Pattern Recognition Letters 34 (2013) 519–526

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

GAS meter reading from real world images using a multi-net system

Marco Vanetti *, Ignazio Gallo, Angelo Nodari

University of Insubria, Dipartimento di Scienze Teoriche e Applicate, Via Mazzini 5, 21100 Varese VA, Italy

ARTICLE INFO

Article history:
Received 11 May 2012
Available online 7 December 2012

Communicated by G. Borgefors

Keywords:
Object detection
Object segmentation
Text localization
Ocr
Neural networks
Multi-net system

ABSTRACT

We present a new approach for automatic gas meter reading from real world images. The gas meter reading is usually done on site by an operator and a picture is taken from a mobile device as proof of reading. Since the reading operation is prone to errors, the proof image is checked offline by another operator to confirm the reading. In this study, we present a method to support the validation process in order to reduce the human effort. Our approach is trained to detect and recognize the text of a particular area of interest. Firstly we detect the region of interest and segment the text contained using a method based on an ensemble of neural models. Then we perform an optical character recognition using a Support Vector Machine. We evaluated every step of our approach, as well as the overall assessment, showing that despite the complexity of the problem our method provide good results also when applied to degraded images and can therefore be used in real applications.

1. Introduction

In Europe, the existing automatic meter reading technology is not widespread (Deconinck, 2010), for this reason the gas meter reading is usually done on site by an operator and a picture is taken from a mobile device as reading proof. Since this operation is prone to errors, to confirm the reading, the same proof image is checked offline by another operator. The validation is expensive because it is time consuming and for this reason we propose a method to automatically read a gas meter using a single picture.

In literature there are several works developed for the location and recognition of text in the whole image (Jung et al., 2004). In (Shen et al., 2006) a segmentation method is used to detect edges, strokes, corners and other features in the image, following a bottom up approach. The combination of these features are used in order to detect the bounding boxes which surround the text within the image. In (Jung, 2001) a neural network-based texture discriminator is employed to locate the text area, while in (Chen et al., 2004) a set of Haar-like features, used to locate the text inside the image, are found using the AdaBoost algorithm. The localization of the text is followed by the character recognition, this last phase is performed using commercial OCR softwares (Chen et al., 2004). Lee et al. (2010) employs a modified K-means algorithm to select candidate text regions using texture, edge and color information, then each candidate region is verified using a Markov random field model. Pan et al. (2011) propose a three-stages pipeline, where the text area is detected using a histogram of oriented gradients, then a conditional random field is used to filter out the non-text components and finally the text components are grouped together with a learning-based energy minimization method. A very different approach is proposed in (Coates et al., 2011), where an unsupervised K-means based algorithm is used to learn a set of features from raw pixel intensities, then both the text localization and character recognition are performed using these features and a linear Support Vector Machine (SVM) classifier.

Like other specific applications that search for text located on specific objects, such as address block location (Jain, 1994) and license plate location (Gazcon et al., 2012), our approach is trained to detect the specific area of the image where it is possible to find the meter counter, and after that an ad hoc Optical Character Recognition (OCR) is performed.

The main problems in gas meter reading images are poor resolution due to a low-quality camera, blur, poor lighting, low contrast, reflections, shadows, occlusions, etc. Fig. 1(g) shows some problematic examples used in our experiments and it is evident that under these conditions, the detection of the meter counter becomes a very competitive task. Because of these problems and very high visual variability of the different meters models and brands, we could not rely on knowledge-based information derived from specific meter models to locate the counter.

Recently, neural networks have been successfully applied in many similar problems (Jung and Han, 2004, Jung, 2001). In this paper we used a set of supervised neural models to address and solve the many problems described above and hosted in the images of our domain. The approach we propose is summarized in Fig. 1(a)–(f) and can be divided in five main stages: firstly we

* Corresponding author.
E-mail addresses: marco.vanetti@uninsubria.it (M. Vanetti), ignazio.gallo@uninsubria.it (I. Gallo), angelo.nodari@uninsubria.it (A. Nodari).

0167-8655/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2012.11.014

Fig. 1. Optical character recognition example produced by our algorithm. (a) An input image. In (b) the detected region of interest obtained by our algorithm. In (c) the digits are separated from the background while in (d) all the digits and their bounding box have been detected. The relevant part of the digits is detected in (e) and classified in (f). In (g) some examples of problematic images from all the dataset used in this work.

detected the region of interest in which the meter counter appears and then we segmented the digits found in order to separate the characters from the background noise. For these two phases we used the same supervised neural algorithm summarized in Section 2.1. In the last three steps we carried out an accurate detection of relevant digits and then a classification using a Support Vector Machine (SVM).

