Computers and Electronics in Agriculture 171 (2020) 105304

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Deep learning in olive pitting machines by computer vision

Manuel de Jódar Lázaro[a,*], Antonio Madueño Luna[b], Alberto Lucas Pascual[a], José Miguel Molina Martínez[c], Antonio Ruiz Canales[d], José Miguel Madueño Luna[e], Meritxel Justicia Segovia[f], Montserrat Baena Sánchez[g]

[a] Ph.D. student at the Technical University of Cartagena, Spain
[b] Aerospace Engineering and Fluid Mechanical Department (University of Seville), Spain
[c] Food Engineering Department (Technical University of Cartagena), Spain
[d] Engineering Department (Miguel Hernández University of Elche), Spain
[e] Graphics Engineering Department (University of Seville), Spain
[f] Ph D. Agricultural Engineer at Agriculture Service DP Jaén, Spain
[g] Agency of Agrarian and Fishing Management of Andalusia (A.G.A.P.A), Spain

ARTICLE INFO

Keywords:
Artificial neural networks (ANNs)
Olive pitting machines
Table olives
Physical neural network
NeuroMem CM1K

ABSTRACT

Olive pitting machines are characterized by the fact that their optimal functioning is based on an appropriate adjustment: selection of a feed plate adapted to the olive variety and its caliber, geometrical characteristics of the feed chain, etc. The first of these elements sets the optimal way for olives to enter the feed chain and, therefore, it prevents empty pockets or more than one olive to be placed in the same pocket. The second element sets the appropriate position for the olive to be pitted and prevents it to be pitted by a secondary axis.

The proposed study analyzes the appropriate placement of olives in the pockets of the feed chain by using the following items:

1. A computer vision system with an external trigger, which is capable of taking a picture of every pocket passing in front of the camera.
2. A classifying neural network that, appropriately trained, differentiates between four possible pocket cases: empty, normal, incorrectly de-stoned olive in any of its angles (when the olive is de-stoned transversally instead of longitudinally, also known as “boat”) and anomalous case (two olives in the same pocket, small parts of it or foreign elements, such as small branches or stones).

A preliminary analysis, carried out with the MATLAB Neural Network Toolbox, has enabled to test the viability of using a neural network to perform this type of classification.

The main objective of this paper is to illustrate the use of a physical chip with neural networks, NeuroMem CM1K (General Visions, 2016. CM1K), for sorting purposes.

Therefore, it is necessary to identify the minimum resolution required to classify the images of olives in olive pitting machines and their adequate position to be pitted considering an input vector of up to 256 bytes, which is the maximum dimension supported by NeuroMem CM1K.

As described before, a camera with an external trigger will be used for image capturing synchronized with the feed chain.

Given that the image classification speed must be higher than 15 Hz to be operatively convenient, the industrial feasibility of this system will be assessed in order to implement it in an olive pitting machine, the operating speed of which starts at a rate of 900 olives/minute.

The use of the physical chip NeuroMem CM1K, for its greater capacity and scalability, has been proven satisfactory and, therefore, it offers a great potential for sorting purposes. As stated in the obtained results contained in following pages, it has been possible to train, for the first time, an artificial neural network (ANN) implemented in a neuromorphic chip to classify the images of the olives in the feed chain of olive pitting machines. Consequently, it sets an alternative system in order to study possible cost, space and energy use reductions in contrast with traditional common computer systems or PLCs.

* Corresponding author.
E-mail address: manueldejodar@gmail.com (M. de Jódar Lázaro).

https://doi.org/10.1016/j.compag.2020.105304
Received 4 June 2019; Received in revised form 10 September 2019; Accepted 22 February 2020

1. Introduction

Neural network is a widely used technology in the industry due to its successful way to classify or improve the production. Some examples are described below.

Parallel Neural Network Chips are used for fish inspection before filleting offshore (Menendez and Paillet, 2008). Each network chip system uses four neural network chips (accounting for 312 neurons) based on a natively parallel, hard-wired architecture that performs real-time learning and nonlinear classification (RBF).

Common tasks are related to fill bottles in factories. The process is quite simple. However, it may be necessary to use an intelligent device to inspect the process (Menendez and Paillet, 2013).

There are other similar papers related to fruits and vegetables:

- Classification of apples using three layers of 9-6-3 neurons, 96.6% accuracy (Yang, 1993).
- (Nagata and Qixin, 1998) developed a grading system for fruit and vegetables using neural network technologies, obtaining a high percentage of accuracy for strawberries and green peppers (94% to 98% and 89%, respectively).
- Applied machine vision and ANN for modelling and controlling grape drying process. This paper presents a new method for predictive modelling of grape drying process for on-line monitoring and controlling of this process. (Behroozi Khazaei et al., 2013).
- Olive Fruits Recognition Using Neural Networks. Olive fruit recognition is performed by analyzing RGB images taken from olive trees. (Gatica et al., 2013).
- Identifying Olive (Olea europaea) Cultivars Using Artificial Neural Networks. Backpropagation neural networks (BPNNs) were used to distinguish among 10 olive (Olea europaea L.) cultivars, originating throughout the Mediterranean basin. (Mancuso and Nicese, 1999).

The combination of computer vision and neural networks is a way to perform tasks that are more complex:

- A neural network chip is used for license plate recognition (Liu et al., 2011). The chip combines a video image-processing module with a neural network module by using equalized image processing algorithms and network classification algorithms.
- Image recognition is simpler than image processing methods for face recognition, mainly due to the lack of a fixed pattern for comparison purposes (Sardar et al., 2011). Santu Sardar published a paper based on automated face recognition, which is a technique employed in a wide range of practical applications, including personnel access control or identification systems.

Regarding table olives, there are some techniques to classify them by using computer vision and neural network according to the papers below.

The book *Computer Vision Technology for Food Quality Evaluation*, 2nd Edition, (Sun, 2008) includes a specific chapter based on different technologies that analyze the quality of food. Chapters 11 to 13 deal with quality evaluations of apples, citrus fruits and strawberries, respectively.

Of particular interest to the problem, chapter 14 explains the classification and evaluation of table olives and describes how to classify them by color, shape or external defects made by insects.

