform choice process, in order to develop an AI application for treating Fragrantica's perfume engineering unstructured database, by storing its information within structured XML documents and database. The platform choices were performed through special issues based available technologies comparative analysis. The developed AI application shall provide automatic retrieval of data from Fragrantica database, making possible the use of its information for perfume engineering related AI training. The main expected contributions are the converting procedures developed in order to turning unstructured data, as images and text, into XML structured documents which shall be extensible enough to support different context solutions.

Citation: ***Antônio Ricardo D'Araujo Amâncio Oliveira, Mariana Lima Acioli Murari and Paulo Caetano da Silva.*** "Embedded artificial intelligence application development proposed for a perfume engineering database treatment", *International Journal of Development Research*, 10, (07), 37935-37942.

INTRODUCTION

Databases are important for keeping events records, monitoring and identifying trends, supporting decision making, and training artificial intelligence applications. The bigger the phenomena data the greater results coherence and accuracy attained performing such tasks. As the communication and information technologies evolved, international research supportive open access online databases became more common. However, there are many kinds of data stored under different multimedia content such as text, images, videos, and other kinds of non-structured data, so retrieving information from those databases can be a real challenge. Perfume engineering is responsible for developing several fragrances perceived in industrialized daily consumed products. It also demands database analysis through AI applications for perfume development, profit maximization, easing production and productivity management, forecasting product acceptance and consumption trend studies, and cognition of volatile components from product samples. Artificial Intelligence (AI) training requires extensive databases so that their predictions and actions correspond to those predicted with accuracy. The most convenient ways to store and sort data are not the best for processing them. When it comes to heterogeneous databases its necessary to treat and organize raw data before processing it, in order to ease AIs' training, implementation, and validation processes. Organizing and processing a heterogeneous database would require a careful item to item analysis. Handling it manually would not be practical given the amount of stored data. Conventional computational algorithms might not be suitable for converting images into structured data, once it demands pattern recognition and classification criteria are not Boolean. So, an AI application (AIApp) would better perform data extraction from such heterogeneous platforms providing specific treatment and organization. Programming platforms for AIApp development offer different facilities that impact applications' compatibility, computational resource usage, complexity, efficiency, and ease of local or cloud implementation. C, Java, and Python are all valid programming platforms examples, but it's important to compare them, from the AIApp demands approach, in order to find out which one better suits the solution's deployment. It is also important to choose a platform for embedding the AIApp which enhances or optimizes the application's performance, following an analogous procedure. This paper proposes an embedded AIApp for structuring and organizing a perfumery heterogeneous database through XML documents, hence XML has been adopted as a way to integrate multiple data sources and data formats in a standard format, in order to make its information useful for perfume engineering-related AI training. The rest of this paper is organized as follows. Section II contextualizes why Fragrantica's database is the proposed AIApp's target. Section III presents a brief introduction to AI, the AI programming language choice process, the platform selection process for embedding the AIApp, and some researched AI applications and methods for image recognition. Section IV describes the proposed AIApp. Section V brings the conclusions and future works related to the proposed project.

Fragrantica

Fragrantica is a free access database example of perfume engineering with enough content available for AIs training, due to the number of specialists in perfumery and public conducted research deposed there over the years. However, it is filled with structured (numbers, keys, labels, etc.) and non-structured (mostly images and text) data which makes it hard to retrieve useful computational information from it. Fragrantica's database (Fragrantica, 2017) showed to be the one with the biggest amount of free available data, costumer perceptional research data, professionals and costumers feeding among different perfumery databases such as OSMOZ (Osmoz, 2019), basenotes (Basenotes, 2019), Now Smell This (NOW SMELL THIS, 2019), parfumo (Parfumo, 2016) and Michael Edwards (Michael Edwards, 2019). So, it was selected as the target database. Fragrantica's database demands a specific data treatment that requires repetitive, accurate, and busywork data would be sorted in a standardized structured way, in order to become useful for computational analysis and AIs applications' training. The proposed AI application shall provide automatic retrieval of data from the Fragrantica database, through pattern recognition and data sorting, making possible the use of its information for perfume engineering-related AI training.

Artificial Intelligence

Artificial intelligence is a software system or mechanism based on human intelligence adaptation. The development of such "intelligent" machines and computers seek to optimize problem-solving for situations which require human reasoning to be achieved. According to Intel (2017), artificial intelligence, while supported by Moore's law, promises to transform the society much more than industrial revolutions. This technique accelerates problem-solving in many areas such as medicine, finance, education, and science, because of their learning and decision-making capabilities. Solving a problem with traditional structural algorithms is a valid alternative when the proposed problem has some peculiar characteristics such as its solution has to be known and broken down into small, concise instructions; all of its exceptions must be known hence the inputs with unexpected exceptions are not handled properly. Alternatively, AI applications allow solving problems whose solution is unknown but learned by recognizing tolerance patterns and applications, both during its training and throughout its operation. They "learn" through self-modifying algorithms that increase or decrease the result acceptance over the standard used in their training. According to Intel (2017), they are mostly applied in pattern recognition and system behavior inferences research. The use of artificial intelligence for image recognition based on previously registered ("learned") patterns increases every day. However, through the bibliographic survey carried out, no records about AI developed for perfume engineering-related database treatment were found.