2. The proposed method

The first two steps of the proposed method are based on our previous work (Nodari et al., 2011) where the automatic meter counter detection and digit segmentation of a gas meter are accomplished with the multi-net for object detection (MNOD) algorithm, which we first proposed in (Gallo and Nodari, 2011).

Since our previous digit segmentation algorithm may produce several artifacts, we need a robust method to locate accurately each digit. In this paper we then performed a Fourier analysis on a signal based on a local integration on the segmentation image in order to find phase and period of the digit sequence. In a post processing step, we selected the relevant digits to process in the final classification phase where a SVM is used.

2.1. The MNOD algorithm

MNOD is a multi-net system (Sharkey, 1999) which consists of an ensemble of supervised neural models able to detect an object in a cognitive manner, locating the object through the use of a segmentation process. This model is tolerant to many of the problems that afflict the images in a real scenario and it is characterized by a high generalization ability and good robustness (Gallo and Nodari, 2011).

The MNOD can be represented by a directed acyclic graph, composed by multiple source nodes, having no incoming edges, used as feature extractors, internal nodes which aggregate output of other nodes and a single terminal node, which has not outgoing edges, that provides the final segmentation map. Each node n is properly configured with its own parameters $\boldsymbol{P}$ and acts like an independent module $C_{\boldsymbol{P}}^{n}$ providing an output segmentation map. The process start from source nodes, which apply operators and filters on the input images in order to create their specific features, that enhance the input data peculiarities. Internal nodes process the maps provided by other input nodes first resizing them according to the parameter I_S. After the resizing, internal nodes generate the pattern vectors for the neural network using pixel values that fall within a sliding window of size W_S and give in output a map image where each pixel has an intensity value proportional to the probability it

belongs to the object. In the present work, each internal node consists in a feed-forward multi-layer perceptron (MLP) and is trained using the Resilient Backpropagation learning algorithm proposed by Riedmiller et al. (1993).

The particular aspect of this model lies in the connection between nodes, since the links between nodes in the structure define the flow of image segmentation process that cross the whole structure from the source nodes to the terminal node containing the final segmentation. Source nodes extract different information from the input images, as an example we used color channels and brightness (Br) for the luminance of a visual target. As suggested in (Dalal and Triggs, 2005), information on edges were extracted using 1-D $[-1, 0, 1]$ masks at $\sigma = 0$, obtaining the two source nodes: horizontal edges (HE_s) and vertical edges (VE_s), where s is the scale of the input image used as a resize factor to compute the feature.

The MNOD configuration was chosen considering which nodes in which layers gave better results, using a trial and error procedure. In particular, we used the following strategy to optimize the results: nodes belonging to the first layer are used to find areas of interest, and nodes of subsequent layers are used to eliminate the false positives and to confirm the true positives.

2.2. Counter detection

The counter is detected using the MNOD algorithm described in the previous section where each node is trained on a set of different points extracted from all training images. In general each training example t_i has the following structure $[\langle features\rangle; \langle expected values\rangle]$ where the *feature part* depends from the parameter W_s and the configuration of each internal node, while the *expected values* are the expected segmentation labels $o_{ij} \in \{0, 1\}$ of the pixels surrounding the training point t_i.

Since the output map produced by a trained MNOD presents lack of precision near the boundaries of the detected objects, it could miss some basic details or add unnecessary background information during the segmentation step. Then we performed a post-processing step in order to improve the segmentation using the fast watersheds algorithm (Vincent et al., 1991). The idea, behind this technique, consists of placing two sources of water inside and outside the boundary of each predicted object of interest. In particular, we first identify each blob in the MNOD output image and then we erode and dilate them to create a source of water inside and outside the object. In Fig. 2 there are three examples of post-processing through the use of the fast watersheds algorithm. In column (b) we can see the initialization of internal and external sources of water for each blob, while in columns (c) and (d) the segmentation results of the watersheds algorithm. Rows (1) and (2) of Fig. 2 represent the typical response of the detection system proposed while the row (3) provides an answer with false positive that occurs when the image acquisition is not optimal (for example the image (a3) in Fig. 2 is low-contrast and has a very low brightness).

In all situations where the system detects more than one area that could potentially contain a counter (see example in Fig. 2 – image (3d)), we select only one object (best-area A_D), looking for the best ratio between the width and height of the bounding box and selecting the area whose color histogram is closer to the histogram of a typical counter. These a priori knowledge was statistically inferred using the training dataset employed during the detection evaluation experiments.

