Reading Meter Numbers in the Wild

Alessandro Calefati[1], Ignazio Gallo[1], and Shah Nawaz[1]

[1]Department of Theoretical and Applied Science, University of Insubria, Varese, Italy
{a.calefati,ignazio.gallo,snawaz}@uninsubria.it

***Abstract*—In this work we introduce a pipeline to detect and recognize various utility meter numbers in the wild. The system leverages on deep neural networks for detection and recognition. In the detection phase, we employ a fully Convolutional Neural Network to perform a pixel-wise classification, while the recognition phase employs another deep neural network to predict the length and individual digits in a meter. We qualitatively showed that the proposed approach is robust against severe perspective distortions, different lighting conditions and blurred images. Furthermore, it is capable of detecting small scale digits. Our approach is suitable for billing companies aiming to increase efficiency, lowering the time consumed by manual checks performed in the billing process. Finally, we release the dataset used in this work to benchmark the task.**

I. Introduction

In recent years, deep neural networks [1], [2], [3], [4], [5], [6] have replaced traditional Optical Character Recognition based methods for text detection and recognition. The strength of deep neural networks lies in unifying localization, segmentation and recognition steps [1], [2], [3]. Similarly, recognizing multi-digit numbers in images is considered a more specific task in text recognition. It is challenging due to the considerable variability in the visual appearance of numbers in the wild on account of a large range of fonts, colors, styles, orientations and arrangements. In addition, environmental factors such as lighting, shadows, glares and occlusions further complicate the automatic multi-digit numbers detection and recognition tasks. Recently, recognizing multi-digit numbers in images has gained considerable attention [1], [3], [4]. Some random examples of multi-digit numbers are shown in Fig. 1. These images are taken from various meter typologies such as electricity, gas and water.

In this paper, we focus on detecting and recognizing multi-digit numbers from various utility meter typologies. Although this specific task reduces the set of characters to be recognized, the complexities associated with text recognition in natural images remain. Furthermore, we observed that, in the literature, there is no standard benchmark dataset available to evaluate strategies on such a task. We found out that privacy is a major concern in releasing such datasets because typically meter images contain information related to customers. To overcome the lack of a standard dataset, we are releasing a "cropped" version in an attempt to benchmark such a task. We called it "cropped" because it contains only meters numbers without customer information.
Main contributions of this work are:

Fig. 1: Random images extracted from the dataset. Note that meter typologies are different, ranging from mechanical gas meters to digital electricity meters.

- a pipeline to detect and recognize multi-digit numbers in various meter typologies.
- a "cropped" version of the dataset, containing images of various meter typologies to benchmark multi-digit reading task, without customer information.

The rest of the paper is structured as follows: we explore the related literature in Section II; details of the proposed approach are discussed in Section III; dataset used are described in Section IV while Section V reports settings used to perform experiments and corresponding results. Finally conclusion are in Section VI.

II. Related work

Text detection in natural images has been tackled in few papers [2], [3], [7], [8]. All these attempts differ for task addressed or models used for detection and recognition. Previous works in the literature use traditional techniques such as Histogram of Gradient (HOG) features to perform text detection. Minetto *et al.* [9] proposed T-HOG: a HOG-based texture descriptor that uses a partition of an image to detect a single line of text. This approach, however, suffers from orientation issues. Given that HOG deals with lines, texts in several orientations become a problem in this approach. Boran *et al.* [8] adopted a more traditional approach, with the joint use of HOG features and Support Vector Machine [10] to detect Chinese words from images. Many works employ Maximally Stable Extremal Region (MSER) to perform text detection [11], [12].

Dai *et al.* [13] uses the traditional two-stage object detection strategy that consists of region proposal extraction followed by region classification. The problem of these approaches is that the region proposal step has a high cardinality, forcing a strategy that removes false positives.

With the rise of Convolutional Neural Networks (CNNs) and their stunning results, we opt to perform the detection and recognition using deep models. In [2] authors developed a pipeline for text detection and recognition from the natural scene using deep models. The purpose of our work is similar; however, we are focusing on a more specific task: multi-digit meter numbers detection and recognition on various meter typologies. Next, for the recognition, we use a model similar to the proposed by Goodfellow *et al.* [1]. Similarly, Gómez *et al.*[3] proposed an end-to-end CNN to predict numbers in a meter. Although the approach obtained promising results, it suffers from severe perspective distortions. In our approach, after the detection phase, we apply image transformations to align it in a horizontal position, making it possible to deal with distortions. After this step, we apply the classification model to obtain predicted values. Moreover, our model recognizes digits from various meter typologies, unlikely from [3], where authors worked solely with mechanical gas meters.

