Proceedings of the Twelfth International Conference on
Engineering Computational Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 8, Paper 5.2
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.8.5.2

Optimized Parallel Software Architecture for Industrial Materials Sorting Systems

M.O. Martinez-Rach[1], H. Migallón[1], O. López-Granado[1], C. Pérez-Vidal[2] and R. Morales[3]

[1]Department of Computer Engineering, Miguel Hernández University, Elche, Spain
[2]Department of Systems Engineering and Automation, Miguel Hernández University, Elche, Spain
[3]Research and Development Department, Jovisa S.L. , Muro de Alcoy, Spain

Abstract

The introduction of new technologies in the industry makes possible to develop new functionalities in production lines, but it requires to solve some challenges in several areas. One of these challenges is the hardware and software design of the computational system, taking into account both the time requirements of the industrial system and the economic requirements. In the field of sorting and collecting pieces of different composition but with similar appearance, hyperspectral cameras have recently been introduced in the industry. They provide more information than conventional vision cameras, but it must be transmitted, pre-processed and analysed depending on the final application of the system. Obviously, the computing cost increases as the volume of data to be processed does. The system's computing power must be increased if it is unable to process all the information in real time. Distributed software architectures must therefore be designed to efficiently use the computational power of parallel computing systems, since computational power increases primarily by the number of processing elements increment rather than by increasing the power of each processing element itself. In this paper we present a software architecture design that distributes heterogeneous work and accelerates computationally expensive processes to meet the industrial system time requirements.

Keywords: Industry 4.0, parallel systems, software architecture, environmental sustainability, hyperspectral image, sorting materials

1 Introduction and motivation

Currently, there is a growing concern in different productive sectors about environmental sustainability, which is driving the change towards what has been called the circular economy. At the heart of this transformation is the problem of the waste accumulation in the environment of many types of materials such as plastic, textile, etc., posing as a challenge the demand for innovative solutions for effective waste recovery and reuse.

Waste separation and sorting is an essential step prior to its elimination, since waste such as plastics, paper, rubber, bottles, glass, etc. can be valorized by having an added value through recycling. Different techniques and technologies are used for waste separation and sorting such as magnetic separation, air separation, screening systems, X-ray, induction sorting system, colour sorting technology, thermal imaging, vision-based classification methods, etc.

Recently, classification methods based on digital images of municipal waste, as the ones presented in [1–7], use the images of the objects to perform their classification by means of some specific algorithm. However, for an effective recycling, it is necessary to separate the waste according to its chemical composition, so that the different sets of waste are as pure as possible. When litter of different chemical composition has similar morphological and visual characteristics, algorithms based on visual digital images are unable to distinguish between these different materials. Among the image based algorithms used to discriminate products, the most advanced ones are those based on artificial intelligence (AI) models. These AI models must be trained for a specific set of products, but any change in the format, texture, colour, or shape of the waste objects may require a new training process. Besides, if this training process is based on digital images in the visible spectrum, AI based methods will also not be able to distinguish between similar objects with different chemical compositions.

Sorting waste according to its chemical composition requires more information than traditional vision cameras can provide. However, thanks to the advances in multispectral and hyperspectral technology, today it is possible to integrate these sensors into industrial production lines like waste classification and sorting processing lines.

A multispectral or hyperspectral image (HSI) can be considered as a huge three dimensional data cube (usually called hypercube) with two spatial dimensions (width and height), and one spectral dimension (depth). For each value in the spectral dimension (wavelength), we have a 2D image of the scene captured by the spectral sensor. This image corresponds to the spectral information obtained only at the corresponding wavelength. A conventional video image (visible spectrum) follows the same data structure, but the depth is fixed to 3, which corresponds to the red, green and blue (RGB) channels, while a multispectral or hyperspectral sensor generates hypercubes with a depth of several tens or hundreds channels, respectively.