A usual way to classify the olives is by using computer vision (Diaz, 2004). The paper analyzes the images captured by a camera connected to a PC. The image analyzed 66 olives per matrix. In this paper, the author used a Bayesian math model for pre-classification to perform this process. A neural network software was used with 15 sorting parameters and a hidden layer. The result was successful: the network was able to classify more than four types of olives. The results could be improved by using high-resolution images.

Other chipsets have been successfully used in several trials obtaining positive results:

Lohi is a chip manufactured by Intel. It integrates programmable synaptic learning rules. This chip can solve LASSO optimization problems. (Davies et al., 2018) presented an unambiguous example of spike-based computation using a range of different neuron models and some preliminary Lohi results.

Moran et al. (2018) presented a new work demonstrating, for the first time, spinal image segmentation using a deep learning network. It was implemented on a neuromorphic chip, an IBM TrueNorth Neurosynaptic System. The results compared to human-generated segmentations of spinal vertebrae and disks.

Moradi et al. (2018) presented a work which established a novel routing methodology to minimize memory requirements and latency, maximizing programming flexibility. The authors validated the proposed scheme in a multi-core neuromorphic processor chip prototype, DYNAP-SE.

Frenkel et al. (2019) presented a digital spiking neuromorphic processor. This chip obtains a minimum energy per synaptic operation. They demonstrated an efficient implementation of the spike-driven synaptic plasticity learning rule for high-density embedded online learning.

Google's Coral Dev (Fried, 2019) is a single-board computer based on the Raspberry Pi form factor, designed to run TensorFlow Lite models. The chip Edge TPU is implemented on this board and it is capable of up to 4 trillion operations per second (TOPS).

Lobachev et al. (2018) used the physical deployment model employed in the Intel Movidius VPU modules and Raspberry Pi microcontrollers to investigate and model the neural network integration. This network model utilizes a lower power approach and faster responses as well as it improves overall system efficiency.

Hubbard (2019) selected the Nvidia Jetson platform, specifically the Nvidia Jetson Nano, to build a neural network for detecting objects. It requires a very high-speed device with the machine learning runtime.

At the time of writing this paper, the chip selected was the NeuroMem CM1K. In 2019, the manufacturer Nepes IA in collaboration with General Vision launched a new powerful chip with 575 Neurons (NM500).

Kim (2019) proposed a system to control vehicle speed by recognizing the traffic information images marked on road with an Advanced Driver Assistance System, ADAS. This application used this newly developed neuromorphic artificial intelligent chip NM500.

The following article (General Visions, 2018 CogniPat SDK for MATLAB) proves the benefits of this chip under the MATLAB platform to learn and recognize extracted data vectors, signals and images.

There is another platform on the market to develop research projects: the NeuroMem USB Dongle (General Visions, 2019 NeuroMem USB Dongle) uses 4 NM500 chips packaged, accounting for 2304 neurons in total.

2. Materials and methods

2.1. Neural network

A technology based on ANNs has been used due to its noticeable ability to obtain complicated and imprecise data, such as image analysis. Therefore, it can be used to deduce patterns and detect correlations which are hard to be seen by humans or other computational techniques.

The architecture of a neural network is composed of multiple elementary processors (Goodfellow et al., 2016). It is an adaptive system with an algorithm which can adjust its weight values in order to meet the performance requirements of the problem based on representative samples.

Therefore, it is possible to state that an ANN is a distributed computer system with the following features:

– It is composed of an input layer, an output layer and one or more hidden layers, which can account for hundreds or thousands of them in complex systems.
– Every unit is an independent perceptron, which has an extremely low processing rate.
– The units within the input layer allow the access to hidden layers and these do the same to access output layers in turn.
– Each node connecting different neurons has a weight value, w.

The backpropagation process, used for perceptron training purposes, consists on the following:

– A training sample propagates forwards, through the network.
– Input error calculation is performed, usually squared error. Both desired and output values are considered to obtain the aforementioned calculation.
– This error calculation is reduced using the stochastic gradient descent method (SGD).

The error optimal value is defined as the value obtained as a global minimum. The goal is to obtain it during the training stage. However, local minimum values can be obtained instead.

Backpropagation is the most efficient way to set this value (Hecht-Nielsen, 1992). First, errors are calculated in output units considering the difference between the desired and predetermined values. After that, they propagate through the network using the weights and hit the minimum in the most optimal way.

Such working structure is repeated in every network. Considering deep learning networks, the concept of hidden layer must be raised because this type of layer is based on the universal approximation theorem and, therefore, a neural network with a single hidden layer containing a finite number of neurons may be trained to approximate an arbitrary random function.

It is important to state that the most important feature of artificial neural networks lies in their learning capacity from a training pattern combination, meaning that they are able to find a model which fits the obtained data.

Fig. 1 shows the diagram of the physical neural network used by the CM1K chip.

Software emulation of a neural network is computationally demanding, which implies that operations will take a long time to be carried out. The use of a chip that physically implements a neural network (that is the case of this paper with CM1K chip) will speed up the training and response processes of the neural network.

2.2. The NeuroMem chip CM1K

The NeuroMem neural network shown in Fig. 1 is a pattern recognition accelerator chip which is also trainable in real-time by learning examples, (General Visions, TM_TestNeurons_SimpleScript). A NeuroMem chip is a fully parallel silicon neural network: it is a chain of identical elements (i.e. neurons) which can store and process information simultaneously. They are addressed in parallel and have their own "genetic" material to learn and recall patterns without running a single line of code and without reporting to any supervising unit.

Input data coming from a variety of sources can be converted into a pattern vector which can then be sent to the neurons for either learning or recognition, (General Visions, 2019 TM_NeuroMem_Technology_Reference_Guide). In the first case (learning), the neurons decide autonomously if the input pattern and its associated category (1) represent novelty and should be stored in the next available neuron, and (2) bring conflict to committed neurons which should be corrected. In the second case (recognition), the neurons decide autonomously which one has the closest match and queue their responses per decreasing level of confidence.

The CM1K (General Visions, 2017 TM_CM1K_Hardware_Manual) is composed of the following modules: a Neuron Interconnect module, a chain of neurons, daisy-chained and interconnected, a recognition stage (optional usage) and finally, a I2C slave (optional usage).

The CM1K has two possible classifiers: K-Nearest Neighbor (KNN) or Radial Basis Function (RBF), and more precisely, a Restricted Coulomb Energy (RCE) neural network (Halgamuge et al., 1994). We used the second one in this work.

