FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automation of Cable and Connection Lists in Substation Projects

Diogo Viamonte Santos da Costa Fernandes



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Doutor José Eduardo Roque Neves dos Santos

March 27, 2026

© Diogo Viamonte Fernandes, 2026

Automation of Cable and Connection Lists in Substation Projects

Diogo Viamonte Santos da Costa Fernandes

Mestrado em Engenharia Eletrotécnica e de Computadores

Approved in oral examination by the committee:

President: Prof. José Rui da Rocha Pinto Ferreira
Referee: Prof. Sérgio Filipe Carvalho Ramos
Referee: Prof. José Eduardo Roque Neves dos Santos

March 27, 2026

Resumo

A produção de listas de cabos e de ligações a partir de desenhos de subestações elétricas é manual, demorada e propensa a erros, com impacto na segurança e na conformidade com a regulamentação. As normas E-REDES exigem nomenclatura estruturada e rastreabilidade ao nível dos condutores; a Bragalux opera neste contexto e foi o parceiro industrial do trabalho.

Esta tese descreve um pipeline baseado em PDF para extração de terminais e apoio à nomenclatura de cabos. O desenho em CAD é exportado para PDF, preservando a informação vetorial; a extração de texto e a validação em várias etapas — reconhecimento de padrões, filtragem de conteúdo, validação espacial e atribuição de réguas — produzem listas de terminais em formato E-REDES. A conversão CAD-to-PDF e o processamento em batch permitem analisar projetos sem bloquear o uso do CAD.

A extração de terminais foi validada num conjunto piloto e num modelo de projeto E-REDES completo com múltiplos painéis, mediante comparação com dados de referência e métricas de precisão e recall. Os resultados indicam precisão adequada à produção para a extração de terminais, que pode assim apoiar a revisão por técnicos. A nomenclatura de cabos fica limitada pela fragmentação das rotas na exportação para PDF, que quebra a continuidade geométrica necessária ao emparelhamento rota–terminal.

O âmbito cobre a documentação de subestações E-REDES e o limite entre o que é automatizável com extração geométrica e o que exige abordagens semânticas ou nativas em CAD; essa fronteira é discutida nas conclusões.

Abstract

The production of cable and connection lists from electrical substation drawings is manual, time-consuming, and error-prone, with implications for safety and regulatory compliance. E-REDES standards require structured naming and conductor-level traceability; Bragalux operates in this context and was the industrial partner for this work.

This thesis describes a PDF-based pipeline for terminal extraction and cable naming support. CAD drawings are exported to PDF, preserving vector information; text extraction and multi-stage validation — pattern recognition, content filtering, spatial validation, and ruler assignment — produce terminal lists in E-REDES format. CAD-to-PDF conversion and batch processing allow projects to be analysed without blocking CAD use.

Terminal extraction was validated on a pilot dataset and on a full E-REDES project template across multiple panels, using reference data and precision and recall metrics. Results indicate production-ready accuracy for terminal extraction, which can therefore support technician review. Cable naming remains limited by route fragmentation in PDF export, which breaks the geometric continuity needed for route–terminal matching.

The scope is E-REDES substation documentation and the boundary between what can be automated with geometric extraction and what requires semantic or CAD-native approaches; that boundary is discussed in the conclusions.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework for achieving a better and more sustainable future for all. It includes 17 goals[1] to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis.

The specific Sustainable Development Goals mentioned have the following names:

SDG 7 Ensure access to affordable, reliable, sustainable and modern energy for all

SDG 8 Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

SGD	Target	Contribution	Performance Indicators and Metrics
7	7.1	Enhancing the efficiency of SCADA systems can help increase the reliability of solar energy production, facilitating universal access to clean energy.	Percentage of solar plants with improved ...
	7.2	Improving the management of solar plants helps enhance the efficiency and reliability of renewable energy.	Increase in renewable energy share ...
8	8.1	Enhancing renewable energy infrastructure promotes resilience against climate-related hazards and supports sustainable energy sources.	Increase in resilience metrics ...

[1] https://sdgs.un.org/goals

Agradecimentos

Chegado o final desta etapa, quero agradecer a todos que tornaram o meu sucesso uma realidade.

À Bragalux, SA. e ao Eng. Bruno Rodrigues, o meu sincero agradecimento pela oportunidade que me deram de integrar este projeto. Ao meu orientador, Eng. Bruno Coelho, e aos Eng. Nuno Santos e Hugo Silva, um obrigado muito grande: pela paciência, pela disponibilidade e pela boa disposição que me deram incondicionalmente. Ao Prof. José Neves dos Santos, agradeço muito por ter aceite o convite, apesar de já não exercer funções e pelos contributos ao longo de todo este processo.

Aos meus pais, devo um agradecimento muito sentido por serem a base da pessoa que me tornei e por tudo o que sacrificaram para eu chegar aqui. À minha família, obrigado por acreditarem em mim e por estarem sempre do meu lado. Aos meus amigos, um obrigado por partilharem os bons e os maus momentos e por me puxarem para cima sempre que precisei. E à Paula, um obrigado muito especial: és a minha rocha e a minha fonte de energia em tudo o que faço.

Há mais gente a quem devo gratidão e que não cabe aqui por nome. A todos, o meu obrigado. Fico com o que aprendi e com as pessoas que tornaram este percurso possível.

"Invention is the most important product of man's creative brain."

