



Sobia Barkat

Automated Meter Reading System

Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

30 Nov 2024

PREFACE

This thesis aims to pursue a project that involves developing an automated system to read and record utility meter data. In the whole project, much attention was given to designing a robust solution that could process various types of input data and ensure the accuracy of the results. Much attention had to be paid to data processing, error reduction, and validation in this project to develop a system that could satisfy practical needs for various conditions of the real world.

This could not have been possible without my mentors, colleagues, and loved ones, their encouragement fueling this project. I am deeply grateful for their pursuits, which were responsible for completing this project in the first place. I do hope that this thesis will contribute to the ongoing developments in the field of automated data processing and may perhaps ignite another wave of future innovations.

Espoo, 30 Nov 2024

Sobia Barkat

Abstract

Author: Sobia Barkat
Title: Automated Meter Reading System
Number of Pages: 54 pages
Date: 30 November 2024

Degree: Master of Engineering
Degree Programme: Information Technology
Professional Major: Networking and Services
Supervisors: Sami Sainio, Principal Lecturer
Ville Jääskeläinen, Principal Lecturer

Advancements in automation technology have transformed several industries, including utility management companies. These advanced systems offer improvements in their services regarding capability, accuracy, and performance. One example of improvements is in Automated Meter Reading (AMR) frameworks. Since traditional meter reading methods are time-consuming and prone to errors, this has proven to be a fruitful field for AMR deployment. These systems have replaced manual human interpretations with new calculating processes, like image processing and optical character recognition (OCR), for extracting accurate readings from electricity meters. This thesis describes a software-based solution that presents the plan, advancement, and assessment of an imaginative AMR framework to collect data from meters.

This study explores challenges in the automation of utility meter readings, including design differences in meters, lighting conditions, and image distortion. This thesis solves these problems via advanced preprocessing techniques and adaptive recognition algorithms.

Keywords: Automated Meter Reading, Image Enhancement, Computer Vision, Image Processing

The originality of this thesis has been checked using the Turnitin Originality Check service.

Table of Contents

| | | |
|-------|---|----|
| 1 | Introduction | 1 |
| 1.1 | Background | 1 |
| 1.2 | Problem Statement | 2 |
| 1.3 | Objectives | 3 |
| 1.4 | Rationale and Research Scope | 3 |
| 1.4.1 | Rationale | 3 |
| 1.4.2 | Research Scope | 3 |
| 1.5 | Thesis Outline | 4 |
| 2 | Literature Review | 5 |
| 2.1 | Overview of Automated Meter Reading (AMR) Systems | 5 |
| 2.2 | Optical Character Recognition (OCR) | 5 |
| 2.2.1 | Types and Methods of OCR | 7 |
| 2.2.2 | Applications for OCR | 7 |
| 2.3 | Image Processing | 8 |
| 2.3.1 | Image modeling and representation | 8 |
| 2.3.2 | Image Enhancement | 9 |
| a) | Median Filter | 9 |
| b) | Mean Filter | 10 |
| 2.3.3 | Image Restoration | 10 |
| 2.3.4 | Image Analysis | 11 |
| 2.4 | Applications of Python Tesseract and OpenCV | 13 |
| 2.5 | Comparative Analysis of Existing AMR Technologies | 14 |
| 2.6 | Challenges in Manual Meter Reading | 15 |
| 2.7 | Advancements in OCR Technologies | 16 |
| 2.8 | Emerging Trends in AMR Systems | 16 |
| 3 | System Design and Implementation | 18 |
| 3.1 | Challenges in Manual Meter Reading | 18 |
| 3.1.1 | Hardware components: | 19 |
| 3.1.2 | Software requirements: | 19 |
| 3.2 | Overall System Architecture | 19 |
| 3.2.1 | Image Upload Module | 20 |
| 3.3 | OCR and Image Processing Module | 20 |
| 3.3.1 | Python Tesseract Implementation | 21 |
| 3.3.2 | OpenCV Techniques | 23 |

| | | |
|-----|--|----|
| 3.4 | Integration of OCR and Image Processing..... | 26 |
| 3.5 | Data Storage and Management..... | 27 |
| 3.6 | Bill Calculation Module..... | 28 |
| 4 | Testing, Evaluation, and System Validation..... | 29 |
| 4.1 | Testing Methodology..... | 29 |
| 4.2 | Result Analysis and System Validation..... | 34 |
| 4.3 | Advantages of the Developed System..... | 39 |
| 4.4 | Comparative Cost-Benefit Analysis..... | 39 |
| 4.5 | Comparative Performance Analysis..... | 40 |
| 4.6 | Limitations, Challenges, and Potential Improvements..... | 41 |
| 5 | Conclusion and Future Work..... | 42 |
| 5.1 | Summary of Findings..... | 42 |
| 5.2 | Recommendations for Future Research..... | 42 |

References

List of Figures

| | |
|--|----|
| Figure 1.0 Working of the current system | 2 |
| Figure 2.0 Flow process of OCR..... | 7 |
| Figure 2.1 Median Filter to reduce noise..... | 10 |
| Figure 2.2 Skeletonization..... | 11 |
| Figure 2.3 Different types of utility meters and pictures taken at different angles..... | 15 |
| Figure 3.0 Design of the system..... | 19 |
| Figure 3.1 Python Tesseract Techniques..... | 23 |
| Figure 3.2 Flow process of OpenCV..... | 26 |
| Figure 3.3 Flow process of developed system..... | 28 |
| Figure 4.0 Preprocessing techniques for image enhancement..... | 32 |
| Figure 4.1 Working with OpenCV..... | 33 |
| Figure 4.2 Processed image for further analysis..... | 33 |
| Figure 4.3 Data extraction from the meter image..... | 34 |
| Figure 4.4 a) Meter reading scanning..... | 35 |
| b) Saving data..... | 36 |
| Figure 4.5 a) Data storage with different parameters..... | 37 |
| b) Data retrieval..... | 37 |
| Figure 4.6 Storing of information into system..... | 38 |
| Figure 4.7 a) Different options related to meter reading..... | 39 |
| b) Information related to specific meters..... | 39 |
| Figure 4.8 Comparative Cost Benefit Analysis of Systems..... | 41 |

List of Abbreviations

| | |
|--------|-------------------------------------|
| AI | Artificial Intelligence |
| AMR | Automated Meter Reading |
| CNN | Convolutional Neural Network |
| LSTM | Long Short-Term Memory |
| IoT | Internet of Things |
| ML | Machine Learning |
| MYSQL | My Structured Query Language |
| OCR | Optical Character Recognition |
| OpenCV | Open-Source Computer Vision Library |
| PSM | Page Segmentation Mode |
| SQLite | Structured Query Language Lite |
| RGB | Red Green Blue |
| RNN | Recurrent Neural Network |
| ROI | Region of Interest |

1 Introduction

1.1 Background

Owing to the demand for increased accuracy and efficiency, utility metering framework development has moved beyond human readings to include advanced and automated methods. The billing system and control management of public utilities are still not fully automated in many countries[1]. Presently, a person from the electricity, water, or gas board goes to every building and takes the meter reading manually. These meter readings are used for electricity, water, or gas bill calculation[2]. Conventional manual meter reading strategies are tedious and inclined to human mistakes, prompting incorrect charging and client disappointment. Service organizations are working to modernize their operations by cutting costs and using automation to handle repetitive tasks. This helps employees focus on more important and productive work. Automated Meter Reading (AMR) systems are becoming a common part of this change.

Most advancements in AMR systems focus on upgrading utility meters with expensive hardware like sensors and communication modules to allow automatic meter readings. While these systems are effective, upgrading an entire network is costly and difficult. A simpler solution is for the company's employees to capture a photo of the meter using a device and upload it to a server. The system then processes the data automatically, avoiding the need for costly hardware upgrades.

This work aims to provide an overview of how to enable a digital photo based system for reading electric meters without the need for power mains to be upgraded or modified. The idea is to create a system that works with the current infrastructure, so no expensive upgrades are needed. The system uses photos of utility meters taken by smartphones or cameras. Python Tesseract for OCR and OpenCV for image processing can automatically read the numbers from these photos and add them to the billing system. The aim is to use the common

availability of smartphones and the accuracy of modern image processing tools to make the process simple and efficient.

1.2 Problem Statement

XYZ Company is an electricity distributor that manually collects utility meter readings. The process includes taking photos of the meters and manually entering the interpreted information into data storage. This method is labor intensive and carries a risk of human error. Despite the technical solutions available, XYZ Company had ongoing problems in ensuring the accuracy and usefulness of meter readings, which directly impacted customer loyalty and operational costs. Additionally, XYZ faces challenges in adopting modern solutions due to the high costs of replacing existing infrastructure, such as installing smart meters and upgrading network systems to support automated data transfer. These changes require significant financial investment, which is currently beyond the company's budget and operational capacity. Figure 1.0 shows the current working state.