In a CM1K (General Visions, DS_CM1K), a neuron is a memory with some associated logic to compare an incoming pattern with the reference pattern stored in its memory and react (i.e. fire) according to its

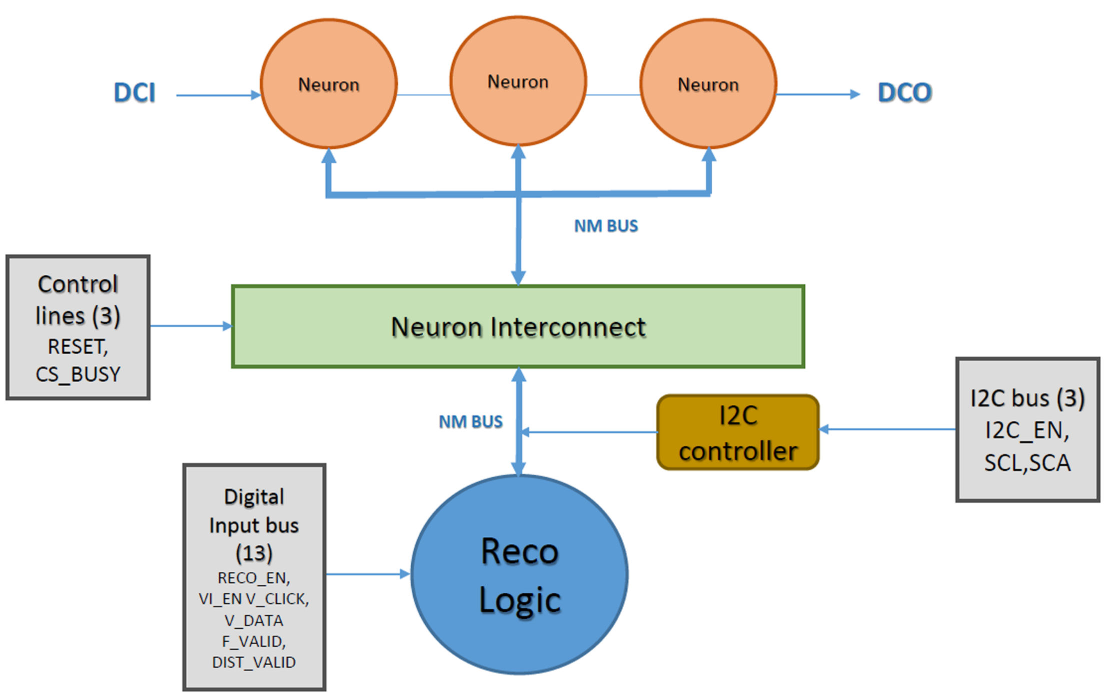

Fig. 1. Diagram of the CM1K chip neural network.

similarity range. A neuron also has a couple of attribute registers such as a context and category value. Once a pattern is broadcasted, the neurons communicate briefly with one another (for 16 clock cycles) to determine which one holds the closest match in its memory. The "Winner-Takes-All" neuron de-activates itself when its category is read, thus leaving the lead to the next "Winner-takes-All", if applicable, and so on. A single CM1K matching a pattern of 256 bytes against 1024 models delivers the equivalent of 192 GiGaOps per second.

2.3. MATLAB neural network toolbox

We are using the MATLAB Neural Network Toolbox (The MathWorks Inc., Neural-network 1994–2017) in order to train a deep neural network and classify images taken by a camera.

Once trained, the network will be able to differentiate the most common errors in the olive-pitting process, including "boat" de-stoning and empty pockets. This preliminary stage allows us to classify binarized images properly in order to test the most common error cases in physical chip.

A neural network with multiple hidden layers may be useful to solve sorting problems dealing with complex data, such as images. There are plenty of developed applications, e.g., the classification of wine cellars according to certain patterns (The MathWorks Inc., wine-classification-with-neural-pattern-recognition 1994–2017). Every layer can learn different characteristics in different levels of abstraction. However, training the neural network with multiple hidden layers may be challenging.

An effective way to train a neural network with multiple layers is to train one layer at once (The MathWorks Inc., train autoencoder 1994–2017), which can be achieved by using a specific network called 'autoencoder' in every selected hidden layer. This feature is enabled in the used (MATLAB, 2017a) and later versions. An autoencoder is a neural network in which the inputs are trained to emulate the outputs (Moller, 1993). Autoencoders use unsupervised learning (Haykin, 1999) which employs an algorithm that collects data in an unlabeled data set according to some hidden features of the data. Autoencoders are usually symmetrical, taking the middle layer as reference, and consist of two parts. The encoder comprises both the input and the middle layer whilst the decoder consists of the middle and the output layer. The autoencoder is divided into two neural networks that work as an encoder-decoder algorithm.

The neural network, the input of which is the encoder input and the output is the middle layer of the autoencoder, compresses the data. On the other hand, the neural network, the input of which is the middle layer and the output is the output layer of the autoencoder, decompresses them. The encoder also extracts a set of features.

The autoencoder is often used for data dimensionality reduction, features extraction or reinforcement learning.

The mathematical model to determine autoencoder (Pierre, 2012):

$$\min\ E(A, B) = \min_{A,B} \sum_{t=1}^{m} E(x_t) = \min_{A,B} \sum_{t=1}^{m} \Delta(AoB(x_t), x_t)$$

For any A ∈ Ą and B ∈ Ɓ, the autoencoder transforms an input vector $x \in F^n$ into an output vector $A \text{ o } B(x) \in F^n$.

- F is set.
- m is a positive integer.
- is a set of m (training) vectors in F^n.
- Δ is a distorsion function.

(The MathWorks Inc., Train Stacked Autoencoder for Image Classification 1994–2019) provides a clarifying example about this type of neural structure for image classification using an autoencoder neural network. In this example, handwritten number images of up to 28 pixels are included to be identified and classified by the MATLAB Neural Network Toolbox, using up to 5000 images for training purposes.

In our case, a two-autoencoder stacked neural network has been used to extract different image features and also a Softmax layer (Demuth et al., 2014) to determine different classes.

As shown in Fig. 2a diagram, the first trained autoencoder (Autoencoder 1) uses a hidden layer of 100 neurons that comprises a maximum input of 256 dimensions (16 × 16 pixels) into 100-dimensional feature vectors.