2.3. Digit segmentation

As in the previous section, the digits within A_D are segmented using the MNOD algorithm described in Section 2.1. In order to train a MNOD's internal node for this sub-task we consider as *expected values* of each training example t_i the expected segmentation labels $o_{ij} \in \{0, 1\}$ of the manually segmented digits of interest. In this case, we assume a single type of source node in order to obtain a simple MNOD configuration. This is a valid assumption because the digit segmentation works only on the previous

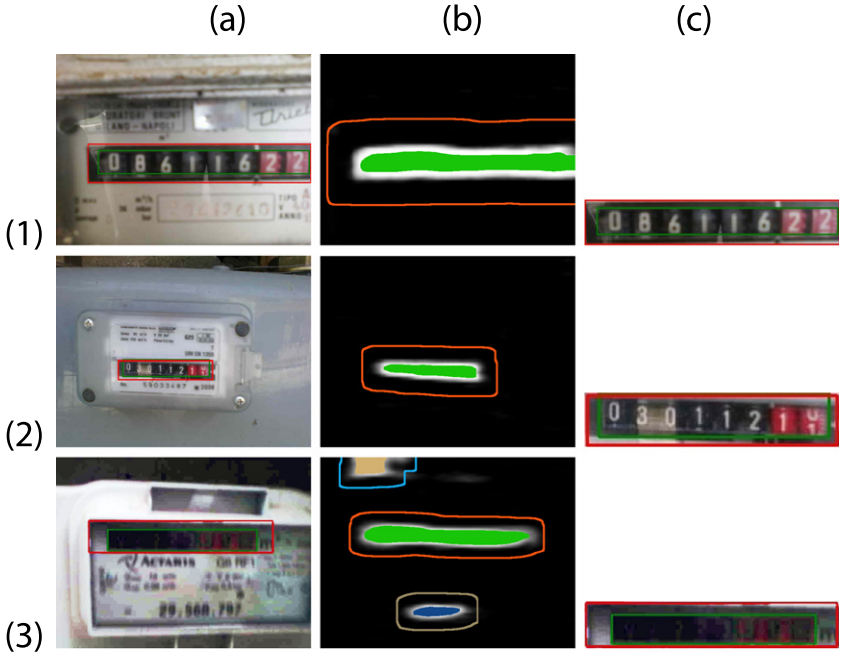


Fig. 2. Some examples of post-processed images of the counter detection phase when applied to some test images. In column (a) three input images with the expected bounding box in green and the bounding box found in red. In column (b) the fast watersheds initialization of internal and external sources of water for each blob. In columns (c) the final resulting areas of the Watersheds algorithm and the selected area A_D. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

detected area containing only white digits on a simplified background. The segmentation image A_S, as shown in Fig. 1(c), can be employed to significantly improve the OCR phase as described in the following sections.

2.4. Fourier analysis

Given A_S, we compute a signal $x(n)$ obtained by locally integrating the A_S image using a fixed sliding window size:

$$x(n) = \sum_{r=0}^{H} \sum_{c=0}^{W/100} S(r, n+c), \quad n = 0 \dots W \tag{1}$$

where W and H are respectively the width and the height expressed in pixels of A_S and n is the position of the sliding window with respect to the left margin. Sums in Eq. (1) can be efficiently computed using integral images. By applying a Fourier analysis on $x(n)$ we can extract the largest amplitude frequency component $x_d(n)$. We noticed that the horizontal coordinate of each digit location d_j in the image corresponds, or falls very close, to the maximum m_j of the periodic function $x_d(n)$ with period T. The region containing the counter digits can finally be divided into $D = W/T$ candidate regions R_j, centered on m_j and having width equal to T. Each R_j contains a potential digit and in Fig. 1(d) an example of candidate regions are marked in green.

2.5. Significant digits detection

A crucial step, during the digits detection, is to distinguish between decimals and integers. Typically the digits belonging to the fractional part are surrounded by a red area, whose shape may vary between counter models. This detail is important because from a legal standpoint, the decimal digits can be discarded in the reading process. Given $R_j \in A_D$, a region which surrounds the digit d_j, we first use the normalized red channel RN in order to extract a feature enhancing the red areas. Then we solve the following maximization:

$$P = \arg\max_{p \in [D/2, D]} \left| \frac{\sum_{j=1}^{p} Avg(R_j)}{p} - \frac{\sum_{j=p+1}^{D} Avg(R_j)}{D-p} \right| \tag{2}$$

where $Avg(R_j)$ is the average intensity value of the RN feature in the area R_j surrounding the digit d_j and D is the total number of digits found in the previous "Fourier analysis" phase. The first value found $p \in P$ represents the index of the last significant digit, then all the digits in $\{R_{p+1}, \dots, R_D\}$ can be discarded. Notice that in Eq. (2) we exploit the knowledge about the position of the not-significant digits that is they are always on the right side of the meter counter.

