FABIANO AIZAWA VALENÇA DE MELLO

KEYPOINT DETECTION ON MULTI-DIAL METERS FOR AUTOMATIC METER READING USING YOLOV7

(*pre-defense version, compiled at March 6, 2023*)

Documento criado para entrega do Trabalho de Conclusão de Curso 2 em Inteligência Artificial.

Área de concentração: *Ciência da Computação*.

Orientador: David Menotti.

CURITIBA PR

2023

RESUMO

A leitura Automática de Medidores (AMR em inglês) é a solução menos custosa e mais rápida para a tarefa arduosa da leitura de medidores residênciais e industriais antigos, quando comparada com a substituição desses medidores pelos novos *smartmeters*. Apesar de vários métodos e pesquisa feitas na área, encontramos poucos trabalhos anteriores que abordaram o problema de AMR para medidores residenciais com múltiplos ponteiros. Neste trabalho, focamos neste tipo de medidores, destacando as diferenças entre medidores com um único ponteiro e propomos uma abordagem que, até onde sabemos, nunca foi implementada para medidores com múltiplos ponteiros. Nossas principais contribuições são: (a) mostramos que detecção de *keypoints* em medidores com múltiploos ponteiros usando um detector de objetos (YOLOv7) é uma abordagem viável, e (b) expandimos o, que para o melhor do nosso conhecimento, é o único *dataset* público de medidores com múltiplos ponteiros, a UFPR-ADMR, com anotações para possibilitar detecção de *keypoints*. Nosso método atingiu uma taxa de detecção geral de 99.15% para todos os dígitos e *keypoints*.

ABSTRACT

Image-based Automatic Meter Reading (AMR) is the less costly and faster solution to the laborious task of reading old residential and industrial meters, when compared to replacing outdated models with smart meters. Even with many methods and research done in this area, we found few previous works that tackled the AMR problem for residential multi-dial meters. In this work we focused on these type of meters, highlighting the differences with single-pointer meters (gauges) and propose an approach that to the best of our knowledge was never implemented for multi-dial meters. Our main contributions are: (a) we show that detecting keypoints on multi-dial meters using an object detector (YOLOv7) is a viable approach, and (b) we expand upon the, as far as we know, only public real-world multi-dial meter dataset, the UFPR-ADMR dataset, with annotations for enabling keypoint detection. Our method achieved an overall 99.15% detection rate for all digits and keypoints.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AMR	Automatic Meter Reading
RoI	Region of Interest
CNN	Convolutional Neural Network

CONTENTS

1 INTRODUCTION

Measuring and billing of residential utilities (energy, water, and gas), especially in developing countries, is a laborious task, because old mechanical meters are still prevalent, requiring an employee of the utility company to visit each residence to perform the readings (Bishwokarma et al., 2020; Salomon et al., 2020; Laroca et al., 2019). Automatic meter reading (AMR) is the task of automatically measuring the values of a meter. While smart meters can perform this task, replacing the old meters would have a great monetary and environmental cost (Howells et al., 2021; Salomon et al., 2020). Because of that, many image-based methods for AMR were proposed, as their implementation would not require the replacement of the meters. Figure 1.1(a) shows examples of residential digit meters from the UFPR-AMR dataset (Laroca et al., 2019) and Figure 1.1(b) residential pointer meters from the UFPR-ADMR dataset (Salomon et al., 2020).

(a) residential digit meters

(b) residential pointer meters

Figure 1.1: Examples of residential meters.

Traditional methods, often those also used in similar tasks (*e.g.*, text detection, circle fitting), have been applied in AMR methods, however they have issues in unconstrained scenarios (Calefati et al., 2019; Salomon et al., 2020), because images captured in real field scenarios may present challenging environmental factors (such as the ones in Fig. **??**) and other variations (*e.g.*, shooting angle, different models, etc.). In contrast, Deep learning-based methods have great generalization capabilities, and Convolutional Neural Networks (CNNs) in particular are well suited for detection and classification problems, being widely used in recently developed AMR methods.

(a) uneven lighting (b) blur (c) distant capture (d) reflections

(e) dirt (f) glare (g) broken glass

Figure 1.2: Images from Salomon et al. (2020) showing samples of challenging scenarios present in the UFPR-ADMR dataset.

In the AMR literature there is a notable lack of publicly available datasets (Salomon et al., 2020), with most works using private datasets, either self-gathered or provided by companies. Additionally, most AMR methods dealing with residential meters focus on digit meters, and for pointer meters, almost all are designed for industrial single-dial meters (also known as gauges). Deep-learning based methods for residential pointer meters with multiple dials (multi-dial meters) were only found in Salomon et al. (2020) and its related works, which also presents the only publicly available dataset we found for this type of meter (UFPR-ADMR). It was based on this dataset that we built the dataset used in our experiments.

While digit meters present topological similarities regardless of model (most notably having a counter area, where the digits to be read are located 1.1(a)), there are significant differences between gauges and multi-dial meters that might pose challenges when attempting to apply methods successful for gauges on multi-dial meters. Other than the evident difference in number of dials, in gauges the pointer is used to read a value in a scale, and in multi-dial meters, each dial represents one digit of the reading, which means misreading a single dial might cause a significant error in the measurement, especially in the most significant dials (Salomon et al., 2020).

To the best of our knowledge, popular deep learning-based methods for angle calculation in AMR of pointer meters, such as keypoint[1] detection and feature extraction, were never implemented for multi-dial meters. Object detection models, such as the YOLO family, are most often used in this field as a first step to detect the RoI. They were used before for keypoint

[1]In the context of this work, a keypoint refers to specific points that can be detected in a pointer meter in order to calculate the angle of the pointer to find the reading value (i.e., start of the scale, the marks on the scale, dial position, center of the meter, edge positions, etc.)

detection (Ueda et al., 2020; Zhou et al., 2022), however the point of the dial was never directly detected with these models, they first detect the bounding box for the entire arm of the pointer, then calculate the point of the dial.

We will focus solely on keypoint detection on residential multi-dial meters, as it was an approach not explored in the previous works we found on the subject (Salomon et al., 2020; Salomon et al., 2022). We will also expand upon the UFPR-ADMR dataset so that keypoint detection is possible in a subset of that dataset. As the goal of this work was only to study the detection of the keypoints, following steps to calculate the final reading, such as perspective correction and angle calculation, were not performed, leaving them to a future work.

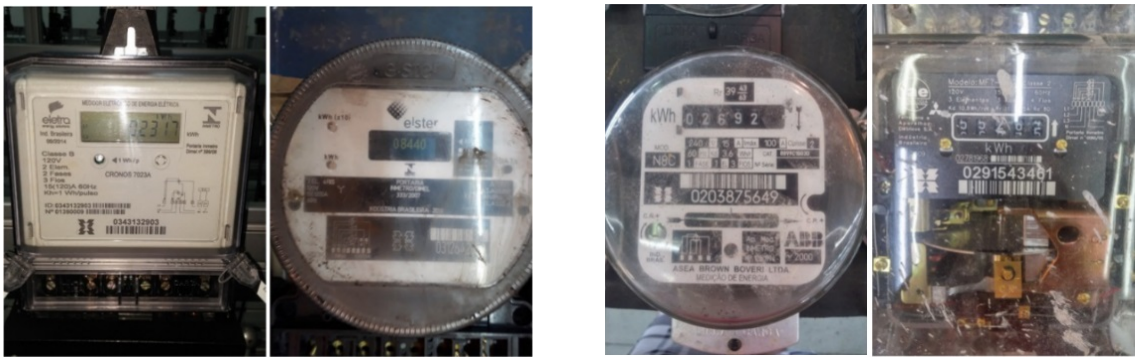
In this work we will show that it is viable to apply YOLOv7, a state-of-the-art object detection model, to directly detect all the keypoints necessary to calculate the angle of the dial of a multi-dial meter while highlighting the differences between the approaches and the types of meter. The two main contributions of this work are: (a) we show that using an object detection model for keypoint detection on multi-dial meters in a single stage is a viable strategy, and (b) we expand upon the UFPR-ADMR dataset by adding extra annotations to 500 images (more than 2200 individual digits), labeling them with the coordinates for the point of the dial, center of the dial, and start of the scale for each digit.

The remainder of this work will be organized as follows. In Section 2 we review related works and their methods. Section 3 describes our proposed approach. The results and the dataset used are shown and analyzed in Section 4. Finally, the conclusions are stated in Section 5

2 RELATED WORK

Automatic meter reading has two main categories of meters being studied in the literature: digit meters (Fig. 2.1) and pointer meters (Fig. 2.2).

Digit meters include meters with digital electronic displays (Fig. 2.1(a)) and meters with analog rolling digit displays (Fig. 2.1(b)). AMR methods for digital and analog displays largely include the same general steps, and in practice, many methods that work for analog meters will also work for digital meters. This is also reflected in the datasets created for these meters, as they often include both types (Azeem et al., 2020; Shuo et al., 2019; Bishwokarma et al., 2020; Waqar et al., 2019; Calefati et al., 2019; Dzeha et al., 2021). Most approaches include some form of these steps: detection of the counter region (Region of Interest - RoI), digit segmentation, and digit recognition. Various methods can be applied to achieve these steps, from traditional image processing algorithms to deep learning techniques.