Given that our objective is the design of a hardware and software infrastructure for the implementation of an industrial line for the separation of waste based on its chemical composition, we decided to work with hyperspectral sensors, which produce a huge amount of information that needs to be transmitted, pre-processed and analysed. In this work we will focus on the design of the industrial system in general and in the computational system in particular allowing to perform all these tasks in real time.

This work is not focused on the specific processing algorithms to detect different chemical compositions based on the collected hyperspectral information, but rather on the infrastructure design that allows us the maximum possible computation time to execute high complexity detection algorithms in real time, i.e the time requirements that makes the industrial system economically profitable.

The physical conditions of the acquisition system (conveyor belt speeds, lighting system, environmental characteristics, etc.) as well as the type of sensor used (spatial and spectral resolution and capturing frequency) affect to the quality of the collected data. Depending on the quality of the data, a more or less sophisticated pre-processing step must be included before the classification algorithm. Therefore, a good industrial design will produce data with less noise, allowing a simple pre-processing step which will permit the use of more complex and accurate classification algorithms.

Building a modular system or software which allows the modification or incorporation of different pre-processing, detection or sorting algorithms with the least possible impact on the final system is an important challenge. The idea is to obtain a modular system design that can be used in several recycling plants, being able to adapt to the different physical parameters of the industrial plant configuration. The computer system must manage the classification algorithms as well as the hyperspectral sensor, the conveyors, the lighting systems, the operation and alarm sensors, the robotic picking arms, etc.

The present work is mainly focused on the analysis of the feasibility of a robotic system for recycling, proposing a modular software system. This software uses the computer system efficiently to apply different classification and segmentation algorithms, introduced as modules, to treat different materials together. The design of the software is therefore independent of the nature of the materials to be classified and sorted and also of the way in which the pick and place tasks are carried out, either by clamping, blowing or any other technology.

One of the goals is to minimize the cost of the computational infrastructure. We will use a conventional multi-core architecture on which we will distribute the heterogeneous work (sensor data acquisition, classification, segmentation, communication with other systems, etc.) and we will dimension it so that there are sufficient computational resources to apply acceleration techniques in the processes that require them. The sorting algorithm itself is beyond the scope of this work. However, the company JOVISA S.L. asked us for a system to classify plastic and cardboard. Due to the nature of the spectra of both materials, we propose an analytical solution with a low computational cost and an accuracy close to 100%. Although the classification of other types of materials may require more complex classification algorithms, the proposed software architecture has been designed in such a way that there is sufficient time to perform these more complex algorithms. Thanks to this software architecture design, the hardware requirements of the system will be able to adapt to the computational demands of the sorting algorithm.

The rest of the paper is structured as follows: Section 2 provides an introduction to hyperspectral image processing. Section 3 shows, firstly, the main characteristics of the industrial prototype used from a hardware point of view (Section 3.1), and secondly, Section 3.2 shows the developed software architecture that allows the system to be versatile and scalable from a software and hardware point of view. Finally, in Section 4, we present some numerical results and draw some conclusions.

2 Introduction to the hyperspectral imaging

Hyperspectral or HSI technology has been widely used in remote sensing [8] [9]. In particular, there are several satellites that provide hyperspectral images for Earth observation, such as SPOT 6, RapidEye, Pleiade, Sentinel 2, or Landsat 8. These sensors provide multispectral information of the Earth's surface with low spatial resolution (10 meters for Sentinel 2) and very low imaging frequency (5 days in Sentinel 2 for the same location). Sensors of this type are also installed in aircraft or drones to inspect sensitive areas.

Currently, there are hyperspectral sensors designed for industrial applications that do not differ in their basic operation, but are intended for their use at short distances and for the analysis at a higher spatial resolution (millimetres rather than meters).

These hyperspectral cameras are characterized by their spectral resolution, i.e. the number of wavelength bands per pixel, and their maximum operating frequency or frames per second (fps). In addition, these cameras are line cameras, i.e. their spatial resolution is Npx x 1, where Npx is the number of pixels per line provided by the camera. The pixel size width will depend on the spatial resolution of the hyperspectral camera, the optical lens angle, and the distance from the camera to the objects to be scanned. Once these parameters have been fixed, the capturing frequency must be adapted so that we get a squared pixel.