MATERIALS AND METHODS

This section aims to present the methodology to select the technologies that will allow developing the proposed AI. It consisted of analyzing available technologies to verify which ones would better support the application's development.

AI programming language choice

Several programming languages can be used for AI applications' implementation. However, each one of them has its own peculiar aspects (load time, memory consumption, number of instructions, programming complexity, etc.) which makes a programming language better or worse than others for the solution's development. The Fragrantica database has information such as product description, product fragrance composition, and customer satisfaction surveys organized as image files. The proposed AI application aims to process and convert image non-structured data into structured data. A programming platform analysis showed which one would best fit the solution needs. According to previously done bibliography research, the most used programming languages for image processing and recognition AI application development are Java, C++, and Python. An empirical comparison between seven different programming languages (C, C ++, Java, Perl, Python, Rexx, and Tel) was conducted by Prechelt (2000) and allowed verifying selection criteria for the most suitable one. In his case study, Prechelt (2000) obtained eighty codes in the evaluated languages developed by different programmers, but for the same purpose: to solve the phone coding problem. This challenge consists of associating telephone numbers, previously loaded in a file, with strings, returning all the possible words formed by each telephone number. Programs were compared considering different criteria such as loading and data initialization, processing, memory consumption, code extension, inserted comments, reliability, runtime, and productivity. The statistic results Prechelt (2000) obtained are shown in Figures 1 and 2. Based on Prechelt's (2000) data, Python codes had smaller programming time and also fewer code lines than those in C / C ++ and Java (Figures 1b and 2b, respectively). Programs in Java consumed more memory than Python, which consumed more memory than programs in C / C ++ (Figure 2a). Python programs had highest the boot and data loading time. On the other hand, Programs in C / C ++ were the fastest ones in this step. Java programs performed this task faster than Python ones but, slower than C / C ++ ones. Programs in Python spent less processing time than those in Java and C / C ++ (Figure 1a), according to Prechelt (2000).

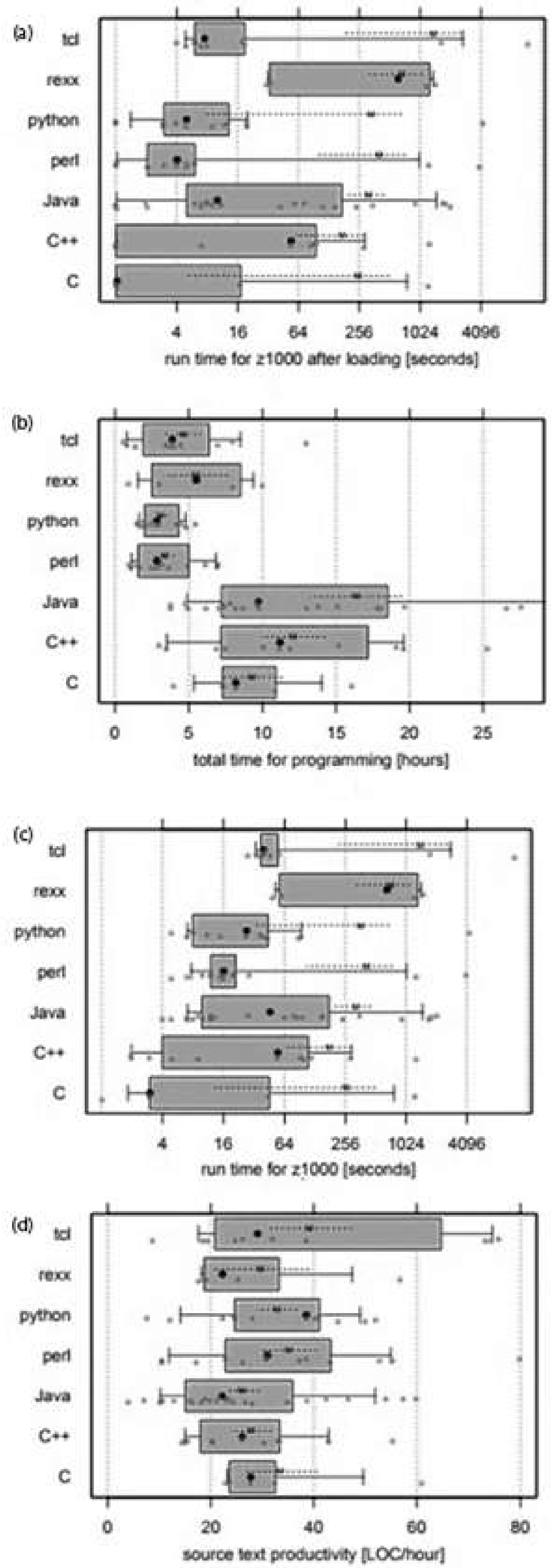


Figure 1 - Statistical performance comparison between C, C ++, Java, Perl, Python, Rexx, and Tel. Part 1. Adapted from Prechelt (2000)

Analyzing the lines of code per program (Figure 2b) and the number of comments graphs, it is notorious C / C ++ programs require a greater effort to be performed and require greater programming care when choosing a suitable compiler and handling exceptions. Those comparatives analyze are only valid for the telephone code application. Shifting the application, the results obtained can be influenced by each language's libraries and specific characteristics. Hence, different applications demand applying different libraries and functions set, as each one of them has its way to manage hardware resources, functions that accomplish the same task will have different computational cost for different kinds of solution development.