As shown in Fig. 2b, the second autoencoder (Autoencoder 2) is used to simplify and optimize the neural network comprising again the data, from 100 dimensions (which is the number of the dimensional feature vectors in Autoencoder 1) to an output of 50-dimensional feature vectors.

The Softmax layer, in Fig. 2c, takes those 50-dimensional feature vectors from Autoencoder 2 and assigns a class to every possible vector. In our case, there are three possible classes: empty pocket, "boat" olive and normal. Anomalous cases (two olives in the same pocket, broken olives, etc.) have not been included.

Finally, it is possible to go further and adjust it through a back-propagation process throughout the stacked multilayer network (deep network), including the encoder 1 of the Autoencoder 1, the encoder 2 of the Autoencoder 2 and the Softmax layer, as shown in Fig. 2d. This process readjusts the entire network and retrains it using the training data under supervision.

Once built and trained successfully, this type of neural network for deep learning (as mentioned before, there are two layers in our case) will allow us to analyze the effect of image resolution on the sorting capacity of the network.

2.3.1. Preliminary tests: minimum resolution

Several tests to identify the minimum resolution accepted by physical chip are carried out with the purpose of setting the lowest processing rate. To do so, real images are to be processed in MATLAB and it should be estimated if the system is able to identify them.

A test comprising 10,000 images and random sequences of 300

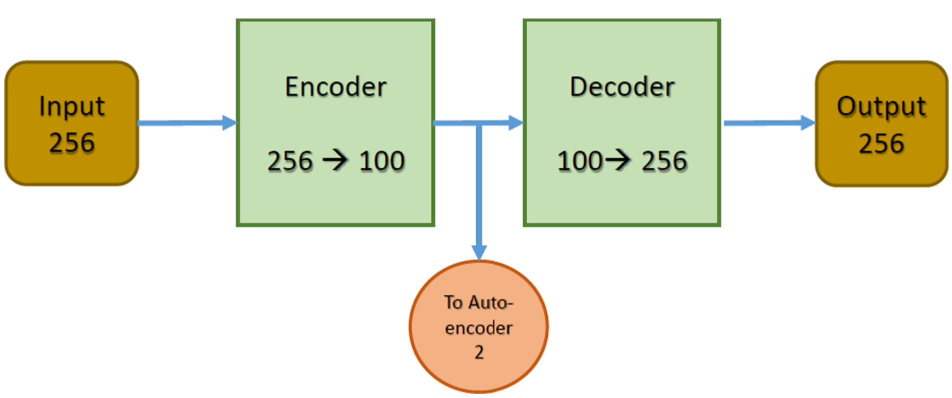

Fig. 2a. Autoencoder 1 diagram.

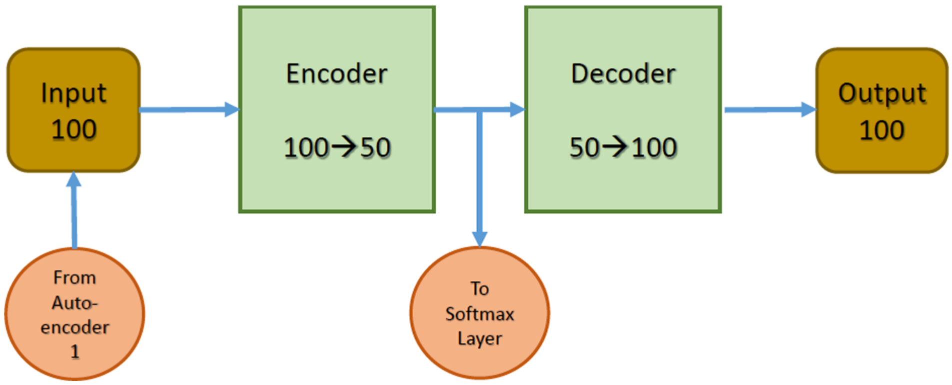

Fig. 2b. Autoencoder 2 diagram.

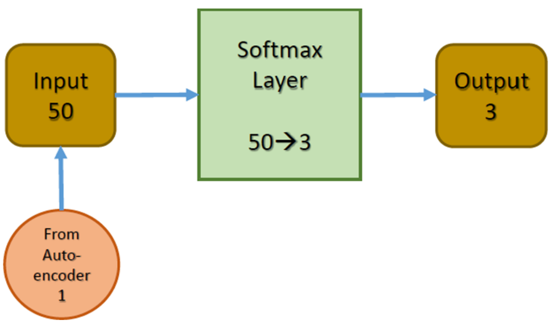

Fig. 2c. Softmax layer diagram.

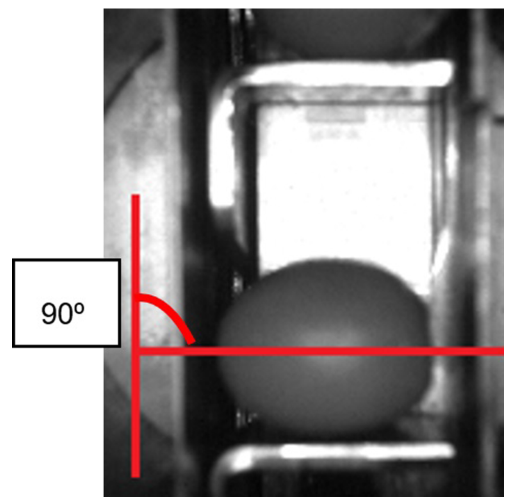

Fig. 3. Both the punch needle and the olive are perpendicular to the X axis.

images, in which the orientation of the olive has been calculated, has been set in order to validate the preliminary tests. Once tested in a given resolution, it is checked with the same images but at a lower resolution, reducing a pixel in every direction. Verifications conclude (as exposed in the section about results) that the minimum resolution accepted is 11 × 11. Below this resolution, the system is not able to identify the image.

Therefore, we will focus on the 16 × 16 pixels that should allow us to know the different options we will have when using this physical chip which are available in the market today: CM1K (General Visions, 2016 CM1K Chip) and 2 × NM500 (General Visions, 2019 NM500 chip).