The choice of hyperspectral camera model depends on the size of the materials to be classified, which determines the camera pixel resolution, and the material composition, which determines the minimum spectral resolution.

3 Proposed system

After confirming the feasibility of the classification of different materials using HSI techniques under laboratory conditions, we detected some obstacles that hinder the implementation of these systems in a real industrial environment. We saw after an initial analysis of the computing times of the processes involved in the industrial classification and recycling system, that the laboratory prototype was not viable as it was planned. The huge amount of data received from the hyperspectral sensor needs to be pre-processed and then analysed according to the final system requirements, and thus, a proper industrial design will significantly reduce the computational cost of pre-processing associated with noise removal.

In this section, we will first describe the industrial system from a hardware point of view and then we will analyse it from a software point of view in order to identify the main functionalities and the main challenges to be addressed. Subsequently, the developed multi-level parallel software architecture will be described in detail in Section 3.2, showing the feasibility of the system.

The aim of our proposal is not to develop a closed system for sorting a limited number of materials, but on the contrary, our development should be able to run as a stand-alone system as well as being adaptable to existing processing lines. Our proposal is a scalable system, i.e. it can be adapted to processing lines with different workloads, being the main objective that the multi-level parallel software does not require relevant adaptations and/or modifications. Obviously, depending on the number and type of materials to be sorted, the software module in charge of sorting will have to be modified.

3.1 Hardware system

The hardware system of the developed prototype production line (Figure 1) consists in a first approximation of four basic parts:

- System for transporting the material to be sorted.
- HSI acquisition system.
- Material retrieval and sorting system.
- Computer system, including communication networks.

The system that collects and transports the material to be sorted, is essentially made up of the conveyor belts needed to transport the material to be sorted to the HSI acquisition system. As in any industrial system with these characteristics, these conveyors must be managed by intelligent systems with communication capabilities, so that they can be remotely controlled. Programmable Logic Controllers (PLCs) with TCP/IP communication capabilities have been used in all systems that require centralized control, such as conveyor belts, lighting system and refrigeration system, among others

Obviously, the goal is to speed-up the conveyor belt as much as possible while maintaining an accurate classification of the different materials as well as allowing an effective pick and place process. However, this speed is limited not only by the

Figure 1: Prototype production line.

technology or characteristics of the conveyor, but also by the characteristics of the material, and by the type of the robotic system used for the pick and place process, depending on whether it is a collaborative environment or not. In addition, it must be taken into account that if the speed is too low, the system will not be economically viable, and if the speed is too high, depending on the morphology of the objects, they may move on the belt due to air friction, causing errors in the robotic arms responsible for picking them.

The HSI acquisition system is a key part of the proposed system, both at the level of resolution and spectral characteristics of the multispectral camera used, but also, and not least, with respect to the design of the physical structure of the camera installation, lighting system, and thermal conditioning. As previously said, the resolution of the camera together with the mounting distance, affects the physical equivalent size of each pixel captured, whereas the required spectral resolution is determined by the chemical characteristics of the different materials to be classified.

As mentioned above, multispectral cameras are line cameras, not frame cameras. This means that they do not provide images in two dimensions, but rather spectral information, usually several hundred frequencies for a single line of pixels. Therefore, the physical dimension corresponding to a pixel depends on the distance of the camera installation with respect to the conveyor that transports the material, and on the frequency operation of the camera that must be adjusted with respect to the speed of the conveyor belt. In order to obtain realistic dimensions of the shape of the objects, it is necessary to work with square pixels, which requires a fine tuning between the camera acquisition speed, the conveyor speed, and the camera installation height. Therefore, a compromise has to be found between the conveyor speed and the multispectral camera installation height, taking into account the spatial resolution of the multispectral sensor.