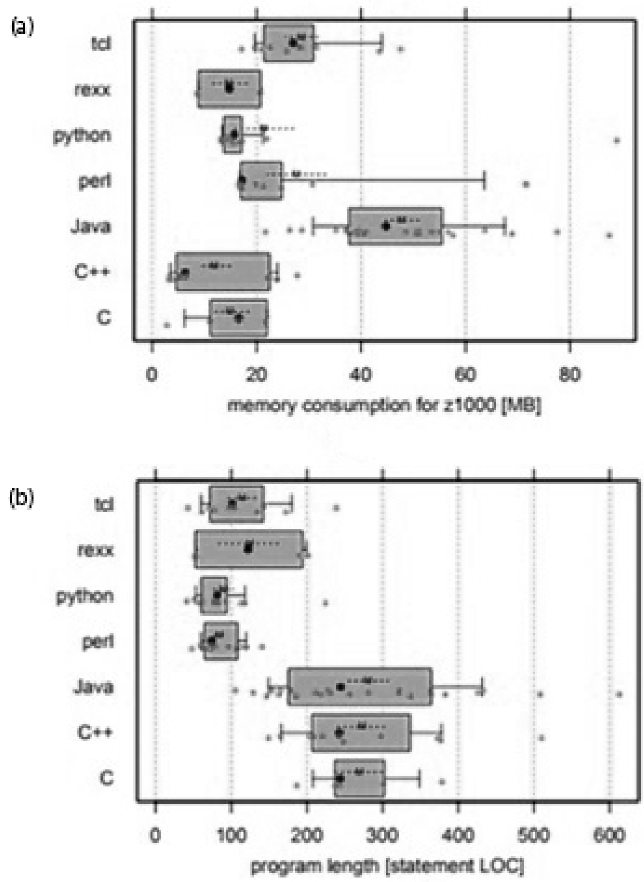


Figure 2. Statistical performance comparison between C, C ++, Java, Perl, Python, Rexx, and Tel. Part 2. Adapted from Prechelt (2000)

Aruoba and Villaverde (2017) performed comparative studies between different programming languages, (C ++, Fortran, Java, Julia, Matlab, Python, R, Mathematica, and hybrid programs) for a neoclassical stochastic growth model mathematical processing application. The tests were performed on Mac and Windows platforms using more than one compiler for the same language on both operational systems. The run-time test (Figure 3) showed C ++ as the most efficient. Java programs were two to three times slower than C ++ ones. Python programs showed to be more than thirty times slower than the equivalent one in C ++, according to Aruoba and Villaverde (2017). Comparing the codes extension for this application (Figure 4), Python codes, although slower, were smaller than Java ones, which were smaller than C ++ ones (Aruoba and Villaverde, 2017). No case studies comparing different programming languages related to image processing and recognition AI applications were found. Characteristics such as multiplatform programs and the number of support tools (libraries), criteria were not discussed by Aruoba and Villaverde (2017) and Prechelt (2000) but had to be considered for selecting the AI language development.