III. Proposed approach

The proposed approach is graphically represented in Fig. 2 and it is split into two phases: detection and recognition.

A. Detection phase

The detection phase is carried out with the model proposed in [14]. This phase in labelled in Fig. 2 with name "CNN-1". This model takes the original image resized to 224×224 pixels and produces correspondingly-sized output image with the inference. The training set contains pairs of images: the original one and the ground truth with pixel values in $\{0, 1\}$, where 0 indicates background and 1 tags numbers. After the inference stage, we crop the area of interest from the original image. Furthermore, we rotate it to obtain a horizontally aligned image. We apply the following strategy starting from the output image obtained with the first model (Fig. 2-b):

- contours extraction (curves joining continuous points along the boundary with same color or intensity);
- selection the best contour, containing the area of interest. We keep the region with the highest ratio $r_i = min(w_i, h_i)/max(w_i, h_i)$, where w_i and h_i are the width and height of the i-th rotated rectangle containing the contour for each of the proposed areas of interest;
- computation of the tilt angle to obtain an horizontally aligned image from the selected area of interest;
- dilation of width and height of the selected area with an increment $d = 0.3 \cdot min(w, h)$ (see the example in Fig. 2-c and all the examples in Fig. 3);
- crop of the image using the expanded area;

B. Recognition phase

The recognition phase is based on a deep neural network that takes as input a meter image and is capable of producing the actual meter reading as output, as shown in Fig. 2 labelled with "CNN-2". The network is composed of a convolutional backbone of 8 blocks, each made up by a convolutional layer with Rectified Linear Units (ReLU) as activation function, a max pooling layer and a batch normalization layer. The number of convolutional filters is 48, 64, 128, 160, 192, 192, 192 for convolutional layers, while the kernel size is 5×5 for all of them. All pooling layers use a kernel of 2×2 and alternate stride 2 and 1. After the convolutional part we stack 2 fully connected layers, followed with 6 small fully connected layers where the first one has only 7 neurons to predict the labels length $(0, \ldots, 5$, and more than $5)$, while others have 11 neurons representing the digit variables with 11 possible values. Finally, outputs of each fully connected layer are treated as a typical classification and trained using the Cross-Entropy loss function. The final loss function is defines as:

$$\sum_{j=1}^{K} H_{y'_j}(y_j) = \sum_{j=1}^{K} \left(-\sum_{i} y'_{ji} \log(y_{ji}) \right) \tag{1}$$

where K represents the number of digits to predict, y is the predicted probability distribution and y' is the true distribution (the one-hot vector with the digit labels). The network architecture with its main parameters is shown in Fig. 4.

Once the data is ready, we apply the classifier to get digits from images. This model is trained with manually assigned labels. For example, if an image contains the value "$00040,87$", the label would be "40". The model has the ability to discard the decimal and leading zeros parts during training and testing phases. We observed that the majority of numbers in meter images does not exceed more than 5 digits, so we set the maximum length to 5 thus $K = 6$. The last fully connected layer is made up of two different kinds of output neuron: one for predicting the length of the digit in the image and the other one predicting the output values. Possible values for the length prediction are 7: values from 0 to 5 and one output indicating a length of "more than 5". Similarly the output neurons for recognizing digits have 11 possible values, from 0 to 9 plus a blank character.

IV. Dataset

The dataset used in this work is obtained from private multi-utility companies. We cannot release the original data because of privacy concerns. However, we are publishing a version which does not contain customer information. We create various splits in the dataset: one for detection and one for the recognition phase. A third split test the whole pipeline.

