**AUTOMATED VISUAL INSPECTION OF METER READING**

*Submitted to*

***HRD, SAIL***

*An In-Depth Analysis and Implementation Report*

*by*

PIYUSH KUMAR

***B.TECH in Information Technology***

***at***





*Guided by*

**Mr. Aditya Dubey**

*Sr. Manager, C & IT*

BOKARO STEEL PLANT (SAIL)



# DECLARATION CERTIFICATE

I certify that

1. The work contained in the report is original and has been done by myself under the general supervision of my supervisor.
2. The work has not been submitted to any other Institute for any other degree or diploma.
3. I have followed the guidelines provided by the Institute in writing the report.
4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references.
6. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

**Date:** **14/07/24 PIYUSH KUMAR**

**BTECH/10191/21**

**Department of Computer Science and Engineering**

**Birla Institute of Technology, Mesra**

***CONTENTS***

**ABSTRACT …………………………………………………………………………… i**

**ACKNOWLEDGEMENT …………………………………………………………..... ii**

**LIST OF FIGURES ……..…………………………………………………………..... iii**

**LIST OF TABLES …………………………………………………………………..... iv**

**CHAPTER 1 – INTRODUCTION ………………………………………………….. 1**

**1.1 INTRODUCTION TO YOLOV8 MODEL ………………………….………….. 1**

**1.2 OBJECTIVE ………………………………………………………..…………….. 3**

**1.3 TECHNOLOGIES USED ………………………………………………………... 3**

**CHAPTER 2 – REVIEW OF PAST WORK ………………………………………. 4**

**2.1 LITERATURE REVIEW…….…………………………………….…………….. 4-5**

**2.2 GAP ANALYSIS ………………………………………………………………….. 6**

**2.3 HOW THIS PROJECT ADDRESSES THE GAPS………………….……….…. 7**

**CHAPTER 3 – METHODOLOGY ………………………………………………….. 8**

**3.1 PROBLEM STATEMENT ………………………………………………………. 8**

**3.2 DATASET PREPARATION .………………………….………………………. 8-11**

**3.3 WORKFLOW OVERVIEW DIAGRAM……………………………………… 12**

**3.4 METER DETECTION…………………………………………………………... 13**

**3.5 IMAGE ANALYSIS…………………………………………………………….. 14-21**

**CHAPTER 4 – RESULTS & ANALYSIS……………….....................…………… 22**

**4.1 PROJECT OUTCOMES …………………………….....................…………… 22-23**

**4.2 COMPARISON WITH OBJECTIVES …………………………..…………… 23-24**

**CHAPTER 5 – CONCLUSIONS CONCLUSION & LEARNING OUTCOMES 25**

**5.1 CONCLUSION ……………………………..................................……………… 25**

**5.2 LEARNING OUTCOMES & PERSONAL EXPERIENCE………………….. 25**

**5.3 FUTURE WORKS…………………...................................……………………... 26**

**REFERENCES ………………………………………………………………………. 27**

***LIST OF FIGURES***

Figure 1.1 Visualization of an anchor box in YOLO 7

Figure 1.2 Results of the Yolov5 model 8

Figure 1.3 Results of the Yolov8 model 8

Figure 2.1 Comparison of the past work 12

Figure 3.1 Example of Meter Dataset 15

Figure 3.2 Removed (extensive blurred) 16

Figure 3.3 Removed (meter not fully available) 16

Figure 3.4 Annotated Images 17

Figure 3.5 Workflow Diagram 18

Figure 3.6 Snippet for Fine Tuning 19

Figure 3.7 Uncropped Image & Cropped Image 19

Figure 3.8-3.9 Resizing Image 20

Figure 3.10 Grayscale Image 20

Figure 3.11 Gaussian Filter is applied 21

Figure 3.12 Bilateral Filter is applied 21

Figure 3.13 Threshold Image 22

Figure 3.14 Sobel & Canny Edge Detection Comparison 22

Figure 3.15 Snippet for Contour Approximation Snippet 23

Figure 3.16 ROI Selection Snippet 23

Figure 3.17 Snippet of contourArea() and arcLength() 24

Figure 3.18 Drawing operations snippet 24

Figure 3.19 Contours drawn 24

Figure 3.20 ROI Extraction Snippet 25

Figure 3.21 Extracted ROI of meter 25

Figure 3.22 After Erosion 26

Figure 3.23 After Dilation 26

Figure 3.24 Seven Segments' states in a table using a dictionary 27

Figure 3.25 Snippet for Seven-Segment Implementation 27

Figure 4.1 Training and Validation Metrics 29

Figure 4.2 Confusion Matrix 29

Figure 4.3 Evaluation Curves 30

# *ACKNOWLEDGEMENT*

I would like to express my deepest gratitude to all the individuals and organizations who have contributed to the successful completion of this project on "Application of Automated Visual Inspection for Electric Meter Reading using Traditional Image Processing."

First and foremost, I extend my profound gratitude to my mentor at Steel Authority of India Limited, Mr. Aditya Dubey (Sr. Manager) , for his invaluable guidance, unwavering support, and constructive feedback throughout the project duration. His expertise and mentorship have been instrumental in shaping my ideas and ensuring the project's smooth progress.

I would also like to acknowledge and extend my heartfelt gratitude to my collaborators, Apurva, Abhay, and Madhur from VIT Vellore. Their contributions were instrumental in integrating the project with the Django framework and developing the end-to-end application. Their technical expertise, dedication, and teamwork greatly enhanced the scope and quality of the project.

Thank you all for your invaluable support and contributions.

This project would not have been possible without the collective efforts, support, and the encouragement and understanding from my family and friends were paramount.

Thank you for being an integral part of my journey towards achieving my goals.



Date: 14/07/2024

Piyush Kumar

BTECH/10191/21

***ABSTRACT***

This project develops an automated visual inspection system for electric meter reading using traditional image processing techniques. The system aims to enhance accuracy and efficiency in meter reading. By integrating the YOLOv8 object detection model, it accurately identifies electric meters in real-time from user-uploaded images, rejecting invalid submissions and prompting for re-upload when necessary.

Upon detection, images undergo preprocessing steps such as resizing, cropping, rotation, color conversion, and blurring (Gaussian blur, bilateral filter), followed by thresholding and edge detection (adaptive thresholding, Canny edge detection). Postprocessing includes contour approximation, drawing operations (bounding boxes, text labels), morphological transformations (dilation and erosion), and region of interest (ROI) extraction, leading to the identification and interpretation of seven-segment displays for numeric extraction.

The extracted meter readings, along with user ID and address, are stored in an Excel file for easy access and record-keeping. This approach leverages the simplicity and efficiency of traditional image processing methods, making it suitable for environments with limited computational resources. The project demonstrates the viability of classical techniques in achieving reliable meter readings and sets a foundation for future enhancements incorporating advanced machine learning models. The results show a significant improvement in the accuracy and reliability of meter reading processes, providing a scalable and efficient solution for utility management.

This project highlights the relevance of traditional techniques in modern applications and offers a comprehensive solution for automated electric meter reading, enhancing user experience and operational efficiency.

***CHAPTER 1 ~* INTRODUCTION**

**1.1 Introduction to** **Yolov8 model**

Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions .YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification and pose estimation tasks.

In conclusion, the YOLOv8 documentation serves as a comprehensive resource for users and developers interested in leveraging the capabilities of YOLOv8 for object detection tasks.

As YOLOv8 continues to evolve, the documentation will likely play a crucial role in facilitating widespread adoption and understanding of this powerful **computer vision** tool.

**Why Yolov8?**

YOLOv8 is an anchor-free model. This means it predicts directly the center of an object instead of the offset from a known [**anchor box**](https://blog.roboflow.com/what-is-an-anchor-box/).

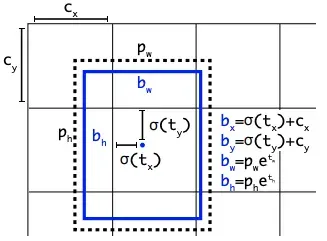
****

Fig. 1.1 Visualization of an anchor box in YOLO

YOLOv5 uses predefined boxes (anchor boxes) and predicts how much the actual object deviates from these boxes. YOLOv8 makes fewer box predictions, which simplifies and speeds up the process. YOLOv5 makes many predictions based on multiple anchor boxes, which can be more complex and slower.

Fewer predictions in YOLOv8 make Non-Maximum Suppression (NMS) step faster and more efficient. While YOLOv5 more predictions, mean this step can be slower and more complex.

