# Analog Meter Reader

by

**Keeran Dhakal**
Department of Computer Science
College of Information Science and Technology
University of Nebraska at Omaha
Omaha, NE 68182-0500
402-415-3459
kdhakal@unomaha.edu

For

**CSCI 8300, Image Processing and Computer Vision**

Fall, 2023

Date: Thursday, December 7, 2023

# Abstract

Analog meters are still very prevalent in industries despite the fact that they are continuously being replaced by digital meters. This report discusses the automation of the reading of analog meters so that minimum human intervention is required in tasks involving such meters. This report talks about the use of object detection models and their fine-tuning for the localization of objects as well as their results and analysis. It also discusses image processing techniques that can be applied to the image for accurate reading of the analog meters. This project also explores the robustness of the reading made by the system taking into account the orientation of the object in the image.

# Table of Contents

# 1. Introduction

## 1.1. Problem

In this day and age, the use of technology has automated everything. From human-like chatbots to self-driving cars, we have come a long way in this field of automation. One similar task this project talks about is working on a system with automated readings of analog meters and gauges. The system first receives the images as input, processes the image, performs various operations, and outputs the reading based on the operations.

## 1.2. Motivations

The development and popularity of digital meters have suppressed the use of analog meters in various places. Despite this, there are still analog meters that are being utilized in various industries for measurement, the reason being that it is easy to intuitively interpret the results with the analog meter. With the increasing use of technology and automation, it only makes sense that the reading of analog meters is automated. Hence, the use of image processing and computer vision is employed to automate the reading process of analog meters.

## 1.3. Significance

The use of automated reading in industries can automate the whole pipeline of any industry using the feedback loop system. The human interference necessary to avoid any undesirable scenarios due to anomalies will be dramatically reduced. Hence, the automatic reading of the analog meters is one of the interesting topics to work on.

There are a few experiments that have already been done on this. Since the reading of meters in large industries is a critical matter, the reading has to be accurate for the system to be employed. A lot of experiments on this have focused on needle detection and number recognition in the meter. One other thing to incorporate would be the orientation of the meter in the image, which would make it look like the needle is pointing differently when this is not the case. One can fix this issue with the fixed positions of the meter and the camera or by using the relative perspective of the image. I have used the second idea to account for the orientation change.

## 1.4. Challenges

Experiments already done are more focused on the images taken from a static position. The path I will be taking is to make the system robust for the change of camera positions as well. Since the orientation of the image is to be studied based on the subjects available in the image, there will be a significant amount of image processing involved, and hence the image quality of the meter is one of the things that matters a lot. So, getting good images to work with will be one of the requirements as well as a challenge. Other than that, locating the meter is also another important part of the project before further processing; hence, object detection and meter recognition will be another important challenge.

## 1.5. Objectives

### 1.5.1. Project Objectives

- To accurately read the analog meter from the images.
- To detect and identify the position of the meter in the image.
- To incorporate the orientation of the image to reduce the error.

### 1.5.2. Academic Objectives

- To be familiar with image processing techniques and computer vision algorithms.
- To learn about the use of different tools and technologies in the field of AI and computer vision.

# 2. Overview

## 2.1. History of the Problem

The one problem that has been practiced in all fields with the advancements in technology is automation. Due to the minimization of human interference, this has been a popular problem that has been practiced frequently. Similarly, the work on analog meter reading has also seen a lot of automation work. Chen and Wang [1] has worked on the analog meter reading for meters such as multimeters in which one has to incorporate the range that it is set on to measure. Similarly, Peixoto et al. [5] has implemented the analog meter reading with the use of CNN as an object detector to detect the analog meter in the image. Sowah et al. [7] has experimented with this by combining various learning algorithms for the pointer identification methods. This method also employs the number recognition in the meter to do the reading based on where the tip is at. Salomon et al. [6] has also performed similar experiment using computer vision algorithms for automated meter reading for energy meters in unconstrained scenarios. Howells et al. [2] has also performed the analog gauge reading in real-time on mobile phones using CNN networks. Lauridsen et al. [4] has also presented their work on reading circular analogue gauges using digital image processing based pipeline.

## 2.2. State of the Art

The problem has been worked on in various ways. The solutions on the automatic analog meter reading have produced accurate readings as well. This problem generally uses various algorithms and techniques in combination such as object detection and inclination detection. State-of-the-art object detection models such as YOLO, SSD, RCNN, etc. are usually employed for object detection in these tasks. The pre-trained object detection models are trained using a custom small dataset to get accurate results on the relevant data. Inclination detection is normally done using classical image processing techniques.

# 3. Techniques

## 3.1. Principles, Concepts, and Theoretical Foundations of the research problem

This problem mainly considers image processing and computer vision as the foundation of all the tasks that will be accomplished. Generally, there are three steps to the task. The first step is object detection and segmentation which is very crucial to precisely detect the analog meter. Object detection and segmentation are done using neural network models such as YOLO, SSD, and RCNN models, the YOLO model in this case. Compensating for the wrong orientation of the meter in images will also be attempted here.

A generic formula to calculate the cross-entropy loss function for CNN models is:

$$L = -\sum_{c=1}^{m} (y_c * log\hat{y_c})$$

where $L$ is the cross-entropy loss, m is the number of classes, $y_c$ is the ground truth, and $\hat{y_c}$ is the predicted value.

The second step is finding the exact position of the needle. This is one of the most important parts of this project. Different experiments performed have different approaches to determining the position of the needle. Image processing using hough transform, keypoint detections on needle head, and gauge center are some ways of locating the needle.

The final step is interpreting the result. The approaches that have been used in the previous step for determining the needle position also determine how the result is going to be interpreted. If one uses angle calculation to the horizontal line for position detection, then the actual number would be obtained from the angle calculation and fluctuation per degree of angle.

## 3.2. Relevant Techniques that would be useful

### 3.2.1. Object Detection

Object detection is a computer vision technique for locating an object in an image or a video. This technique uses machine learning or deep learning methods to localize the object and output the bounding box of the objects present in the image along with their classifications.

### 3.2.2. Hough Transform

Hough transform is an image processing technique that is used to identify various shapes in images. The method was initially used to detect lines in an image; however, was later modified to detect complex shapes such as circles, ellipses, etc.