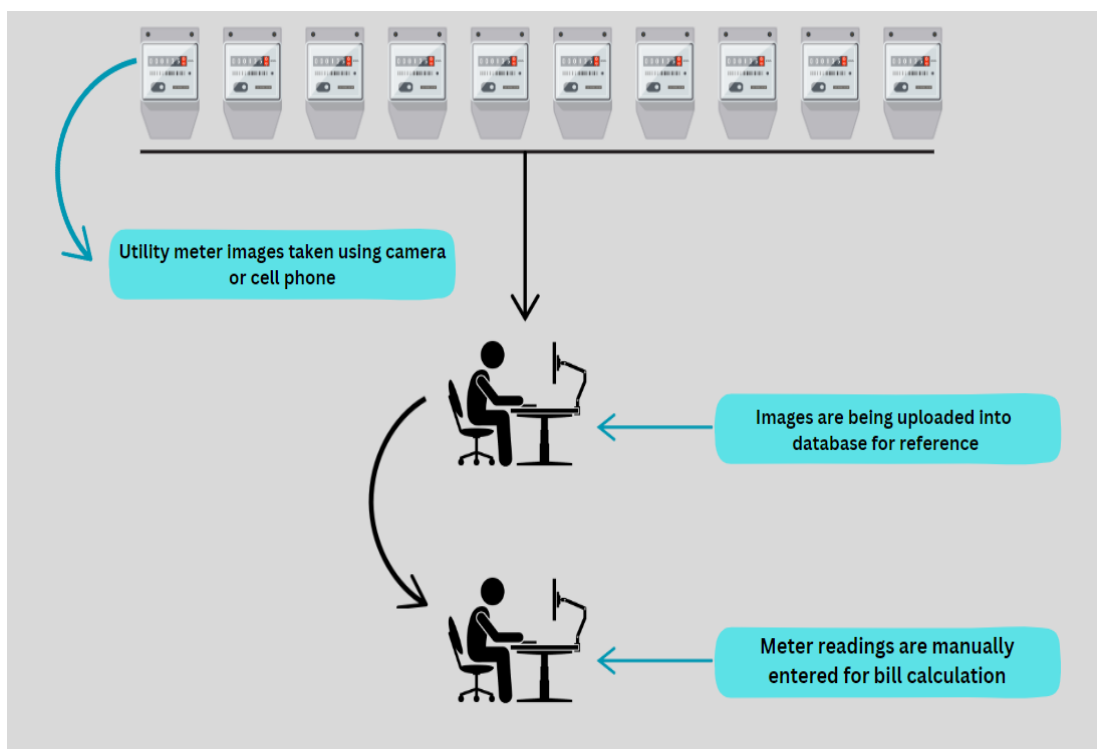


Figure 1.0 Working of the current system

1.3 Objectives

The main goal of this thesis is to develop an Automated Meter Reading (AMR) system that extracts data from digital photos of utility meters. The system uses OpenCV and Python Tesseract to automatically process the images and read the meter values. This reduces the chances of errors that happen when meter readings are entered manually into the billing system. The project focuses on improving the accuracy and efficiency of the billing process for the power distribution company by providing an easy-to-use software solution. The system shows that image processing and OCR technologies make utility meter readings more reliable and help to manage labor better.

1.4 Rationale and Research Scope

1.4.1 Rationale

XYZ Company operates in a third-world country, serving over 8.2 million consumers. While smart meters are available, upgrading them is too expensive. A single smart meter costs around 250 euros, and additional costs for installation and training make it even more expensive. Replacing all the meters would cost 2.05 billion euros, which is not affordable given the region's economic situation. This is why a software-based solution is a better and more practical choice.

1.4.2 Research Scope

This thesis focuses on creating a simple and affordable Automated Meter Reading (AMR) system using image processing and OCR techniques. The project includes:

- Building a basic system that uses OpenCV and Python Tesseract to read numbers from photos of utility meters.
- Developing software that lets employees upload meter photos and automatically extract the meter readings.

- Integration of the extracted data into the existing billing system.
- Testing the system by comparing its accuracy and reliability with the current manual process, including handling errors.

The project also looks at whether this solution is cost-effective. It shows how Python, Tesseract, and OpenCV can be used for AMR without relying on expensive or specialized hardware and software.

1.5 Thesis Outline

The content of the thesis can additionally be divided into four more parts.

Chapter 2: *Literature Review*: This chapter presents the background of previous studies and innovations in the field of energy meter reading. The various manual techniques, AMR innovations, and ongoing advances in image processing and OCR are reviewed. The limitations in current methodologies are reviewed to justify the need for the proposed framework.

Chapter 3: *System Design and Implementation*: This chapter provides the step-by-step procedure of system development. It describes the system requirements and architecture and outlines the different modules that are the basis of the system. Later practical implementation of the system has been discussed, highlighting the use of Python Tesseract, OpenCV and OCR.

Chapter 4: *Testing, Evaluation and Discussion*: This chapter covers the results of the performance testing conducted in the system, especially concerning number recognition aspects of accuracy and efficiency. The system design provides a comparison between the manual and automated systems, and case studies are presented to test the effectiveness of the system.

Chapter 5: *Conclusion and Future Work*: This final chapter summarizes the findings of research, and potential improvements for future work are explained.

2 Literature Review

2.1 Overview of Automated Meter Reading (AMR) Systems

Automated Meter Reading (AMR) technology automatically reads and records meter data using image processing techniques [2]. It is commonly used in monitoring equipment and billing systems [3]. AMR helps utility companies collect energy consumption data without manual effort. Many systems rely on hardware improvements, like special meters or communication devices, to send data. Over time, AMR has improved from simple one-way systems to advanced two-way systems that allow remote data collection and control.

AMR systems often depend on hardware upgrades, such as installing smart meters or other devices to improve functionality. Initially, these systems used one-way communication, where meters sent information to utility companies but couldn't receive anything in return. Over time, they evolved into two-way systems that allow utility providers to do much more. These advanced systems can gather data remotely, monitor energy usage, and adjust consumption when needed. For example, during peak hours, utilities can manage the load by reducing or limiting energy use, which helps improve efficiency and reliability.

Focusing on hardware solutions has made AMR systems expensive and difficult to set up. This has slowed down their use, especially in places where older systems are still in place. There is a growing need for software solutions that can work with existing meters without needing big changes to the infrastructure. However, most advancements in AMR still focus on improving hardware, while software approaches like image processing and OCR to automate meter readings are not explored much.

2.2 Optical Character Recognition (OCR)

Optical character recognition (OCR) converts scanned documents PDFs into computer-readable and editable formats, e.g. ASCII characters. OCR is the

process of classifying optical patterns in a digital image corresponding to alphanumeric or other characters [4]. Despite the title OCR, the processing of digital images still needs to be done. Therefore, it is essential to have basic knowledge of image processing and character recognition. The word "recognition" plays an important role in our lives. It is a fundamental property of all human beings: when people see an object, they first gather all the information about it and compare its properties and behaviors with the existing knowledge stored in their minds.[4] If we find a suitable match, we recognize it. If we see an object, it has features that describe it, that is, a set of characteristics of that object. These sets, when combined, create a fixed shape and size of that object, called the object model [4]. If multiple objects share the same model, those models belong to a model class. Pattern recognition is the process by which the features of objects are assigned to a pattern class. Given a pattern, its recognition/classification can consist of one of two functions:

- supervised classification (e.g., discriminant analysis) in which the input pattern is identified as a member of a predefined class.
- unsupervised classification (e.g., clustering) in which the pattern is assigned to a previously unknown class [5].

Pattern recognition can be used to develop applications for fingerprint identification, voice recognition, face recognition, character recognition, and signature recognition. It can also be used in scientific research fields such as astronomy, engineering, statistics, medicine, machine learning, and neural networks. The design of a pattern recognition system includes the following three aspects: 1) data acquisition and processing, 2) data representation, and 3) decision-making [5].

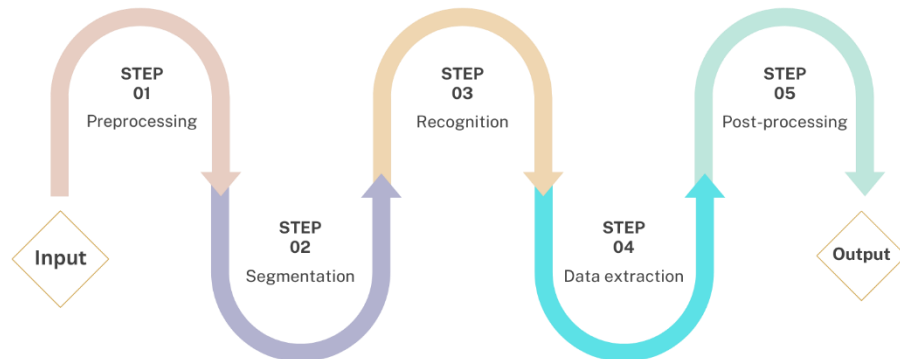


Figure 2.0 Flow process of OCR

2.2.1 Types and Methods of OCR

OCR systems can be divided into traditional and modern methods. Traditional methods include pattern matching, where images are compared to templates, and feature extraction, which looks at shapes like edges and loops to identify characters. These methods don't work well with complex data. Modern OCR incorporates machine learning to recognize complex handwriting and adapt to various fonts and layouts[6].

OCR can also be online or offline. Online OCR uses cloud services to process text quickly, making it great for mobile apps. Offline OCR works without the internet, which is better for secure environments or limited connectivity. Tools like EasyOCR are often used for offline OCR.

2.2.2 Applications for OCR

OCR technology is used in many fields and has changed how organizations handle text. It is commonly used to scan printed books, historical records, and reports into digital formats, making old texts easier to access and preserve. Other

uses include reading license plates in traffic systems and automating tasks like extracting information from invoices and receipts.

In industries, OCR has applications beyond document processing, including healthcare for digitizing patient records, banking for processing checks and forms, and logistics for tracking shipments through label recognition[7]. In healthcare, it manages patient files and improves security. In banking, it is used for verifying documents and detecting fraud. Shipping companies use OCR to track packages by reading labels. Its flexibility makes OCR an important tool for automation and digital transformation.

2.3 Image Processing

Extracting text from an image and preparing it for recognition is done through digital image processing. This process includes segmenting characters, removing noise, skeletonizing shapes, and normalizing them for better accuracy. Before recognizing characters, it is important to separate each one from the text image. Digital image processing works with two-dimensional images converted into binary data—combinations of 0s and 1s [8].

Digital image processing is widely used in many fields, including satellite remote sensing, image storage for businesses, medical imaging, radar, sonar, robotics, and quality inspection in industries [9]. The operations in digital image processing can be divided into five main categories, detailed below.

