IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 19, 2021, accepted April 20, 2021, date of publication May 4, 2021, date of current version May 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077415

Towards Image-Based Automatic Meter Reading in Unconstrained Scenarios: A Robust and Efficient Approach

RAYSON LAROCA, ALESSANDRA B. ARAUJO, LUIZ A. ZANLORENSI, EDUARDO C. DE ALMEIDA, AND DAVID MENOTTI, (Senior Member, IEEE)
Department of Informatics, Federal University of Paraná, Curitiba 81530-000, Brazil

Corresponding author: Rayson Laroca (rblsantos@inf.ufpr.br)

This work was supported in part by the Coordination for the Improvement of Higher Education Personnel (CAPES) (Social Demand Program) under Grant 88887.516264/2020-00, and in part by the National Council for Scientific and Technological Development (CNPq) under Grant 308879/2020-1.

ABSTRACT Existing approaches for image-based Automatic Meter Reading (AMR) have been evaluated on images captured in well-controlled scenarios. However, real-world meter reading presents unconstrained scenarios that are way more challenging due to dirt, various lighting conditions, scale variations, in-plane and out-of-plane rotations, among other factors. In this work, we present an end-to-end approach for AMR focusing on unconstrained scenarios. Our main contribution is the insertion of a new stage in the AMR pipeline, called *corner detection and counter classification*, which enables the counter region to be rectified – as well as the rejection of illegible/faulty meters – prior to the recognition stage. We also introduce a publicly available dataset, called Copel-AMR, that contains 12,500 meter images acquired in the field by the service company's employees themselves, including 2,500 images of faulty meters or cases where the reading is illegible due to occlusions. Experimental evaluation demonstrates that the proposed system, which has three networks operating in a cascaded mode, outperforms all baselines in terms of recognition rate while still being quite efficient. Moreover, as very few reading errors are tolerated in real-world applications, we show that our AMR system achieves impressive recognition rates (i.e., $\geq$99%) when rejecting readings made with lower confidence values.

INDEX TERMS Computer vision, image-based automatic meter reading, unconstrained scenarios.

I. INTRODUCTION

Automatic Meter Reading (AMR) refers to the technology whose goal is to automatically record the consumption of electric energy, gas and water for both monitoring and billing [1], [2]. Although smart meters are gradually replacing old meters, in many regions (especially in developing countries) the reading is still done manually in the field, on a monthly basis, by an employee of the service company who takes a picture as reading proof [3], [4].

As such a procedure is prone to errors [5]–[7], the picture needs to be verified by another employee in some situations, for example, when the consumer makes a complaint about the amount charged and when the registered consumption differs significantly from that consumer's average. This offline checking is known to be a laborious task [8], [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani.

In this context, image-based techniques for AMR are much needed, especially taking into account that it is not feasible to quickly replace old meters with smart ones [10]–[12]. The idea behind image-based AMR, which is an specific scenario of scene text detection and recognition, is that the aforementioned inspection can be carried out automatically, reducing mistakes introduced by the human factor and saving manpower [3], [13]. As pointed out by Salomon *et al.* [14], the consumers themselves can capture photos of meters using a mobile device (e.g., a cell phone or a tablet). This eliminates the need for employees of the service company traveling around to perform local meter reading at each consumer unit, resulting in cost savings (especially in rural areas).

Although AMR (hereinafter AMR refers to image-based AMR) has received great attention in recent years, most works in the literature are still limited in several ways. In general, the experiments were performed either on

proprietary datasets [4], [10], [15] or on datasets containing images captured on well-controlled environments [3], [11], [16]. This is in stark contrast to related research areas, such as automatic license plate recognition, where in recent years the research focus shifted to unconstrained scenarios (with challenging factors such as blur, various lighting conditions, scale variations, in-plane and out-of-plane rotations, occlusions, etc.) [17], helping to advance the state of the art considerably. In addition, there are many works focused on a single stage of the AMR pipeline [7], [12], [18], which makes it difficult to accurately evaluate the presented methods in an end-to-end manner (e.g., the results achieved by a recognition model may vary considerably depending on how accurate the counter region is detected). Another factor that makes it difficult to assess existing methods, or their applicability, is that the authors commonly do not report the execution time of the proposed approaches or the hardware in which they performed their experiments [11], [16], [18]. Finally, to the best of our knowledge, no previous work dealt with cases where it is not possible to perform the meter reading due to occlusions or faulty meters, even though such cases are relatively common in practice.

Considering the above discussion, in this work we present a novel end-to-end approach for AMR that leverages the high capability of Convolutional Neural Networks (CNNs) to achieve impressive results on real-world scenarios while still being quite efficient – it is capable of processing 55 frames per second (FPS) on a high-end Graphics Processing Unit (GPU). For our system to be both robust and efficient, we focused on achieving the best speed/accuracy trade-off at each stage when designing it. Our main contribution is the insertion of a new stage in the AMR pipeline, called *corner detection and counter classification*, where a multi-task network detects the four corners of the counter and simultaneously classifies it as legible/operational or illegible/faulty. Prior to the recognition stage, legible counters are rectified using the predicted positions of the corners, thus improving the results obtained in distorted/inclined counters due to oblique views, and illegible counters are rejected. We remark that while improving the recognition performance has an important role in reducing manual intervention, automatically classifying and filtering out illegible meter readings is of paramount importance to the service company, as such cases still require human review.

As part of this work, we introduce a publicly available dataset,[1] called Copel-AMR, that contains 12,500 meter images acquired in the field by the service company's employees themselves (i.e., the images were taken in real-world conditions), including 2,500 images of faulty meters or cases where the reading is illegible. More specifically, we consider as *faulty* the meters where it is not possible to perform the meter reading because no reading is displayed (e.g., electronic meters where the display screen is blank) and as *illegible* the meter images where it is not possible to perform the meter reading due to occlusions in the counter region (e.g., dirt, reflections, and/or water vapor on the meter glass). To the best of our knowledge, this is the first public dataset for end-to-end AMR captured "in the wild" and also the only one with images of illegible/faulty meters. The proposed dataset has six times more images – and contains a larger variety in different aspects – than the largest dataset found in the literature for the evaluation of end-to-end AMR methods.

We experimentally evaluate the proposed approach in two public datasets: UFPR-AMR [3] and Copel-AMR (described in Section III). Our system achieves state-of-the-art results by outperforming 10 deep learning-based baselines in both datasets (we are not aware of any work in the AMR literature where so many methods were evaluated in the experiments). The importance of the *corner detection and counter classification* stage is demonstrated, as our system made 34% fewer reading errors in the legible/operational meters of the Copel-AMR dataset – where the images were captured in unconstrained scenarios – when feeding rectified counters into the recognition network. Moreover, simultaneously to the prediction of the counter corners, our network is able to filter out most of the illegible/faulty meters (i.e., 98.9%), thereby reducing the overall cost of the proposed system since the counter rectification and recognition tasks are skipped in such cases, while correctly accepting 99.82% of the legible/operational meters.

In summary, our paper has three main contributions:

- A robust and efficient approach for AMR that achieves state-of-the-art results in two public datasets and that significantly reduces the number of images that are sent to human review by filtering out most images containing illegible/faulty meters. Our system explores three carefully designed and optimized networks, operating in a cascaded mode, to achieve the best trade-off between accuracy and speed.
- A public dataset for end-to-end AMR with 12,500 fully-annotated images acquired on real-world scenarios by the service company's employees themselves, being 10,000 of them of legible/operational meters and 2,500 of illegible/faulty meters. The dataset contains a well-defined evaluation protocol to assist the development of new approaches for AMR as well as the fair comparison among published works;
- A comparative assessment of the proposed approach and 10 baseline methods based on deep learning, unlike most works in the literature that reported only the results obtained by the proposed methods or compared them exclusively with traditional approaches – often carrying out experiments exclusively on proprietary datasets. It is observed that most of the reading errors made by our AMR system occurred in challenging cases, where even humans can make mistakes, as one digit becomes very similar to another due to artifacts in the counter region.

[1]The Copel-AMR dataset is publicly available to the research community at https://web.inf.ufpr.br/vri/databases/copel-amr/. Access is granted **upon request**, i.e., interested parties must register by filling out a registration form and agreeing to the dataset's terms of use.

The remainder of this paper is organized as follows. We review related works in Section II. The Copel-AMR dataset is introduced in Section III. In Section IV, we present the proposed approach in detail. The experiments carried out and the results achieved are described in Section V. Conclusions and future works are given in Section VI.

II. RELATED WORK

Image-based AMR is a specific application of scene text detection and recognition [15], [19]. However, there are some fundamental differences for the general task of detecting and recognizing scene text that should be highlighted: (i) in the AMR context, there is only one region of interest in each image (i.e., the counter region) and all digits are within it; (ii) recognition networks for AMR need to learn 10 classes (digits 0 to 9) while networks for general scene text recognition need to learn 36 character classes (26 letters and 10 digits; the number of classes can be even higher depending on the language); and (iii) AMR presents an unusual challenge in Optical Character Recognition (OCR): rotating digits in electromechanical meters. Typically, rotating digits are a major cause of errors in such meters, even when robust approaches are employed for digit/counter recognition [3], [19]. This last point further emphasizes the importance of the proposed dataset, as recognition models trained exclusively on images from general datasets for robust reading (e.g., ICDAR 2013 [20]) are likely to fail in these cases.

Over the last decade, a number of methods have been proposed for image-based AMR. Prior to the widespread adoption of deep learning in computer vision, most approaches to this task explored image enhancement techniques and handcrafted features with a similar pipeline, i.e., (i) counter detection followed by (ii) digit segmentation and (iii) digit recognition [5], [6]. Most limitations of such methods may be attributed to the fact that handcrafted features are easily affected by noise and are generally not robust to images captured under unconstrained environments [7], [11], [21].

