This is an author-prepared version. The published version is available at the SPIE Digital Library (DOI: 10.1117/1.JEI.28.1.013023).

Convolutional Neural Networks for Automatic Meter Reading

Rayson Laroca[a,*], Victor Barroso[a], Matheus A. Diniz[b], Gabriel R. Gonçalves[b], William Robson Schwartz[b], David Menotti[a]
[a]Federal University of Paraná, Department of Informatics, Laboratory of Vision, Robotics and Imaging, Av. Coronel Francisco Heráclito dos Santos 100, Curitiba, Brazil, 81530-000
[b]Federal University of Minas Gerais, Department of Computer Science, Smart Sense Laboratory, Av. Antônio Carlos 6627, Belo Horizonte, Brazil, 31270-010

Abstract. In this paper, we tackle Automatic Meter Reading (AMR) by leveraging the high capability of Convolutional Neural Networks (CNNs). We design a two-stage approach that employs the Fast-YOLO object detector for counter detection and evaluates three different CNN-based approaches for counter recognition. In the AMR literature, most datasets are not available to the research community since the images belong to a service company. In this sense, we introduce a new public dataset, called UFPR-AMR dataset, with 2,000 fully and manually annotated images. This dataset is, to the best of our knowledge, three times larger than the largest public dataset found in the literature and contains a well-defined evaluation protocol to assist the development and evaluation of AMR methods. Furthermore, we propose the use of a data augmentation technique to generate a balanced training set with many more examples to train the CNN models for counter recognition. In the proposed dataset, impressive results were obtained and a detailed speed/accuracy trade-off evaluation of each model was performed. In a public dataset, state-of-the-art results were achieved using less than 200 images for training.

Keywords: automatic meter reading; convolutional neural networks; deep learning; public dataset.

*Rayson Laroca, rblsantos@inf.ufpr.br

1 Introduction

Automatic Meter Reading (AMR) refers to automatically record the consumption of electric energy, gas and water for both monitoring and billing.[1–3] Despite the existence of smart readers,[4] they are not widespread in many countries, especially in the underdeveloped ones, and the reading is still performed manually on site by an operator who takes a picture as reading proof.[2,5] Since this operation is prone to errors, another operator needs to check the proof image to confirm the reading.[2,5] This offline checking is expensive in terms of human effort and time, and has low efficiency.[6] Moreover, due to a large number of images to be evaluated, the inspection is usually done by sampling[7] and errors might go unnoticed.

Performing the meter inspection automatically would reduce mistakes introduced by the human factor and save manpower. Furthermore, the reading could also be executed fully automatically using cameras installed in the meter box.[1,8] Image-based AMR has advantages such as lower cost and fast installation since it does not require renewal or replacement of existent meters.[9]

A common AMR approach includes three phases, namely: (i) counter detection, (ii) digit segmentation and (iii) digit recognition. Counter detection is the fundamental stage, as its performance largely determines the overall accuracy and processing speed of the entire AMR system.

Despite the importance of a robust AMR system and that major advances have been achieved in computer vision using deep learning,[10] to the best of our knowledge, only in Ref. 11, published very recently, Convolutional Neural Networks (CNNs) were employed at all AMR stages. Previous works relied, in at least one stage, on handcrafted features that capture certain morphological and color attributes of the meters/counters. These features are easily affected by noise and might not be robust to different types of meters.

Deep learning approaches are particularly dependent on the availability of large quantities of training data to generalize well and yield high classification accuracy for unseen data.[12] Some previous works[2,6,11] employed large datasets (e.g., more than $45{,}000$ images) to train and evaluate their systems. However, these datasets were not made public. In the AMR literature, the datasets are usually not publicly available since the images belong to the [electricity, gas, water] company. In this sense, we introduce a new public dataset, called UFPR-AMR dataset, with $2{,}000$ fully annotated images to assist the development and evaluation of AMR methods. The proposed dataset is three times larger than the largest public dataset[13] found in the literature.

In this paper, we design a two-stage approach for AMR. We first detect the counter region and then tackle the digit segmentation and recognition stages jointly by leveraging the high capability of CNNs. We employ a smaller version of the YOLO object detector, called Fast-YOLO,[14] for counter detection. Afterward, we evaluate three CNN-based approaches, i.e. CR-NET,[15] Multi-Task Learning[16] and Convolutional Recurrent Neural Network (CRNN),[17] for the counter recognition stage (i.e., digit segmentation and recognition). CR-NET is a YOLO-based model proposed for license plate character detection and recognition, while Multi-Task and CRNN are segmentation-free approaches designed respectively for the recognition of license plates and scene text. These approaches were chosen since promising results have been achieved through them in these applications. Finally, we propose the use of a data augmentation process to train the CNN models for counter recognition to explore different types of counter/digit deformations and their influence on the models' performance.

The experimental evaluation demonstrates the effectiveness of the CNN models for AMR. First, all counter regions were correctly located through Fast-YOLO in the proposed dataset and also in two public datasets found for this task.[5,13] Second, the CR-NET model yielded promising recognition results, outperforming both Multi-Task and CRNN models in the UFPR-AMR dataset. Finally, an impressive recognition rate of 97.30% was achieved using Fast-YOLO and CR-NET in a set of images proposed for end-to-end evaluations of AMR systems, called Meter-Integration subset,[5] against 85% and 87% achieved by the baselines.[2,5] In addition, the CR-NET and Multi-Task models are able to achieve outstanding frames per second (FPS) rates in a high-end GPU, being possible to process respectively 185 and 250 FPS.

Considering the aforementioned discussion, the main contributions of our work are summarized as follows:

- A two-stage AMR approach with CNNs being employed for both counter detection and recognition. In the latter, three different types of CNN are evaluated;

- A public dataset for AMR with $2{,}000$ fully and manually annotated images/meters (i.e., 10,000 digits) with a well-defined evaluation protocol, allowing a fair comparison between different approaches for this task;

- The CNN-based approaches outperformed all baselines in public datasets and achieved impressive results in both accuracy and computational time in the proposed UFPR-AMR dataset.

The remainder of this paper is organized as follows. We briefly review related works in Section 2. The UFPR-AMR dataset is introduced in Section 3. The methodology is presented in Section 4. We report and discuss the results in Section 5. Conclusions and future works are given in Section 6.

2 Related Work

AMR intersects with other Optical Character Recognition (OCR) applications, such as license plate recognition[18] and robust reading,[19] as it must reliably extract text information from images taken under different conditions. Although AMR is not as widespread in the literature as these applications, a satisfactory number of works have been produced in recent years.[3,6,7,11,20] Here, we briefly survey these works by first describing the approaches employed for each AMR stage. Next, we present some papers that address two stages jointly or using the same method. Then, we discuss the deep learning approaches and datasets used so far. Finally, we conclude this section with final remarks.

Counter Detection: Many pioneering approaches exploited the vertical and horizontal pixel projections histograms for counter detection.[1,8,21] Projection-based methods can be easily affected by the rotation of the counter. Refs. 2, 6, 7, 13, 20, 22 took advantage of prior knowledge such as counter's position and/or its colors (e.g., green background and red decimal digits). A major drawback of these techniques is that they might not work on all meter types and the color information might not be stable when the illumination changes. Other works include the use of template matching[7] and the AdaBoost classifier.[3] In the latter, normalized gradient magnitude, Histogram of Oriented Gradients (HOG) and LUV color channels were adopted as low-level feature descriptors.

Digit Segmentation: Projection and color-based approaches have also been widely employed for digit segmentation.[9,22,23] The use of morphological operations with Connected Components Analysis (CCA) was considered in Refs. 6, 20. However, it presents the drawback of depending largely on the result of binarization as it cannot segment digits correctly if they are connected or broken. In Ref. 8, a binary digit/non-digit Support Vector Machine (SVM) was applied in a sliding window fashion, while Gallo et al.[2] exploited Maximally Stable Extremal Regions (MSER). In Ref. 2, the MSER algorithm failed to segment digits in images with problems such as bluring and perspective distortions.

Digit Recognition: Template matching[21–23] along with simple measures of similarity have been widely used for digit recognition. Nevertheless, it is known that if a digit is different from the template due to any font change, rotation or noise, this approach produces incorrect recognition.[18] Thus, many authors have employed an SVM classifier for digit recognition. In Refs. 5, 8, simple features such as pixel intensity were used in training, while HOG descriptors were adopted as features in Refs. 2,7. Although some promising results have been attained, it should be noted that it is not trivial to find the appropriate hyper-parameters of SVM classifiers as well as the best features to be extracted. The open-source Tesseract OCR Engine[24] was applied in Refs. 5, 6, 25, however, satisfactory results were not obtained in any of them. Cerman et al.[6] achieved a remarkable improvement in digit recognition when using a CNN inspired by the LeNet-5 architecture instead of Tesseract.

AMR presents an unusual challenge in OCR: rotating digits. Typically, this is the major cause of errors, even when robust approaches are employed for digit recognition.[3,26] In Ref. 23, this problem was addressed using a Hausdorff distance based algorithm, achieving excellent recognition results in real time. Note that all images were extracted from a single meter and, as pointed out by the authors, a controlled environment was required since there were no preprocessing stage and no algorithm for angle correction.