Nikola Tesla

Contents

1 Introduction **1**
- 1.1 Research Environment . . . 1
- 1.2 Problem Statement and Objectives . . . 1
- 1.3 Scope and Work Proposal . . . 2
- 1.4 Motivation and Applications . . . 3
- 1.5 Structure of the Work . . . 3

2 Literature Review **4**
- 2.1 Problem Definition Through Standards . . . 4
- 2.2 CAD Systems and Automation in Electrical Engineering . . . 5
  - 2.2.1 CAD-Based Extraction Approaches . . . 5
  - 2.2.2 Image-Based Deep Learning Approaches . . . 6
  - 2.2.3 Methodology Comparison . . . 7
- 2.3 Naming Methodologies and Component Identification . . . 8
- 2.4 Validation Tools and Report Generation . . . 8
- 2.5 Synthesis and Research Gaps . . . 9
  - 2.5.1 Synthesis . . . 9
  - 2.5.2 Extraction Methodology Analysis . . . 10
  - 2.5.3 Research Gaps . . . 11
  - 2.5.4 Comparative Analysis of Methodologies . . . 11
  - 2.5.5 Conclusion . . . 13

3 E-REDES Standards and Current Workflow **14**
- 3.1 Panel Classification and Cable Association Rules . . . 14
  - 3.1.1 Substation Organization and Structure . . . 14
  - 3.1.2 Full Project Template Structure . . . 15
  - 3.1.3 Panel Classification Rules . . . 15
- 3.2 E-REDES Cable Naming Rules . . . 17
- 3.3 Terminal Strip and Conductor Identification . . . 18
- 3.4 Cable Type Selection and Section Requirements . . . 20
- 3.5 Documentation Requirements . . . 21
- 3.6 Current Manual Workflow in Bragalux . . . 22
- 3.7 Automation Requirements Derivation . . . 24

4 System Architecture and Methodology **25**
- 4.1 System Overview . . . 25
  - 4.1.1 Design Rationale . . . 26
- 4.2 CAD to PDF Conversion Pipeline . . . 28

4.2.1 Format Detection and Routing . . . 28
4.2.2 Vector Extraction Methods . . . 28
4.2.3 Layer Preservation Techniques . . . 29
4.2.4 Output Generation . . . 29
4.3 Terminal Extraction Architecture . . . 29
4.3.1 Text Extraction from PDF . . . 30
4.3.2 Terminal Pattern Recognition . . . 30
4.3.3 Text Merging and Validation . . . 32
4.3.4 Spatial Validation and Circle Detection . . . 33
4.3.5 Terminal List Generation . . . 34
4.4 Route Detection Architecture . . . 37
4.4.1 Route Extraction Methods . . . 37
4.4.2 Layer-Based Filtering . . . 38
4.4.3 Parallel Route Identification . . . 38
4.4.4 Route Validation and Geometric Analysis . . . 39
4.5 Cable Naming Architecture . . . 41
4.5.1 Route-Terminal Matching . . . 41
4.5.2 Equipment and Panel Logic . . . 41
4.5.3 Limitations of PDF-Based Connection Line Extraction . . . 42
4.6 System Integration . . . 44
4.6.1 Pipeline and Data Flow . . . 44
4.6.2 Errors and Propagation . . . 44

5 Implementation, Datasets, and Validation 45
5.1 Terminal Extraction Implementation . . . 45
5.1.1 Technology Stack . . . 45
5.1.2 PDF Text Extraction . . . 45
5.1.3 Terminal Pattern Recognition . . . 45
5.1.4 Text Merging . . . 46
5.1.5 Tolerance Calibration . . . 46
5.1.6 Spatial Validation . . . 47
5.2 Computer-Aided Design (CAD) to PDF Conversion . . . 47
5.3 Route Detection Implementation . . . 48
5.4 Cable Naming Implementation . . . 48
5.5 Datasets and Test Cases . . . 48
5.6 Validation Methodology . . . 50

6 Results and Discussion 51
6.1 Terminal Extraction Results . . . 51
6.1.1 Pilot dataset (SE Pracana) . . . 51
6.1.2 Full project template . . . 52
6.1.3 Error Analysis: False Positives and False Negatives . . . 52
6.2 CAD to PDF Conversion Results . . . 53
6.3 Route Detection Results . . . 54
6.4 Cable Naming Results . . . 54
6.5 Discussion . . . 55

7 Conclusions 56

References **61**

List of Figures

1.1 Bragalux SA. logo 2

2.1 CAD System Architecture 5

3.1 Panel Classification Decision Tree 17

4.1 System Architecture Overview 26

5.1 Row layout (schematic): ruler and terminal numbers on the same horizontal line. 46
5.2 Column layout: ruler above or below vertically arranged terminal numbers, merged by proximity into ruler:number identifiers. 47
5.3 Reference data Excel format: terminals by ruler, with page and unified coordinate (page and grid cell) for traceability. 49
5.4 Example E-REDES schematic excerpt showing the grid; the unified coordinate in the reference data (page and cell) refers to this grid for traceability. 49
5.5 Validation workflow: reference data, extraction, matching, metrics, error analysis. 50

6.1 Schematic region where contact block numbers match the terminal pattern; without geometric validation such text can be incorrectly reported as terminals. 53

List of Tables

2.1 Research Approaches - Extraction Methodologies and Capabilities 9
2.2 Commercial Electrical Computer-Aided Design (ECAD) Tools - Capabilities and Limitations 10
2.3 Accuracy Comparison Across Extraction Methodologies 13

3.1 E-REDES project template structure: panels and page counts. 15
3.2 Panel Types and Identifiers 16
3.3 High Voltage (HV) & Medium Voltage (MV) Cable Naming Pattern Components 17
3.4 Low Voltage (LV) Cable Naming Pattern Components 18
3.5 Voltage Level Codes and Thresholds 18
3.6 Terminal Strip Functions and Numbering Formats 19
3.7 Conductor Color Coding by Circuit Type 19
3.8 Circuit-to-Cable-Type Mapping 20
3.9 Minimum Cable Sections by Circuit Function 21
3.10 E-REDES Requirements and Automation Implications 22
3.11 Common Error Types in Manual Workflow 23

4.1 Terminal Extraction Tolerance Parameters 35
4.2 Route Detection Tolerance Parameters 35
4.3 Equipment-to-Panel Mapping Rules 41