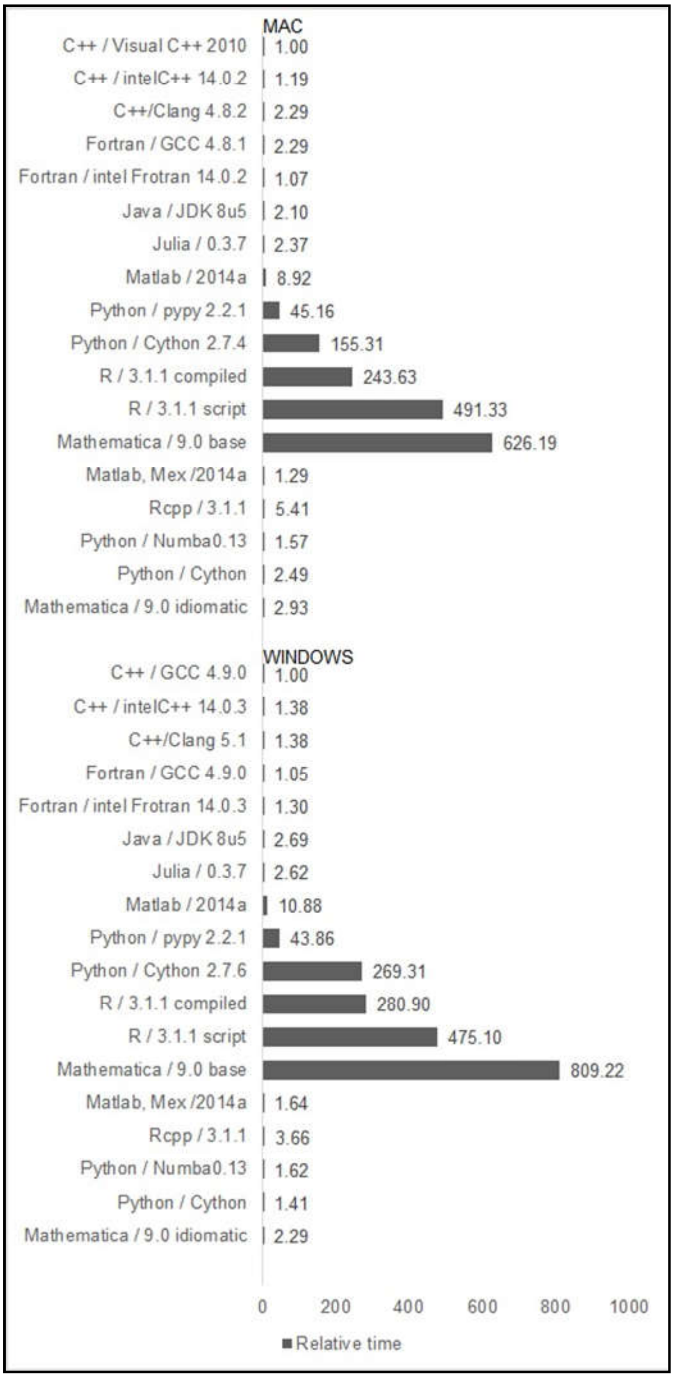


Figure 3. Relative program execution time. Adapted from Aruoba and Villaverde (2017)

Java language showed an intermediate performance to Python and C / C ++ in both studies. Although it is slower than C ones, Java algorithms have fewer exceptions handling errors than C does. Even with shorter algorithms, Python proved to be less efficient than C and Java for loading files to be handled by the algorithm, which is a critical task for the proposed AI. C algorithms are faster than Java ones, but they aren't fully multiplatform and suffer significant interference from the compiler used. On the other hand, Java is a multiplatform language, so, it demands no adaptation to run on any of the three main existing operational systems platforms (Linux, Mac, and Windows). The macroeconomics application required more mathematical content processing than the Prechelt's (2000) test and Java showed better run time than Python. Access online data and operate XML files can demand high insensibility to exception treatment errors, hence it is not desired to corrupt data that shall be used for AIApp.

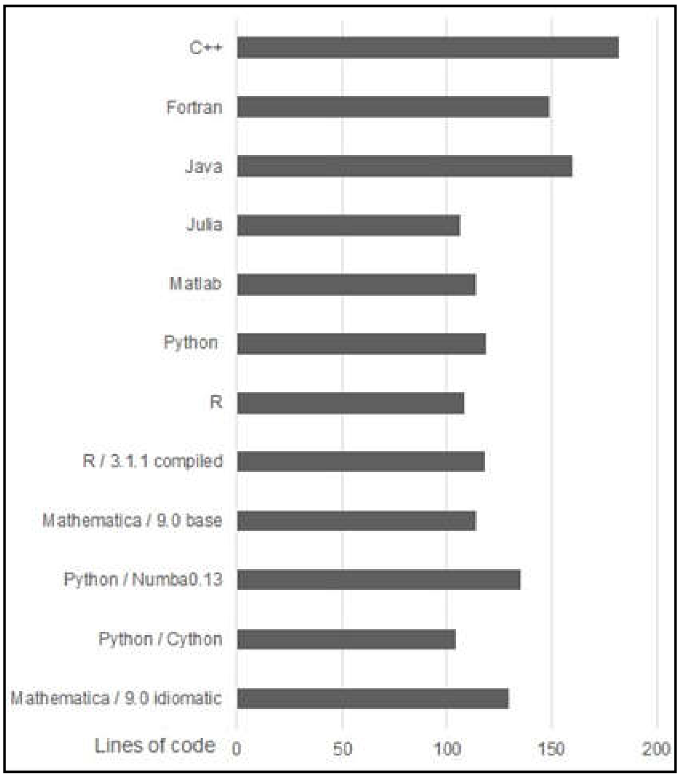


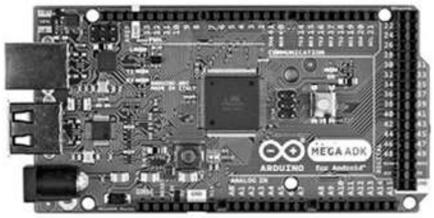
Figure 4. Aruoba and Villaverde codes' length. Adapted from Aruoba and Villaverde (2017)

The proposed application shall open, read, and create XML files, so efficiency when it comes to loading files to be processed is a relevant characteristic. Analyzing the amount of documentation and forums related to image treatment through AI available, for each one of the programming languages, Java platform showed to have the most extensive and consolidated documentation. Java also has a greater amount of image processing libraries, which may reduce the proposed application's programming time due to code reuse. Even if the solution is developed in a cloud environment JSON would still allow implementing the AIApp in Java, despite programming complexity increment, hence it is possible to convert JSON (Ecma, 2017) to XML (Lian Quin, 2019) using Json.NET (Newtonsoft, 2019), for example. Considering Prechelt's (2000) and Aruoba and Villaverde's (2017) special issues Java programming language proved to be the most suitable one for the proposed AI application's development.

Platform selection for Embedding the AIApp

According to the literature, three solutions were mentioned as suitable for embedding and processing image recognition applications: Raspberry (Figure 5b), for the proposed AI Cloud Computing Servers. The Arduino platform provides solutions in wiring code for image recognition and it is also compatible with Python code. However, it was not considered worth due to the high processing and memory demand for proposed AI application before the actual micro-controller processing capacity.