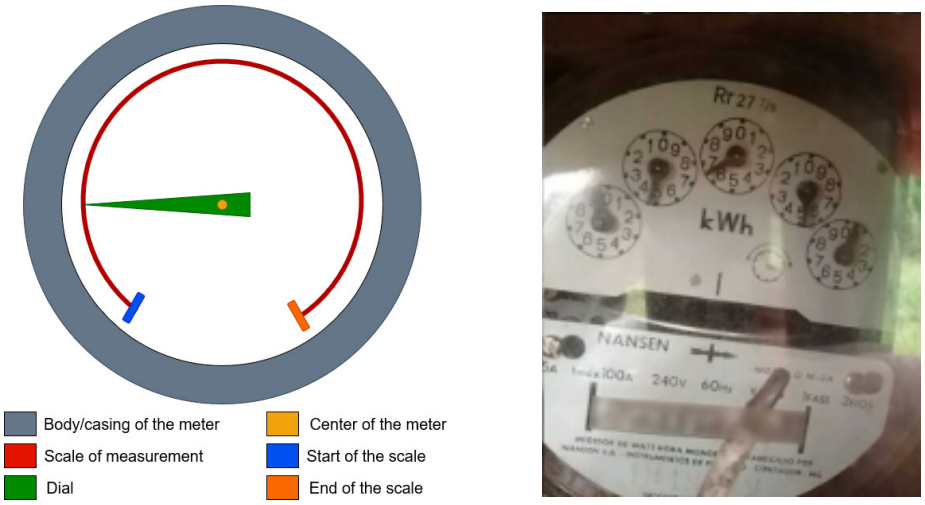


(a) meters with electronic display

(b) meters with analog rolling digit display

Figure 2.1: Samples of digit meters from the UFPR-AMR dataset

Pointer meters use rotating pointers in a circular scale to measure their readings (Fig. 2.2). A widely used method being currently applied for AMR of pointer meters is finding the angle of the pointer. This is achieved by mainly 2 methods: keypoint detection and pointer extraction. Keypoint detection methods detect the keypoints of the meter, to then calculate the reading using these keypoints (usually calculating the angle between the dial and start of the scale) (Fang et al., 2019; Ueda et al., 2020; Zhou et al., 2022; Howells et al., 2021). Pointer extraction methods extract the region of the image which contains the dial (through traditional image processing techniques, or deep learning techniques) to find the angle of the dial (Zhou et al., 2021; He et al., 2019; Zuo et al., 2020; Peng et al., 2021). However all the AMR methods for pointer meters found in the literature, with the exception of Salomon et al. (2020) and its related works, were designed for gauges (Fig. 2.2(a)). Although in Salomon et al. (2020) an approach for multi-dial meters (Fig. 2.2(b)) is presented, it differs from other works, as the angle of the dial is not calculated, the reading value is generated based on the dial position detected in a previous step.



(a) diagram of a common gauge

(b) multi-dial meter from UFPR-ADMR

Figure 2.2: The two types of pointer meters, gauges and multi-dial meters.

2.1 DIGIT METERS

In this section we will present AMR methods designed for digit meters that use deep learning techniques. It's in this category of meters that most works dealing with residential (water, electricity/energy, gas) meters are present.

Son et al. (2019) presented an AMR method for analog gas meters. Their method consists of a 3-stage process in which deep learning techniques are used in the first and last stage. The first stage, region detection, is achieved using YOLOv3, the second stage, digit segmentation, uses traditional image processing methods, and the last stage, digit recognition, uses a CNN with a modified VGG network structure. The dataset used was a private, self-gathered dataset, composed of 5000 annotated images. Their method achieved a recognition rate of 85.71% accuracy for the meter value readings, 60.90% accuracy for the ID readings, and 57.14% accuracy for value + ID readings.

Azeem et al. (2020) presented an AMR method for analog and digital electricity meters. Their method consists of 3 stages, using deep learning in all of them. Counter detection, digit segmentation, and digit recognition are all achieved using a Mask-RCNN approach. The dataset used was the public UFPR-AMR dataset. Against 3 other methods compared, their method achieved better F-Measure for the counter detection, and higher accuracy for digit recognition.

Shuo et al. (2019) presented an AMR method for digital electricity meters. Their method consists mainly of 3 parts: Meter type identification and digital area location, image enhancement, and digital segmentation and recognition. Meter type identification and digital area location are achieved using MobileNetv2-SSD, image enhancement uses traditional image processing techniques, and digital segmentation and recognition are achieved using a SVM classifier. The dataset used was a privately selected dataset, composed of 2300 images. Their method achieved an overall accuracy of 88.67% on LCD and analog meters.

Košcevic and Subašic (2018) presented an AMR method for analog and digital residential meters. The method has 2 main parts, the first part consists of the detection of the counter and serial number regions, and the second part performs the digit reading. The first part has 2 steps of image processing then the counter and serial number detection is performed using a modified Faster R-CNN. The second part also uses a modified Faster R-CNN for digit reading. The dataset used was a private self-gathered dataset, composed of 775 images. Their method successfully read 97.01% of counter regions and 63.51% of serial number regions, with an average time of 0.36 seconds per image.

Bishwokarma et al. (2020) proposed an AMR system for analog and digital electricity meters using smartphones. The system uses a mobile device to send images to a server for processing. On the server, the task of counter detection and digit recognition are performed by a deep neural network (YOLOv3). The dataset used was a small private self-gathered dataset, composed of 410 images. For each class of digit recognized, the accuracy of the network varied between 50% and 97.5%.

Peng and Chen (2020) presented an AMR method for analog water meters. The method uses an end-to-end approach using deep learning. The network used is a R-FCN model. The dataset used was a private self-gathered dataset, composed of 160785 images (of which 15000 are training images). The method achieved a recognition accuracy of 81.70% and the recognition rate is 52.93%.

Waqar et al. (2019) presented an AMR method for analog and digital electricity meters. The method uses deep learning for extraction and recognition of digits. The model used is the Faster R-CNN. The dataset used was private, obtained from electricity companies, and was composed of 10310 images. The model achieved 76.05% accuracy in the test images.

Calefati et al. (2019) proposed an AMR method for analog and digital meters. Their approach has 2 phases, a detection phase, and a recognition phase. Each phase uses a different CNN to achieve its goal. The dataset used was obtained from private utility companies (the authors will publish a version which does not contain customer information). The end-to-end accuracy of the method was 85.60%.

Li et al. (2019) proposed an AMR system for analog water meters using cameras and cloud technology. The system uses an installed camera in the meter to capture images and send them to a cloud server for processing. The server uses a CNN for digit recognition. The dataset used was a private self-gathered dataset consisting of 6000 images. The proposed network successfully reduced computational load, storage and running time.

Dzeha et al. (2021) presented an AMR system for analog and digital electricity meters using mobile phones. The smartphone captures an image and sends it to a server for processing. The feature extraction and character recognition are performed on the server using a CNN. The dataset used was the MNIST dataset. The accuracy claimed was 99.09%, and time to extract information was approximately 1.52 seconds.

Zhu et al. (2022) proposed an AMR method for analog water meters. Their approach focuses on a new method for data augmentation and a new module to connect with the CNN used in digit recognition. The network used in recognition is the YOLOv4-tiny model. The dataset used was their own collected dataset available publicly at Github (at the time of writing the page was not available) consisting of 1277 images. The data augmentation method showed improvements when compared to the baseline, and the new module presented improvements on the baseline network and also when compared against other modules.

Cerman et al. (2016) proposed an AMR method for residential meters. Their method has 2 stages, a digit detection, and a digit recognition phase. Digit detection uses traditional image processing methods, and for their second stage, they compare 2 approaches, one using Tesseract OCR, and the other using a CNN. The dataset used was a company's internal collection, consisting of a total of almost 50 thousand frames of electricity, gas and water meters. The digit detection achieved an average accuracy of 81.42% for the 3 types of meter, and the CNN achieved an average accuracy of 97.34% (compared to the Tesseract OCR with 85.76%).

Chouiten and Schaeffer (2014) presented an AMR method for analog gas meters using mobile phones. Their method uses machine learning for the detection of the region of interest. The last stage, optical character reading, is achieved using GOCR (an open source Optical Character Recognition model). The dataset used was private, provided by a utility company. The method achieved an average of 89% accuracy across various models of mobile devices.

2.2 POINTER METERS

In this section we will present AMR methods for pointer meters that use deep learning techniques. In this section, with the exception of Salomon et al. (2020), all works deal with industrial gauges.

Fang et al. (2019) presented an AMR method for pointer meters. The method has 3 steps: keypoint detection, scale circle fitting, and finally value calculation. The first step, keypoint detection is achieved using an improved Mask R-CNN (ResNet-18), then the following steps use the detected keypoints for the calculations. The dataset used for training and testing was a private self-gathered dataset, consisting of 1160 images. Their method was compared to traditional methods (Hough transform), showing better accuracy in their data, however the run time was slower than the traditional method.

Zhou et al. (2021) presented an AMR method for industrial pointer meters at gas gathering stations. The method first detects the location of the meter and extracts the pointer region using deep learning techniques then various steps of image processing are done and finally the value is calculated. The first step, meter location detection is performed using a YOLO network. The dataset used was a private self-gathered dataset, composed of 1500 images. There was no data in regards to accuracy or time achieved by their method.