In deep learning-based methods [4], [15], [18], on the other hand, usually the entire counter region is fed into the recognition network and all digits are predicted simultaneously (instead of first segmenting and then recognizing each of them). As major advances have been achieved in computer vision through deep learning [22], in this section we review works that employed deep learning-based approaches in the AMR context.

We also focus on studies related to digit-based meters, even though there are some recent works that addressed the recognition of dial meters [13], [14], [23]. Such works usually explore the angle between the pointer and the dial to perform the reading.

Object detectors have been explored frequently to deal with counter detection. For example, Koščević & Subašić [10] employed Faster R-CNN [24] to detect counters and serial numbers on images of residential meters, whereas Tsai *et al.* [12] applied and fine-tuned Single Shot MultiBox Detector (SSD) [25] for counter detection in electricity meters. In both studies, only proprietary datasets were used to evaluate the detectors.

Similarly, Laroca *et al.* [3] tackled counter detection using Fast-YOLOv2 [26]. Considering that all counter regions were correctly detected in their experiments, the authors stated that very deep models are not necessary to handle this task. For counter recognition, three CNN-based approaches were evaluated, with the CR-NET model [27] outperforming two segmentation-free models [28], [29] in terms of recognition rate. It should be noted that the images they used were acquired in a warehouse of the service company by one of the authors; in other words, the images are not as unconstrained as those collected in the field by the service company's employees (e.g., there is no external lighting or occlusions caused by railings or vegetation).

Rather than exploring object detectors, Calefati *et al.* [30] employed a Fully Convolutional Network (FCN) for semantic segmentation [31] to handle the detection stage. Then, the counter region was aligned horizontally through the application of traditional image-processing techniques, such as contours extraction and mathematical morphology, in the segmentation mask. Finally, a CNN model was employed to produce the meter reading from the aligned counter region. Although their experiments were carried out on real-world images, only a cropped version of their dataset is available for the research community (as only the region containing the digits was kept in each image, it is not possible to use the released dataset for the evaluation of end-to-end methods). In addition, the accuracy rates obtained in some digit positions were significantly lower than in others due to the low variability in such positions (their dataset is biased and so is their recognition model; this phenomenon was also observed by Laroca *et al.* [3]). Such a limitation must be addressed before an AMR solution can be used in practice.

Yang *et al.* [18] combined an FCN and Connectionist Temporal Classification (CTC) without any intermediate recurrent connections for counter recognition in water meter images. Their network achieved better recognition results than two baselines, showing that such a network is capable of learning contextual information and thus eliminating the need for recurring layers. However, it is important to note that their experiments were carried out only on manually-cropped counter regions and that such a segmentation-free approach may not be as robust in cases where the region of interest (here, the counter) is not detected as precisely [32].

Taking into account the importance of designing highly efficient methods in the AMR context, Li *et al.* [11] proposed a light-weight CNN for counter recognition that splices a certain number of 1×1 and 3×3 kernels to reduce the network parameters with little loss in the recognition rate. The results reported by them are impressive considering the accuracy/speed trade-off obtained; nevertheless, their experiments were performed exclusively on a private dataset with well-controlled images quite similar to each other (i.e., the images were captured by a camera installed in the meter box and preprocessed manually by the authors; thus, they have no blur,

FIGURE 1. Some images extracted from the Copel-AMR dataset. Note that there are both electromechanical and electronic meters and that meters of different types/models often have different screen sizes and layouts. The last two images in each row are from faulty or illegible meters. As requested by Copel, the regions containing consumer identification were blurred on each image due to privacy constraints.

scale variations, shadows, occlusions, significant rotations, among other challenging factors).

Marques *et al.* [4] fine-tuned the Faster R-CNN and RetinaNet [33] object detectors for counter recognition. Although the authors reported mean Average Precision (mAP) rates above 90% with both detectors, only a small subset of counter images from a private dataset was employed in their experiments and the hardware used (i.e., the GPU) was not specified, making it difficult to compare their methodology with previous works both in terms of efficiency and recognition rate. Waqar *et al.* [7] also employed Faster R-CNN for counter recognition, however, a low recognition rate of 76% was reported in their experiments. As in [30], the accuracy achieved in some digit positions was considerably lower than in others, probably due to the fact that the authors did not take into account the bias in the distribution of the digit classes in the training set when fine-tuning the Faster R-CNN model. Despite the fact that the authors claimed that their method can be deployed in real-time applications, no experiments related to execution time were performed/reported.

There are some works in which the authors chose to perform the meter reading directly in the input image, i.e., without counter detection. For instance, Liao *et al.* [21] simply employed YOLOv3 [34] for this task, whereas Gómez *et al.* [15] proposed a CNN model that directly outputs the meter reading in a segmentation-free manner. Although promising results were reported in these works, such approaches are not robust to severe perspective distortions and small-meter images [3], [30].

Considering the many limitations of existing works, we propose a novel end-to-end AMR system that contains a unified approach for *corner detection and counter classification* in order to (i) improve the recognition results (especially in unconstrained scenarios) through counter rectification and (ii) significantly reduce the number of images that are sent to human review by filtering out images containing illegible/faulty meters. The proposed system is empirically evaluated in two public datasets that have well-defined evaluation protocols and that enable the evaluation of end-to-end AMR methods. One of the datasets, called UFPR-AMR [3], has $4K$ images collected in a warehouse of the service company by one of its authors, i.e., under controlled capture conditions, while the other (introduced in Section III) contains $480p$ images acquired in the field by the service company's employees themselves, i.e., under unconstrained capture environments. In our experiments, detailed information regarding both the hardware/frameworks used and the execution time required to run our AMR system is also provided in order to enable an accurate analysis of its speed/accuracy trade-off, as well as its applicability.

III. THE COPEL-AMR DATASET

The Copel-AMR dataset contains 12,500 meter images acquired in the field by the employees of the Energy Company of Paraná (Copel), which directly serves more than 4 million consuming units, across 395 cities and 1,113 locations (i.e., districts, villages and settlements), located in the Brazilian state of Paraná [35]. Thus, Copel-AMR is composed of images captured in unconstrained scenarios, which typically include blur (due to camera motion), dirt, scale variations, in-plane and out-of-plane rotations, reflections, shadows, and occlusions. In 2,500 images (i.e., 20% of the dataset), it is not even possible to perform the meter reading due to occlusions or faulty meters. Although such situations are found on a daily basis by meter readers, there is no work in the literature addressing them or public datasets containing images of illegible/faulty meters, to the best of our knowledge. Figure 1 shows the diversity of the dataset. Note that as the model of the meters being installed/replaced has

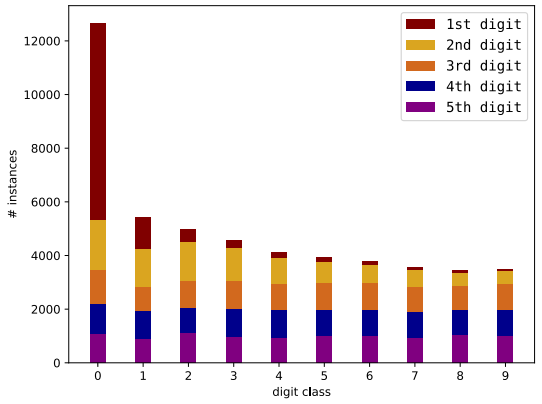


FIGURE 2. Distribution of the digit classes in the Copel-AMR dataset. In the AMR context, it is common that the digit '0' has many more instances than the others, as a brand new meter starts with 00000.

changed over the years, there is a wide variety of meter types in our dataset.

The images have a resolution of 480 × 640 or 640 × 480 pixels, depending on the orientation in which they were taken. Considering that the meter is operational and that there are no occlusions, these resolutions are enough for the meter reading to be legible.

For each image in our dataset, we manually labeled the meter reading, the position (x, y) of each of the four corners of the counter, and a bounding box (x, y, w, h) for each digit. Corner annotations – which can be converted to a bounding box – enable the counter to be rectified, while bounding boxes enable the training of object detectors as well as the application of a wider range of data augmentation techniques. As far as we are aware, the Copel-AMR dataset is the only one to provide so much labeled information for each image.

As the Copel-AMR dataset contains 10,000 images of legible/operational meters and each meter reading consists of 5 digits, we manually labeled a total of 50,000 digits. The distribution of the digit classes in the dataset is shown in Figure 2. We observed that the digit '0' has many more instances than the others, which was expected, due to the fact that a brand new meter starts with 00000 and the leftmost digit positions take longer to be increased.

In electromechanical meters, it is possible that the digits (usually, the rightmost one) are rotating (see an example in the 3rd row and 4th column in Figure 1). In such cases, following the protocol adopted at Energy Company of Paranà (Copel), we considered the lowest digit as the ground truth (e.g., a digit rotating from '4' to '5' is labeled as '4'), except between digits '9' and '0' where the digit should be labeled as '9'.

With the advances of deep learning-based techniques and the availability of ever larger datasets, in many cases it is time-consuming to divide the datasets multiple times and then average the results among multiple runs. Hence, public datasets introduced in recent years commonly have a single division of the images into training, validation and test sets [3], [14]. In the same direction, we randomly split the Copel-AMR dataset as follows: 5,000 images for training, 5,000 images for testing and 2,500 images for validation, following the split protocol (i.e., 40%/40%/20%) used in the UFPR-AMR dataset. We preserved the percentage of samples for illegible/faulty meters, that is, there are 1,000 images of illegible/faulty meters in each of the training and test sets, and 500 images in the validation one. For reproducibility purposes, the subsets generated are explicitly available along with the proposed dataset.

