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, David Menotti
Department of Informatics, Federal University of Paraná, Curitiba, Brazil
{rblsantos, abaraujo, lazjunior, eduardo, menotti}@inf.ufpr.br

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 six 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., ≥ 99%) when rejecting readings made with lower confidence values.

1. 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].

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 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 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, 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\(^1\). 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 (UFPR-AMR) 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 3). Our system achieves state-of-the-art results by outperforming six deep learning-based baselines in both datasets. The importance of the corner detection and counter classification stage is demonstrated, as our system made 37% fewer reading errors in 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 99.78% 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 AMR with 12,500 fully-annotated images acquired on real-world scenarios by the service company’s employees themselves. The dataset contains a well-defined evaluation protocol in order 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 six 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.

The remainder of this paper is organized as follows. We review related works in Section 2. The Copel-AMR dataset is introduced in Section 3. In Section 4, we present the proposed approach in detail. The experiments carried out and the results achieved are described in Section 5. Conclusions and future works are given in Section 6.

2. RELATED WORK

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, 19]. 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, 20].

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 [21], in this section we review works that employed deep learning-based approaches in the AMR context. We focus on studies related to digit-based meters, even though there are some recent works that addressed the recognition of dial meters [13, 14, 22]. Such works usually explore the angle between the pointer and the dial to perform the reading.

\(^1\)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.
Object detectors have been explored frequently to deal with counter detection. For example, Koščević & Subašić [10] employed Faster R-CNN [23] 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) [24] 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 [25]. 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 [26] outperforming two segmentation-free models 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. [27] employed a Fully Convolutional Network (FCN) for semantic segmentation [28] 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 employed 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 [29].

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 $1 \times 1$ and $3 \times 3$ 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, scale variations, shadows, occlusions, significant rotations, among other challenging factors).

Marques et al. [4] fine-tuned the Faster R-CNN and RetinaNet [30] 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 [27], 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.

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. [20] simply employed YOLOv3 [31] 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, 27].

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 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 3) 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 hard-
ware/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.

### 3. 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 [32]. 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. Fig. 1 shows the diversity of the dataset.

The images have a resolution of $480 \times 640$ or $640 \times 480$ 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 the Copel-AMR dataset contains 10,000 images of legible 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 Fig. 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 Fig. 1). In such cases, following the protocol adopted at 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 av-
erage 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 5,250 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 me-
ters, 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 gener-
ated are explicitly available along with the proposed dataset.

4. PROPOSED APPROACH

As illustrated in Fig. 3, the proposed approach consists of
three main stages: (i) counter detection, (ii) corner detection
and counter classification, and (iii) counter recognition.

Given an input image, the counter region is located using a
modified version of the Fast-YOLOv3 model, called Fast-
YOLOv3-SmallObj. Then, in a single forward pass of the pro-
posed 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 pro-
posed 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 5.1).
In this way, our networks become robust to images captured
under different conditions with significantly less manual ef-
fort, as the network parameters are adjusted only once.

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

4.1. 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 por-
tion of the input image, as can be seen in Fig. 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, in this work, we first locate the counter in the in-
put image and then detect its corners in the cropped patches.

As the counter region is rectified prior to the recogni-
tion stage in our system’s pipeline, the counter detector does
not need to be very sophisticated or rely on very deep mod-
els. In other words, our AMR system can tolerate less ac-
curate 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-YOLOv3 model for this task since, despite be-
ing much smaller than state-of-the-art object detectors, Fast-
YOLO variants (also known as Tiny-YOLO) are still able to
detect some objects quite precisely [33] and have been em-
ployed in various research areas in recent years [14, 34, 35].

Nevertheless, we noticed in preliminary experiments that
Fast-YOLOv3 failed in some cases where the meter was rela-
tively far from the camera (usually in images where the read-
ing is illegible). Therefore, we performed some modifications
to the network in order to improve its performance in detect-
ing small objects. More specifically, following insights from
[31] and [36], 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-YOLOv3-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.

Table 1. The Fast-YOLOv3-SmallObj model, a modified version of Fast-YOLOv3 to improve the detection of small objects. We added layers 24-30 so that the network predicts bounding boxes at 3 different scales (layers 16, 23, and 30) instead of 2 (layers 16 and 23).

