

# Under pressure: learning-based analog gauge reading in the wild

Maurits Reitsma<sup>1</sup>, Julian Keller<sup>1</sup>, Kenneth Blomqvist<sup>1</sup> and Roland Siegwart<sup>1</sup>

<sup>1</sup>Autonomous Systems Lab  
 ETH Zürich, Switzerland

**Abstract**— We propose an interpretable framework for reading analog gauges that is deployable on real world robotic systems. Our framework splits the reading task into distinct steps, such that we can detect potential failures at each step. Our system needs no prior knowledge of the type of gauge or the range of the scale and is able to extract the units used. We show that our gauge reading algorithm is able to extract readings with a relative reading error of less than 2%.

## I. INTRODUCTION

Robots are often tasked to inspect industrial plants. One critical part of inspecting industrial plants is reading analog gauges. While digital gauges exist, the vast majority of gauges are still analog. Analog gauges are designed to be easy for humans to read, but they are challenging to read using state-of-the-art computer vision algorithms. There are a plethora of different gauge designs. They vary in units and measurement scales. The gauge face is typically protected with glass, which shows reflections and can be covered with dirt. The gauges can be mounted in tricky positions which are impossible for the robot to approach and perceive head on, so reading needs to be possible from a variety of different orientations and angles. To build robots that can inspect and interact with legacy infrastructure, we need to develop algorithms that can reliably read analog gauges.

In this paper, we present a learning-based system to autonomously read analog industrial gauges [18]. Our system aims to achieve robust performance across gauges found in real-world scenarios with diverse appearance, scales and units. We do not require prior information about the gauge, ensuring broad applicability in industrial environments.

Our approach builds on the insight that all analog gauges we have seen have a face for the dial, a needle and a scale with notches along the way. We first detect and segment gauge faces and needles in an image. For each detection, we apply a keypoint detection algorithm to detect major notches on the scale of the gauge. As there can be a perspective shift and most gauges don't actually have a circular scale or they have a scale with a larger radius than the gauge face, we fit an ellipse to the detected keypoints. We then run optical character recognition (OCR) on the gauge face to detect units and scale markings on the gauge face, which we anchor to the ellipse. We fit a linear model with a RANSAC loop to the detected markings, as some might have been misread. We finally find the intersection point between the needle and the estimated ellipse and predict the gauge reading by interpolation using our linear model.

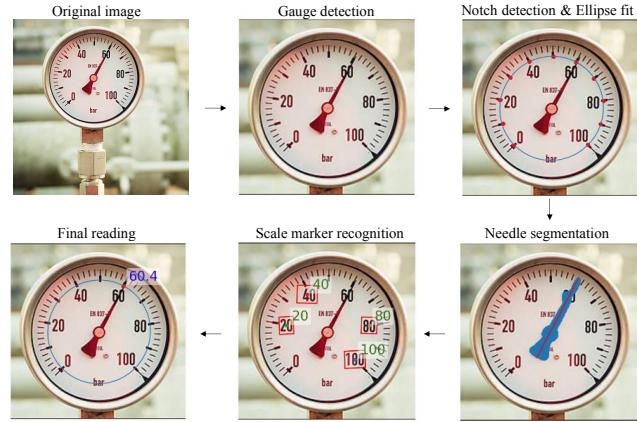


Fig. 1. Method overview. We first detect the gauge in the image, followed by detecting the notches and fitting an ellipse through them. Then, we segment the needle and fit a line through it. We detect and read the characters of the scale markers. Finally, we use the extracted information to compute the final reading.

We evaluate our algorithm on a set of analog gauges collected from the web, as well as a proprietary dataset of high quality pressure gauge images collected in an operating oil refinery. Additionally, we evaluate our system on the dataset provided by Howells *et al.* [10].