(a)



Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

(b)

Processor
• Broadcom BCM2387 chipset.
• 1.2GHz Quad-Core ARM Cortex-A53 (64Bit)
802.11 b/g/n Wireless LAN and Bluetooth 4.1 (Bluetooth Classic and LE)
• IEEE 802.11 b / g / n Wi-Fi. Protocol: WEP, WPA WPA2, algorithms AES-CCMP (maximum key length of 256 bits), the maximum range of 100 meters.
• IEEE 802.15 Bluetooth, symmetric encryption algorithm Advanced Encryption Standard (AES) with 128-bit key, the maximum range of 50 meters.
GPU
• Dual Core Video Core IV® Multimedia Co-Processor. Provides Open GL ES 2.0, hardware-accelerated Open VG, and 1080p30 H.264 high-profile decode.
• Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure
Memory
• 1GB LPDDR2
Operating System
• Boots from Micro SD card, running a version of the Linux operating system or Windows 10 IoT
Dimensions
• 85 x 56 x 17mm
Power
• Micro USB socket 5V1, 2.5A
Connectors:
Ethernet
• 10/100 BaseT Ethernet socket
Video Output
• HDMI (rev 1.3 & 1.4)
• Composite RCA (PAL and NTSC)
Audio Output
• Audio Output 3.5mm jack
• HDMI
• USB 4 x USB 2.0 Connector
GPIO Connector
• 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip
• Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector
• 15-pin MIPI Camera Serial Interface (CSI-2)
Display Connector
• Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot
• Push/pull Micro SDIO

Figure 5. Arduino (a) and Rapsberry (b) features

For example, Bayir and Albayrak (2018) developed an Arduino -supported neural network model in order to obtain information about the general situations of beehives. However, tasks such as real-time measurement and prediction, and the neural network train quality were limited or compromised due to Arduino's prototyping board lack of computational power. The Raspberry PI3 is a low cost, credit-card sized computer, which has a clock close to that of ordinary computers, has enough RAM and ROM for running the onboard plating along with an operating system, unlike Arduino micro-controllers. Despite showing greater computational power and resources than the Arduino prototyping board, Raspberry boards do lack processing capacity and go under short with some processes in image processing, when it comes to greater complexity treatment of unwanted variables and noises, which requires a high-performance cost of a processor. For example, Ortiz, Marin, and Gualdron (2016) developed a security system to generate an alarm event evaluating possible situations of robberies through computer visualization and a neural network AIApp within a Raspberry PI, whose greatest drawback was the impossibility to apply a treatment of unwanted variables techniques due to the board limited processor. Cloud computing helps to deal with several aspects related to the aforementioned challenges hence its computational power is not as limited as local devices (PC, prototyping boards) does. Cloud computing platforms put down obstacles such as updates, infrastructure, interoperability between devices, deployment cost, insufficient hardware resources, lack of scalability, and much more. But cloud IA applications demands high network bandwidth and low network latency (Zhang et al., 2018), which are critical QoS and QoE parameters. Bandwidth requirements depend on how many cloud users are online, how they are geographically distributed (tracking accuracy), and how the distribution scheme (which affects the virtual environment scalability) will handle it (Skorin-Kapov et al., 2004). Raspberry platform better supports AIApps than Arduino. Due to its low cost and complexity, Raspberry is more suitable for testing and validating AIApps. On the other hand, Cloud computing better suits AIApps definitive implementation and usage.

AI applications and methods for image recognition

The best suitable database treatment, through image recognition, solutions among the case studies identified in the literature review were Cannesson's et al. (2007) and Pan's et al. (2016). The former developed an application that automatically calculates the ejection fraction using two-dimensional echocardiographic images. The application calculated values were similar to manual calculation results and also reached lower variability than traditional visual recognition. According to the American College of Cardiology Foundation, this AI has clinical potential. The latter developed a method of facial recognition for single-sample training image cases, through LPP (Locality Preserving Projection) and optimized features transfer. The true challenges for facial recognition methods and applications consist of inadequate image training samples from the same face (PAN et al., 2016). When there is only a single image sample to train the application, keeping up with accuracy becomes more critical. Several factors such as brightness and angle also make facial recognition a challenge. The conventionally used learning methods also struggle to deal with such obstacles (Pan et al., 2016). According to Pan et al. (2016), there are two main types of single training sample image recognition methods: global face recognition and local face recognition. The first one is based on Principal Component Analysis (vector orthogonalization for components transform, observations, and non-linearly correlated values). It provides generalization capability to applications. It can also be optimized through noise models depending on the kind of image selected. Global face recognition applications can also be trained in crafting virtual samples from the training images in order to increase adaptability. In Pan's et al. (2016) case study, virtual samples were generated by adding noise, filtering, applying wavelet transform, applying image reconstruction (based on the Contourlet transformation), analyzing the image kernel main component, applying the discriminatory generalized analysis and generating multiangle features with Gabor filter.