IV. PROPOSED APPROACH

As illustrated in Figure 3, the proposed approach consists of three main stages: (i) counter detection, (ii) corner detection and counter classification, and (iii) counter recognition.[2] Given an input image, the counter region is located using a modified version of the Fast-YOLOv4 model, called Fast-YOLOv4-SmallObj. Then, in a single forward pass of the proposed *Corner Detection and Counter Classification Network (CDCC-NET)*, the cropped counter is classified as legible/operational or illegible/faulty and the position (x, y) of each of its corners is predicted. Finally, illegible counters are rejected, while legible ones are rectified and fed into our recognition network, called Fast-OCR.

In the remainder of this section, each stage of the proposed system is better described. It is worth noting that, for each stage, we train a single network on images from both datasets in which we perform experiments (see Section V-A). In this way, our networks become robust to images captured under different conditions with significantly less manual effort, as the network parameters are adjusted only once.

A. COUNTER DETECTION

In unconstrained scenarios, locating the corners (2D points) of the counter directly in the input image is a challenging task for three main reasons: (i) one or more corners may not be visible due to occlusions caused by dirt, reflections, and other factors; (ii) the counter region may occupy a very small portion of the input image, as can be seen in Figure 1; and (iii) some text blocks (e.g., meter specifications and serial number) are very similar to the counter region in certain meter models. Therefore, we first locate the counter in the input image and then detect its corners in the cropped patches.

As the counter region is rectified prior to the recognition stage in our system's pipeline, the counter detector does not need to be very sophisticated or rely on very deep models. In other words, our AMR system can tolerate less accurate detections of the counter region, as the corners will be later located and the counter rectified. Taking this into account as well as the importance of having an efficient system in real-world applications, we initially decided to use the Fast-YOLOv4 model [36] for this task since, despite being much smaller than state-of-the-art object detectors, Fast-YOLO variants (also known as Tiny-YOLO) are still able to detect

[2]The entire system, i.e., the architectures and weights, is publicly available at https://web.inf.ufpr.br/vri/publications/amr-unconstrained-scenarios/.

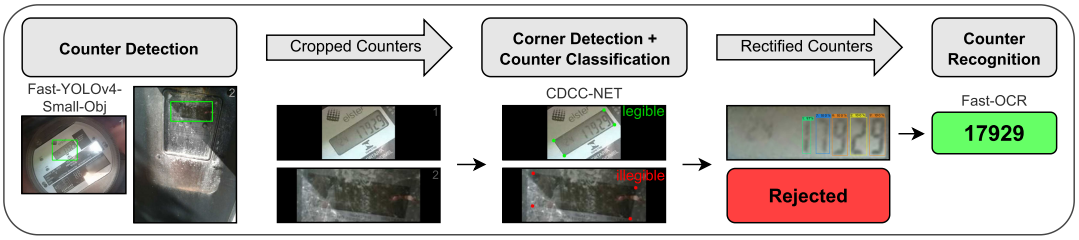


FIGURE 3. The pipeline of the proposed AMR system. Given an input image, the counter region is located using a modified version of the Fast-YOLOv4 model. Then, in a single forward pass of the proposed CDCC-NET, the cropped counter is classified as legible/operational or illegible/faulty and the position (x, y) of each of its corners is predicted. Finally, illegible counters are rejected, whereas legible ones are rectified and fed into our recognition network, called Fast-OCR.

some objects quite precisely [37] and have been employed in various research areas in recent years [14], [38], [39].

Nevertheless, we noticed in preliminary experiments that Fast-YOLOv4 failed in some cases where the meter was relatively far from the camera (usually in images where the reading is illegible). Therefore, we performed some modifications to the network in order to improve its performance in detecting small objects. More specifically, following insights from [34], [40], [41], we added a few layers to the network so that it predicts bounding boxes at 3 different scales instead of 2. This was done by (i) taking the feature map from the next-to-last layer and upsampling it by a factor of 2; (ii) concatenating a feature map from earlier in the network with the upsampled features; and (iii) adding some convolutional layers to process this combined feature map and predict a similar tensor but with twice the size. Table 1 shows the modified architecture, which hereinafter is referred to as *Fast-YOLOv4-SmallObj*. Observe that the final feature map is now 48 × 48 instead of 24 × 24 pixels (for an input size of 384 × 384 pixels), which makes fine details better visible; consequently, small objects can be detected more accurately. There are 18 filters (instead of 255) in layers 15, 22 and 29 so that the network predicts 1 class instead of 80.

The input size of 384 × 384 pixels was chosen based on careful assessments carried out in the validation set, where we sought the best balance between speed and accuracy with different input dimensions (from 320 × 320 to 608 × 608 pixels). It is remarkable that, according to our experiments, for this task, Fast-YOLOv4-SmallObj performed better than the Fast-YOLOv4 model with larger input sizes, while requiring comparable or even less floating-point operations (FLOP) in each forward pass. As an example, Fast-YOLOv4-SmallObj, which requires 6.8 billion floating-point operations (BFLOP), achieved 99.88 mAP in the validation set, whereas Fast-YOLOv4 with an input size of 608 × 608 pixels reached 99.13% mAP while requiring 14.5 BFLOP in each forward pass.

We exploit several data augmentation strategies to train our network, such as random cropping, shearing, conversion to grayscale, and random perturbations of hue, saturation and brightness. As each image contains a single meter, only the detection with the highest confidence value is considered in cases where more than one counter region is predicted.

TABLE 1. The Fast-YOLOv4-SmallObj model, a modified version of Fast-YOLOv4 to improve the detection of small objects. We added layers 38-44 so that the network predicts bounding boxes at 3 different scales (layers 30, 37, and 44) instead of 2 (layers 30 and 37).

#	Layer	Filters	Size	Input	Output
0	conv	32	3 × 3/2	384 × 384 × 3	192 × 192 × 32
1	conv	64	3 × 3/2	192 × 192 × 32	96 × 96 × 64
2	conv	64	3 × 3/1	96 × 96 × 64	96 × 96 × 64
3	route [2]			× 1/2	96 × 96 × 32
4	conv	32	3 × 3/1	96 × 96 × 32	96 × 96 × 32
5	conv	32	3 × 3/1	96 × 96 × 32	96 × 96 × 32
6	route [5, 4]				96 × 96 × 64
7	conv	64	1 × 1/1	96 × 96 × 64	96 × 96 × 64
8	route [2, 7]				96 × 96 × 128
9	max		2 × 2/2	96 × 96 × 128	48 × 48 × 128
10	conv	128	3 × 3/1	48 × 48 × 128	48 × 48 × 128
11	route [10]			× 1/2	48 × 48 × 64
12	conv	64	3 × 3/1	48 × 48 × 64	48 × 48 × 64
13	conv	64	3 × 3/1	48 × 48 × 64	48 × 48 × 64
14	route [13, 12]				48 × 48 × 128
15	conv	128	1 × 1/1	48 × 48 × 128	48 × 48 × 128
16	route [10, 15]				48 × 48 × 256
17	max		2 × 2/2	48 × 48 × 256	24 × 24 × 256
18	conv	256	3 × 3/1	24 × 24 × 256	24 × 24 × 256
19	route [18]			× 1/2	24 × 24 × 128
20	conv	128	3 × 3/1	24 × 24 × 128	24 × 24 × 128
21	conv	128	3 × 3/1	24 × 24 × 128	24 × 24 × 128
22	route [21, 20]				24 × 24 × 256
23	conv	256	1 × 1/1	24 × 24 × 256	24 × 24 × 256
24	route [18, 23]				24 × 24 × 512
25	max		2 × 2/2	24 × 24 × 512	12 × 12 × 512
26	conv	512	3 × 3/1	12 × 12 × 512	12 × 12 × 512
27	conv	256	1 × 1/1	12 × 12 × 512	12 × 12 × 256
28	conv	512	3 × 3/1	12 × 12 × 256	12 × 12 × 512
29	conv	18	1 × 1/1	12 × 12 × 512	12 × 12 × 18
30	**detection**				
31	route [27]				12 × 12 × 256
32	conv	128	1 × 1/1	12 × 12 × 256	12 × 12 × 128
33	upsample		2×	12 × 12 × 128	24 × 24 × 128
34	route [33, 23]				24 × 24 × 384
35	conv	256	3 × 3/1	24 × 24 × 384	24 × 24 × 256
36	conv	18	1 × 1/1	24 × 24 × 256	24 × 24 × 18
37	**detection**				
38	route [35]				24 × 24 × 256
39	conv	64	1 × 1/1	24 × 24 × 256	24 × 24 × 64
40	upsample		2×	24 × 24 × 64	48 × 48 × 64
41	route [40, 15]				48 × 48 × 192
42	conv	128	3 × 3/1	48 × 48 × 192	48 × 48 × 128
43	conv	18	1 × 1/1	48 × 48 × 128	48 × 48 × 18
44	**detection**				

B. CORNER DETECTION AND COUNTER CLASSIFICATION

In order to rectify the cropped counter patch, we need to first locate the four corners of the counter. For this purpose, we designed a *multi-task* network [42], called *CDCC-NET*, that analyzes the counter region detected in the previous stage and predicts 9 outputs: eight float numbers referring to the corner positions (x_0/w, y_0/h, …, x_3/w, y_3/h) and an array containing two float numbers regarding the probability of the counter being legible/operational or illegible/faulty.

We consider as input to the network a counter region slightly larger than the one detected in the previous stage in order to try to ensure that all corners are within the cropped patch even in less accurate detections.

TABLE 2. CDCC-NET's layers and hyperparameters. It is relatively shallow and has two dense layers for each of the 9 outputs (i.e., x_0/w, y_0/h, x_1/w, y_1/h, x_2/w, y_2/h, x_3/w, y_3/h, [legible, illegible]).