### 3.2.3. Keypoint Detection

Keypoint detection is the detection of an object or entity by detecting a few major key points in it. Several features or key points that are sufficient to describe the object are the minimum number of points that are to be determined. To detect a needle, one has to detect its head and its tail.

### 3.2.4. Angle Identification

Angle identification is the process of determining the inclination of a meter pointer or needle. The angle of the needle is calculated using a reference line, such as a horizontal plane. This technique is mostly used for the interpretation of results.

## 3.3. Algorithms, Process modules, and solution methods

The first task that is done in this project is localizing the analog meter object. That is done through the use of the object detection method. For the object detection method, a pre-trained object detection model is fine-tuned and used. The pre-trained model used is YOLOv7. Wang et al. [8] has presented version 7 of the YOLO model. The model is trained on the COCO dataset.

The YOLO algorithm is based on four approaches:

### 3.3.1. Residual Blocks

This step comprises diving an input image into N*N grids of equal shape. Each grid cell is responsible for localizing objects and determining the class of objects that are being covered in the cell along with their confidence score.

### 3.3.2. Bounding Box Regression

The rectangular bounding boxes are then identified based on the number of objects in the image. YOLO determines these bounding boxes by outputting their attributes in the following format.

$$Y = [p_c,\ b_x,\ b_y,\ b_h,\ b_w,\ c]$$

where $p_c$ is the probability of the object present inside the cell, $b_x, b_y$ is the x and y coordinates of the center of the bounding box, $b_h, b_w$ is the height and width of the bounding box and $c$ is the corresponding class of the bounding box.

### 3.3.3. IOU

IOU stands for Intersection over Union. A lot of times, a single object can have a lot of bounding boxes in which not a lot of them are relevant. IOU plays a role in discarding the boxes that are not relevant.

$$IOU = \frac{Intersection\,Area}{Union\,Area}$$

The grids with IOU less than the specified threshold are discarded.

### 3.3.4. Non Max Suppression

Non-maximum suppression (NMS) is a technique used to remove the redundant bounding boxes and to select only the most confident bounding box that has high overlap.

# 4. Approaches

## 4.1. Methodologies I applied in this research

### 4.1.1. Understanding the work in the domain

Reading various research papers was done to ensure the validity of the pathway that was taken during the project. This process also helped in identifying any appropriate solutions that have already been practiced.

### 4.1.2. Data Collection

Relevant datasets that were required in some stages of this project have been collected based on some of the papers that I read. For fine-tuning purposes, a small dataset of 28 train images and 8 validation images was collected and used for training.

### 4.1.3. Object Detection

Various algorithms and techniques for object detection were studied. A popular object detection model, YOLOv7, was selected based on its performance and ease of use and fine-tuned using the collected dataset to get accurate detection of analog meter objects. The fine-tuned model worked well, but the original model would detect the meters better while classifying it as a clock. Since analog meters were similar in appearance to clocks, I assumed this would happen. Because I only required region detection and not the classification part of the model, I went with the original model, not the fine-tuned one, as its performance was better. The lower performance of the fine-tuned model might be because we only had 28 training images and 8 validation images.

### 4.1.4. Pointer Detection

Certain image processing techniques for line detection, such as the Hough transform and Line Segment Detector (LSD), were evaluated to see what worked best in this case. Hough transform returned a whole line after detecting the lines in the image, whereas LSD returned a combination of lines to identify a single line for pointer position identification. More lines are susceptible to noise and are difficult to remove, so the Hough transform method was selected to work on the pointer detection task.

### 4.1.5. Readings Calculation

Once the pointer detection is done, the position of the pointer is used for calculating the slope of the pointer, which was adjusted to account for the coordinate progression difference in the actual coordinate plane and the image coordinate plane. The actual coordinate plane has Y-axis values increasing from bottom to top, while the image coordinate plane has Y-axis values increasing from top to bottom. So, the slope would be somewhat inverted. The relative difference in the coordinate plane was taken into account to calculate the angle of the image from the slope. There is yet another caveat: The slope would only say how the line is oriented; where the pointer is pointing is another thing that you have to deduce from the line coordinates before calculating the angle the pointer makes with the horizontal plane. The angle would then be appropriately converted based on where the pointer lies before calculating the reading of the meter.

## 4.2. Techniques I used to solve the problem

Most of the techniques that are used in this project are not novel approaches. As this project is a combination of two or more project components, it focuses on the robustness of the approaches that were used. The image processing techniques were studied and used, which is to be expected at this level. But the new thing that has been tried is to accompany the change in orientation of the meters in the images. Giving the program the relative readings in horizontal planes allowed for this to be accomplished. The relative readings in the horizontal plane are values at the angles of $0^\circ$ and $180^\circ$.

The procedures that were conducted throughout the research can be explained in the following way:

- Detection of meter region in the image with the use of YOLOv7 object detection technique.
- Cropping out the region of the region detected from the original image.
- Pointer needle identification using hough transformation for detection of straight lines.
- Meter center point detection for the meter.

- Calculate the slope of the detected lines and, hence, the angle of the pointer to the horizontal plane.
- Calculate the reading based on the angle identified.
- Testing the set of images by comparing their outputs with the actual reading and calculating the root mean squared error.

### 4.3. Processes I engaged in this research

Programs are written in Python programming language, along with the use of frameworks such as PyTorch as well as other available packages such as OpenCV, NumPy, etc. The experiments were divided into sub-projects and worked on independently. These tasks were performed gradually and, hence, stacked one sub-project on top of another until the project was completed.

The results are demonstrated in the later section. The project is an end-to-end pipeline where an input image is given to the system, the appropriate reading analysis is done, and the final reading is based on the analysis presented as an output. Outputs from various stages are also presented in the later section for the analysis of each stage after the image is passed through.

The main criterion to judge the result was to compare it with the actual reading of the meter. This was done by getting a set of images containing analog meters and their corresponding readings and labels in a file. The pipeline that was employed in the project outputs the reading when an image is passed through the program. The output was then compared with the actual reading in the label file and a root mean squared error was calculated based on all the observations from the set of images.

### 4.4. Facilities and supplies used for this research

- MacBook Air 2020 Laptop
- MacOS Ventura 13.6
- Google Colaboratory Platform
- Python Programming Language
- Deep Learning Frameworks such as pytorch and tensorflow
- python packages such as numpy, opencv, etc.