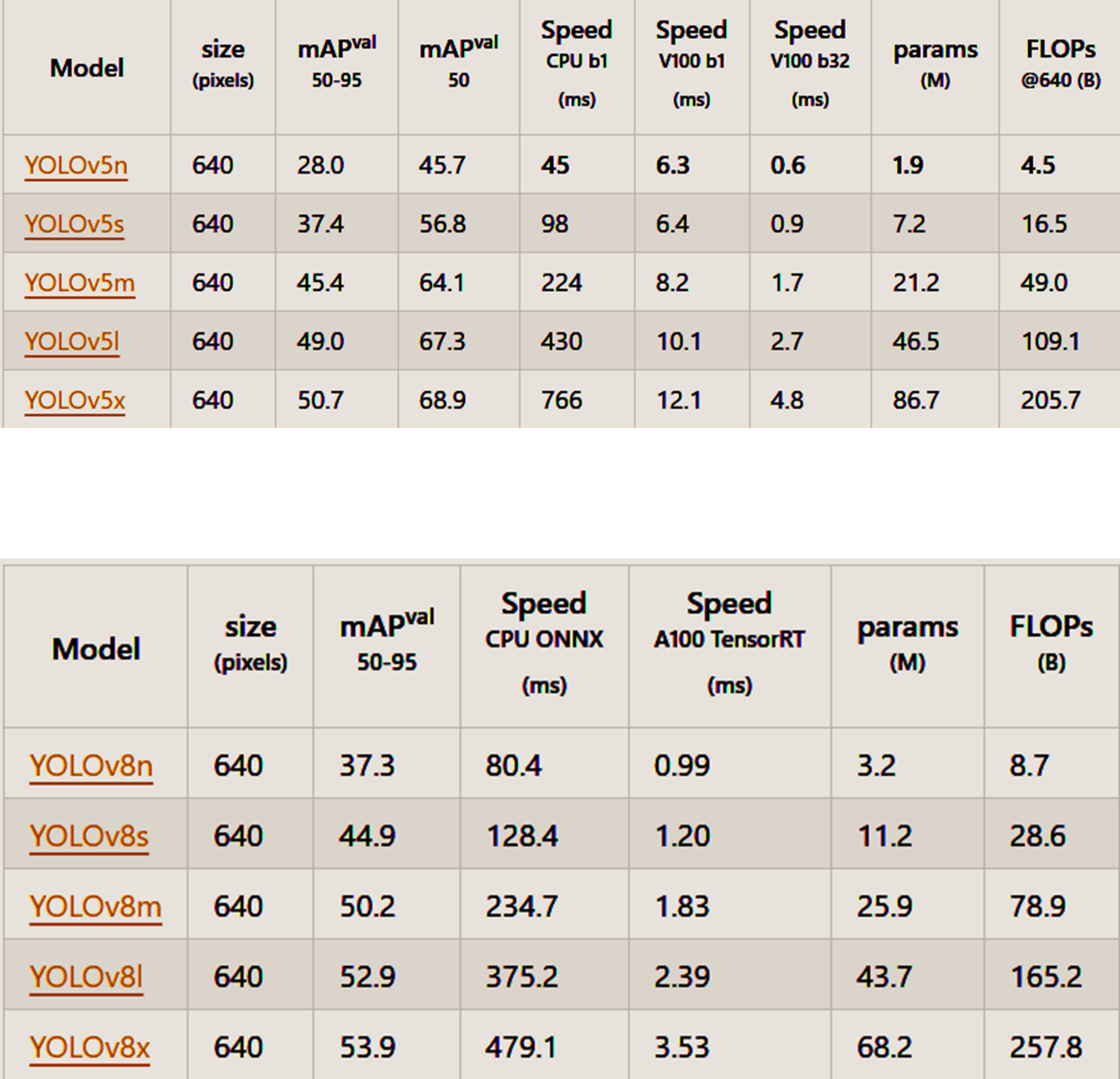
****Below are the results of these models trained on COCO dataset, which includes 80 pre-trained classes.

Fig. 1.2 Results of the Yolov5 model

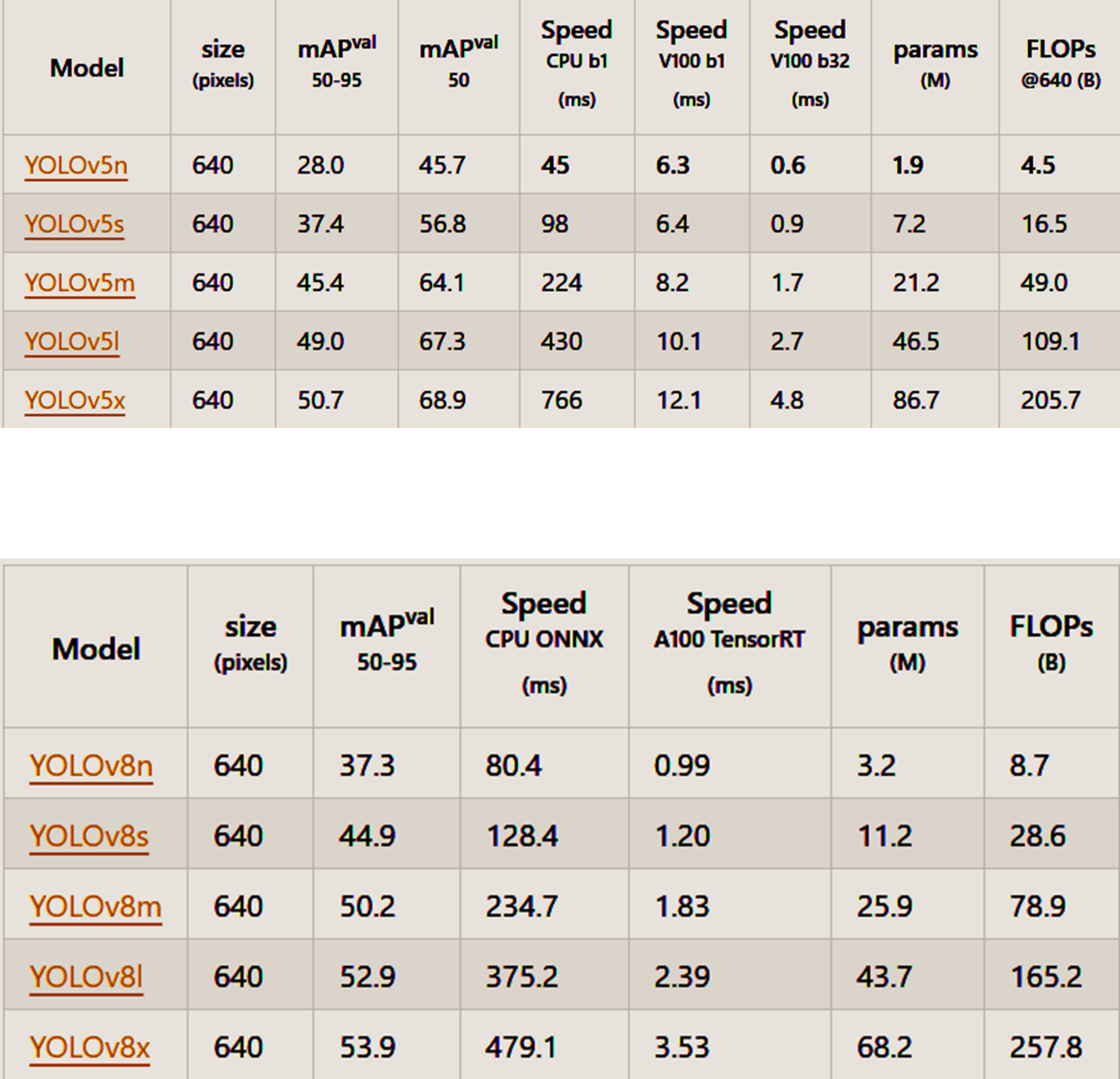


Fig. 1.3 Results of the Yolov8 model

**1.2 Objectives**

The primary objective of this project is to develop an end-to-end application for automated electric meter reading, leveraging advanced object detection and traditional image processing techniques. This project was initiated based on the existing infrastructure and resources available at SAIL, where electric meters are already installed and readings have historically been recorded manually by laborers and agents. The goal is to minimize costs and labor requirements by automating the reading process.

**Specific Objectives:**

1. **Object Detection with YOLOv8**

* Develop and implement a custom-trained YOLOv8 object detection model.
* Detect the electric meter in the uploaded image by forming a contour around it.

1. **Numeric Extraction with Traditional Image Processing**

* Utilize traditional image processing techniques to extract numeric data from the detected electric meter.
* Process the uploaded image through various methods such as resizing, color conversion, blurring, edge detection, and contour approximation to accurately identify and extract the numeric information displayed on the meter.

**1.3** **Technologies Used**

**1.3.1 YOLOv8 Model for Meter Detection:**

**Framework:** PyTorch

**Model:** YOLOv8

**1.3.2 Image Processing Techniques:**

**Libraries:** OpenCV

**1.3.3 Web Application Using Django Framework:**

**Backend Framework:** Django (Python)

**Frontend Technologies:** HTML, CSS, JavaScript

**Integration:** Django Rest Framework (for API development)

***CHAPTER 2 ~*** **Review of Past Work**

**2.1 Literature Review**

The recognition of digits has been a significant research area since the 1980s. Digit recognition using classifiers has extensive applications, from postal code recognition to automated check reading. However, this task is challenging due to the variability in digit size, thickness, and positioning relative to the image margin. These challenges necessitate robust and adaptable recognition techniques.

**Existing Techniques**

**Backpropagation Networks:**

*Y. LeCun et al. (1998***)** presented the application of backpropagation networks to digit recognition. This pioneering work demonstrated the effectiveness of neural networks in handling the complexities of digit shapes and variability.

**Convolutional Neural Networks (CNNs):**

*X. Han et al. (2019***)** provided a comprehensive summary of the latest developments in CNNs, highlighting their significant impact on image recognition technology. The paper elaborates on the application of CNNs in numeral recognition, emphasizing the improved accuracy due to their hierarchical feature extraction capabilities. However, CNNs still encounter error rates due to the inherent similarities in digit shapes (e.g., 1 vs. 7, 5 vs. 6).

*Caiyun Ma et al. (2020***)** focused on preprocessing techniques, such as normalizing images of varying sizes and stroke thickness, to eliminate irrelevant information and retain essential features. This preprocessing step is crucial for enhancing the accuracy of CNNs in digit recognition.

**Hybrid Models:**

*R. Sudhakar et al. (2021)* developed a hybrid model integrating a non-linear regression model and an optimization-driven deep learner for video super-resolution. The low-resolution frames were framed and fed into both a Fractional-Group Search Optimizer-based Deep

Belief Network (FrGSO-DBN) classifier and a non-linear regression model, showcasing the potential of hybrid approaches in improving recognition tasks.

**Euclidean Distance for Keyword Recognition:**

*V.C. Bharathi et al. (2020)* proposed a method for recognizing query words using the Euclidean distance metric. This approach involved retrieving keywords from their indexed positions in documents, effectively addressing the variability in digit writing styles. Despite its usefulness, the method struggled with digits that have high visual similarity (e.g., 3 vs. 8, 9 vs. 8).

**Deep Learning Advances:**

*Dan ClaudiuCiresan et al. (2010)* showcased the superior performance of deep and large neural networks on handwritten digit recognition tasks. Their work demonstrated that increasing the depth and complexity of neural networks significantly improves accuracy.

*Li Deng et al. (1998)* utilized the MNIST database of handwritten digits, a benchmark dataset in machine learning research, to train and evaluate various models. This database has been fundamental in advancing digit recognition technologies.