#	Layer	Filters	Size	Input	Output
0	conv	16	$3 \times 3/1$	$192 \times 64 \times 3$	$192 \times 64 \times 16$
1	max		$2 \times 2/2$	$192 \times 64 \times 16$	$96 \times 32 \times 16$
2	conv	32	$3 \times 3/1$	$96 \times 32 \times 16$	$96 \times 32 \times 32$
3	max		$2 \times 2/2$	$96 \times 32 \times 32$	$48 \times 16 \times 32$
4	conv	64	$3 \times 3/1$	$48 \times 16 \times 32$	$48 \times 16 \times 64$
5	max		$2 \times 2/2$	$48 \times 16 \times 64$	$24 \times 8 \times 64$
6	flatten			$24 \times 8 \times 64$	12288
#	Layer	Connected to	Units	Input	Output
7	non-shared dense$_{[0..8]}$	#6	128	12288	128
8	non-shared dense_corners$_{[0..7]}$	#7	1	128	1
9	non-shared dense_counter_class$_{[8]}$	#7	2	128	2

CDCC-NET's architecture is shown in Table 2. As can be seen, there are three shared convolutional layers with 16/32/64 filters, each followed by a max-pooling layer with a 2×2 kernel and stride $= 2$. There are also two fully connected (or dense) layers for each of the 9 outputs (i.e., these two layers are not shared). Observe that in the second non-shared dense layer there is a single unit for the prediction of each of the eight corner coordinates (a single float number is predicted for each task), and two units for the prediction of the probabilities of the counter being legible or illegible (here we employed the softmax function to enforce that the sum of the probabilities is equal to 1).

The input size of the CDCC-NET model is 192×64 pixels. These dimensions were defined by halving the input size used in the previous stage, as here the region of interest is already cropped, and by adapting it to the mean aspect ratio of the counters in the Copel-AMR dataset ($w/h \approx 3$). Thus, all images are resized to 192×64 pixels before being fed into the network. However, to avoid distortions when resizing the images, we first add black borders on them so that they have an aspect ratio (w/h) close/equal to 3.

The main difference between CDCC-NET and existing networks for corner detection in other applications, such as license plate recognition [43], [44], is that the proposed network is relatively shallow and has a specific dense layer for predicting each output, while the existing models usually have more intermediate layers with many more filters and a single dense layer to predict all output values. Another approach worth mentioning is that proposed by Lyu *et al.* [45] for multi-oriented scene text detection, where each corner point is redefined and represented by a horizontal square $C = (x_c, y_c, ss, ss)$, where x_c, y_c are the coordinate of a corner point (such as x_1, y_1 for top-left point) as well as the center of the horizontal square (ss is the length short side of the rotated rectangular bounding box). Then, the corner points are detected as default bounding boxes through a model with a backbone adapted from VGG16 [46] containing several extra convolutional layers and a few deconvolution modules.

Considering that the number of training images is still limited to train such a multi-task network (i.e., there is no public dataset for AMR with hundreds of thousands of labeled images) and also the fact that the counter region is well-aligned in most cases (especially in the UFPR-AMR dataset [3]), we created many artificial images through data augmentation in order to prevent overfitting. We performed random variations of hue, saturation and brightness to the original images, in addition to randomly rotating and cropping them. We also randomly permuted the position of the digits on the counters to eliminate undesirable biases in network learning related to the corner positions and certain classes of digits, for example, the network might learn a false correlation between the top-left/bottom-left corners and digits '0' since most occurrences of the class '0' are in the leftmost digit position (see Figure 2). Figure 4 shows some examples of the images generated by us. Note that the bounding box of each digit, which is labeled in our dataset, is required to apply this data augmentation technique.

FIGURE 4. Some representative examples of the images generated by us for training CDCC-NET (and also Fast-OCR; see the next section). The images in the first column are the originals and the others were generated automatically.

After locating the four corners, we rectify the counters classified as legible/operational by calculating and applying a perspective transform from the coordinates of the four corners in the source image (src) to the corresponding vertices in the "unwarped" image (dst). We defined these vertices as follows: (0, 0) indicates the top-left corner; $(max_w - 1, 0)$ corresponds to the top-right corner; $(max_w - 1, max_h - 1)$ is the bottom-right corner; and $(0, max_h - 1)$ refers to the bottom-left corner, where max_w indicates the maximum distance between the bottom-right and bottom-left x coordinates or the top-right and top-left x coordinates, and max_h is the maximum distance between the top-right and bottom-right y coordinates or the top-left and bottom-left y coordinates.

For the sake of completeness, following [47], the 3×3 matrix of the perspective transform is calculated so that:

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = \text{map\_matrix} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

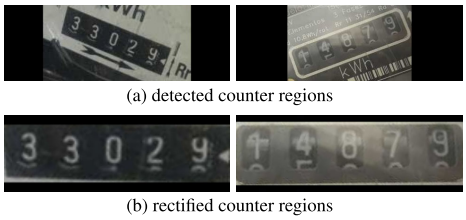
where

$$dst(i) = (x'_i, y'_i), src(i) = (x_i, y_i), i = 0, 1, 2, 3 \tag{2}$$

and the counter region is rectified using the specified matrix:

$$dst(x, y) = src\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right) \tag{3}$$

Counters classified by CDCC-NET as illegible/faulty, on the other hand, are rejected. To the best of our knowledge, in the AMR context, this is the first work in which the region of interest is rectified prior to the recognition stage. As illustrated in Figure 5, rectified counters become more horizontal, tightly-bounded, and easier to read.



(a) detected counter regions

(b) rectified counter regions

FIGURE 5. Two examples of counter regions before and after the rectification process. It should be observed that the rectified counters become more horizontal, tightly-bounded, and easier to read.

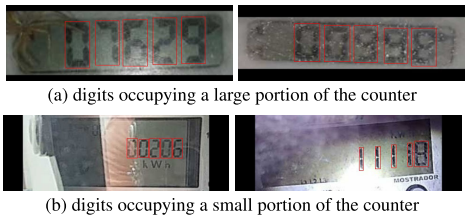
C. COUNTER RECOGNITION

Once the counter region has been located and rectified, the digits must be recognized. To this end, we built a new lightweight detection network, called *Fast-OCR*, that incorporates features from existing models focused on the speed/accuracy trade-off, such as YOLOv2 [26], CR-NET [27] and Fast-YOLOv4. In this work, we handle counter recognition as an object detection problem since object detectors have been successfully applied to several character recognition tasks in recent years [48]–[50]. Accordingly, the Fast-OCR model is trained to predict 10 classes (i.e., 0-9) using the cropped counter patch as well as the class and bounding box (x, y, w, h) of each digit as inputs.

TABLE 3. The architecture of the Fast-OCR model, proposed for counter recognition.

#	Layer	Filters	Size	Input	Output	BFLOP
0	conv	32	$3 \times 3/1$	$384 \times 128 \times 3$	$384 \times 128 \times 32$	0.085
1	max		$2 \times 2/2$	$384 \times 128 \times 32$	$192 \times 64 \times 32$	0.002
2	conv	64	$3 \times 3/1$	$192 \times 64 \times 32$	$192 \times 64 \times 64$	0.453
3	max		$2 \times 2/2$	$192 \times 64 \times 64$	$96 \times 32 \times 64$	0.001
4	conv	128	$3 \times 3/1$	$96 \times 32 \times 64$	$96 \times 32 \times 128$	0.453
5	max		$2 \times 2/2$	$96 \times 32 \times 128$	$48 \times 16 \times 128$	0.000
6	conv	256	$3 \times 3/1$	$48 \times 16 \times 128$	$48 \times 16 \times 256$	0.453
7	conv	128	$1 \times 1/1$	$48 \times 16 \times 256$	$48 \times 16 \times 128$	0.050
8	conv	256	$3 \times 3/1$	$48 \times 16 \times 128$	$48 \times 16 \times 256$	0.453
9	max		$2 \times 2/2$	$48 \times 16 \times 256$	$24 \times 8 \times 256$	0.000
10	conv	512	$3 \times 3/1$	$24 \times 8 \times 256$	$24 \times 8 \times 512$	0.453
11	conv	256	$1 \times 1/1$	$24 \times 8 \times 512$	$24 \times 8 \times 256$	0.050
12	conv	512	$3 \times 3/1$	$24 \times 8 \times 256$	$24 \times 8 \times 512$	0.453
13	conv	45	$1 \times 1/1$	$24 \times 8 \times 512$	$24 \times 8 \times 45$	0.009
14	**detection**					
15	route [11]				$24 \times 8 \times 256$	
16	conv	256	$1 \times 1/1$	$24 \times 8 \times 256$	$24 \times 8 \times 256$	0.025
17	upsample		$2\times$	$24 \times 8 \times 256$	$48 \times 16 \times 256$	
18	route [17, 6]				$48 \times 16 \times 512$	
19	conv	512	$3 \times 3/1$	$48 \times 16 \times 512$	$48 \times 16 \times 512$	3.624
20	conv	45	$1 \times 1/1$	$48 \times 16 \times 512$	$48 \times 16 \times 45$	0.035
21	**detection**					

The architecture of Fast-OCR is shown in Table 3. The input size is 384×128 pixels considering both the number of max-pooling layers in the network (i.e., the dimensions of the output layer must be large enough to enable the detection of multiple digits horizontally spread side by side) and also the mean aspect ratio of the counters in the Copel-AMR dataset ($w/h \approx 3$). As Fast-YOLOv4, the proposed network performs detection at 2 different scales (layers 14 and 21). This is particularly important for counter recognition in unconstrained scenarios due to the fact that the digits may occupy either a small or a large portion of the counter region depending on the meter model, as illustrated in Figure 6, and also on how accurately the counter corners were detected in the previous stage. As in the networks introduced in [26], [27], the convolutional layers in Fast-OCR mostly have 3×3 kernels and the number of filters is doubled after each max-pooling layer. In addition, there are 1×1 convolutional layers between 3×3 convolutions to reduce the feature space from preceding layers.