2.3.1 Image modeling and representation

Image representation and models focus on finding ways to describe images mathematically. There are two main types of representation. The first is external representation, which focuses on the boundary or outline of the image and is used to study shape characteristics. The second is internal representation, which looks at the pixels inside the image and is used to analyze properties like color

and texture. In some cases, both methods are needed to understand the image fully.

2.3.2 Image Enhancement

Image enhancement is the process of improving features in an image, like edges, boundaries, or contrast, to make it clearer for viewing and analysis [9]. It doesn't add new information but makes features easier to see. Common techniques include adjusting brightness and contrast, reducing noise, sharpening edges, applying filters, enlarging images, and adding colors.

i) Noise reduction/removal

Image noise appears like random grains or specks and happens when pixel values don't match what they should be. It can be caused by light, heat in sensors, or issues during image capture or transmission [10]. In binary images, noise is when a black pixel turns white or a white pixel turns black. If not fixed early, it can cause problems in later processing steps.

There are two main types of noise:

- **Salt noise:** A black pixel appears as white.
- **Pepper noise:** A white pixel appears as black.

To fix this, filters like mean, median, Laplacian or mode filters are often used to clean up the image.

a) Median Filter

The median filter is an effective way to remove noise. It's a non-linear method that helps smooth images while keeping edges sharp. To use it, a window or template like 3x3 or 5x5 is applied over the image. The pixel values inside this window are sorted, and the middle (median) replaces the processed pixel [11]. This filter works well for reducing noise without losing important details.

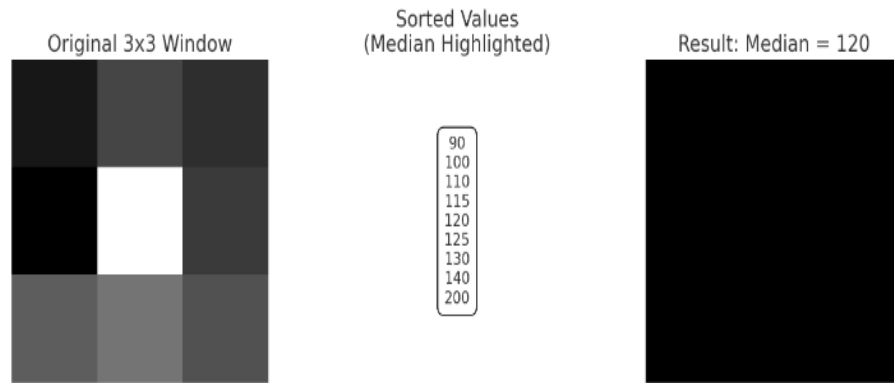


Figure 2.1 Median Filter to reduce noise

b) Mean Filter

The mean filter is one of the simplest ways to smooth an image. It works by averaging all the pixel values within a chosen window. The result replaces the original pixel value. While it's easy to use, the mean filter often causes blurring around the edges. This makes it less ideal when preserving sharp details is important. It's a basic linear filter suited for general smoothing tasks.

2.3.3 Image Restoration

Images captured by cameras or electronic devices can lose quality due to environmental issues. Problems like bad lighting, color errors, and blurring often occur. Blurring happens when the camera moves while taking the photo or if the lens isn't properly focused [8].

Image restoration aims to fix these issues by recovering the original image from a blurred or noisy version. It's also called deblurring or deconvolution. To restore an image, it's important to understand what caused the problem and choose the right method [8]. Unlike image enhancement, which makes images look better, restoration works to bring back lost details as accurately as possible.

2.3.4 Image Analysis

Image analysis is the process of extracting useful information from images. This can include finding shapes, detecting edges, removing noise, or counting objects. It also involves measuring properties like size and region. Techniques such as segmentation, skeletonization, slant correction, and size normalization are used.

i) Skeletonization

Skeletonization is the result of the thinning process, which peels the color of the text until it reaches the most medial one-pixel width [12]. During this process, the pixels of an object are removed without affecting the general shape of the pattern. The output object, after skeletonization, should be one pixel thick through the middle of the object, and the topology of the object should be preserved.



Figure 2.2 Skeletonization [13]

ii) Segmentation

Segmentation is a mid-level processing technique used to analyze the image and can be defined as a processing technique used to classify or cluster an image into several disjoint parts by grouping the pixels to form a region of homogeneity based on the pixel characteristics like gray level, color, texture, intensity and other features[14]. It is the process that finds the boundary and the object in an image. Using the segmentation process, we can separate and isolate the object of interest from the background of an image. The different approaches to segmenting an image are listed below.

- a) **Discontinuity-based methods:** An intensity-based method uses a threshold to divide an image into two groups. Pixels are grouped based on their intensity values—those within a specific range are placed in one group, while the rest go into another.
- b) **Discontinuity-based methods:** A discontinuity-based method identifies objects by detecting sudden changes in pixel intensity. When neighboring pixels show sharp differences in intensity, these changes are marked as edges. These edges are then connected to outline the object's boundary.
- c) **Similarity-based methods:** A similarity-based method compares neighboring pixels to check if their characteristics are alike. Pixels within the same region share similar traits, while those in different regions are distinct. If the two neighboring pixels are similar, they are grouped together, and the region expands. This process continues until no more similar pixels are found.
- d) **Active contour model:** An active contour model helps analyze the shape of an object. It requires prior knowledge of the image's intensity, boundary shape, and texture for accurate results. The goal is to adjust an initial contour so it moves toward the object's boundary. Active contours can be classified into edge-based models, which focus on boundaries, or region-based models, which work with areas inside the object.

iii) Size Normalization

Size normalization ensures all characters are resized to a standard size. After the image is split into individual characters, resizing helps make further processing consistent and straightforward. It removes issues caused by different writing sizes or scanning resolutions. This process involves two steps:

- **Upscaling:** Small images are enlarged to match the standard size. Rows are duplicated to increase the image size, depending on how much resizing is needed.
- **Downscaling:** Large images are reduced by removing rows. The number of rows removed depends on how much the image needs to shrink.

By standardizing character sizes, normalization simplifies recognition and improves accuracy.

iv) Image Data Compression

Image data compression reduces the number of bits needed to store or send an image while keeping all the information intact. This helps speed up transmission and lowers storage requirements. Visual data often takes up a lot of space, and although storage devices have large capacities, their speed usually decreases as capacity increases. Compression techniques aim to make images smaller by removing redundant data and reducing the number of samples needed for transmission.

2.4 Applications of Python Tesseract and OpenCV

Python Tesseract and OpenCV are popular tools for processing photos and performing OCR. Python Tesseract focuses on recognizing text, while OpenCV enhances and prepares photos for better OCR results. These tools combine to extract text from photos, such as scanned documents or utility meter photos [15]. OCR involves two steps: first, finding the text in the photo, and second, reading and recognizing it [16].

In AMR systems, these tools extract meter readings from photos. OpenCV improves the quality of the photo, corrects any distortions, and highlights areas that need attention [17]. Python Tesseract then takes care of recognizing the text [1].

Combining these tools makes it possible to create software-based AMR systems that don't rely on expensive hardware. These systems can also integrate with existing platforms for utility management. This proposal shows how Python Tesseract and OpenCV can build efficient and scalable AMR solutions, offering a simple alternative to traditional hardware-based methods.

2.5 Comparative Analysis of Existing AMR Technologies

AMR technologies can be broken down into two categories: hardware-based and software-based. Like smart meters, hardware systems send data directly to utility providers using built-in communication features. They are handy for remote monitoring and can even handle real-time updates [1]. Plus, they're part of the smart grid—a network that links meters wirelessly [1].

The disadvantage of hardware-based systems is their high cost. Installation and maintenance of smart meters can be expensive, making them less practical in some areas. On the other hand, software solutions are more accessible and flexible. They work with devices like smartphones or digital cameras to capture meter readings. This is particularly useful in regions where replacing meters is not easy.

Although software-based systems have potential, most research has been focused on hardware. Software solutions are cheaper and easier to use with existing equipment but present challenges. For example, they require sharp images and good image processing tools to work effectively. This comparison demonstrates the advantages of software-based AMR systems while highlighting the gaps that must be filled.

2.6 Challenges in Manual Meter Reading

Manual meter reading comes with a lot of problems for utility companies. Workers have to visit each meter in person, take the reading, and then manually enter the data into the system. It's a slow and labor-heavy process.

Errors can happen during recording or while entering the data, leading to wrong bills and customer complaints. The costs are high because they require a lot of manpower, and delays in the billing cycle mean companies collect payments later than they should.

Because of these challenges, there's a real need for a better solution. Hardware-based AMR systems help, but they're expensive and tricky to manage. That's why this research focuses on a software-based option. It uses existing technology to automate the process, offering a simpler, cheaper way to solve these problems.



Figure 2.3 Different types of utility meters and pictures taken on different angles

2.7 Advancements in OCR Technologies

OCR has become much better over the past few years, due to advancements in Artificial Intelligence and Machine Learning. Before that, OCR systems struggled with blurred images, tilted text, and unusual fonts because they had a specific set of instructions to follow for reading the text.

Today, OCR is improving and using intelligent technologies such as deep learning that help it recognize text more accurately[18]. Tools such as Google Vision API and EasyOCR use convolutional neural networks to identify text patterns that work even on low-quality images.

Modern OCR-based systems also use advanced models such as Long Short-Term Memory(LSTM) networks. For example, Tesseract uses LSTM to more accurately read handwriting and merge letters.

OCR can now work on smartphones. This allows features like instant translation or augmented reality, making OCR tools more useful and accessible.

2.8 Emerging Trends in AMR Systems

Trends in AMR technologies include the adoption of advanced communication protocols, the integration with Internet of Things (IoT) solutions for real-time data collection, and the use of AI-driven analytics to optimize energy consumption and to improve grid management[19].