# 5. Work Conducted

## 5.1. Tasks performed in this research

- **Understanding the work in the domain**

  This task was done in the initial phase of the project for one to two weeks. Research papers related to this type of project was collected and studied. A basic graph of what needs to be done was prepared after this step.

- **Dataset Collection**

  This task was done overlapping with the understanding of the work tasks. While the above task was being conducted, I happened to know about a Kaggle video dataset of analog pressure meters. The images in the collected data were derived from the frames of the videos I found in the Kaggle dataset prepared by Julius Grassmé [3].

- **Object Detection**

  Various models, both pre-trained and fine-tuned, were tested for this task. This was worked on immediately after the preliminary tasks. YOLOv7 was selected to perform this task of object detection because of its availability, higher level of performance, and ease of use for training and inference. This task was worked on for around four weeks before proceeding to move ahead with the next step.

- **Pointer Detection Task**

  This task incorporating various image processing algorithms, was worked on in the later half of the project. Various algorithms, such as Hough transform and Line Segment Detector (LSD), were tested to see what would be a better fit for the project. Finally, the Hough Transform was chosen because it would detect the lines as a whole, not in a piecewise linear form. This task was worked on for around 3 weeks.

- **Readings Calculation**

  The reading calculation and interpretation tasks were done in the final weeks of the project duration once the object detection and pointer detection tasks were accomplished. This was done using the slope and angle approach since this approach would also take into account the orientation of the meter in the image and was computationally inexpensive as well. This was worked on for around 2 weeks before moving on to the documentation of the project.

- **Documentation**

  The approaches that were taken, the results that were achieved, and the analysis that was performed were all documented in the final project report that is to be submitted. This was worked on for a week.

### 5.2. Schedule, timeline, and milestones

| Order | Dates | Tasks/activity | Prerequisites (Knowledge, Skill or Tools) | Results Obtained |
|---|---|---|---|---|
| 1 | From Sep 7 to Sep 21 | Understanding other works | - | Proper understanding of approaches |
| 2 | From Sep 7 to Sep 21 | Dataset Collection | - | Dataset Preparation |
| 3 | From Sep 21 to Oct 21 | Object Detection Task | Python, PyTorch, opencv, numpy | Object Detection and Region Identification |
| 4 | From Oct 21 to Nov 15 | Pointer Detection | Python, PyTorch, opencv, numpy | Pointer Inclination Detection with slope and angle |
| 5 | Nov 15 to Dec 1 | Readings Calculations | Object Detection and Pointer Detection | Meter reading output along with results and analysis of project |
| 6 | Dec 1 to Dec 7 | Documentation | LaTeX | Documentation of the project |

# 6. Results and Analysis

## 6.1. Results

### 6.1.1. Object Detection

The YOLOv7 model was fine-tuned with a custom analog meter dataset collected from frames of available videos posted by Julius Grassmé [3]. The dataset comprises 28 training images and 8 validation images. The configuration file for training was customized to include only one class, "meter." The training was done multiple times with varying configurations to ensure that the best model was found. The training was conducted in the Google Colaboratory using a T4 GPU. The training time took around half an hour for a batch size of 2 and epochs of 100.

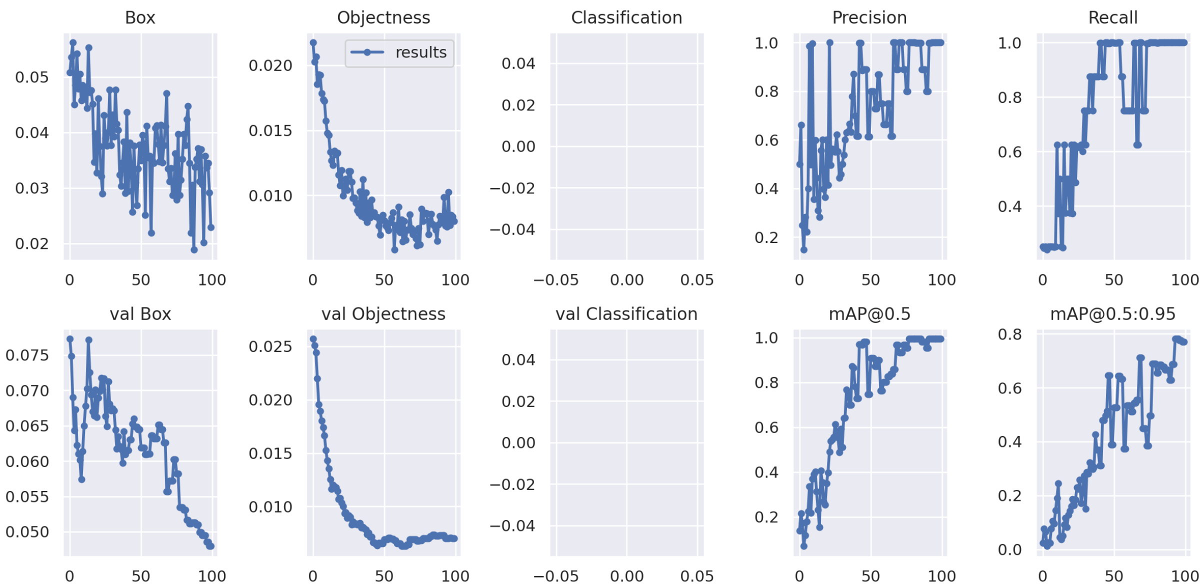



Figure 6.1: Training metrics for 100 epochs and batch size of 2

```
  Epoch   gpu_mem       box       obj       cls     total    labels  img_size
  97/99      2.3G   0.03446  0.008482         0   0.04294         6       640: 100% 14/14 [00:11<00:00,  1.27it/s]
             Class      Images      Labels          P          R     mAP@.5  mAP@.5:.95: 100% 2/2 [00:00<00:00,  6.46it/s]
               all           8           8          1          1      0.995       0.776

  Epoch   gpu_mem       box       obj       cls     total    labels  img_size
  98/99      2.3G   0.02918  0.008379         0   0.03756         8       640: 100% 14/14 [00:11<00:00,  1.19it/s]
             Class      Images      Labels          P          R     mAP@.5  mAP@.5:.95: 100% 2/2 [00:00<00:00,  7.83it/s]
               all           8           8          1          1      0.995        0.77

  Epoch   gpu_mem       box       obj       cls     total    labels  img_size
  99/99      2.3G   0.02286  0.008018         0   0.03088        12       640: 100% 14/14 [00:11<00:00,  1.21it/s]
             Class      Images      Labels          P          R     mAP@.5  mAP@.5:.95: 100% 2/2 [00:00<00:00,  3.12it/s]
               all           8           8          1          1      0.995        0.77
```