Miscellaneous: Nodari & Gallo[25] exploited an ensemble of Multilayer Perceptron (MLP) networks to perform the counter detection and digit segmentation without preprocessing and postprocessing stages. Since low F-measure rates were achieved, extra techniques were added in Ref. 5, an extension of Ref. 25. In summary, a watershed algorithm was applied to improve counter detection and Fourier analysis was employed to avoid false positives in digit segmentation. Although better results were attained, only 100 images were used to evaluate their system performance, which may not be representative enough. It should be noted that, to the best of our knowledge, this was the first work to make the images used in the experiments publicly available.

Gao et al.[3] designed a bidirectional long short-term memory (LSTM) network for counter recognition. In their approach, a feature sequence is first generated by a network that combines convolutional and recurrent layers. Then, an attention decoder predicts, recurrently, one digit at each step according to the feature representation. A promising accuracy rate was reported, with most of the errors appearing in cases of half digits.

Gómez et al.[11] presented a segmentation-free AMR system able to output readings directly without explicit counter detection. A CNN architecture was trained in an end-to-end manner where the initial convolutional layers extract visual features of the whole image and the fully connected layers predict the probabilities for each digit. Even though an impressive overall accuracy was achieved, their approach was evaluated only on a large private dataset which has almost $180k$ training samples and mostly images with the counter well centered and occupying a good portion of the image. Thus, as pointed out by the authors, small-meter images pose difficulties to their system.

Datasets: To the best of our knowledge, only Refs. 5, 13 made available the datasets used in their experiments. These datasets are composed of gas meter images with resolution of 640×480 pixels (mostly) and the counter occupying a large portion of the image, which facilitates its detection. Additionally, both datasets are small (253 and 640 images, respectively) and the cameras used to capture them were not specified. It is important to note that in the dataset introduced in Ref. 5, 153 images are divided into different subsets for the evaluation of each stage and only 100 images are for the end-to-end evaluation of the AMR system. Also, there is no split protocol in Ref. 13, which prevents a fair comparison between different approaches.

Deep Learning: Recently, deep learning approaches have won many machine learning competitions and challenges, even achieving superhuman visual results in some domains.[27] Such a fact motivated us to employ deep learning for AMR, since we could find only three works[3,6,11] employing CNNs in this context and all of them made use of large private datasets, overlooking the public datasets. This suggests that these models are able to generalize only with many training samples (e.g., 177,758 images in the segmentation-free system proposed in Ref. 11). Moreover, (i) conventional image processing with handcrafted features was used in at least one stage in Refs. 3, 6,

(ii) the images used in Ref. 3 are mostly sharp and very similar, which does not represent real-world conditions, and (iii) the poor digit segmentation accuracy obtained in Ref. 6, i.e. 81%, through a sequence of conventional image processing methods, discourages its use in real-world applications.

Final Remarks: The approaches developed for AMR are still limited. In addition to the aforementioned points (i.e., private datasets and handcrafted features), many authors do not report the computational time of their approaches, making it difficult an accurate analysis of their speed/accuracy trade-off, as well as their applicability. In this paper, CNNs are used for both counter detection and recognition. We evaluate the CNNs that achieved state-of-the-art results in other applications in both the proposed and public datasets, reporting the accuracy and the computational time to enable fair comparisons in future works.

3 The UFPR-AMR Dataset

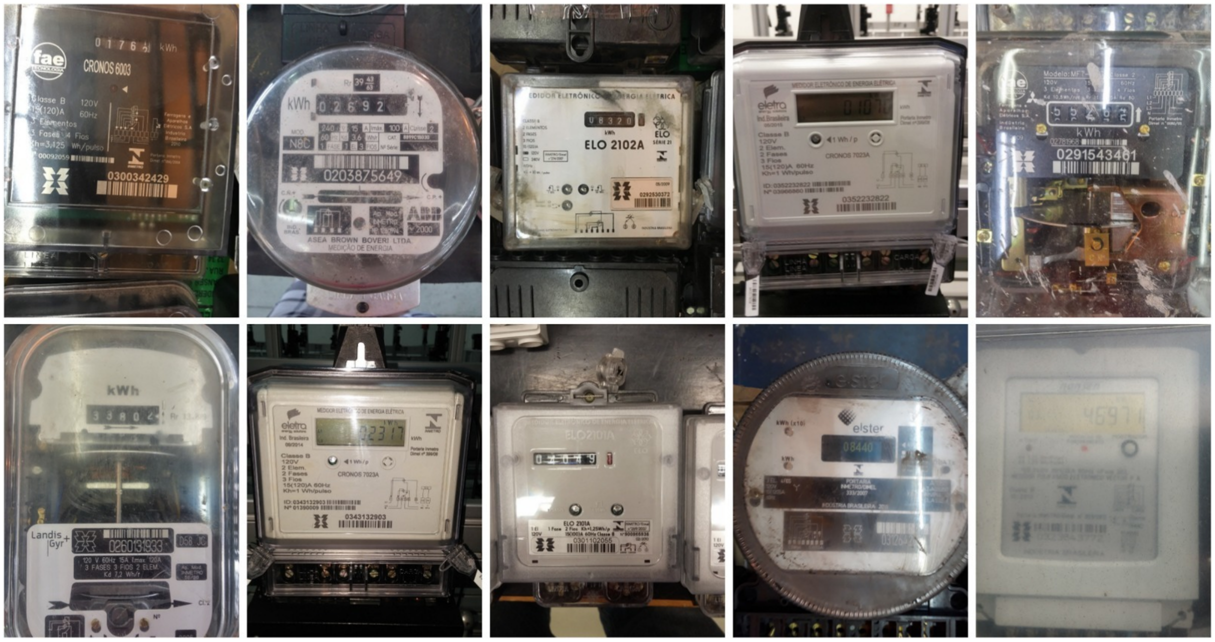


Fig 1 Sample images of the UFPR-AMR dataset (some images were slightly resized for display purposes). Note the diversity of meter types and conditions, as well as the existence of several textual blocks similar to the counter region.

The proposed dataset contains $2{,}000$ images taken from inside a warehouse of the Energy Company of Paraná (Copel), which directly serves more than 4 million consuming units in the Brazilian state of Paraná.[28] Therefore, our dataset presents electric meters of different types and in different conditions. The diversity of the dataset is shown in Fig. 1. One can see that (i) the counter occupies a small portion in the image, which makes its location more difficult; (ii) there are several similar textual blocks (e.g., meter specifications and serial number) to the counter region. The UFPR-AMR dataset is publicly available to the research community at https://web.inf.ufpr.br/vri/databases/ufpr-amr/.

Meter images commonly have some artifacts (e.g., blur, reflections, low contrast, broken glass, dirt, among others) due to the meter's conditions and the misuse of the camera by the human operator, which may impair the reading of electric energy consumption. In addition, it is possible that the digits are rotating or in-between positions (e.g., a digit going from 4 to 5) in some types of counters. In such cases, we consider the lowest digit as the ground truth, since this is the protocol adopted at Copel. The exception, to have a reasonable rule, is between digits 9 and 0, where it should be labeled as 9.

The images were acquired with three different cameras and are available in the JPG format with resolution between 2,340 × 4,160 and 3,120 × 4,160 pixels. The cameras used were: *LG G3 D855*, *Samsung Galaxy J7 Prime* and *iPhone 6s*. As the cameras (cell phones) belong to different price ranges, the images presumably have different levels of quality. Additional information can be seen in Table 1.

Every image has the following annotations available in a text file: the camera in which the image was taken, the counter position (x, y, w, h), the reading, as well as the position of each digit.

Table 1 Additional information about the UFPR-AMR dataset: (a) how many images were captured with each camera; (b) dimensions of counters and digits (width × height in pixels). It is noteworthy the large variation in the sizes of counters and digits.

(a)

Camera	**Images**
LG G3	947
J7 Prime	584
iPhone 6s	469
Total	2,000

(b)

Info	**Counters**	**Digits**
Minimum Size	238 × 96	35 × 63
Maximum Size	1,689 × 365	168 × 283
Average Size	674 × 178	76 × 134
Aspect Ratio	3.79	0.57

All counters of the dataset (regardless of meter type) have 5 digits, and thus 10,000 digits were manually annotated.

Remark that a brand new meter starts with `00000` and the most significant digit positions take longer to be increased. Then, it is natural that the less significant digits (i.e., 0 and 1) have many more instances than the others. Nonetheless, digits 4-9 have a fairly similar number of examples. Fig. 2 shows the distribution of the digits in the UFPR-AMR dataset.

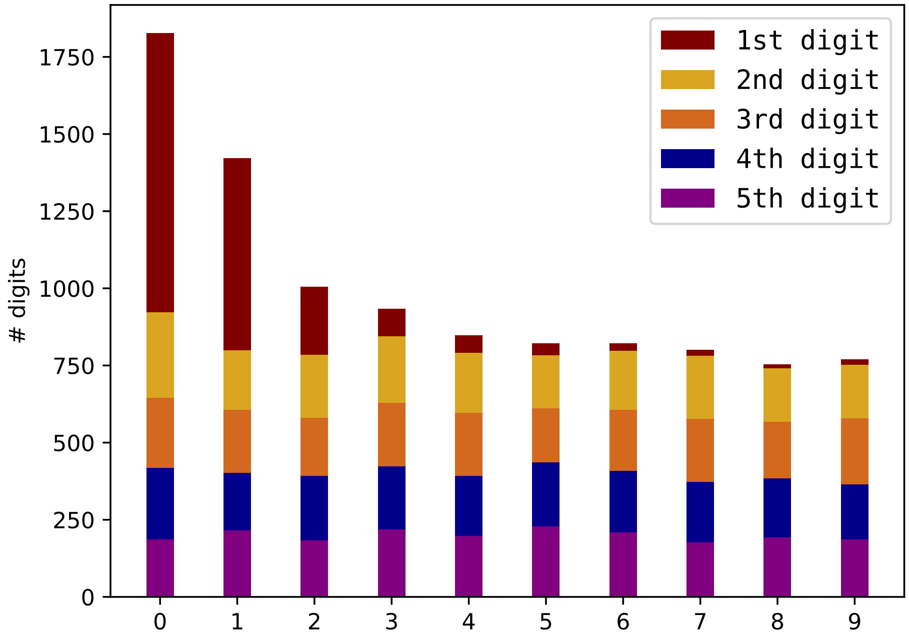


Fig 2 Frequency distribution of digits in the UFPR-AMR dataset. It is worth noting that the first position (i.e., the most significant) consists almost exclusively of 0s and 1s. On the other hand, the frequency of digits in the other positions is very well balanced.