5.1 Text merging tolerance values 46
5.2 Validation dataset characteristics 49
5.3 Validation metric definitions 50

6.1 Terminal extraction results by ruler type for SE Pracana pilot dataset. 51
6.2 Terminal extraction validation results for full E-REDES project template panels. . 52

List of Acronyms

AC Alternating Current
ACCP Command, Control and Protection Cabinets
API Application Programming Interface
ARC Cable Regrouping Cabinets
BOM Bill of Materials
CAD Computer-Aided Design
COM Component Object Model
DC Direct Current
DIFB Differential Protection Panel
DWG Drawing
EC Command Building
ECAD Electrical Computer-Aided Design
HV High Voltage
IED Intelligent Electronic Device
LV Low Voltage
ML Machine Learning
MLLM Multimodal Large Language Model
MV Medium Voltage
OCR Optical Character Recognition
OCG Optional Content Groups
PEA Outdoor Equipment Area
PCL Local Control Post
QIT Lightning and Outlet Panel
SACA Alternate Current Auxiliary Services
SACC Direct Current Auxiliary Services
SDIC Digital Input/Output Control Panel
SDIT Digital Input/Output Panel
SLD Single-Line Diagram
SPICE Simulation Program with Integrated Circuit Emphasis
SQST Quality Monitoring Panel

TP Power Transformer

TSA Auxiliary Services Transformer

UC Central Unit

YOLO You Only Look Once

Chapter 1

Introduction

Electrical substations require detailed documentation of all cable connections for construction, operation, maintenance, and regulatory compliance. In Portugal, the E-REDES standards [1] require strict naming formats for every cable and require clear documentation of conductor-level connectivity.

Engineers currently spend approximately 2 to 4 hours manually producing cable lists for new substation installations, depending on the complexity of the project. This manual process introduces errors that compromise system safety and can result in failed regulatory audits. Common errors include incorrect cable naming, missing connections, and terminal misidentification.

This work addresses these challenges through automated terminal extraction and PDF-based processing: multi-stage validation for terminal extraction accuracy, exploration of cable naming within geometric limits, and batch workflow integration. Validation uses real industrial data from Bragalux projects, so the results reflect production conditions rather than idealized test cases.

1.1 Research Environment

Bragalux, SA., shown in Figure 1.1, is a company that specializes in the construction and maintenance of electrical infrastructure and substations and hosted this research. The company completes up to five new substation projects annually and performs interventions on more than fifty existing installations each year. This industrial partnership provided access to real engineering drawings and operational workflows that ensure that the system requirements reflect the actual E-REDES project practices.

1.2 Problem Statement and Objectives

The manual generation of cable lists from substation drawings is labor-intensive and error-prone. Typical substations contain numerous inter-panel and panel-to-equipment cables, each with multiple conductors and terminal points. Documentation requires cross-referencing of multi-page drawings to verify connections, cable types, and terminal designations.



Figure 1.1: Bragalux SA. logo

Based on Bragalux operational data, the manual cable list production process typically requires approximately 2 to 4 hours for new substation installations, depending on project complexity. For integrated project interventions that involve equipment additions or modifications within existing substations, the process requires one to two hours, depending on the depth of the intervention. Common errors include incorrect cable naming, missing connections, terminal misidentification, and inconsistent panel identifier assignment.

These challenges motivate the development of automated extraction and documentation support. The E-REDES standards require structured cable naming and conductor-level traceability. Existing research has addressed symbol detection and text extraction from electrical drawings [2], [3], but gaps remain in integrated utility-specific automation; the literature review identifies these gaps.

The objectives of this thesis are the following:

- Extract terminal connection information from substation drawings and generate terminal lists suitable for technician review and manual cable list production.
- Implement and evaluate cable naming automation using E-REDES grammars and panel classification, and document its limitations where geometric extraction is insufficient.
- Validate the system using substation documentation from Bragalux projects and demonstrate the accuracy and completeness of terminal extraction versus manual workflows.
- Provide a PDF-based workflow architecture that allows batch processing without freezing Computer-Aided Design (CAD) software during analysis.

1.3 Scope and Work Proposal

The scope focuses on support for E-REDES substation documentation for Bragalux. The system processes CAD drawings exported as vector PDF files and outputs terminal lists in E-REDES-compliant format. In scope: terminal extraction from multi-page PDF with multi-stage validation

and ruler assignment; implementation of E-REDES cable naming grammars and panel classification, with evaluation and documentation of limitations; CAD-to-PDF conversion and batch workflow; validation of Bragalux project data and comparison with manual terminal identification. The work proceeds through requirements analysis for E-REDES and Bragalux, system design, Python implementation, testing on Bragalux drawings, and evaluation of terminal extraction accuracy.

1.4 Motivation and Applications

Bragalux's project volume and manual process challenges demonstrate the motivation for automation support. With up to five new substation projects annually and more than fifty interventions per year, ensuring complete terminal identification across all projects requires systematic validation. Based on the time requirements in Section 1.2, the annual manual effort ranges from 60 to 200 hours or more. Terminal extraction automation identifies and lists all terminals for technician check; the system supports verification of completeness and accuracy, while technicians remain responsible for producing cable documentation manually. As project volume grows, manual methods require time proportional to the number of drawings and terminals because each drawing is processed sequentially with human review; automation allows completeness to be checked systematically across many drawings at once.

The system provides complete terminal identification for technician review, improved accuracy through automated validation, and E-REDES-compliant terminal list formatting that supports manual cable list production. Primary users include Bragalux engineering teams, on-site technicians, E-REDES auditors, and maintenance staff.

1.5 Structure of the Work

This work is organized into seven chapters.

Chapter 1 introduces the research environment, the problem statement and objectives, the scope, and the motivation.