<table>
<thead>
<tr>
<th>#</th>
<th>Layer</th>
<th>Filters</th>
<th>Size</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>conv 24</td>
<td>16</td>
<td>3 × 3</td>
<td>384 × 384</td>
<td>384 × 384 × 16</td>
</tr>
<tr>
<td>1</td>
<td>max 25</td>
<td></td>
<td>2 × 2</td>
<td>384 × 384</td>
<td>192 × 192 × 16</td>
</tr>
<tr>
<td>2</td>
<td>conv 26</td>
<td>32</td>
<td>3 × 3</td>
<td>192 × 192</td>
<td>192 × 192 × 32</td>
</tr>
<tr>
<td>3</td>
<td>max 27</td>
<td></td>
<td>2 × 2</td>
<td>192 × 192</td>
<td>96 × 96 × 32</td>
</tr>
<tr>
<td>4</td>
<td>conv 28</td>
<td>64</td>
<td>3 × 3</td>
<td>96 × 96</td>
<td>96 × 96 × 64</td>
</tr>
<tr>
<td>5</td>
<td>max 29</td>
<td></td>
<td>2 × 2</td>
<td>96 × 96</td>
<td>48 × 48 × 64</td>
</tr>
<tr>
<td>6</td>
<td>conv 30</td>
<td>128</td>
<td>3 × 3</td>
<td>48 × 48</td>
<td>48 × 48 × 128</td>
</tr>
<tr>
<td>7</td>
<td>max 31</td>
<td></td>
<td>2 × 2</td>
<td>48 × 48</td>
<td>24 × 24 × 128</td>
</tr>
<tr>
<td>8</td>
<td>conv 32</td>
<td>256</td>
<td>3 × 3</td>
<td>24 × 24</td>
<td>24 × 24 × 256</td>
</tr>
<tr>
<td>9</td>
<td>max 33</td>
<td></td>
<td>2 × 2</td>
<td>24 × 24</td>
<td>12 × 12 × 256</td>
</tr>
<tr>
<td>10</td>
<td>conv 34</td>
<td>512</td>
<td>3 × 3</td>
<td>12 × 12</td>
<td>12 × 12 × 512</td>
</tr>
<tr>
<td>11</td>
<td>max 35</td>
<td></td>
<td>2 × 2</td>
<td>12 × 12</td>
<td>6 × 6 × 512</td>
</tr>
<tr>
<td>12</td>
<td>conv 36</td>
<td>1024</td>
<td>3 × 3</td>
<td>12 × 12</td>
<td>12 × 12 × 1024</td>
</tr>
<tr>
<td>13</td>
<td>conv 37</td>
<td>256</td>
<td>1 × 1</td>
<td>12 × 12</td>
<td>12 × 12 × 256</td>
</tr>
<tr>
<td>14</td>
<td>conv 38</td>
<td>512</td>
<td>3 × 3</td>
<td>12 × 12</td>
<td>12 × 12 × 512</td>
</tr>
<tr>
<td>15</td>
<td>max 39</td>
<td></td>
<td>2 × 2</td>
<td>12 × 12</td>
<td>6 × 6 × 512</td>
</tr>
<tr>
<td>16</td>
<td>detection 40</td>
<td>18</td>
<td>1 × 1</td>
<td>12 × 12</td>
<td>12 × 12 × 18</td>
</tr>
<tr>
<td>17</td>
<td>route [13] 41</td>
<td></td>
<td></td>
<td></td>
<td>12 × 12 × 256</td>
</tr>
<tr>
<td>18</td>
<td>conv 42</td>
<td>128</td>
<td>1 × 1</td>
<td>12 × 12</td>
<td>12 × 12 × 128</td>
</tr>
<tr>
<td>19</td>
<td>upsample 43</td>
<td></td>
<td>2 × 2</td>
<td>12 × 12</td>
<td>24 × 24 × 128</td>
</tr>
<tr>
<td>20</td>
<td>route [19, 8] 44</td>
<td></td>
<td></td>
<td></td>
<td>24 × 24 × 384</td>
</tr>
<tr>
<td>21</td>
<td>conv 45</td>
<td>256</td>
<td>3 × 3</td>
<td>24 × 24</td>
<td>24 × 24 × 256</td>
</tr>
<tr>
<td>22</td>
<td>conv 46</td>
<td>18</td>
<td>1 × 1</td>
<td>24 × 24</td>
<td>24 × 24 × 18</td>
</tr>
<tr>
<td>23</td>
<td>detection 47</td>
<td></td>
<td></td>
<td></td>
<td>24 × 24 × 256</td>
</tr>
<tr>
<td>24</td>
<td>route [21] 48</td>
<td></td>
<td></td>
<td></td>
<td>24 × 24 × 256</td>
</tr>
<tr>
<td>25</td>
<td>conv 49</td>
<td>128</td>
<td>1 × 1</td>
<td>24 × 24</td>
<td>24 × 24 × 128</td>
</tr>
<tr>
<td>26</td>
<td>upsample 50</td>
<td></td>
<td>2 × 2</td>
<td>24 × 24</td>
<td>48 × 48 × 128</td>
</tr>
<tr>
<td>27</td>
<td>route [26, 6] 51</td>
<td></td>
<td></td>
<td></td>
<td>48 × 48 × 256</td>
</tr>
<tr>
<td>28</td>
<td>conv 52</td>
<td>128</td>
<td>3 × 3</td>
<td>48 × 48</td>
<td>48 × 48 × 128</td>
</tr>
<tr>
<td>29</td>
<td>conv 53</td>
<td>18</td>
<td>1 × 1</td>
<td>48 × 48</td>
<td>48 × 48 × 18</td>
</tr>
</tbody>
</table>