The dataset is split into three sets: training (800 images), validation (400 images) and test (800 images). We adopt this protocol (i.e., with a larger test set) since it has already been adopted in other datasets[29,30] and to provide more samples for analysis of statistical significance. It should be noted that this division was made randomly and the sets generated are explicitly available along with the UFPR-AMR dataset. Additionally, experiments carried out by us suggested that dividing the dataset multiple times and then averaging the results is not necessary, as the proposed division is representative.

4 Methodology

Meters have many textual blocks that can be confused with the counter's reading. Moreover, the Region of Interest (ROI) (i.e., the counter) usually occupies a small portion of the image and its position varies according to the meter type. Therefore, we propose to first locate the counter region and then perform its recognition in the detected patch. We tackle both stages by leveraging the high capability of state-of-the-art CNNs. It is remarkable that, to the best of our knowledge, this is only the second work in which both stages are addressed using CNNs[11] and the first with the experiments being performed on public datasets.

In the following sections, we describe the CNN models employed for counter detection and counter recognition. It is worth noting that all parameters (e.g., CNNs input size, number of epochs, among others) specified here are defined based on the validation set and presented in Section 5, where the experiments are reported.

4.1 Counter Detection

Recently, great progress has been made in object detection through models inspired by YOLO,[14,31,32] a CNN-based object detection system that (i) reframes object detection as a single regression problem; (ii) achieved outstanding and state-of-the-art results in the PASCAL VOC and COCO detection tasks.[33] For that reason, we decided to fine-tune it for counter detection. However, as we want to detect only one class and the computational cost is one of our main concerns, we chose to use a smaller model, called Fast-YOLO,[14] which uses fewer convolutional layers than YOLO and fewer filters in those layers. Despite being smaller, Fast-YOLO (architecture shown in Table 2) yielded outstanding results, i.e. detections with Intersection over Union (IoU) ≥ 0.8 with the ground truth, in preliminary experiments. The IoU is often used to assess the quality of predictions in object detection tasks[34] and can be expressed by the formula

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})}\,, \tag{1}$$

where B_p and B_{gt} are the predicted and ground truth bounding boxes, respectively. The closer the IoU is to 1, the better the detection. For this reason, we believe that very deep models are not necessary to handle the detection of a single class of objects.

For counter detection, we use the weights pre-trained on ImageNet[35] and perform two minor changes in the Fast-YOLO model. First, we recalculate the anchor boxes for the UFPR-AMR dataset using the algorithm available in Ref. 36. Anchors are initial shapes that serve as references at multiple scales and aspect ratios. Instead of predicting arbitrary bounding boxes, YOLO only adjusts the size of the nearest anchor to the size of the object. Predicting offsets instead of coordinates simplifies the problem and makes it easier for the network to learn.[33] Then, we reduce the

Table 2 Fast-YOLO network used to detect the counter region.

	Layer	Filters	Size	Input	Output
0	conv	16	$3 \times 3/1$	$416 \times 416 \times 3$	$416 \times 416 \times 16$
1	max		$2 \times 2/2$	$416 \times 416 \times 16$	$208 \times 208 \times 16$
2	conv	32	$3 \times 3/1$	$208 \times 208 \times 16$	$208 \times 208 \times 32$
3	max		$2 \times 2/2$	$208 \times 208 \times 32$	$104 \times 104 \times 32$
4	conv	64	$3 \times 3/1$	$104 \times 104 \times 32$	$104 \times 104 \times 64$
5	max		$2 \times 2/2$	$104 \times 104 \times 64$	$52 \times 52 \times 64$
6	conv	128	$3 \times 3/1$	$52 \times 52 \times 64$	$52 \times 52 \times 128$
7	max		$2 \times 2/2$	$52 \times 52 \times 128$	$26 \times 26 \times 128$
8	conv	256	$3 \times 3/1$	$26 \times 26 \times 128$	$26 \times 26 \times 256$
9	max		$2 \times 2/2$	$26 \times 26 \times 256$	$13 \times 13 \times 256$
10	conv	512	$3 \times 3/1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$
11	max		$2 \times 2/1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$
12	conv	1024	$3 \times 3/1$	$13 \times 13 \times 512$	$13 \times 13 \times 1024$
13	conv	1024	$3 \times 3/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 1024$
14	conv	30	$1 \times 1/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 30$
15	detection				

number of filters in the last convolutional layer from 125 to 30 to output 1 class instead of 20. The number of filters in the last layer is given by

$$\mathit{filters} = (C + 5) \times A\,, \tag{2}$$

where A is the number of anchor boxes (we use A = 5) used to predict bounding boxes. Each bounding box has four coordinates (x, y, w, h), a objectness value[37] (i.e., how likely the bounding box contains an object) along with the probability of that object belonging to each of the C classes, in our case $C = 1$ (i.e., only the counter region).[33] Remark that the choice of appropriate anchor boxes is very important, and thus our boxes are similar to counters in size and aspect ratio.

We employ Fast-YOLO's multi-scale training.[33] In short, every 10 batches, the network randomly chooses a new image dimension size from 320×320 to 608×608 pixels (default values). These dimensions were chosen considering that the Fast-YOLO model down samples the image by a factor of 32. As pointed out in Ref. 33, this approach forces the network to learn to predict well across a variety of input dimensions. Then, we use 416×416 images as input since the best results (speed/accuracy trade-off in the validation set) were obtained with this dimension as input. It is remarkable that, although YOLO networks have a $1:1$ input aspect ratio, previous works[15,30] have attained excellent object detection results (over 99% recall) in images with different aspect ratios (e.g., $1{,}920 \times 1{,}080$). All image resizing operations were performed using bilinear interpolation.

In cases where more than one counter is detected, we consider only the detection with the highest confidence since each image/meter has only one counter. To avoid losing digits in cases where the counter is not very well detected, we add a margin (with size chosen based on the validation set) on the detected patch so that all digits are within it for the recognition stage. A negative recognition result is given in cases where no counter is found.

4.2 Counter Recognition

We employ three CNN-based approaches for performing counter recognition: CR-NET,[15] Multi-Task Learning[16] and CRNN.[17] These models were chosen because promising results were obtained through them in other OCR applications, such as license plate recognition and scene text recognition. It is noteworthy that, unlike CR-NET, the last two models do not need the coordinates of each digit in the training phase. In other words, Multi-Task Learning and CRNN approaches only need the counter's reading. This is of paramount importance in cases where a large number of images is available for learning (e.g., millions or hundreds of thousands), since manually labeling each digit is very costly and prone to errors.

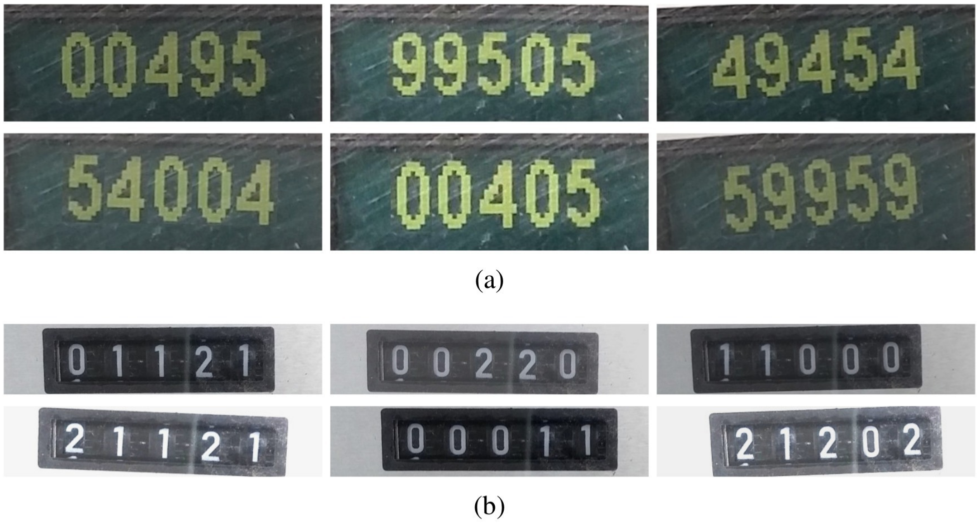
The remainder of this section is organized into four parts, one to describe the data augmentation method, which is essential to effectively train the deep models, and one part for each CNN approach employed for counter recognition.