(a) digits occupying a large portion of the counter

(b) digits occupying a small portion of the counter

FIGURE 6. The digits may occupy either a large (a) or a small (b) portion of the counter region depending on the meter model. The bounding boxes of the digits are outlined in red for better viewing.

In Table 3, we also list the number of FLOP required in each layer to highlight how small the Fast-OCR network is compared to some deeper detection models. For example, for this task, our network requires 6.6 BFLOP while YOLOv3 [34] and YOLOv4 [40] require 65.4 and 59.5 BFLOP, respectively.

It is important to note that, thanks to the versatility and ability of detection networks to learn the general features of objects (here, the digits) regardless of their positions, and also to the confidence value tied to each prediction, Fast-OCR's output can be easily adapted/improved (i.e., without making any changes to its architecture) through post-processing heuristics. For example, a variable number of digits is predicted for each counter patch fed into the network, and some digits can be discarded based on the confidence values which they were predicted or through geometric constraints; in other words, Fast-OCR can be applied – without any modification – to counters with different numbers of digits. In fact, we believe that heuristic rules can also be explored to identify illegible/faulty meters erroneously classified as legible/operational in the previous stage, since it is very likely that counters classified as legible/operational should have been classified as illegible/faulty in cases where Fast-OCR has not predicted any digit with a high confidence value.

As the Fast-OCR model is trained from scratch, many training samples are needed for the network to generalize well. Therefore, in addition to using the original images of the counter region, we exploit the images artificially generated in the previous stage to train Fast-OCR and improve its robustness. Note that using the exactly same images in both

stages is possible since patches of the counter region are fed as input into both CDCC-NET and Fast-OCR models.

V. EXPERIMENTS

A. SETUP AND BASELINES

In this work, we conducted experiments on images from the Copel-AMR and UFPR-AMR [3] datasets. The UFPR-AMR dataset contains 2,000 images acquired in relatively well-controlled environments, with a resolution between 2,340 × 4,160 and 3,120 × 4,160 pixels, and is split into three subsets: training (800 images), validation (400 images) and testing (800 images). In order to train, validate and test our networks, we merge the respective subsets from both datasets (i.e., exactly the same networks are used regardless of which datasets we are running experiments on). As the UFPR-AMR dataset does not have any annotations related to the corners of the counters, we manually labeled their positions in its 2,000 images so that we can use images from both datasets to train/evaluate the CDCC-NET model. All annotations made by us are publicly available to the research community along with the Copel-AMR dataset.

It is important to point out that in several recent works in the literature [10], [15], [21], [30] the authors reported only the results obtained by the proposed methods or compared them exclusively with traditional approaches, overlooking deep learning-based approaches designed for AMR. This makes it difficult to make a fair comparison between recently published works. Taking this into account, in this work, the end-to-end results achieved by the proposed system are compared (both in terms of recognition rate and execution time) with those obtained by several baseline methods [3], [15], [21], [30], [40] trained by us on *exactly* the same images as the proposed method.[3] These specific methods, described in Section II, were chosen/implemented by us for two main reasons: (i) they were recently employed in the context of AMR (except [40], which was recently introduced) with promising/impressive results being reported, and (ii) we believe we have the knowledge necessary to train/adjust them in the best possible way in order to ensure fairness in our experiments, as the authors provided enough details about the architectures used, and also because we designed/employed similar networks (even the same ones in certain cases) in recent works in the context of license plate recognition and related areas [29], [32], [50]. Note that, in our experiments, we adapted all networks so that their input layers have the same aspect ratio ($w/h = 3$).

Regarding the baselines, Gómez *et al.* [15] evaluated their recognition network on a dataset containing mostly images where the counter is well centered and occupies a good portion of the image; therefore, in our experiments we first detect the counter region in the input image with the Fast-YOLOv4-SmallObj model and then apply their network to the detected region. The same was done with the recognition network proposed by Calefati *et al.* [30], as they dealt with the counter detection stage using an FCN for semantic segmentation [31] that we do not have the knowledge necessary to train/adjust in the best possible way. We consider reimplementing/retraining it out of scope for this work since our detection model (i.e., Fast-YOLOv4-SmallObj) is able to achieve high F-measure rates in both datasets (see Section V-B1) and also due to the fact that the segmentation task is generally more time-consuming than the detection one.

The YOLO-based models were trained using the Darknet framework,[4] while the other models were trained using Keras.[5] In Darknet, the following parameters were used: $65K$ iterations (max batches), batch size = 64, and learning rate = $[10^{-3}, 10^{-4}, 10^{-5}]$ with decay steps at $26K$ and $45.5K$ iterations. In Keras, we employed the following parameters: initial learning rate = 10^{-3} (with *ReduceLROnPlateau*'s patience = 3 and factor = 10^{-1}), batch size = 128, max epochs = 100, and patience = 7 (patience refers to the number of epochs with no improvement after which training will be stopped). All networks were trained using the Stochastic Gradient Descent (SGD) optimizer, and all experiments were carried out on a computer with an AMD Ryzen Threadripper 1920X 3.5GHz CPU, 48 GB of RAM, SSD (read: 535 MB/s; write: 445 MB/s), and an NVIDIA Titan V GPU. We remark that all parameter values were defined based on experiments performed in the validation set.

In addition to the baselines mentioned above, we trained and evaluated all models for scene text recognition implemented in [51] (i.e., CRNN [28], RARE [52], R2AM [53], STAR-Net [54], GRCNN [55], Rosetta [56] and TRBA [57]), which is the open source repository (PyTorch[6]) of Clova AI Research used to record the 1st place of ICDAR2013 focused scene text and ICDAR2019 ArT, and 3rd place of ICDAR2017COCO-Text and ICDAR2019 ReCTS (task1). For reasons of space and clarity, instead of reporting the results obtained by all these models, we included in our overall evaluation only the results obtained by the model that performed faster (CRNN [28]) and by the one that obtained the best recognition rates (TRBA [57]) in our experiments.

It is worth noting that, as stated in Section II, recognition models trained exclusively on images from datasets for general robust reading are likely to fail in AMR scenarios due to some domain-specific characteristics (e.g., rotating digits). However, we believe that first pre-training them using images from large-scale scene text recognition datasets and then fine-tuning them on images from AMR datasets can enable even better results to be achieved.

B. RESULTS AND DISCUSSION

In this section, we report the experiments carried out to verify the effectiveness of the proposed AMR system. We first

[3]The architectures and weights of the baselines implemented/trained by us are also publicly available at https://web.inf.ufpr.br/vri/publications/amr-unconstrained-scenarios/.

[4]https://github.com/AlexeyAB/darknet/

[5]https://keras.io/

[6]https://pytorch.org/

assess the counter detection stage separately since the regions used in the following stages are extracted from the detection results, rather than cropped directly from the ground truth – note that a detection failure probably leads to another failure in the subsequent stages. Similarly, we then report the results reached by CDCC-NET in both corner detection and counter classification tasks. Finally, we evaluate the entire AMR system in an end-to-end manner (without any prior knowledge as to which dataset each test image belongs to or whether it is from a legible/operational or illegible/faulty meter) and compare the reading results achieved in legible/operational images with those obtained by 10 baseline methods.

1) COUNTER DETECTION

Detection tasks in the AMR context are often evaluated by considering a predicted bounding box to be correct if its Intersection over Union (IoU) with the ground truth is greater than 50% [3], [14]. Nevertheless, such a low threshold (IoU $>$ 0.5) was deliberately defined by Everingham *et al.* [58] to account for inaccuracies in bounding boxes in the training data, as defining the bounding box for a highly non-convex object (e.g., a person with arms and legs spread) is somewhat subjective. Taking into account that the counters are convex objects and that they were carefully labeled in both datasets, in Table 4 we report the performance (in terms of F-measure) of the Fast-YOLOv4-SmallObj model over different IoU thresholds, from 0.5 to 0.95, similarly to the COCO [59] primary metric (mAP@IoU = [0.5:0.95]). As we consider only one meter per image, the precision and recall rates are identical.

TABLE 4. F-measure values obtained over different IoU thresholds, from 0.5 to 0.95, in the counter detection stage. Note that Fast-YOLOv4-SmallObj achieves considerably better results at higher IoU thresholds (i.e., 0.8-0.95), which indicates that its predictions are much better aligned with the ground truth.

Model	IoU Threshold						
	0.5	0.6	0.7	0.8	0.9	0.95	0.5:0.95
UFPR-AMR [3]							
Fast-YOLOv4 (608 × 608)	98.1%	97.8%	97.6%	94.6%	75.9%	38.6%	83.8%
Fast-YOLOv4-SmallObj	99.9%	99.9%	99.9%	98.1%	80.0%	38.9%	86.1%
Copel-AMR							
Fast-YOLOv4 (608 × 608)	99.0%	98.3%	95.3%	84.2%	52.2%	17.0%	74.3%
Fast-YOLOv4-SmallObj	99.7%	99.5%	98.8%	96.1%	75.3%	28.8%	83.0%

For comparison purposes, we also list in Table 4 the detection results obtained by the original Fast-YOLOv4 model. Observe that the results achieved by Fast-YOLOv4-SmallObj are considerably better at higher IoU thresholds (i.e., 0.8-0.95), which indicates that the bounding boxes predicted by the modified architecture are much better aligned with the ground truth. This is relevant since although our AMR system can tolerate less accurate detections at this stage, such imprecise predictions may still impair counter rectification and, consequently, counter recognition because one or more corners may not be within the detected bounding box. Considering the detections with IoU $>$ 0.5 with the ground truth as correct, as in some previous works, the Fast-YOLOv4-SmallObj failed in only one image from the UFPR-AMR's test set and in only 14 of the 5,000 test images of the Copel-AMR dataset. Nevertheless, we highlight that it is still possible to correctly perform the subsequent tasks in most cases where our network has failed at this stage, as the four corners are usually within the detected region (especially considering that we use as input to the next stage a counter region slightly larger than the one detected). For the record, we tried to further improve the results achieved at this stage by adding attention modules to each model; however, Fast-YOLOv4's results improved only marginally whereas Fast-YOLOv4-SmallObj's results have not improved at all.