2.6. Accurate digit detection

Experiments in Section 3.5 show that, cropping each digit from the counter image using a fine bounding box around its borders, leads to a high quality digit classification. To obtain an accurate bounding box around each digit we first applied a threshold on each candidate image R_j and then we processed the resulting image with the Canny algorithm in order to obtain a set of contours (Suzuki and Abe, 1985). Among all the contours obtained in the previous step, our approach selects the contour C_j with the largest area and, in particular, the rectangle $\widehat{R}_j$ that contains C_j. The result of this procedure is shown in Fig. 1(d), where the the red rectangles are the selected $\widehat{R}_j$.

2.7. Digit classification

In this phase we used a SVM with radial basis function based kernel as a discriminative model to classify the sub-images containing each digit. Therefore, we have to address a supervised classification problem, with 10 classes, one for each digit from 0 to 9. Patterns used to train the SVM model are formed by cropping, from A_D, the digit area $\widehat{R}_j$ and stretching the obtained sub-image to a constant size $F \times F$ using linear interpolation. The feature vector is composed by concatenating the intensity values of the pixels within the sub-image, thus the dimension of the vector is F^2. As a further refinement, before extracting the feature vector, the sub-image can be processed using a Sobel filter in order to highlight the edges of the digits and discard areas with constant intensity.

3. Experiments

In this section we show the results of the experiments performed to evaluate each step and the overall reading strategy.

To assess the digit detection system, precision P and recall R have been used. Detections are considered true or false positives depending on the area which overlaps with the ground truth bounding boxes. Let B_p the predicted bounding box and B_{gt} the ground truth bounding box, we compute the Jaccard index J according to the following formula:

$$J(B_p, B_{gt}) = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{3}$$

and in order to be considered a correct detection, the Jaccard index must be greater than 0.5.

A measure that is used to summarize P and R values is the F-measure (Makhoul et al., 1999) $F = 1/(\lambda P + (1-\lambda)R)$ where in our experiments $\lambda = 0.5$, which is the standard value for this harmonic mean of precision and recall.

In order to evaluate the segmentation step, intersection/union between truth and predicted foreground pixels is computed:

$$Acc. = tp/(fn + fp + tp) \tag{4}$$

where tp is the number of correctly classified foreground pixels, fn is the number of foreground pixels classified as background and fp are the number of background pixels classified as foreground. This is a common metric used to evaluate segmentation algorithms because penalizes both over- and under-estimates predictions. For more details on these measures and how an object is considered correctly detected or segmented see Everingham et al. (2005).

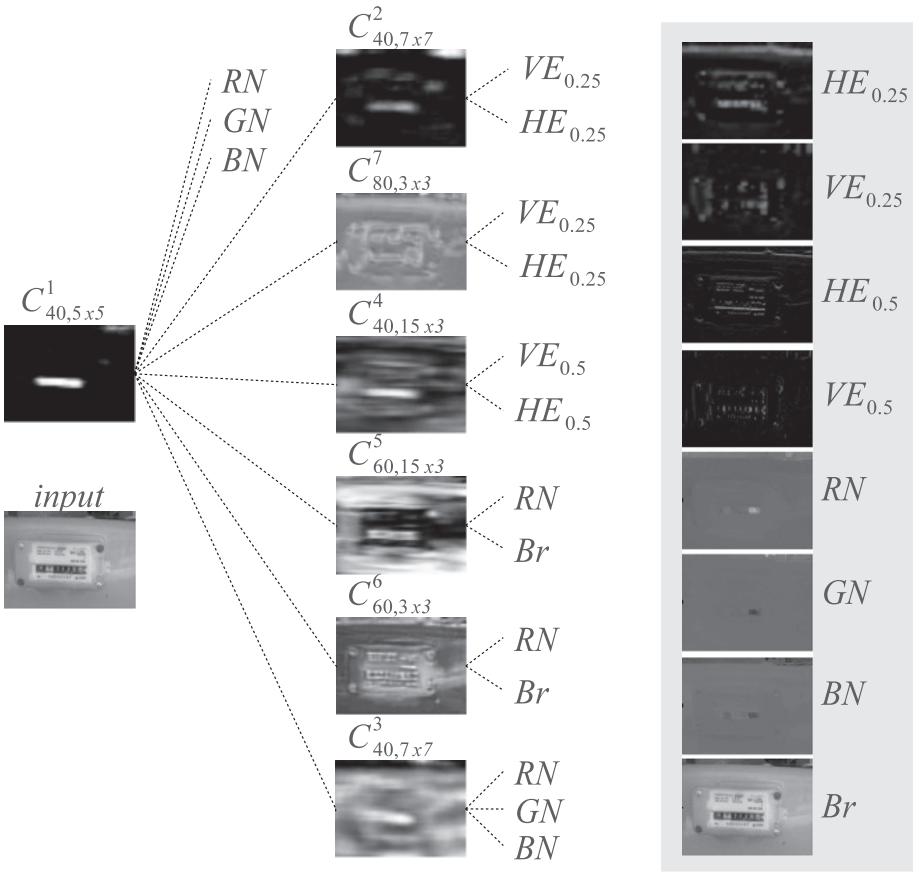
For our experiments we made four different datasets that will be described in the following sections and are available online[1] in order to facilitate the comparison with future works. For all these datasets the division between training and test sets is done once and in a random way and the sets are fixed for all the experiments.

