IPSJ Transactions on Computer Vision and Applications Vol.7 1–9 (Jan. 2015)

[DOI: 10.2197/ipsjtcva.7.1]

Research Paper

Neural 1D Barcode Detection Using the Hough Transform

Alessandro Zamberletti[1] Ignazio Gallo[1,a)] Simone Albertini[1] Lucia Noce[1]

Received: January 28, 2014, Accepted: October 2, 2014, Released: January 28, 2015

Abstract: Barcode reading mobile applications to identify products from pictures acquired by mobile devices are widely used by customers from all over the world to perform online price comparisons or to access reviews written by other customers. Most of the currently available 1D barcode reading applications focus on effectively decoding barcodes and treat the underlying detection task as a side problem that needs to be solved using general purpose object detection methods. However, the majority of mobile devices do not meet the minimum working requirements of those complex general purpose object detection algorithms and most of the efficient specifically designed 1D barcode detection algorithms require user interaction to work properly. In this work, we present a novel method for 1D barcode detection in camera captured images, based on a supervised machine learning algorithm that identifies the characteristic visual patterns of 1D barcodes' parallel bars in the two-dimensional Hough Transform space of the processed images. The method we propose is angle invariant, requires no user interaction and can be effectively executed on a mobile device; it achieves excellent results for two standard 1D barcode datasets: WWU Muenster Barcode Database and ArTe-Lab 1D Medium Barcode Dataset. Moreover, we prove that it is possible to enhance the performance of a state-of-the-art 1D barcode reading library by coupling it with our detection method.

Keywords: barcode detection, 1D barcode, Hough Transform, Neural Network, Multilayer Perceptron

1. Introduction

In the last few years, online shopping has grown constantly, and so have the number of customers that use smartphones or tablets to purchase products online. Most of those mobile devices integrate high quality cameras; as such, many researchers and companies focused on solving the problem of identifying products shown in camera captured images on the basis of their visual features [1]. However, the task of recognizing both the brands and the models of products from pictures has yet to be efficiently solved; this is mostly due to the large number of issues that need to be addressed when using images captured by customers, such as poor light conditions, occlusions and variations in poses and scales.

As proved by many successful applications currently available in most mobile application stores, an easier and more efficient way to approach the object identification task in the field of e-commerce lies in exploiting barcodes, as they univocally identify almost every item in the market and have strong characteristic visual patterns [2].

While both the detection and the decoding tasks have already been exhaustively faced for two-dimensional (2D) barcodes (e.g., Quick Read codes) [3], [4], [5], the same does not hold for one-dimensional (1D) barcodes, even though Universal Product Codes (UPC) and European Article Numbers (EAN) barcode standards are widely diffused all over the world.

The task of reading 1D barcodes from camera captured images has been approached in different ways [6], [7], [8], [9], [10], [11], [12]. Most of the currently available barcode reading mobile applications analyze the gray intensity profiles of single lines in the given images to identify typical patterns associated with the parallel bars of 1D barcodes, thus they usually require the user to place the barcodes in a specific position within the camera screen [12]. Some industrial approaches obtain excellent results using hardware implementations of their barcode reading softwares [13] while exploiting prior knowledge related to the specific domain, e.g., the dimension and the position in which a barcode may appear inside the processed image. Other works propose different techniques of decoding 1D barcodes to deal with camera related issues, such as poor light conditions or lack of focus [9], [10], [11].

Currently, most of the 1D barcode related works in literature address the decoding phase and treat the underlying detection task as a side problem; nonetheless, we argue that the task of detecting multiple arbitrary rotated barcodes in real world images is crucial to reduce the amount of user interaction involved in the subsequent decoding process. Moreover, real time angle invariant barcode detection algorithms can be exploited by automated systems to identify products without defining placement or ordering constraints.

Even though 1D barcodes can be effectively detected by general purpose object detection methods, this is not an optimal solution, as most of the interesting applications of barcode reading algorithms lie in the mobile field and the majority of currently available mobile devices do not meet the minimum working requirements of those object detection approaches.

In this work, we propose an Hough Transform [14] based method for angle invariant 1D barcode detection in camera captured images. In our method, a properly trained supervised ma-

[1] University of Insubria, Department of Theoretical and Applied Science, Via Mazzini, 5, 21100 Varese, Italy

a) ignazio.gallo@uninsubria.it

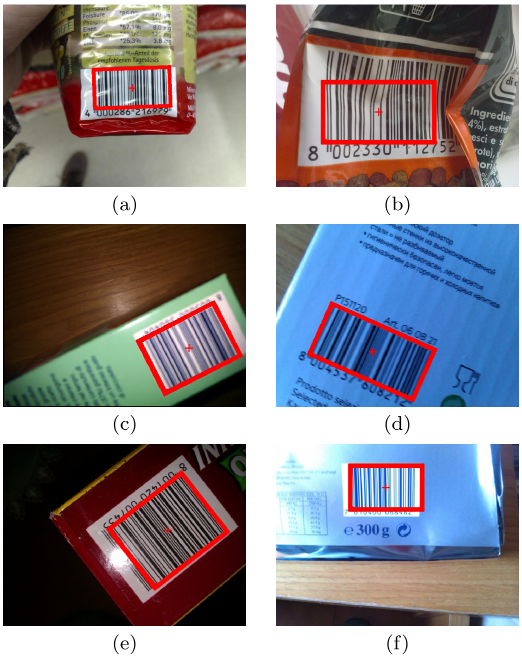
(a) (b) (c) (d) (e) (f)

Fig. 1 Examples showing the bounding boxes automatically identified by the proposed method for arbitrary rotated 1D barcodes in images acquired using smartphones. The results produced by the proposed method are satisfying even when the 1D barcodes are warped, occluded or partially illegible due to reflections.

chine learning model identifies the rotation angle of every barcode in the processed image by analysing the two-dimensional Hough Transform discrete space of the given image; a subsequent simple and fast procedure detects the bounding boxes surrounding those barcodes by exploiting the information received from the machine learning model.

The main novelties of our approach are: (i) the ability of detecting the exact positions and rotation angles of 1D barcodes without exploiting prior knowledge nor user interaction, (ii) the capacity of precisely identifying the bounding boxes for 1D barcodes in the processed image, this is particularly interesting because the identified bounding boxes can be used in a subsequent decoding phase to efficiently read barcodes without wasting any time searching for them in the processed image.