AMR systems are becoming more advanced with new technologies. One key change is the use of IoT devices that help utility meters send real-time data directly to servers. This means meters can be monitored and controlled remotely, so there is less need for manual checks.

The use of machine learning in AMR systems can predict energy usage patterns and spot problems like leaks or faulty devices by analyzing unusual spikes in

consumption. Blockchain is also being tested to make energy transactions more secure and transparent, especially in systems where energy is shared between users.

AMR systems are also connected to smart grids. These grids adjust the energy supply based on how much energy people are using at any moment, which helps to save energy and to ensure it is distributed efficiently. These changes show how AMR plays a big role in making energy management smarter and more sustainable.

3 System Design and Implementation

This chapter describes the design and architecture of the automatic meter reading system. The system is focused on uploading utility meter images, applying OCR with Python Tesseract, and using OpenCV for image processing. After enhancing the images, the meter readings are taken, and the electricity bills are calculated. Figure 3.0 shows the design of the system.

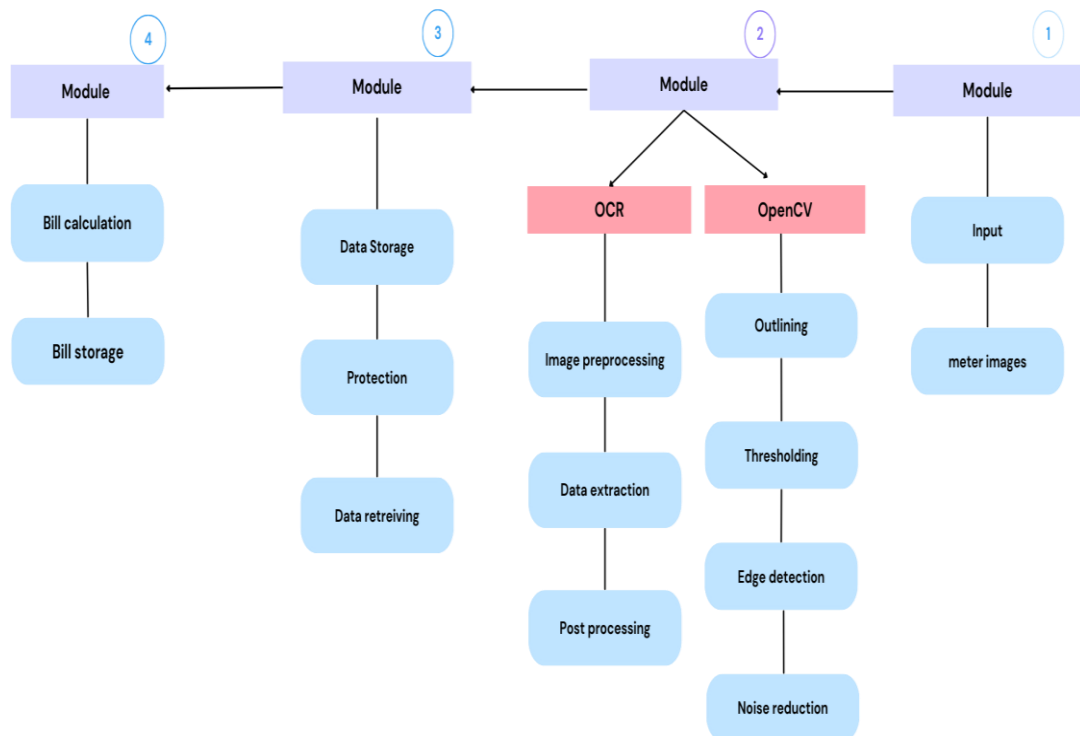


Figure 3.0 Design of the system

3.1 Challenges in Manual Meter Reading

The AMR framework needs hardware and software to process utility meter photos accurately and efficiently.

3.1.1 Hardware components:

On the hardware side, a specified device with camera, for example, a cell phone or a dedicated meter scanner, is essential for capturing high-quality images of the meter readings. The high-quality images are required for clarity and accuracy to enable reliable processing. A PC or server with at least 8 GB RAM and a multi-core processor is also required to deal with the computational demands of the image processing tasks. A strong system is essential to manage databases that store customer details, meter readings, and transaction records.

3.1.2 Software requirements:

On the software side, Python is used as the primary programming language because of its extensive libraries and support for AI and image-processing tasks. The key libraries include OpenCV for image processing and Tesseract for OCR functionality. An operating system within the system is required e.g. Windows, Mac, or Linux operating system. For database management, MySQL or SQLite is used to save, retrieve and manage data.

3.2 Overall System Architecture

The AMR system design consists of different parts that work together seamlessly. It starts with the image upload module, where employees upload photos of electricity meters. These photos are processed by the OCR and image processing module, which extracts the meter readings. The data is then sent to the data storage and management module, where it is saved and organized in a database. Finally, the bill calculation module uses this data to generate and present bills to the customers.

The system is designed to be flexible, allowing updates or fixes to one part without disrupting the others. Below are the main modules of the system.

- **Module 1:** This module is utilized to upload pictures.

- **Module 2:** This module handles OCR and picture handling by analyzing and managing obtained pictures.
- **Module 3:** This module stores and tracks expenses and meter readings.
- **Module 4:** This module uses extracted information and algorithms or provided formulas to calculate the cost of electricity.

3.2.1 Image Upload Module

This module allows employees to upload photos of utility meters for processing and data extraction. It acts as the system's starting point, where employees interact with the interface to upload photos. The module is designed to be user-friendly, making it simple to add photos of utility meters.

The system reviews each uploaded photo to make sure it is clear and properly aligned before moving on. Supported formats, like JPEG and PNG, are processed further to get details such as timestamps and unique meter IDs. If a photo doesn't meet the quality standards, the module alerts employees with an error message and provides tips to correct it. This helps ensure that only usable photos move on to the next processing stage. The steps in this module are as follows:

- Consistent image upload functionality.
- Accurate image processing by working on image quality.
- Correspondence with the image processing module for further analysis.

3.3 OCR and Image Processing Module

Image processing is fundamental to assessing and evaluating utility meter pictures in software-based AMR systems. There are two aspects to the process of reading an electric meter. One is a different angle problem, which means poor displays localization effect owing to the variations of camera position(level), shot lengths, and angles[20].

The OCR and image processing module is an important part of the AMR system. It takes photos of utility meters, finds the digital readings, and converts them into a format the system can use. This module is designed to handle problems like low photo quality, shadows, reflections, and uneven surfaces, ensuring accurate results.

It uses two tools: OpenCV and Python Tesseract. OpenCV cleans up the photos and separates the meter readings while Tesseract reads the numbers. These tools work together to turn the readings into usable data. The module has two main steps:

3.3.1 Python Tesseract Implementation

Tesseract OCR is used to recognize text from photos. As an open-source library, it is highly flexible and can read alphanumeric characters from images. For utility meters, where only numbers are needed, Tesseract has been adjusted to focus on recognizing numeric information efficiently.

i) Preprocessing for OCR:

During this step, preprocessing is done for better accuracy in text recognition. It starts by converting the whole image into a grayscale in order to reduce the complexity of the processing by removing a variety of information, for example, color data. Then, adaptive thresholding is applied to produce a binary image where the figures are sharply separated from the background. This step is quite effective if the image is captured at low light or variable illumination.

ii) Tesseract Design:

Tesseract framework design includes parameters that can be customized such as the Page Segmentation Mode that can be adjusted to optimize OCR for specific layouts like single lines or blocks of text[18].

Tesseract's setup includes selecting the right paging mode (PSM). For instance, the system uses `--psm 7`, which is designed to read text in a single line, like the

row of numbers on a utility meter. Custom training data is also used to improve Tesseract's accuracy. This data helps the system recognize the specific patterns of numbers found on meter readings and reduces errors caused by similar-looking characters.

iii) Post-Processing:

Post-processing in Tesseract includes techniques such as spell checking and contextual analysis to refine recognized text and reduce errors in the final output[18]. The Tesseract result often contains interference or misrecognized characters. Post-processing channels are aimed at cleaning and validating the extracted information. For example, the system checks that the identified characters match the expected numeric patterns (e.g. numbers only should be recognized between 0 and 9) and removes any random images. Figure 3.1 illustrates the process of Data extraction.