Chapter 2 reviews the literature on cable documentation and automation and identifies research gaps.

Chapter 3 details the characteristics of Portuguese substations, E-REDES standards, and the current workflow at Bragalux, establishing the automation of requirements.

Chapter 4 presents the methodology and architecture of the system, including terminal extraction, route detection, and cable naming components.

Chapter 5 describes the implementation, datasets, and validation procedure, documenting technical decisions and reproducibility.

Chapter 6 summarizes the results and discusses the performance of the system compared to manual approaches, with particular focus on the accuracy of cable and terminal extraction.

Chapter 7 presents conclusions and opportunities for future improvement, including limitations and research directions.

Chapter 2

Literature Review

This chapter reviews the existing work on cable documentation automation in electrical projects and identifies the gaps that motivate the system developed in this thesis. Chapter 1 defined the problem and scope; here, we examine how the literature addresses extraction from drawings, application of naming rules, validation, and report generation.

The chapter is organized as follows. Section 2.1 frames the problem through utility standards and naming rules. Sections 2.2 through 2.4 review CAD and image-based extraction, naming and component identification, and validation and report generation. Section 2.5 summarizes the gaps that the system architecture aims to address, based on the E-REDES requirements.

2.1 Problem Definition Through Standards

The automation problem is defined by utility-specific standards: a system must extract information from drawings, apply naming rules to produce identifiers, validate that outputs comply with those rules, and generate documentation in the required structure. The following sections review how existing work addresses extraction, naming and component identification, and validation and report generation; Section 2.5 synthesizes the gaps. This section only sets the frame: which utility defines the requirements and what kind of requirements they are, so that the review and gaps make sense. The detailed E-REDES rules—naming grammars, panel classification, deliverables, validation criteria—are set out in the next chapter.

E-REDES is the primary electricity distribution operator in mainland Portugal, managing the High Voltage (HV), Medium Voltage (MV), and Low Voltage (LV) grids under a state concession. Most of the work carried out by the substations team falls within the voltage levels and project types operated by E-REDES; E-REDES norms and standards are therefore the reference for this thesis. Those standards define naming rules as formal patterns for cable identifiers and panel classification, required deliverables such as connection lists, cable schedules, and technical specifications, and validation criteria including syntax and completeness. Any solution that targets this context must address extraction, rule-based name generation, validation, and formatted report production; the literature is reviewed against that pipeline.

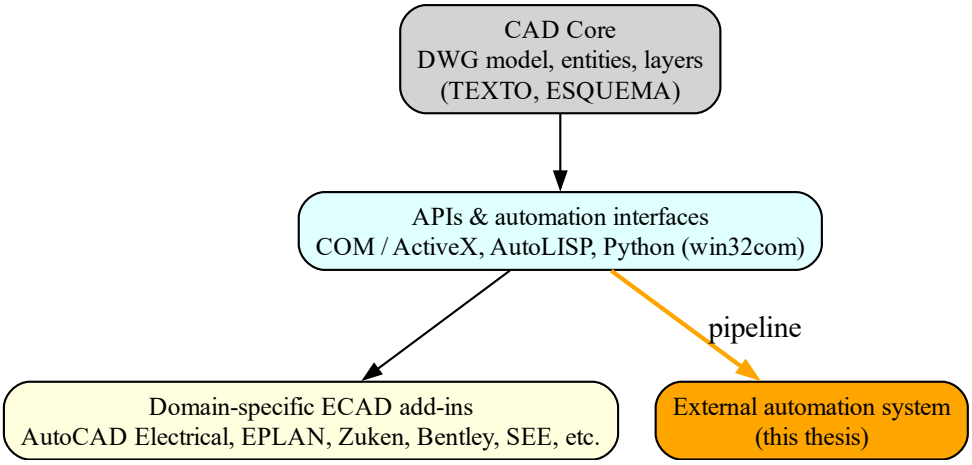


Figure 2.1: CAD System Architecture

E-REDES succeeded the former EDP distribution operator, which was rebranded in 2021; project types and documentation structure remain largely unchanged. Therefore, previous work on the manual application of these standards still applies: structured project methodologies [4], EDP-type project implementation [5], electrical project design [6], and project workflows [7].

2.2 CAD Systems and Automation in Electrical Engineering

This section examines existing approaches for automating cable documentation, organized by extraction methodology. Subsection 2.2.1 covers CAD-based approaches that access drawing data directly through programmatic interfaces. Subsection 2.2.2 examines image-based deep learning approaches that process rasterized or vectorized diagrams. Subsection 2.2.3 compares these methodologies and identifies their limitations for utility-specific automation.

2.2.1 CAD-Based Extraction Approaches

The CAD systems used in electrical engineering provide programmatic interfaces through scripting languages and Application Programming Interface (API)s [8]. Figure 2.1 illustrates the layered architecture of the automation interfaces of the CAD systems. These systems implement scripting languages like AutoLISP and Python, Component Object Model (COM) interfaces for programmatic access to drawing entities and properties, and specialized add-ins for domain-specific electrical functionality.

AutoCAD Electrical automates symbol placement, connection documentation through wire numbering, and Bill of Materials (BOM) generation. However, utility-specific requirements necessitate customization for particular naming conventions and documentation structures. So it cannot be applied directly to E-REDES projects without extensive configuration.

AutoCAD provides COM interfaces for programmatic access to drawing entities, object properties, including layer assignments and attribute data, and automation routines [8]. Current implementations focus on manual processes with limited automation support for naming rule applications. The COM interfaces provide programmatic access to CAD object models and entity collections [9]. However, name generation according to formal naming grammars requires algorithmic implementation beyond simple scripting, involving rule evaluation engines that are not provided by standard CAD automation interfaces.