2.3.2. Preliminary tests: Classification

First, it is necessary to set with parameters what an olive in an incorrect position is in order to establish if the olive is, indeed, in an incorrect position. As stated before, the olive is in an incorrect position for pitting when it is not perpendicular to the punch needle of its major axis. Fig. 3 shows the correct position of an olive, 90° from vertical.

If an olive is not in the correct position, it may be incorrectly destoned because the punch needle enters the olive in the same direction of the minor axis (Fig. 4). If an olive is in an incorrect position, its stone may crush, remain inside or spoil the fruit. Moreover, it may damage the machine since the punch needle or even the pocked bedding may break.

"Boat" position is established when the olive is between −45° and 45°. However, it is set based on an empirical approach for classification that may vary according to the needs of a specific factory, in which a wider or closer classification may be preferred or even the "boat" position may be extended from [−80°, −90°] to [80°, 90°].

The neural network needs very few images (20 images) to identify empty pockets due to the uniformity of images. On the other hand, the overexposure of empty pocket images leads to an excessively trained neural network, which begins to identify small details in order to distinguish between empty images and it produces a sub-classification, resulting in identification errors.

2.4. Hardware used in image capturing

The camera was selected considering that it must provide an external trigger, since image capturing must synchronize with the pocket and the illumination trigger. Another important point is that the camera itself provides the ROI or the image scale for future uses in real time. In this case, ROI and scale are handled using MATLAB software.

Several cameras have been tested and a choice for the computer

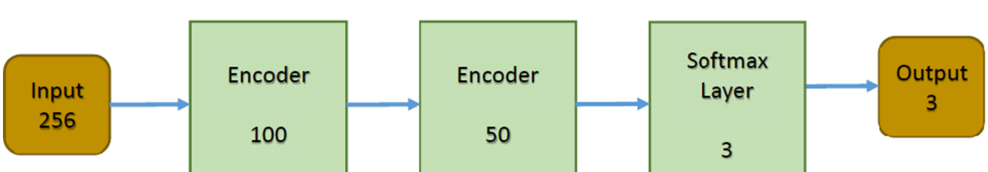

Fig. 2d. Stacked neural network diagram.

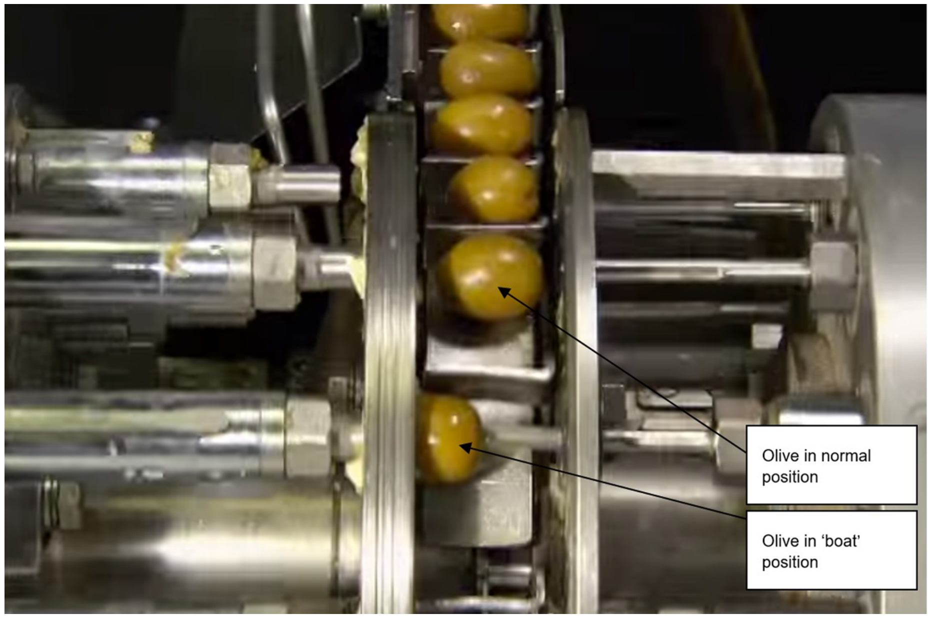

Fig. 4. Punch needles in an olive pitting machine (https://www.youtube.com/watch?v=c4-z3Jo6fOg).

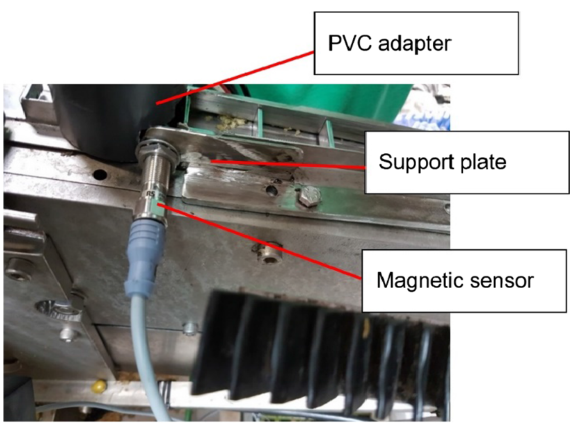

Fig. 5. Magnetic sensor used to detect the passage of the pockets in the chain.

vision system is an UI-1220SE-C USB industrial camera (IDS Imaging Development Systems GmbH, 2016). The camera trigger is activated in synchronization with the movement of the olive pitting, slicing and stuffing machine with a magnetic sensor (Fig. 5).

The detecting distance of the magnetic sensor is 8 mm, as seen in

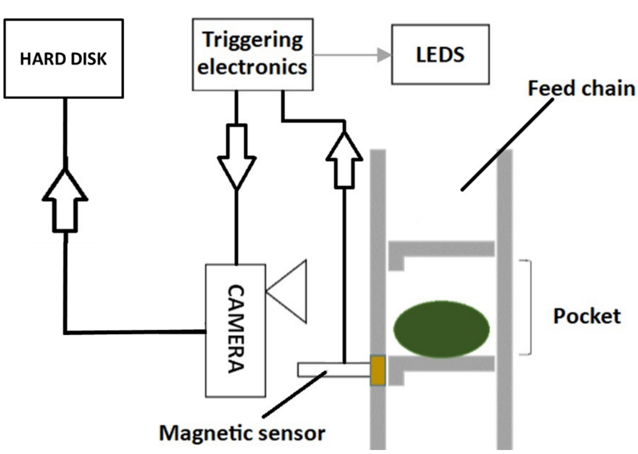

Fig. 6. Graphic of the implemented system.