We show that our method yields a relative reading error of around 2%, beating the work of Howells *et al.* [10] on their dataset, while not requiring any prior gauge information, showcasing the potential of our algorithm. We analyze failure cases of our algorithm and show how our system may be improved in the future. We release our software as open source to allow future engineers and researchers to leverage our work and make further progress on the problem.<sup>1</sup>

## II. RELATED WORK

Several works in recent years have tackled autonomous analog gauge reading. Localizing the gauge is generally the first stage in a pipeline for reading gauges in recent works on the subject [19], [10], [3], [16], as it reduces the background noise for the further steps.

Similarly to us Howells *et al.* [10] use a learning-based approach. Their method extracts the start and end point of the scale, the center of the gauge and the needle tip to read the gauge. However, this assumes prior knowledge about the range of the scale of the gauge. We do not make this assumption which makes our algorithm more general.

<sup>1</sup><https://github.com/ethz-asl/analog-gauge-reader>

Some recent works do not make assumptions about the scale of the gauge. Dumberger *et al.* [3] incorporate an optical character recognition (OCR) step to read the scale. However, they impose additional restrictions. They assume that the gauge is not being partially covered by objects or and require the scale to be circular and spanning at least a range of 180 degrees. We avoid this making our algorithm capable of handling a wider range of gauge types and scenarios.

Another recent approach similar to ours, which incorporates an OCR stage, was proposed by Milana *et al.* [16]. However, there are some key differences between their method and ours. Milana *et al.* employed a technique where they warp the image, transforming the gauge face from an ellipse into a circular shape. They then directly compute the final prediction using polar coordinates on the warped image. In contrast, our approach assumes a circular shape of the gauge scale, rather than assuming the entire gauge face to be circular. This key distinction allows our method to apply to a wider range of gauge types. Furthermore, our approach offers an additional advantage when dealing with gauges that have multiple scales. Our approach allows the distinction between scale markers of the inner and outer scale of the gauge.

### III. METHODOLOGY

We propose a robust and modular analog gauge reading algorithm with the following stages: (1) gauge detection, (2) notch detection, (3) ellipse fitting, (4) needle segmentation, (5) character recognition of scale markings and units, (6) computing reading through linear model and interpolation.

To achieve a more robust system, we leverage some assumptions. Firstly, we assume that the gauge has a linear scale with at least 5 visible major notches on the scale aligned in an arc on the gauge face. We further assume that there are numerical scale markers along the scale and that the gauge face is flat. Additionally, we assume that the point between the start and end notch of an unrotated gauge is on the bottom. An overview is shown in Figure 1. In the following subsections, we describe each stage in detail.

#### A. Gauge detection

As a first step, we detect the gauge and crop the image to separate different gauges in the image and to remove any background noise. We use the smallest off-the-shelf YOLOv8 detector [11] to detect the gauge. We found the smallest model to be more than sufficient for this task. The model has been pretrained on the COCO dataset [15], which we further fine-tune on a small dataset of around 2,000 labeled gauge images. After detection, we crop and resize the image using bilinear interpolation to a resolution of (448, 448).

#### B. Notch detection

After gauge detection, we detect notches on the scale. We do this by designing a simple keypoint detector. We split notch detection into three different classes of keypoints: intermediate notches, the start notch and the end notch. The start and end notches are used to fix the orientation in a later

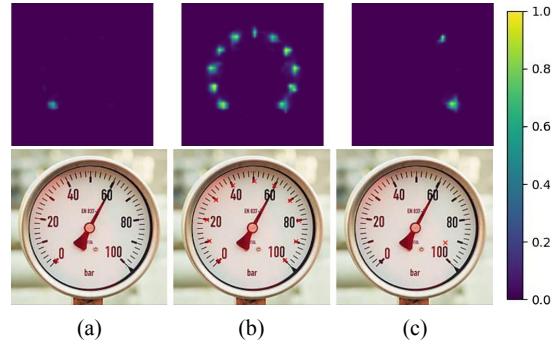


Fig. 2. Notch detection overview. First, we predict a heatmap to predict the probability of each pixel being a notch, then we cluster this heatmap to extract the individual notch points. In (b) we predict all notches, while in (a) and (c) we predict the start and end notch respectively.

stage of the pipeline, as detailed in Section III-E. We formulate keypoint detection as a heatmap prediction problem, followed by clustering to separate keypoint detections from each other. Figure 2 shows the different steps.