4.2.1 Data Augmentation

It is well known that unbalanced data is undesirable for neural network classifiers since the learning of some patterns might be biased. For instance, some classifiers may learn to always classify the first digit as 0, but this is not always the case (see Fig. 2), although it is by far the most common. To address this issue, we employ the data augmentation technique proposed in Ref. 16. Using this technique, we are able to create a new set of images, where each digit class is equally represented in every position. This set consists of permutations of the original images. The order and frequency of the digits in the generated counters are chosen to uniformly distribute the digits along the positions. Note that the location of each digit (i.e., its bounding box) is required to apply this data augmentation technique.

Some artificially generated images when applying the method in the UFPR-AMR dataset are shown in Fig. 3. We also perform random variations of brightness, rotation and crop coordinates to increase even more the robustness of our augmented images, creating new training examples for the CNNs. As can be seen, the data augmentation approach works on different types of meters.

The adjustment of parameters is of paramount importance for the effectiveness of this technique since the presence of very large variations in brightness, rotation or cropping, for instance, might impair the recognition through the generation of images that do not match real scenarios. Therefore, the parameter ranges were empirically determined based on experiments performed on the validation set, i.e., brightness variation of the pixels $[0.5; 2]$, rotation angles between $-5°$ and $5°$ and cropping from -2% to 8% of the counter size. Once these ranges were established, new counter images were generated using random values within those ranges for each parameter.



(a)

(b)

Fig 3 Data augmentation examples, where the images in the upper-left corner of (a) and (b) are the originals, and the others were generated automatically. In (a) and (b), counters of different types and aspect ratios are shown.

4.2.2 CR-NET

CR-NET is a YOLO-based model proposed for license plate character detection and recognition.[15] This model consists of the first eleven layers of YOLO and four other convolutional layers added to improve non-linearity. In Ref. 15, CR-NET (with an input size of 240×80 pixels) was capable of detecting and recognizing license plate characters at 448 FPS. Laroca et al.[30] also achieved great results applying CR-NET for this purpose.

The CR-NET architecture is shown in Table 3. As in the counter detection stage, we recalculate the anchors for our data and make adjustments in the number of filters in the last layer. Furthermore, we adapt the input image size taking into account the aspect ratio of the counters, which have a different aspect ratio when compared to license plates in Ref. 15. Then, we use as input an image with resolution of 400×106 pixels since the results obtained when using other sizes (e.g., 360×95 and 440×116) were worse or similar, but with a higher computational cost.

We consider only the five digits detected/recognized with highest confidence, since commonly more than five digits are predicted. However, we noticed that the same digit might be detected

Table 3 CR-NET with some modifications for counter recognition: input size of 400×106 pixels and 75 filters in the last layer.

	Layer	Filters	Size	Input	Output
0	conv	32	$3 \times 3/1$	$400 \times 106 \times 3$	$400 \times 106 \times 32$
1	max		$2 \times 2/2$	$400 \times 106 \times 32$	$200 \times 53 \times 32$
2	conv	64	$3 \times 3/1$	$200 \times 53 \times 32$	$200 \times 53 \times 64$
3	max		$2 \times 2/2$	$200 \times 53 \times 64$	$100 \times 26 \times 64$
4	conv	128	$3 \times 3/1$	$100 \times 26 \times 64$	$100 \times 26 \times 128$
5	conv	64	$1 \times 1/1$	$100 \times 26 \times 128$	$100 \times 26 \times 64$
6	conv	128	$3 \times 3/1$	$100 \times 26 \times 64$	$100 \times 26 \times 128$
7	max		$2 \times 2/2$	$100 \times 26 \times 128$	$50 \times 13 \times 128$
8	conv	256	$3 \times 3/1$	$50 \times 13 \times 128$	$50 \times 13 \times 256$
9	conv	128	$1 \times 1/1$	$50 \times 13 \times 256$	$50 \times 13 \times 128$
10	conv	256	$3 \times 3/1$	$50 \times 13 \times 128$	$50 \times 13 \times 256$
11	conv	512	$3 \times 3/1$	$50 \times 13 \times 256$	$50 \times 13 \times 512$
12	conv	256	$1 \times 1/1$	$50 \times 13 \times 512$	$50 \times 13 \times 256$
13	conv	512	$3 \times 3/1$	$50 \times 13 \times 256$	$50 \times 13 \times 512$
14	conv	75	$1 \times 1/1$	$50 \times 13 \times 512$	$50 \times 13 \times 75$
15	detection				

more than once by the network. Therefore, we first apply a non-maximal suppression algorithm to eliminate redundant detections. Although *highly unlikely* (i.e., $\approx 0.1\%$), it is also possible that less than five digits are detected by the CR-NET, as shown in Fig 4. In such cases, we reject the counter's recognition.

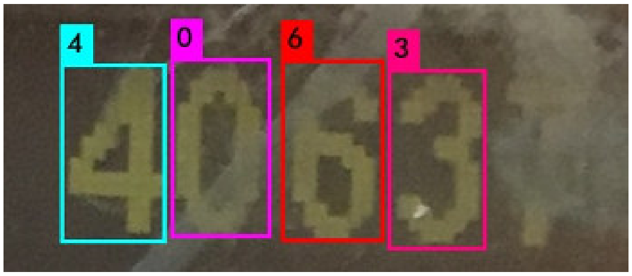


Fig 4 A counter where less than 5 digits were detected/recognized by the CR-NET. We could employ leading zeros (e.g., $4063 \rightarrow 04063$), however, this could result in a large error in the meter reading.

4.2.3 Multi-Task Learning

Multi-Task Learning is another approach for character string recognition developed for license plates.[16,38] This method skips the character segmentation stage and directly recognizes the character string of an image (here, the cropped counter). Since there might be multiple characters, each character is modeled as a task on the network.

For the UFPR-AMR dataset, we use a similar architecture adding the restraint that each character must be a digit, transforming the output space from 36 (their work considers numbers and letters) to 10 for each digit. The architecture holistically segments and recognizes all five characters due to its multi-task output.

Table 4 shows the architecture of the model, which is very compact with only 4 convolutional layers followed by a fully connected shared layer and two fully connected layers for each digit, indexed from 1 to 5. Each output represents the classification of one of the digits. Thus, no explicit segmentation is performed in this approach.

Table 4 Multi-Task layers and hyperparameters.

	Layer	Filters	Size	Input	Output
0	conv	128	$5 \times 5/1$	$220 \times 60 \times 1$	$220 \times 60 \times 128$
1	max		$2 \times 2/2$	$220 \times 60 \times 128$	$110 \times 30 \times 128$
2	conv	128	$3 \times 3/1$	$110 \times 30 \times 128$	$110 \times 30 \times 128$
3	conv	192	$3 \times 3/1$	$110 \times 30 \times 128$	$110 \times 30 \times 192$
4	max		$2 \times 2/2$	$110 \times 30 \times 192$	$55 \times 15 \times 192$
5	conv	256	$3 \times 3/1$	$55 \times 15 \times 192$	$55 \times 15 \times 256$
6	max		$2 \times 2/2$	$55 \times 15 \times 256$	$27 \times 7 \times 256$
7	flatten			$27 \times 7 \times 256$	48384
	Layer	**Units**		**Input**	**Output**
8	dense	4096		48384	4096
9	dense[1..5]	512		4096	512
10	dense[1..5]	10		512	10

4.2.4 Convolutional Recurrent Neural Network

CRNN[17] is a model designed for scene text recognition that consists of convolutional layers followed by recurrent layers, in addition to a custom transcription layer to convert the per-frame predictions into a label sequence. Given the counter patch, containing the digits, the convolutional layers act as a feature extractor, which is then transformed into a sequence of feature vectors and fed into an LSTM[39] recurrent layer. This layer handles the input as a sequence labeling problem, predicting a label distribution $y = y_1, y_2, ..., y_t$ for each feature vector $x = x_1, x_2, ..., x_t$ from the feature map.

The Connectionist Temporal Classification (CTC)[40] cost function is adopted for sequence decoding. The CTC has a softmax layer with a label more than the original 10 digits. The activation of each feature vector corresponds to a unique label that can be one of the ten digits or a 'blank' (i.e., the absence of digit). Thus, this model is able to predict a variable number of digits, differently from Multi-Task where 5 digits are always predicted. As the classification is done through the whole feature map from the convolutional layers, digit segmentation is not required.

We evaluate different network architectures with variations in the input size and in the number of filters and convolutional layers. As shown in Table 5, the input size is 160×40 pixels and there are only one LSTM layer (instead of two, as in Ref. 17) since the best results (considering the speed/accuracy trade-off) in the validation set were obtained with these parameters.

5 Experimental Results

In this section, we report the experiments carried out to verify the effectiveness of the CNN-based methods in the UFPR-AMR dataset and also in public datasets. All experiments were performed

Table 5 CRNN layers and hyperparameters.

Layer		Filters	Size	Input	Output
0	conv	64	$3 \times 3/1$	$160 \times 40 \times 1$	$160 \times 40 \times 64$
1	max		$2 \times 2/2$	$160 \times 40 \times 64$	$80 \times 20 \times 64$
2	conv	128	$3 \times 3/1$	$80 \times 20 \times 64$	$80 \times 20 \times 128$
3	max		$2 \times 2/2$	$80 \times 20 \times 128$	$40 \times 10 \times 128$
4	conv	256	$3 \times 3/1$	$40 \times 10 \times 128$	$40 \times 10 \times 256$
5	conv	256	$3 \times 3/1$	$40 \times 10 \times 256$	$40 \times 10 \times 256$
6	max		$2 \times 2/2 \times 1$	$40 \times 10 \times 256$	$40 \times 5 \times 256$
7	conv	512	$3 \times 3/1$	$40 \times 5 \times 256$	$40 \times 5 \times 512$
8	batch				
9	conv	512	$3 \times 3/1$	$40 \times 5 \times 512$	$40 \times 5 \times 512$
10	batch				
11	max		$2 \times 2/2 \times 1$	$40 \times 5 \times 512$	$40 \times 2 \times 512$
12	conv	512	$2 \times 2/2 \times 1$	$40 \times 2 \times 512$	$40 \times 1 \times 512$

Layer		Input	Hidden Layer	Output
13	LSTM	$512 \times 1 \times 40$	256	11

on a computer with an AMD Ryzen Threadripper 1920X 3.5GHz CPU, 32 GB of RAM and an NVIDIA Titan Xp GPU (3,840 CUDA cores and 12 GB of RAM).