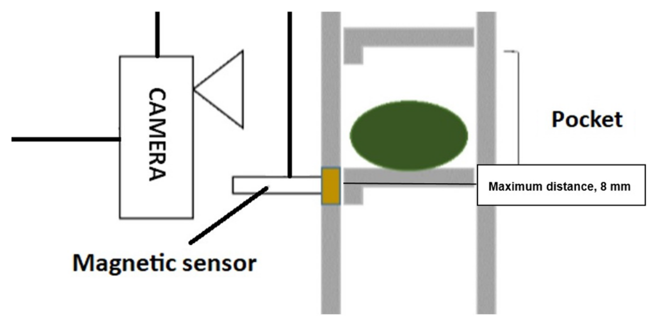

Fig. 7. Maximum distance between the sensor and the pocket.

Fig. 7. Therefore, it must be a flush sensor, allowing the pockets to move without hitting the sensor.

The magnetic sensor (see Fig. 5) activates the illumination trigger (a set of four 900-lumen of warm white light output power LED diodes, 3000 K) when it detects the passage of each pocket of the feed chain.

The generated light has pulses of about 2 ms and the minimum interval of time that takes an olive to pass in front of it is 60/2500 = 24 ms (when the machine is at its maximum operating speed of 2500 olives/minute). Consequently, LEDs are lit a total of 2/24*100 = 8.34% of the time frame, which prevents the device from warming, and the image is taken using an appropriate visibility and an exposure time of the camera of 1 ms.

The images are stored in the hard disk of a PC for later analysis. These images are snipped to specifically show the region of interest (ROI) and create 11 × 11 or 16 × 16 pixel images. Fig. 6 shows a diagram explaining how the system operates. Every part is represented by modules (industrial PC, camera, magnetic sensor and triggering electronics and LED lighting). There is a detailed part of the same diagram in Fig. 7 to state that the maximum distance between the sensor and the feed chain cannot exceed 8 mm.

The electronic circuit controls the pulse to turn on and turn off the LEDs and coordinate with the magnetic sensor and the trigger of the camera. There is an electronic circuit to clean the vision area in case of bad visibility. See Fig. 8.

Fig. 9 shows the implemented system over an olive pitting, slicing and stuffing machine (for small calibers, 'Manzanilla' and 'Hojiblanca' olive varieties) in a table olive factory in Seville.

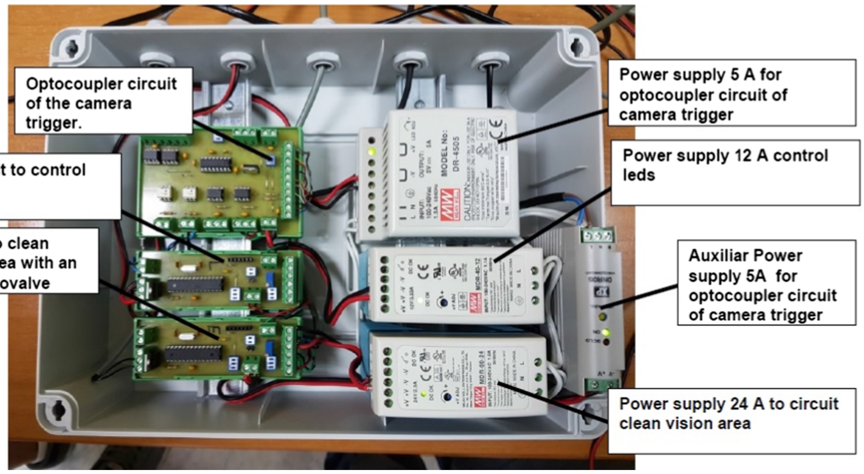

Fig. 8. Electronic circuit of the external trigger.

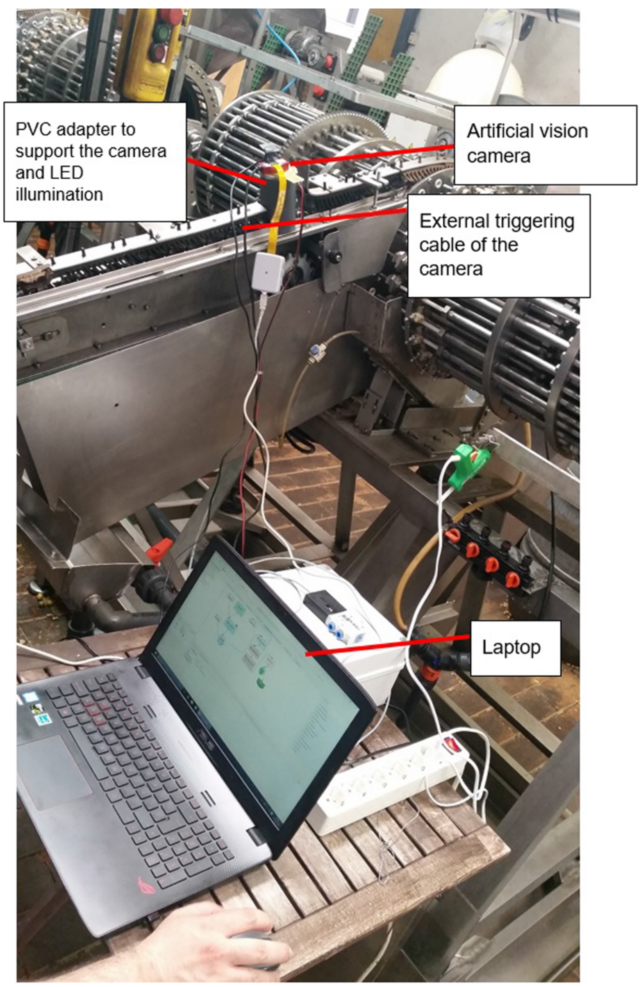

Fig. 9. System implemented in an olive pitting machine.

2.5. Neural chip used

The neuromorphic chip CogniMem CM1K used in this work has 1024 neurons and an input vector of 256 bytes has been used in a BrainCard (General Vision, 2017 Braincard). A BrainCard is a trainable

Fig. 10. Olive imagery samples, 16 × 16 ROI-scaled.