*1) Keypoint prediction:* To arrive at a vision model that can be trained from a small number of examples, we use a pretrained vision transformer DINoV2 backbone [17]. To predict heatmaps from the extracted features, we add a decoder structure consisting of a two-stage 1x1 CNN layer network, followed by a bilinear upsampling step. The upsampling is followed by a sigmoid activation. To construct the heatmaps for training, we place a Gaussian distribution centered at each keypoint location. We normalize heatmaps to have values in the range [0, 1]. We predict three heatmaps. One for intermediate notches, one for the start notch and one for the end notch.

To train the model, we freeze the backbone parameters and optimize the rest using a binary cross-entropy loss function and stochastic gradient descent and the Adam optimizer [12].

To extract keypoints from the predicted heatmaps, we use the Mean-Shift algorithm [2], clustering heatmap indices which have a value higher than 0.5.

#### C. Ellipse fit

Gauge scales are typically on an arc. To approximate the arc of the scale, we fit an ellipse through the coordinates of the keypoints of the detected notches. To fit an ellipse to a set of two-dimensional data points, we use the algorithm by Halir *et al.* [7]. It is a numerically stable version of the algorithm proposed by Fitzgibbon *et al.* [6]. Using direct least squares solving a corresponding eigensystem, the algorithm efficiently determines the optimal ellipse fit for the given data points. The ellipse fitting requires at least 5 notches to be detected.

#### D. Needle segmentation and line fit

We tackle the problem of estimating the position and direction of the gauge needle in two steps. In the first step, we segment pixels belonging to the needle using an object instance segmentation model. In the second step, we fit a line through the segmented pixels. Our approach implicitly assumes that the needle is directed towards the symmetry

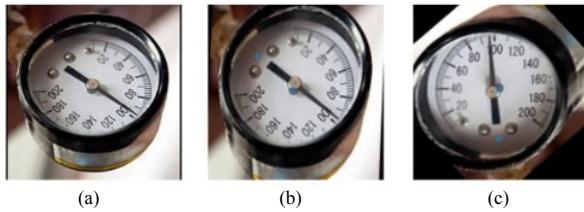


Fig. 3. Visualization of the perspective and rotation correction. (a) shows the original cropped image. (b) shows the image after perspective correction. (c) shows the image after rotation correction.

axis of the needle, which is a valid assumption for the vast majority of analog industrial gauges.

For the instance segmentation model, we use a COCO pretrained [15] YOLOv8 model [11]. We fine-tune the model on our dataset of around 2,000 labeled images of gauges. To fit the line through the segmented needle pixels, we use Orthogonal Distance Regression [1], as it minimizes the shortest distance of each point to the line.

#### E. Detection of scale numbers and unit

As we don't know the range of the gauge scale, we extract the numbers and unit displayed on the gauge face. We do this using an optical character recognition (OCR) model. We use a two-stage pre-trained deep neural network. The first network is dedicated to text detection, and the second network does text recognition. For text detection, we use the DBNet model [14], which is a segmentation model using a ResNet-18 backbone [9]. For text recognition, we use an ABINet model [4]. The ABINet architecture uses a combination of a vision model and a language model to predict the text in the detected bounding box. For both models, we use the MOCR library [13].

The OCR pipeline is sensitive to rotations and perspective shifts. To correct for this, we use the detected ellipse, start notch, and end notch. To correct for perspective, we warp the gauge image, such that the computed ellipse becomes circular. We assume that we have a circular scale of some, however large, radius. We use the start and end notch to estimate the orientation of the gauge by assuming the point between the start and end notch is on the bottom of the gauge. This allows us to estimate the gauge orientation and saves computation time, compared to exhaustively sampling many rotations. Both transformations are shown in Figure 3.

#### F. Computing the final reading

After running the previous stages of the pipeline, we compute the reading of the gauge using the following steps, which we explain in the following two subsections: (1) scale marker projection to ellipse and scale interpolation and (2) needle-to-ellipse intersection.