We first assess counter detection since the counter regions used for recognition are from the detection results, rather than cropped directly from the ground truth. This is done to provide a realistic evaluation of the entire AMR system, where well-performed counter detection is essential to achieve outstanding recognition results. Next, each approach for counter recognition is evaluated and a comparison between them is presented.

Counter detection is evaluated in the UFPR-AMR dataset and also in two public datasets,[5,13] while counter recognition is assessed only in the UFPR-AMR dataset. This is because (i) two different sets of images were used to evaluate digit segmentation and recognition in Ref. 5, and thus it is not possible to use these sets in the counter recognition approaches (where these stages are performed jointly); (ii) Ref. 13 performed digit recognition on a subset of their dataset which was not made publicly available.

We will finally evaluate the entire AMR pipeline in a subset of 100 images (640 × 480) taken from the public dataset introduced by Vanetti et al.[5] This subset, called Meter-Integration, was used to perform an overall evaluation of the AMR methods proposed in Refs. 2,5. It should be noted that other subsets of the dataset, containing different images, were used to evaluate each AMR stage independently and the training images (in the overall evaluation) are from these subsets.[5] Aiming at a fair comparison, we employ the same protocol.

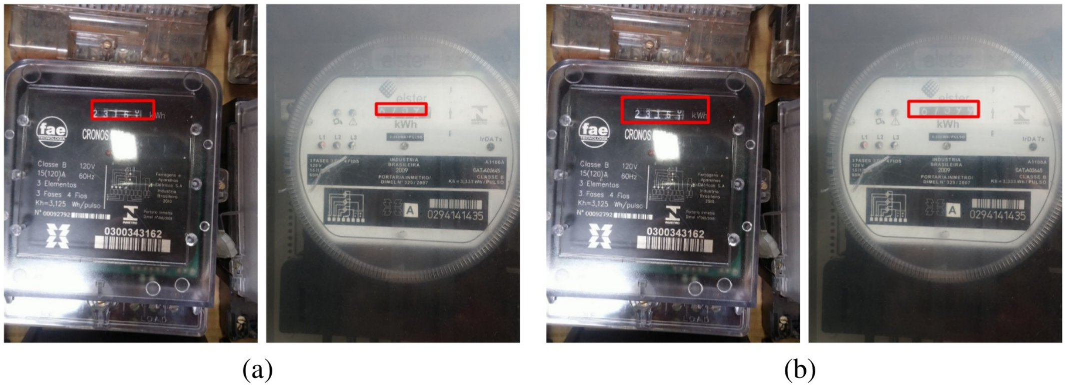
5.1 Counter Detection

For evaluating counter detection, we employ the bounding box evaluation defined in the PASCAL VOC Challenge,[34] where the predicted bounding box is considered to be correct if its IoU with the ground truth is greater than 50% (IoU > 0.5). This metric was also used in previous works,[5,25] being interesting once it penalizes both over- and under-estimated objects.

According to the detection evaluation described above, the network correctly detected 99.75% of the counters with an average IoU of 83%, failing to locate the counter in just two images (798/800). However, in these two cases, it is still possible to recognize the digits from the detected counters, since they were actually detected (with IoU ≤ 0.5) and all digits are within the ROI after adding a margin (as explained in Section 4.1). In the validation set, a margin of 20% (of the bounding box size) is required so that all digits are within the ROI. Thus, we applied a 20% margin in the test set as well. Fig. 5 shows both cases where the counters were detected with IoU ≤ 0.5 before and after adding this margin. Note that, in this way, all counter digits are within the located region using Fast-YOLO.

Some detection results achieved by the Fast-YOLO model are shown in Fig. 6. As can be seen, well-located predictions were attained on counters of different types and under different conditions.

In terms of computational speed, the Fast-YOLO model takes about 3.30 ms per image (303



(a)

(b)

Fig 5 Bounding boxes predicted by the Fast-YOLO model before (a) and after (b) adding the margin (20% of the bounding box size).

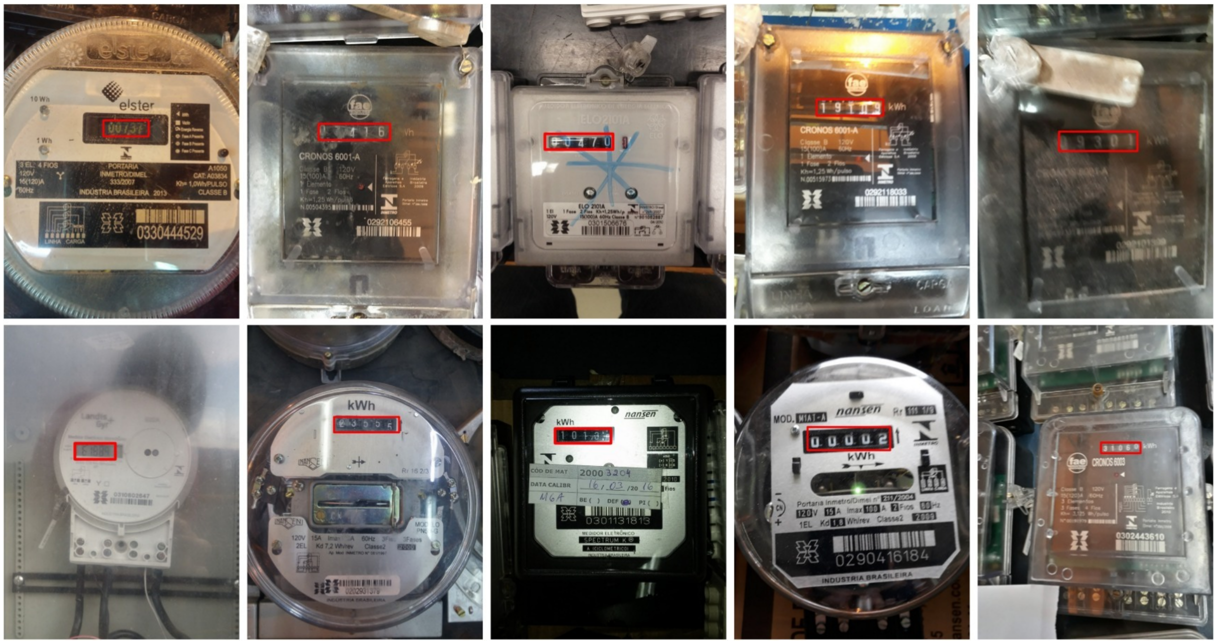


Fig 6 Samples of counter detection obtained with the Fast-YOLO model in the UFPR-AMR dataset.

FPS). The model was trained using the Darknet framework[41] and the following parameters were used for training the network: $60k$ iterations (max batches) and learning rate = [10^{-3}, 10^{-4}, 10^{-5}] with steps at $25k$ and $35k$ iterations.

5.1.1 Counter Detection on Public Datasets

To demonstrate the robustness of Fast-YOLO for counter detection, we employ it on the public datasets found in the literature[5, 13] and compare the results with those reported in previous works.

Vanetti et al.[5] employed a subset of 153 images of their dataset specially for the evaluation of counter detection, being 102 for training and 51 for testing. In Ref. 13, a larger dataset (with 640 images) was introduced, but no split protocol was defined.

As the dataset introduced in Ref. 5 has a split protocol, we employed the same division in our experiments. We randomly removed 20 images from the training set and used them as validation. For the experiments performed in the dataset introduced in Ref. 13, we perform 5-fold cross-validation with images assigned to folds randomly in order to achieve a fair comparison. Thus, in each run, we used 384 images (60%) for training and 128 images (20%) for each validation and testing, i.e., a 3/1/1 split protocol.

As mentioned in the related work section, both datasets are composed of gas meter images. Such a fact is relevant since gas meters usually have red decimal digits that should be discarded in the reading process.[2,5,11,13] Therefore, we manually labeled, in each image, a bounding box containing only the significant digits for training Fast-YOLO. These annotations are also publicly available to the research community at https://web.inf.ufpr.br/vri/databases/ufpr-amr/.

Table 6 F-measure values obtained by Fast-YOLO and previous works in the public datasets found in the literature.

Approach	F-measure	
	Dataset[13]	Dataset[5]
Nodari & Gallo[25]	–	70.00%
Gonçalves[13]	96.09%	88.24%
Vanetti et al.[5]	–	96.00%
Fast-YOLO	**100.00%**	**100.00%**
Fast-YOLO (IoU > 0.7)	98.59%	92.16%

The Fast-YOLO model correctly detected 100% of the counters in both datasets, outperforming the results obtained in previous works, as shown in Table 6. It is noteworthy the outstanding IoU values attained: on average 83.39% in the dataset proposed in Ref. 5 and 91.28% in the dataset introduced in Ref. 13. We believe that these excellent results are due to the fact that, in these datasets, the counter occupies a large portion of the image and the meters/counters are quite similar when comparing with the UFPR-AMR dataset. Fig. 7 shows a counter from each dataset detected using Fast-YOLO.