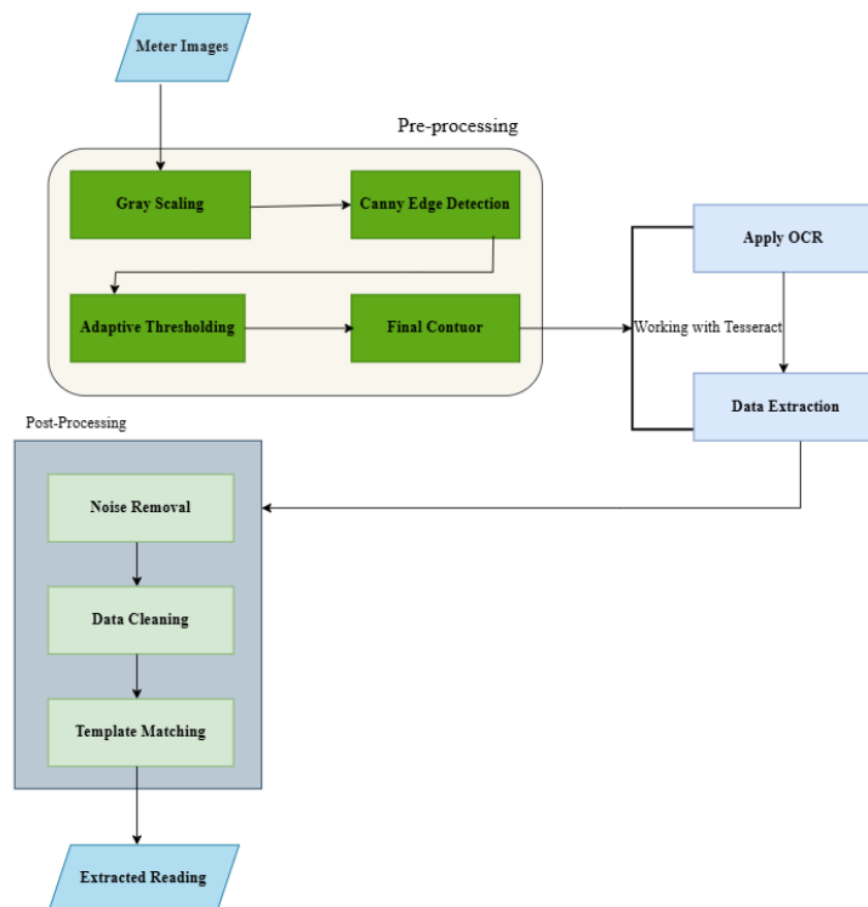


Figure 3.1: Python Tesseract techniques

3.3.2 OpenCV Techniques

The OpenCV library is designed to enhance the quality of the images, which are then passed to the OCR engine for further processing. OpenCV makes data extraction easier by offering tools like edge detection, thresholding, and contour analysis. These tools improve image quality, making it ready for accurate text recognition [21]. Different methods are applied to enhance the images and are given below:

Outline Extraction: This method is used to detect the borders, making it easier to isolate the numbers.

Image Threshold: This strategy helped segment the image and focus on areas with valuable information, such as meter reading numbers.

Edge detection: This technique focuses on determining the edges of the numbers.

Noise reduction: These filtering techniques remove unwanted objects from images that can slow down OCR accuracy.

OpenCV plays an important role in this module; it provides the tools required for accurate image analysis and control. The use of OpenCV starts with image enhancement and reaches out to digit segmentation, enabling the system to deal with different meter plans and challenging image conditions.

i) Image Enhancement and Noise Reduction:

OpenCV configures the image for digit recognition using the following optimization strategies: smoothing the Gaussian mask for noise reduction that may interfere with the OCR and leveling histograms to develop more contrast between figures and background.

Gaussian Blur is used to smooth images by reducing noise and detail, which helps improve the accuracy of feature detection and preprocessing in computer vision tasks[21].

ii) Region of Interest (ROI) Detection:

Region of Interest (ROI) detection is a technique that focuses on isolating specific parts of an image, such as text or objects, for targeted processing, improving accuracy and reducing computational overhead[21]. The system focuses on the meter reading area by identifying the region of interest (ROI). For this purpose, it uses edge detection algorithms, like Canny, to outline the contours of the display panel. After detecting the edges, contour approximation techniques are applied to locate rectangular sections that likely contain the meter reading. Once the ROI is identified, it is sent for further processing in the next stages.

iii) Digit Segmentation:

Digit segmentation separates each digit in an image by detecting their outlines with contour detection and placing them in bounding boxes. This helps OCR recognize the digits more accurately [18]. After ROI detection, the system segments individual digits for separate processing. Contour detection is utilized to identify and isolate every digit within the reading area. Every digit is isolated as a sub-picture, resized to a uniform scale, and sent to the OCR engine afterward. This ensures that even distorted or irregularly spaced digits are recognized precisely.

iv) Template Matching for validation:

Template matching is a technique in computer vision where a predefined template is compared against an input image to find matching regions commonly used for object or character recognition[21]. In addition, with OCR, the system integrates template matching as secondary validation. Pre-stored templates of standard digit shapes are compared to the segmented digits using the Match Template function of OpenCV. This approach identifies errors in the OCR results, especially for blurred or partially obscured images. If the OCR result differs radically from the corresponding rendering results, the system flags the reading for further review. Figure 3.2 demonstrates the working of OpenCV.

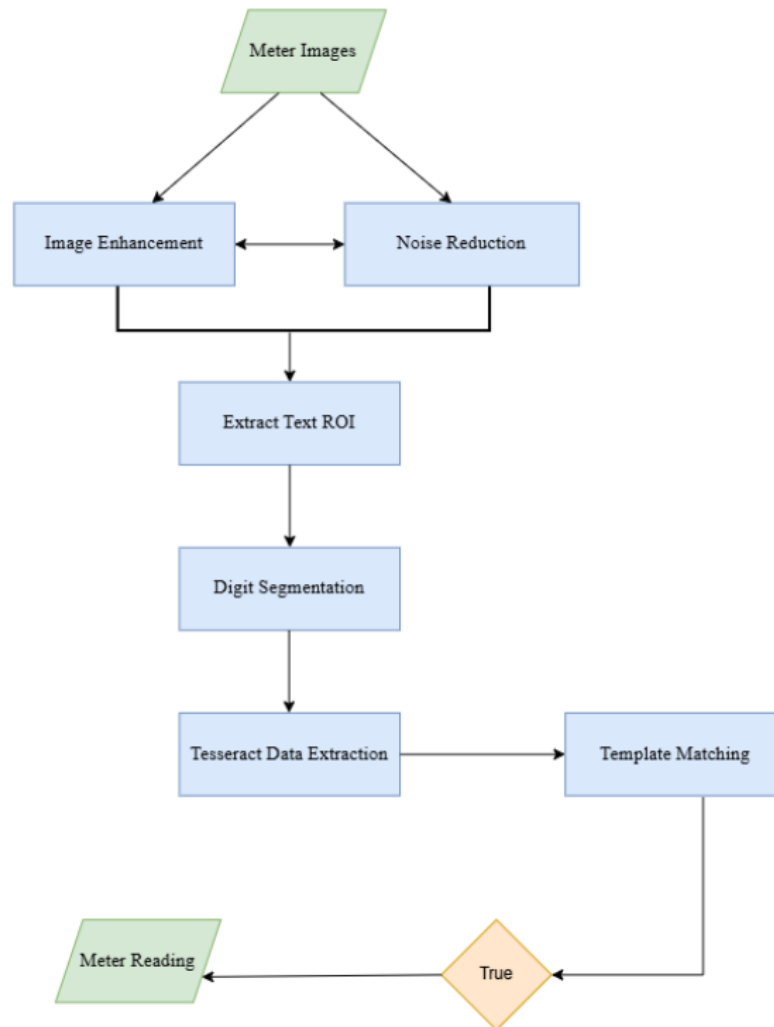


Figure 3.2: flow process of OpenCV

v) Dealing with complex scenarios:

OpenCV techniques are especially useful for handling complex situations, such as:

Reflections on glass panes: Adaptive thresholding dynamically calculates thresholds for smaller regions in an image, enabling better segmentation under uneven lighting conditions[22]. Adaptive thresholding and basic adjustments like zooming and cropping are used to reduce reflections.

Distorted or rotated images: Perspective correction is applied to fix tilted or distorted photos, ensuring the numbers are properly aligned.

Partial occlusion: If parts of the numbers are blocked, edge enhancement and contour tracing help fill in the missing sections.

3.4 Integration of OCR and Image Processing

This module combines OCR and image processing to work together. OpenCV prepares the photos so Tesseract can read the text properly. After Tesseract processes the numbers, OpenCV checks the results and adjusts to improve accuracy.

Using OCR with image processing makes text extraction more efficient, improving both accuracy and reliability in recognizing data [23]. The system works well using both tools, even with poor-quality photos or bad lighting. It's a reliable solution for automated meter reading.

The setup is also flexible for the future. It can handle new types of meters or add features like better error correction. Figure 3.3 shows how the system works.