Ueda et al. (2020) proposed an AMR system for pointer meters using smartphones. The system uses a mobile device to capture and send frames to a server for processing. On the

server, various steps are performed for calculating the reading value, and the steps that use deep learning techniques are the meter detection and pointer region extraction steps, with each step using a different Single Shot Multibox Detector (SSD) network. The dataset used was a private self-gathered dataset, and consists of 12600 training images. Their method that achieved best accuracy was using multiple frames, with homography transform, achieving an error of less than 0.015 with a frequency of 0.9352.

He et al. (2019) proposed an AMR method for pointer meters. The method first uses deep learning to segment the meter dial and extract the pointer area, then calculates the reading. The framework used for the segmentation is the Mask-RCNN, and the value is calculated using the angle method. The dataset used was a private self-gathered dataset composed of 1430 RGB images. Their method had time costs of 2.8724 seconds on CPU and 0.0931 seconds on GPU, and was shown to achieve better performance in instance segmentation and average reference error when compared to other methods.

Zuo et al. (2020)[1] presented an AMR method for pointer meters. The approach consists of 4 stages: feature extraction, classifier, perspective transformation, and reading calculation. The first stage uses a Mask-RCNN (with the RoiAlign replaced by the PrRolPooling method). The dataset used was a private dataset composed of 1430 RGB images. The improved Mask-RCNN has achieved better performance than the original and the average running time per picture is 0.0931 seconds by the GPU while 2.8724 seconds in the CPU.

Peng et al. (2021) presented an AMR method for pointer meters. The method first detects the dial position and meter class, then extracts the pointer to calculate the readings. The dial position is detected with a YOLOv4 network, and the feature extraction is achieved with a modified U-Net network. The dataset used was a selection of 1806 images from substations. The method proposed presents a lower error detection rate and missed detection rate than other ML methods on meter detection, and also presents improvements on reading recognition when compared to the control groups.

Zhou et al. (2022) proposed an AMR method for pointer meters. The method has 2 main stages: a detection stage, and a recognition stage. The detection stage uses a CNN (based on the YOLOv5 framework) to find the meter positioning and achieve feature detection, then the recognition stage uses the features detected to calculate the readings. The dataset used was a self-gathered dataset consisting of 6725 images. The YOLOv5 framework used for meter position performed better than the other frameworks it was compared to, with a significant increase in frames per second detected.

Howells et al. (2021) presented an AMR method for pointer meters using mobile phones. The method has 3 stages: (1) gauge detection, (2) perspective correction, and (3) gauge reading. CNNs are used in stage 1, for location detection and homography inferention, and in stage 3, for detection of the keypoints used in the angle calculation. The dataset used is a combination of

[1]Zuo et al. (2020) and He et al. (2019) have the same authors, albeit in a different order. Both papers appear to present the same work.

self-gathered real data and synthetic 3D data. Their method achieved 25 fps on an iPhone 11 and performed better when compared against the baselines and the comparison, achieving a mean of 98.6% of accuracy on gauge detection and correctly detecting all keypoints.

Salomon et al. (2020) presented an AMR baseline for residential pointer meters with multiple dials. Their method has 4 main steps: image acquisition, dial detection, dial recognition, and final reading. The detection, recognition and reading stages are based on deep learning, and the detection models evaluated for their approach were models based on YOLO (Fast-YOLOv3, YOLOv3) and Faster R-CNN (ResNet-50, ResNet-101, ResNeXt-101). A new dataset was created and made public, the UFPR-ADMR dataset (shared upon request). The best results were achieved by the ResNeXt-101 with a recognition rate of 93.60% per dial and 75.25% per meter.

Salomon et al. (2022) expanded upon their approach in their following work. They introduce a novel regression approach for object detection, then a fine tuned ResNet and Xception models are explored for feature extraction, then finally a custom FCN is used to predict the final readings. They also expanded their dataset. Their best results achieved a 92.5% recognition rate per meter, and 98.11% per dial.

2.2.1 Feature Extraction and Keypoint Detection for the Angle Method

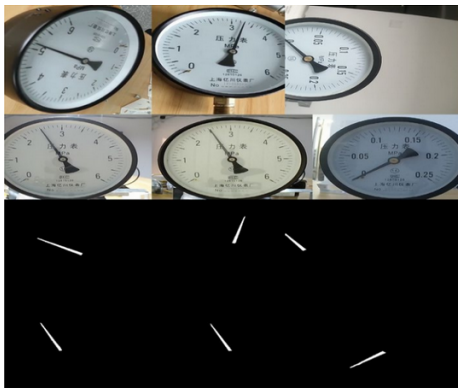
The methods that use the angle method shown in this section can be further divided into feature extraction and keypoint detection methods for calculating the angle between the pointer and the starting point of the scale.

Feature extraction methods (Zhou et al., 2021; He et al., 2019; Zuo et al., 2020; Peng et al., 2021) use a specialized network or traditional image processing methods to extract the region of the image that represents the dial, then use techniques such as straight line fitting to find the appropriate angle. In particular, the Machine Learning techniques used here are algorithms based on classification, such as Regional Convolutional Neural Networks (most often Mask-RCNN models).

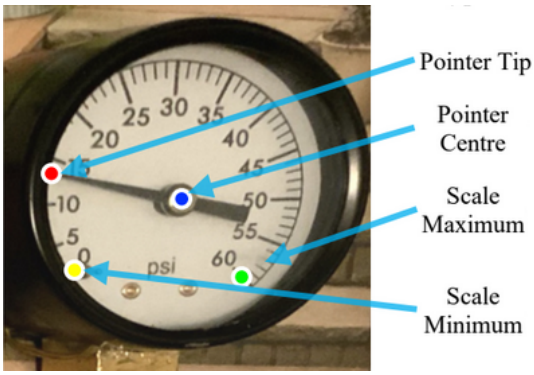
Keypoint detection methods (Fang et al., 2019; Ueda et al., 2020; Zhou et al., 2022; Howells et al., 2021) use deep learning to detect the points in the meter relevant to calculate the angle, such as: center, edge of the dial, minimum and maximum of the scale. The ML models found here are often algorithms based on regression, such as the YOLO family and Single Shot Multibox Detectors, or specially trained CNNs for this task.

2.3 SUMMARY

In this section we present tables summarizing the AMR methods found in the literature that use deep learning techniques. Dataset denotes if the work used a publicly available dataset. Time and hardware denote if the work presented times achieved and hardware used.



(a) Pointer extraction from Peng et al. (2021)



(b) Keypoint detection from Howells et al. (2021) using specially trained CNNs

Figure 2.3: The two types of approaches for the angle method.

Tab. 2.1 summarizes the works that designed an AMR method for digit meters. For meter types, "A" denotes meters with an analog rolling digit display, and "D" denotes digital electronic displays.

Tab. 2.2 summarizes the works that designed an AMR method for pointer meters.

Table 2.1: Digit Meters

Year	Work	Dataset	Data type	Meter type	DL stages	Model used	Time	Hardware	Notes
2019	Son et al. (2019)		Gas	A	S1, S3	YOLO, VGG		X	
2020	Azeem et al. (2020)	X	Energy	A/D	all	Mask-RCNN		X	
2019	Shuo et al. (2019)		Energy	A/D	S1	MobileNetv2-SSD		X	
2018	Košcevic and Subašic (2018)		G/E	A/D	S3,S5	Faster R-CNN	X	X	
2020	Bishwokarma et al. (2020)		Energy	A/D	all	YOLOv3		X	Mobile
2020	Peng and Chen (2020)		Water	A	all	R-FCN		X	
2019	Waqar et al. (2019)		Energy	A/D	all	Faster RCNN			
2019	Calefati et al. (2019)		Utilities	A/D	all	CNN	X	X	
2019	Li et al. (2019)		Water	A	all	CNN			
2021	Dzeha et al. (2021)	X	Energy	A/D	all	CNN	X		Mobile
2022	Zhu et al. (2022)	X*	Water	A	all	YOLOv4-tiny		X	
2016	Cerman et al. (2016)		Energy	A	S2	CNN	X	X	
2014	Chouiten and Schaeffer (2014)		Gas	A	S1	CNN		X	Mobile

* The dataset was not available at the time of writing

Table 2.2: Pointer meters

Year	Work	Dataset	Data type	Meter type	DL stages	Model used	Time	Hardware	Notes
2019	Fang et al. (2019)		Industrial	Gauge	S1	Mask-RCNN*		X	
2021	Zhou et al. (2021)		Industrial	Gauge	S1	YOLO		X	
2020	Ueda et al. (2020)		Industrial	Gauge	S1,S3	SSD			Mobile
2019	He et al. (2019)		Industrial	Gauge	S1	Mask-RCNN	X	X	
2020	Zuo et al. (2020)		Industrial	Gauge	S1	Mask-RCNN	X	X	
2021	Peng et al. (2021)		Industrial	Gauge	S1,S2	YOLOv4/U-Net	X	X	
2022	Zhou et al. (2022)		Industrial	Gauge	S1,S2	YOLOv5	X	X	
2021	Howells et al. (2021)	X	Industrial	Gauge	S1, S3	MobileNetv2(PyTo)	X	X	Mobile
2020	Salomon et al. (2020)	X	Energy	Multi-dial	all	ResNeXt-101	X	X	Residential

2.4 CONCLUDING REMARKS

Most of the AMR methods for residential meters (water, electricity, gas) involve digit meters (Azeem et al., 2020; Shuo et al., 2019; Košcevic and Subašic, 2018; Bishwokarma et al., 2020;

Waqar et al., 2019; Calefati et al., 2019; Dzeha et al., 2021; Zhu et al., 2022; Cerman et al., 2016; Chouiten and Schaeffer, 2014), while methods for pointer meters involve mostly gauges in industrial settings (Fig. 2.4(b)) (Fang et al., 2019; Zhou et al., 2021; Ueda et al., 2020; He et al., 2019; Zuo et al., 2020; Peng et al., 2021; Zhou et al., 2022; Howells et al., 2021). There are also methods that focus on mobile application solutions (Ueda et al., 2020; Bishwokarma et al., 2020; Dzeha et al., 2021; Howells et al., 2021; Chouiten and Schaeffer, 2014). This work will focus on the less studied multi-dial residential pointer meters (Fig. 2.4(a)), that we found only in the UFPR-ADMR dataset (Salomon et al., 2020).