3.1. Counter detection

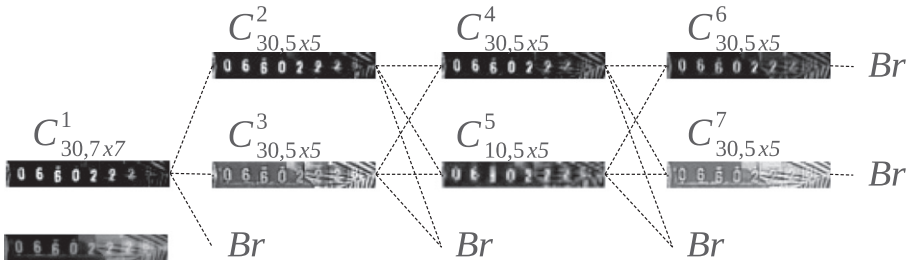
The dataset used for the evaluation of this step is called *gas-meter-counter* and consists of 102 training images and 51 testing images.

The MNOD configuration used in this experiment is shown in Fig. 3(a). Seven different MLP neural networks $C^n_{I_S,W_S}$ were used, configured with the following set of predefined source nodes: horizontal and vertical edges $\{HE, VE\}$, normalized color channels $\{RN, GN, BN\}$ and normalized red channel and brightness $\{RN, Br\}$. The individual nodes C^2–C^7 were selected and trained

[1] <http://artelab.dicom.uninsubria.it/download>.



(a) Meter counter detection



(b) Digit segmentation

Fig. 3. In (a) the MNOD configuration used for the meter detection step. The image shows the activation map produced by each node of the MNOD configuration when a test image is presented in input. In (b) the configuration used for the digit segmentation step. Each node $C^n_{l_S,W_S}$ is representative of a neural network trained on the training set.

using the AdaBoost algorithm Freund et al. (1995) starting from a preconfigured set of nodes (weak learners). The internal nodes C^2–C^7 use the features extracted from all the input images through the source nodes, while node C^1 aggregates the output of all the internal nodes and a few of source nodes.

The results shown in Table 1 demonstrates how, applying the Watershed algorithm and the rule for the best area selection, we obtain a significant increase in terms of precision and recall.

Table 1
Detection results obtained on the *gas-meter-counter* dataset comparing the MNOD result, the post-processing fast watersheds algorithm applied to the MNOD output and the application of the decision rule for the best area selection. The table on the right shows the segmentation results computed on the *segmented-gas-meter-counter* dataset.

Detection	MNOD (%)	Watersheds (%)	Best-Area (%)	Segmentation	MNOD
Precision	3400	5900	9600	Acc. bg	92.8
Recall	7800	9200	9600	Acc. fg	49.5
F-measure	4700	7200	9600	Avg Acc.	71.6

3.2. Digit segmentation

For the digit segmentation problem a second dataset called *segmented-gas-meter-counter* was built and consists of 62 training and 32 testing images, cropped from the *gas-meter-counter* dataset.

The MNOD configuration, used in this experiment, is composed by seven different neural networks and is shown in Fig. 3(b). Nodes C^7 and C^6 identify the areas of the image containing the

digits, while all the other nodes improve the segmentation for each digit.