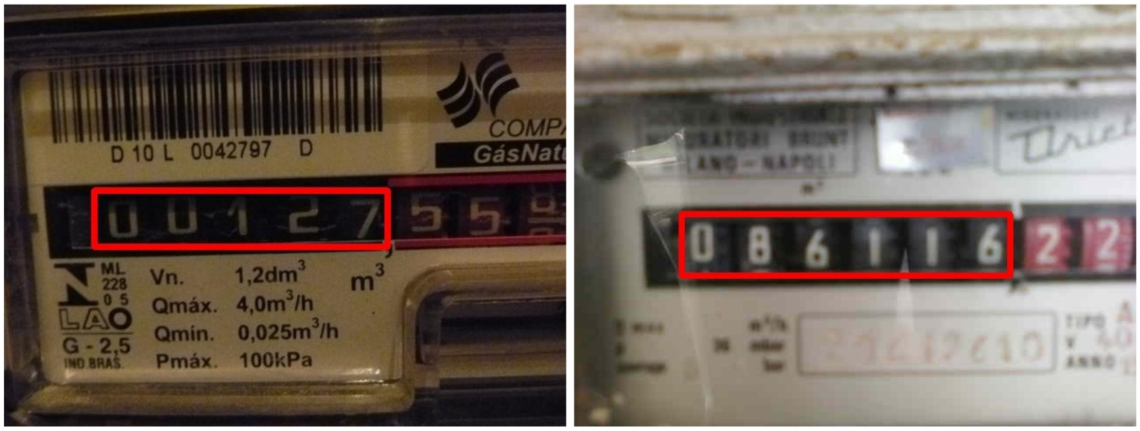
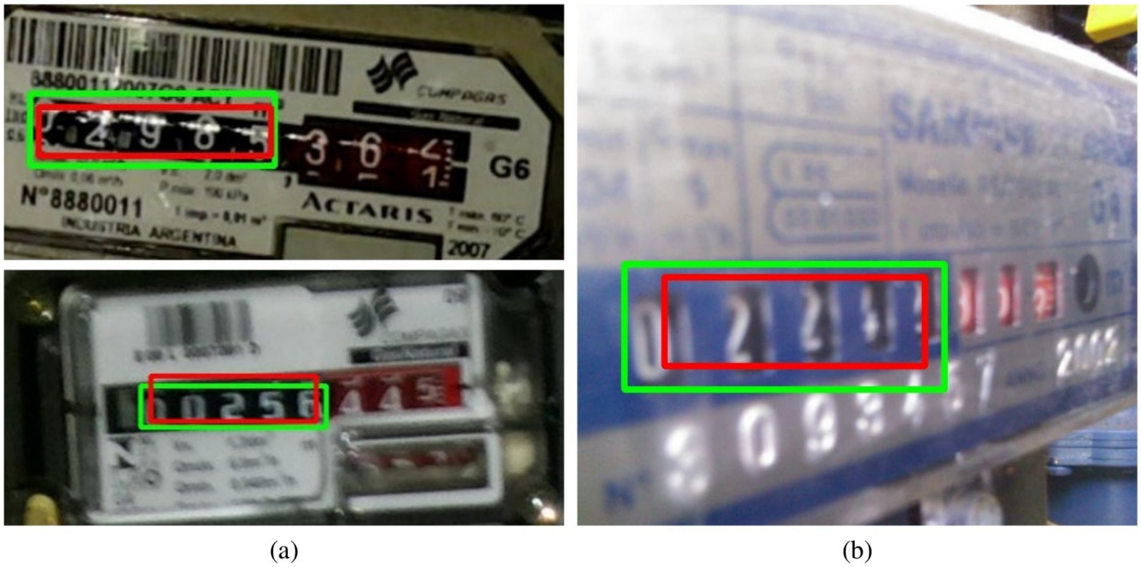


Fig 7 Examples of counter detection obtained with the Fast-YOLO model. Note that the counter region in the images of the Dataset[13] (left) and Dataset[5] (right) is quite larger than in the UFPR-AMR dataset.

Additionally, we reported the results with a higher detection threshold (i.e., IoU > 0.7). It is remarkable that more than 90% of the counters were located with an IoU (with the ground truth) greater than 0.7 in both datasets. We noticed that the detections with a lower IoU occurred mainly in cases where the meter/counter was inclined or tilted, as illustrated in Fig. 8.

5.2 Counter Recognition

For this experiment, we report the mean of 10 runs for both digit and counter recognition accuracy. While the former is the number of correctly recognized digits divided by the number of digits in the test set, the latter is defined as the number of correctly recognized counters divided by the test set



(a) (b)

Fig 8 Samples of counters detected with a lower IoU with the ground truth. (a) and (b) show images of the datasets proposed in Refs. 13 and 5, respectively. The predicted position and ground truth are outlined in red and green, respectively. Observe that all digits would be within the ROI with the addition of a small margin.

size, since each image has only a single meter/counter. Additionally, all CNN models were trained with and without data augmentation, so that we can analyze how data augmentation (described in Section 4.2.1) affects the performance of each model.

For a fair comparison, we (i) generated $300{,}000$ images and applied them for training all CNNs (more images were not generated due to hardware limitations); (ii) disabled the Darknet's (hence CR-NET's) built-in data augmentation, which creates a number of images with changed colors (hue, saturation, exposure) randomly cropped and resized; and (iii) evaluated different margin values (less than the 20% applied previously) in the predictions obtained by Fast-YOLO, since each approach might work better with different margin values.

The recognition rates achieved by all models are shown in Table 7. We performed statistical paired t-tests at a significance level $\alpha = 0.05$, which showed that there is a significant difference in the results obtained with different models. As expected, the results are greatly improved when taking advantage of data augmentation. The best results were achieved with the CR-NET model, which correctly recognized 94.13% of the counters with data augmentation against 92.30% and 87.69% through CRNN and Multi-Task Learning, respectively. This suggests that segmentation-free approaches require a lot of training data to achieve promising recognition rates, as in Ref. 11 where 177,758 images were used for training.

Table 7 Recognition rates obtained in the UFPR-AMR dataset using Fast-YOLO for counter detection and each of the CNN models for counter recognition.

Approach	Accuracy (%)	
	Digits	Counters
Multi-Task (original training set)	24.64 ± 0.25	00.00 ± 0.00
CRNN (original training set)	92.85 ± 0.93	77.75 ± 2.39
CR-NET (original training set)	97.78 ± 0.17	91.95 ± 0.52
Multi-Task (with data augmentation)	95.96 ± 0.25	87.69 ± 0.40
CRNN (with data augmentation)	97.87 ± 0.21	92.30 ± 0.56
CR-NET (with data augmentation)	**98.30 ± 0.09**	**94.13 ± 0.50**

It is important to highlight that it was not possible to recognize any counter when training the Multi-Task model without data augmentation. We performed several experiments reducing the size of the Multi-Task network to verify if a smaller network could learn a better discriminant function. However, better results were not achieved. This is because the dataset is biased and so is the recognition. Even though the first digit has the strongest bias (given the large amount of 0 and 1s in that position), the other digits still have a considerable bias due to the low number of training samples. For example, the Multi-Task network may learn to predict the last digit/task as '5' on every occasion it sees a particular combination of the other digits that is present in the training set. In other words, the network may learn correlations between the outputs that do not exist in practice (in other applications this may be beneficial, but in this case it is not). Such a fact explains why the segmentation-free approaches had a higher performance gain with data augmentation, which balanced the training set and eliminated the undesired correlation between the outputs.

To assess the speed/accuracy trade-off of the three CNN models, we list in Table 8 the time required for each approach to perform the recognition stage. We report the FPS rate achieved by each approach considering only the recognition stage and also considering the detection stage (in parenthesis), which takes about 3.30 ms per image using Fast-YOLO. The reported time is the average time spent processing all images, assuming that the network weights are already loaded. For completeness, for each network, we also list the number of parameters as well as the number of billion floating-point operations (BFLOP) required for a single forward pass over a single image.

Table 8 Results obtained in the UFPR-AMR dataset and the computational time required for each approach to perform counter recognition. In parentheses is shown the FPS rate when considering the detection stage.

Approach	BFLOP	Parameters	Time (ms)	FPS	Accuracy (%)
Multi-Task	3.45	209M	0.6956	1437 (250)	87.69 ± 0.40
CRNN	2.50	7M	5.1751	193 (118)	92.30 ± 0.56
CR-NET	5.37	3M	2.1071	475 (185)	**94.13 ± 0.50**

The CR-NET and Multi-Task approaches achieved impressive FPS rates. Looking at Table 8, the difference between using each one of them is clear. The CR-NET model achieved an accuracy of 94.13% at 475 FPS, while the Multi-Task model was capable of processing 1437 FPS with a recognition rate of 87.69%. When considering the time spent in the detection stage, it is possible to process 185 and 250 FPS using the CR-NET and Multi-Task models, respectively.