Sossa (2013) was another previous work that tackled AMR for multi-dial meters, however they were not included in the review above since deep learning techniques were not used. The authors used only traditional image processing techniques (Scale-invariant feature transform for RoI and image binarization finding the pointer). Since it uses image binarization, it is closer to the feature extraction approaches found in the previous section.

The baseline presented by Salomon et al. (2020) on their dataset does not use feature extraction or keypoint detection, it uses a CNN to directly classify the result upon detecting the dials. As such, the aim of this work is studying the feasibility of detecting keypoints with deep learning (Fang et al., 2019; Ueda et al., 2020; Zhou et al., 2022; Howells et al., 2021) on residential multi-dial meters, and its possible challenges when compared to its detection on gauges.

There may be difficulties present in detecting keypoints of multi-dial meters as the meters will often present multiple keypoints belonging to different scales in close proximity to one another. There are also other significant topology differences between the 2 types of pointer meters, such as the start and end of the scales forming a full circle in multi-dial meters, with each dial representing one digit (0-9) of the value (Fig. 2.4(a)), as opposed to gauges that have a distinct start and end (Fig. 2.4(b), 2.2(a)). When detecting keypoints to calculate the reading, the value can be calculated using the angle between the dial and the start of the scale (Zhou et al., 2021; Ueda et al., 2020; He et al., 2019; Zuo et al., 2020; Peng et al., 2021; Zhou et al., 2022; Howells et al., 2021), therefore, the accuracy of such a method is dependent on how accurate the keypoint detection is.

Another characteristic to note is that multi-dial meters do not have a standard number of dials (Fig. 2.2(b), 2.4(a)). Furthermore, different scales in a meter have alternating directions (clockwise, counter-clockwise) and may also share numbers, as seen in Fig. 2.4(a).

(a) multi-dial meter from UFPR-ADMR

(b) image of an industrial gauge from Ueda et al. (2020)

Figure 2.4: Two examples of real pointer meters.

3 PROPOSED APPROACH AND MOTIVATION

The proposed approach aims to explore the possibility of applying keypoint detection, popular in AMR methods for industrial gauges, for residential multi-dial meters, a subject largely unstudied, seen before only on Salomon et al. (2020) and its related works as far as we are aware. Our approach uses YOLOv7 (Wang et al., 2022), an object detection model to detect each digit, and all keypoints of each digit (center of the dial, edge of the pointer, and zero on the scale) in a single stage. Fig. 3.1 shows a result of the detection for our approach.

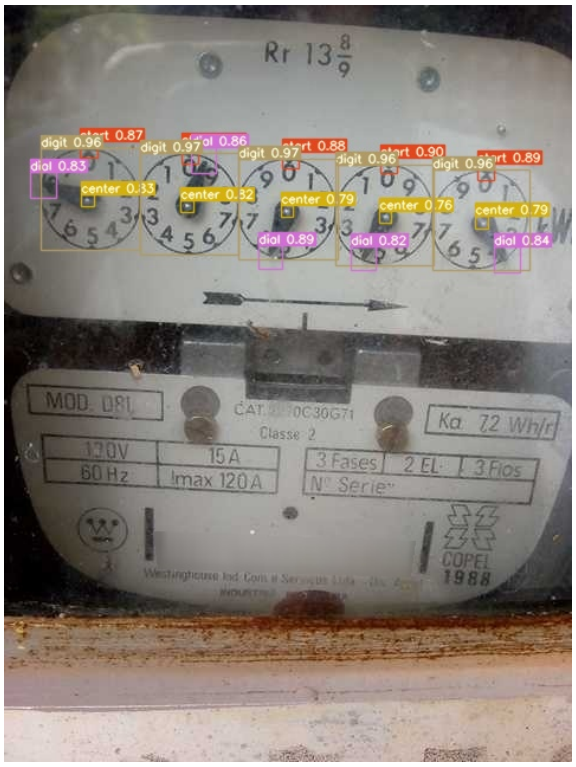


Figure 3.1: Detection of all digits and keypoints using YOLOv7.

3.1 YOLOV7 OBJECT DETECTOR

The YOLO (You Only Look Once) is an algorithm proposed for object detection (Redmon et al., 2015). It works by dividing the input image into regions and predicting bounding boxes and probabilities of an object being present for each region. The YOLO family showed frequent utilization in the AMR field, and at the time of experimenting, YOLOv2 was the oldest and YOLOv5 was the most recent found in the literature.

YOLOv7 is a single stage real-time object detection model based on a Fully Connected Neural Network architecture. It was released in July 2022 and according to the authors it was the fastest and most accurate object detector at the time (Wang et al., 2022). Other than architectural

changes when compared to previous YOLO models (mainly the usage of the E-ELAN network), the model's weights were trained on the COCO dataset (Lin et al., 2014). It showed better results than all previous YOLO models.

When calculating the angle of the dial, the two approaches found in the literature are detecting the keypoints and extracting the region of the image which contains the dial. Since using object detectors to detect keypoints was an approach implemented before (Ueda et al., 2020; Zhou et al., 2022), and the dataset used as base (Salomon et al., 2020) already had annotations with bounding boxes for the digits fitting for use with a YOLO model, the approach chosen was that of keypoint detection with YOLOv7 (based on the PyTorch framework). It was chosen for this approach because it was the best and most recent object detector at the time of experimenting.

3.2 DIGIT AND KEYPOINT DETECTION

The model is trained to detect all digits and all the keypoints in the input image in a single stage. The keypoints the model is trained to detect are: the center of the dial, the point of the dial, and the starting position of the scale (Fig. 3.3). Since YOLO detects a bounding box, the values that should be used in a future step to calculate the angle of the dial are the center coordinates of the bounding boxes. Fig. 3.2 shows the flowchart of our single stage approach.

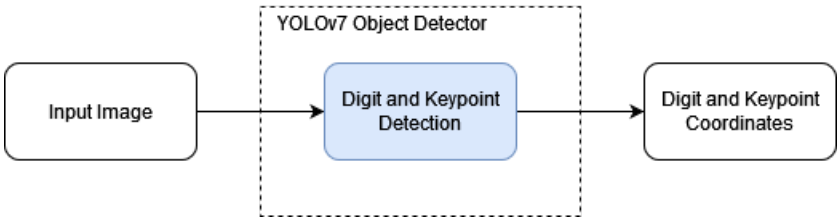


Figure 3.2: Flowchart of the single stage approach.

The keypoints detected for each digit are shown in Figure 3.3. From these keypoints the value of the digit can be calculated in a future step by measuring the angle between the dial and start from the center. For this dataset the network does not need to detect whether a digit is in a clockwise or counter-clockwise orientation because this information can be inferred from the total number of digits (the least significant digit is always in a clockwise configuration, and each digit alternates orientation).

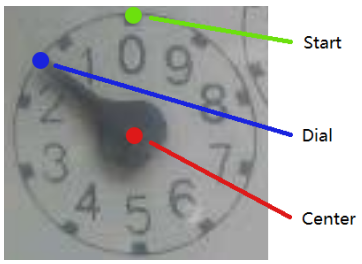


Figure 3.3: Keypoints of one digit.

Since the YOLOv7 labeling format requires a bounding box for each object, the coordinates of the keypoint were used as the center of the bounding box, and the size of the box was based on the size of the annotated bounding box for the corresponding digit, with each keypoint using a different proportion. The proportions were adjusted empirically, increasing from a starting value of 0.05 of the digit's bounding box until satisfactory results were achieved in the detection of each keypoint. The final proportions were 0.125 for the center, 0.25 for the dial, and a sixth for the start. The results of a detection can be seen in Fig. 3.1

The bounding box of the dial is the main difference between our approach and the two others that used object detectors to find keypoints (Ueda et al., 2020; Zhou et al., 2022). In their work they did not directly detect the dial keypoint, they detected the entire dial arm and calculated the position based on that.