*1) Projection and interpolation of scale markers:* From the text recognitions from the OCR stage, we consider only the subset of recognitions that can be interpreted as numerical values. For each such detection, we project the center of the corresponding bounding box to the ellipse to anchor it.

By projecting scale markers onto the ellipse, we can determine the needle angle for each marker. Using the angles,

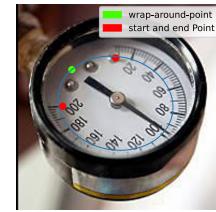


Fig. 4. Example of wrap-around point on ellipse. We take the middle point between the start and end point. This allows us to force the wrap-around point to be outside of the scale.

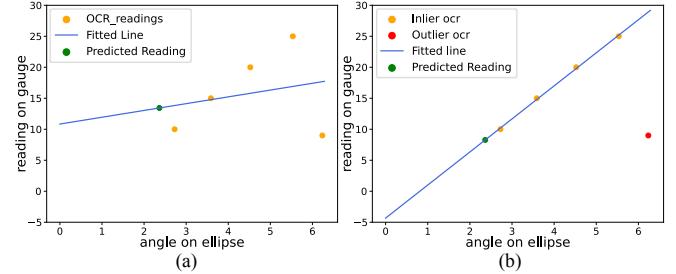


Fig. 5. Interpolation of the scale markers to fit the line mapping the ellipse angle to the gauge value reading. (a) Shows the least squares fit, which is heavily disturbed by outliers. (b) Shows the improved least squares fit by using RANSAC.

we fit a correspondence line that maps angles on the ellipse to a reading on the scale. This can be done using least squares on the angles of the scale markers, resulting in a model which we can interpolate.

However, linear regression with least squares is sensitive to outliers. Outliers can occur for many reasons, such as faulty OCR readings and detections of unrelated numbers on the gauge, such as a serial number. We remove outliers using RANSAC [5]. We fit a linear model mapping ellipse angles to value readings, filtering out any values that do not fit the model. Figure 5 shows an example of this, where RANSAC improves the final prediction. We fit a line, as we assume that the scale is linear. In cases where the scale represents a different function, another type of mapping would have to be chosen, possibly at runtime.

To support gauges with multiple scales, we can distinguish between scale markers for the inner and outer scale, by checking if they lie inside or outside of the ellipse. By making this distinction, we can proceed to fit two linear models, one for each scale, and subsequently calculate a reading for each scale.

When mapping a linear scale using angles from  $0$  to  $2\pi$ , the position of the zero angle becomes significant. We refer to this point as the "wrap-around point" on the ellipse. We choose the wrap-around point to be the point in between the scale start and end points, as shown in Figure 4. This guarantees that the chosen wrap-around point is outside of the scale, preventing discontinuities in the line fitting.

*2) Intersection of needle and ellipse:* The final step is to determine the point on the ellipse that the needle is pointing at. We do this by finding the intersection of the needle line and the ellipse. We first rotate and shift the needle line, such that the ellipse is centered at the origin. The intersection is computed using the equation of the line  $y = mx + c$  and the

ellipse. By computing the roots of the following quadratic equation, we find the two intersection points:

$$\frac{x^2}{a^2} + \frac{(mx + c)^2}{b^2} = 1$$

We then pick the intersection point that lies on our needle line. If this applies to neither or both points, we pick the one that is closer to either end of our needle line, as determined by the segmentation. This heuristic is based on the observation that the needle tip is usually closer to the gauge scale than the tail of the needle.

After determining the intersection point, we compute the angle of this point on the ellipse using the formula from Section III-F.1. With the needle angle, we predict the reading using the line model mapping angles to readings.

#### IV. EXPERIMENTS

We test our system on two datasets. The first consists of images of many different gauge types that are crawled from the web. The other consists of proprietary gauge images collected in an oil refinery. Both sets contain a variety of different gauge types.