The Nashville Electric Service implementation automates cable modeling using AutoCAD Electrical and Inventor [8]. The Substation Design Suite automates cable creation, conductor sizing based on current ratings, and routing algorithms. The BOM generation updates automatically when parts are replaced in the model, although these outputs are inventory lists rather than formatted technical reports that meet the utility-specific documentation requirements. The implementation required utility-specific symbol libraries, drawing templates, and naming procedures. This case demonstrates that CAD-based automation requires significant customization effort and produces outputs that do not necessarily match the utility-specific documentation structures.

Commercial Electrical Computer-Aided Design (ECAD) software provides specialized capabilities for electrical design, including automated cable naming and report generation. EPLAN Electric P8 provides broad electrical design capabilities, including automatic cable numbering, terminal strip generation, and automated report generation [10]. The Zuken E3.series offers electrical design capabilities that include wiring harness design, cable routing, and automated documentation [11]. Bentley OpenUtilities Substation provides specialized functionality for substation design [12]. SEE Electrical [13], Smap3D Electrical [14], Elecdes Design Suite [15], and Siemens Capital Suite [16] all provide automated documentation capabilities.

These commercial ECAD tools automate cable naming and report generation but require extensive configuration to match utility-specific naming conventions and documentation structures. That overhead limits their use for E-REDES projects without substantial customization. The tools are also standalone, so they do not integrate with existing CAD workflows where drawings are already in AutoCAD; converting or recreating drawings in another package disrupts the workflow.

2.2.2 Image-Based Deep Learning Approaches

Deep learning approaches extract information from electrical diagrams represented as raster images or vectorized PDF. These methodologies can be grouped by their primary contribution: text classification and extraction, symbol detection, and wiring optimization.

For text classification and extraction, Putra et al. [2] extract cable connection information from single-line wiring diagrams. Their system uses Machine Learning (ML) text classification and a

convolutional neural network for line detection. The system employs ML.NET with a Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimization algorithm. It categorizes text annotations into circuit numbers, cable types, deck locations, and device names with 89.6% average accuracy. The approach extracts existing identifiers rather than generating names from normative rules, so it cannot be used for drawings with incomplete or missing labels—common in E-REDES projects where cable names must be generated algorithmically. It also needs training data that may not generalize across drawing styles or utility-specific conventions, which limits use for E-REDES without extensive retraining.

In symbol detection and wiring optimization, Ekeke et al. [17] present an attention-based framework that integrates the attention mechanisms in You Only Look Once (YOLO) v8 for symbol recognition. The framework achieves 0.859 mean average precision with Efficient Channel Attention modules. Their graph-based A* pathfinding algorithm reduces the wire length by 43% while adhering to routing constraints. The focus is on geometric optimization rather than cable naming. The approach does not handle utility-specific naming conventions or documentation requirements, so it is not directly applicable to E-REDES cable documentation automation. It also assumes complete route information, which many drawings lack when routes are partial segments or connection information is encoded by spatial relationships rather than explicit paths.

Regarding the visual recognition of Single-Line Diagram (SLD), Yang et al. [3] achieve 99.53% F1-score for symbol detection. Their approach uses a Swin Transformer backbone with Faster R-CNN and text super-resolution networks. Related work by Hemker et al. [18] extracts the circuit topology from the schematics to generate Simulation Program with Integrated Circuit Emphasis (SPICE) netlists. Yao et al. [19] and Bhanbhro et al. [20] process multi-scale drawings and detect symbols, respectively. These approaches extract component symbols and existing identifiers well but do not generate names from utility-specific rules or produce formatted technical reports that meet regulatory requirements. High symbol detection accuracy shows that extraction is feasible; the missing piece is name generation, so they cannot be applied to E-REDES projects where names must be generated algorithmically from extracted information.

2.2.3 Methodology Comparison

CAD-based approaches give direct access to drawing data through programmatic interfaces and allow real-time extraction of entity properties and relationships. They require CAD software to be frozen during processing, so technicians cannot work in parallel, and they lack built-in support for utility-specific naming rules. Image-based approaches achieve high accuracy for symbol detection and text extraction but lose vector geometry when rasterizing and do not handle name generation from utility-specific rules. Both address extraction but neither delivers an integrated solution that combines extraction, name generation, validation, and formatted report production for E-REDES compliance.

Extraction methodologies identify components and connections, but generating compliant cable names needs more than extraction. Section 2.3 examines naming and component identification and how existing research tackles name generation, which extraction alone cannot solve.

2.3 Naming Methodologies and Component Identification

The E-REDES cable naming patterns incorporate voltage level codes, phase identification for three-phase systems, sequential numbering, and panel identifiers. Panel classification systems determine panel identifiers through conditional logic that evaluates equipment associations and panel hierarchies.

Research approaches extract cable information and identify components from electrical drawings using various methodologies—text classification [2], attention-enhanced detection [17], transformer-based recognition [3], topology extraction [18], adaptive object detection [19], symbol detection [20] and template matching [9]. While effective for extraction, these approaches do not address name generation. The extraction methodologies identify symbols and text but lack the algorithmic rule evaluation needed to generate E-REDES-compliant cable names from extracted information.

Ontology-based approaches formalize naming rules and knowledge representation in power systems. Elnagar et al. create knowledge graphs from unstructured text [21]. OntoPowerSys provides a power systems ontology for cross-domain interactions [22]. These approaches do not generate names from CAD drawings, focusing on knowledge representation rather than extraction and generation workflows. Although ontologies provide formal structures to represent naming rules, they do not address the extraction and generation pipeline required for automated cable documentation from CAD drawings.

2.4 Validation Tools and Report Generation

Current CAD automation implementations do not include automated validation against utility-specific naming rules [8]. The Nashville Electric Service case generates reports with self-checking capabilities, but validation focuses on BOM completeness rather than naming rule compliance. Their automatically generated BOM updates when parts are replaced, but these remain inventory lists rather than formatted technical reports that meet the requirements of utility-specific documentation. So existing validation covers data completeness but not compliance with utility-specific naming syntax and semantic rules required for E-REDES projects.