This approach also differs from other methods found in 2.2.1 as it does not detect the RoI first, as Salomon et al. (2020) found that for this dataset they achieved better results without it. Another difference is that most methods for industrial gauges are highly dependant on the types of meters found in their datasets or manually pre-determined values, as different types of meters might not use the same scale (having varying minimum and maximum values between models) while residential multi-dial meters, regardless of number of digits, will always have only a starting point at the value zero.

As the goal was only to study the feasibility of keypoint detection on this dataset, the model does not calculate the reading value of the meters, as problems such as perspective correction and parallax (the dial and the scale are not in the same plane) are not significantly different than on gauges. Given our method is a single stage object detector, it can be easily implemented into a new end-to-end approach, or an existing one. Fig. 3.4 shows a hypothetical approach including our method.

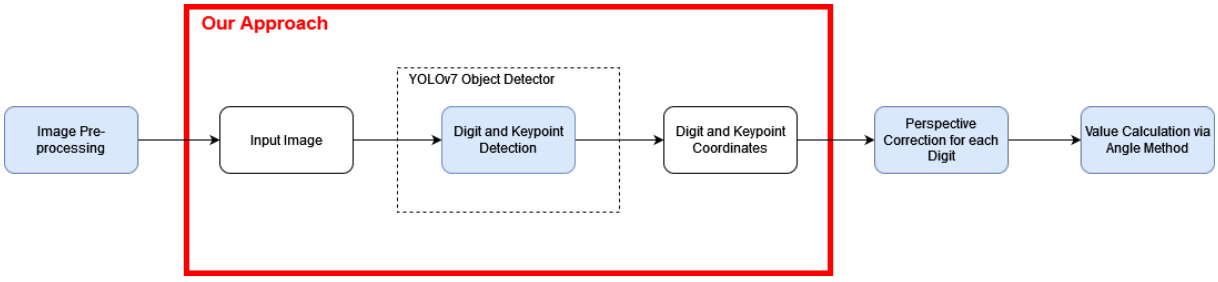


Figure 3.4: Flowchart of a possible end-to-end approach including our method.

An advantage this approach may present when compared to other keypoint detection methods and Salomon et al. (2020) is that small inaccuracies in the dial position might not affect the final result, as each digit is an integer. And for cases where the angle is found to be close to a mark on the scale, the previous digit can be used as context to decide if the digit should be slightly above or below the mark by checking if the previous digit is below or above the value 5, in the worst case (all middle digits are found to be close to the "5" mark, which shouldn't be possible in a calibrated meter) being dependant on the accuracy of the least significant digit.

4 RESULTS AND DISCUSSION

The performance of the trained model was evaluated based on the number of digits and keypoints it correctly detected on the 100 test images. The model used was the one based on the PyTorch framework, and was trained for 250 epochs with 200 training images and 200 validation images using multiple GPUs (Quadro RTX 8000, NVIDIA GeForce RTX 2080 SUPER and NVIDIA TITAN Xp). The pre-trained weights chosen were the medium sized weights trained on the COCO Dataset (Lin et al., 2014), available by default on the YOLOv7 model (Wang et al., 2022), as they strike a balance between size and precision when compared to their tiny and large versions. The initial learning rate was 0.01 and the final learning rate was 0.1. The IoU training threshold was 0.2. The average time of each training epoch was 18 seconds.

4.1 DATASET

The dataset used as a base was the UFPR-ADMR (Salomon et al., 2020) which is the only public dataset containing residential multi-dial meters we found. It includes 2000 images and is already annotated with digit locations, so for this work 500 images were annotated to include the 3 extra classes necessary for the keypoints to be detected. Each Image has one meter, and each meter has either 4 or 5 digits, with each digit having 3 keypoints, giving a total of 2273 digits (54.6% of the meters in our dataset are 5-digit meters) and 6819 keypoints for our dataset.

The UFPR-ADMR dataset consists of images with 640x480px resolution, and has the following annotations: the value of the readings, the coordinates (in pixels) of the four corners of the irregular quadrilateral containing the digits, and for each digit: the coordinates (in pixels) of the upper left corner, the lenght and height (in pixels) of the box containing the digit. The 500 images used for this work were further annotated with the coordinates (in pixels) of the center, dial and start for each digit, as seen in Fig. 3.3.

For the training of the method shown in this work, only the bounding box of the digit and the keypoints were used. The dataset is public, available upon request (Salomon et al., 2020) and our annotations are available at: https://github.com/FabianoAizawa/Keypoints-for-UFPR-ADMR Fig. 4.1 shows an annotated image from our dataset.

4.1.1 Division of the Dataset

The 500 images in our dataset were divided into 200 training images, 200 validation images and 100 testing images. The training images contain 914 digits (57% 5-digit meters) and the validation images contain 915 digits (57.5% 5-digit meters). In the 400 images used in training, there are a total of 1829 digits and 7316 keypoints.

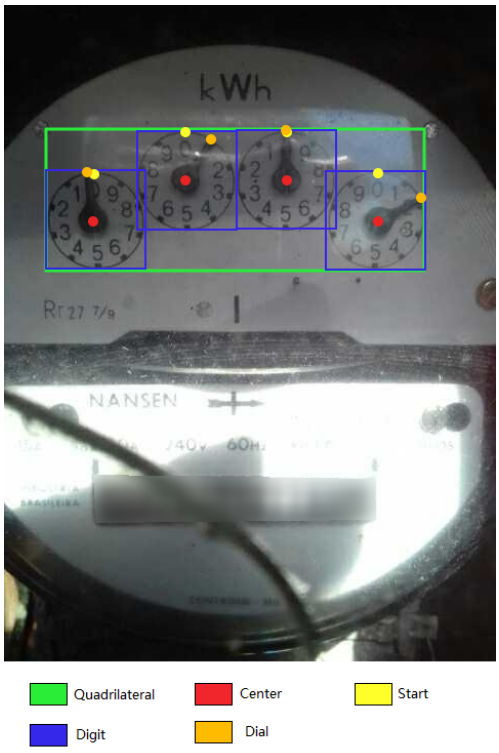


Figure 4.1: Example of an annotated image.

There are 100 images in the testing set, containing in total 444 digits (with 3 keypoints for each digit, there are 1332 keypoints in this set). The test images have on average 17.76 objects to be detected (16 objects for meters with 4 digits, and 20 for meters with 5).

4.2 DATA AUGMENTATION

YOLOv7 allows configuration of its parameters (such as learning rates, IoU thresholds, weight decay, etc.) along with data augmentation via its hyper-parameters. Initial experimentation found that the model was having difficulties detecting dials which the color was too similar to the background of the meter (mostly gray metallic dials, Fig. 4.2). To counteract this problem, the hyper-parameter of HSV hue augmentation was increased to 0.5 (saturation and value augmentation were already at sufficient values). All other parameters were not changed.

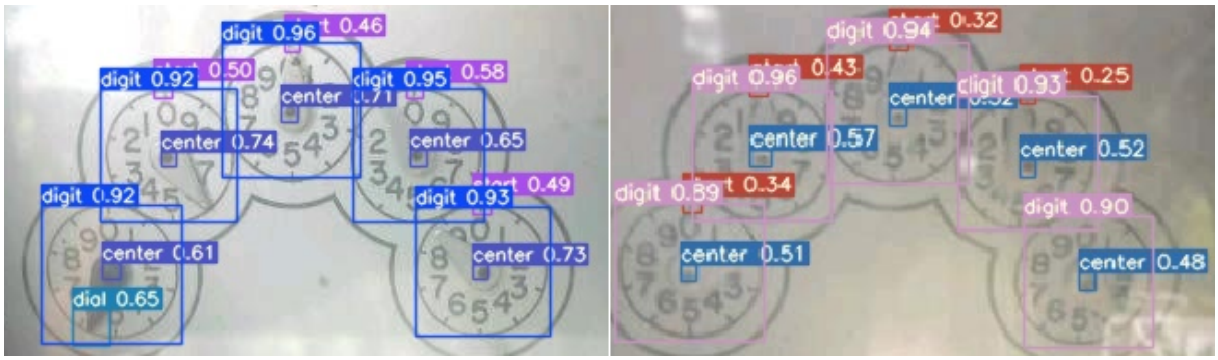


Figure 4.2: Examples of detection before data augmentation was adjusted.

4.3 OVERALL RESULTS

The model achieved an average of 99.15% of digits and keypoints detected, with an average detection time of 13ms for each image. The overall results for each object can be seen in Table 4.1.

Object	Detection Rate (%)	Detected	Total
All	99.15%	1761	1776
Digit	100%	444	444
Center	100%	444	444
Dial	98.64%	438	444
Start	97.97%	435	444

Table 4.1: Overall Results of our model.

Our model correctly detected all digits and center keypoints, and achieved greater than 98% rate at detecting dials and 97% rate at start keypoints. Figure 4.3 shows results from detections with our model.

Figure 4.3: Examples of detection on different meter models.

On a per image accuracy, we achieved 90% detection rate, meaning in 90 of the 100 test images, we correctly detected all digits and all keypoints without detecting any false positive.