To assess the performance of our system, we partition our dataset into several categories. The categorization is based on the specific difficulties posed by each image, allowing us to measure how our algorithm deals with these challenges. For the first category, we select images of gauges taken from the front. The second category consists of images viewing the gauge at an angle. The third contains images where the gauge is rotated and the fourth category is for images where the gauges are both rotated and viewed at an angle. The fifth and final category contains gauges that look significantly different from the gauges in our training set, for example having a non-conventional needle type and color, or a scale covering only a small part of the gauge face while having a very large radius.

We compare the estimated reading with expert-annotated labels. For cases where a reading cannot be computed or the reading significantly deviates from the true value, we identify the stage of the pipeline responsible for the failure and examine the underlying reasons.

We perform an ablation study, to evaluate the effect of the perspective and rotation correction on the OCR stage. Furthermore, we compare our algorithm method published by Howells *et al.* [10] on their dataset. This dataset consists of 450 images each of 5 different gauge types. We excluded meter\_d because as it has a nonlinear scale.

Next to evaluations on datasets, we deployed our system on a mobile robotic inspection and maintenance system, as shown in Figure 6. Our system allows for consecutive readings of gauges at different locations and heights within industrial plants.

#### V. RESULTS

In this section, we assess our experiments using two distinct approaches. First, we measure the average absolute and relative end-to-end reading error of our system.

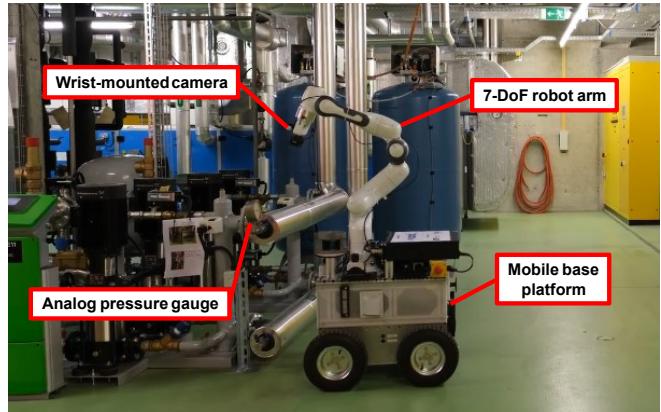


Fig. 6. Real-world gauge reading deployment on a mobile robotic base platform with manipulator and wrist-mounted camera.

Test set	OCR Success RE	Full RE
Front	1.15%	444%
Angled	0.77%	246%

TABLE I  
THE MEAN ABSOLUTE RELATIVE READING ERROR (RE) FOR TWO OF THE TEST SETS. FOR "OCR SUCCESS" WE CONSIDER EXAMPLES, WHERE OCR IS SUCCESSFUL. FOR "FULL" WE CONSIDER ALL EXAMPLES, WHERE A READING COULD BE COMPUTED.

Second, we evaluate the performance of each stage within our pipeline using various metrics and visual analysis. We explore different failure modes and limitations of our system.

##### A. End-to-end results

The relative reading error we consider in this section refers to the reading error expressed as the percentage relative to the entire range of the scale. When assessing reading error, we differentiate between two scenarios: whether the OCR successfully recognizes at least two scale markers correctly or fails to do so. The failure of the OCR stage can be attributed to any of the reasons outlined in section V-B.4.

As evident from Table I, if the scale markers are successfully recognized, our algorithm produces very accurate readings with a deviation of approximately 1% from the correct values on average. These results highlight the potential of our algorithm in determining gauge readings if the OCR stage operates effectively.

It is evident from the results in Table I that an incorrect detection of a scale marker negatively impacts the accuracy of the readings. Strategies to mitigate such cases are discussed

Image type	Our RF	Our RE	Howells RE
meter_a	100%	*	5.3%
meter_b	0%	0.96%	7.0%
meter_c	0%	1.46%	0.9%
meter_e	73.2%	3.1%	1.2%
meter_f	1.99%	2.5%	2.3%
Mean		2.01%	3.3%

TABLE II  
THE MEAN ABSOLUTE RELATIVE READING ERROR (RE) ON HOWELLS *et al.*. WE ALSO ADD THE READING FAILURE (RF) AS THE SHARE OF FRAMES WHERE READING FAILED.