In our experiments, we prove that the proposed model can obtain excellent results for three different 1D barcode datasets and that it is also effective in detecting barcodes that are warped, partially occluded or illegible due to reflections, some examples are provided in **Fig. 1**. We publicly release the source code used in our experiments as it can be used by most of the barcode reading algorithms presented in literature [*1]; we also introduce a new dataset specifically designed to evaluate the performances of angle invariant 1D barcode detection methods. A live demo of our approach is available online and can be used to evaluate the proposed method with user uploaded images [*2].

The method presented in this manuscript is an extension of our previous work on barcode detection [15], in this extended version we provide a deeper analysis of all the components involved in the method's pipeline in order to understand how they are affected by parameter changes.

We have also previously proposed a robust 1D barcode decoding system [10] which can be combined with the detection method presented in this paper to provide an effective end-to-end system for 1D barcode reading on mobile devices or desktop applications.

2. Related Works

2.1 Hough Transform

The classical Hough Transform [14] is a feature extraction technique commonly used in Image Processing and Computer Vision for the detection of regular shapes such as lines, circles or ellipses.

The Hough Transform for lines detection adopts a voting procedure to identify the set of linear shapes L in a given image ι. The normal form equation of a generic line $l \in L$ in ι can be defined as follows:

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \qquad (1)$$

where $\rho \geq 0$ is the distance of l from the origin of ι and $\theta \in [0, 2\pi)$ is the angle of l with the normal.

Let the two-dimensional Hough Transform space H be the (ρ, θ) plane, for an arbitrary point $(x_i, y_i) \in I$, Eq. (1) corresponds to a sinusoid in H. If two points $(x_0, y_0), (x_1, y_1) \in I$ belong to the same line l, their corresponding sinusoids intersect in a point $(\rho_l, \theta_l) \in H$; the same holds true for all the points of l. Note that the coordinates of the point $(\rho_l, \theta_l) \in H$ correspond to the main parameters of l, therefore it is possible to detect the set of linear shapes L by identifying the points of intersection in the Hough Transform space H of the given image ι.

In a discrete implementation, the Hough Transform algorithm uses a two-dimensional array A, called *accumulator*, to represent the plane H. In its first step, the algorithm executes an edge detection algorithm on ι. Let ι_e be the edge map computed for ι, for each pixel $p \in I_e$ the Hough Transform algorithm determines whether p corresponds to an edge in ι, if so, for every line l_p (in the discrete space defined by A) that may pass through p, the algorithm increases the value of the element in A that corresponds to the main parameters of l_p. Finally, the linear shapes in ι are identified by applying a local threshold operator to A to detect its peak elements.

2.2 Barcode Detection

The barcode detection task consists in locating the barcodes that appear in a given image; the output of a barcode detection algorithm should consist of a set of bounding boxes that surround those barcodes.

This task has been faced using many different techniques, for example: (i) in Refs. [12], [16] scan lines are drawn over the image to detect the exact position of a barcode, (ii) Basaran et al. [17] rely on the properties of the Canny Edge Detector [18] to identify edges corresponding to the parallel bars of 1D barcodes, (iii) Gallo and Manduchi [9] assume that the regions in the image characterized by weak horizontal gradients and strong vertical gradients correspond to barcodes. In order for those cited models to operate effectively, the barcodes shown in the processed images need to satisfy a set of constraints, e.g., none of them can

[*1] http://github.com/SimoneAlbertini/BarcodeDetectionHough
[*2] http://artelab.dicom.uninsubria.it/projects/projects.html

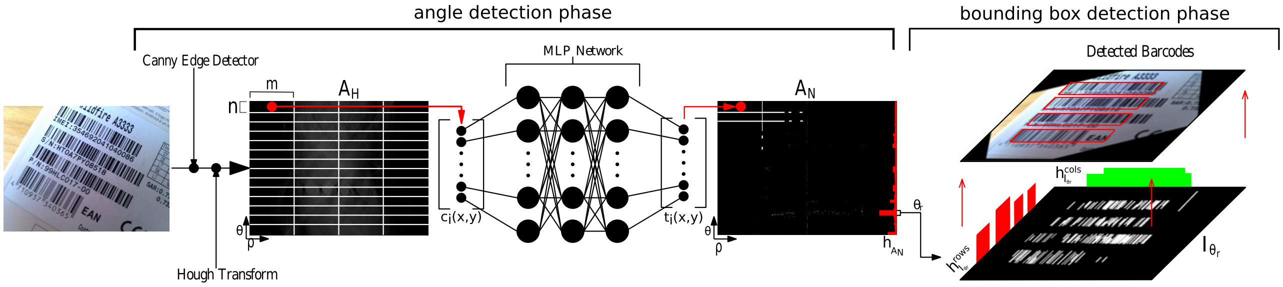

Fig. 2 A visual overview of the sequence of steps performed by the proposed model to detect the bounding boxes for an image extracted from the *Rotated Barcode Database*.

detect arbitrary rotated 1D barcodes.

2.3 Barcode Decoding

The barcode decoding task consists in exploiting the information provided by a barcode detection algorithm to identify the sequence of numbers represented by the barcodes appearing in a given image. As for barcode detection, the task of decoding 1D barcodes is really challenging because it has to deal with issues that mostly depends on the final domain of application; for example, when designing a barcode reading algorithm for mobile devices, the problem of decoding blurred barcodes needs to be addressed in order to obtain a robust and effective algorithm.

There are many interesting works in literature that face the 1D barcode decoding task: (i) Gallo and Manduchi [9] exploit deformable templates to efficiently read extremely blurred barcodes, (ii) in [8], [12], [16] the authors adopt different thresholding techniques to decode the typical black parallel bars of 1D barcodes, (iii) Zamberletti et al. [10] use a supervised neural network to improve the performance of the Zebra Crossing (ZXing) [19] algorithm, (iv) Muñiz et al. [20] decode 1D barcodes by “reading” the sequence of intensity values in the accumulator matrix of the linear Hough Transform algorithm. Note that the work of Muñiz et al. [20] does not perform any detection as it assumes that the images provided as input to the model consist in 1D barcodes cropped exactly to their bounding boxes.

Based on the results provided by the authors, the algorithm proposed by Gallo and Manduchi [9] proves to be significantly more robust than the others when applied to images acquired by mobile devices. The methods presented in [9], [10], [19] are able to efficiently read 1D barcodes even on low end mobile phones.