4.4 DIGIT DETECTION

Our model correctly detected all 444 digits, even on the meter models where digits overlap with the neighboring digits (Fig. 4.4). The digits overlap on these models because the annotations for the bounding boxes of the digits also overlap in the UFPR-ADMR dataset, as they include the surrounding scale numbers as part of the digit. The overlapping digits did not directly impact in the detection of the other keypoints, however, the compact configuration of this type of meter leads to a problem, as we will see in Section 4.7.1.

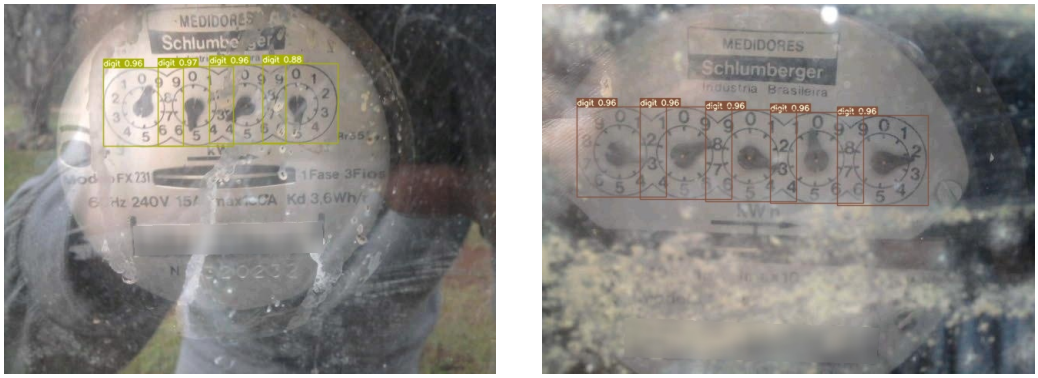


Figure 4.4: Two examples of meters with overlapping digits. The 3 keypoints for each digit are not being shown in the images, but in these two images they were all detected.

The average confidence of the digits is very high, greater than 0.95, so any false-positives can be discarded due to a low confidence value. The incorrect digit bounding box with the highest confidence detected in our test set had 0.68 confidence (Shown in Fig. 4.5), while the lowest confidence value for a digit was a single outlier at 0.74 (the second lowest was at 0.88 confidence).

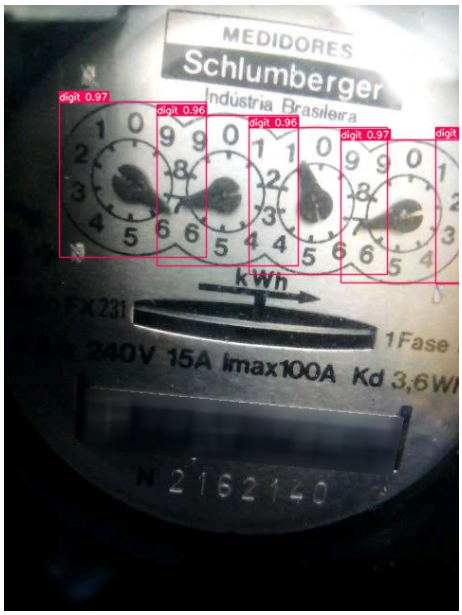


Figure 4.5: The test image which had the highest confidence value for an incorrect digit bounding box (0.68 confidence). The 3 keypoints not shown for each actual digit were correctly detected.

Most incorrect bounding boxes found for the digits were found in images of meters with overlapping digits were the most or least significant digit were partially outside the image, so our model was predicting with a lower confidence that the edge of another digit was in that part of the image (Fig. 4.5).

A confidence value of 0.7 or higher is already enough to discard all false positives in the digit detection, however it's also possible to discard digits via their lack of keypoints or via their coordinates and dimensions.

4.5 FALSE POSITIVES

Given how reliably we can detect the digits in each image, incorrect bounding boxes in the detection of keypoints are only considered to be false positives in the case of an incorrect bounding box having a confidence value higher or equal than the lowest confidence value of the corresponding keypoint class. This is because with the number of digits we can pick the corresponding number of keypoints starting with the highest confidence value and discard the rest.

As an example, in Fig. 4.6 the network correctly detected 4 digits, 4 centers, and 4 starts. However, with a confidence threshold of 0.3 in the detection, it detected 5 dials, the 4 correct dials, and one more with a confidence of 0.32 at the shadow of the least significant dial. Since the digits were all correctly detected, we know there are 4 dials in the image, discarding the incorrect dial with the lowest confidence.

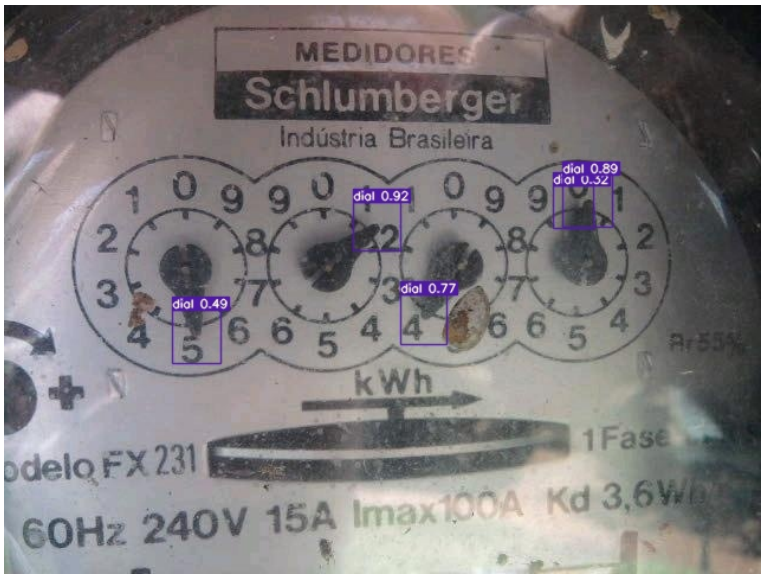


Figure 4.6: An image from the testing set were one dial was incorrectly detected.

With this definition, on the training dataset 5 images detected false positives, 3 on the detection of the dial keypoint and 2 on the detection of the start keypoint. Figure 4.7 shows the 3 images were false positives were detected on the dial keypoints. In these images we see that the false positives were caused by uneven illumination, shadows and reflections on an image with

poor quality, respectively. We can also see on the second image that the confidence value of the least significant dial is much lower than the other dials because of uneven lighting and the shadow of the dial. Additionally the most significant dial of the second image was not detected because of a reflection.

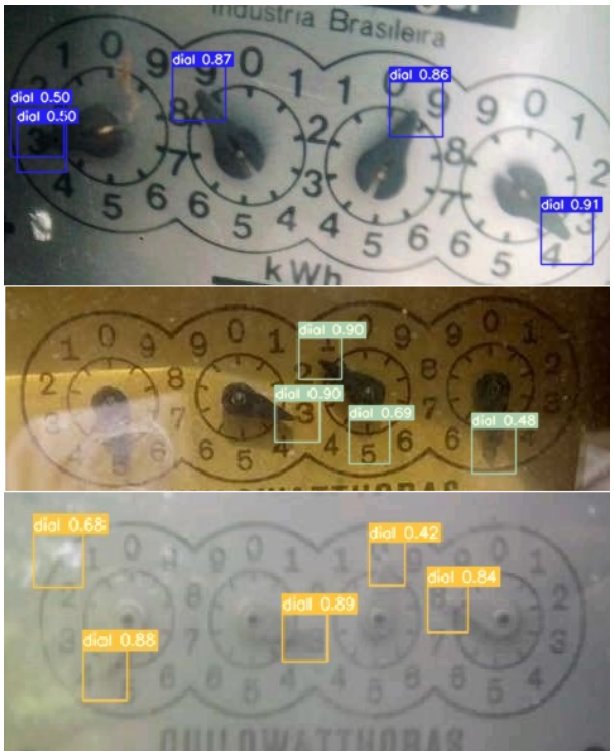


Figure 4.7: Figure showing all the false positives detected on the dial keypoint detection. In the second image the most significant dial was not detected. All other digits and keypoints not shown were correctly detected for these images.

The false positives detected on the start keypoints will be discussed in depth in Section 4.8.

4.6 CENTER KEYPOINT

All 444 center keypoints were detected with no false positives. Figures 3.1 and 4.3 show detection results including the detection of the center keypoints.

4.7 DIAL KEYPOINT

Detection of the dials achieved a detection rate of 98.64%, detecting 438 of the 444 dials in the testing set. There were 3 false positives detected (Fig. 4.7) and 5 other images were one dial was not detected.

4.7.1 Error Analysis

The detection of the dial proved to be the most challenging aspect for our approach, a total of 8 images presented either a false positive (Fig. 4.7), or failed to detect all dials.

The false positives detected are shown in Fig. 4.7 and represent common challenges in image recognition in real scenarios, such as contrast (first image), reflections and shadows (second image), and blurriness and glare (third image). It was here that most incorrect bounding boxes were detected, in particular on the shadows of the dials. Other than the images shown in Fig. 4.7 all other incorrect boxes had low confidence scores and were not treated as false positives as the confidence values were below the actual dial with the lowest confidence scores, however it is still worth noting that shadows presented difficulties for our model.