Test set	Failure rates			
	Gauge	Ellipse	Needle	OCR
Front	0%	5%	2.5%	40%
Angled	0%	8%	0%	57.5%
Rotated	0%	5%	5%	42.5%
Angled & Rotated	0%	17%	0%	50%
Untypical	0%	50%	50%	-

TABLE III

FAILURE RATES OF DIFFERENT PARTS OF THE PIPELINE. WE DEFINE A FAILURE CASE OF A STAGE, WHEN IT EITHER HEAVILY DISTORTS THE READING NEGATIVELY OR WHEN IT PREVENTS THE ALGORITHM FROM COMPUTING A READING AT ALL.

Test set	Gauge IoU	Needle IoU	OCR %
Front	0.81	0.80	33 %
Angled	0.80	0.72	18 %
Angled & Rotated	0.81	0.75	29 %
Untypical	0.81	0	40 %

TABLE IV

EVALUATION OF DIFFERENT STAGES IN OUR PIPELINE. WE COMPUTE THE MEAN OF THE GAUGE DETECTION IOU, THE MEAN OF THE NEEDLE DETECTION IOU AND IN THE FINAL COLUMN THE MEAN SHARE OF THE DETECTED SCALE MARKERS. IOU REFERS TO INTERSECTION OVER UNION.

in detail in section V-B.4.

The results of the comparison we conduct between our approach and the method proposed by Howells *et al.* [10] using the dataset they provided are presented in Table II. Our approach yielded comparable or even better results than Howells *et al.* for most of the gauges.

For the gauge type of meter\_a our OCR model fails, which resulted in an inability to compute any reading. Additionally, for meter\_e our algorithm fails to compute a reading regularly due to issues in the OCR stage, as well as the fact that the type of notches in this meter were not represented in our training set. Consequently, the start and end point detection failed in approximately half of the cases for this meter. It is worth emphasizing again that our approach does not require prior knowledge about the full range of the gauge's scale, unlike the method proposed by Howells *et al.*. This aspect makes our approach applicable to a wider range of gauges.

### B. Individual stage results

In Table III we assess the frequency of failures caused by each stage of the pipeline. We define a failure of a stage, when it either heavily distorts the reading negatively, or when it prevents the algorithm from computing a reading in the first place. For instance, if we fail to detect multiple scale

Test set	% Detected	% Non-assigned
Front	94.6 %	2.2 %
Angled	82.9 %	9.7 %
Angled & Rotated	77.5 %	8.5 %
Untypical	.30 %	58.3 %

TABLE V

EVALUATION OF THE NOTCH DETECTION. WE COMPUTE THE MEAN SHARE OF DETECTED ANNOTATED NOTCHES AND MEAN SHARE OF NON-ASSIGNED PREDICTED NOTCHES. BOTH AVERAGES ARE TAKEN OVER ALL THE IMAGES IN EACH SET.

Test set	Nothing	Persp.	Rot. & Persp.
Front	35%	37%	33%
Angled	18%	20%	18%
Angled & Rotated	0	0	29%

TABLE VI

THE MEAN SHARE OF DETECTED SCALE MARKERS WITH NO CORRECTIONS, WITH ONLY PERSPECTIVE CORRECTION AND WITH BOTH PERSPECTIVE AND ROTATION CORRECTION.

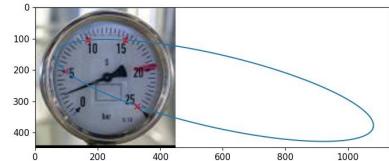


Fig. 7. A gauge, with only four notch detections. The fitted ellipse does not fit the scale well.

markers using OCR, we subsequently cannot fit a line and therefore cannot compute a reading.

In Table IV we evaluate the results of different parts of the pipeline. To assess the gauge detection and needle segmentation stages, we calculate the intersection over union (IoU) metric. For the OCR stage, we evaluate the performance by checking if each annotated scale marker has a predicted scale marker with an IoU of at least 0.5. If such a match exists, we define it as a detected scale marker. Otherwise, it is classified as undetected. In Table IV the average value of the share of the detected scale markers is depicted.