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-YOLOv3-SmallObj performed better than the Fast-YOLOv3 model with larger input sizes, while requiring comparable or even less floating-point operations (FLOP) in each forward pass. As an illustration, Fast-YOLOv3-SmallObj, which requires 6.0 billion floating-point operations (BFLOP), achieved 99.85 mAP in the validation set, whereas Fast-YOLOv3 with an input size of 512 × 512 pixels reached 99.73% while requiring 8.3 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.

4.2. 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 [37], 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 (x0/w, y0/h, ..., x3/w, y3/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.

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).

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., x0/w, y0/h, x1/w, y1/h, x2/w, y2/h, x3/w, y3/h, [legible, illegible]).

<table>
<thead>
<tr>
<th>#</th>
<th>Layer</th>
<th>Filters</th>
<th>Size</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>conv 54</td>
<td>16</td>
<td>3 × 3</td>
<td>192 × 64</td>
<td>192 × 64 × 16</td>
</tr>
<tr>
<td>1</td>
<td>max 55</td>
<td></td>
<td>2 × 2</td>
<td>192 × 64</td>
<td>96 × 32 × 16</td>
</tr>
<tr>
<td>2</td>
<td>conv 56</td>
<td>32</td>
<td>3 × 3</td>
<td>96 × 32</td>
<td>96 × 32 × 32</td>
</tr>
<tr>
<td>3</td>
<td>max 57</td>
<td></td>
<td>2 × 2</td>
<td>96 × 32</td>
<td>48 × 16 × 32</td>
</tr>
<tr>
<td>4</td>
<td>conv 58</td>
<td>64</td>
<td>3 × 3</td>
<td>48 × 16</td>
<td>48 × 16 × 64</td>
</tr>
<tr>
<td>5</td>
<td>max 59</td>
<td></td>
<td>2 × 2</td>
<td>48 × 16</td>
<td>24 × 8 × 64</td>
</tr>
<tr>
<td>6</td>
<td>flatten 60</td>
<td></td>
<td></td>
<td>24 × 8</td>
<td>12288</td>
</tr>
</tbody>
</table>

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 ≈ 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 [38, 39], 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.

After locating the four corners, we rectify the counters classified as legible/operational by applying a perspective transformation to the cropped patches. Counters classified 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 Fig. 4, rectified counters become more horizontal, tightly-bounded, and easier to read), as well as the first system able to deal with images in which it is not possible to perform the meter reading due to occlusions or faulty meters.

![detected counter regions](image)

![rectified counter regions](image)

Fig. 4. 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.

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 Fig. 2). Fig. 5 shows some examples of the images generated by us.

![Some representative examples of the images generated by us for training CDCC-NET. The images in the first column are the originals and the others were generated automatically.](image)

4.3. Counter Recognition

Once the counter region has been located and rectified, the digits must be recognized. To this end, we built a new detection network, called Fast-OCR, that incorporates features from existing models focused on the speed/accuracy trade-off, such as YOLOv2 [25], CR-NET [26] and Fast-YOLOv3. 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 [40–42]. 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.