3. Proposed Model

A detailed description of the proposed method is given in the following paragraphs.

Given an image ι, we apply the Canny Edge Detector [18] to ι to obtain its edge map ι_e; this step can be performed efficiently even on a mobile device, for example by exploiting the implementation of the Canny Edge Detector algorithm provided by the *OpenCV* library [21], like we currently do in our experiments and mobile application. Once the edge map ι_e has been determined, we compute its Hough Transform in the two-dimensional space $H = (\rho, \theta)$. Finally, as described in Section 3.1, we detect the rotation angle for the barcode in ι and subsequently identify its bounding box by exploiting both a neural generated accumulator matrix and the linear segments identified by the Hough Transform, as described in Section 3.2. In **Fig. 2** we provide a simple visual representation of the sequence of steps performed by our proposed method to detect the barcode bounding boxes for an image belonging to the *Rotated Barcode Database* (introduced in Section 4.1).

3.1 Angle Detection

Let A_H be the accumulator matrix for the two-dimensional Hough Space $H = (\rho, \theta)$ of an image ι; as described in Section 2.1, every element $a \in A_H$ in position (ρ_a, θ_a) denotes a potential line in ι_e having rotation angle θ_a and distance from the origin of the axes ρ_a. The goal of the angle detection phase is to identify which elements of A_H are associated with barcode bars in ι.

Even though this task may look easy at first glance because of the regularities usually shown by 1D barcodes (parallel black bars, guard bars on the left and right, numbers under the bars, etc.), we were not able to solve it using a set of handcrafted rules because of all the issues that arise when identifying barcode bars in the Hough Transfom accumulator matrices of real world images. For example, a barcode may appear warped, this causes the elements denoting its barcode bars in the accumulator matrix to have different values for the angle coordinate θ. The barcode may also appear partially illegible due to reflections and this causes the expected regular sequence of elements denoting its barcode bars in the accumulator matrix to be irregular, as some of those bars are not correctly identified by the edge detection algorithm. Most of the other examples that can be provided are related to the issues usually faced by the decoding algorithms described in Section 2.3 (lack of focus, scale variations, etc.). For those very reasons, we decided to employ a supervised machine learning algorithm in place of handcrafted rules to efficiently solve the angle detection task.

As shown in **Fig. 3**, a regular grid of cells C is superimposed over A_H; the height and the width of each cell are defined as n and m respectively. Every cell $c_i \in C$ is processed by a Multi Layer Perceptron (MLP) [22] neural network that produces a new cell t_i as follows. Let $c_i(x, y)$ be the value of the element of c_i in position (x, y) with $0 \leq x < n$ and $0 \leq y < m$, the value assigned by the MLP network to its corresponding element $t_i(x, y)$ is defined as follows:

$$t_i(x, y) = \begin{cases} 1 & \text{if } c_i(x, y) \text{ is a barcode bar in } \iota. \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

where an element $c_i(x, y)$ denotes a barcode bar in ι if the sinu-

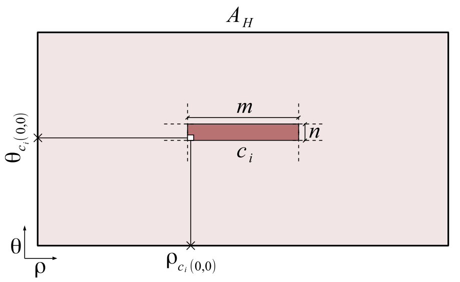

Fig. 3 An example showing how the regular grid of cells C is superimposed over the Hough accumulator matrix A_H and how the elements $c_i(x, y)$ are mapped into the (ρ, θ) plane.

soid defined by Eq. (1) for the pair $(\rho_{c_i(x,y)}, \theta_{c_i(x,y)})$ corresponds to a barcode bar in ι. As stated by Eq. (2), the goal of the neural network is to assign an high intensity value to $t_i(x, y)$ if it corresponds to a $c_i(x, y)$ that denotes a barcode bar in the original image ι.

Let Ω be a dataset splitted into training and test sets defined as Ω_{train} and Ω_{test} respectively, in which $\Omega_{\text{train}} \cap \Omega_{\text{test}} = \emptyset$ and $\Omega_{\text{train}} \cup \Omega_{\text{test}} = \Omega$; in order to accomplish the task described by Eq. (2), the neural network model needs to be properly trained using a set of patterns generated from the samples in $\Omega_{\text{train}} \subset \Omega$ as follows.

Let $\omega_i \in \Omega_{\text{train}} \subset D$ be a training image, each training pattern extracted from ω_i consists in a pair (in, out) in which:

- *in* - is the linear representation of a cell extracted from the Hough Transform accumulator matrix of the training image ω_i.
- *out* - is the linear representation of the same cell associated with *in*, in which, according to Eq. (2), the elements of *in* that denote barcode bars in the original training image ω_i are assigned 1 as intensity value; the remaining are assigned 0.

Once all the training patterns for the training images in Ω_{train} have been extracted from A_H, they are used to train the MLP using the resilient backpropagation algorithm with the default parameter values proposed by Igel et al. [23].

The trained network can be exploited to solve the task described by Eq. (2) simply by processing all the cells defined for the accumulator matrix A_H of the image ι; once all those cells have been processed, they are combined together to generate a new accumulator matrix A_N, called *neural accumulator matrix*, in which the elements having high intensity values represent potential barcode bars.

Finally, we identify the correct rotation angle of the barcode in ι by analizing A_N. Taking into account that the main feature of a 1D barcode is that its bars are parallel, we expect the elements denoting those bars in the Hough Transform accumulator matrix of ι to appear in the same row of A_N; for this reason, we define the likelihood l_r of a barcode appearing in ι rotated by the angle associated with a row r in A_N as the sum of all the elements of r. This process is repeated for all the k rows of A_N to obtain an histogram $h_{A_N} = [b_0, \ldots, b_k]$ in which each bin b_j, $0 \leq j \leq k$ represents the likelihood that the elements of the row j denote the bars of a barcode in ι. Examples showing the angle detection procedure applied to images from *ArTe-Lab 1D Medium Barcode Dataset* are provided in Figs. 2 and **4**.