**Principal Component Analysis (PCA):**

*Vineet Singh et al. (2018)* explored the use of single-layer neural networks combined with principal component analysis (PCA) for digit recognition. PCA was employed to reduce the dimensionality of the input data, enhancing the efficiency and performance of the neural network.

**2.2 Gap Analysis**

Digit recognition faces several challenges, primarily due to the diverse ways in which individuals write the same digit. Similarities between certain digits (e.g., 1 and 7, 5 and 6) add to the complexity. The table below summarizes the analysis of digit recognition methods by various researchers.

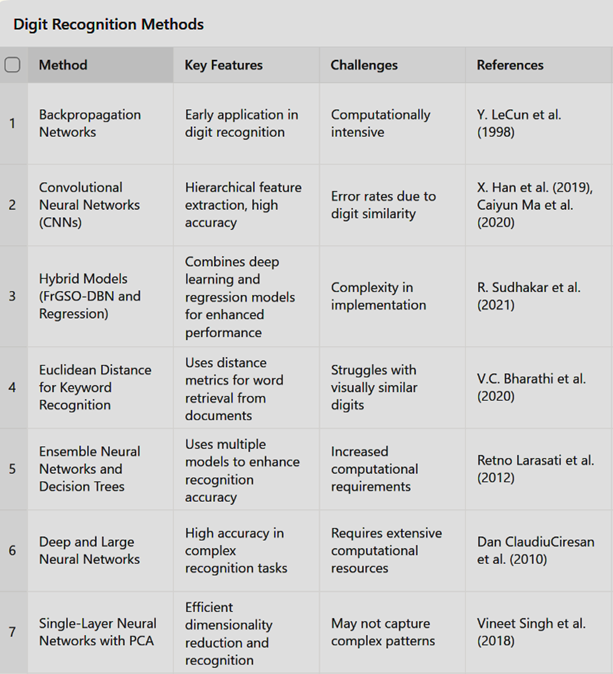


Fig. 2.1 Comparision of the past work

**2.3 How This Project Addresses the Gaps**

**Efficiency:** The project utilizes traditional image processing techniques which require less computational power, making it suitable for environments with limited resources.

**Simplicity:** By employing classical methods, the project ensures ease of implementation and debugging, providing a practical solution for real-world applications.

**Balanced Approach:** Combining YOLOv8 for initial object detection with traditional techniques for numeric extraction provides a robust yet efficient solution. This approach leverages the strengths of both methods, ensuring high accuracy without the high computational costs associated with deep learning.

***CHAPTER 3 ~*** **METHODOLOGY**

**3.1 Problem Statement**

Manual reading of electric meters is labor-intensive and prone to errors, leading to inaccurate billing and inefficient resource management. As the demand for precise and timely meter readings grows, traditional manual methods are insufficient.

Automated visual inspection technologies presents a promising solution, but current systems face challenges with real-time object detection and image processing, particularly in varied lighting conditions and complex backgrounds. This project aims to develop an advanced automated visual inspection system using YOLOv8 and traditional image processing techniques to accurately detect and read electric meters from uploaded images. By addressing the limitations of existing methods, this project seeks to enhance the accuracy, efficiency, and reliability of meter reading processes.

**3.2 Dataset Preparation**

**3.2.1 Acquisition:**

The dataset, comprising **1400 raw images**, was provided at SAIL. These images are real-world examples that can be found under typical real-life lighting conditions, capturing various electric meters in diverse environments. The images include different types of electric meters, each photographed under varying circumstances such as indoor and outdoor settings, different times of the day, and various weather conditions. This diversity ensures that the dataset is comprehensive and robust, reflecting the challenges faced in practical scenarios.

The images exhibit a range of lighting conditions from well-lit to poorly-lit environments, which is critical for training a model that can perform accurately in real-world applications. The variety in background noise, meter placements, and image qualities adds to the dataset's complexity, making it an excellent resource for developing a highly accurate and reliable object detection model. This extensive and varied dataset provides a solid foundation for creating an end-to-end application capable of performing electric meter reading with minimal manual intervention.



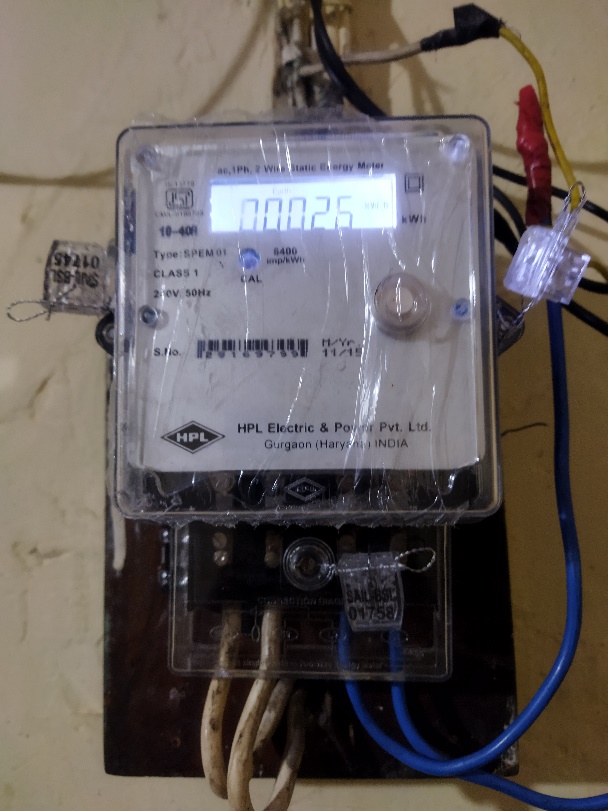
****



Fig. 3.1 Example of Meter Dataset

**3.2.2 Cleaning**

To ensure the quality and reliability of the dataset, a thorough cleaning process was undertaken. This involved carefully reviewing each of the **1400 raw images** and removing those that did not meet the necessary standards for training the model.

First, any images that were completely **distorted** and **unusable** were identified and removed.

Next, images that were **excessively** **blurred** were excluded from the dataset. Clear and sharp images are crucial for effective training and accurate model performance, so any images that did not meet this clarity criterion were discarded.

Additionally, the dataset contained some **random** **uploads** from customers that were not relevant to the project. These images did not contain electric meters or were otherwise irrelevant, so they were carefully reviewed and removed. This helped in maintaining the focus and relevance of the dataset, ensuring that only pertinent images were included.

This refinement was essential for creating a reliable and accurate labeled dataset for training the YOLOv8 model. New dataset comprised of **981 images**.

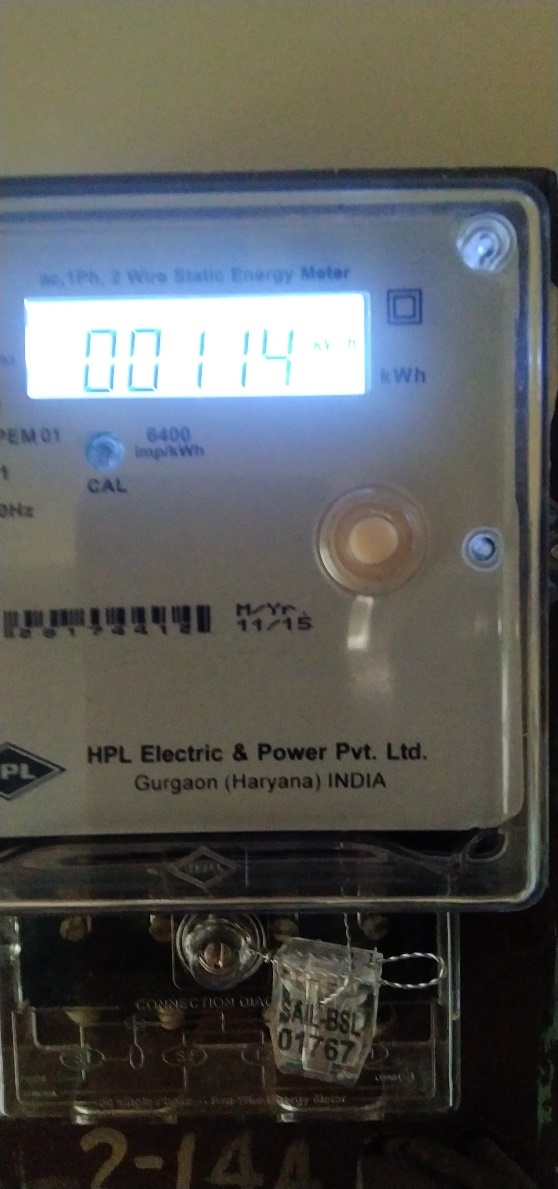


Fig. 3.2 Removed (extensive blurred)

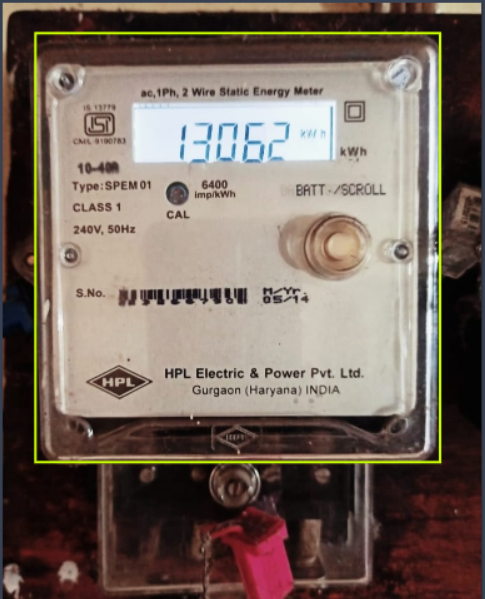
Fig. 3.3 Removed (meter not fully available)

**3.2.3 Labeling of the Images using Roboflow**

The annotation of the dataset was performed using the Roboflow platform, which offers efficient tools for labeling and managing image datasets. Initially, the cleaned dataset of 900 images was uploaded to Roboflow.