The architecture of Fast-OCR is shown in Table 3. The input size is \(384 \times 128\) 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-YOLOv3, 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 Fig. 6, and also on how accurately the counter region was detected in the previous stage. As in the networks introduced in [25, 26], the convolutional layers in Fast-OCR mostly have \(3 \times 3\) kernels and the number of filters is doubled after each max-pooling layer. In addition, there are \(1 \times 1\) convolutional layers between \(3 \times 3\) convolutions to reduce the feature space from preceding layers.

<table>
<thead>
<tr>
<th>#</th>
<th>Layer</th>
<th>Filters</th>
<th>Size</th>
<th>Input</th>
<th>Output</th>
<th>BFLOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>conv</td>
<td>32</td>
<td>3 x 3/1</td>
<td>384 x 128 x 3</td>
<td>384 x 128 x 32</td>
<td>0.085</td>
</tr>
<tr>
<td>1</td>
<td>max</td>
<td>2 x 2/2</td>
<td>384 x 128 x 32</td>
<td>192 x 64 x 32</td>
<td>192 x 64 x 64</td>
<td>0.002</td>
</tr>
<tr>
<td>2</td>
<td>conv</td>
<td>64</td>
<td>3 x 3/1</td>
<td>192 x 64 x 32</td>
<td>192 x 64 x 64</td>
<td>0.453</td>
</tr>
<tr>
<td>3</td>
<td>max</td>
<td>2 x 2/2</td>
<td>192 x 64 x 64</td>
<td>96 x 32 x 64</td>
<td>96 x 32 x 128</td>
<td>0.001</td>
</tr>
<tr>
<td>4</td>
<td>conv</td>
<td>128</td>
<td>3 x 3/1</td>
<td>96 x 32 x 64</td>
<td>96 x 32 x 128</td>
<td>0.453</td>
</tr>
<tr>
<td>5</td>
<td>max</td>
<td>2 x 2/2</td>
<td>96 x 32 x 128</td>
<td>48 x 16 x 128</td>
<td>48 x 16 x 256</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>conv</td>
<td>256</td>
<td>3 x 3/1</td>
<td>48 x 16 x 128</td>
<td>48 x 16 x 256</td>
<td>0.453</td>
</tr>
<tr>
<td>7</td>
<td>conv</td>
<td>128</td>
<td>1 x 1/1</td>
<td>48 x 16 x 256</td>
<td>48 x 16 x 256</td>
<td>0.050</td>
</tr>
<tr>
<td>8</td>
<td>conv</td>
<td>256</td>
<td>3 x 3/1</td>
<td>48 x 16 x 256</td>
<td>48 x 16 x 256</td>
<td>0.453</td>
</tr>
<tr>
<td>9</td>
<td>max</td>
<td>2 x 2/2</td>
<td>48 x 16 x 256</td>
<td>24 x 8 x 256</td>
<td>24 x 8 x 256</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>conv</td>
<td>512</td>
<td>3 x 3/1</td>
<td>24 x 8 x 256</td>
<td>24 x 8 x 256</td>
<td>0.453</td>
</tr>
<tr>
<td>11</td>
<td>conv</td>
<td>512</td>
<td>1 x 1/1</td>
<td>24 x 8 x 512</td>
<td>24 x 8 x 512</td>
<td>0.050</td>
</tr>
<tr>
<td>12</td>
<td>conv</td>
<td>512</td>
<td>3 x 3/1</td>
<td>24 x 8 x 512</td>
<td>24 x 8 x 512</td>
<td>0.453</td>
</tr>
<tr>
<td>13</td>
<td>conv</td>
<td>45</td>
<td>1 x 1/1</td>
<td>24 x 8 x 512</td>
<td>24 x 8 x 45</td>
<td>0.009</td>
</tr>
<tr>
<td>14</td>
<td>detection</td>
<td></td>
<td></td>
<td>24 x 8 x 256</td>
<td>24 x 8 x 256</td>
<td>0.025</td>
</tr>
<tr>
<td>15</td>
<td>route [11]</td>
<td></td>
<td></td>
<td>24 x 8 x 256</td>
<td>48 x 16 x 256</td>
<td>0.025</td>
</tr>
<tr>
<td>16</td>
<td>conv</td>
<td>256</td>
<td>1 x 1/1</td>
<td>24 x 8 x 256</td>
<td>48 x 16 x 256</td>
<td>0.005</td>
</tr>
<tr>
<td>17</td>
<td>upsample</td>
<td></td>
<td></td>
<td>2 x 24 x 8 x 256</td>
<td>48 x 16 x 256</td>
<td>0.025</td>
</tr>
<tr>
<td>18</td>
<td>route [17, 6]</td>
<td></td>
<td></td>
<td>48 x 16 x 512</td>
<td>48 x 16 x 512</td>
<td>0.025</td>
</tr>
<tr>
<td>19</td>
<td>conv</td>
<td>512</td>
<td>3 x 3/1</td>
<td>48 x 16 x 512</td>
<td>48 x 16 x 512</td>
<td>3.624</td>
</tr>
<tr>
<td>20</td>
<td>conv</td>
<td>45</td>
<td>1 x 1/1</td>
<td>48 x 16 x 512</td>
<td>48 x 16 x 45</td>
<td>0.035</td>
</tr>
</tbody>
</table>