Figure 6.2: Training metrics for last few epochs

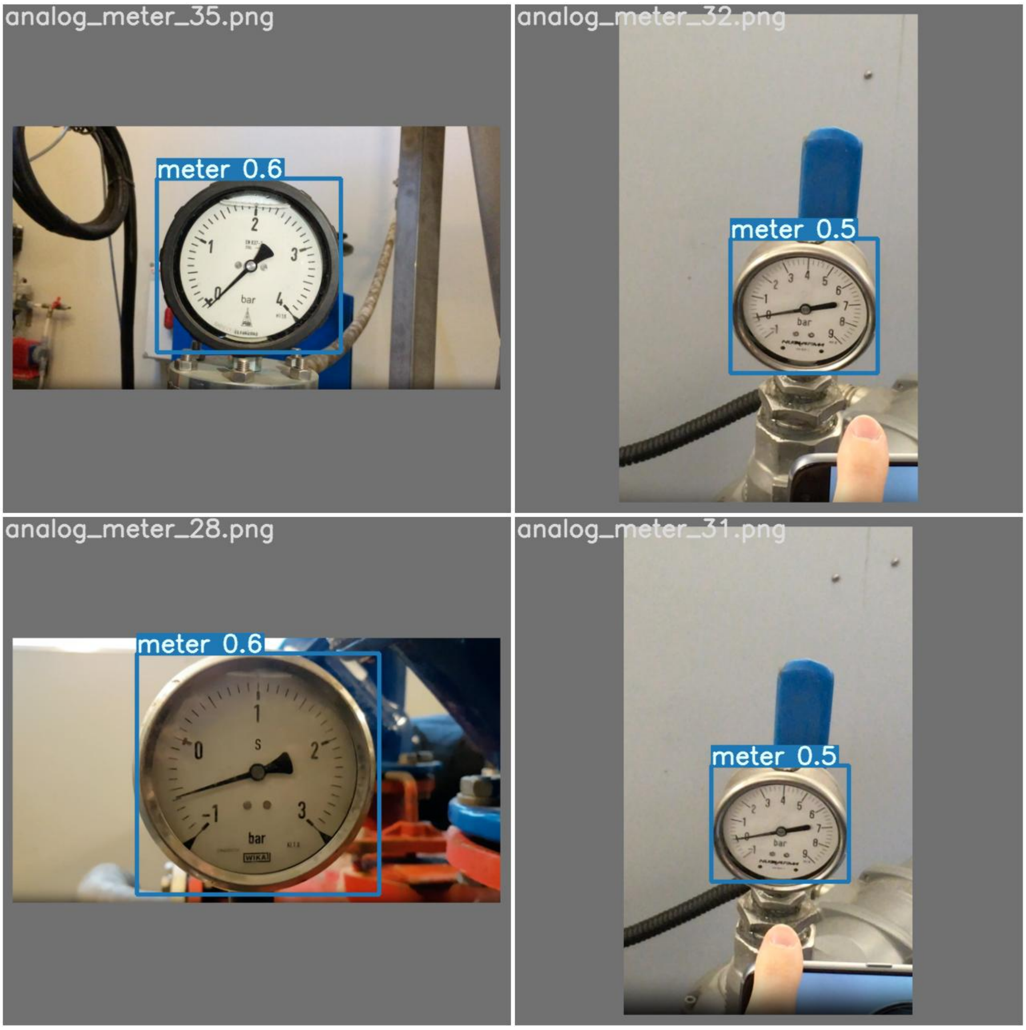
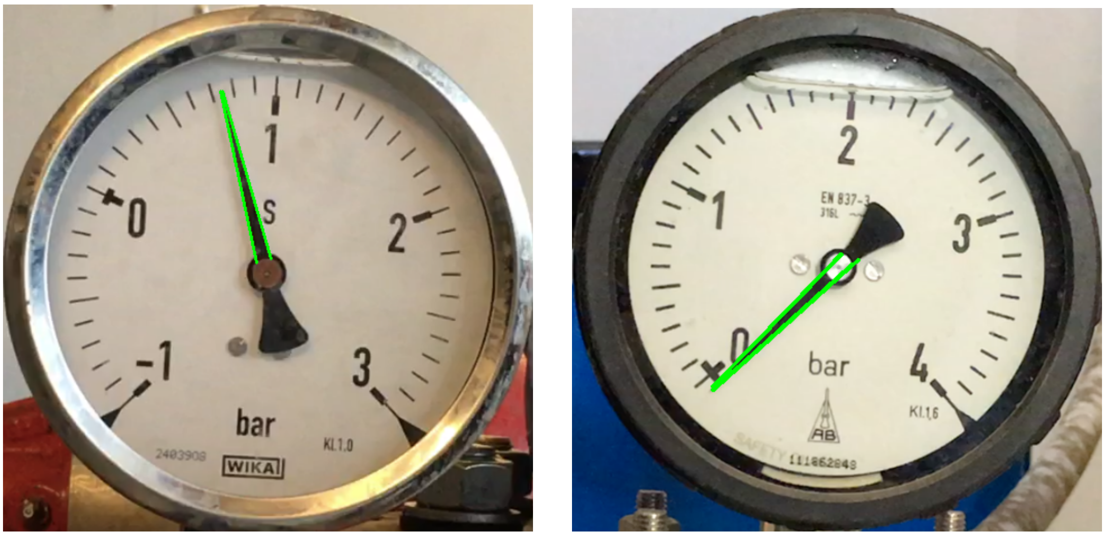



Figure 6.3: Model Prediction on validation images

Since there were only a few training images to fine-tune the model, the fine-tuned model performed similarly or inferiorly in terms of detecting the meter and its region compared to the pre-trained model. Hence, the original model was used rather than the fine-tuned one. The region identified after using the object detection method is used to crop the image and proceed with the next step.