In the following sections, we will conduct a detailed analysis of the performance of each individual stage. By closely examining failure cases, we aim to propose solutions to prevent such occurrences in the future.

1) *Gauge detection:* Tables III and IV show the effectiveness of our gauge detection stage, as it successfully works for all the images in the test set. Additionally, the results in Table IV demonstrate that the gauge detection remains stable even when dealing with varying perspectives or gauges that differ significantly from those in the training set.

These findings are aligned with results of previous works [19], [10], who also used off-the-shelf detection models. Given that our detection model is not limited to the detection of a single gauge, it allows our system to read multiple gauges within the same image.

2) *Notch detection and ellipse fitting:* Table V shows the performance of notch detection on two different metrics. The first measures the share of annotated keypoints that have at least one predicted keypoint in close proximity. The second shows the share of predicted keypoints that are close to at least one annotated keypoint. The second metric is needed to evaluate the number of wrongly predicted notches, which can disturb the ellipse fit.

We find that the notch detection and the subsequent ellipse fitting work very robustly even with different perspectives, rotations, reflections and lighting conditions. This is supported by the corresponding results presented in Table V.

From the results in Table V, we can see that the performance drops slightly for angled and rotated gauges. Furthermore, for gauges that look significantly different from the

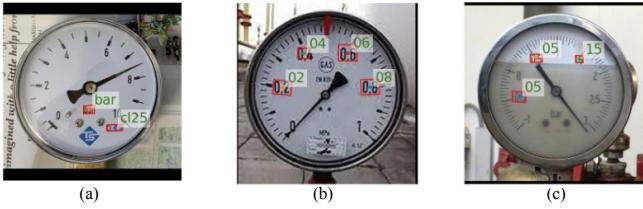


Fig. 8. OCR failures. (a) scale markers are not recognized. (b) decimal numbers are not recognized correctly (c) negative and decimal scale markers are not recognized.

gauge types in our training data, the performance deteriorates substantially. To address this issue, a potential solution is to augment the training set with a more diverse range of gauges, such that our notch detection model generalizes better.

If our system fails to detect five or more notches, the ellipse fit will not provide any useful results, resulting in the "ellipse failure" cases reported in Table III. An example of such a failure case can be seen in Figure 7.

The ellipse fitting failure could be addressed by either also predicting minor notches or segmenting pixels on the scale, and fitting the ellipse through the dense points. Another option would be to add a regularization term to the ellipse fitting which would bias the ellipse closer to a circular shape in the case of low data points. The axis of the needle could also be used as a constraint in ellipse fitting. An end-to-end differentiable approach predicting and supervising directly on an ellipse could also be explored.

3) *Needle segmentation*: The needle segmentation stage has shown good performance for most images in the dataset across various conditions, as shown in Table IV.

However, in our proprietary dataset, there were instances where the presence of needle shadows perturbed the segmentation result. This could be attributed to the absence of such cases in our training data. A potential solution would be to augment the training dataset with examples that include shadows, which could lead to a more robust model.

The proprietary dataset also contained gauges with a white needle on an almost white background, which our segmentation model failed to identify. These cases explain the zero IoU score in Table IV for the needle segmentation of untypical gauges. This is likely due to a bias in our training set, which almost exclusively contains needles with good contrast between the needle and the background. Similarly, this could be addressed with a more diverse training dataset for the segmentation model.

4) *Optical character recognition*: Thanks to our design choice, the OCR model only needs to recognize at least two scale markers for our algorithm to predict an accurate reading. This means that even with an imperfect OCR model, we still get reasonable results.

The OCR stage of our pipeline is the most common failure mode of our system. The OCR model struggles to recognize the digits of the scale markers in many cases, even for images captured from the front perspective. This is evident from the low success rate and the limited detection of scale markers shown in Tables III and IV. Figure 8(a) shows an example of a gauge where the OCR stage fails completely. In this case,

no reading can be obtained.