Using Roboflow's intuitive interface, precise bounding boxes were drawn around each electric meter in the images. These bounding boxes defined the regions of interest necessary for the YOLOv8 model's training. The platform allowed for easy adjustments of annotations to ensure accuracy, and it also provided automatic annotation suggestions to expedite the process.

After completing the annotations, the dataset was thoroughly reviewed for consistency and accuracy. This review ensured that the labeled dataset was of high quality, providing reliable data for training the YOLOv8 model. By leveraging Roboflow, the project achieved an efficient and effective labeling process, crucial for developing a robust object detection system.

****

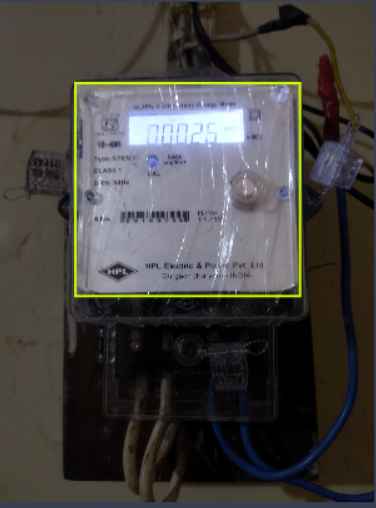
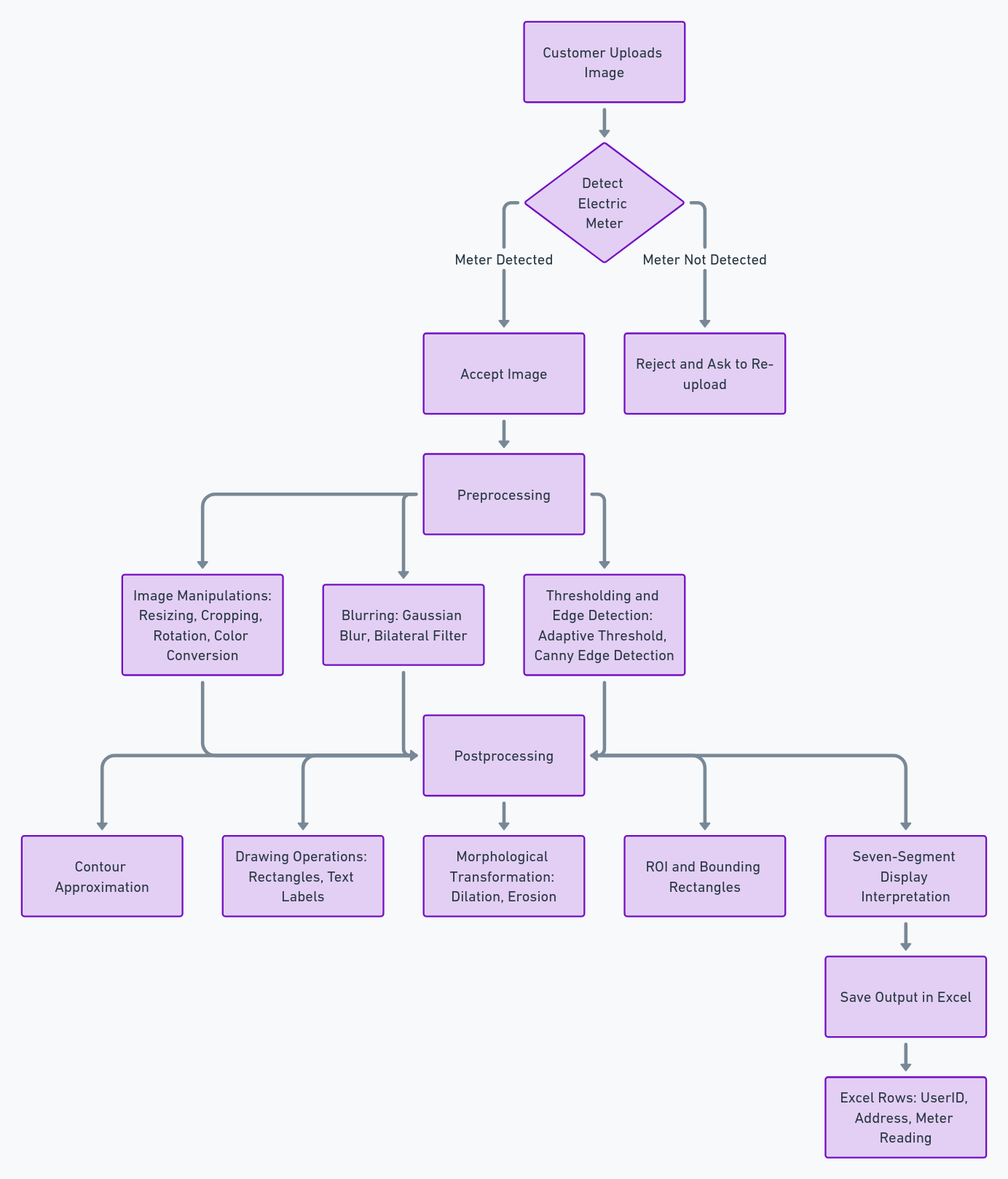
****

Fig. 3.4 Annotated Images

**3.3 Workflow Overview Diagram**



**3.4 Meter Detection**

Fig. 3.5

**Model Training of Yolov8**

To create a reliable meter detection system, the YOLOv8 model was trained on a carefully prepared dataset of annotated electric meter images. The training process included several key steps:

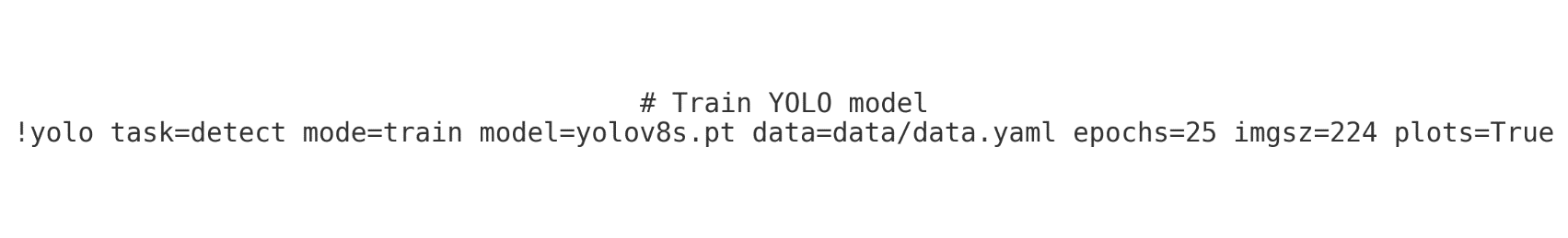
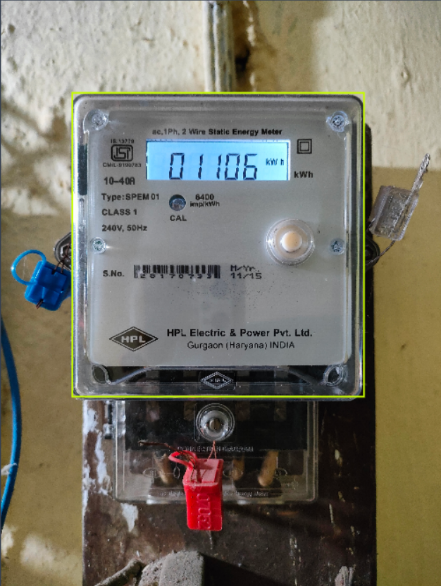
1. **Data Preparation:** A diverse collection of electric meter images was assembled, featuring various lighting conditions, angles, and backgrounds. Each image was precisely annotated to indicate the location and boundaries of the meters.
2. **Model Fine-tuning:** The pre-trained YOLOv8 model was fine-tuned using the annotated dataset. This involved adjusting the model’s hyperparameters and architecture to optimize its performance for meter detection.

Fig. 3.6

1. **Data Augmentation:**  To enhance the model’s robustness and generalization capabilities, data augmentation techniques were applied. Transformations such as rotation, scaling, and color adjustments were used on the training images, effectively increasing the dataset’s diversity.
2. **Training and Validation:** The fine-tuned YOLOv8 model was trained over multiple epochs (100 to be exact) , with its performance regularly evaluated on a validation set to monitor progress and prevent overfitting.

**Output :** The model detects the meter, draws contour and extracts the ROI for the next step.

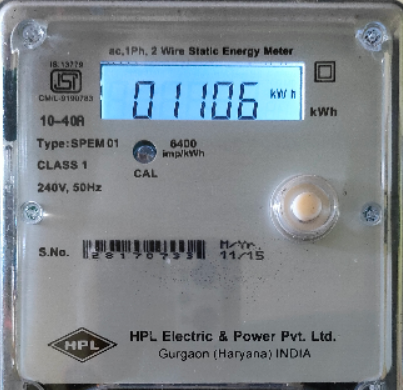
****

Fig. 3.7 Uncropped and Cropped Image

**3.5 Image Analysis**

As the image is uploaded and the meter is detected by the yolov8 model, it is passed through various pre~processing and post~processing techniques :

**3.5.1 Preprocessing of the Images :**

Preprocessing is a crucial step in the image analysis pipeline, ensuring that the images are suitably formatted and enhanced for accurate meter reading and numeric extraction. The following steps were implemented in the preprocessing phase:

***Resizing*** :

Purpose: Standardizes the image dimensions to ensure consistency and optimal input size for the detection and processing algorithms.

Implementation: Images are resized to a predefined dimension, balancing between retaining sufficient detail and reducing computational load.