### 6.1.2. Pointer Identification

The cropped image after object detection was first converted to a gray image and smoothed using the Gaussian blur technique. The resulting image was then subjected to the Canny edge detection method to detect the edges in the cropped image since a line detection method such as the Hough transform works better with the edge-detected image. Finally, the pointer identification was done using the Hough Transformation method, and the resulting lines obtained were checked for any outlier lines. The outlier lines are then filtered out based on the distance from the center, the slope of the line, and the endpoints of the line.




(a) Meter 1 (b) Meter 2

Figure 6.4: Lines detection using Hough Transform

### 6.1.3. Calculating Readings

After the line detection for pointers, the slopes of the lines were calculated. The following formula was employed to calculate the slopes of the lines.

$$Slope(s) = \frac{y_2 - y_1}{x_2 - x_1}$$

The calculated angle can't be directly used because of the difference in the coordinate system. Hence, certain changes are made to incorporate the difference. The slope would only mean how the line was inclined. There was still the ambiguity of whether the pointer would point upwards or downwards. This was resolved by using the line's ends and observing where it landed: the upper or lower half. That is how the region it points to was detected.

The number of slopes calculated would be the same as the number of lines that were detected. The definition of the slope of a line states that it is the tangent of the angle the line makes with the horizontal plane. Using the same approach, the angle that the detected lines make with the horizontal plane was calculated.

$$Angle(\theta) = tan^{-1}(Slope)$$

The angles formed by the lines were averaged, and an approximate angle was calculated to be used in calculations. Based on where the pointer is pointing, we would make corresponding changes to the angle. If the pointer was pointing to the upper half, the approximate angle would be the same. But if the pointer was pointing to the lower half, the approximate angle was updated based on its value.

$$Angle(\theta) = approximate\_angle + 180^\circ \qquad if\ approximate\_angle <= 90^\circ$$

$$Angle(\theta) = approximate\_angle - 180^\circ \qquad if\ approximate\_angle > 90^\circ$$

Finally, the reading is then calculated using the angle calculated above using the following formula:

$$Value = V_0 - \theta * (V_0 - V_{180})/180$$

where Value is the final value of the meter reading, $V_0$ is the meter reading at 0°, $V_{180}$ is the meter reading at 180°, and $\theta$ is the updated angle.

The values of the meter reading at 0° and 180° make the meter reading result robust in cases of misoriented meters in images.


(a) Meter 1

(b) Meter 2

Figure 6.5: Calculated angles and readings of the analog meters

## 6.2. Analysis

### 6.2.1. Object Detection

The prediction from the model in test images is compared with the predictions of the original model on the same images. The original model localizes the meter object on the image with high confidence while mapping it to another class, ”clock.” The clock is one of the 80 classes of objects that the original model was trained on. Consequently, the fine-tuned model also classifies a wall clock as a meter. The similarity between the arrangements of numbers and pointer needles, along with the circular shape of both clocks and meters, is the main reason for this. The class for the original model was renamed from ”clock” to ”meter” in the inference process for the comparison.

From the images, we can see that the fine-tuned model predicts the bounding box of meter objects with a low confidence score, whereas the original model predicts the bounding box with high scores. Similarly, the fine-tuned model sometimes predicts two bounding boxes for an image with a single object, while this is not the case with the original model. The original model is still more reliable since it is trained on more clock images, which have a similar appearance as the analog meters.