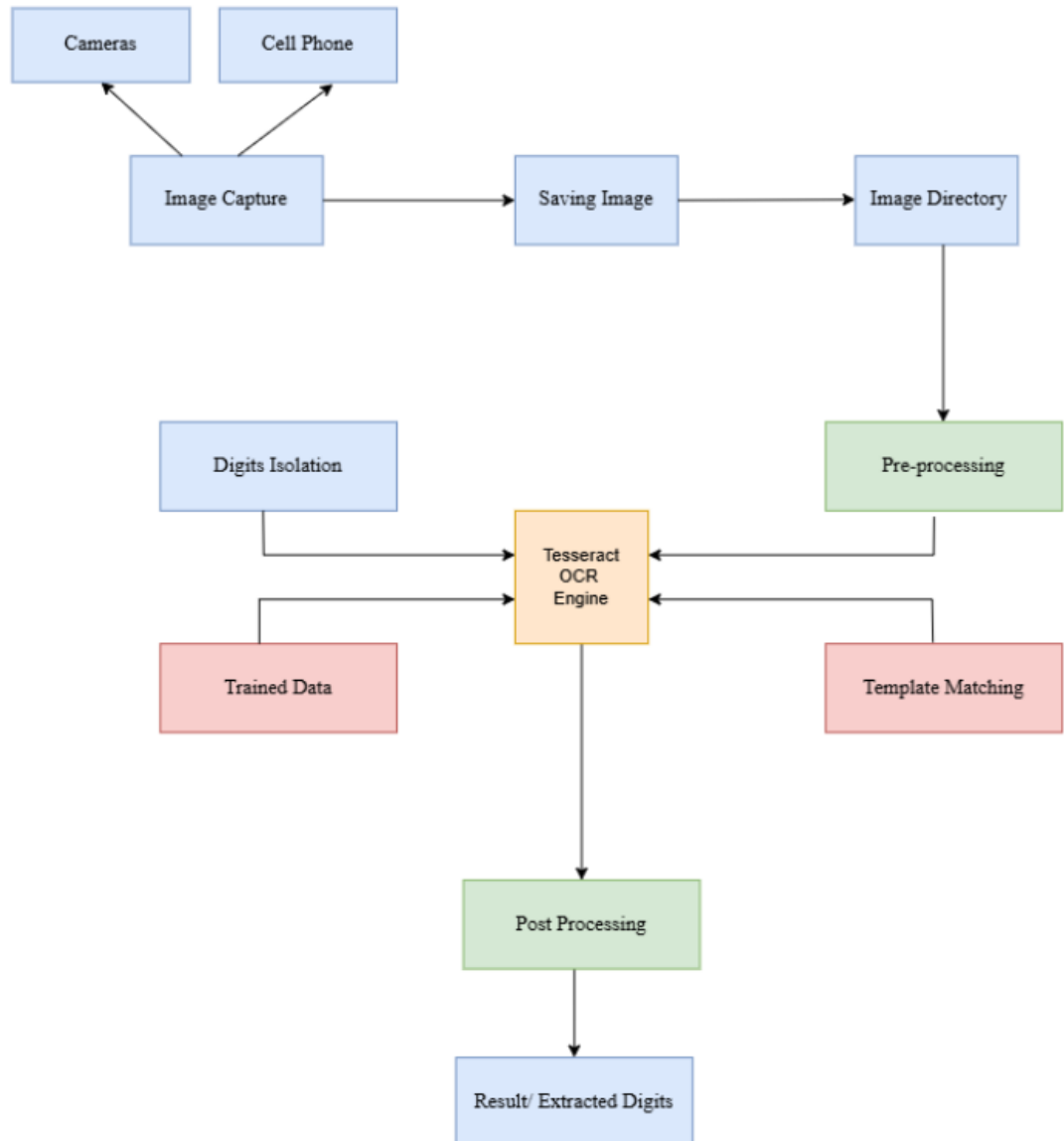


Figure 3.3 Flow process of the developed system

3.5 Data Storage and Management

The database is an important part of the AMR system. It keeps raw photos, processed readings, customer details, and billing reports. It is built to handle more data and customers as the system grows without slowing down.

To keep data safe, it has backup and recovery options. Fast search tools make finding data for audits, analysis, or research easy. The database helps with tasks like:

- Save data (extracted reading) with image reference and location.
- Data protection against unauthorized access.
- Optimize databases to manage large numbers of meter readings efficiently.
- Simplify billing calculations by providing an interface to retrieve stored statements.

3.6 Bill Calculation Module

The bill calculation module changes the processed data into meaningful outcomes. Using the provided formulas, it takes the data set's current and past meter readings to record energy utilization. The determined consumption is then increased by the material tax rate to determine the total cost. It supports dynamic rate structures whereby utility companies can update the rates based on the weather season of the year when energy is used, energy demand, or customer classification. The final bill details include the customer meter ID, billing period, amount of energy consumed, and the total cost in a reasonable and concise format. The calculation formulas in this module are publicly available, hence helping to improve trust between users and energy providers. The module performs the following tasks:

- Get the meter reading from the database.
- Calculate the invoice by applying the characterized unit costs and calculations/equations.
- Apply additional fees or taxes to calculate the final bill.
- Follow the receipt amount defined in the database.

4 Testing, Evaluation, and System Validation

4.1 Testing Methodology

For extracting meter reading digits, Tesseract OCR is used. To ensure consistent results, careful image preprocessing is essential. An adaptive preprocessing approach was adopted since the dataset includes photos of varying sizes, shapes, and backgrounds. This step improves the quality of the photos before they are processed by the OCR.

First, preprocessing starts with resizing the images and converting them into grayscales to reduce noise and edge detection efficiently. This transformation combined the red, green, and blue (RGB) channels into a single intensity channel using the formula:

$$\text{Gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

Hence, Gaussian blur and adaptive thresholding are used in this context to reduce interference with the background, thereby enhancing edges. The "resize" parameter is tuned to standardize the dimensions of the images, while binary thresholding highlights the digits at their best. The following functions are used for this purpose:

$$\text{blurred} = \text{cv2.GaussianBlur}(\text{gray}, (5, 5), 0)$$

$$\text{thresh} = \text{cv2.adaptiveThreshold}(\text{blurred}, 255, \text{cv2.ADAPTIVE_THRESH_GAUSSIAN_C}, \text{cv2.THRESH_BINARY}, 11, 2)$$

Gaussian Blur:

- This step smooths the image to remove small details and noise, making it easier to process.

Adaptive Thresholding:

- Instead of using one rule for the whole image, this method looks at small areas (11x11 pixels) and sets a brightness limit for each one.
- It calculates the average brightness in the area and subtracts 2 to decide the threshold.
- Pixels brighter than this are turned white, and darker ones are turned black.

This method works well for images with uneven lighting, ensuring the text stands out clearly. Figures 4.0, 4.1, and 4.2 demonstrate the image preprocessing steps.

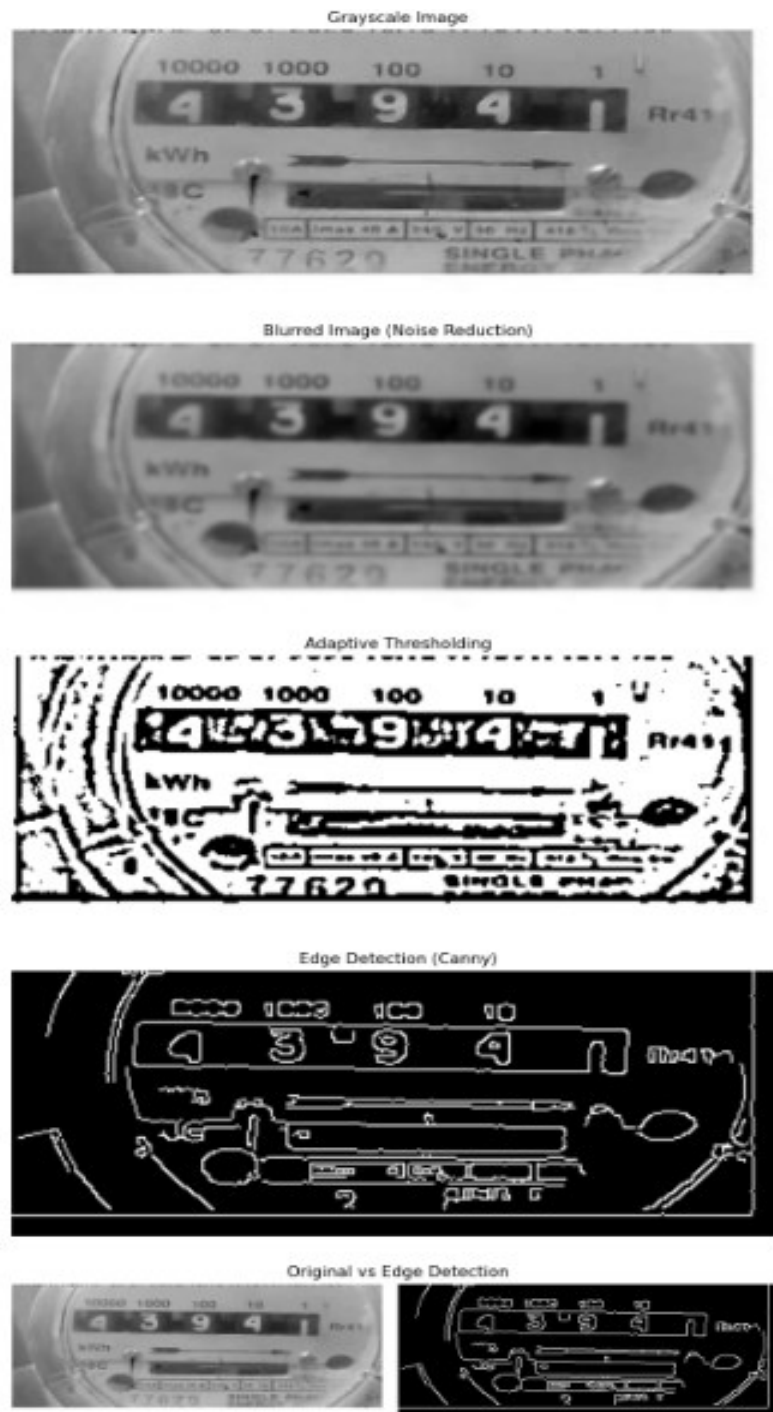


Figure 4.0 Preprocessing techniques for image enhancement

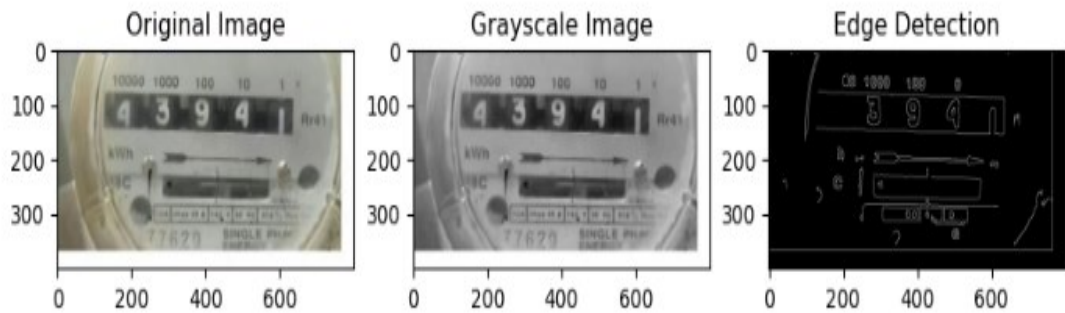


Figure 4.1 Working with OpenCV

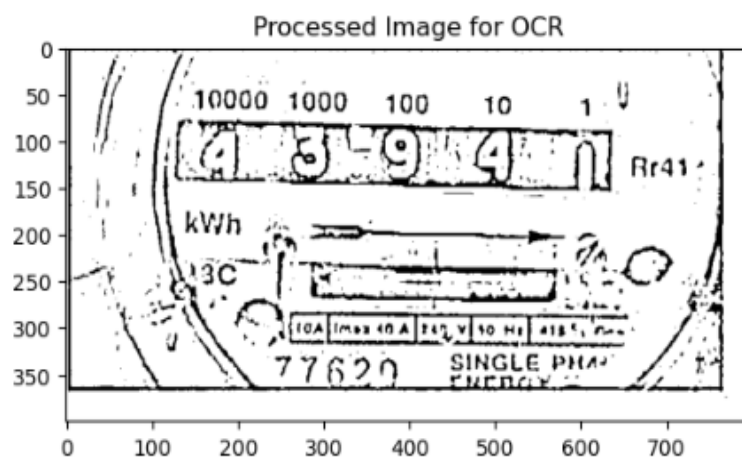


Figure 4.2 Processed image for further analysis

OCR and Tesseract setup with the preprocessed images: The setup of Tesseract is configured to focus only on recognizing numbers by using the parameter:

```
tessedit_char_whitelist=0123456789
```

This ensures that only digits are processed. Additionally, the mode `--psm 7` is used, which helps Tesseract read single lines of text more accurately. This is especially useful for meter readings where the digits might not always be perfectly aligned. Figure 4.3 shows how the meter reading is extracted. Figure 4.3 demonstrates the process of extracting meter readings.