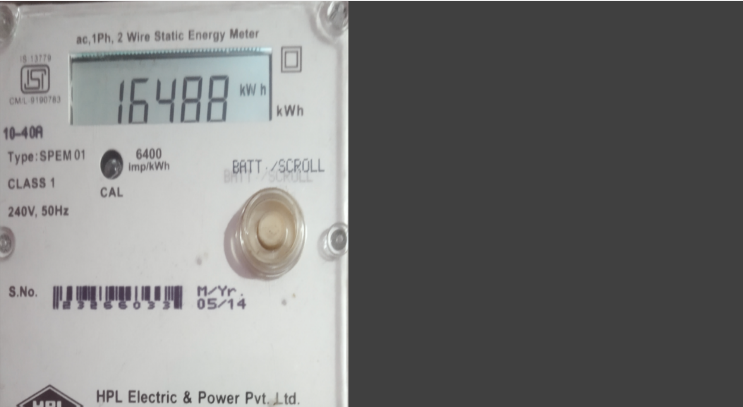
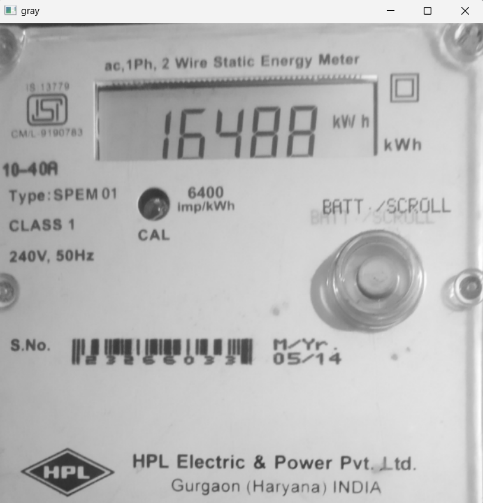


Fig. 3.8 Fig. 3.9



***Color Conversion :***

Purpose: Simplifies the image data by converting it to a more analyzable format.

Implementation: Images are converted from RGB (color) to grayscale, reducing the complexity and enhancing the contrast of the meter readings.

Fig. 3.10

***Blurring :***

Gaussian Blur:

Purpose: Reduces image noise and detail, smoothing the image to highlight significant features.

Implementation: A Gaussian filter is applied to blur the image, which helps in reducing high-frequency noise and makes edge detection more reliable.

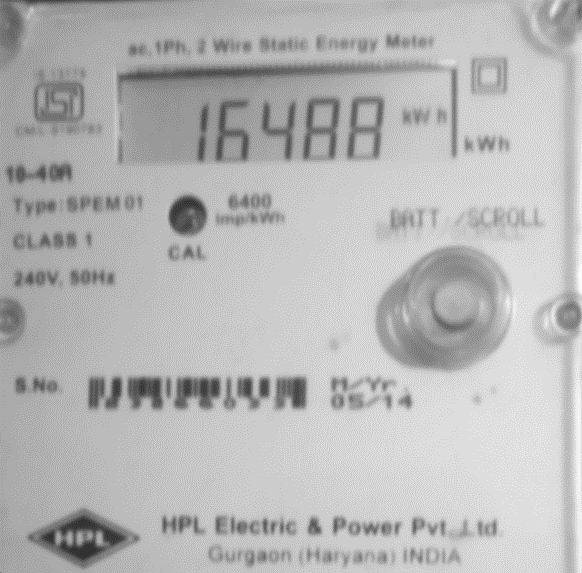


Fig. 3.11

Bilateral Filter:

Purpose: Preserves edges while reducing noise, enhancing the image quality.

Implementation: A bilateral filter is applied to maintain sharp edges while smoothing other parts of the image, crucial for accurate edge detection and contouring.

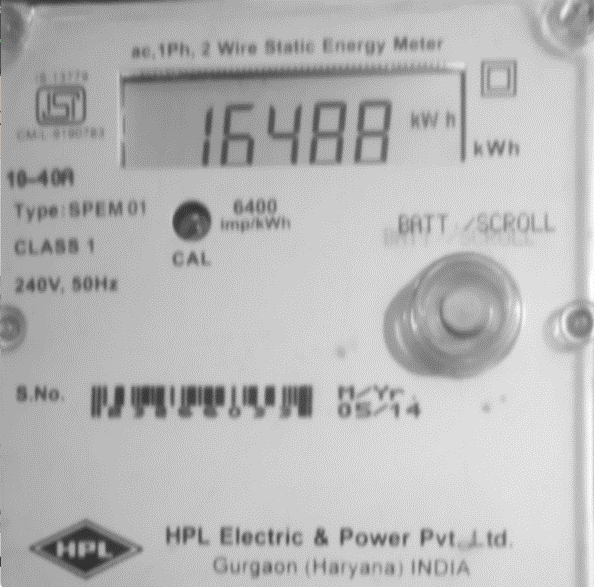


Fig. 3.12

***Thresholding :***

Purpose: Converts grayscale images to binary images, distinguishing the foreground (meter readings) from the background.

Implementation: Adaptive thresholding is used to handle varying lighting conditions within the image, ensuring that the numeric segments are distinctly separated from the background.

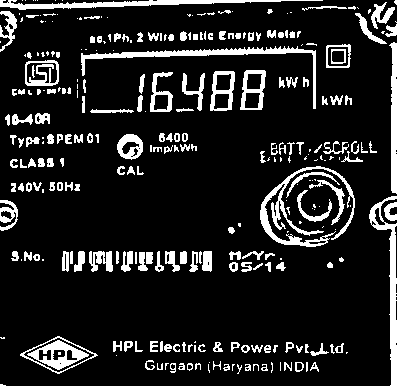


Fig. 3.13

***Canny Edge Detection :***

Purpose: Identifies and highlights the edges within the image, crucial for contour detection.

Implementation: Two most common used edge detection algorithms are :

1) Canny Edge Detection - Canny edge detection is a widely used method that combines the strengths of edge detection and noise suppression. It is highly effective in identifying the contours of objects within an image.

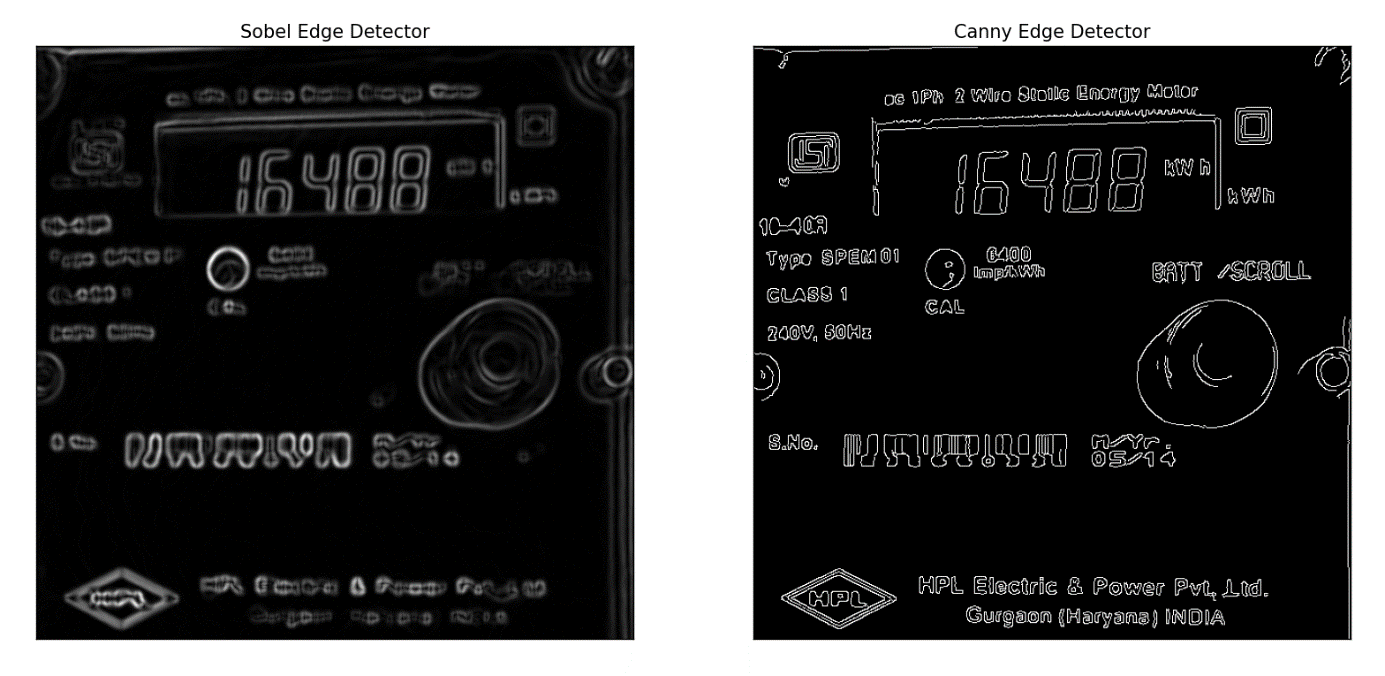
2) Sobel Edge Detection - Sobel edge detection is a simple and efficient method that uses convolution with specific kernels to calculate the gradient magnitude and direction at each pixel.

Fig. 3.14

***Contour Approximation :***

Purpose: Detects and approximates the contours within the image to identify the shapes and boundaries of objects.

Implementation: Contours are detected and approximated to simplify the shapes, making it easier to identify the regions containing the numeric readings. This step is critical for isolating the numeric segments for further analysis.



Fig. 3.15

**3.5.2 Postprocessing of the Images :**

**ROI Selection**

There are a few common ways to select an ROI:

**Manual Selection**: The user manually selects the ROI by drawing a bounding box, polygon, or other shape around the area of interest using a graphical user interface (GUI) or programming tools.

**Automatic Selection**: The ROI is automatically detected and selected based on the output of previous processing steps, such as object detection, edge detection, or image segmentation.

**Predefined ROI**: The ROI is predetermined based on the application or the known location of the object of interest in the image.

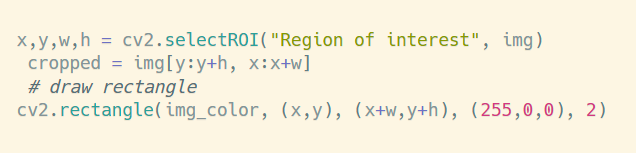


Fig. 3.16

In most cases, it simply wouldn't be realistic to render an image before manually specifying our region of interest.

OpenCV made this incredibly easy through the contourArea() and arcLength() function.