Let b_r be a bin in h_{A_N} and $max(h_{A_N})$ be the maximum value among the bins of h_{A_N}, if b_r is equal to $max(h_{A_N})$ then we assume that some of the elements of r denote the bars of a barcode in ι. Given θ_r as the rotation angle specified by the row r of A_N, without further operations, a set of scan lines [12], [16] rotated by θ_r could be performed all over the image ι to decode the barcode associated with r. However, this is an expensive operation, as those scan lines must be performed over all the lines of ι having rotation angle θ_r. For this reason, we decided to try a more efficient and novel way to decode the barcode by running a single scan line in the middle of its bounding box, as described in Section 3.2.

3.2 Bounding Box Detection

Given the neural accumulator matrix A_N for a given image ι and the rotation angle θ_r identified by the angle detection phase described in Section 3.1, we want to determine the bounding boxes for the 1D barcodes in ι. There are many reasons why we want to identify those bounding boxes, e.g., they can be exploited to speed up the decoding process because we do not need to run scan lines all over the images but just in the middle part of the detected bounding boxes, or to define commercial applications that overlay images/videos over the areas occupied by 1D barcodes to create particular visual effects and enhance the user experience during the decoding phase.

To identify those barcode bounding boxes, we start by determining the set S of all the linear segments in ι by applying the same technique used by Matasyz et al. [24] to the edge map ι_e. Given the rotation angle θ_r identified by the angle detection phase, we define $S_{\theta_r} \subseteq S$ as the subset of linear segments in S whose rotation angles differ by at most ±5° from θ_r and we create a binary image I_{θ_r} in which the intensity value assigned to the pixels belonging to the segments in S_{θ_r} is 1, while the others are assigned 0. The binary image I_{θ_r} is then rotated so that the majority of its segments are parallel to the vertical. Similarly to Section 3.1, we define two histograms $h^{\text{rows}}_{I_{\theta_r}}$ and $h^{\text{cols}}_{I_{\theta_r}}$ that describe the intensity profile of the rows and the columns of I_{θ_r} respectively, as shown in Fig. 2. More specifically, each bin of those histograms is computed as the sum of the elements of a row/column in I_{θ_r}. Finally, we apply a smoothing filter to each histogram to remove low value bins corresponding to isolated non-barcode segments and we determine the bounding box for the barcodes as the intersection area between the rows and the columns associated with the remaining non-zero bins in $h^{\text{rows}}_{I_{\theta_r}}$ and $h^{\text{cols}}_{I_{\theta_r}}$ respectively. All the previously described operations can be performed in parallel for each barcode in ι.

3.3 Discussion

The computational complexity of the proposed model strictly depends on the size of the accumulator A_H. Note that, due to its aspect ratio, it is possible to successfully decode a non-deformed 1D barcode using a scan line if the rotation angle of the scan line differs by at most ±30° from the one of the barcode [9]. This feature enables us to obtain good results even when a single row in A_H is associated with multiple consecutive rotation angles. In theory, we could speed up the detection process by employing accumulator matrices having just $\frac{180}{30}$ rows, in which each row

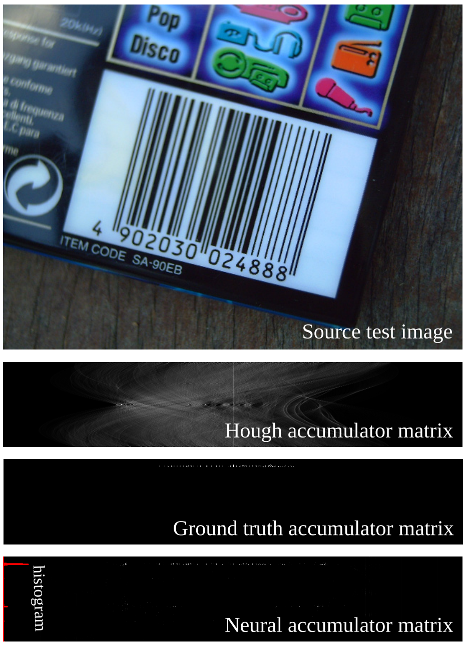

Fig. 4 An example showing the accumulator matrices generated by the angle detection phase for an image in the *ArTe-Lab 1D Dataset*. In the right part of the figure, from top to bottom: the original accumulator matrix A_H; the ground-truth accumulator matrix in which elements denoting barcode bars in the original image have been assigned an high intensity value; the neural accumulator matrix A_N generated by our trained neural model for A_H.

represents 30 consecutive angles. In practice, as shown in our experiments, this is not feasible, since the neural network is not able to correctly recognize the visual patterns produced by barcode bars because of the limited number of training patterns that can be extracted from those "condensed" accumulator matrices using the training procedure described in Section 3.1.

As proved in Section 4.3, the capability of the MLP network to detect warped barcodes depends on the parameters n and m. This is due to the fact that the bars of a warped barcode (e.g., a barcode printed on an irregular object) are not parallel, therefore some of the points generated in A_H for such bars lie on different subsequent rows; if we increase n, each cell provided as input to the MLP network spans over multiple subsequent rows of A_H and this enables the neural model to successfully identify those multiple rows patterns characteristic of warped 1D barcodes.

As described in Section 4.3, in our experiments we explore the possibility of employing overlapping cells instead of the regular grid C superimposed over A_H; as shown by the results we obtained, we decided not to use this approach, as it substantially increases the time required by the neural model to process a single image and does not increase the overall angle detection performance.

The computational complexity of the bounding box detection phase described in Section 3.2 depends on the size of the input image ι. In our experiments we always rescale ι to a 640×480 pixels resolution without losing overall detection accuracy.

4. Experiments

4.1 Datasets

In this section we provide an overall description of the datasets used to evaluate the performance of the proposed model.

In our experiments, we employ two standard 1D barcode datasets: the *ArTe-Lab 1D Medium Barcode Dataset* [10] and the *WWU Muenster Barcode Database* [12]; as described in the next paragraphs, those two datasets do not contain images of barcodes rotated by more than ±30° from the vertical, for this reason we decided to build an additional dataset, called *Rotated Barcode Database*, specifically designed to evaluate the performances of angle invariant 1D barcode detection algorithms. In order to enable other researchers to compare their results with ours, this last dataset has been made publicly available on our website. Since our method involves a supervised machine learning algorithm, we split each dataset into training and test sets: we randomly select 66% of the dataset's images as training set Ω_{train} and the remaining 33% as test set Ω_{test}. Moreover, in order to evaluate the accuracy of the bounding box detection phase described in Section 3.3, we define the ground truth figure-ground segmentation masks for all the images of the previously cited datasets; those masks have also been made available online on our website.