Onabanjo et al. generate reports from electrical diagrams using Multimodal Large Language Model (MLLM)s [23]. The system uses ChatGPT, Gemini, and Claude to generate calculations and technical descriptions from single-line diagrams. The approach produces textual content, not formatted technical reports that match the utility-specific documentation structure. Template-based formatting is required to convert generated content into reports that match the utility-specific structure [23]. Although MLLMs demonstrate the ability to generate technical content, they do not address the structured formatting requirements and naming rule compliance validation needed for E-REDES documentation, which require additional template-based processing that limits automation effectiveness.

Table 2.1: Research Approaches - Extraction Methodologies and Capabilities

Source	Input Format	Symbol Detection	Text Extraction	Connection Mapping	Name Generation	Report Generation	Validation
[2]	Raster/Vector PDF	✓	✓	✓	×	∼ (Lists)	×
[17]	Images	✓	×	✓	×	×	×
[3]	Images	✓	✓	×	×	×	×
[18]	Schematics	✓	×	✓	×	∼ (SPICE netlists)	×
[19]	Multi-scale drawings	✓	✓	×	×	×	×
[20]	Single-line diagrams	✓	×	×	×	×	×
[9]	Diagrams	✓	✓	×	×	×	×
[8]	AutoCAD files	✓	✓	✓	×	∼ (BOM lists)	∼ (BOM)
[23]	Single-line diagrams	✓	✓	×	×	∼ (Content)	×
[21]	Unstructured text	×	✓	×	×	×	×
[22]	Cross-domain data	×	×	×	×	×	×

Legend: ✓ = Fully addresses; ∼ = Partially addresses; × = Does not address

The E-REDES standards define the structural requirements for technical documentation: specific sections, formatting standards, and content organization. Connection lists document equipment interconnections extracted from wiring diagrams, specifying source and destination terminals. Cable schedules list cables with assigned names, conductor types, cross-sections, terminations, and routing information. Template-based report generation inserts extracted and generated information into predefined structures matching these requirements.

2.5 Synthesis and Research Gaps

2.5.1 Synthesis

Research approaches extract information from electrical diagrams using three primary methodologies: PDF-based extraction, image-based deep learning, and native CAD processing. Table 2.1 synthesizes these approaches: most cover symbol detection and text extraction, but none cover automated name generation from utility-specific rules. Table 2.2 analyzes commercial ECAD tools separately; all require extensive customization for utility-specific compliance.

Table 2.1 shows that most approaches cover symbol detection and text extraction but none cover automated name generation from utility-specific rules. All listed approaches achieve symbol detection; most achieve text extraction; several achieve connection mapping. No source addresses name generation; only two provide partial report generation, and that does not match utility-specific documentation requirements.

Putra et al. [2] cover extraction and connection mapping most fully but still lack name generation. The commercial CAD automation approach by Humel and Weaver [8] provides partial report generation, but outputs inventory lists rather than formatted technical reports matching utility-specific documentation requirements.

Table 2.2: Commercial ECAD Tools - Capabilities and Limitations

Software	Cable Naming	Report Quality	Specialization	Customization	Integration
EPLAN [10]	✓Automatic	Comprehensive	General ECAD	∼ Required	Standalone
Zuken [11]	✓Automatic	Comprehensive	Wiring harness	∼ Required	Standalone
OpenUtilities [12]	∼ Physical	Comprehensive	Substation/3D	∼ Required	Standalone
SEE Electrical [13]	✓Automatic	Basic	General ECAD	∼ Required	Standalone
Smap3D [14]	✓Automatic	Basic	3D electrical	∼ Required	Standalone
Elecdes DS [15]	✓Automatic	Basic	General ECAD	∼ Required	Standalone
Siemens CS [16]	✓Automatic	Enterprise	Enterprise systems	∼ Required	Standalone

Legend: ✓= Fully addresses; ∼ = Partially addresses; × = Does not address

This pattern demonstrates that extraction methodologies are well-established, but the gap in automated name generation remains unaddressed.

Table 2.2 shows that all listed commercial ECAD tools require customization to match utility-specific naming conventions. All seven tools provide automatic cable naming and require customization (∼ in the Customization column). EPLAN and Zuken, for example, provide broad report generation, but the need for customization is a major barrier for E-REDES projects. Because the tools are standalone, they do not integrate with existing CAD workflows where drawings are in AutoCAD or ZWCAD; commercial tools therefore cannot be applied directly to E-REDES projects without extensive configuration.

2.5.2 Extraction Methodology Analysis

Research approaches use three primary extraction methodologies, each with distinct advantages and limitations for E-REDES cable documentation automation:

PDF-based extraction: Putra et al. [2] extract from raster and vector PDF using Optical Character Recognition (OCR) and PDF parsing algorithms, achieving 89.6% accuracy. This methodology supports both scanned raster PDF and digital vector PDF, matching workflows where CAD drawings are exported to PDF format. Vector PDF preserve geometric information, while raster PDF require OCR processing. Putra et al. [2] demonstrate that PDF-based extractions achieve precision comparable to that of image-based approaches while maintaining vector information. The approach extracts existing identifiers rather than generating names from utility-specific rules, so it cannot be used for drawings with incomplete or missing labels—common in E-REDES projects where names must be generated algorithmically from extracted information. It also requires high-quality rasterization or vectorization, which may not preserve all geometric relationships needed for spatial validation in complex drawings.

Image-based extraction: Ekeke et al. [17], Yang et al. [3], Yao et al. [19], and Bhanbhro et al. [20] process rasterized images using deep learning models. These approaches require vector drawing rasterization and achieve high symbol detection accuracy, 99.53% F1-score for Yang et al. [3], but lose vector geometric information. The loss of vector precision reduces the effectiveness

of spatial validation and prevents the accurate coordinate-based terminal matching required for the generation of the E-REDES connection list.