FIGURE 7. Representative samples of counters detected by the Fast-YOLOv4-SmallObj model – images taken under unconstrained conditions, e.g., with significant reflections, rotations, and scale variations.

Some detection results are shown in Figure 7. As can be seen, well-located predictions were attained on meters of different models and on images acquired under unconstrained conditions, that is, with significant reflections, rotations, and scale variations.

2) CORNER DETECTION AND COUNTER CLASSIFICATION

To assess the performance of CDCC-NET in the *corner detection* task, following [44], we report in Table 5 the mean pixel distance between the predicted corner positions and the ground truth on each dataset, in addition to how many FPS the proposed model is capable of processing (we report the average across 10 runs). We normalize the distances by dividing them by the respective image dimensions. To enable a comparative evaluation, we also list in Table 5 the results obtained by three CNN models recently proposed for the detection of corners on license plate images: Smaller-LocateNet [43], LocateNet [43] and Hybrid-MobileNetV2 [44]. For a fair comparison and considering that the classification of the meters as legible/operational or illegible/faulty is essential in the proposed AMR pipeline, we added an output layer (softmax) to each baseline so that they can also perform such classification. We emphasize that, according to our experiments, this additional layer does not significantly affect the results obtained in the corner detection task (the normalized mean pixel distance achieved with and without that layer varied slightly in the fourth decimal place).

As can be seen, CDCC-NET presents the best balance between accuracy and speed among the evaluated models. More specifically, (i) Smaller-LocateNet is considerably less accurate in predicting the corner positions than the other networks, even though it runs faster; (ii) CDCC-NET

TABLE 5. Comparison of the corner detection results obtained by CDCC-NET and three baselines. The proposed model presents similar accuracy to Hybrid-MobileNetV2 but is twice as fast. Also, it performs almost as fast as Smaller-LocateNet, even though it predicts much more accurate corner positions.

Model	FPS	Mean pixel distance between the predicted corners and the ground truth (normalized)		
		UFPR-AMR	Copel-AMR	Average
Smaller-LocateNet [43]	**207**	0.0059	0.0173	0.0116
LocateNet [43]	166	0.0031	0.0098	0.0065
CDCC-NET (Ours)	191	0.0017	0.0055	0.0036
Hybrid-MobileNetV2 [44]	97	**0.0016**	**0.0046**	**0.0031**

is both faster and more accurate in locating the corners than LocateNet; and (iii) CDCC-NET predicts the positions of the corners almost as precisely as Hybrid-MobileNetV2, despite being able to process twice as many FPS.

Figure 8 shows the evolution of the training and validation losses of CDCC-NET over time (we omitted the losses related to counter classification for better viewing). As it can be seen, CDCC-NET learns to predict the position of the four corners simultaneously and converged after 14/15 epochs.

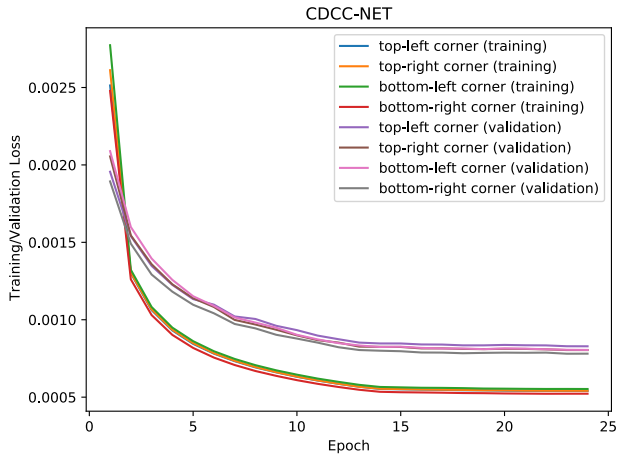


FIGURE 8. Training and validation losses of CDCC-NET over time. For each corner, the loss plotted is the mean between the *x* and *y* coordinates, e.g., the 'top-left corner' loss is the mean between the losses of the x_0 and y_0 tasks.

It should be noted that we evaluated deeper networks in place of CDCC-NET (i.e., with more convolutional layers and/or more filters), however, the end-to-end reading results achieved by the proposed AMR system improved only slightly – not justifying the higher computational cost. In fact, we carried out an experiment in which we rectified all counter regions using the ground-truth annotations, as if the four corners were detected perfectly on each image, and the results improved less than we expected (from 95.87% to 96.53%), implying that most reading errors made by our AMR system were not caused by a poor rectification of the counter region, but by other challenging factors (see Section V-B3 for more information and qualitative results).

TABLE 6. Results achieved by CDCC-NET in the counter classification task. It is able to filter out 98.9% of the illegible/faulty meters, while correctly accepting 99.82% (on average of both datasets) of the legible/operational meters.

Dataset \ Class	Legible/Operational	Illegible/Faulty
UFPR-AMR	99.88%	-
Copel-AMR	99.75%	98.90%
Average	99.82%	98.90%

Table 6 shows the results achieved by CDCC-NET in the *counter classification* task, which is significantly less challenging than corner detection. It is clear that CDCC-NET handles counter classification very well – in images from both datasets – since it correctly filtered out 989 of the 1,000 images in the Copel-AMR's test set (98.9%) where it is not possible to perform the meter reading due to occlusions or faulty meters, thereby reducing the overall cost of the proposed system by skipping the counter rectification and recognition tasks in such cases, while correctly accepting 799 of the 800 images in the UFPR-AMR's test set (99.88%) and 3,990 of the 4,000 legible/operational images in the Copel-AMR's test set (99.75%).

FIGURE 9. Some qualitative results achieved by CDCC-NET in corner detection and counter classification. For better visualization, we draw a polygon from the predicted corner positions. Counters classified as legible/operational are outlined in green while those classified as illegible/faulty are outlined in red.

According to Figure 9, CDCC-NET is able to successfully predict the four corners of the counter and simultaneously classify it as legible/operational or illegible/faulty, regardless of the meter model and other factors that are common in images acquired in uncontrolled environments such as rotations, reflections and shadows.

3) OVERALL EVALUATION (END-TO-END)

In this section, for each dataset, we report the number of correctly recognized counters divided by the number of legible/operational meters in the test set (legible/operational meters classified as illegible/faulty in the previous stage are considered as a reading error of the proposed method). A correctly recognized counter means that all digits on the counter were correctly recognized, as a single digit recognized incorrectly can result in a large reading/billing error. As in the previous stage, to enable an accurate analysis regarding the speed/accuracy trade-off of the evaluated methods, we report

how many FPS each method is capable of processing (in the average of 10 runs), including the time required to load the respective models and weights.

Note that our end-to-end system classifies the counter regions as legible/operational or illegible/faulty in the UFPR-AMR's dataset as well, even though it does not have images labeled as illegible/faulty. This procedure was adopted to better simulate a real-world scenario, where there are images of both legible/operational and illegible/faulty meters.

The results obtained by the proposed system and the baselines are shown in Table 7. As can be seen, our system performed the correct reading of 94.75% of the meters in the UFPR-AMR's test set and 96.98% of the legible/operational meters in the Copel-AMR's test set, considerably outperforming all baselines in terms of recognition rate. It is remarkable that the proposed approach is able to process 55 FPS (i.e., it took ≈18 ms to process each image), meeting the efficiency requirements of real-world applications.

At first, we were surprised by the fact that the recognition rates reached in the proposed dataset were higher than those obtained in the UFPR-AMR dataset [3], where the images were acquired in relatively more controlled conditions. However, through an inspection of the reading errors made by all methods, we noticed some inconsistencies in the way rotating digits were labeled in the UFPR-AMR dataset (not always the lowest digit was chosen as the ground truth). We also observed that the UFPR-AMR's test set contains some images where it is very difficult to perform the correct reading, even for humans, due to factors such as water vapor, reflections and dirt on the meter glass, as well as the poor positioning of the camera by the person who took the photo, causing the digits to appear only partially in the image even though the counter region appears entirely – we emphasize that the UFPR-AMR dataset was collected by one of its authors and not by employees of the service company, unlike the Copel-AMR dataset. In fact, in a few cases, it is even difficult to verify if the labeled reading (i.e., the ground truth) is correct. Although we believe that such images should be rejected by the system due to the great possibility of reading errors (some of these images are shown in Figure 10), we employed the original labels in our evaluations to enable fair comparisons with other works in the literature, since most authors tend to use the annotations originally provided as part of the dataset.

To highlight the importance of rectifying the counter region prior to the recognition stage, we included in Table 7 the results achieved by a modified version of our approach in which the detected counter region is fed directly into Fast-OCR (i.e., without counter rectification). As can be seen, the *corner detection and counter classification* stage is **essential** for accomplishing outstanding results in unconstrained scenarios, as our system made 34% fewer reading errors (i.e., (96.98% − 95.43%)/(100% − 95.43%)) in the legible/operational meters of the Copel-AMR dataset when feeding rectified counters into the recognition network.

FIGURE 10. Some images from the UFPR-AMR dataset in which all evaluated methods failed to correctly perform the reading. In the first three images, there is dirt or water vapor on the meter glass, while in the fourth image the rightmost digit is labeled as '6' and not '5', contrary to the protocol normally adopted.