Although these multispectral cameras have been designed for industrial environments, they are not immune to both radiated and conducted noise sources. As mentioned above, one of the objectives is to have systems that can operate within the strict time requirements. These requirements are given by the industrial infrastructure (conveyor belt speed, conveyor belt width, number of picking and sorting robots, robot operating speed,etc.), but also by the computational cost of the different processes (pre-process, classification and segmentation) to be executed. The more immune the system is to noise sources, the less pre-processing or conditioning is required for the signal acquired by the multispectral camera, thus helping to meet the timing requirements of the overall system.

Multispectral cameras capture the electromagnetic signal reflected from an object in a broad frequency spectrum, just as conventional vision cameras reflect the electromagnetic signal in the visible spectrum. Logically, in order to obtain this reflection signal, first it must be emitted toward the object. In order to design a robust and low-cost system, we propose the use of low-cost emitting sources, being halogen emitters that emit both in the visible spectrum and in the non-visible spectra close to the visible spectrum, such as the near infrared (NIR), our choice.

One option to avoid radiated noise sources, such as sunlight, other light sources in the industrial environment, or radiated electromagnetic noise, would be to encapsulate the system in a dark chamber where no electromagnetic signal can enter, a solution adopted in other systems, such as PICVISA's commercial product with reference Ecopick[1]. However, this solution prevents us from achieving several of our objectives, such as having a scalable, versatile and low-cost system. Our proposal requires only that there be no direct sunlight, which is easily achievable; the harmful effects of the rest of the noise are treated computationally.

Conducted noise sources are all sources of vibration that can affect the operation of the multispectral camera. These cameras are not immune to vibration, in the case of the model used in this project, one of the effects of vibration is the loss of information. If this loss of information is sporadic, it is easily solved by computational means, but if the loss is more intensive, it causes unwanted computational overhead and possible errors in objects detection. Therefore, a camera anchoring system has been designed that must first be decoupled (with as little physical contact as possible) from the sources of vibration, i.e. conveyor belts and robotic systems. To minimise this noise, the camera was installed with vibration mounts. These mounts isolate the camera from the vibrations of the rest of the system. In addition, the camera requires a thermal conditioning system through fans installed in such a way that they do not

[1] `https://picvisa.com/ecopick-robot-inteligencia-artificial\-clasificacion-materiales-residuos/`

transmit vibrations to the hyperspectral camera.

The material recovery and sorting system is the most conventional part of the system, it does not require any specifications different from the conventional systems based on conveyors controlled by PLC and robotic arms, the robots use suction to pick up parts in our prototype. The only aspect to take into account is that in order to homogenize the signal reflected by the belt, i.e. when there is no material, it is necessary to automatically clean the belt periodically.

Finally, as mentioned above, the computing system is a conventional multicore system, i.e. a shared memory parallel computing system. The communication systems are, on the one hand, an Ethernet network that will allow connectivity through TCP/IP sockets with the PLCs and with the robotic arms, and, on the other hand, the communication with the hyperspectral camera requires a higher bandwidth communication, in particular a Camera Link[2] protocol that requires a dedicated frame grabber.

3.2 Application description

In this section we describe the application structure design as well as the optimizations performed to accomplish with the real time requirements of a real system designed for a waste-recycle company. The waste-recycle prototype, see Figure 1 is composed by a conveyor belt with a 80cm width size. The hyperspectral camera used is the FX17[3] model from Specim whose wavelength range goes from 900 to 1700 nm which is able to capture 640 pixels x 224 wavelength bands at 670 frames per second (fps). The camera height as well as the camera operating frequency have been set to obtain a squared pixel of 1.08 mm. The system is also composed by one Robot RS007L[4] from Kawasaki Robotics Inc. For testing purposes, the developed system has been configured to classify only two main materials: plastic and cardboard, that will be picked by the robot and stored in two different hoppers. Besides, there is a PLC that controls the machine conveyor belt, lighting, robot operation signals and industrial robot vacuums system. The computer used to perform the capture, pre-preprocess, classification and segmentation of the different material objects is a 2 CPU based Intel(R) Core(TM) i7-6800K CPU at 3.4GHz with 6 cores each one and with 64Gb RAM running Windows 10 operating system.