The dataset for the detection phase contains 4566 labelled images for the train and 3365 for the test. The split consists of image pairs: RGB input image and a mask with pixel values in $\{0, 1\}$ to represent background and area of interest, respectively. Fig. 5 shows three examples of detection used as ground truth. From the same figure, we can see that selected

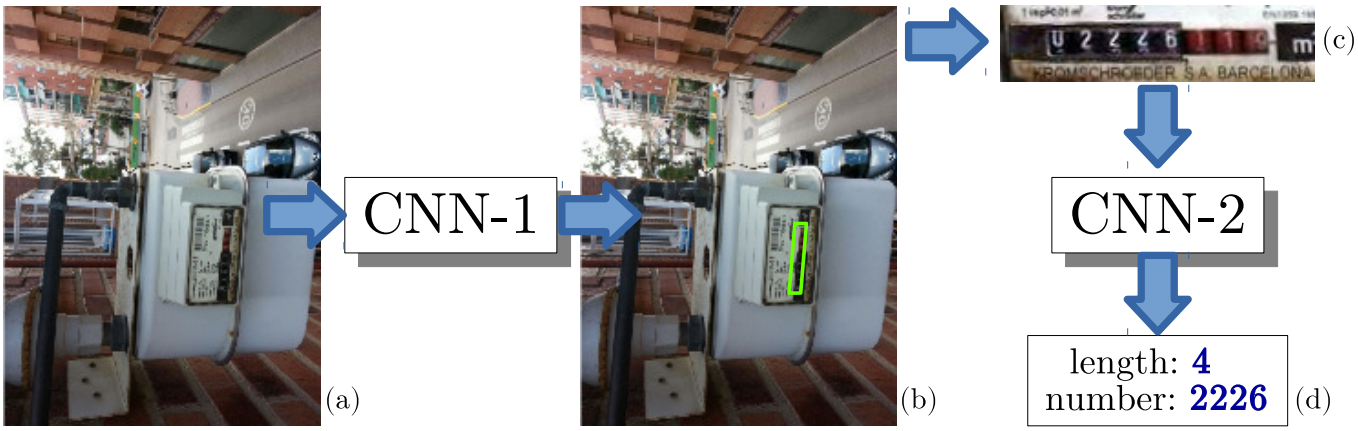


Fig. 2: Graphical representation of the proposed pipeline. On the left we have the CNN used for detection. It receives as input an image (a) and gives as output an image map with the area of interest (b). On the right, the second CNN performs the reading starting from the cropped and rotated detected region (c), producing the final reading containing length and predicted numbers (d).

Fig. 3: Cropped samples extracted from the dataset before the recognition phase.

areas of interest do not have well-defined boundaries. Thus shapes and size can vary based on the annotator. We called this dataset split *meter detection*.

The recognition split consists of 30240 images divided into 24195 and 6045 for train and test sets, respectively. We call it *meter recognition*. It contains pairs of input RGB images and integer numbers to represent meter values. In this split, images are different for each utility meter typology. This makes the task even more challenging because images of the mechanical gas meter are visually different from digital electricity meter. In addition, we observed that countries have different formats of utility meters. For example, European meters are different from South Asian meters.

Finally, we build another set of 15452 images called *test split* to test the full pipeline. For fair comparisons, the sets mentioned above are disjoint. This dataset is challenging because it contains images for various meter typologies. Furthermore, the dataset has images with scale variations, strong glares etc. (See examples in Fig. 6 and 7).

V. Experiments

We perform a series of experiments to show the effectiveness of the proposed approach. These consist of evaluating detection and recognition phases independently. In addition, we evaluate the entire pipeline.

A. Implementation details

The hyperparameters for the detection model are fairly standard. We train the model for 100000 iterations with a batch size of 8 and learning rate set to $1e^{-4}$. Similarly, for the recognition phase, we use a batch size of 32 and learning rate set to $1e^{-2}$. During the training of the recognition model, we do not set a maximum number of iterations. However, we use patience to stop the training process, and we set this value to 100.

B. Experiment on detection phase

In the first experiment, we use images from the detection split. Feeding an input image to the model, we get the mask containing labels of the area of interest. Then, we employ the mask to calculate the Intersection over Union (IoU) metric, defined in Eq. 2:

$$IoU = \frac{AO}{AU} \tag{2}$$

where *AO* is the area of the overlapping rectangle between predicted and ground truth, while *AU* is the area of union rectangle between predicted and ground truth.

Results on detection phase are shown in Table I. Note that, we calculate IoU for various thresholds ranging from 0.1 (Poor) to 0.9 (Excellent). We found out that a value of 0.4 (Good) produces an accuracy of 91.06%. However, increasing this value leads to lower results. We observed that such situation arises from different ways of annotating the data (see examples available in Fig.5).