We obtain the contour area and parameter iteratively in a for-loop, like the following:

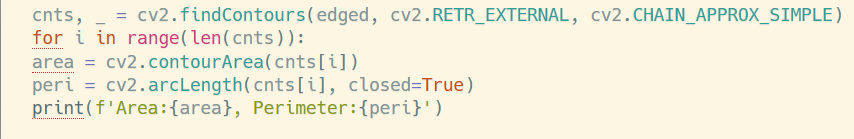


Fig. 3.17

In effect, we're looping through each contour that the findContours() operation found, and computing two values each time, area and peri.

**Drawing operations** are used to visualize and annotate specific features within an image. By drawing rectangles around detected objects and adding text labels, we can clearly highlight and identify important elements. This step is particularly useful for debugging and presenting results.

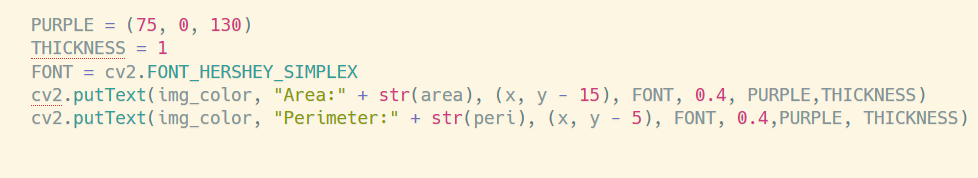
****

Fig. 3.18

****

Fig. 3.19

**ROI Extraction**

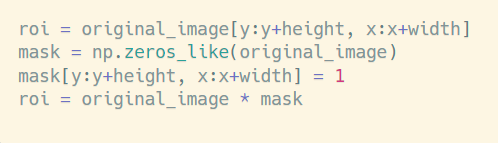
The basic steps for ROI extraction are:

*1. Obtain the Coordinates:* Determine the coordinates (e.g., x, y, width, height) of the selected ROI.

*2. Crop the Image:* Use the ROI coordinates to crop the original image and extract only the selected region.

*3. Apply a Mask:* Alternatively, you can create a binary mask that is the same size as the original image, with 1s (white) in the

ROI and 0s (black) elsewhere. Then, you can apply this mask to the original image to extract the ROI.

****

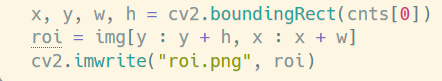
****

Fig. 3.20

Extracted ROI :

****

Fig. 3.21

Now the the image is ready to move on to the next phase of the digit recognition pipeline:

**Morphological Transformations**

After isolating the region of interest (ROI), the image might still contain noise, especially if the ROI was obtained through thresholding methods. These methods can inadvertently include "non-features" (noises) in the resulting image. To address these imperfections, a series of morphological operations are performed.

**Erosion**: Erosion reduces the boundaries of foreground objects by sliding a kernel over the image and setting a pixel to 1 only if all the pixels under the kernel are 1. This process removes small noise and shrinks the foreground objects.

**Dilation**: Dilation is the opposite of erosion. It sets a pixel to 1 if at least one pixel under the kernel is 1, effectively expanding the foreground objects.

These operations are generally performed on binary images, where pixel values are either white (1) or black (0). The convention is to keep the foreground white and the background black. By combining erosion and dilation (erosion followed by dilation or vice versa), we can achieve better noise reduction and object enhancement

**Opening and Closing**:

Another name for Erosion, followed by Dilation is the Opening. It is useful in removing noise in our image. The reverse of Opening is Closing, where we first perform Dilation followed by Erosion, particularly suited for closing small holes inside foreground objects.

Here, we have a grayscale image where the foreground is mostly black and the background is white. After **binarizing and inverting** the image to get a white foreground on a black background, we perform **erosion** to eliminate white noise.

This step, however, **shrinks** the foreground objects.



Fig. 3.22

We then apply **dilation** to restore the objects to their original size.

****

Fig. 3.23

**Seven-Segment Display Detection**

The seven-segment display, commonly used in devices like digital clocks, electronic meters, calculators, and banking security tokens, represents digits using seven individual segments. This section focuses on isolating and identifying each digit displayed by these segments.

To perform digit classification based on the state of each segment, we represent the segments' states in a lookup table using a dictionary. Each digit (0-9) corresponds to a specific combination of segment states, represented as binary values.

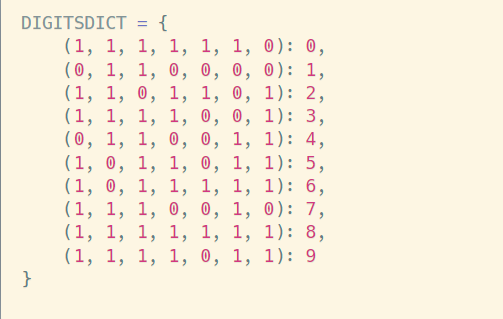
****

Fig. 3.24

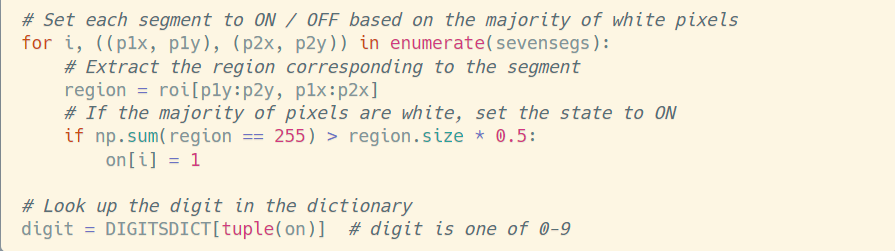
****For each digit, we examine the pixel values in each of the seven segments. If the majority of pixels in a segment are white, we classify that segment as being activated (state of 1); otherwise, it is deactivated (state of 0). After evaluating all seven segments, we form a binary array representing the digit, which we use to look up the corresponding value in our dictionary.

Fig. 3.25

***CHAPTER 4 ~* RESULTS** **& ANALYSIS**

**4.1 Project Outcomes**

The custom-trained YOLOv8 model achieved high accuracy in detecting electric meters in user-uploaded images, demonstrating its effectiveness and robustness. The integration of traditional image processing techniques for numeric extraction further enhanced the system's reliability.

**Meter Detection Results:**

1. **High Detection Accuracy:**

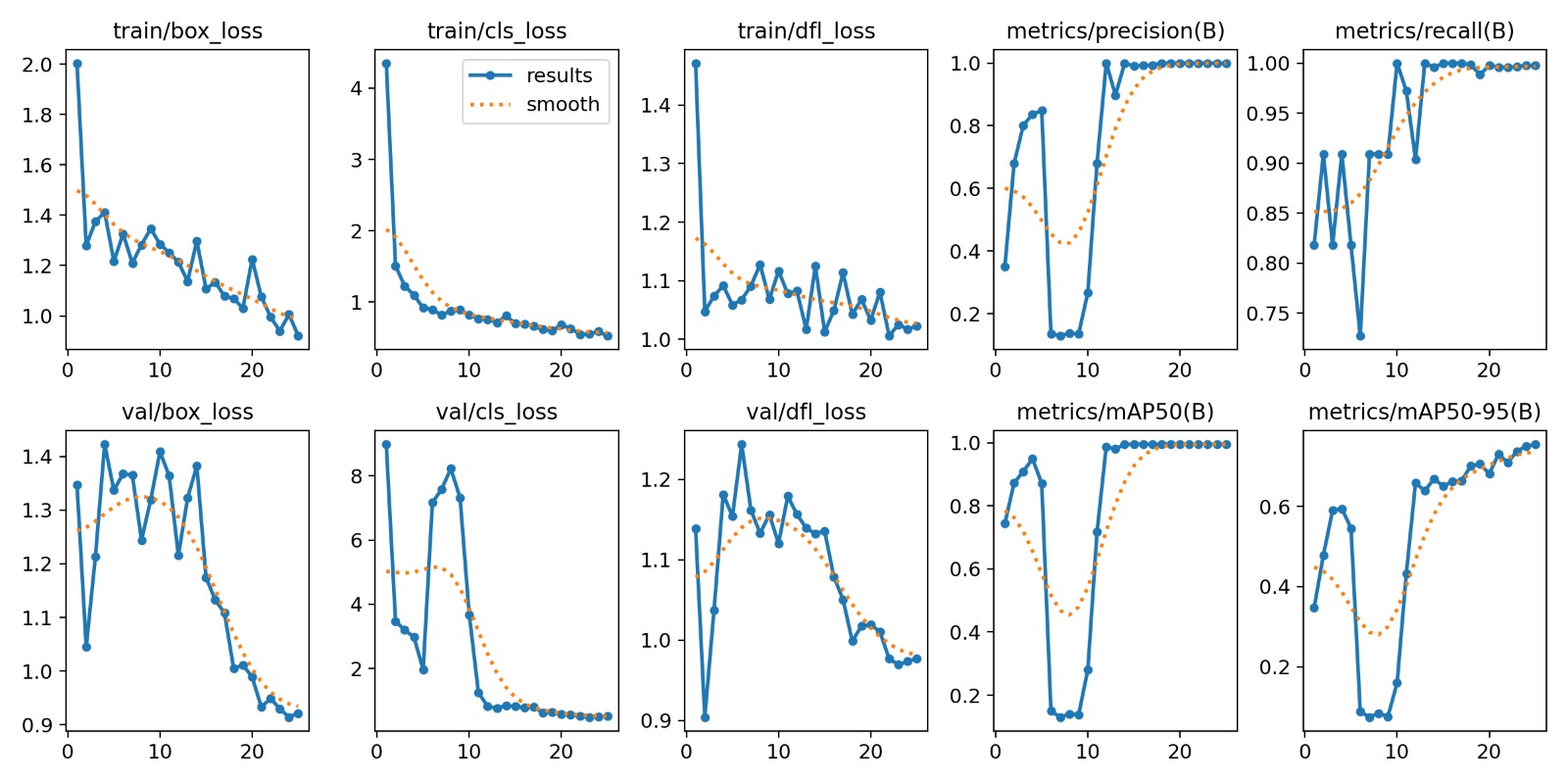
The YOLOv8 model was able to accurately detect electric meters with a high degree of precision. The model's performance metrics, including precision, recall, and mean Average Precision (mAP), indicated its capability to correctly identify meters in various lighting conditions and backgrounds.