It is worth noting that: (i) even though the Multi-Task network has many more parameters than CR-NET and CRNN, it is still the fastest one; (ii) the CRNN model requires a lower number of floating-point operations for a single forward pass than the CR-NET and Multi-Task networks, however, it is still the model that takes more time to process a single image. In this sense, we claim that there are several factors (in addition to those mentioned above) that affect the time it takes for a network to process a frame, e.g., the input size, its specific characteristics and the framework in which it is implemented. For example, two networks may require exactly the same number of floating-point operations (or have the same number of parameters) and still one be much faster than the other. Although much effort was made to ensure fairness in our experiments, the comparison might not be entirely fair since we used different frameworks to implement the networks and there are probably differences in implementation and optimization between them. The CR-NET model was trained using the Darknet framework,[41] while the CRNN and Multi-Task models were trained using PyTorch[42] and Keras,[43] respectively.

Fig. 9 illustrates some of the recognition results obtained in the UFPR-AMR dataset when employing the CR-NET model (i.e., the one with the best accuracy). It is noticeable that the model is able to generalize well and correctly recognize counters from meters of different types and in different conditions. Regarding the errors, we noticed that they occurred mainly due to rotating digits and artifacts in the counter region, such as reflections and dirt.

5.3 Overall Evaluation on the Meter-Integration Subset

The Meter-Integration subset[5] was used to evaluate the AMR methods proposed in Refs. 5, 13. Thus, we decided to perform experiments on this dataset and compare the results with those obtained in both works. As previously mentioned, the training images are from other subsets of the dataset proposed in Ref. 5. Remark that there are only 102 and 62 training images for counter detection and recognition, respectively.

We employ only the CR-NET model in this experiment since it outperformed both Multi-Task and CRNN models in the UFPR-AMR dataset. The mean accuracy of 10 runs is reported for both digit and counter recognition accuracy. As the counters in the training set have from 4 to 7 digits and not a fixed number of digits, we adopted a 0.5 confidence threshold (we report it for sake of

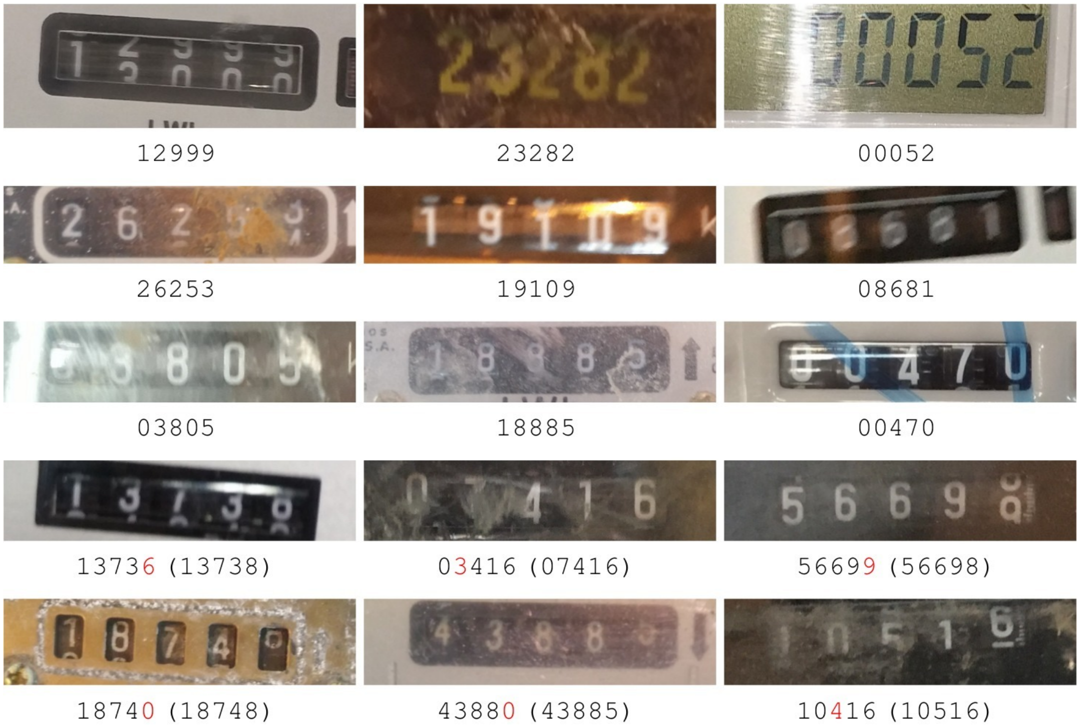


Fig 9 Results obtained by the CR-NET model in the UFPR-AMR dataset. The first three rows show examples of successfully recognized counters, while the last two rows show samples of incorrectly recognized counters. Some images were slightly resized for display purposes.

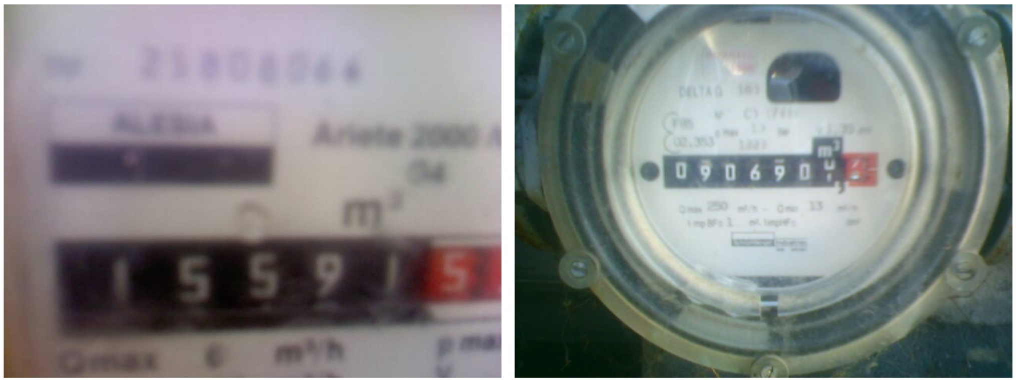
reproducibility) to deal with a variable number of digits, instead of always considering 5 digits per counter. This threshold was chosen based on 12 validation images (i.e., 20%) randomly taken from the training set. Table 9 shows the results obtained in previous works and using the Fast-YOLO and CR-NET networks for counter detection and recognition, respectively.

Table 9 Results obtained in the Meter-Integration subset by previous works and using Fast-YOLO & CR-NET.

Approach	Accuracy (%)	
	Digits	Counters
Gallo et al.[2] (original training set)	–	85.00
Vanetti et al.[5] (original training set)	–	87.00
Fast-YOLO & CR-NET (original training set)	97.94 ± 0.85	94.50 ± 1.72
Fast-YOLO & CR-NET (data augmentation)	$\mathbf{99.56 \pm 0.34}$	$\mathbf{97.30 \pm 1.42}$

As expected, the recognition rate accomplished by our deep learning approach was considerably better than those obtained in previous works (87% → 94.50%), which employed methods based on conventional image processing with handcrafted features. It is noteworthy the ability of both Fast-YOLO and CR-NET models to generalize with very few training images in each stage, i.e., 102 for counter detection and 62 for counter recognition.

The results were improved when using data augmentation, as in the experiments carried out on the UFPR-AMR dataset. The accuracy achieved was 97.30%, significantly outperforming the baselines. It is worth noting that, on average, only 2-3 counters were incorrectly classified and generally the error occurred in the rightmost digit of the counter. Two samples of errors are shown in Fig. 10: the last digit 1 was incorrectly labeled as 0 in one of the cases, probably due to some noise in the image, while in the other case the last digit was detected/recognized with confidence lower than 0.5, apparently due to the m^3 text touching the digit (there were no similar examples in the training set).



`15590 (15591)`

`090690- (0906900)`

Fig 10 Incorrect readings obtained with the Fast-YOLO & CR-NET approach, where the last digit was incorrectly classified (left), and the last digit was detected/recognized with a confidence value below the threshold (right).

6 Conclusions

In this paper, we presented a two-stage AMR approach with CNNs being employed for both counter detection and recognition. The Fast-YOLO[14] model was employed for counter detection, while three CNN-based approaches (CR-NET,[15] Multi-Task Learning[16] and CRNN[17]) were employed for counter recognition. In addition, we proposed the use of data augmentation for training the CNN models for counter recognition, in order to construct a balanced training set with many more examples.

We also introduced a public dataset that includes $2{,}000$ images (with $10{,}000$ manually labeled digits) from electric meters of different types and in different conditions, i.e., the UFPR-AMR dataset. It is three times larger than the largest dataset found in the literature for this task and contains a well-defined evaluation protocol, allowing a fair comparison of different methods. Furthermore, we labeled the region containing the significant digits in two public datasets[5,13] and these annotations are also publicly available to the research community.

The counter detection stage was successfully tackled using the Fast-YOLO model, which was able to detect the region containing the significant digits in all images of every dataset evaluated in this work. For counter recognition, the CR-NET model yielded the best recognition results in the UFPR-AMR dataset (i.e., 94.13%), outperforming both Multi-Task and CRNN models which achieved 87.69% and 92.30%, respectively. These results were attained by taking advantage of data augmentation, which was essential to accomplishing promising results. In a public dataset,[5] outstanding results (i.e., an overall accuracy of 97.30%) were achieved using less than 200 images for training the Fast-YOLO and CR-NET models, significantly outperforming both baselines.

The CR-NET and Multi-Task models achieved impressive FPS rates on a high-end graphic card. When considering the time spent in the detection stage, it is possible to process 185 and 250 FPS using the CR-NET and Multi-Task models, respectively. Therefore, these approaches can be employed (taking a few seconds) in low-end setups or even in some mobile phones.