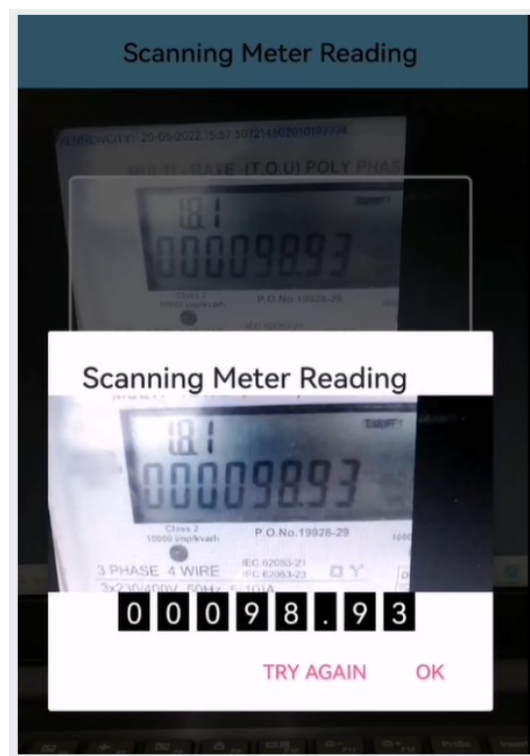


Figure 4.3 Data extraction from meter image

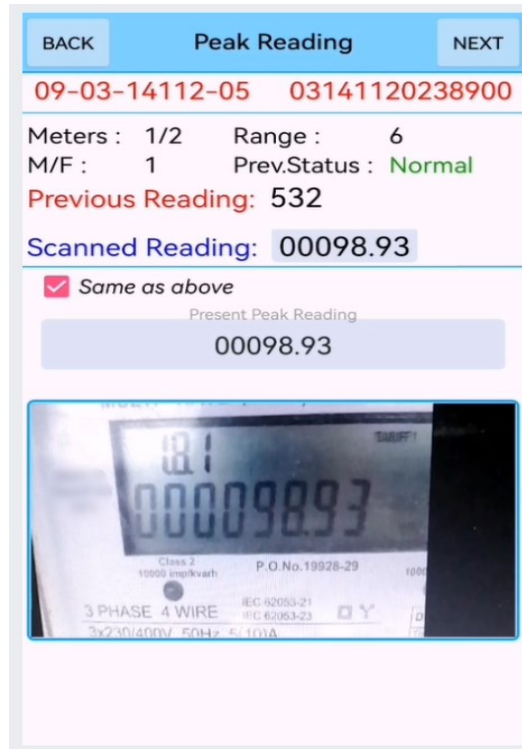
4.2 Result Analysis and System Validation

The system was tested under different conditions and showed excellent performance, accurately reading photos of utility meter readings. Image preprocessing steps like grayscale conversion and adaptive thresholding helped focus on the relevant data, ensuring accuracy during processing.

Using OpenCV for pattern matching, the system successfully identified and extracted numbers even when faced with challenges like varying counter designs, font styles, and image clarity. The results demonstrated a high success rate in recognizing numbers, proving the system's accuracy and reliability for real-world use. Figures 4.4 a and b show how the system extracts meter readings and saves the data.



(a)



(b)

Figure 4.4 (a),(b) Meter reading scanning and saving data

The system tries to make data as accurate and reliable as possible by storing it organized for easy access and verification. However, errors can still happen, and manual checks might be needed, especially at the start. Each entry is processed to improve accuracy, making it useful for real-time tasks and future audits. The storage system also allows quick access to current and historical data, supporting traceability and authentication. Figures 4.5 a and b show how data is stored and retrieved.

| Reference No. | Meter No. |
|--------------------------------------|-----------|
| <input type="radio"/> 03141120238900 | 11043 |
| <input type="radio"/> 03141120239000 | 167783 |
| <input type="radio"/> 03141120239100 | 9053 |
| <input type="radio"/> 03141120239200 | 167189 |
| <input type="radio"/> 03141120239300 | 167479 |
| <input type="radio"/> 03141120239400 | 167456 |
| <input type="radio"/> 03141120239500 | 166994 |
| <input type="radio"/> 03141120239600 | 167846 |
| <input type="radio"/> 03141120239700 | 293583 |
| <input type="radio"/> 03141120239800 | 168158 |
| <input type="radio"/> 03141120239900 | 293352 |
| <input type="radio"/> 03141120240000 | 167327 |
| <input type="radio"/> 03141120240100 | 167401 |
| <input type="radio"/> 03141120240200 | 17263 |
| <input type="radio"/> 03141120240300 | 13906 |
| <input type="radio"/> 03141120240400 | 168276 |

(a)

08-01-13125-01 527/527

| Reference No. | Meter No. |
|---|-----------|
| <input type="radio"/> 01131250000031 | 5835 |
| <input checked="" type="radio"/> 01131250000081 | 471609 |
| <input type="radio"/> 01131250000131 | 6662683 |
| <input type="radio"/> 01131250000181 | 2696028 |
| <input type="radio"/> 01131250000231 | 2696030 |
| <input type="radio"/> 01131250000240 | 7050005 |
| <input type="radio"/> 01131250000291 | 2678112 |
| <input type="radio"/> 01131250000631 | 2695716 |
| <input type="radio"/> 01131250000640 | 784474 |
| <input type="radio"/> 01131250000681 | 376461 |
| <input type="radio"/> 01131250000731 | 2696024 |
| <input type="radio"/> 01131250000748 | 1290582 |
| <input type="radio"/> 01131250000781 | 693821 |
| <input type="radio"/> 01131250000798 | 125031 |

Reading Date: 03-08-2024 Not Arrived.
Please Wait!

Billing Month : 202408
 Days in Month 202407 : 31
 Prev. Reading Date : 20240703
 Current Date : 20240802
 Days Passed : 30
 Days Remaining : 1

OK

(b)

Figure 4.5, (a) data storage with different parameters, (b) data retrieval

Meter information and all the users' data are stored in the system, allowing for tracking various parameters. This data not only supports bill calculations but also helps improve the system's overall performance. Figure 4.6 displays the stored user and utility meter information.

| Sr No. | Proposed TF REF NO | T/F CAPACITY | TF CODE | Meter No. | Location | Readings 24-08-2024 | MF | TL-20-09-24 | off peak | peak |
|--------|--------------------|--------------|---------|------------|----------|---------------------|-----|-------------|----------|---------|
| 1 | | 100-KVA | | 715105 | | 6337.50 | 40 | Opened | DEF | DEF |
| 2 | | 200-KVA | | 00715368 | | 2817.81 | 160 | 3310.33 | 2751.11 | 559.22 |
| 3 | | 75-KVA | | 00715313 | | 12807.88 | 160 | 12844.6 | 10026.34 | 2818.26 |
| 4 | | 200-KVA | | 00714647 | | 8360.39 | 160 | 8764.02 | 6560.06 | 2203.97 |
| 5 | | 100-KVA | | 00715281 | | 6737.41 | 40 | 7075.59 | 5824.49 | 1251.1 |
| 6 | | 200-KVA | | 2999001792 | | 3303.98 | 160 | 3508.04 | 3014.93 | 493.11 |
| 7 | | 200-KVA | | 40341 | | 14297.18 | 160 | 14644.7 | 12427.27 | 2217.42 |
| 8 | | 200-KVA | | 224559 | | 1121.30 | 160 | 1206.52 | 1004.86 | 201.65 |
| 9 | | 200-KVA | | 224606 | | 1478.79 | 160 | 1622.6 | 1434.39 | 188.21 |
| 10 | | 100-KVA | | 00717838 | | 933.71 | 160 | 1041.98 | 857.35 | 184.63 |
| 11 | | 200-KVA | | 40548 | | 2057.79 | 160 | 2161.22 | 1818.5 | 342.72 |
| 12 | | 200-KVA | | 027474 | | 1373.40 | 160 | 1616.18 | 1295.77 | 320.41 |
| 13 | | 200-KVA | | 105972 | | 7850.27 | 160 | 8150.93 | 5791.51 | 2359.42 |
| 14 | | 200-KVA | | 00715049 | | 3590.81 | 160 | 3885.34 | 3333.84 | 551.51 |

Figure 4.6 Storing of information into the system

The system equally allows employees to easily retrieve, update, and view stored data, thus making the system very flexible and user-friendly. The system organizes data to allow quick access and increase the other operations' efficiency. Figure 4.7 (a),(b) shows the different options for meter reading and information related to specific meters.

| 09-03-14112-05 | | PEAK FIRST |
|---|--------------|----------------------|
| 03141120238900 | | |
| D.D.(E) CDA, SUIT NO.404 3RD FLOOR BLOCK-E PARLIMENT LODGES, IBD | | |
| Mobile No. | 000000000000 | EDIT |
| Previous Status | Normal | |
| Meter No. | 11043 | |
| Meter Make | Escorts | |
| Total Meters | 2 | |
| Feeder Code | 75009 | |
| Sanction Load | 14.02 | |
| Tariff Code | 03 | |
| Prev Reading: | 532 3114 | |

(a)

| 09-03-14112-05 | 03141120238900 |
|----------------------------------|------------------------------|
| <input type="radio"/> | 101-Excess/Over Billed |
| <input type="radio"/> | 102-Meter Reversed/Tempered |
| <input type="radio"/> | 103-Digits Running Reverse |
| <input type="radio"/> | 104-Round Complete |
| <input type="radio"/> | 105-Meter Replaced |
| <input type="radio"/> | 106-RCO with Different Meter |
| <input checked="" type="radio"/> | 107-Normal/Seasonal Effect |

(b)

Figure 4.7: different options related to meter reading and information related to specific meters.