The end-to-end results achieved by the baselines ranged from 86.09% to 94.73%. As some of them were originally evaluated only on private datasets, it is now possible to assess their applicability – both in terms of speed and accuracy – more accurately. For example, Gómez *et al.* [15] reported a promising recognition rate of 94.17% on a proprietary dataset using their segmentation-free network. Nevertheless, in our comparative assessment, this model reached the lowest recognition rate among the baselines. In general, as observed in [3], the recognition models based on object detectors (e.g., CR-NET and Fast-OCR – which are based on YOLO [37]) performed better than those where counter recognition is done holistically (e.g., [15], [28], [30], [57]).

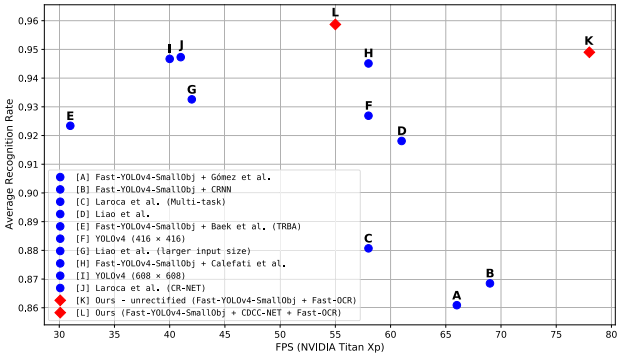


FIGURE 11. Performance comparison of our AMR system with 10 deep learning-based baselines. For better viewing, both the proposed method (L) and its version without counter rectification (K) are depicted as red diamonds, whereas all the baselines (A-J) are depicted as blue circles.

In terms of execution time, we noticed that all methods evaluated are relatively fast, i.e., they are all capable of processing more than 30 FPS on a high-end GPU, especially the version without counter rectification of our system, which is able to process 78 FPS. As can be seen clearly in Figure 11, the proposed system outperformed all baselines in terms of average recognition rate while being relatively fast, and its version without counter rectification is much faster than the baselines that reached similar results.

We highlight that if an AMR system can run at 30+ FPS on a high-end GPU, it probably also works well on cheaper hardware (this is relevant to the service company). In this

TABLE 7. Recognition rates obtained by the proposed AMR system, a version without counter rectification of our system, and 10 baselines in both datasets used in our experiments. We emphasize that exactly the same images were used for training the proposed methods and the baselines.

Approach	FPS	Recognition Rate		
		UFPR-AMR	Copel-AMR	Average
Fast-YOLOv4-SmallObj + Gómez et al. [15]	66	88.25%	83.93%	86.09%
Fast-YOLOv4-SmallObj + CRNN [28]	69	89.62%	84.07%	86.85%
Laroca et al. [3] (Multi-task)	58	89.12%	87.02%	88.07%
Liao et al. [21]	61	91.37%	92.25%	91.81%
Fast-YOLOv4-SmallObj + Baek et al. [57] (TRBA)	31	93.87%	90.80%	92.34%
YOLOv4 (416 × 416) [40]	58	91.63%	93.75%	92.69%
Liao et al. [21] (larger input size)	42	92.25%	94.27%	93.26%
Fast-YOLOv4-SmallObj + Calefati et al. [30]	58	93.87%	95.15%	94.51%
YOLOv4 (608 × 608) [40]	40	94.00%	95.33%	94.67%
Laroca et al. [3] (CR-NET)	41	94.25%	95.20%	94.73%
Ours - unrectified (Fast-YOLOv4-SmallObj + Fast-OCR)	**78**	94.37%	95.43%	94.90%
Ours (Fast-YOLOv4-SmallObj + CDCC-NET + Fast-OCR)	55	**94.75%**	**96.98%**	**95.87%**

FIGURE 12. Examples of meter readings performed correctly by the proposed system. It is remarkable that it performed well in images of meters of different models and captured in unconstrained conditions (e.g., with various lighting conditions, reflections, shadows, scale variations, and considerable rotations).

sense, for simpler/constrained scenarios, we believe that the proposed AMR system can be employed in low-end setups or even in some mobile phones (taking a few seconds).

Figure 12 shows some meter readings performed correctly by the proposed system. It is noticeable that our end-to-end system is able to generalize well, being robust to meters of different models and images captured in unconstrained conditions (e.g., with various lighting conditions, reflections, shadows, scale variations, considerable rotations, etc.).

Some reading errors made by our AMR system are shown in Figure 13. As one may see, they occurred mainly in challenging cases, where one digit becomes very similar to another due to artifacts in the counter region. Another portion of the errors occurred on rotating digits (also called half digits), which is known to be a major cause of errors in electromechanical meters [3], [19], even when robust approaches/models are employed for digit/counter recognition. It should be noted that (i) errors in the least significant digits are tolerable, as they do not significantly impact the amount charged to consumers; and (ii) reading errors in the most significant digits can be filtered by the service company through heuristic rules, for example, the reading must be greater than or equal to the reading taken in the previous month.

Finally, considering that very few reading errors are tolerated by Copel [35] and other service companies, we present

FIGURE 13. Examples of reading errors made by our system. The ground truth is shown in parentheses. Observe that most of the errors occurred in challenging cases, where even humans can make mistakes, as one digit becomes very similar to another due to rotating digits or artifacts in the counter region.

TABLE 8. Recognition rates reached by the proposed AMR system when discarding/rejecting the readings returned with lower confidence values by the Fast-OCR network. Our system achieves impressive recognition rates (i.e., ≥ 99% on average) when using a confidence threshold that rejects 15% of the images.

Dataset	Rejection Rate				
	0%	5%	10%	15%	20%
UFPR-AMR	94.75%	97.63%	98.47%	98.82%	99.22%
Copel-AMR	96.98%	98.61%	99.00%	99.24%	99.44%
Average	95.87%	98.12%	98.74%	99.03%	99.33%

in Table 8 the end-to-end results achieved by the proposed system when discarding/rejecting the readings returned with lower confidence values by the Fast-OCR network (in practice, in a mobile application, the employee would have to capture another image). It is noteworthy that our AMR system achieved an average recognition rate above 98% by rejecting only 5% of the meter readings. Moreover, recognition rates above 99%, which are acceptable to service companies, are achieved by setting a confidence threshold that rejects 15% of the images. In this way, we consider that the proposed approach can be reliably employed on real-world applications.

VI. CONCLUSION AND FUTURE WORK

In this work, we presented an end-to-end, robust and efficient approach for AMR that achieves state-of-the-art results in two public datasets while being able to significantly reduce the number of images that are sent to human review by filtering out images in which it is not possible to perform the meter reading due to occlusions or faulty meters. Our main contribution is the insertion of a new stage in the AMR pipeline, called *corner detection and counter classification*, which enables the counter region to be rectified prior to the recognition stage. As the proposed system made 34% fewer reading errors in the legible/operational meters of the Copel-AMR dataset when feeding rectified counters into the recognition network, we consider this strategy (corner detection + counter rectification) essential for accomplishing outstanding results in unconstrained scenarios.

Our AMR system, which presents three new networks operating in a cascaded mode, performed the correct reading of 94.75% and 96.98% of the meters in the UFPR-AMR and Copel-AMR test sets (outperforming all baselines), respectively, while being able to process 55 FPS on a high-end GPU. It is notable that the proposed approach achieves impressive end-to-end recognition rates (i.e., ≥99%) when discarding/rejecting the readings made with lower confidence values, which is of paramount importance to the service companies since very few reading errors are tolerated in real-world applications due to the fact that a single digit recognized incorrectly can result in a large reading/billing error.

We also introduced a publicly available dataset for AMR with 12,500 fully-annotated images acquired on real-world scenarios by the service company's employees themselves, including 2,500 images of faulty meters or cases where the meter reading is illegible due to factors such as shadows and occlusions. The proposed dataset has six times more images and contains a larger variety in different aspects than the largest dataset found in the literature for the evaluation of end-to-end AMR methods. It also contains a well-defined evaluation protocol to assist the development of new approaches and the fair comparison among published works.

As future work, we plan to design a methodology for the simultaneous detection of the counter region and its corners, aiming to perform counter rectification with an even better speed/accuracy trade-off. We also intend to explore the meter's model/type in the AMR pipeline and investigate in depth the cases where the counter has rotating digits, considering that this is a major cause of reading errors in electromechanical meters. Finally, we want to carry out an extensive

assessment with various approaches/models for detecting and recognizing general scene text to compare how robust and efficient they are for the specific AMR scenario.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation with the donation of the GPUs used for this research. They would also like to thank the Energy Company of Paraná (Copel), in particular the manager of the reading division Dihon Pereira Brandão, for providing the images for the creation of the Copel-AMR dataset.

REFERENCES

[1] T. Khalifa, K. Naik, and A. Nayak, "A survey of communication protocols for automatic meter reading applications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 168–182, 2nd Quart., 2011.

[2] Y. Kabalci, "A survey on smart metering and smart grid communication," *Renew. Sustain. Energy Rev.*, vol. 57, pp. 302–318, May 2016.

[3] R. Laroca, V. Barroso, M. A. Diniz, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "Convolutional neural networks for automatic meter reading," *J. Electron. Imag.*, vol. 28, no. 1, 2019, Art. no. 013023.

[4] R. C. da Silva Marques, A. C. Serra, J. V. F. Franca, J. O. B. Diniz, G. Braz, J. D. S. de Almeida, M. I. A. da Silva, and E. M. G. Monteiro, "Image-based electric consumption recognition via multi-task learning," in *Proc. 8th Brazilian Conf. Intell. Syst. (BRACIS)*, Oct. 2019, pp. 419–424.

[5] M. Vanetti, I. Gallo, and A. Nodari, "GAS meter reading from real world images using a multi-net system," *Pattern Recognit. Lett.*, vol. 34, no. 5, pp. 519–526, Apr. 2013.

[6] I. Gallo, A. Zamberletti, and L. Noce, "Robust angle invariant GAS meter reading," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Nov. 2015, pp. 1–7.