C. Experiment on recognition phase

In the second experiment, we use images from the recognition split. Note that images used in this experiment contain only the area of interest, previously detected from the model. Sample images are shown in Fig. 3. We compute the overall accuracy considering a prediction right only if numbers inside

TABLE I: IoU evaluation with thresholds ranging from 0.1 (Poor) to 0.9 (Excellent) with step size of 0.1.

IoU Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy	99.37	99.16	98.35	91.06	69.87	40.41	17.42	2.63	0.002

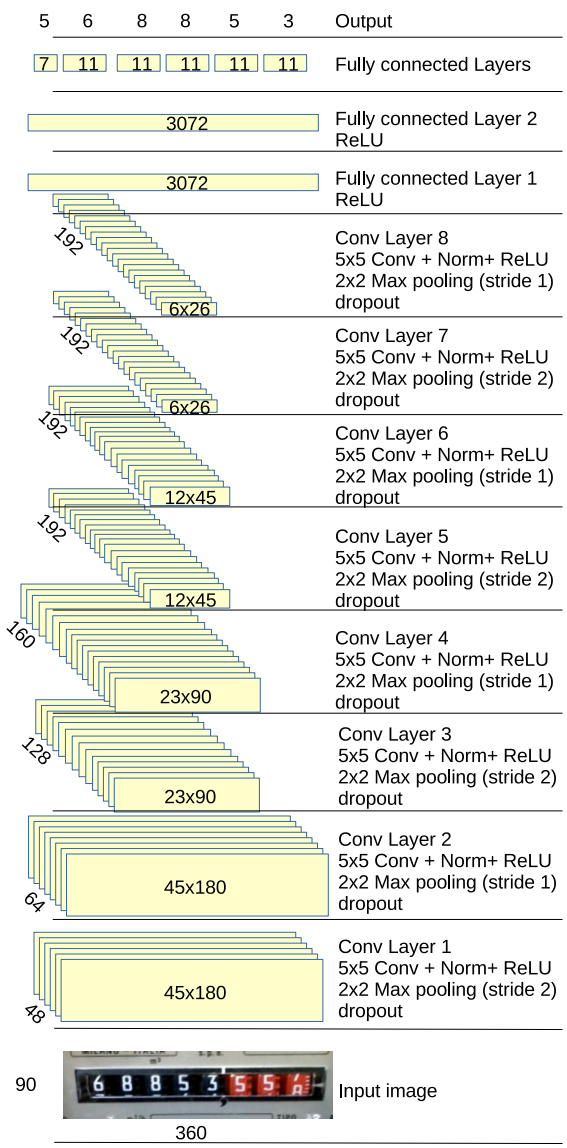


Fig. 4: The CNN architecture used to recognize numbers starting from the output crop of CNN-1. This model automatically predicts the length (5 in this example) and the values of each digits to compose the number (68853 in this example), without any segmentation step.

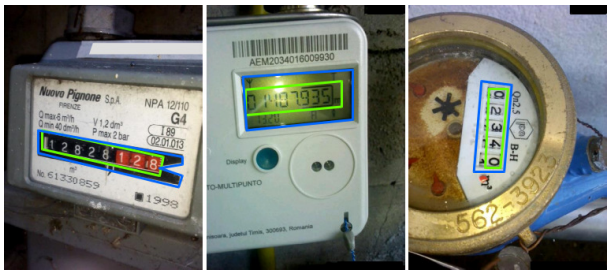


Fig. 5: Examples of images with superimposed ground truth (area with green borders) and the relative prediction of the CNN-1 model (area with blue borders). These examples highlight the lack of a clear boundary in the area of interest and consequently the dependency of the ground truth from the annotator.

Fig. 6: Examples of scale variation images from the dataset.

the image are classified correctly. In addition, we calculate the accuracy for each number position in the image. In Table II, we calculated the accuracy performance for each position of the digits in the meters. Furthermore, we calculated the overall accuracy for the recognition phase is 82.70% over the entire meter values. We obtain this value from already cropped (6045) test images available in the recognition split.

D. End-to-end experiment

In the third experiment we evaluate the entire pipeline. First, we apply the detection phase to get the area of interest and then we feed it to the recognition phase to obtain the final prediction.

Our pipeline achieves promising results considering that

Fig. 7: Examples of images with strong glare from the dataset.

TABLE II: Accuracy computed on recognition phase.