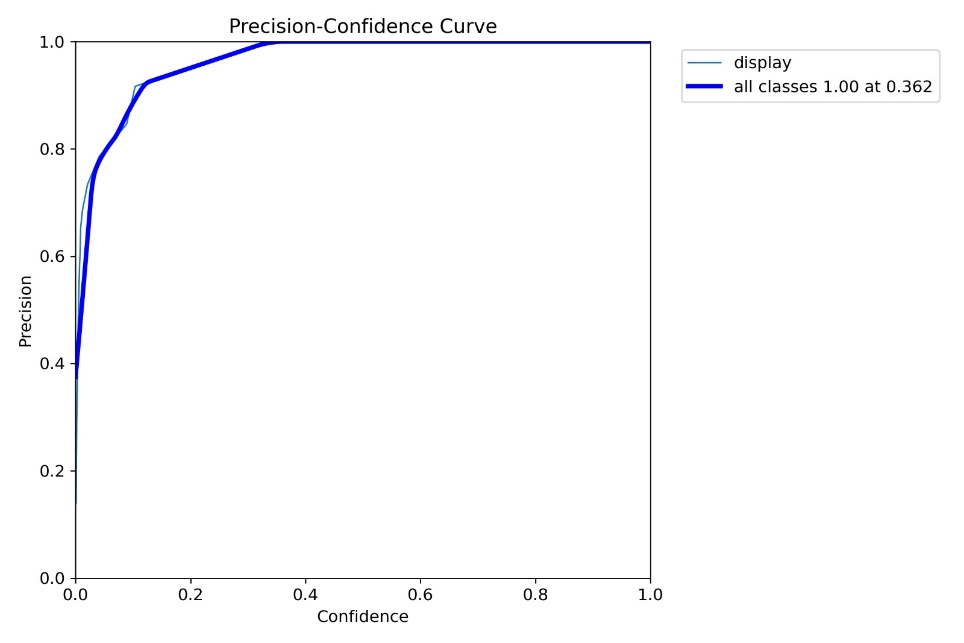
1. **Reduction in False Positives and Negatives:**

Through rigorous training and validation, the model achieved a significant reduction in both false positives (incorrectly identifying non-meter objects as meters) and false negatives (failing to identify actual meters). This ensured that only valid meter images were processed, enhancing the overall accuracy and efficiency of the system.

1. **Efficient Handling of Diverse Images:**

The system was tested on a diverse set of images, including those with varying angles, lighting conditions, and obstructions. The YOLOv8 model consistently performed well across these different scenarios, demonstrating its robustness and adaptability.

 Fig. 4.1 Training and Validation Metrics



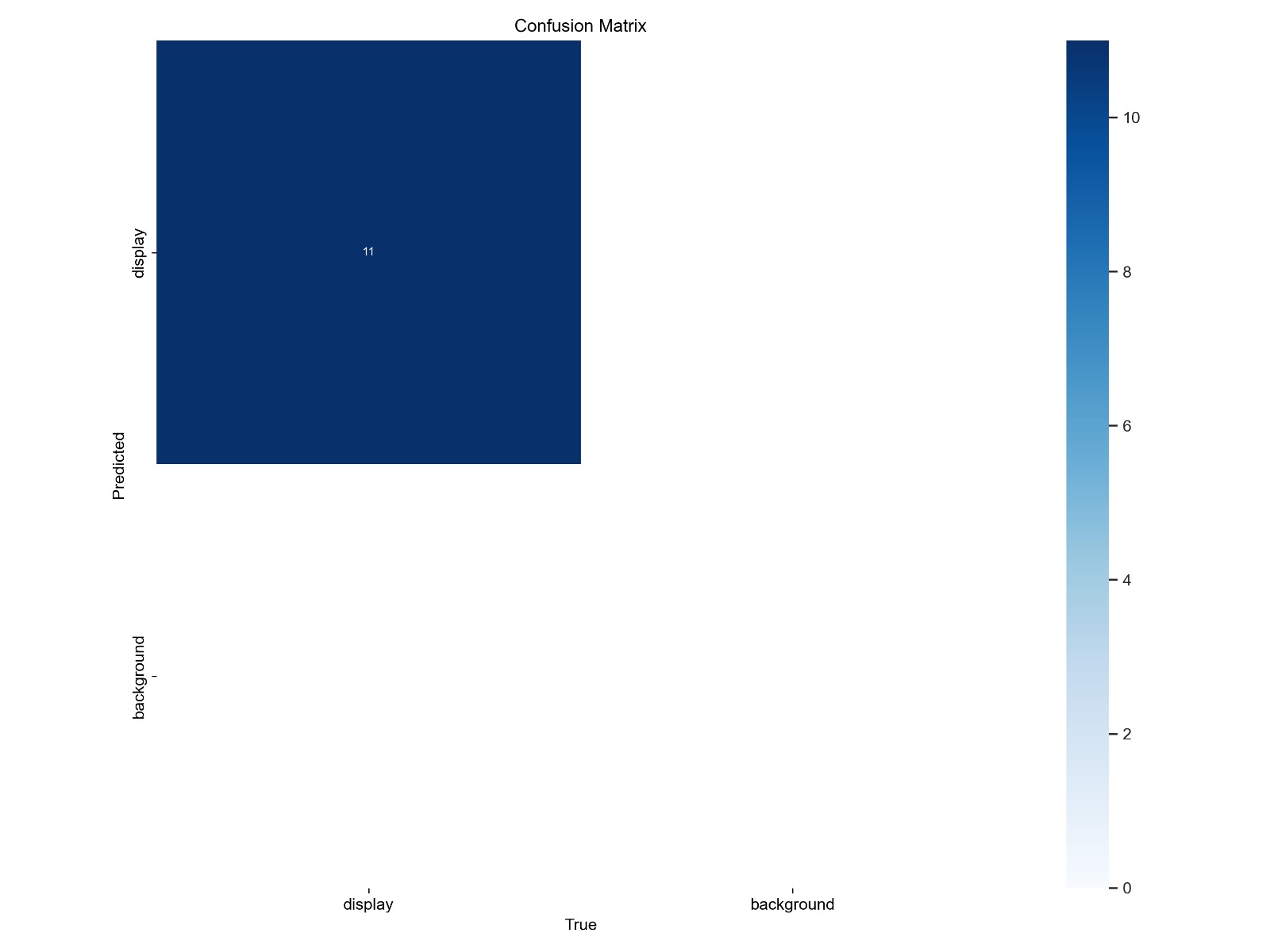
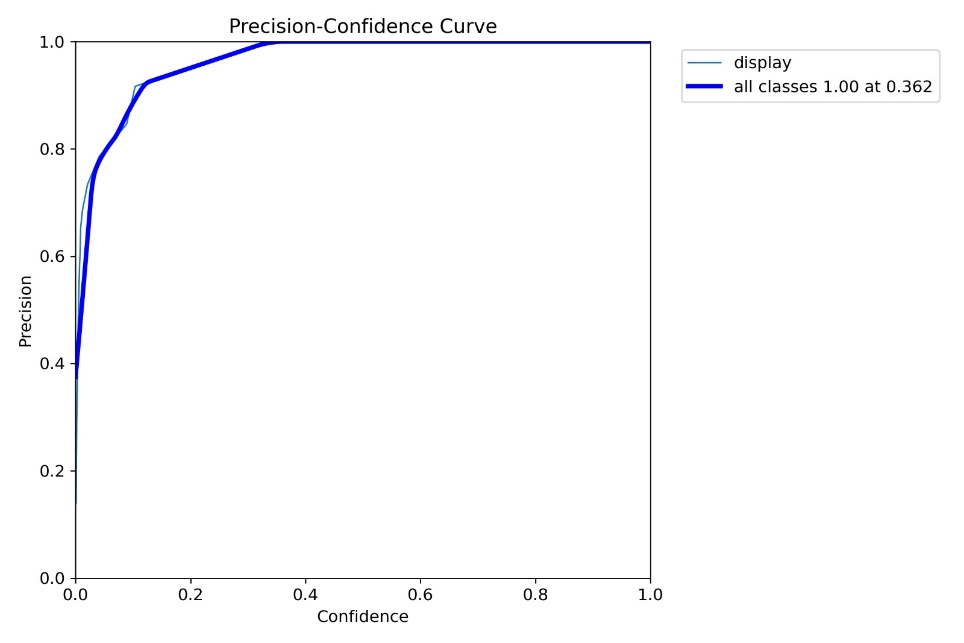
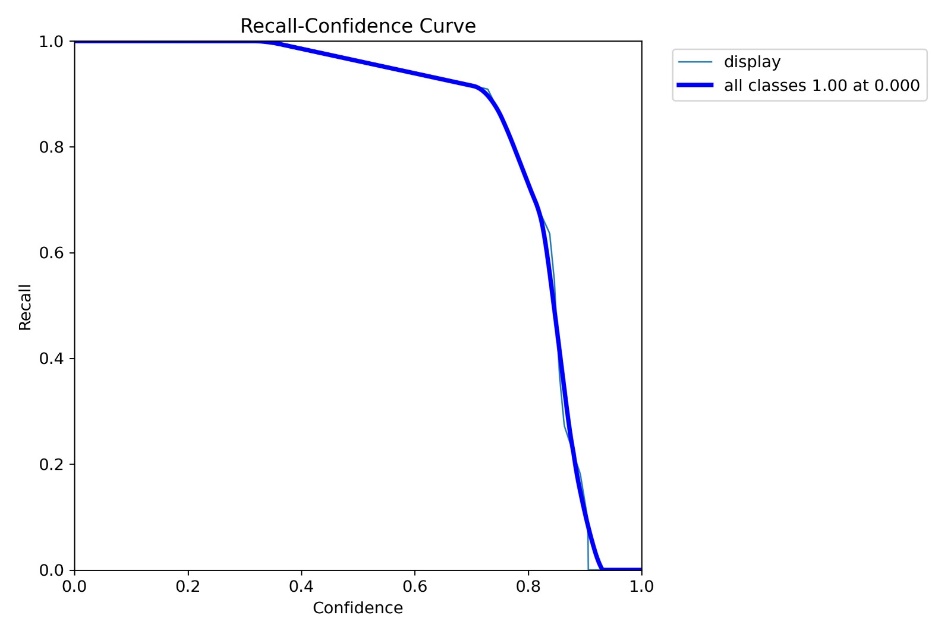
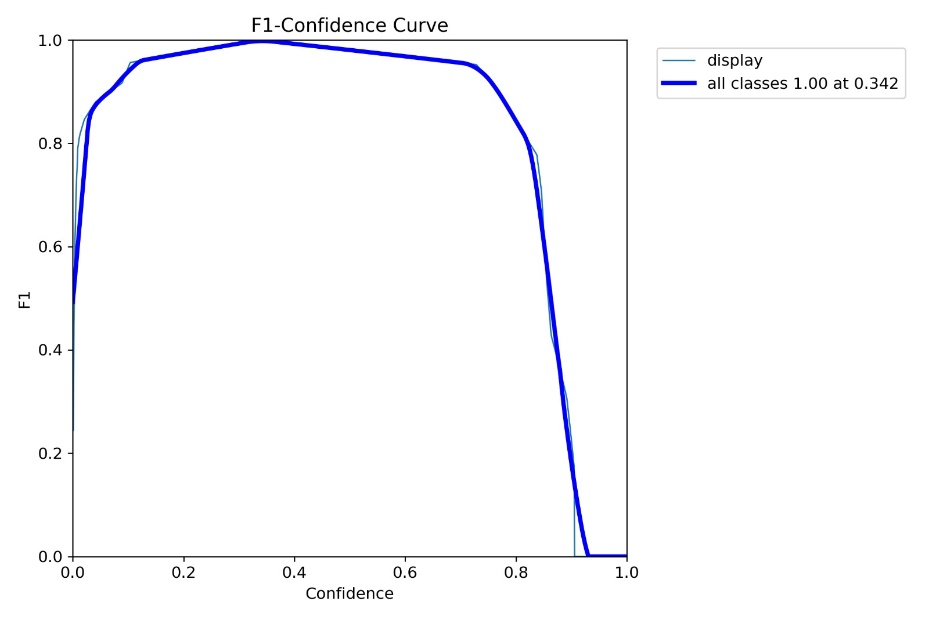
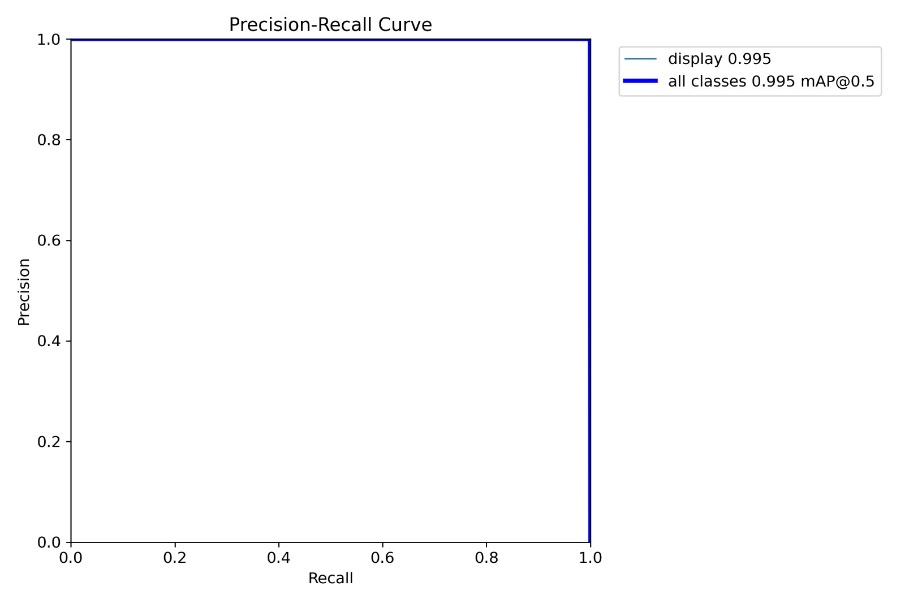