CAD-native extraction: Nashville Electric [8] accesses CAD data directly through COM interfaces, enabling real-time extraction of entity properties and relationships but requiring CAD software to be frozen during processing, preventing technicians from working simultaneously. This dependency prevents parallel processing workflows and creates bottlenecks when multiple drawings must be processed simultaneously.

2.5.3 Research Gaps

The literature review reveals four gaps that this thesis addresses:

Gap 1: Automated name generation. Extraction methodologies identify equipment symbols, connection paths, and existing identifiers [2], [3], [17], [18], [19], [20], [24], none generate names according to utility-specific rules. Name generation requires algorithmic implementation of voltage level code mapping, phase identification, sequential numbering, and panel assignment rules.

Gap 2: Integrated pipeline. Individual components exist for extraction [2], [3], [17], report generation [8], [23], and validation [8]. However, no integrated system combines vectorized PDF extraction, algorithmic name generation, rule-based validation, and formatted report production in a unified workflow.

Gap 3: PDF-based workflow integration. Research approaches work with either rasterized images [3], [17], [19], [20] or CAD-native formats [8]. Although PDF extraction demonstrates feasibility [2], no system integrates PDF extraction with name generation, validation, and report formatting.

Gap 4: Utility-specific compliance. Commercial ECAD tools [10], [11], [12], [13], [14], [15], [16] provide generic automation that requires extensive customization. No system addresses utility-specific compliance requirements out of the box.

2.5.4 Comparative Analysis of Methodologies

A comparative analysis reveals distinct trade-offs between extraction methodologies that explain why no single approach addresses all E-REDES requirements. CAD-based approaches, e.g. [8], provide direct access to drawing data with real-time entity property extraction, achieving high precision for symbol detection and text extraction. However, they require the CAD software to freeze during processing, preventing technicians from working simultaneously and creating workflow bottlenecks when multiple drawings must be processed simultaneously. The Nashville Electric Service case demonstrates that CAD-based automation requires significant customization effort—utility-specific symbol libraries, drawing templates, naming procedures—and produces outputs such as inventory lists that do not match utility-specific documentation structures.

Image-based deep learning approaches, e.g. [3], [17], achieve high symbol detection accuracy (99.53% F1-score for Yang et al.) but lose vector geometry when rasterizing. Without that geometry, spatial validation is weaker and coordinate-based terminal matching for E-REDES connection

lists is less accurate. Deep learning approaches also need large training sets covering terminal format variation across drawing styles, which limits generalization to E-REDES conventions without retraining.

PDF-based extraction, e.g. [2], preserves vector information while enabling batch processing, achieving 89.6% accuracy for text classification. However, this approach extracts existing identifiers rather than generating names according to utility-specific rules, preventing application to drawings with incomplete or missing labels common in E-REDES projects where cable names must be generated algorithmically.

The severity of the gap is quantified by Table 2.1: zero of the eleven research sources address name generation, and only two provide partial report generation that does not meet the utility-specific documentation requirements. This pattern demonstrates that extraction methodologies are well-established, but the gap in automated name generation remains unaddressed. The technical challenges explaining these gaps are grounded in the E-REDES requirements: utility-specific naming rules require algorithmic rule evaluation beyond simple pattern matching, as standards define formal grammars with voltage codes, phase identifiers, and panel associations; geometric ambiguity in route-terminal matching requires semantic understanding that geometric-only approaches cannot provide; and E-REDES standards define complex conditional logic such as panel classification and equipment associations that require knowledge-based systems rather than geometric analysis alone.

Commercial ECAD tools address generic automation, but require extensive customization for utility-specific compliance. EPLAN Electric P8 [10] and Zuken E3.series [11] provide comprehensive automation capabilities, but customization requires the development of utility-specific symbol libraries, the configuration of naming rules through proprietary interfaces, creation of documentation templates matching E-REDES structures, and integration with existing CAD workflows such as AutoCAD or ZWCAD that may require format conversion or drawing recreation. This customization overhead, as demonstrated in the Nashville Electric Service case [8] which required utility-specific symbol libraries, drawing templates, and naming procedures, creates a significant barrier to adoption for E-REDES projects. The standalone nature of these tools also prevents seamless integration with existing CAD workflows, which requires workflow disruption when drawings must be converted or recreated in different software environments.

Table 2.3 synthesizes the precision metrics across extraction methodologies. The table reveals that image-based deep learning approaches achieve the highest symbol detection accuracy, 99.53% F1-score for Yang et al. [3], while PDF-based extraction achieves comparable accuracy, 89.6% for Putra et al. [2], while preserving vector information. CAD-native approaches provide direct data access, but lack reported accuracy metrics in the literature. The comparison demonstrates trade-offs between accuracy, vector preservation, and workflow integration that explain why no single approach addresses all E-REDES requirements simultaneously.

Table 2.3: Accuracy Comparison Across Extraction Methodologies

Methodology	Symbol Detection	Text Extraction	Vector Preservation
Image-based Deep Learning	99.53% F1 (Yang et al.)	High (OCR)	× Lost in rasterization
PDF-based Extraction	89.6% (Putra et al.)	High (PDF parsing)	✓ Preserved
CAD-native Extraction	Not reported	High (COM access)	✓Native format

Source: Synthesized from literature review. Symbol detection accuracy from Yang et al. [3], PDF extraction accuracy from Putra et al. [2]. Vector preservation indicates whether geometric information is maintained.

2.5.5 Conclusion

This chapter reviews existing approaches for automating cable naming and report production. Extraction methodologies work with PDF [2], images [3], [17], [19], [20], or CAD-native formats [8]. Deep learning approaches achieve high accuracy for symbol detection and text extraction. Commercial ECAD tools provide generic automation [10], [11], [12], [13], [14], [15], [16]. However, four gaps remain: automated name generation, integrated extraction-generation-validation-reporting pipelines, PDF-based workflow integration, and utility-specific compliance.