A more detailed description of the datasets used in our experiments is given in the following paragraphs.

ArTe-Lab 1D Medium Barcode Dataset [10]. It consists of 215 1D barcode images acquired using a Nokia 5800 mobile phone. This dataset is not specifically designed to evaluate the performances of angle invariant algorithms; as such, the barcodes appearing in the images are rotated by at most ±30° from the vertical. Each image contains at most one non-blurred EAN barcode. In our experiments, we do not employ the extended version of this dataset because the proposed method is not specifically designed to deal with unfocused images.

WWU Muenster Barcode Database [12]. It consists of 1055 1D barcode images acquired using a Nokia N95 mobile phone. As for the *ArTe-Lab 1D Medium Barcode Dataset*, this dataset has not been specifically designed for angle invariant detection algorithms, for this reason most of the barcodes that appear in the images are not rotated from the vertical. Each image contains at most one non-blurred EAN or UPC-A barcode.

Rotated Barcode Database. It consists of 368 1D barcode images acquired using multiple smartphones; all the images are scaled to a 640×480 pixels resolution. This dataset is specifically designed to evaluate the performances of angle invariant barcode detection algorithms; as such, the barcodes appearing in the images are rotated by arbitrary angles. Each image may contain multiple EAN and UPC barcodes, moreover, the barcodes may appear warped, illegible due to reflections or partially occluded. The dataset is publicly available for download and use [*3].

4.2 Evaluation metrics

We measure the performances of the two main phases of the proposed model using the *overall angle detection accuracy* OA^{θ} for the angle detection phase of Section 3.1 and the *overall bounding box detection accuracy* OA^{bb} for the bounding box detection phase of Section 3.3. Those two metrics are defined as follows.

[*3] http://artelab.dicom.uninsubria.it/downloads.html

Overall angle detection accuracy. Given a set of test images Ω_{test}, the overall angle detection accuracy achieved by the proposed model for Ω_{test} is computed as follows:

$$OA^{\theta} = \frac{tp}{tp + fn + fp} \tag{3}$$

where tp is the number of barcode rotation angles successfully detected in Ω_{test}, $tp + fn$ is the total number of 1D barcodes that appear in the images of Ω_{test} and fp is the number of objects wrongly identified as barcodes. The rotation angle detected for a barcode b is considered correct if it differs by at most ±10° from the truth rotation angle θ_b. In Section 4.3, we provide a sensitivity analysis of the angle detection phase by reducing this tolerance value to check the precision and the robustness of the proposed method, results are shown in **Fig. 5**.

Overall bounding box detection accuracy. Given a set of test images Ω_{test}, the overall bounding box detection accuracy OA^{bb} is calculated by redefining tp in Eq. (3) as the number of barcode bounding boxes correctly detected. Let bb_b be the bounding box for a barcode b, a detected bounding box d_b is considered correct for b if the following condition holds:

$$\frac{|bb_b \cap d_b|}{|bb_b \cup d_b|} \geq 0.5 \tag{4}$$

4.3 Results

In this section we discuss the results obtained by the two phases of the model presented in Section 3 for the three datasets described in Section 4.1.

In all our experiments we adopt an MLP neural network composed by a single hidden layer whose size is equal to $n \times m$, where n and m are the height and the width of each cell in the regular grid C superimposed over the Hough Transform accumulator matrix, as described in Section 3.1. The MLP is trained using resilient backpropagation with the default parameter configuration of Igel et al. [23], using 150 “background” and 50 “foreground” training patterns randomly extracted from each image in the given train set Ω_{train}. A training pattern $p = (in, out)$ is a “background” training background pattern if it satisfies the following condition:

$$\forall x \in out, x \neq 1 \tag{5}$$

if this condition is not met then p is a “foreground” training pattern.

For every experiment, we define the accumulator A_H as a matrix having 180 rows and $\sqrt{2} \cdot max(h, w)$ columns, where h and w are the height and the width of ι respectively.

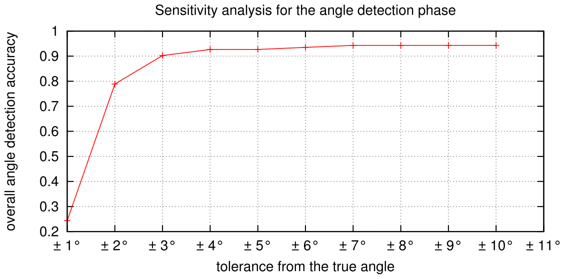

Fig. 5 Overall angle detection accuracy OA^{θ} achieved by our angle detection method for the *Rotated Barcode Database* while varying the tolerance from the true angle values. The predictions generated by this phase almost always fall within ±4° from the correct values.

Angle detection evaluation. In our first experiment we analyze the performances of the angle detection phase described in Section 3.1 while varying the size of the cells extracted from A_H; the results we obtain are shown in **Fig. 6**. It is possible to observe that, as stated in Section 3.3, the parameters n and m deeply affect the overall angle detection accuracy. The best value for m is 3; lower values do not allow the MLP network to detect warped barcodes while higher values introduce too much noise in the patterns processed by the MLP network and reduce its effectiveness. Overall, we achieve excellent angle detection performances: if we set $n = 3$ and $m = 61$, we obtain a 100% OA^{θ} for the simple *ArTe-Lab 1D Medium Barcode Dataset* and an average of 95.5% OA^{θ} for the other two datasets. In this configuration, the time required to process an image is roughly 200 ms on a Samsung I9300 Galaxy S III.

These overall accuracies have been measured by considering a prediction correct if it falls within ±10° from the true angle value. However, it may be interesting to evaluate the precision of our method by measuring the distance between the predictions it generates and the true values in a more detailed way; for this reason, we compute the overall angle accuracy OA^{θ} achieved by our method while varying the accepted tolerance from ±10° to ±1° from the true values. Results are shown in Fig. 5 ; the predictions generated by our method almost always fall within ±4° from the true values. Note that, as stated in Section 3.1, such degree of precision is acceptable thanks to the typical geometric proportions of 1D barcodes.