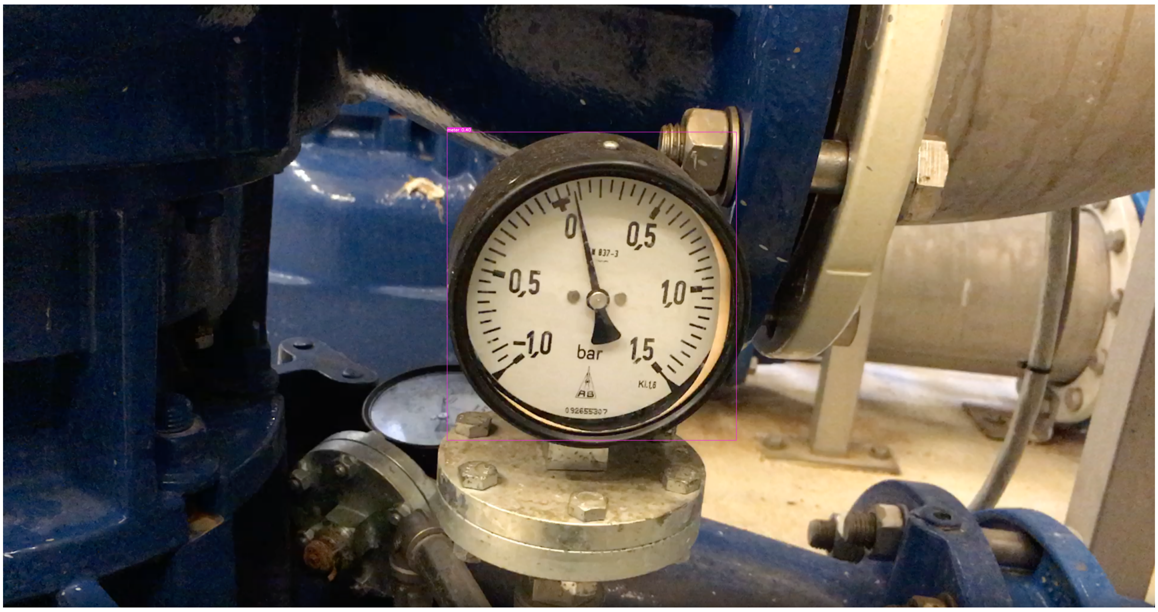



Figure 6.6: Fine-tuned model prediction on test image 1


Figure 6.7: Fine-tuned model prediction on test image 2

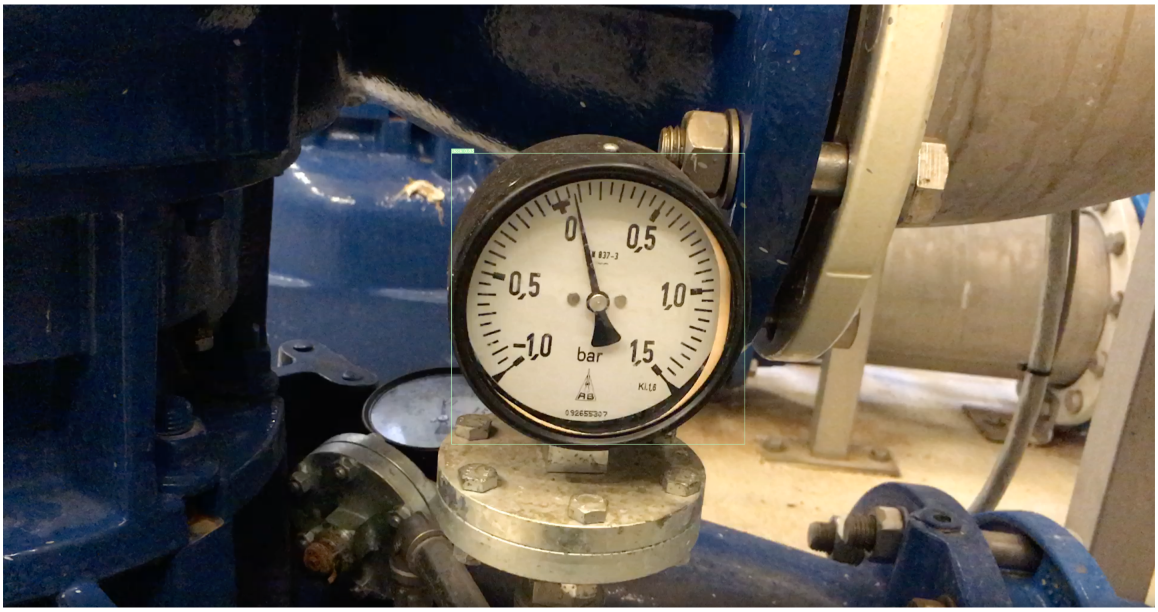



Figure 6.8: Original model prediction on test image 1

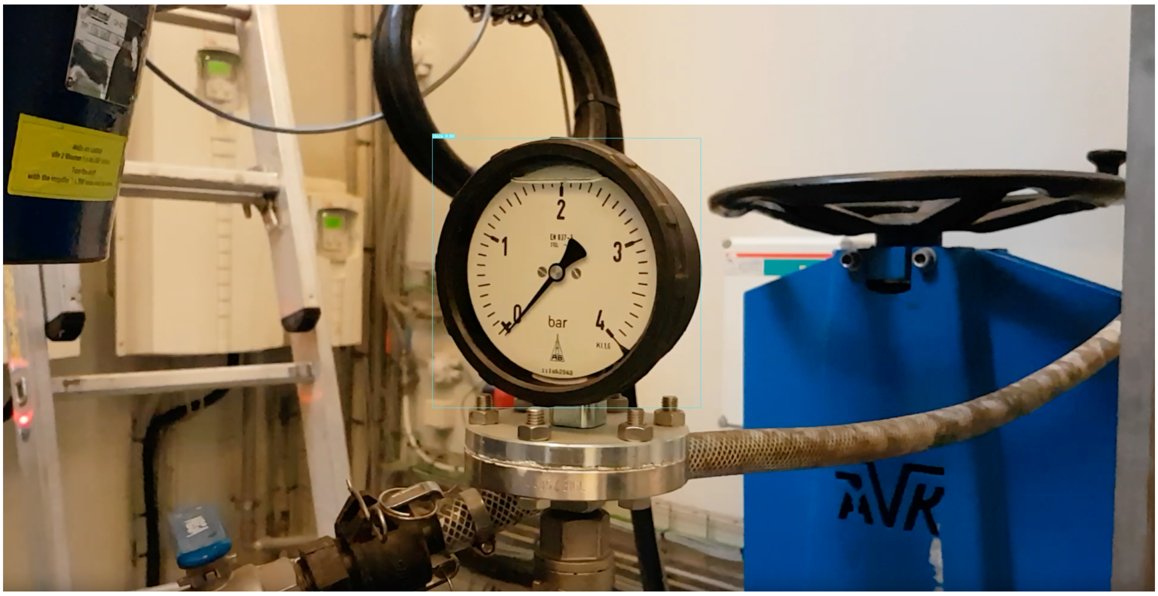
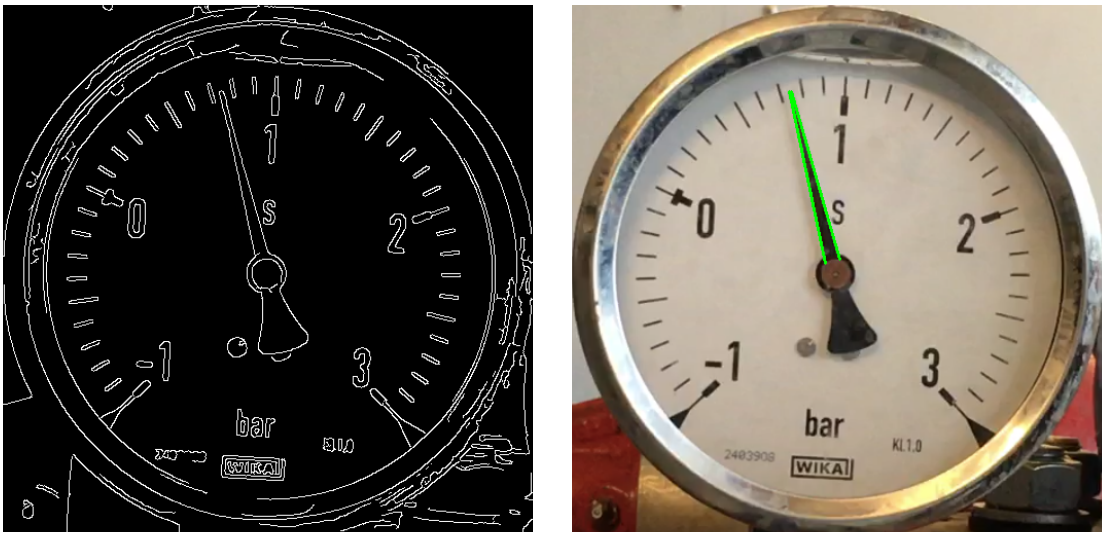



Figure 6.9: Original model prediction on test image 2

### 6.2.2. Pointer Identification

The pointer identification stage consisted of various steps, such as gray image conversion, gaussian smoothing, and canny edge detection, before applying the Hough transform to the resulting image. Hough transform works efficiently with edge images. A few outlier lines were also detected, but they were handled using the post-processing code using line end coordinates, the slope of the line, and how close it is to the center of the line. The post-processing operation left us with the only lines that we needed.