As future work, we intend to create an extension of the UFPR-AMR dataset with more than

10,000 images of meters of different types and under different conditions acquired by the company's employees to perform a more realistic analysis of deep learning techniques in the AMR context. Additionally, we plan to explore the meter's model in the AMR pipeline and investigate in depth the cases where the counter has rotating digits since this is one of the main causes of errors in AMR.

Acknowledgments

This work was supported by grants from the National Council for Scientific and Technological Development (CNPq) (# 428333/2016-8, # 311053/2016-5 and # 313423/2017-2), the Minas Gerais Research Foundation (FAPEMIG) (APQ-00567-14 and PPM-00540-17) and the Coordination for the Improvement of Higher Education Personnel (CAPES) (DeepEyes Project). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. We also thank the Energy Company of Paraná (Copel) for allowing one of the authors (Victor Barroso) to collect the images for the UFPR-AMR dataset.

References

1 D. Shu, S. Ma, and C. Jing, "Study of the automatic reading of watt meter based on image processing technology," in *IEEE Conference on Industrial Electronics and Applications*, 2214–2217 (2007).

2 I. Gallo, A. Zamberletti, and L. Noce, "Robust angle invariant GAS meter reading," in *International Conference on Digital Image Computing: Techniques and Applications*, 1–7 (2015).

3 Y. Gao, C. Zhao, J. Wang, *et al.*, "Automatic watermeter digit recognition on mobile devices," in *Internet Multimedia Computing and Service*, 87–95, Springer Singapore (2018).

4 Y. Kabalci, “A survey on smart metering and smart grid communication,” *Renewable and Sustainable Energy Reviews* **57**, 302 – 318 (2016).

5 M. Vanetti, I. Gallo, and A. Nodari, “Gas meter reading from real world images using a multi-net system,” *Pattern Recognition Letters* **34**(5), 519–526 (2013).

6 M. Cerman, G. Shalunts, and D. Albertini, “A mobile recognition system for analog energy meter scanning,” in *Advances in Visual Computing*, 247–256, Springer International Publisher (2016).

7 D. Quintanilha *et al.*, “Automatic consumption reading on electromechanical meters using HoG and SVM,” in *Latin American Conference on Networked and Electronic Media*, 11–15 (2017).

8 V. C. P. Edward, “Support vector machine based automatic electric meter reading system,” in *IEEE International Conference on Computational Intelligence and Computing Research*, 1–5 (2013).

9 Y. Zhang, S. Yang, X. Su, *et al.*, “Automatic reading of domestic electric meter: an intelligent device based on image processing and ZigBee/Ethernet communication,” *Journal of Real-Time Image Processing* **12**, 133–143 (2016).

10 Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature* **521**(7553), 436 (2015).

11 L. Gómez, M. Rusiñol, and D. Karatzas, “Cutting sayre’s knot: Reading scene text without segmentation. application to utility meters,” in *13th IAPR International Workshop on Document Analysis Systems (DAS)*, 97–102 (2018).

12 J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” *IEEE Signal Processing Letters* **24**, 279–283 (2017).

13 J. C. Gonçalves, “Reconhecimento de dígitos em imagens de medidores de consumo de gás natural utilizando técnicas de visão computacional,” Master’s thesis, Universidade Tecnológica Federal do Paraná - UTFPR (2016).

14 J. Redmon, S. Divvala, R. Girshick, *et al.*, “You only look once: Unified, real-time object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016).

15 S. M. Silva and C. R. Jung, “Real-time brazilian license plate detection and recognition using deep convolutional neural networks,” in *30th Conference on Graphics, Patterns and Images (SIBGRAPI)*, 55–62 (2017).

16 G. R. Gonçalves, M. A. Diniz, R. Laroca, *et al.*, “Real-time automatic license plate recognition through deep multi-task networks,” in *31th Conference on Graphics, Patterns and Images (SIBGRAPI)*, 110–117 (2018).

17 B. Shi, X. Bai, and C. Yao, “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 2298–2304 (2017).

18 S. Du, M. Ibrahim, M. Shehata, *et al.*, “Automatic license plate recognition (ALPR): A state-of-the-art review,” *Trans. on Circuits and Systems for Video Technology* **23**, 311–325 (2013).

19 D. Karatzas *et al.*, “ICDAR 2015 competition on robust reading,” in *International Conference on Document Analysis and Recognition (ICDAR)*, 1156–1160 (2015).

20 A. Anis, M. Khaliluzzaman, M. Yakub, *et al.*, “Digital electric meter reading recognition based on horizontal and vertical binary pattern,” in *Int. Conference on Electrical Information and Communication Technology*, 1–6 (2017).

21 S. Zhao, B. Li, J. Yuan, *et al.*, "Research on remote meter automatic reading based on computer vision," in *IEEE PES Transmission and Distribution Conference and Exposition*, 1–4 (2005).

22 L. A. Elrefaei, A. Bajaber, S. Natheir, *et al.*, "Automatic electricity meter reading based on image processing," in *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 1–5 (2015).

23 M. Rodriguez, G. Berdugo, D. Jabba, *et al.*, "HD MR: a new algorithm for number recognition in electrical meters," *Turkish Journal of Elec. Engineering & Comp. Sciences* **22**, 87–96 (2014).

24 R. Smith, "An overview of the Tesseract OCR Engine," in *International Conference on Document Analysis and Recognition*, **2**, 629–633 (2007).

25 A. Nodari and I. Gallo, "A multi-neural network approach to image detection and segmentation of gas meter counter.," in *IAPR Conference on Machine Vision Applications*, 239–242 (2011).

26 L. Zhao, Y. Zhang, Q. Bai, *et al.*, "Design and research of digital meter identifier based on image and wireless communication," in *International Conference on Industrial Mechatronics and Automation*, 101–104 (2009).

27 J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks* **61**, 85–117 (2015).

28 Copel, "Energy Company Of Paraná." `http://www.copel.com/hpcopel/english/`. Accessed: 2018-04-24.

29 G. R. Gonçalves, S. P. G. da Silva, D. Menotti, *et al.*, "Benchmark for license plate character segmentation," *Journal of Electronic Imaging* **25**(5) (2016).

30 R. Laroca, E. Severo, L. A. Zanlorensi, *et al.*, "A robust real-time automatic license plate recognition based on the yolo detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–10 (2018).

31 B. Wu, F. Iandola, P. H. Jin, *et al.*, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 446–454 (2017).

32 S. Tripathi, G. Dane, B. Kang, *et al.*, "LCDet: Low-complexity fully-convolutional neural networks for object detection in embedded systems," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 411–420 (2017).

33 J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525 (2017).

34 M. Everingham, L. Van Gool, C. K. I. Williams, *et al.*, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision* **88**, 303–338 (2010).

35 J. Deng, W. Dong, R. Socher, *et al.*, "ImageNet: A large-scale hierarchical image database," in *Conference on Computer Vision and Pattern Recognition*, 248–255 (2009).

36 AlexeyAB, "YOLOv2 and YOLOv3: how to improve object detection."

37 B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**, 2189–2202 (2012).

38 J. Špaňhel, J. Sochor, R. Juránek, *et al.*, "Holistic recognition of low quality license plates by

CNN using track annotated data," in *IEEE Intern. Conference on Advanced Video and Signal Based Surveillance*, 1–6 (2017).

39 F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: continual prediction with LSTM," in *International Conference on Artificial Neural Networks*, **2**, 850–855 vol.2 (1999).

40 A. Graves, S. Fernández, F. Gomez, *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine Learning (ICML)*, 369–376 (2006).

41 J. Redmon, "Darknet: Open source neural networks in C." http://pjreddie.com/darknet/ (2013–2019).

42 A. Paszke *et al.*, "Automatic differentiation in PyTorch," (2017).

43 F. Chollet *et al.*, "Keras." https://keras.io (2015).

Rayson Laroca received his bachelor's degree in software engineering from the State University of Ponta Grossa, Brazil. Currently, he is a master's student at the Federal University of Paraná, Brazil. His research interests include machine learning, pattern recognition, and computer vision.

Victor Barroso is an undergraduate student in computer science at the Federal University of Paraná, Brazil. His research interests include machine learning, computer vision, pattern recognition and its applications.

Matheus A. Diniz is a master's student at the Federal University of Minas Gerais, Brazil, where he also received his bachelor's degree in computer science. His research focuses on deep learning techniques applied to computer vision and surveillance.

Gabriel R. Gonçalves is a PhD student at Federal University of Minas Gerais. He received a bachelor's degree in computer science from Federal University of Ouro Preto, Brazil and a master's degree in computer science from the Federal University of Minas Gerais, Brazil. His research interests includes machine learning, computer vision and pattern recognition, specially applied to smart surveillance tasks.

William Robson Schwartz is an associate professor in the Department of Computer Science at the Federal University of Minas Gerais, Brazil. He received a PhD from the University of Maryland, College Park, Maryland, USA. His research interests include computer vision, smart surveillance, forensics, and biometrics, in which he authored more than 100 scientific papers and coordinated projects sponsored by several Brazilian Funding Agencies. He is also the head of the Smart Surveillance Interest Group.

David Menotti is an associate professor at the Federal University of Paraná, Brazil. He received his BS and MS degrees in computer engineering and applied informatics from the Pontifical Catholic University of Paraná, Brazil, in 2001 and 2003, respectively, and his PhD degree in computer science from the Federal University of Minas Gerais, Brazil, in 2008. His research interests include machine learning, image processing, pattern recognition, computer vision, and information retrieval.