3.2.1 Software architecture

First of all, we decided to use QT[5], a cross-platform development framework, because it is multiplatform (Windows, Linux, MacOS,...). Moreover, since the software will be based on threads to distribute the heterogeneous work between the different available

[2] `https://www.automate.org/vision/vision-standards/vision-standards-camera-link`

[3] `https://www.specim.com/products/specim-fx17/`

[4] `https://kawasakirobotics.com/products-robots/rs007l/`

[5] `https://www.qt.io/`

cores, QT has a communication system based on signals and slots for the intercommunication of the different threads. Finally, it will allow the development of a complete graphical user interface (GUI).

In Figure 2 we show the different processes/threads involved in the application. As can be seen, the application is mainly composed by four threads, namely *main*, *camera*, *segment* and *sender*. The *main* thread is in charge of creating all the threads and global objects as well as the management of the GUI. Currently, the GUI only shows the raw data received from the hyperspectral camera, the classification data (plastic, cardboard and belt), and the segmentation results (object detection), see Figure 3. When QT Application object is created and launched, it sends a *startCamera* signal to the camera thread that starts capturing lines from the spectral camera. Each camera line is then pre-processed and classified according to the different materials. In order to perform the segmentation process, each classified line is stored in a storage buffer or window. By default, the storage buffer or window has a length of 1600 lines, although this length can be determined by a configuration parameter. After the first window is completely classified, a *startSegmentation* signal is sent to the *segment* thread, that performs the segmentation of the different objects in the image for each material. Afterwards, every 400 new classified lines a new segmentation process is performed. From these segmentations, we obtain the different material objects centroids (mass object centre). Then we have to determine which of these centroids corresponds to new detected objects or if they were detected in previous segmentations and should not be included again. There is a global object that contains all the detected centroids. Finally, the physical robots request to the sender thread for objects that are near to its current position to pick them. Besides, the *segment* thread communicates to the *main* thread though a QT communicator object to display the segmented image in the graphical interface.

It would not be possible to perform the work of these four threads sequentially and meet the time requirements initially set by the frequency operation of the hyperspectral camera. The creation of these threads results in a first level of parallelisation that distributes the heterogeneous work. If the work assigned to any of the threads does not meet the time requirements, this work must be accelerated by using more processing elements (cores).

Apart from the above-mentioned threads, shown in Figure 2, a Robot Operating System[6] (ROS) thread is used to communicate with the Programmable Logic Controller (PLC) that controls the machine conveyor belt, camera lights, robot operation signals and industrial robot vacuums system. In the prototype tested, it was necessary to install a single PLC, but the QT application must generate a thread for each installed PLC with which it needs to communicate.

[6] `https://www.ros.org/`

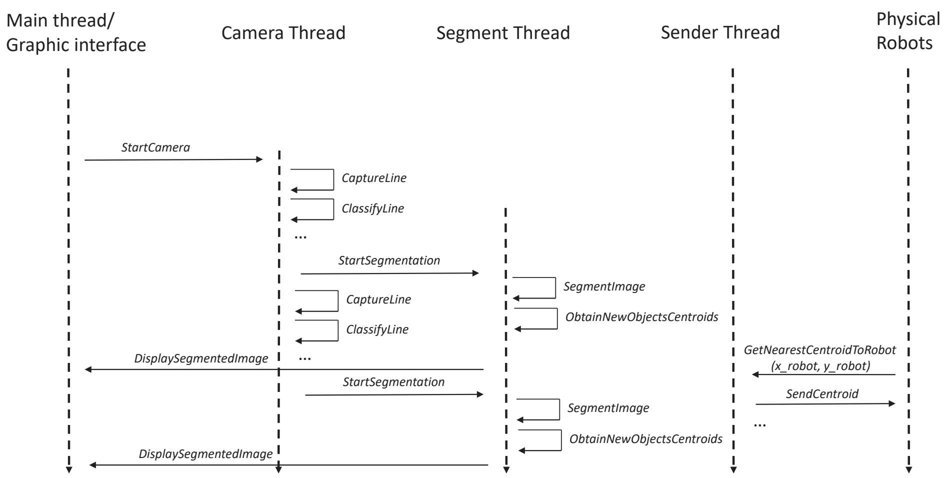


Figure 2: Application thread architecture.