For Pan et al. (2016), feature transfer learning is an inductive one. So, considering a given source domain and task, a target domain and task, the feature transfer learning looks for a common LD (low-dimensional feature representation) that associates source tasks with target tasks by reducing the difference between them. According to Pan et al. (2016) the popular methods for LD face recognition, include Eigenface, Fisherface, and Laplacianface. The most important task is processed in the transfer source selection stage, where each selected sample which shows the same target training sample macro characteristic, pass through the further feature transfer steps: source images selection; true testing sample replacement; and source images of the macro characteristic of the testing sample selection. The algorithm validation used NIST's FERET (Face recognition technology) data and the results obtained (such as features extraction and global information loss) were better than those from other methods described by Pan et al. (2016) in their paper, considering the receiver operating characteristic curve of each one of them. An attenuation correction synthesis for Positron Emission Tomography (PET) data was proposed in Burgos et al. (2014) case study. It says that PETs and Magnetic Resonance (MR) imaging scanners must have their data photon corrected in order to provide greater precision in neuro-oncological, epileptic, and neurodegenerative diseases detection. According to Burgos et al. (2014), the most used techniques that perform this image treatment are emission, segmentation, and approaches based on atlas. The first one of them estimates attenuation maps based on exploiting information in the PET.

The segmentation-based methods apply uniform linear attenuation coefficients to tissue classes from MRI (magnetic resonance image) segmentation. Atlas-based methods use models from patient data and apply attenuation maps to deform it in order to match the patient's anatomy. The method proposed by Burgos et al. (2014) follows the principle of multi-atlas propagation. It exploits the concept of morphological similarity in order to synthesize an attenuation correction map from an MRI. The main challenge is that linear coefficients for body tissues, air, and sinus tissue, for example, have very close coefficients and are not easily differentiable. Thus, images' generation and reconstruction algorithms must have the robustness to perform satisfactory accuracy. The algorithm uses a pre-acquired set of aligned MRI/CT image pairs, from multiple subjects. The code generates the CT (computed tomography) intensities, corresponding to similar MRIs, by propagation in a voxel-wise fashion. An atlas database with pairs of T1-weighted MRI and CT brain images in Burgos et al. (2014)'s algorithm validation experiment. The similarity measure shows the match quality between the target subject MRI and each of the warped atlas database MRIs. The convolution-based fast Local Normalized Correlation Coefficient (LNCC) and the local Normalized Sum of Square Differences (NSSD) were the criteria used to evaluate the similarity measures. The developed application validation consisted of applying data from 41 different patients. The same test was performed by currently used platforms. The accuracy criteria for pseudo-computed tomography and for the attenuation map were previously set. These criteria ended up being the method's most outstanding advantage before the conventional ones in the PET generated image reconstruction. The method and results presented by Pan et al. (2016), might be useful for developing the proposed solution hence Fragrantica's online database only has one image sample for each perfume, whose data and information shall be extracted and regrouped. The image classification accuracy criteria, the image segmentation approach, as well as the voxels treatment applied in Burgos et al. (2014) case study, shall be used to the proposed solution for the Fragrantica database at the results validation stage and during the image information extraction and treatment stage. Both special issues difficulties and obstacles shall perform during the development of the proposed IA, so the presented solutions shall provide directions about how to deal with the upcoming ones.

RESULTS

A method, a system, and a program product for searching unstructured data, through SQL query, were developed by Choi et al. (2006). It searches the unstructured data, as text data or image data, returns the search results, and makes the attributes available as structured data in a relational SQL database. A method and apparatus for organizing and integrating structured and non-structured data across heterogeneous systems were developed by William et al. (2005). The invention identifies, aggregates, and makes accessible information from multiple heterogeneous files. Even though methods and strategies for turning unstructured data into structured data are extensible to different applications, its processing and treating make each developed solution specific or even unique due to databases specific information content. Fragrantica's database has its information organized as text and images sets. Each one of them demands different kinds of processing and treating steps before being converted in a structured data format as XML which has been adopted as a way to integrate data in a standard format (Santos, 201X). Whenever a user or an external source (another AIApp, for example) requests specific perfumery information, if there is no structured data previously retrieved, the proposed AI application shall interact with Fragrantica's online database, through unstructured data searching mechanisms.

Once requested data is found, applying an AI algorithm for classification, it is converted into structured data. Then, it is saved within an XML document which is returned to the requester as search results and stored in an XML database for a future retrieval request. Storing retrieved data into XML database decreases the system computational cost and effort, hence it has to perform navigation and information gathering through XML mechanisms as XLink (W3C, 2010), XLpath (Santos, 201X) or XPath+ (Silva, 2009) instead of interacting with Fragrantica's database and running the AI algorithm in order to perform the same procedure. The proposed solution has a two-step implementation model (Figure 6). The first step (Figure 6a), is for testing, evaluation, and validation purpose.

Some Fragrantica's database webpages reference will be available in a local (micro-controlled board, such as RaspBerry) directory as well as the Java AI Application and the XML database. The device's internal storage size will not be a problem due to its extensibility through SD cards inserted. Java has interoperability between operating systems (OS), so a RaspBerry device running a version of Linux OS, will not add extra costs to the project.