(a) Edge Image (b) Hough Transformed Image

Figure 6.10: Edge image and hough transformed image for meter 1

A different approach using the Line Segment Detector (LSD) was also tried for pointer needle line detection. The resulting lines extracted from LSD were piecewise linear in most cases, which complicates the operation of pointer identification along with slope and angle calculation. The LSD algorithm also seemed to detect the circular region in a piecewise linear fashion, which only increased the outlier lines in our case.


Figure 6.11: Line Detection using Line Segment Detector

### 6.2.3. Calculating Readings

The pointer line identification would only make sense if the center of the meter was identified. Hence, the Hough transform for the detection of circles was used to identify the circles and the center of the circle. A few issues with this approach were that a little noise in the image was able to skew the results, and unnecessary circles would be generated. Similarly, the orientation of the images would also play a big factor here. A slightly disoriented meter image would result in inaccurate circle generation, and hence the inaccurate center. Also, the meter region identification was very good in the previous step; just taking the midpoint of the rectangular bounding box would almost approximate the center of the meter.

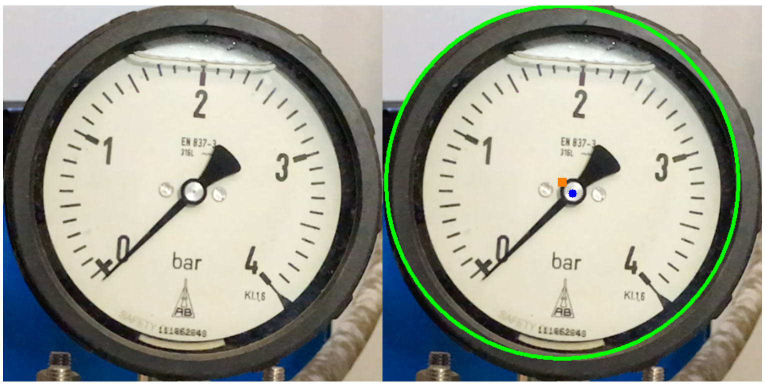



Figure 6.12: Good detection of circles using Hough Transform (Red - HT circle center, Blue - Bounding box midpoint)

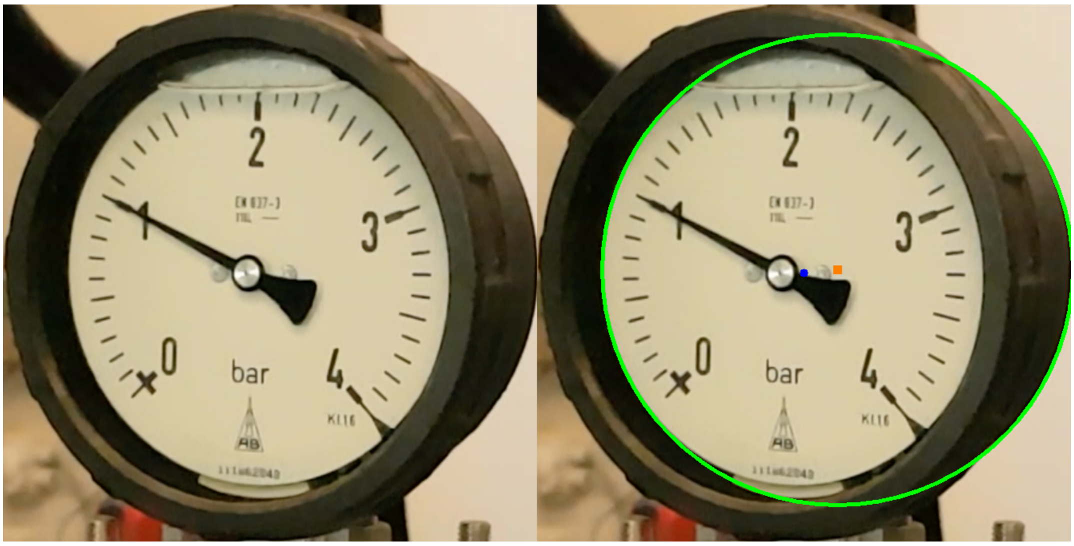



Figure 6.13: Bad detection of circles using Hough Transform (Red - HT circle center, Blue - Bounding box midpoint

After the identification of pointer lines as well as meter center point, we were able to calculate the slope of the lines and the angles that they make with the horizontal plane. The angles were updated based on where the pointer lies, and a value was calculated. This gave us the reading of the meter. The results of running the program for a few images are shown below:

| Image Name | True Reading | Read Value | Angle Identified | Deviation |
|---|---|---|---|---|
| analog_meter_0 | -0.05 | -0.0473 | 223.97 | 0.0027 |
| analog_meter_10 | 0.0 | 0.0067 | 193.06 | 0.0067 |
| analog_meter_19 | 0.0 | 0.0364 | 195.14 | 0.0364 |
| analog_meter_29 | 0.7 | 0.769 | 103.96 | 0.0694 |

Table 6.1: Few instances of results

#### 6.2.4. Evaluation of the Results

A set of 30 images was collected, and the corresponding readings were labeled in a text file. These images were selected from the pool of images that we used for fine-tuning the YOLOv7 model based on the brightness and clarity of the images. Even though we are reusing the images, it would not hurt the performance since the task we are performing in pointer identification is completely different from the one that we perform in object detection. The 30 images were run through the program, and the output of the program in the form of a reading value was then compared with the actual reading value from the label text file. For the evaluation of the output of the images, an error metric, root mean squared error

(RMSE), was calculated based on all the observations from all 30 images. The formula for the calculation of RMSE is as follows:

$$RMSE = \sqrt{\sum_{n=1}^{N} \frac{(Predicted)_i - (Actual)_i}{N}}$$

where predicted is the read value from the program output, actual is the true value of the reading, and $N$ is the number of images run through the program.