The segmentation results showed in Table 1 highlight an accuracy of the test set close to 50% while the high accuracy of background detection assures that many problems due to noise present in the proximity of the digits have been removed.

3.3. Fourier analysis robustness

To test the robustness in the identification of the digits using the Fourier analysis proposed in Section 2.4, we performed an experiment to study the detection accuracy varying the noise in the image. We iteratively introduced an artificial noise into the images A_S taken from *segmented-gas-meter-counter* dataset in order to emulate the typical noise suffered by a real image. The noise was added by drawing numbers, lines and other geometrical elements using random sizes and locations. The amount of noise introduced was quantified as the ratio between the number of pixels changed on the total number of the pixels of A_S. Having the truth bounding box B_{gt} for each digit, and given the regions R_j defined in Section 2.4, the accuracy was estimated by checking how many regions B_{gt} fall inside each region R_j. A region R_j is considered correct if it contains only one B_{gt}. In this way it is possible to count how many regions are correctly identified on the total number of ground truth digits. The results of this experiment are shown in the Fig. 4. At the same time we have exasperated this process to test the limits of applicability of the proposed method according to the amount of noise introduced. A noise higher than 10% in the region containing the digits is very unlikely in practice and it is also difficult to read for a human being. As showed in Fig. 4, within this threshold there is a detection accuracy higher than 90%, as a remark of the robustness of our approach.

3.4. Significant digits detection

Here we tested the significant digits detection strategy presented in Section 2.5 in order to analyze its accuracy. Within the *gas-meter-counter* dataset, we manually labeled the area belonging to the non-significant digits. Results showed that the proposed strategy is able to recognize correctly the position P of the last significant digit in the 91.86% of all the images of the dataset used.

3.5. Accurate digit detection

Results of the accurate digit detection (ADD) phase strongly depend on the A_S image, used directly to estimate the rectangle in which the digit is inscribed. To assess the accuracy of the ADD phase we used the metric introduced in Eq. (3) where B_p is the rectangle found by the ADD method and B_{gt} is the ground truth rectangle. Employing the *segmented-gas-meter-counter* dataset, where the ground truth rectangles have been derived from the ground truth segmentation masks, we obtained an accuracy of 75.4% where the errors are almost exclusively from segmentation faults, since that the ADD phase itself is based on a process that is very stable and does not exploit supervised learning.

3.6. Digit classification

We collected a manually annotated dataset to train and test the digit classification phase. We experimented two different strategies to extract digit patterns from an image, the first one with no constraints on the proportions of the digit inside the pattern (see *coarse BB* in Fig. 5(a)) and the second one requiring that the digit is scaled in order to fill the whole pattern area (see *Fine BB* in Fig. 5(b)). The results showed an increase of the Kappa in classification (Cohen, 1960) from 0.2 to 0.4, using the *Fine BB* instead of the *coarse BB* and varying the stretching factor F. This convinced us to develop the accurate digit detection phase in order to have a significantly more accurate classification. Table 2 shows the results just discussed, notice that the best stretching factor is $F = 20$ pixels and how, using the Sobel operator, the classification performance is further improved.

We compared our method with Tesseract (Smith et al., 2007), an algorithm that represents the state of the art in OCR, to address the problem of digits classification. Tesseract is an algorithm which is focused on reading printed documents and has many features regarding the formatting of text, paragraphs, fonts and support dozens of different languages, but can not be directly used to read text from natural images because this is still an open problem (Jung et al., 2004). This OCR algorithm was configured to recognize 11 classes of characters, in particular the digits from 0 to 9 and the "not a number" class. In a first experiment, using the default pretrained version of Tesseract, on the 100 examples taken from the *segmented-gas-meter-counter* dataset, we obtained an accuracy of 44.27%. Subsequently we conducted another experiment re-training the Tesseract algorithm using the *segmented-gas-meter-counter* dataset obtaining an accuracy of 50.18% a result much lower than the result obtained by our algorithm. This result shows the complexity of the digit classification task in this context, highlighting that standard OCR algorithms are not suitable for the problem faced in this study. This is due to the extreme variability of the digit font type, dimension and resolution, therefore it is necessary an ad hoc solution as proposed in this study.