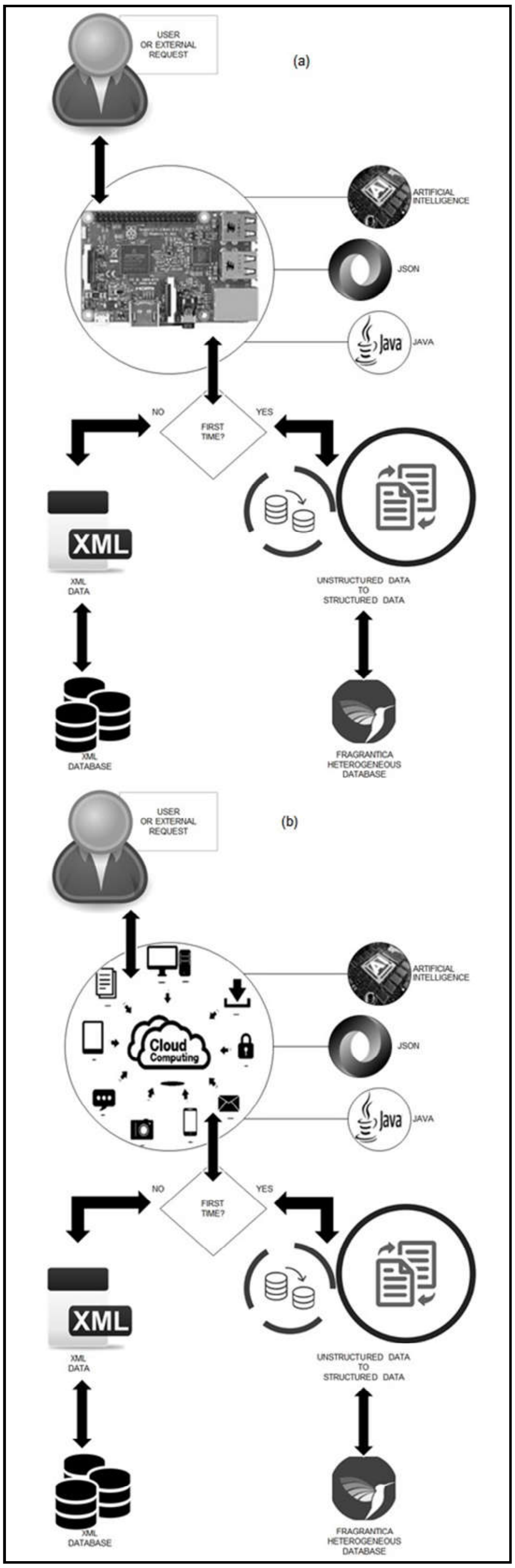


Figure 6 - Fist implementation step (a) and second implementation step (b).

The update step will be a user-requested operation which will be demand data retrieval step if it has never been done before with the device. So, the user will be asked about which product name he or she desires to get information from. Then, the developed application shall automatically access Fragrantica's database, extract the product-related images aside from other unstructured data and, generate an XML file containing a table filled with the searched product information. The information will be returned to the user and recorded in the XML file which will be used as the reference for sample comparison when the same information is requested twice or more. The Java programming environment NetBeans8.2 is compatible with a plenty of libraries and functions which will provide the unstructured data treatment, the XLM with plenty of interaction between the application and the Fragantica's online database. As discussed in section III, RaspBerry Device struggles handling the advanced image processing demanded by the solution proposed, however, it allows checking application's functionality, AI calibration accuracy and testing developed search mechanisms parameters through a low-cost prototyping platform. In order to test Fragrantica's unstructured data conversion into XML structured data (second step), any discussed micro-controlled board will not provide the computational power demanded or meet QoS and QoE parameters. So, a cloud domain implementation (Figure 6b) would work properly, as discussed in section 3.2. In a user or external request, the cloud application will fist navigate its XML online database searching for the information. If data is already available within an XML document, it is returned to the requester. If no data is found, the cloud AI application searches for it on actual Fragantica's online unstructured database. Once it is identified, the data converting process gets started. The retrieved structured data returns to the requester and it is also stored as an XML document in the online XML database.

DISCUSSION

The image recognition and database reorganization with artificial intelligence application development is necessary since the bibliographical survey did not reveal any developed tool capable of treating Fragrantica's image database. The main expected contributions are the converting procedures developed in order to turn unstructured data, as images and text, into XML structured documents for both platforms: RaspBerry and cloud. While the former provides low-cost functional verification and prototyping the latter shall grant computational power, QoE, and QoS parameters for converting Fragrantica's unstructured data into XML structured data. Despite the information retrieval procedures' strict specificity, which makes it harder to extend its usage to other applications, the aforementioned ones shall be extensible enough to support different context solutions. As a future task, the proposed AIApp shall be deployed, using the selected technologies among the ones discussed in this paper, and the results obtained will be discussed as well as eventual QoS, document structuring and any other relevant issues faced.

REFERENCES

ARUOBA, S. Boragan. VILLAVERDE, Jesus Fernandez . A comparison of programming languages in macroeconomics. Journal of economic dynamics and control. Usa, v. 58, p.265-273. 22 may. 2015.

BASENOTES COPYRIGHT 1999 2019. About Basenotes. Availlable at: http://www.basenotes.net/about/. Accessed on: 25th jun. 2019.