Another interesting experiment consists in understanding

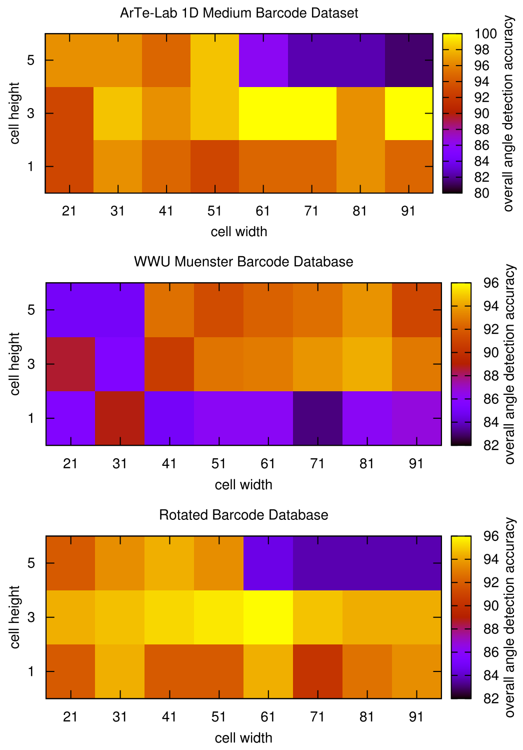

Fig. 6 Overall angle detection accuracy OA^{θ} achieved by the proposed angle detection method described in Section 3.1 for the three 1D barcode datasets presented in Section 4.1 while varying the height n and the width m of the cells extracted from the Hough Transform accumulator matrix.

whether our neural filtering technique can be improved by processing multiple times every element of the accumulator matrix A_H. Given $n = 3$ and $m = 61$ as the pair of parameters that provide the best overall results, instead of extracting those cells as described in Section 3.1 (using a regular grid superimposed over A_H), we extract them using a $n \times m$ sliding window approach. The sliding window strides over A_H from left to right and from top to bottom with different horizontal and vertical step values. The results we obtain from adopting this method are shown in **Fig. 7**. Note that, when the horizontal step on the ρ-axis is less than 61 or the vertical step on the θ-axis is less than 3, every element in A_H is processed multiple times by the MLP and those multiple predictions are averaged to obtain the neural accumulator matrix A_N. Our results prove that this "overlapped" approach performs well when the vertical step is equal to 1, mainly because every row of A_H is processed 3 different times by the MLP. However, this technique increases the time required by the MLP to process a single accumulator matrix approximately by an order of magnitude; moreover the best results are achieved when the horizontal step is equal to 61 and the vertical step is equal to 3. In such situation the "overlapped" approach corresponds to our simple grid approach.

In **Fig. 8**, we evaluate the performance of the MLP on Rotated Barcode Database using the optimal cell size ($n = 3$ and $m = 61$), while varying both the number of hidden layer and the number of hidden units. The best results are obtained when using 2 hidden layers with 200 hidden units each. As the number of hidden units increases, the MLP starts to overfit the training data and loses its generalization ability. The MLP has been trained using the default parameter configuration proposed by Igel et al. [23]. Since the MLP has to be implemented in a mobile enviroment, in our experiments we employ a MLP just having a single hidden layer and the same amount of input, hidden, and output units: 3×61; this is the configuration that provides the best compromise between performance and computational complexity.

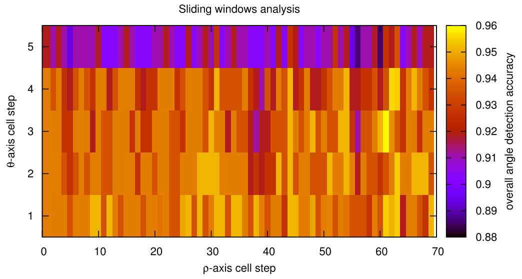

Fig. 7 Overall angle detection accuracy OA^{θ} achieved by our angle detection method for the *Rotated Barcode Database* using cells of size 3×61 that stride over the accumulator matrices, while varying the step values for the horizontal ρ-axis and the vertical θ-axis.

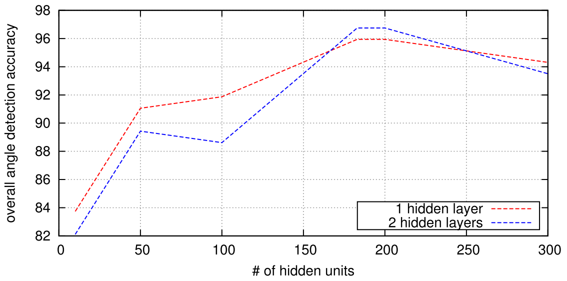

Fig. 8 Overall angle detection accuracy OA^{θ} achieved by the Multilayer Perceptron for the test set of Rotated Barcode Database, while varying both the number of hidden layers and the number of hidden units.

Bounding box detection evaluation. We evaluate the overall bounding box detection accuracy OA^{bb} obtained by the bounding box detection phase for the three datasets described in Section 4.1; the results are presented in **Table 1**. Unfortunately, we cannot provide any comparison with other barcode detection algorithms as they do not usually detect region of interests within the processed images; to our knowledge, the only method that performs a similar detection process is the one in [9], however we cannot use it to provide a comparison because its source code is not currently available. The bounding box detection accuracies we obtain are close to 85% OA^{bb} for all the analyzed datasets, this is a good result considering that our method does not impose constraints and requires no user interaction. On our mobile implementation, the completion time for the bounding box detection phase is 70 ms per 640×480 pixels image.

Modifying ZXing [19]. We perform a final experiment to prove that it is possible to improve the performance of an existing barcode reading algorithm by replacing its detection algorithm with our method. We decided to use the ZXing [19] algorithm because it is one of the most widely used open source multi-format barcode reading algorithms currently publicly available.

ZXing is a scanline based barcode reading algorithm that looks for the characteristic patterns of 1D barcodes by reading multiple rows of the processed images. The number of rows that are analysed when looking for barcodes is determined by the boolean *TRY_HARDER* parameter. When *TRY_HARDER* is *true*, ZXing has an increased chance of detecting barcodes within the processed images because it scans twice the lines it scans when *TRY HARDER* is *false*. As shown in [10], *TRY_HARDER* affects both the recall and the computational time of ZXing: for the *ArTe-Lab 1D Medium Barcode Dataset*, the recall of ZXing goes from 0.62 with *TRY_HARDER=false* to 0.82 with *TRY_HARDER=true*, while the computational time required to successfully decode an image increases by an order of magnitude when setting *TRY_HARDER=true*. Nonetheless, even when using *TRY_HARDER* and similarly to most of the other libraries for 1D barcode detection, in its original configuration, ZXing cannot detect rotated 1D barcodes because all the scanlines are horizontally traced. However, as shown in **Table 2**, by properly replacing the standard detection algorithm integrated in ZXing with one that exploits either the bounding boxes or the rotation angle identified by the proposed solution, the resulting algorithm becomes able to successfully decode arbitrary rotated 1D barcodes and therefore achieves significantly better results on all the three evaluated datasets.