3.7. Overall evaluation

To evaluate the accuracy of the overall solution we set up a test involving all the phases described in the previous sections. They

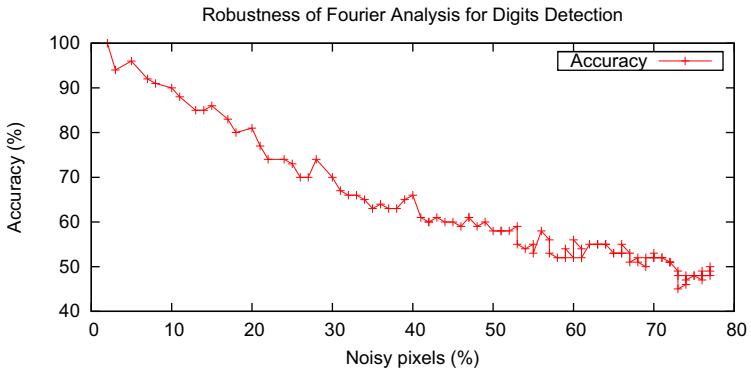
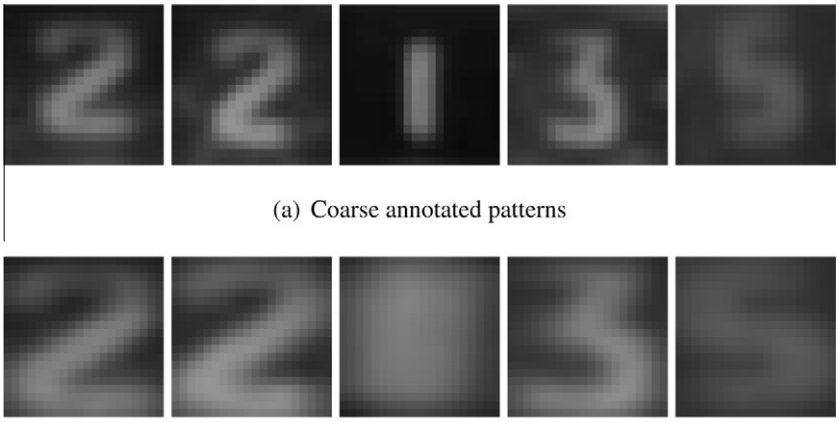


Fig. 4. Percentage of correctly identified digits' regions using the Fourier analysis related with the percentage of artificially introduced noise



(a) Coarse annotated patterns

(b) Fine annotated patterns

Fig. 5. Sample patterns used to train and test our OCR model. In (a) the bounding boxes locating the digits were roughly placed during the manual annotation. In (b) the digit is stretched in order to fills all the available space.

Table 2
Digit classification results in terms of Kappa value, using coarse bounding boxes (coarse BB) and fine bounding boxes (fine BB) varying the pattern size. Simple pixel intensities (PI) and Sobel operator aggregation strategies were used to construct a pattern. The best result is in bold.

Pattern size ($F \times F$ pixels)	Coarse BB		Fine BB	
	PI	Sobel	PI	Sobel
5×5	0.52	0.52	0.93	0.91
10×10	0.61	0.77	0.97	0.97
15×15	0.61	0.76	0.95	0.97
20×20	0.61	0.77	0.96	**0.98**
25×25	0.61	0.75	0.97	0.97
30×30	0.61	0.76	0.97	0.97
35×35	0.59	0.72	0.96	0.97

exploit a supervised learning method which is trained using the respective training datasets discussed in the previous sections.

The obtained overall accuracy is 87%, computed using a new dataset, called *meter-integration*, containing 100 meter counter images not present in all the other datasets, associated with hand-annotated meter readings. This value is grater than the values obtained in the previous steps because, for each step, the used measure refers to the specific goal performed by that step and shows how accurately a specific algorithm plays its task. As an example to measure the segmentation accuracy we used a pixel-based metric which is very restrictive in the error commissions. As a result, paradoxically, only the error of the segmentation step is greater than the total error of the entire algorithm, because in the overall evaluation phase an automatic reading is considered correct only if the value read by our method and the ground truth is the same.

For completeness, a similar experiment was performed using the Tesseract OCR framework, configured as described in Section 3.6 on the A_D, but the method was not able to correctly recognize any reading.

The total time required for an automatic reading is 844.53 ms and in particular counter detection takes 478.73 ms, digit segmentation 315.45 ms, digit detection 44.34 ms and digit classification 6.01 ms. The most time consuming phase is the counter detection, because the size of the input images (640 px $\times$ 480 px) are generally very large compared to the area detected on which all subsequent steps are computed. The tests were performed using a single thread C# code, on an Intel® Core™i5 2410 M processor @2.30 Ghz.