3.2.2 Hyperspectal detection and objects segmentation

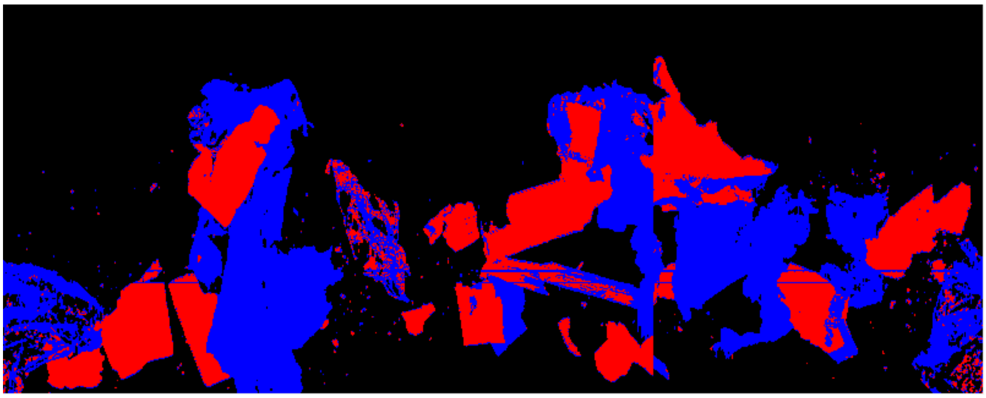
As mentioned above, the waste recycling classification machine developed for the company's test purposes only requires the classification of plastic and cardboard materials. Therefore, the classification is performed using a threshold value using the minimum value of the normalized spectrum of each pixel. Remark, that first we have to determine if the pixel corresponds to an object pixel or if belongs to the conveyor belt. In order to determine if a pixel belongs to the conveyor belt we use the mean of the raw pixel spectrum values.

Although both the pre-processing and the classification algorithms performed in this example machine are quite simple, we want to remark that our application architecture has been developed in a modular way so that both algorithms can be replaced easily. This classification procedure was used to test that the system works in real time, and that our architecture design allow the acceleration of these algorithms using parallelism techniques.

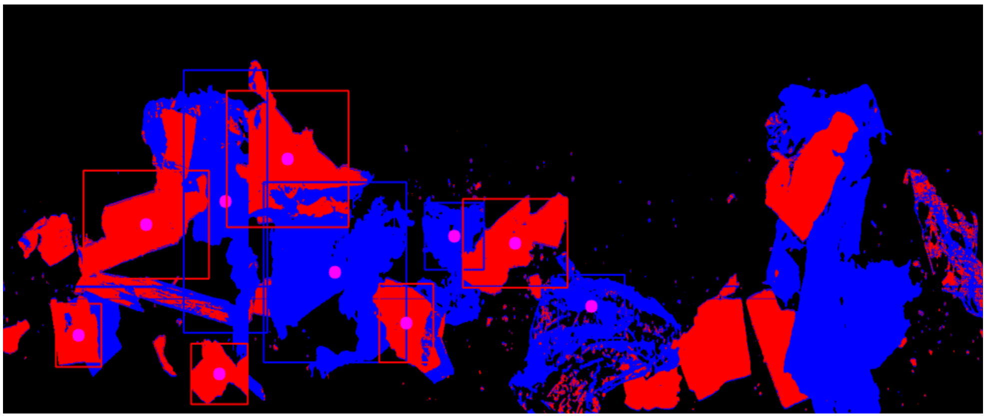
In particular, in order to speed up the classification process, the OpenMP[7] paradigm has been used. The intrinsic parallelism associated with the classification of each pixel has been exploited by using dynamic work sharing as a heterogeneous cost computing. Taking into account the number of threads running in the application shown in Figure 2, the maximum number of additional cores available is equal to the number of cores available in the computer minus the number of threads running (5 in our system). If two additional cores are used, the sorting work is divided between three cores (the two additional cores and the one assigned to the *camera* thread), so the minimum number of cores required for the platform is 7 in this case. The complexity of the classification software module determines the number of additional cores required. We want to remark that both the pre-processing and the classification process times are