Table 1 Overall bounding box detection accuracy OA^{bb} achieved by the proposed method for three 1D barcode datasets.

Dataset	OA^{bb}
ArTe-Lab 1D Dataset [10]	0.86
Muenster BarcodeDB [12]	0.83
Rotated Barcode Database	0.84

Table 2 Comparison between the overall barcode reading accuracy [10] achieved by the ZXing algorithm, two modified versions of ZXing that exploit our method (OurBB and OurAN) and the ZXing algorithm applied 6 times to 6 different rotation angles for every processed image (ZXing-6rot).

Dataset	Barcode Reading Algorithm			
	ZXing	OurBB	OurAN	ZXing-6rot
ArTe-Lab [10]	0.82	0.85	**0.95**	0.85
Muenster [12]	0.73	0.81	**0.93**	0.78
Rotated DB	0.61	0.82	**0.93**	0.77

More in details, in Table 2, we compare the results achieved by the original ZXing algorithm with the ones obtained by two modified versions of ZXing that exploit our detection method to decode rotated barcodes (OurBB and OurAN) and by those obtained by the classic ZXing algorithm applied 6 times to 6 different rotation angles (ZXing-6rot). The results of Table 2 have been measured using the same metric of Zamberletti et al. [10].

- ZXing has been evaluated in its *TRY_HARDER=true* configuration to obtain the best possible results for the three evaluated datasets. ZXing with *TRY_HARDER=true* requires on average 122 ms to decode an image from the ArTe-Lab 1D Dataset (roughly 1 ms per scanline).
- OurBB denotes a modified version of ZXing that exploits the bounding boxes detected by our method and traces 10 uniformly sampled scan lines, rotated by the angle detected by our method, over the components intersecting the detected bounding boxes in the original images. This modified ZXing requires roughly 280 ms to decode a single image from ArTe-Lab 1D Dataset: 200 ms for the Hough angle detection phase described in Section 3.1, 70 ms for the bounding box detection phase described in Section 3.2 and 10 ms to decode 10 scanlines using ZXing.
- OurAN denotes a modified version of ZXing that exploits the rotation angles identified by our method and traces scan lines rotated by those angles over the whole images (similarly to the classic ZXing algorithm). The time required to decode an image in this configuration is roughly 322 ms: 200 ms for the Hough angle detection phase of Section 3.1 and 122 ms for the execution of ZXing with *TRY_HARDER=true* over the whole image.
- ZXing-6rot denotes an algorithm obtained by applying 6 times the original ZXing algorithm to 6 different angles for every processed image. The average time required by ZXing-6rot to process an image is equal to 6 times the time required to process a single angle using ZXing: 732 ms.

OurAN obtains better results than OurBB since, as expected from the results presented in Table 1, some bounding boxes are not correctly detected by our bounding box detection method (Section 3.2), therefore the scan lines traced over the rows intersecting those bounding boxes in the original do not produce any results. On the other hand, OurAN exhaustively looks for barcodes rotated by the angles detected by our Hough based procedure (Section 3.1) within the whole image, this increases the chances to successfully decode 1D barcodes but also increases the computational time required to process an image (from 280 ms to 322 ms).

OurBB and OurAN achieve better overall performances than both ZXing and ZXing-6rot algorithms for all the three evaluated datasets. ZXing-6rot is also substantially slower than all the other evaluated algorithms and cannot overcome OurBB or OurAN, especially for the *Rotated Barcode Database*, because the assumption of Gallo and Manduchi [9] that 1D barcodes can be successfully decoded when a scanline is traced within ±30° from the true rotation angles of the barcodes does not hold true for deformed barcodes, e.g., the ones printed over non planar surfaces (typical of the *Rotated Barcode Database*).

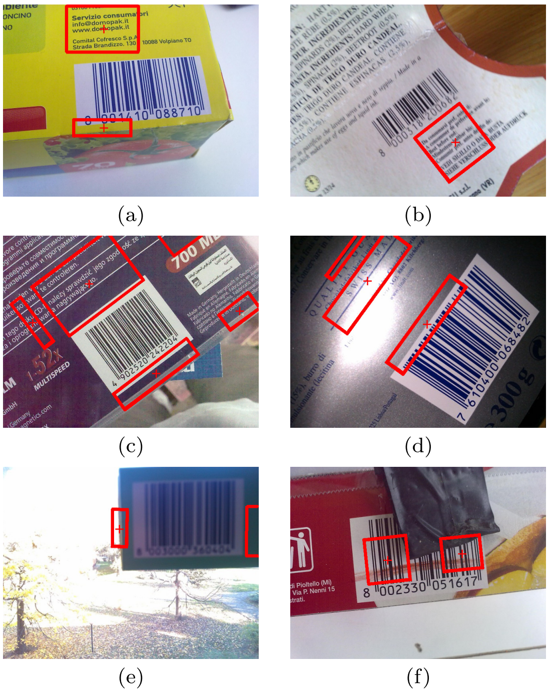

Fig. 9 Negative examples of bounding boxes identified by our method for some sample images extracted from three 1D barcode datasets. In some instances, the edges of text characters in the images are wrongly identified as the bars of a 1D barcode.

Failure examples. In **Fig. 9**, we present some negative examples of bounding boxes identified by the proposed method for different images extracted from the three datasets described in Section 4.1. It is possible to observe that: (i) the errors committed by the bounding box detection phase for the images (a) and (b) are caused by the vertical edges of text characters that resemble the bars of barcodes and share the same orientation of the real barcodes; in such situation, even though the rotation angles of the barcodes are correctly identified by the angle detection phase, the subsequent segment extraction, performed by the bounding box detection phase, wrongly extracts those characters' edges because they are stronger than the edges produced by the bars of the real barcodes; (ii) in the examples (c) and (d), the distance between the rotation angles detected by the MLP and the true angle values is 90°; as such, the output of the bounding box detection is flawed; (iii) the examples (e) and (f) highlight the limit of the proposed approach when detecting heavy occluded and unfocused 1D barcodes.

5. Conclusion

In this work, we have presented a simple method for detecting one-dimensional barcodes from camera captured images that requires no user interaction and is angle invariant.