4. Conclusion

The localization and recognition of the digits in a gas meter counter is a difficult task due to the quality of images taken by mobile devices. The proposed model, based on an ensemble of neural networks, is able to detect the area of interest and to recognize every single digit in an average of 0.8 s per image. The Fourier analysis applied to the segmented image, in order to find phase and period of the sequence of digits, is a very robust approach to avoid false positives that are common in this context. Finally, the SVM used to classify each single digit has been shown to be an excellent and accurate solution.

For these reasons the proposed method can be applied in a real application context, but being the reading of text in natural images still an open problem, our solution can be applied in support and to validate the manually reading done by an operator.

We are now conducting an experimental phase in a real scenario where our solution is applied to a large database of images.

As future work, we are working to extend the proposed method in order to read even water meters and other useful information such as the model of the counter and the ID code of the meter. Another interesting future development is the unification of accurate digit detection and digit classification in a single phase, using a more powerful descriptor, like statistical or topological features, applied to the coarse bounding box.

Acknowledgements

We want to thank TECNA SA[2] for supporting us with the datasets used in this work.

References

Deconinck, G., 2010. Metering, intelligent enough for smart grids? Secur. Electric. Suppl. Cyber Age 15, 143–157.
Jung, K., Kim, K.I., Jain, A.K., 2004. Text information extraction in images and video: A survey. Pattern Recognition 37 (5), 977–997.
Shen, H., Coughlan, J., 2006. Finding text in natural scenes by figure-ground segmentation. In: Internat. Conf. on Pattern Recognition, pp. 113–118.
Jung, K., 2001. Neural network-based text location in color images. Pattern Recognition Lett. 22, 1503–1515.
Chen, X., Yuille, A.L., 2004. Detecting and reading text in natural scenes. In: Proc. of the 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition 2004, CVPR 2004, pp. 366–373.

[2] <http://www.tecnasolutions.com>.

Lee, S., Cho, M.S., Jung, K., Kim, J.H., 2010. Scene text extraction with edge constraint and text collinearity. In: Internat. Conf. on Pattern Recognition, pp. 3983–3986.

Pan, Y.-F., Hou, X., Liu, C.-L., 2011. A hybrid approach to detect and localize texts in natural scene images. IEEE Trans. Image Process. 20, 800–813.

Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D.J., Ng, A.Y., 2011. Text detection and character recognition in scene images with unsupervised feature learning. In: ICDAR, pp. 440–445.

Jain, A., 1994. Address block location using color and texture analysis. Comput. Vis. Image Understand. 60, 179–190.

Gazcon, N.F., Chesnevar, C.I., Castro, S.M., 2012. Automatic vehicle identification for argentinean license plates using intelligent template matching. Pattern Recognition Lett. 33 (9), 1066–1074.

Jung, K., Han, J., 2004. Hybrid approach to efficient text extraction in complex color images. Pattern Recognition Lett. 25 (6), 679–699.

Nodari, A., Gallo, I., 2011. A multi-neural network approach to image detection and segmentation of gas meter counter. In: 12th IAPR Conf. on Machine Vision Applications, pp. 239–242.

Gallo, I., Nodari, A., 2011. Learning object detection using multiple neural netwoks. In: VISAPP. INSTICC Press, vol. 2011, pp. 131–136.

Sharkey, A.J. (Ed.), 1999. Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, 1st ed. Springer-Verlag, New York, Inc., Secaucus, NJ, USA.

Riedmiller, M. Braun, H., 1993. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE Internat. Conf. on Neural Networks, pp. 586–591.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Proc. CVPR, pp. 886–893.

Vincent, L., Soille, P., 1991. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Machine Intell., 583–598.

Suzuki, S., Abe, K., 1985. Topological structural analysis of digitized binary images by border following. Graph. Models/graph. Models Image Process./ Comput. Vis. Graph. Image Process. 30, 32–46.

Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R., 1999. Performance measures for information extraction.

Everingham et al., M., 2005. The 2005 pascal visual object classes challenge. In: Selected Proceedings of the First PASCAL Challenges Workshop.

Freund, Y., Schapire, R.E., 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In: Proc. Second European Conf. on Comput. Learn Theor. EuroCOLT '95. Springer-Verlag, London, UK, pp. 23–37.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educat. Psychol. Measur. 20 (1), 37–46.

Smith, R., 2007. An overview of the tesseract ocr engine. In: Proc. of the Ninth Internat. Conf. on Document Analysis and Recognition. IEEE Computer Society, Washington, DC, USA, pp. 629–633, vol. 2.