[7] `https://www.openmp.org/`

(a) Raw data



(b) Classification



(c) Segmentation

Figure 3: Graphical information.

critical, because the camera thread can capture one line image every 1492 μs taking into account the maximum capturing frequency of the camera (670fps).

Regarding the segmentation process, after the classification of the first complete window (1600 lines) that corresponds with approximately 1.7 meters of the conveyor belt in our prototype machine, a segmentation process using OpenCV 4.8.1[8] is performed for each possible material. After that, we obtain a set of blobs that corresponds to the different material objects detected in the image that must be picked by the different robots (one robot in the prototype machine). These blobs have an area and a centre position (x, y) or centroid. Before adding a blob to the list of blobs pending to be sent, the blobs that do not satisfy a minimum area (determined by the robot sucker size) are discarded.

After the completion of the first complete image (1600 lines), a new segmentation is performed every 400 lines (configurable value), so as to see a smoother object movement transition in the graphical interface and to keep visual tracking of the objects. This cute visual effect complicates the management of the blob list, because three extra segmentations are performed for each image window and there will be previous detected objects in the list that are in a new position in the current segmentation. To overcome this issue, before performing a new segmentation, the list of object is updated in such a manner that its x centroid value is increased 400. In this way, before adding new segmented blobs we can check if they were detected in previous segmentations. Furthermore, the blobs are not deleted from the list when they are sent to robots, instead, we put a sent mark to avoid the inclusion of blobs detected in the last segmentation that were just sent to robots. Finally, the blobs will be deleted from the list if their x position exceeds the maximum reachable position of the furthest robot.

Although the prototype machine just work with two materials, the segmentation process has been implemented in a modular way and can be configured to work with multiple materials through the software application parameters.

3.2.3 Robots communication subsystem

As shown in Figure 2, the application architecture contains a *sender* thread which is in charge of the communication between the different possible robots and the application. Although the prototype system developed has just one robot, the application architecture and communication system can be configured to work with several robots.

First of all, when the *sender* thread is created and launched, it establishes a socket communication between the computing platform and the different active robots. The number of robots as well as if they are active or not is determined by the application configuration parameters. Besides, each robot can be configured to pick just one type of material or to pick several ones, and also it is possible to assign different hoppers to the materials. Afterward, the computing platform starts receiving packets from the different robots. There are three different types of command packets: BUSY, ACK, and NEW. The BUSY command is sent by a robot when it is moving with a picked

[8] `https://opencv.org/`

object and it can not accept new positions. In this case, the computing platform sends and ACK response. The ACK command is sent by a robot to maintain the socket opened to avoid timeout, and the computing platform also responds ACK. When a NEW command is sent by a robot, in the communication packet also is sent the robot current x and y position. In this case, the computing platform looks for an object in the list which corresponds to the material that the robot is configured to pick with the centroid nearest to the current position of the robot in order to speed-up the picking process. The computing platform compounds a packet with the selected object information including, centroid position, material, and hopper number.

4 Results and conclusions

In this section we will provide the computing time results measured in the prototype developed system. The pre-process and classification of a captured camera line (640pixels x 224 band values) last on average $0.173ms$, which is equivalent to a capturing frequency of 5767 fps. Remember that the FX17 camera has a maximum frequency of 670fps. That means that we have room enough (8.6 times) to develop a more sophisticated classification algorithm. Regarding the segmentation process, in the real developed system, we require $20.314ms$ to perform a segmentation of a window of 1600 lines, which corresponds to a maximum frequency of 19690fps.