We proved the effectiveness of the proposed approach using three EAN/UPC datasets, one of which has been specifically built to evaluate the performance of angle invariant barcode detection methods.

The obtained results prove that our method can be used to precisely detect both the rotation angles and the bounding boxes of one-dimensional barcodes even when such barcodes are partially occluded, warped or illegible due to reflections.

The time required by our approach to process an entire image is roughly 270 ms (200 ms for the Hough based angle detection phase described in Section 3.1 and 70 ms for the bounding box detection phase described in Section 3.2) on a modern mobile device; this is an acceptable result, as it is possible to obtain a robust one-dimensional barcode reading algorithm simply by coupling our approach with a fast scan line decoding algorithm that processes only the center of the detected bounding boxes.

References

[1] Shen, X., Lin, Z., Brandt, J. and Wu, Y.: Mobile Product Image Search by Automatic Query Object Extraction, *Proc. European Conference on Computer Vision* (*ECCV'12*), pp.114–127 (2012).

[2] Pavlidis, T., Swartz, J. and Wang, Y.: Fundamentals of bar code information theory, *Computer*, Vol.23, No.4, pp.74–86 (1990).

[3] Kato, Y., Deguchi, D., Takahashi, T., Ide, I. and Murase, H.: Low Resolution QR-Code Recognition by Applying Super-Resolution Using the Property of QR-Codes, *Proc. International Conference on Document Analysis and Recognition* (*ICDAR'11*), pp.992–996 (2011).

[4] A-Lin, H., Yuan, F. and Ying, G.: QR code image detection using run-length coding, *Proc. International Conference on Computer Science and Network Technology* (*ICCSNT'11*), pp.2130–2134 (2011).

[5] Szentandrási, I., Herout, A. and Dubská, M.: Fast detection and recognition of QR codes in high-resolution images, *Proc. Spring Conference on Computer Graphics* (*SCCG'12*), pp.129–136 (2012).

[6] Wang, K., Zhou, Y. and Wang, H.: 1D bar code reading on camera phones, *International Journal of Image and Graphics*, Vol.7, No.3, pp.529–550 (2007).

[7] Zhang, C., Wang, J., Han, S., Yi, M. and Zhang, Z.: Automatic real-time barcode localization in complex scenes, *Proc. International Conference on Image Processing* (*ICIP'06*), pp.497–500 (2006).

[8] Chai, D. and Hock, F.: Locating and decoding EAN-13 barcodes from images captured by digital cameras, *Proc. International Conference on Information, Communications and Signal Processing* (*ICICS'05*), pp.1595–1599 (2005).

[9] Gallo, O. and Manduchi, R.: Reading 1-D Barcodes with Mobile Phones Using Deformable Templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.33, No.9, pp.1834–1843 (2011).

[10] Zamberletti, A., Gallo, I., Carullo, M. and Binaghi, E.: Neural Image Restoration for Decoding 1-D Barcodes using Common Camera Phones, *Proc. International Conference on Computer Vision Theory and Applications* (*VISAPP'10*), pp.5–11 (2010).

[11] Chen, L.: A directed graphical model for linear barcode scanning from blurred images, *Proc. Asian Conference of Computer Vision* (*ACCV'12*), pp.524–535 (2012).

[12] Wachenfeld, S., Terlunen, S. and Jiang, X.: Robust 1-D Barcode Recognition on Camera Phones and Mobile Product Information Display, *Mobile Multimedia Processing*, Vol.5960, pp.53–69 (2010).

[13] CipherLab, available from ⟨http://www.cipherlab.com.au/⟩.

[14] Duda, R.O. and Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures, *Comm. ACM*, Vol.15, No.1, pp.11–15 (1972).

[15] Zamberletti, A., Gallo, I. and Albertini, S.: Robust Angle Invariant 1D Barcode Detection, *Proc. Asian Conference on Pattern Recognition* (*ACPR'13*), pp.160–164 (2013).

[16] Adelman, R., Langheinric, M. and Floerkemeier, C.: A toolkit for Barcode Recognition and Resolving on Camera Phones, *Proc. International Symposium on Mathematical Progress in Expressive Image Synthesis* (*MEIS'06*) (2006).

[17] Basaran, E., Uluçay, O. and Ertürk, S.: Reading Barcodes Using Digital Cameras, *Proc. International Symposium on Intelligent Manufacturing Systems* (*IMS'06*), pp.835–843 (2006).

[18] Canny, J.: A Computational Approach To Edge Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.8, No.6, pp.679–698 (1986).

[19] Zebra Crossing, available from ⟨https://github.com/zxing/zxing⟩.

[20] Muñiz, R., Junco, L. and Otero, A.: A robust software barcode reader using the Hough Transform, *Proc. International Conference on Information, Intelligence, Systems and Applications* (*IISA'99*), pp.313–319 (1999).

[21] Bradski, G.: The OpenCV Library, *Dr. Dobb's Journal of Software Tools* (2000).

[22] Bishop, C.M.: *Neural Networks for Pattern Recognition* (1995).

[23] Igel, C., Toussaint, M. and Weishui, W.: *Rprop using the natural gradient* (2005).

[24] Matasyz, J., Galambosy, C. and Kittler, J.: Progressive probabilistic Hough Transform for line detection, *Proc. Internation Conference on Computer Vision and Pattern Recognition* (*CVPR'99*) (1999).

Alessandro Zamberletti graduated from the University of Insubria, Varese, Italy (2011). He is currently a Ph.D. candidate in the Applied Recognition Technology research laboratory. His research interests include Object Recognition and Object Class Segmentation from natural scene images.

Ignazio Gallo received his degree in Computer Science at the University of Milan, Italy, in 1998. He became an assistant professor at University of Insubria, Varese in 2003. His research interest are Computer Vision, Image Processing, Pattern Recognition, Neural Computing.

Simone Albertini received his M.S. degree in Computer Science in 2011 and he is attending a Ph.D. course while collaborating with the Applied Recognition Technology Laboratory (Arte-Lab) of the University of Insubria. His research interests are Computer Vision, Pattern Recognition and Automatic Feature Learning.

Lucia Noce received her M.S. degree in 2012 and is currently a Ph.D. student in the Applied Recognition Technology Laboratory (Arte-lab) within the Department of Theoretical and Applied Science at the same University. Her research interests are in the areas of Data Mining and Computer Vision.

(Communicated by *Koichi Kise*)