Phase	1st	2nd	3rd	4th	5th	Acc.
Recognition	94.60	89.22	92.76	95.10	96.60	82.70

TABLE III: Accuracy computed on the full pipeline.

Phase	1st	2nd	3rd	4th	5th	Acc.
Pipeline	92.94	88.99	90.95	92.24	92.92	85.60

Fig. 8: Random examples of upside down images from the dataset.

it recognizes numbers from three different meter typologies. Furthermore, the pipeline is capable to produce numbers from mechanical and digital meters. Moreover, our recognition model is more robust than the one proposed in [3], because it can predict digits from upside down images. In some cases, due to the perspective with which the image is captured, when rotating the area of interest, we obtain a horizontally aligned image but with numbers rotated by 180 degrees as shown in Fig. 8.

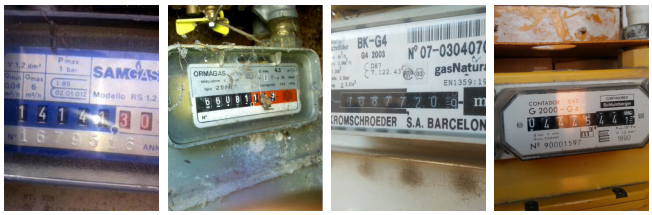
Applying the entire pipeline to the test split, we achieved an accuracy of 85.60%. In addition, we calculate the accuracy of each number based on the position. Results are in Table III. Analyzing results it can be observed that accuracy of the second most significant digit is significantly lower than others. We believe that this happens because of low variability in the dataset for digits in that particular position. We observed that it is difficult to have high variability for most significant digits because most of times they are either "0" or "1".

Average time required to perform both steps in our pipeline on a single image is of 0.12 seconds on a NVIDIA GeForce GTX 1080 GPU, this means that our approach could also be used in real-time scenarios.

Furthermore, we present qualitative evaluation of the pipeline. Fig. 9 shows some examples of correct readings where the performance of the pipeline is particularly robust. In addition, we included some complicated error cases where the pipeline fails to predict the correct output digits.

Output: 1225 Output: 54 Output: 161 Output: 995

Output: 1134 Output: 11591 Output: - Output: 6681



Output: 14141 Output: 66081 Output: 477 Output: 4745

Fig. 9: Qualitatively results of various typologies of the meter. In each row, the first two samples are correctly classified while the remaining show misclassification. Strong glares and shadows are the main issues which lead the algorithm to predict wrong results.

VI. Conclusion

In this paper, we proposed a pipeline to detect and recognize digits of household meters from images. Our method has been tested on different typologies of meter, ranging from mechanical to electric of various kinds: gas, water and electricity using two deep models, one for detection and one for recognition.

The CNN performing the reading of meters predicts the length and values for each digit to compose the number, without any previous segmentation step. This is a key aspect because it allows to quickly create a large-scale ground-truth dataset.

The proposed pipeline is robust against severe perspective distortions, different scales and even upside-down images. Results obtained are promising and the execution time required to apply the whole pipeline makes it possible to employ it in real-time applications.

REFERENCES

[1] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[2] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[3] L. Gómez, M. Rusinol, and D. Karatzas, "Cutting sayre's knot: Reading scene text without segmentation. application to utility meters," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 97–102.

[4] G. Li, S. Xu, X. Liu, L. Li, and C. Wang, "Jersey number recognition with semi-supervised spatial transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1783–1790.

[5] M. Bušta, L. Neumann, and J. Matas, "Deep textspotter: An end-to-end trainable scene text localization and recognition framework," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2223–2231.

[6] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4168–4176.

[7] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 5, pp. 970–983, 2014.

[8] B. Yu and H. Wan, "Chinese text detection and recognition in natural scene using hog and svm," *DEStech Transactions on Computer Science and Engineering*, 2016.

[9] R. Minetto, N. Thome, M. Cord, N. J. Leite, and J. Stolfi, "T-hog: An effective gradient-based descriptor for single line text regions," *Pattern recognition*, vol. 46, no. 3, pp. 1078–1090, 2013.

[10] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[11] I. Gallo, A. Zamberletti, and L. Noce, "Robust angle invariant gas meter reading," in *Digital Image Computing: Techniques and Applications (DICTA), 2015 International Conference on*. IEEE, 2015, pp. 1–7.

[12] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 770–783.

[13] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.