BAYIR, Raif; ALBAYRAK, Ahmet. The Determination of the Developments of Beehives via Artificial Neural Networks. Tehnicki Vjesnik - Technical Gazette, [s.l.], v. 25, n. 2, p.553-557, abr. 2018. Mechanical Engineering Faculty in Slavonski Brod. http://dx.doi.org/10. 17559/tv-20160419130812.

BURGOS, Ninon et al. Attenuation Correction Synthesis for Hybrid PETMR Scanners: Application to Brain Studies. IEEE TRANSACTIONSON MEDICAL IMAGING, VOL. 33, p.2332-2341, dez. 2014. Availlable at: http://ieeexplore.ieee.org/document/6858020/?arnumber=6858020 Accessed on: 1st Dec. 2017

CANNESSON, Maxime et al. A novel two dimensional echocardiographic image analysis system using artificial intelligence: Learned Pattern Recognition for Rapid Automated Ejection Fraction. Journal of the american college of cardiology. Pittsburgh Pennsylvania, v. 49, p.217-226. 16 jan. 2007.

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION- ECMA. ECMA-404: The JSON Data Interchange Syntax. 2 ed. Genebra: Ecma International, 2017. 16 p.

FRAGRANTICA Copyright INC. Perfume Encyclopedia, 2017. Availlable at: <https://www.fragra ntica.com/>. Accessed on: 12nd Oct. 2017.

INTEL Copyright. Artificial Intelligence, 2017. Availlable at: <https://www.intel.com. br/content/ www/br/pt/analytics/artificialintelligence/ overview.html>. Accessed on: 12nd Sep. 2017.

INTERNATIONAL BUSINESS MACHINES CORPORATION (Armonk). United States Patent. Choi, Arthur et al. REAL-TIME AGGREGATION OF UNSTRUCTURED DATA INTO STRUCTURED DATA FOR SQL PROCESSING BY ARELATIONAL DATABASE ENGINE. US n USOO7146356B2, 21 mar. 2003, 05 dez. 2006.

LIAM QUIN. Extensible Markup Language (XML). Availlable at: ¡https://www.w3.org/XML /¿. Accessed on: 25th jun. 2019.

MICHAEL EDWARDS 2019. About us. Availlable at: http://www.fragrancesofthewor ld.com/About. Accessed on: 25th jun. 2019.

NEWTONSOFT COPYRIGHT 2019. Json.NET: Popular high-performance JSON framework for .NET. Availlable at: ¡https://www.newtonsoft.com/json¿. AAccessed on: 25th jun. 2019.

NOW SMELL THIS COPYRIGHT 2005-2019. About Now Smell This. Availlable at: ¡https://nstperfume.com/about-now-smell-this/. Accessed on: 25th jun. 2019.

ORTIZ, Jesus; MARIN, Andres; GUALDRON, Oscar. Implementation of a banking system in embedded systems using artificial intelligence. Advances In Natural And Applied Sciences, [s.l.], v. 17, n. 10, p.95-101, dez. 2016.

OSMOZ - SHARE YOUR FRAGRANCES COPYRIGHT 2001-2019 — CREATED IN PARIS IN 2001. OSMOZ - Share your fragrances. 2019. Availlable at: http://www.osmoz.com/. Accessed on: 25th jun. 2019.

PAN, Jei et al. Single sample face recognition based on LLP feature transfer.IEEEAccess Multidisciplinarity, Rapid review, Open access journal. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China, v. 4, p.2873-2884. 11 abr. 2016.

PARFUMO COMMUNITY. About us. Availlable at: https://www.parfumo.net/About us. Accessed on: 25th jun. 2019.

PRECHELT, Lutz. An empirical comparison of c, c++, java, perl, python, rexx, and tcl. Ieee computer. Karlsruhe, Germany, 14th mar. 2000.

SANTOS, Marcelo Mendona dos; SILVA, Paulo Caetano da; TIMES, Valria Cesrio. XLPATH: A XML LINKING PATH LANGUAGE. [s.l.] , 201X.

SILVA, Paulo Caetano da; TIMES, Valria Cesrio. XPath+: A Tool for Linked XML Documents Navigation. Database And Xml Technologies, [s.l.], n. 5679, p.67-74, 2009. Springer Berlin Heidelberg. http://dx.doi.org/10.1007/978-3-642-03555-56:

SKORIN-KAPOV, Lea et al. Analysis of End-to-End QoS for Networked Virtual Reality Services in UMTS. Ieee Communications Magazine, [s.l.], v. 0, n. 0, p.49-55, abr. 2004.

VANDERBILT UNIVERSITY. United States Patent Application Publication. Stead, William W. et al. METHOD AND APPARATUS FOR ORGANIZING AND INTEGRATING STRUCTURED AND NON-STRUCTURED DATA ACROSS HETEROGENEOUS SYSTEMS. US n US 200602878.90A1, 15 jun. 2005, 21 dez. 2006.

W3C COPYRIGHT 2010. XML Linking Language (XLink) Version 1.1. Available at: ¡https://www.w3.org/TR/xlink11/¿. Accessed on: 25th jun. 2019.

ZHANG, Lanshan et al. Unlocking the Door to Mobile Social VR: Architecture, Experiments and Challenges. Ieee Network, [s.l.], v. 32, n. 1, p.160-165, jan. 2018. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109 /mnet.2017.1700014.

****