4.3 Advantages of the Developed System

The system makes it much easier to manage meter readings. Employees still need to visit the site to take pictures of the meters and upload them to the system. However, the system eliminates the need for someone else to manually enter the statements, saving time and simplifying the process.

After the images are uploaded, the system automatically reads the meter reading numbers. This reduces errors and ensures billing improved accuracy. It also makes things smoother and less stressful for all participants.

The system is also flexible. It can work with different types of meters, even those that look or function differently. Simple adjustments are enough to make it usable in other areas or setups.

Another significant advantage is how fast it processes data. Because the meter reading part is mainly automated, the system can handle more readings in the same amount of time. This means utility companies can scale up their operations without hiring more staff.

Finally, all the data is stored digitally. It's easy to find, check, or use for audits or reports. This system offers a reliable and efficient way to manage utility data.

4.4 Comparative Cost-Benefit Analysis

The developed system makes meter reading smoother. The employees will still need to travel to each site to take photos of the meters, but they do not need to manually enter the reading data into the system. In the old method, employees spent extra time entering data after returning from the field. Now, with the AMR system, they upload the images, and the system automatically extracts the readings. This saves a lot of time.

As an estimate by the company, reading and entering data for 100 meters used to take 10–12 hours, including travel and data entry. But now, it takes only 6–7 hours because manual data entry is no longer needed. The system also reduces errors from manual input, meaning less time is spent fixing mistakes. While travel time stays the same, the process is much quicker and more efficient.

| Comparative Cost-Benefit Analysis of Existing and Proposed System | | |
|---|--|---|
| Aspects | Existing System | Proposed System |
| Labour Costs | High | Significantly Reduced |
| Error Rates | High Probability | Low Probability |
| Data Insertion | Long (Travel, Photo upload and Data Entry) | Much Faster (Travel, image upload only) |
| Billing Accuracy | Medium | High |

Figure 4.8 Comparative Cost-Benefit Analysis of Systems

4.5 Comparative Performance Analysis

The new AMR system was tested to see how it performed compared to older methods. The focus was on three main elements: its accuracy, speed of operation, and the number of errors it made.

The system could read most of the time correctly, way better than the manual methods where errors are common. It also took only 5-10 seconds to process reading from a photo, and it saved a lot of time compared to manual data entry.

4.6 Limitations, Challenges, and Potential Improvements

The system works with the help of OCR, which is highly dependent on the image quality. The varying image quality, poor lighting, different angles, and other factors may affect the working of OCR and may result in wrong readings.

Additionally, OCR technology is not always accurate. The system may also struggle due to meter types because different meters have different digit patterns, font sizes, and shapes. The system may struggle with analog meters with unclear digit arrangements and depend on cameras. If the camera performance is poor, it can lead to ambiguous results.

Advanced image enhancement techniques can help to handle the problem of image quality. Advanced machine learning models trained on various meter types can work with all. More advanced machine learning techniques can be incorporated so that data extraction is always correct. Using the custom algorithm with pointer detection can enhance the recognition of analog meter digits.

5 Conclusion and Future Work

5.1 Summary of Findings

The system developed for automated utility meter reading can extract the meters' readings from images of the meters with minimal human intervention. The main conclusions are:

OCR Accuracy: The system can reasonably recognize numbers from clear, high-quality digital utility meter images. However, performance varied significantly depending on image quality, meter design, and lighting conditions.

Image Improvement: Image preprocessing techniques such as threshold, dilation, and contrast adjustment improved image clarity, allowing for more accurate OCR results.

Automated process: The developed system has automated the process of reading and extracting meter readings, thus reducing manual work and the risk of errors in meter readings.

Challenges and limitations: Although the system successfully automates the process, it still faces some challenges. Environmental issues like poor lighting and blurry photos can reduce the accuracy of OCR technology and affect how well the readings are extracted.

5.2 Recommendations for Future Research

Although the developed system offers considerable accuracy, more research is needed to address its limitations and expand its applicability.

Future research could explore more sophisticated image processing methods, such as image enhancement based on deep learning or real-time reading

capturing by the network-based camera, to improve the quality of input images, especially in harsh environments such as dim or bright light.

Integrating machine learning models to train the system to recognize specific meter types and configurations can improve system adaptability and accuracy. The system can also learn from historical data to improve recognition accuracy.

More research into developing algorithms to automatically detect errors in readings taken (e.g., flagging abnormal readings based on past data) will improve system reliability. Outlier detection can help detect incorrect readings before they are processed further.

Hybrid systems for analog and digital meters: A hybrid approach that supports analog and digital meters largely expands the system's utility. Analog meter reading presents a unique challenge because its mechanical displays and the development methods for both will make the system more universally applicable.

Development of mobile application versions of the system would allow the company's staff to directly use the device's camera application to scan meters and send readings without the extra step of first uploading the picture to the system directory.

References

- [1] Nayaka RJ, Biradar RC. A survey on wireless network applications in automated public utilities control and management. J Telecommunication and Information Technology. 2015;(3):13-24.
- [2] Huang JH, Chen YH, Tsai YL. Utilizing cross-ratios for the detection and correction of missing digits in instrument digit recognition. Mathematics. 2024;12(11):1669. DOI: 10.3390/math12111669.
- [3] Hawkins R. Promoting the digital literacy of undergraduate historians using digitized historic newspapers. 2016
- [4] Arindam Chaudhary, Krupa Mandaviya, Pratixa Badelia and Soumya K Ghosh, "Optical Character Recognition Systems for Different Languages with Soft Computing", Springer International Publishing AG, Picassoplatz 4, CH-4052 Basel, Switzerland,2017.
- [5] Jayanta Kumar Basu, Debnath Bhattacharyya, Tai-hoon Kim,"Use of Artificial Neural Network in Pattern Recognition", International Journal of Software Engineering and its Application, Vol. 4, No. 2, April 2010.
- [6] Govindaraju, V., & Srihari, S. N. (2009). "Advances in Document Recognition." Proceedings of the IEEE, 96(4), 678-686.
- [7] Mori, S., Suen, C. Y., & Yamamoto, K. (1999). "Historical review of OCR research and development." Proceedings of the IEEE, 80(7), 1029-1058.
- [8] AI Bovik,"The Essential Guide to Image Processing", Academic Press Elsevier, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA.
- [9] Anil K. Jain,"Fundamentals of Digital Image Processing", Prentice-Hall of India Private Limited, New Delhi- 110 001, 1997.
- [10] Rohit Verma and Dr. Jalid Ali," A comparative study of various techniques of image noise and efficient noise removal techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, October 2013.

- [11] Yide Ma, Kun Zhan, Zhaobin Wang, "Application of Pulse-Coupled Neural Networks ", Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg, 2010
- [12] Komal Bishnoi, Preeti Gupta, "A Practical Approach of Parallel Thinning Algorithm on Numeral Patterns Using Matlab Simulation", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 5, May 2016
- [13] Stack Overflow. (n.d.). How do you fit a curve to a skeletonized image?. [online] Available at : <https://stackoverflow.com/questions/21868132/how-do-you-%20fit-a-curve-to-a-skeletonizedimage> [Accessed 28 Nov 2024].
- [14] Rajvardhan Oak, "A Study of Digital Image Segmentation Techniques", International Journal of Engineering And Computer Science, Volume 5 Issue 12 Dec. 2016
- [15] Rekha M, Devi PS, Laxmi UV, Sirisha P, Karthika M. Educational training for processing invoice of vendor identification and payments using Python-Tesseract. Turk J Computer Math Educ. 2021;12(11):224-8.
- [16] Bradski G. The OpenCV library. Dr Dobb's J Software Tools. 2000;25(11):120-3.
- [17] Mahamkali N, Vadivel A. OpenCV for computer vision applications. In: Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15); 2015.
- [18] Smith, R. (2007). "An Overview of the Tesseract OCR Engine." *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*. IEEE.
- [19] International Energy Agency (IEA). (2021). *Digitalization and Energy*. IEA Publications.
- [20] Canny J. A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell. 1986;PAMI-8(6):679-98. DOI: 10.1109/TPAMI.1986.4767851.
- [21] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.

- [21] Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing*. Prentice Hall.

- [22] Bradley, D., & Roth, G. (2007). "Adaptive Thresholding Using the Integral Image." *Journal of Graphics Tools*, 12(2), 13-21.

- [23] Shapiro, L. G., & Stockman, G. C. (2001). *Computer Vision*. Prentice Hall.