In the implemented prototype, the camera has a field of view (FOV) of $38°$ and it has been located at a height with respect to the belt conveyor of $1004mm$, which provides a pixel size of $1.08mm$. In order to get a square pixel, the conveyor belt feed must match that pixel size for each captured line, so the PLC triggers the camera every $1.49ms$. Because of the characteristics of the conveyor and the resolution of its encoder, the real developed system can reach a maximum conveyor belt speed of $43.38m/min$. The conveyor belt speed can be increased while maintaining the maximum frame rate of the camera by increasing the pixel size. This will depend on the size of the objects to be processed. Therefore, the pick & place robotic system must be adapted to this parameters.

In this work, we have presented a modular software system applied to a prototype recycling plant. This recycling plant will classify waste according to its chemical composition using HSI techniques in order to obtain higher quality recycled raw materials. This work has demonstrated that the project is technically and economically feasible. A study was also carried out to determine the minimum number of cores required for the parallel computing platform.

Acknowledgements

This research was partially supported by the Spanish Ministry of Science and Innovation and the Research State Agency under Grant PID2021-123627OB-C55 co-financed by FEDER funds (MCIN/AEI/FEDER/UE) as well as by grant CPP2021-

008593, funded by MCIN/AEI/10.13039/501100011033 and by the European Union-NextGenerationEU/PRTR and by Valencian Ministry of Innovation, Universities, Science and Digital Society (Generalitat Valenciana) under Grant CIAICO/2021/278. We also want to thank the company JOVISA S.L for providing its facilities and its industrial prototype for testing our system.

References

[1] A. D. M. Africa, N. P. J. Labay, M. S. B. Ong, J. A. J. Sales, M. M. E. Toyoda, Vision-based algorithm for recyclable waste classification, ARPN Journal of Engineering and Applied Sciences 14 (13) (2019) 2459.

[2] R. Aarthi, G. Rishma, A vision based approach to localize waste objects and geometric features exaction for robotic manipulation, Procedia Computer Science 218 (2023) 1342–1352.

[3] S. Yeaminul Islam, M. G. R. Alam, Computer vision-based waste detection and classification for garbage management and recycling, in: The Fourth Industrial Revolution and Beyond: Select Proceedings of IC4IR+, Springer, 2023, pp. 389–411.

[4] M. Malik, S. Sharma, M. Uddin, C.-L. Chen, C.-M. Wu, P. Soni, S. Chaudhary, Waste classification for sustainable development using image recognition with deep learning neural network models, Sustainability 14 (12) (2022) 7222.

[5] S. Poudel, P. Poudyal, Classification of waste materials using cnn based on transfer learning, in: Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation, 2022, pp. 29–33.

[6] M. Koskinopoulou, F. Raptopoulos, G. D. Papadopoulos, N. Mavrakis, M. Maniadakis, Robotic waste sorting technology: Toward a vision-based categorization system for the industrial robotic separation of recyclable waste, IEEE Robotics & Automation Magazine 28 (2021) 50–60. https://doi.org/10.1109/MRA.2021.3066040 doi:10.1109/MRA.2021.3066040.

[7] V. Ruiz, Á. Sánchez, J. F. Vélez, B. Raducanu, Automatic image-based waste classification, in: From Bioinspired Systems and Biomedical Applications to Machine Learning: 8th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2019, Almería, Spain, June 3–7, 2019, Proceedings, Part II 8, Springer, 2019, pp. 422–431.

[8] A. Goetz, G. Vane, J. Solomon, B. Rock, Imaging spectrometry for earth remote sensing, Science (New York, N.Y.) 228 (1985) 1147–53. https://doi.org/10.1126/science.228.4704.1147 doi:10.1126/science.228.4704.1147.

[9] M. T. Eismann, Hyperspectral Remote Sensing, Vol. PM210, SPIE Press Book, 2012.