The RMSE value for the 30 images that we ran through the program using an error generation script was 0.0698. The RMSE value was found to be less than 0.1, which is a very good result considering this is an analog meter reading output that always has some error factor, even in the case of manual reading of the meters. Also, the true value in the label text file only had a precision of up to 2 decimal points, which is not ideal either. As we can see in the images, 0.1 is the smallest jump that a pointer needle can take, as can be seen from the markings on the meter. And to say that the RMSE of around 0.07 was achieved is a very good result if we compare this error margin visually in the images as well. We selected good-quality images to avoid the problems of having too many outliers and, in the worst case, no detection of pointer needles. So, the images that are not well-lit may have more errors when running through the program. However, tolerance, to some extent, has been developed through post-processing as well.

The algorithms and techniques that we used in this project are not new, considering this is somewhat of a solved problem. The usual route of object detection and line detection techniques were used for the identification of meters and meter pointers, respectively. One thing that a lot of papers seemed to miss was the orientation of the meters in the images. Most of the work on meter reading has focused on the static camera and static meter perspectives. Hence, making this system flexible in terms of the placement of meters and cameras was a new goal that was attempted. Going this route, a few approaches were there to try: fixing orientation using the OCR and fixing orientation using the shape of the meter. The OCR approach was not possible since there was not enough text in the meter images to get a hint of orientation. A Python module named pytesseract was used to see if any texts were detected, but due to the very limited amount of text in the meters, this was not possible. Similarly, the shape approach also could not work because all the images that were taken for the project had circular shapes, which, when rotated at any angle, would have the same circular shape. So, this approach failed as well. Then there was a different route, which is not an image-processing approach, which took the values at the horizontal plane at 0° and 180°, which was thought of accomplishing with a few tweaks in the post-processing code.

The relative position of the meter readings in the disoriented figure was taken care of this way. The details of how this was done are mentioned in the results section.

## 6.3. Discussions

### 6.3.1. Problems and issues arisen during the research

This project should be able to work with videos with a few changes and optimizations as well since videos are the frames of images played sequentially in order. However, the developed program has never been tested in videos, so this can't be said for sure. Also, the hardware of the machine that this project was worked on was limited, so processing videos might have taken dramatically more time than images. So, this project mostly focused on the reading from the images. Similarly, appropriate and adequate datasets for training and fine-tuning were hard to find as well. Hence, the fine-tuned model with little data didn't perform up to the mark, due to which the fine-tuned model was replaced with the pre-trained YOLOv7 model.

### 6.3.2. Limitations and constraints of the research

As this is a proof-of-concept type of project, the resulting program is not a full-fledged application. One can run the program with the required parameters, though, and it should work as expected. The building of the full-fledged application itself might be a project worth taking time, so considering the limited amount of time in hand, this can be deemed a further enhancement.

# 7. Summary

This report highlights the tasks being done for automation in the reading of analog meters that are mostly used in industries. Even though the popularity of digital meters is increasing with each passing day, there are still a lot of analog meters being used due to the intuitiveness of their reading. The system, after completion, should be able to work on detecting meters and analyzing results based on the input image.

These systems, while in use with the feedback loop in any industry, can automate the whole pipeline of the industry and reduce the human intervention that is usually required. This will also help prevent anomaly scenarios that can cause a lot of damage and loss. The main objective behind this project is to use image processing and computer vision algorithms for automation tasks that are useful in common scenarios.

These problems are generally divided into several stages, each of which is an independent task on its own, such as object detection and localization, pointer identification, and analysis. Object detection has seen a lot of state-of-the-art models such as YOLO, RCNN, etc. The YOLOv7 model was chosen for fine-tuning tasks and trained using a custom analog meter dataset. This model, while pre-trained, was also capable of producing great results at detecting the object and their coordinates. The region identification and object detection confidence were better in the pre-trained model rather than in the fine-tuned one. For the task of pointer identification, a few classic image processing algorithms were employed to get a better estimate of the pointer object. One such component used was the Line Segment Detector (LSD), which outputted lines in the form of piecewise lines. Another method tried was the Hough transform. This is a common line detection algorithm that gives good results in line detection tasks. Although this produced line detections with some outlier lines, it was possible to clean up the outliers, so this was chosen for the line detection task.

After the object detection and pointer identification, the next task that was due was to find the value of the reading. The approach that was taken for this step was to identify the slopes from the detected lines, filter unwanted lines, and calculate the angles from the remaining lines. The approximated angle was calculated using the average of the angles. Based on where the detected line endpoints lie in the images, the angle was updated, and finally, a reading was calculated based on the updated angle value, taking into consideration the reading of the meter at horizontal planes.

Even though a lot of experiments have already tried solving problems like this, they seemed to assume that the camera is always in a fixed position and the analog meter face is perpendicular to the camera angle. In this project, we considered the scenario where these conditions were not true and worked to fix the reading even if the meter in the image was not in the correct orientation.

## References


[1] Yung-Sheng Chen and Jeng-Yau Wang. Computer vision-based approach for reading analog multimeter. *Applied Sciences*, 8(8):1268, 2018.

[2] 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.

[3] Malte Pedersen Julius Grassmé. MS Windows NT kernel description, 2018. URL `https://www.kaggle.com/datasets/juliusgrassme/pressure-gauge-reader-data`.

[4] Jakob S Lauridsen, Julius AG Graasmé, Malte Pedersen, David Getreuer Jensen, Søren Holm Andersen, and Thomas B Moeslund. Reading circular analogue gauges using digital image processing. In *14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Visigrapp 2019)*, pages 373–382. SCITEPRESS Digital Library, 2019.

[5] João Peixoto, João Sousa, Ricardo Carvalho, Gonçalo Santos, Joaquim Mendes, Ricardo Cardoso, and Ana Reis. Development of an analog gauge reading solution based on computer vision and deep learning for an iot application. In *Telecom*, volume 3, pages 564–580. MDPI, 2022.

[6] Gabriel Salomon, Rayson Laroca, and David Menotti. Image-based automatic dial meter reading in unconstrained scenarios. *Measurement*, 204:112025, 2022.

[7] Robert R Sowah, Abdul R Ofoli, Eugene Mensah-Ananoo, Godfrey A Mills, and Koudjo MM Koumadi. An intelligent instrument reader: using computer vision and machine learning to automate meter reading. *IEEE Industry Applications Magazine*, 27 (4):45–56, 2021.

[8] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.