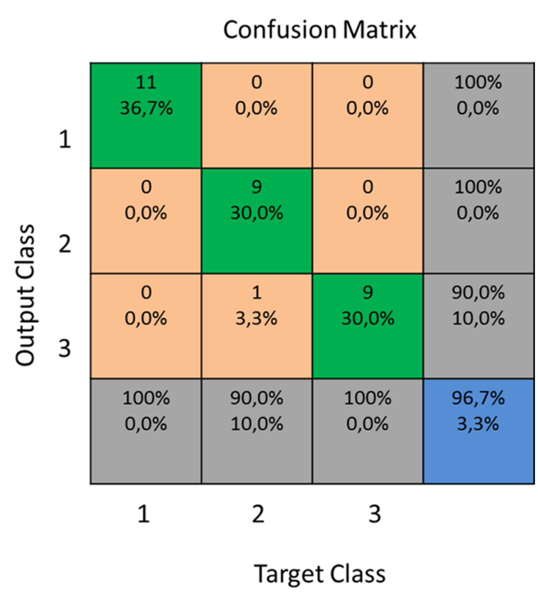

Fig. 11. Matrix of the obtained results using MATLAB (16 × 16 pixels resolution).

Table 1
Shows the average results of the 100 repetitions and standard deviation.

Parameters	Total	“Boat”	Standard Deviation	Empty	Standard Deviation	Normal	Standard Deviation
Samples	1000	69.94	1.36	88.11	1.49	841.95	1.38
Correct	887.24	62.92	1.53	87.16	1.42	737.16	1.48
Errors	24.94	0.33	0.47	0.42	0.50	24.19	1.34
Anomalous cases	88.36	7.06	0.72	0.44	0.50	80.86	1.42
Error rate (%)	[5.24, 2.40]	[1.16, −0.21]	0.007	[1.05, −0,10]	0.006	[3.04, 2.71]	0.002

pattern recognition board for IoT and smart appliances. This chip use (see Section 2.2), RBF or k-NN techniques for sorting purposes. 256-pixel images (16 × 16) will be processed with the CM1K chip.

3. Results and discussion

3.1. Results obtained using MATLAB neural network toolbox

The first one uses 1-byte depth greyscale images, 16 × 16 pixels ROI-scaled. For training purposes, a set of 9 images of empty pockets, 11 of “boat” de-stoned olives and 10 of normal olives has been used. Anomalous case images were not included since the purpose of the test was to prove the feasibility of the classification of these low-resolution images using a neural network. Fig. 10 shows olive images 16 × 16 pixels ROI-scaled.

The structure of the autoencoder of the neural network has been trained with 45 iterations, fixed by the MATLAB neural network library. The results obtained appear in Fig. 11.

As shown in Fig. 11, the neural network has carried out an appropriate classification (< 4% error rate) with 16 × 16 pixels images, which is the maximum resolution supported by the CM1K chip.

3.2. Results obtained using neuromorphic chip

For neuromorphic chip tests, a set of 20 images of empty pockets, 20 of “boat” de-stoned olives and 20 of normal olives has been used to train the neural network. Once trained, and for testing purposes, a set of 10,000 images taken during a test in an olive pitting, slicing and stuffing machine with ‘Hojiblanca’ olive variety in a working factory has been used. From this set, 1000 images were randomly chosen to perform sorting tasks. This test has been repeated 100 times.

Table 1 Average results of 100 repetitions and standard deviation using the NeuroMem CM1K.

As shown in Table 1, the neural network detects between 97.6% and 94.76% of cases correctly [2.40%, 5.24%] error rate. There is a percentage of anomalous cases out from the common errors mentioned classification. These cases may include two olives in the same pocket, broken olives or olives with a pitch close to “boat” position. The percentage of anomalous cases reaches [9.06%, 8.61%].

Four neurons have been used as follows: one neuron for an olive in a correct position, one neuron for an empty pocket and two neurons for “boat” cases because the NeuroMem CM1K detects enough variation so as to need two neurons for classification purposes. The system cannot identify untrained cases (such as broken olives, two olives in the same pocket, etc.) and returns value 255, meaning that the case is unclassifiable (see Section 2.2).

4. Conclusions

Training and testing of a neural network based on a physical chip to classify olives in the feed chain of a pitting machine have been successfully completed. The use of the physical chip NeuroMem CM1K, for its greater capacity and scalability, has been proven satisfactory and, therefore, it offers a great potential for sorting purposes. The CM1K chip detected between 97.6% and 94.76% of cases correctly. Only four neurons have been used for classification purposes: one neuron for empty pockets, one neuron for olives in a correct position and two neurons for “boat” position.

As a preliminary test, a MATLAB neural network has been designed to be trained in order to classify images with different resolutions with the aim of setting the lower resolution needed to classify olives in the feed chain of olive pitting machines. The resolutions of the images have been adapted to the capacities of the CM1K physical chip. The NeuroMem CM1K chip is able to process images and classify them appropriately with 11 × 11 pixels resolution and up to 16 × 16 pixels.

Processing times are high for usual pitting machines. The system reaches to 300 olives per minute, which is far below the 800–2500 olives per minute processed by industrial pitting machines. The reason is that the communication RS232C at 115200-baud rate generates a bottleneck between the CM1K and the PC. This problem will be studied and analyzed in following papers in order to improve time and adapt the system to industrial environments.

The use of a standard camera with an external trigger (UI-1220SE-C USB in this case) is proposed as the appropriate method to capture the images in greyscale (1-byte color depth), with ROI and scale compatible with physical neural chip. In order to capture the images properly, it is necessary to synchronize the trigger of the camera with pulsating illumination.

Besides what has been aforementioned stated, some conclusions obtained during the research process must be raised:

A thorough validation process of training patterns is needed and the better-selected training patterns are, the closer the neural network will be to classify images successfully. If this process is not performed, the error rate may rise up to 30%.

The simplest classification process is the most effective and, therefore, the lower the number of cases is, the better the neural network works because it gets confused with minor differences among cases. The more different the case is in comparison to others, the better the neural work performs. An error rate between 1.16% and −0.21% has been obtained, which proves the right functioning of the neural network to identify “boat” and anomalous cases.