[7] M. Waqar, M. A. Waris, E. Rashid, N. Nida, S. Nawaz, and M. H. Yousaf, "Meter digit recognition via faster R-CNN," in *Proc. Int. Conf. Robot. Automat. Ind. (ICRAI)*, Oct. 2019, pp. 1–5.

[8] M. Cerman, G. Shalunts, and D. Albertini, "A mobile recognition system for analog energy meter scanning," in *Proc. Int. Symp. Vis. Comput.*, 2016, pp. 247–256.

[9] D. B. P. Quintanilha, R. W. S. Costa, J. O. B. Diniz, J. D. S. de Almeida, G. Braz, A. C. Silva, A. C. de Paiva, E. M. Monteiro, B. R. Froz, L. P. A. Piheiro, and W. Melho, "Automatic consumption reading on electromechanical meters using HoG and SVM," in *Proc. 7th Latin Amer. Conf. Netw. Electron. Media (LACNEM)*, 2017, pp. 57–61.

[10] K. Koščević and M. Subašić, "Automatic visual reading of meters using deep learning," in *Proc. Croatian Comput. Vis. Workshop*, 2018, pp. 1–6.

[11] C. Li, Y. Su, R. Yuan, D. Chu, and J. Zhu, "Light-weight spliced convolution network-based automatic water meter reading in smart city," *IEEE Access*, vol. 7, pp. 174359–174367, 2019.

[12] C.-M. Tsai, T. D. Shou, S.-C. Chen, and J.-W. Hsieh, "Use SSD to detect the digital region in electricity meter," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2019, pp. 1–7.

[13] L. Zuo, P. He, C. Zhang, and Z. Zhang, "A robust approach to reading recognition of pointer meters based on improved mask-RCNN," *Neurocomputing*, vol. 388, pp. 90–101, May 2020.

[14] G. Salomon, R. Laroca, and D. Menotti, "Deep learning for image-based automatic dial meter reading: Dataset and baselines," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[15] L. Gómez, M. Rusiñol, and D. Karatzas, "Cutting Sayre's knot: Reading scene text without segmentation. Application to utility meters," in *Proc. IAPR Int. Workshop Document Anal. Syst.*, 2018, pp. 97–102.

[16] K. Kanagarathinam and K. Sekar, "Text detection and recognition in raw image dataset of seven segment digital energy meter display," *Energy Rep.*, vol. 5, pp. 842–852, Nov. 2019.

[17] S. M. Silva and C. R. Jung, "License plate detection and recognition in unconstrained scenarios," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 593–609.

[18] F. Yang, L. Jin, S. Lai, X. Gao, and Z. Li, "Fully convolutional sequence recognition network for water meter number reading," *IEEE Access*, vol. 7, pp. 11679–11687, 2019.

[19] Y. Gao, C. Zhao, J. Wang, and H. Lu, "Automatic watermeter digit recognition on mobile devices," in *Proc. Int. Conf. Internet Multimedia Comput. Service*, 2018, pp. 87–95.

[20] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazàn, and L. P. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

[21] S. Liao, P. Zhou, L. Wang, and S. Su, "Reading digital numbers of water meter with deep learning based object detector," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis.*, 2019, pp. 38–49.

[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[23] P. He, L. Zuo, C. Zhang, and Z. Zhang, "A value recognition algorithm for pointer meter based on improved mask-RCNN," in *Proc. 9th Int. Conf. Inf. Sci. Technol. (ICIST)*, Aug. 2019, pp. 108–113.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[26] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[27] S. Montazzolli and C. Jung, "Real-time Brazilian license plate detection and recognition using deep convolutional neural networks," in *Proc. 30th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2017, pp. 55–62.

[28] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[29] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Multi-task learning for low-resolution license plate recognition," in *Proc. Iberoamerican Congr. Pattern Recognit.*, Oct. 2019, pp. 251–261.

[30] A. Calefati, I. Gallo, and S. Nawaz, "Reading meter numbers in the wild," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*. Perth, WA, Australia: IEEE, 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8945969, doi: 10.1109/DICTA47822.2019.8945969.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[32] G. R. Gonçalves, M. A. Diniz, R. Laroca, D. Menotti, and W. R. Schwartz, "Real-time automatic license plate recognition through deep multi-task networks," in *Proc. 31st SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Oct. 2018, pp. 110–117.

[33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[34] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, pp. 1–6, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[35] Copel. *Energy Company of Paraná*. Accessed: Mar. 3, 2021. [Online]. Available: http://www.copel.com/hpcopel/english/

[36] A. Bochkovskiy. *Fast-YOLOv4*. Accessed: Mar. 3, 2021. [Online]. Available: https://github.com/AlexeyAB/darknet/blob/master/cfg/yolov4-tiny.cfg

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

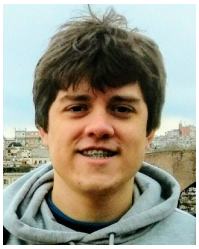
[38] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Goncalves, W. R. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–10.

[39] Z. Jiang, P. Xia, K. Huang, W. Stechele, G. Chen, Z. Bing, and A. Knoll, "Mixed frame-/event-driven fast pedestrian detection," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 8332–8338.

[40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, pp. 1–14, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[41] A. Bochkovskiy. *YOLOv3 and YOLOv2 for Windows and Linux*. Accessed: Mar. 3, 2021. [Online]. Available: https://github.com/AlexeyAB/darknet#how-to-improve-object-detection

[42] Y. Zhang and Q. Yang, "A survey on multi-task learning," 2017, pp. 1–20, *arXiv:1707.08114*. [Online]. Available: http://arxiv.org/abs/1707.08114
[43] A. Meng, W. Yang, Z. Xu, H. Huang, L. Huang, and C. Ying, "A robust and efficient method for license plate recognition," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1713–1718.
[44] H. Yoo and K. Jun, "Deep corner prediction to rectify tilted license plate images," *Multimedia Syst.*, pp. 1–8, May 2020. [Online]. Available: https://link.springer.com/article/10.1007/s00530-020-00655-8
[45] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7553–7563.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–14.
[47] OpenCV. *Geometric Image Transformations*. Accessed: Mar. 3, 2021. [Online]. Available: https://docs.opencv.org/master/da/d54/group__imgproc__transform.html
[48] A. G. Hochuli, A. S. Britto, J. P. Barddal, R. Sabourin, and L. E. S. Oliveira, "An end-to-end approach for recognition of modern and historical handwritten numeral strings," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–9.
[49] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J. Vis. Commun. Image Represent.*, vol. 71, Aug. 2020, Art. no. 102773.
[50] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector," *IET Intell. Transp. Syst.*, vol. 15, no. 4, pp. 483–503, 2021.
[51] Clova AI Research. *What is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis*. Accessed: Mar. 3, 2021. [Online]. Available: https://github.com/clovaai/deep-text-recognition-benchmark/
[52] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4168–4176.
[53] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2231–2239.
[54] W. Liu, C. Chen, K.-Y. Wong, Z. Su, and J. Han, "STAR-Net: A SpaTial attention residue network for scene text recognition," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2016, pp. 1–13.
[55] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017, pp. 334–343.
[56] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 71–79.
[57] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4715–4723.
[58] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[59] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 740–755.

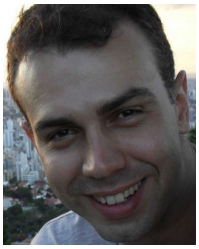


RAYSON LAROCA received the bachelor's degree in software engineering from the State University of Ponta Grossa (UEPG), Brazil, and the master's degree in computer science from the Federal University of Paraná (UFPR), Brazil. He is currently a Ph.D. Student with UFPR. His research interests include machine learning, pattern recognition, and computer vision.

ALESSANDRA B. ARAUJO received the bachelor's degree in information systems from the Faculdades Integradas de Itararé (FAFIT), Brazil, and the master's degree in computer science from the Federal University of Paraná (UFPR), Brazil. Her research interests include machine learning, pattern recognition, and computer vision.

LUIZ A. ZANLORENSI received the bachelor's degree in informatics and the master's degree in applied computing from the State University of Ponta Grossa (UEPG), Brazil, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Federal University of Paraná (UFPR), Brazil. He was a Visiting Researcher with the University of Beira Interior (UBI), Covilhã, Portugal, from 2019 to 2020. His research interests include biometrics, computer vision, machine learning, and pattern recognition.

EDUARDO C. DE ALMEIDA received the Ph.D. degree in computer science from the University of Nantes (AtlasGDD INRIA Team), France, in 2009. He has served as the Deputy Head for the Graduate Program in Computer Science, Federal University of Paraná (UFPR), Brazil, from 2010 to 2012, a Research Associate for the Interdisciplinary Centre for Security, Reliability and Trust (SnT), Luxembourg, from 2013 to 2015, and the Deputy Head for the Undergraduate Program in Computer Science, UFPR, from 2015 to 2017. He also worked as an Engineer of Database Technology with HSBC Bank (now Bradesco) and GVT Telecom (now Vivo), Brazil, from 1998 to 2004. He is currently an Associate Professor with UFPR. His research interests include database systems and data management.



DAVID MENOTTI (Senior Member, IEEE) received the B.S. and M.S. degrees in computer engineering and applied informatics from the Pontifical Catholic University of Paraná, Brazil, in 2001 and 2003, respectively, and the dual Ph.D. degree in computer science in cotutelage from the Federal University of Minas Gerais, Brazil, and the Université Paris-Est/Groupe ESIEE, France, in 2008. He was an Invited Researcher/Postdoctoral Researcher with the Computing Institute, State University of Campinas, São Paulo, Brazil, from 2013 to 2014. He is currently an Associate Professor with the Federal University of Paraná, Brazil. His research interests include computer vision focused on surveillance and biometrics.