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 YOLOv2 [25] and YOLOv3 [31] require 29.4 and 65.4 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, 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, heuristic rules can also be explored to identify illegible/faulty meters erroneously classified as legible/operational in the previous stage.

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 (see Fig. 5) 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.

5. EXPERIMENTS

5.1. 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, 20, 27] 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 six baseline methods [3, 15, 20, 27] trained by us on exactly the same images as the proposed method. These specific methods, described in Section 2, were chosen/implemented by us for two main reasons: (i) they were recently employed in the context of AMR 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, 40, 43]. 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-YOLOv3-SmallObj model and then apply their network to the detected region. The same was done with the recognition network proposed by Calefati et al. [27], as they dealt with the counter detection stage using an FCN for semantic segmentation [28] 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-YOLOv3-SmallObj) is able to achieve high F-measure rates in both datasets (see Section 5.2.1) 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, while the other models were trained using Keras. 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 mentioned above were defined based on experiments performed in the validation set.

5.2. Results and Discussion

In this section, we report the experiments carried out to verify the effectiveness of the proposed AMR system. We first 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 and compare the results achieved with those obtained by six deep learning-based baselines.

5.2.1. 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. [44] 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-YOLOv3-SmallObj model over different IoU thresholds, from 0.5 to 0.95, similarly to the COCO [45] primary metric (mAP@IoU=[0.5:0.95]). As we consider only one meter per image, the precision and recall rates are identical.

For comparison purposes, we also list in Table 4 the detection results obtained by the original Fast-YOLOv3 model. Observe that the results achieved by Fast-YOLOv3-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-YOLOv3-SmallObj model correctly detected all counter regions in the UFPR-AMR dataset and failed in only 22 of the 5,000 test images of the Copel-AMR. However, 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).

Some detection results are shown in Fig. 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.

5.2.2. Corner Detection and Counter Classification

To assess the performance of CDCC-NET in the corner detection task, following [39], 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 aver-

Table 4. F-measure values obtained over different IoU thresholds, from 0.5 to 0.95, in the counter detection stage. Note that Fast-YOLOv3-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                  | 0.5  | 0.6  | 0.7  | 0.8  | 0.9  | 0.95 | 0.95
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UFPR-AMR (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast-YOLOv3</td>
<td>99.9%</td>
<td>99.5%</td>
<td>97.9%</td>
<td>89.9%</td>
<td>47.3%</td>
<td>7.6%</td>
<td>73.6%</td>
</tr>
<tr>
<td>Fast-YOLOv3-SmallObj</td>
<td>100.0%</td>
<td>100.0%</td>
<td>98.9%</td>
<td>91.4%</td>
<td>57.6%</td>
<td>15.8%</td>
<td>77.6%</td>
</tr>
<tr>
<td>Copel-AMR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast-YOLOv3</td>
<td>99.3%</td>
<td>98.8%</td>
<td>97.3%</td>
<td>90.3%</td>
<td>44.5%</td>
<td>9.4%</td>
<td>73.3%</td>
</tr>
<tr>
<td>Fast-YOLOv3-SmallObj</td>
<td>99.6%</td>
<td>99.0%</td>
<td>98.0%</td>
<td>92.4%</td>
<td>55.7%</td>
<td>14.0%</td>
<td>76.4%</td>
</tr>
</tbody>
</table>

Fig. 7. Representative samples of counters detected by the Fast-YOLOv3-SmallObj model – images taken under unconstrained conditions, e.g., with significant reflections, rotations, and scale variations.
improved less than we expected (from previous years were detected perfectly on each image, and the 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 on license plate images: Smaller-LocateNet [38], LocateNet [38] and Hybrid-MobileNetV2 [39]. 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).