Fig. 4.1 Training and Validation Metrics

Fig. 4.2 Confusion Matrix

Fig. 4.3 Evaluation Curves





**Numeric Extraction Results:**

1. **Accurate Numeric Data Extraction:**

The traditional image processing techniques employed for numeric extraction, including edge detection, contour approximation, and thresholding, successfully identified and extracted numeric data from the detected electric meters. This step was crucial for converting the visual information into readable and storable data.

1. **Integration with YOLOv8:**

The seamless integration of YOLOv8 for detection and traditional methods for numeric extraction ensured that the entire process, from detection to data extraction, was streamlined and efficient. This integration was key to achieving the project's objectives.

1. **Reliability and Consistency:**

The combination of advanced object detection and classical image processing methods provided a reliable and consistent output, ensuring accurate meter readings across various test cases. This consistency is critical for applications requiring high reliability, such as utility management.

**4.2 Comparison with Objectives**

The project successfully met its primary and specific objectives, demonstrating the feasibility and effectiveness of the proposed solution.

**Primary Objective:**

* *Objective***:** Develop an end-to-end application for automated electric meter reading.
* *Outcome***:** Delivered a comprehensive solution integrating YOLOv8 for detection and traditional image processing for numeric extraction.

Made an image uploader website through a Django, yet have to complete the integration between the various files.

**Specific Objectives:**

1. **Object Detection with YOLOv8:**

* *Objective***:** Implement a custom-trained YOLOv8 model.
* *Outcome***:** Achieved high detection accuracy with reduced false positives and negatives.

1. **Numeric Extraction with Traditional Image Processing:**

* *Objective***:** Extract numeric data using traditional techniques.
* *Outcome***:** Successfully extracted accurate numeric information, ensuring reliability and efficiency.

***CHAPTER 5 ~*** **CONCLUSION &**

**LEARNING OUTCOMES**

**5.1 Conclusion**

The project achieved its goal of developing an end-to-end application for automated electric meter reading. By leveraging YOLOv8 and traditional image processing techniques, the system provided an efficient, accurate, and scalable solution for meter reading. The success of this project lays the groundwork for future enhancements and potential integration with more advanced machine learning models.

**5.2** **Learning Outcomes & Personal Experience**

**Learning Outcomes:**

1. Gained hands-on experience with YOLOv8 and its application in object detection.
2. Enhanced understanding of traditional image processing techniques and their practical applications.
3. Developed skills in web application development using the Django framework.

**Personal Experience:**

The project provided valuable insights into the challenges and solutions associated with automated meter reading. The integration of various technologies and techniques offered a comprehensive learning experience, reinforcing the importance of interdisciplinary knowledge in solving real-world problems.

***FUTURE WORKS***

Despite significant progress in developing the automated electric meter reading system, several areas require further work to achieve a fully integrated and functional end-to-end solution. The following future work outlines the necessary steps and potential enhancements to complete and improve the project:

1. **Integration of System Components:**

*Objective***:** Combine the separate modules of image uploading, detection, and numeric extraction into a seamless, unified application.

*Details***:** This involves integrating the YOLOv8 model and traditional image processing pipeline within the Django framework to ensure a smooth workflow from image upload to meter reading extraction.

1. **Backend Enhancements:**

*Objective***:** Improve the backend infrastructure to support robust and scalable operations.

*Details***:** Optimize the Django application to handle larger datasets, implement secure user authentication, and ensure data integrity and privacy.

1. **Real-time Processing:**

*Objective***:** Enable real-time processing of meter readings for immediate feedback.

*Details***:** Implement asynchronous processing and real-time notification features to inform users of the status of their uploads and the resulting meter readings.

1. **Extensive Testing and Validation:**

*Objective***:** Conduct thorough testing to ensure accuracy and reliability.

*Details***:** Perform extensive testing on diverse datasets to validate the model's performance and robustness. Identify and fix any issues related to edge cases and diverse image conditions.

***REFERENCES***

[1] J. Canny , “A Computational Approach to Edge Detection.” , 1986

[2] R. Gonzalez and R.Woods “A Threshold Selection Method”

[3] Xin-Yi Gong , “An Overview of Contour Detection Approaches”, 2018

[4] M. Hangarge , “Shape and Morphological Transformation Based Features for Language Identification”, 2008

[5] [W. Lin , Z. Zhao , J. Tao, C. Lian and C. Zhang](https://www.mdpi.com/2076-3417/13/12/7146) , “Research on Digital Meter Reading Method of Inspection Robot Based on Deep Learning”, 2013

[6] [Xianghui Liu; Yancong Deng; Yingkai Sun; Yuting Zhou](https://ieeexplore.ieee.org/document/8686963/authors) , 2019 “Multi-Digit Recognition with Convolutional Neural Network and Long Short-Term Memory”

[7] [OSSpk](https://github.com/OSSpk/Handwritten-Digits-Classification-Using-KNN-Multiclass_Perceptron-SVM/blob/master/README.md) , “A Comparative Study on Handwritten Digits Recognition using Classifiers like K-NN, Multiclass Perceptron and SVM”, 2018

[8] [Li Deng and Dong Yu](https://www.nowpublishers.com/article/Details/SIG-039), “Deep Learning: Methods and Applications," Foundations and Trends in Signal Processing, 2014.

[9] [A. Krizhevsky, I. Sutskever, and G. E. Hinton](https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html), “ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, 2012.”

[10] [C. Olah](https://colah.github.io/posts/2015-08-Understanding-LSTMs/), “Understanding LSTM Networks,” 2015.

[11] [OpenCV: Morphological Transformations](https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html)

[12] [Seven-segment Display](https://pyimagesearch.com/2017/02/13/recognizing-digits-with-opencv-and-python/)