On Figure 4.8 we see an interesting problem that is not possible on gauges. Due to the close proximity of two dial keypoints, a single bounding box was predicted for both keypoints. What likely happened is that both dials generated bounding boxes in the divided regions of the image, however due to the proximity of the keypoints and size of object, the Non-Max Suppression algorithm combined both keypoints into a single object. This problem occurred on the meter type that overlaps digits, as their compact configuration means the keypoints can be closer together. This only happened in the 2 images of Figure 4.8, were the dials were in very close proximity and aligned. In other images were the dial keypoints were close, such as the 2 most significant digits of the first image of Fig. 4.8, Fig. 4.5, and the top right image of Fig. 4.3 the dial keypoints were detected correctly.

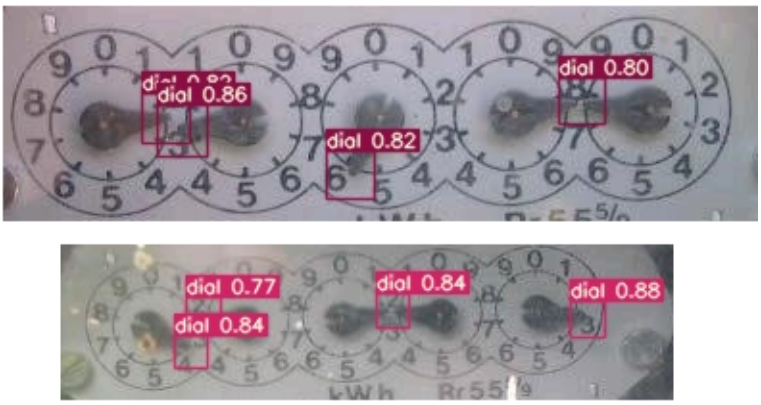


Figure 4.8: Figure showcasing the bounding box merging problem. For both images, digit, center and start keypoints not shown were all correctly detected.

Figure 4.9 shows the final 3 images were dial keypoints were not detected. They showcase classic issues in object detection, such as contrast, blur and glare problems in the image.

Possible improvements to the errors found here are expansion of the dataset and further adjustment of the data augmentation and other parameters.

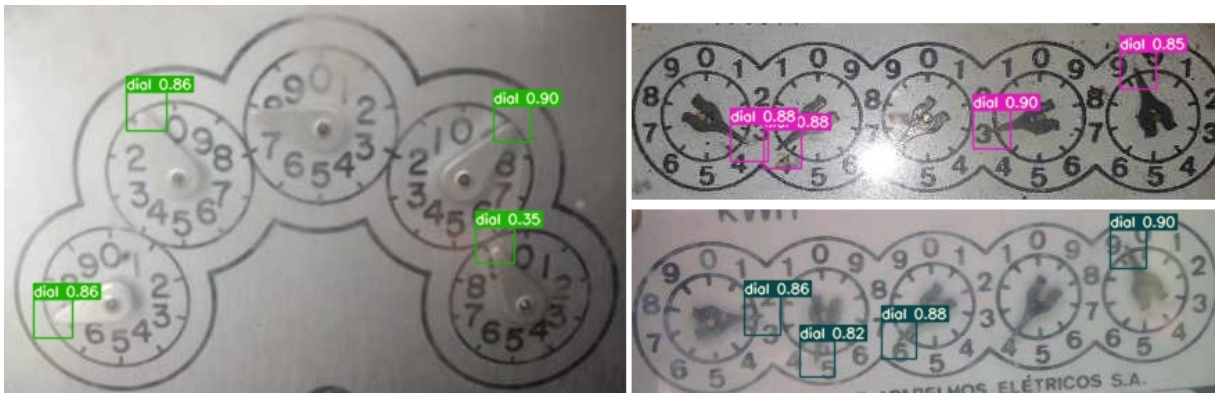


Figure 4.9: Figure showcasing classic image recognition problems. For all images, digit, center and start keypoints not shown were all correctly detected.

4.8 START KEYPOINT

Detection of the start keypoints achieved a detection rate of 97.97%, detecting 435 of the 444 starts in the testing set. The 9 keypoints not detected were in the same two images were the start false positives were found.

4.8.1 Start Keypoint Obstructed by the Dial

We found that the start being obstructed by the dial does not compromise the keypoint detection, as all digits in the testing set that had such characteristic had both the start and dial keypoint detected correctly. Fig. 4.10 shows a few examples of this.

Figure 4.10: Examples of digits where the dial is obstructing the start.

4.8.2 Error Analysis

The two images were the errors occurred are shown in Fig. 4.11, where we can see that all the starting keypoints (red bounding boxes) were detected on the wrong mark because the meter had a severe rotation.

Minor rotation is not a problem, as seen on Fig. 4.12, the wrong mark is only detected when the rotation is enough so that the highest mark in the digit belongs to another number.

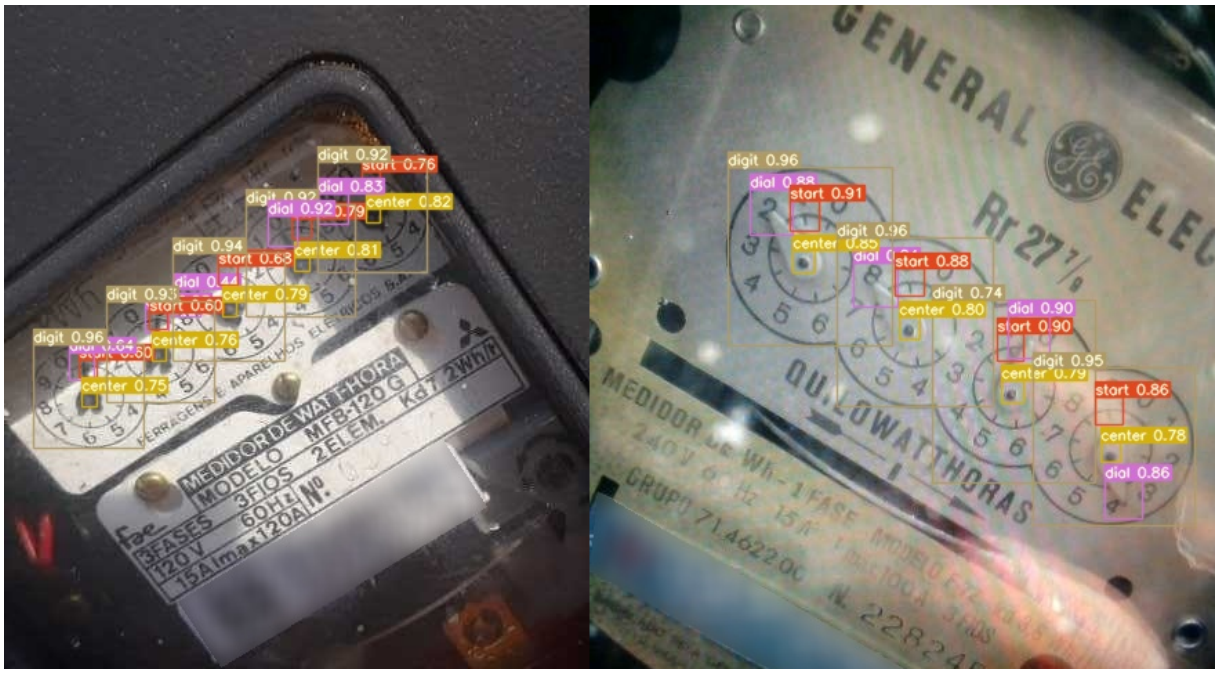


Figure 4.11: Figure showing the two images from the testing set that had a high degree of rotation. Other than the start keypoints, all digits and other keypoints were detected correctly.

This highlights that in our dataset there are few images with high rotation, meaning our model is trained to detect the highest mark on a digit instead of detecting the mark next to the value zero. A possible approach to dealing with this problem is presented in the next section.

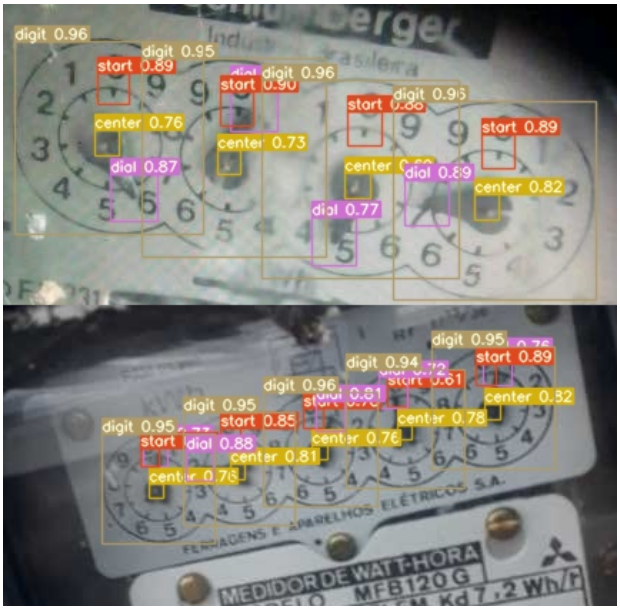


Figure 4.12: Examples of correct detections on slightly inclined meters.

4.8.3 Rectifying Major Rotations

A possible solution to the rotation problem is rectifying rotation in the image is by pre-processing the image before the detection stage. To demonstrate this, we manually rotated one of the images

were the starting points were incorrectly detected and tried detecting the keypoints on the rectified image. The results of the detection can be seen on Fig. 4.13.