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.

<table>
<thead>
<tr>
<th>Model</th>
<th>FPS</th>
<th>Mean pixel distance between the predicted corners and the ground truth (normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>UFPR-AMR</td>
</tr>
<tr>
<td>Smaller-LocateNet [38]</td>
<td>209</td>
<td>0.0059</td>
</tr>
<tr>
<td>LocateNet [38]</td>
<td>165</td>
<td>0.0031</td>
</tr>
<tr>
<td>CDCC-NET (Ours)</td>
<td>191</td>
<td>0.0016</td>
</tr>
<tr>
<td>Hybrid-MobileNetV2 [39]</td>
<td>98</td>
<td>0.0014</td>
</tr>
</tbody>
</table>

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 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.

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 96.23% 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 5.2.3 for more information and qualitative results).

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 since it correctly filtered out 98.9% of the cases 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 since the counter rectification and recognition tasks are skipped in such cases, while correctly accepting 99.78% of the legible/operational meters.

According to Fig. 8, 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.

5.2.3. 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. A correctly recognized counter means that all digits on the counter were correctly recognized, as a single digit recognized incorrectly may imply 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.

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 95.50% of the meters in the
UFPR-AMR’s test set and 96.95% 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 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 Fig. 9), 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. As can be seen, the corner detection and counter classification stage is essential for accomplishing outstanding results in unconstrained scenarios, as our system made 37% fewer reading errors (i.e., \(\frac{96.95\% - 95.15\%}{100\%} = 9.80\%\)) in the Copel-AMR dataset when feeding rectified counters into the recognition network.

The end-to-end results achieved by the baselines ranged from 85.46% 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 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 the proposed system, which is able to process 79 FPS. In this sense, for simpler/constrained scenarios, we believe that it can be employed in low-end setups or even in some mobile phones (taking a few seconds).

In general, as observed in [3], the methods using recognition models based on object detectors (e.g., CR-NET and Fast-OCR) performed better than those where counter recognition is done holistically [15, 27]. The exception to this rule was the approach proposed by Liao et al. [20], which relies on YOLOv3 [31] to perform the meter reading directly in the input image, that is, without first detecting the counter region. We believe that this approach would have made even more

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

<table>
<thead>
<tr>
<th>Approach</th>
<th>FPS</th>
<th>Recognition Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>UFPR-AMR</td>
</tr>
<tr>
<td>Fast-YOLOv3-SmallObj + Gómez et al. [15]</td>
<td>66</td>
<td>87.38%</td>
</tr>
<tr>
<td>Laroca et al. [3] (Multi-task)</td>
<td>58</td>
<td>89.12%</td>
</tr>
<tr>
<td>Liao et al. [20]</td>
<td>61</td>
<td>91.37%</td>
</tr>
<tr>
<td>Liao et al. [20] (larger input size)</td>
<td>42</td>
<td>92.25%</td>
</tr>
<tr>
<td>Fast-YOLOv3-SmallObj + Calefati et al. [27]</td>
<td>59</td>
<td>93.87%</td>
</tr>
<tr>
<td>Laroca et al. [3] (CR-NET)</td>
<td>41</td>
<td>94.25%</td>
</tr>
<tr>
<td>Ours - unrectified (Fast-YOLOv3-SmallObj + Fast-OCR)</td>
<td>79</td>
<td>94.75%</td>
</tr>
<tr>
<td>Ours (Fast-YOLOv3-SmallObj + CDCC-NET + Fast-OCR)</td>
<td>55</td>
<td>95.50%</td>
</tr>
</tbody>
</table>

**Fig. 9.** 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.
reading errors in the Copel-AMR dataset if the regions containing consumer identification were not blurred (we blurred them due to privacy constraints; this was a request from Copel) since they contain relatively large digits.

Fig. 10 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 Fig. 11. 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, 46], 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 [32] and other service companies, we present 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.

**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%) when using a confidence threshold that rejects 15% of the images.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Rejection Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0%</td>
</tr>
<tr>
<td>UFPR-AMR</td>
<td>95.50%</td>
</tr>
<tr>
<td>Copel-AMR</td>
<td>96.95%</td>
</tr>
<tr>
<td>Average</td>
<td>96.23%</td>
</tr>
</tbody>
</table>
6. CONCLUSIONS 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 37\% fewer reading errors in 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 95.50\% and 96.95\% 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 for AMR as well as 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.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research. We also 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