The empty pocket case has nearly 100% reliability rate because differences within the same case are minor and, consequently, the network identifies the pattern with slight differences with tested cases during training. It allows the neural network to identify them easily. Although the other cases use similar olives for testing purposes, there will be always a variation caused by the diversity of the fruit, its position in the pocket and even the moment of image capturing when the olive moves through the feed chain, which explains that anomalous angles [30-70°] get under 5%. This is partly the reason why 100% effectiveness is not obtained.

Overtraining has been analyzed for empty pocket samples and results show that the system uses an extra neuron for training samples over 1000 patterns using quality training patterns and uniform reference for verification tests.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Behroozi Khazaei Nasser, Tavakoli Teymour, Ghassemian Hassan, Hadi Khoshtaghaza Mohammad, Banakar Ahmad, 2013. Applied machine vision and artificial neural network for modeling and controlling of the grape drying process.

Davies Mike, Srinivasa Narayan, Lin Tsung-Han, Chinya Gautham, Cao Yongqiang, Choday Sri Harsha, Dimou Georgios, Joshi Prasad, Imam Nabil, Jain Shweta, Liao Yuyun, Lin Chit-Kwan, Lines Andrew, Liu Ruokun, Mathaikutty Deepak, McCoy Steven, Paul Arnab, Tse Jonathan, Venkataramanan Guruguhanathan, Weng Yi-Hsin, Wild Andreas, Yang Yoonseok, Wang Hong, 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning.

Demuth Howard B., Beale Mark H., De Jess Orlando, Hagan Martin T., 2014. Neural Network Design.

Diaz, R., 2004. Classification and Quality Evaluation of Table Olives.

Frenkel Charlotte, Lefebvre Martin, Legat Jean-Didier, Bol David, 2019. A 0.086-mm2 12.7-pJ/SOP 64k-Synapse 256-Neuron Online-Learning Digital Spiking Neuromorphic Processor in 28-nm CMOS.

Fried Limor, 2019. Making machine learning arduino compatible: A gaming handheld that runs neural networks - [Resources_Hands On].

Gatica Gabriel, Bestb Stanley, Ceronic José, Lefranca Gastón, 2013. Olive fruits recognition using neural networks.

General Visión, 2019. NM500 Chip https://www.general-vision.com/documentation/TM_NeuroShield_GettingStarted.pdf.

General Visions, 2016. CM1K Chip http://www.general-vision.com/hardware/cm1k/.

General Visions, 2017. Braincard https://www.general-vision.com/documentation/TM_BrainCard.pdf.

General Visions, 2017. TM_CM1K_Hardware_Manual. https://www.general-vision.com/documentation/TM_CM1K_Hardware_Manual.pdf.

General Visions, 2018. CogniPat SDK for Matlab. https://www.general-vision.com/download/cp_sdk_ml/.

General Visions, 2019. NeuroMem USB dongle. https://www.general-vision.com/hardware/usbdongle/.

General Visions, 2019. TM_NeuroMem_Technology_Reference_Guide. https://www.general-vision.com/documentation/TM_NeuroMem_Technology_Reference_Guide.pdf.

General Visions. DS_CM1K. https://www.general-vision.com/datasheet/DS_CM1K.pdf.

General Visions. TM_TestNeurons_SimpleScript. http://www.general-vision.com/documentation/TM_TestNeurons_SimpleScript.pdf.

Goodfellow Ian, Bengio Yoshua, Courville Aaron, 2016. Deep Learning.

Halgamuge Saman K., Poechmueller Werner, Glesner Manfred, 1994. An alternative approach for generation of membership functions and fuzzy rules based on radial and cubic basis function networks.

Haykin Simon, 1999. Neural Networks and Learning Machines.

Hecht-Nielsen Robert, 1992. Theory of the Backpropagation Neural Network.

Hubbard Nicholas, 2019. Using Neural Networks for Object Detection in Robotics Competitions.

IDS Imaging Development Systems GmbH, 2016. https://es.ids-imaging.com/store/ui-1220se.html.

Kim Jungyun, 2019. New Neuromorphic AI NM500 and Its ADAS Application.

Liu YiQing, Wei Dong, Zhang Ning, 2011. Vehicle-license-plate recognition based on neural networks.

Lobachev Ivan, Maleryk Roman, Antoschuk Svitlana, Filiahin Denys, Lobachev Mykhaylo, 2018. Integration of neural networks into smart sensor networks.

Mancuso, S., Nicese, F.P., 1999. Identifying Olive (Olea europaea) Cultivars Using Artificial Neural Networks.

Menendez Anne, Paillet Guy, 2008. Fish inspection system using a parallel neural network chip and the image knowledge builder application.

Menendez Anne, Paillet Guy, 2013. Monolithic image perception device and method.

Moller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533.

Moradi, S., Qiao, N., Stefanini, F., Indiveri, G., 2018. A scalable multi-core architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps).

Moran Steven, Gaonkar Bilwaj, Whitehead William, Wolk Aidan, Macyszyn Luke, Iyer Subramanian S., 2018. Deep learning for medical image segmentation – using the IBM TrueNorth neurosynaptic system.

Nagata and Qixin, 1998. Computer Vision Technology for Food Quality Evaluation.

Pierre, 2012. Autoencoders. Unsupervised Learning, and Deep Architectures.

Sardar Santu, Tewari Gaurav, Babu, K.A., 2011. A hardware/software co-design model for face recognition using Cognimem Neural Network chip.

Sun Da-Wen, 2008. Computer Vision Technology for Food Quality Evaluation.

The MathWorks Inc, Train Stacked Autoencoders for Image Classification, 1994-2019 https://es.mathworks.com/help/deeplearning/examples/train-stacked-autoencoders-for-image-classification.html.

The MathWorks, Inc. wine-classification-with-neural-pattern-recognition. 1994-2017. https://es.mathworks.com/videos/wine-classification-with-neural-pattern-recognition-tool-68798.html.

The MathWorks, Inc. Neural-network. 1994-2017. http://es.mathworks.com/discovery/redes-neuronales.html.

The MathWorks, Inc. Train Autoencoder. 1994-2017. http://es.mathworks.com/help/nnet/ref/trainautoencoder.html.

Yang, Q., 1993. Classification of apple surface features using machine vision and neural networks. Comput. Electron. Agric. 9, 1–12. https://doi.org/10.1016/0168-1699(93)90025-V.