Figure 4.13: Correct detection of all keypoints on the rectified image.

4.9 PROOF OF CONCEPT

As a proof of concept, we manually adjusted one of the images that had severe rotation as the pre-processing stage, then detected the keypoints on the adjusted image (our approach) and finally calculated the reading of the meter via the angle method. This is shown in Fig. 4.14. Such an approach would be similar to the possible approach ilustrated in Fig. 3.4.

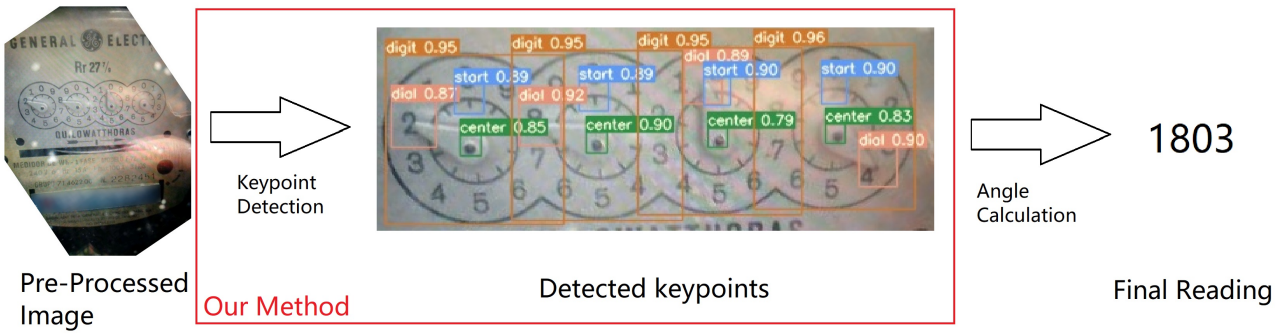


Figure 4.14: The angles found for each digit are: 64º, 291º, 15º and 129º.

The pre-processing of the image can be done in various ways, with the model trained here the correction of the angle can be done with the coordinates of the most and least significant digit, as they are always aligned horizontally. However for the rotation correction to be done this way, the detection of the digits must be done in a separate stage.

4.10 CONCLUDING REMARKS

The approach is promising for AMR of multi-dial meters, however it still needs to be applied in an end-to-end approach in order to test and further improve its accuracy on the final readings. Another point of improvement is the expansion of the dataset and cross-dataset experimentation,

as there is a need to verify whether the particular errors found on the dial keypoint detection are a challenge specific to these types of meters, as with a small dataset we identified possible new challenges in the merging of bounding boxes and the possibility of shadows posing a particular challenge for multi-dial meters given the proportions of the meter features when compared to gauges.

An open question not answered by our work is the viability of the other approach for the calculation of the angle method, feature extraction, and how it compares with keypoint detection.

5 CONCLUSION

Image-based AMR is a viable solution for the laborious task of reading outdated industrial and residential utility meters. In this work we did a review of the AMR literature, presenting its main motivations, challenges and summarizing the methods and techniques used in the field. With that we identified an area where few research was done, AMR methods for residential multi-dial meters.

The only previous work we found in the area was Salomon et al. (2020) and their related dataset, UFPR-ADMR. We expanded upon this dataset annotating 500 images with extra labels necessary to perform keypoint detection. With this new labels we applied a method used in gauges, keypoint detection, while highlighting the differences between the 2 types of meter and pointing the possible challenges unique to multi-dial meters.

While object detectors, such as YOLO and SSD, are usually used in the literature for RoI detection, they have been used before for keypoint detection in gauges. We implemented this approach for multi-dial meters. Considering the relatively small dataset size, using YOLOv7 to detect keypoints in our approach showed promising results, achieving an overall detection rate of 99.15%, showing that keypoint detection for multi-dial meters is a viable strategy.

There are still improvements to be made such as developing an end-to-end approach using our method and expanding the dataset so we can evaluate and improve its accuracy more appropriately.

REFERENCES

Azeem, A., Riaz, W., Siddique, A., and Saifullah, U. A. K. (2020). A robust automatic meter reading system based on mask-rcnn. In *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications(AEECA)*, pages 209–213.

Bishwokarma, R., Paudyal, B., Chapagain, P., Bajgain, S., and Shakya, H. (2020). Deep neural network based automatic system for electricity meter reading in nepal.

Calefati, A., Gallo, I., and Nawaz, S. (2019). Reading meter numbers in the wild. In *2019 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6.

Cerman, M., Shalunts, G., and Albertini, D. (2016). A mobile recognition system for analog energy meter scanning. In Bebis, G., Boyle, R., Parvin, B., Koracin, D., Porikli, F., Skaff, S., Entezari, A., Min, J., Iwai, D., Sadagic, A., Scheidegger, C., and Isenberg, T., editors, *Advances in Visual Computing*, pages 247–256, Cham. Springer International Publishing.

Chouiten, M. and Schaeffer, P. (2014). Vision based mobile gas-meter reading. In *Scientific Cooperations International Workshops on Electrical and Computer Engineering Subfields*, pages 94–97.

Dzeha, E. E., Owusu, D., Mills, G. A., Pi-Bansa, I. B., and Sowah, R. A. (2021). Mobile application for electricity meter reading and billing using image processing and machine learning. In *2021 IEEE 8th International Conference on Adaptive Science and Technology (ICAST)*, pages 1–7.

Fang, Y., Dai, Y., He, G., and Qi, D. (2019). A mask rcnn based automatic reading method for pointer meter. In *2019 Chinese Control Conference (CCC)*, pages 8466–8471.

He, P., Zuo, L., Zhang, C., and Zhang, Z. (2019). A value recognition algorithm for pointer meter based on improved mask-rcnn. In *2019 9th International Conference on Information Science and Technology (ICIST)*, pages 108–113.

Howells, B., Charles, J., and Cipolla, R. (2021). Real-time analogue gauge transcription on mobile phone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2369–2377.

Košcevic, K. and Subašic, M. (2018). Automatic visual reading of meters using deep learning. In *Croatian Computer Vision Workshop*, pages 1–6.

Laroca, R., Barroso, V., Diniz, M. A., Gonçalves, G. R., Schwartz, W. R., and Menotti, D. (2019). Convolutional neural networks for automatic meter reading. *Journal of Electronic Imaging*, 28(1):013023.

Li, C., Su, Y., Yuan, R., Chu, D., and Zhu, J. (2019). Light-weight spliced convolution network-based automatic water meter reading in smart city. *IEEE Access*, 7:174359–174367.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2014). Microsoft coco: Common objects in context.

Peng, J., Xu, M., and Yan, Y. (2021). Automatic recognition of pointer meter reading based on yolov4 and improved u-net algorithm. In *2021 IEEE International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 52–57.

Peng, Y. and Chen, Z. (2020). Application of deep residual neural network to water meter reading recognition. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 774–777.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2015). You only look once: Unified, real-time object detection.

Salomon, G., Laroca, R., and Menotti, D. (2020). Deep learning for image-based automatic dial meter reading: Dataset and baselines. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Salomon, G., Laroca, R., and Menotti, D. (2022). Image-based automatic dial meter reading in unconstrained scenarios. *Measurement*, 204:112025.

Shuo, H., Ximing, Y., Donghang, L., Shaoli, L., and Yu, P. (2019). Digital recognition of electric meter with deep learning. In *2019 14th IEEE International Conference on Electronic Measurement Instruments (ICEMI)*, pages 600–607.

Son, C., Park, S., Lee, J., and Paik, J. (2019). Deep learning–based number detection and recognition for gas meter reading. *IEIE Transactions on Smart Processing Computing*, 8:367–372.

Sossa, H. (2013). Automatic reading of electro-mechanical utility meters.

Ueda, S., Suzuki, K., Kanno, J., and Zhao, Q. (2020). A two-stage deep learning-based approach for automatic reading of analog meters. In *2020 Joint 11th International Conference on Soft Computing and Intelligent Systems and 21st International Symposium on Advanced Intelligent Systems (SCIS-ISIS)*, pages 1–6.

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*.

Waqar, M., Waris, M. A., Rashid, E., Nida, N., Nawaz, S., and Yousaf, M. H. (2019). Meter digit recognition via faster r-cnn. In *2019 International Conference on Robotics and Automation in Industry (ICRAI)*, pages 1–5.

Zhou, D., Yang, Y., Zhu, J., and Wang, K. (2022). Intelligent reading recognition method of a pointer meter based on deep learning in a real environment. *Measurement Science and Technology*, 33(5):055021.

Zhou, T., Wei, H., and Zhang, K. (2021). An automatic pointer meter reading method based on deep learning in gas gathering station. *IOP Conference Series: Earth and Environmental Science*, 793(1):012002.

Zhu, J., Li, M., Jiang, J., Li, J., Wang, Z., and Shen, J. (2022). Automatic wheel-type water meter digit reading recognition based on deep learning. *Journal of Electronic Imaging*, 31(2):023023.

Zuo, L., He, P., Zhang, C., and Zhang, Z. (2020). A robust approach to reading recognition of pointer meters based on improved mask-rcnn. *Neurocomputing*, 388:90–101.