This reveals an insufficient performance of the pre-trained text detection model DBNet [14] for the scale markers on gauges. When qualitatively testing several other detection models like Mask-RCNN [8] or DRRG [20], we did not find any pre-trained model that was performing well universally among all our diverse types of gauges.

Additionally, our text recognition model ABINet [4] lacks the ability to detect decimal points and negative signs, limiting the scale markers we can read to natural numbers. This limitation can be observed in Figure 8(b) and (c).

A potential solution to these issues could involve fine-tuning an existing pre-trained model specifically on the scale markers of gauges or training a text detection architecture designed specifically for scale markers and designing a language model which would be better suited for gauge reading. The text recognition model could be fine-tuned or retrained specifically with decimal and negative numbers.

The results of the ablation study are shown in Table VI and demonstrate the impact of perspective correction and rotation correction. The application of perspective correction shows a slight increase of two percent in the detection of scale markers for gauges captured from an angle. However, there is still a noticeable discrepancy between predictions obtained from the front and from an angle, suggesting that our method of perspective correction may be insufficient. A learning-based approach for perspective correction for analog clocks and by extension for analog gauges was proposed by Yang *et al.* [19] and could be integrated with our algorithm.

The rotation correction plays a crucial role in enabling OCR detection for rotated images, as evident from the results in Table VI. This makes it an essential step for an effective OCR detection. However, it should be noted that rotation correction slightly decreases the detection rate of scale markers in other categories where rotation is not required. This is primarily due to cases where the start and end points are inaccurately estimated, resulting in incorrect gauge orientation prediction.

## VI. CONCLUSIONS

In this paper, we presented an algorithm to read analog gauges with minimal prior information about the gauge. Our gauge reading algorithm is able to read gauges within 2% relative precision. As shown in our experiments, the most error-prone part of our algorithm is the OCR step. Future work will focus on improving this step, possibly by fine-tuning the OCR reader on text specific to gauges and identifying helpful pre-processing steps.

Future work might also focus on building active gauge reading algorithms, where the robot reading the gauge is adjusting its viewpoint to maximize the readability of the gauge and possibly combining readings across images.

## VII. ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871542.

## REFERENCES

- [1] Paul T Boggs and Janet E Rogers. Orthogonal distance regression. *Contemporary Mathematics*, 112:183–194, 1990.
- [2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [3] Stefan Dumberger, Raimund Edlinger, and Roman Froschauer. Autonomous real-time gauge reading in an industrial environment. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 1281–1284, 2020.
- [4] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. 2021.
- [5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.
- [6] A. Fitzgibbon, M. Pilu, and R.B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.
- [7] Radim Halin and Jan Flusser. Numerically stable direct least squares fitting of ellipses. In *Proc. 6th International Conference in Central Europe on Computer Graphics and Visualization. WSCG*, volume 98, pages 125–132. Citeseer, 1998.
- [8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [10] Ben Howells, James Charles, and Roberto Cipolla. Real-time analogue gauge transcription on mobile phone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2369–2377, 2021.
- [11] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, Jan. 2023.
- [12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [13] Zhanghui Kuang, Hongbin Sun, Zhizhong Li, Xiaoyu Yue, Tsui Hin Lin, Jianyong Chen, Huaqiang Wei, Yiqin Zhu, Tong Gao, Wenwei Zhang, Kai Chen, Wayne Zhang, and Dahua Lin. Mmocr: A comprehensive toolbox for text detection, recognition and understanding, 2021.
- [14] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11474–11481, 2020.
- [15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, 2014.
- [16] Edoardo Milana, Oscar Hernán Ramírez-Agudelo, and Jacob Estevam Schmidt. Autonomous reading of gauges in unstructured environments. *Sensors*, 22(17), 2022.
- [17] Maxime Oquab, Timothée Darret, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.
- [18] Maurits Reitsma, Julian Keller, and Kenneth Blomqvist. Analog Gauge Reader, 2023.
- [19] Charig Yang, Weidi Xie, and Andrew Zisserman. It's about time: Analog clock reading in the wild, 2022.
- [20] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. pages 9699–9708, 2020.