This thesis aims to address these gaps through an integrated system that extracts information from vectorized PDF exported from CAD systems, with terminal extraction as primary deliverable and cable naming automation explored within the limits of geometric extraction. The approach is based on PDF-based extraction methodologies [2], extending them with multi-stage validation and E-REDES-compliant output. The next chapter examines the E-REDES standards that define these requirements.

Chapter 3

E-REDES Standards and Current Workflow

This chapter examines the E-REDES standards that define cable naming rules, terminal identification, cable type selection, and documentation requirements, establishing the specific automation challenges that the system architecture must address. The standards analysis reveals complex requirements that include panel classification rules, cable naming grammars, terminal identification formats, and documentation structure specifications. The current manual workflow in Bragalux is analyzed to identify automation opportunities and validate the requirements of the system, establishing the foundation for the system architecture presented in the next chapter.

3.1 Panel Classification and Cable Association Rules

3.1.1 Substation Organization and Structure

This section presents the E-REDES standards that define the substation organization and cable naming rules. DIT-C13-500 specifies the substation layout and main components [25]. Portuguese substations consist of two main areas: the outdoor switchyard (Outdoor Equipment Area (PEA)) where exterior equipment is installed, and the control building (Command Building (EC)) which houses control and command equipment.

Within the control building, protection and control cabinets (Command, Control and Protection Cabinets (ACCP)) contain intelligent electronic devices (Intelligent Electronic Device (IED)s) for protection, command and control functions. These cabinets are organized by voltage level: HV, MV panels and low-voltage (LV) auxiliary service panels for metering, communications, power quality and auxiliary services. Each cabinet contains terminal strips for cable connections and IEDs mounted on swinging frames.

The grouping of cabinets (Cable Regrouping Cabinets (ARC)) installed in the outdoor switchyard reduces the number of cables between the two areas. These cabinets group circuits associated with specific equipment such as current transformers, voltage transformers, power transformers, and auxiliary transformers.

Table 3.1: E-REDES project template structure: panels and page counts.

Panel	Pages
Alternate Current Auxiliary Services (SACA)	20
Direct Current Auxiliary Services (SACC)	27
Power Transformer (TP) and Voltage Regulation	23
Inter-Busbars and HV Busbar Voltage	19
HV Line	19
TP	21
HV Busbar Voltage	9
TP Arrival	12
Auxiliary Services Transformer (TSA), Neutral Reactance and MV Busbar Voltage	15
Capacitor Bank	17
MV Line	12
Metering	10
MV Line Metering	21
Communications	11
Local Control Post (PCL) Central Unit (UC)	8
MV Inter-Busbars	10
Internal Arc	9
Power Quality	9
QIT	7
Total	**279**

Source: E-REDES project template.

3.1.2 Full Project Template Structure

The E-REDES project template represents a complete substation design comprising 19 plan types totaling 279 pages. Table 3.1 presents the complete organization.

The project template includes HV and MV panels such as transformer lines, inter-busbar connections, and busbar potential measurement; LV auxiliary service panels including SACA, SACC, metering, communications, and power quality; and specialized panels such as capacitor banks, Lightning and Outlet Panel (QIT), and internal arc protection.

3.1.3 Panel Classification Rules

The naming of LV cables requires panel identifiers following the pattern defined in D00-C13-500 [1]. Panel identifiers encode panel type, voltage level, and equipment associations. The panel classification rules determine how the panels receive the identifiers as follows.

DIT-C13-500 classifies panels by voltage level and function [25]. Table 3.2 summarizes panel types and their corresponding panel identifiers for cable naming. HV panels connect busbars to distribution lines, power transformer primaries, or other busbars, i.e. interconnection panels. MV panels connect busbars to distribution lines, power transformer secondaries, or other busbars and also include transformer secondary connections, capacitor bank connections, and auxiliary transformer and neutral reactance connections. LV panels provide auxiliary services: metering,

Table 3.2: Panel Types and Identifiers

Panel Type	Panel Identifier	Voltage Level
60kV panels	P5NP	HV
30kV panels	P3NP	MV
15kV panels	P2NP	MV
10kV panels	P1NP	MV
SACA	SACA	LV
SACC	SACC	LV
Metering panel	CONT	LV
Communication panel	COM	LV
Uninterruptible power supply panel	UPC	LV
Digital Input/Output Panel (SDIT)	SDIT	LV
Digital Input/Output Control Panel (SDIC)	SDIC	LV
Quality Monitoring Panel (SQST)	SQST	LV
QIT	QIT	LV
Differential Protection Panel (DIFB)	DIFB	LV

Source: D00-C13-500 [1], DIT-C13-500 [25]

communications, uninterruptible power supply, digital input/output, quality monitoring, and differential protection functions.

Panel identifiers map panel types to alphanumeric codes used in LV cable naming. D00-C13-500 specifies the panel identifier assignment rules [1].

The HV and MV panels use voltage-level-based patterns: P5NP for 60kV panels, P3NP for 30kV panels, P2NP for 15kV panels and P1NP for 10kV panels, where NP represents the panel number.

The LV auxiliary service panels use fixed identifiers as shown in Table 3.2.

Inter-panel cables require assignment rules when cables connect different panels. D00-C13-500 specifies two assignment rules [1]. If the cable information is associated with the HV or MV equipment of a specific panel, the cable belongs to that panel. If the previous rule does not apply, the cable belongs to the panel with the lower sequential number. These rules ensure a consistent panel assignment for cables connecting multiple panels.

Implementing panel classification algorithmically requires systems to identify equipment associations, evaluate panel type based on equipment composition, assign panel identifiers according to mapping rules in Table 3.2, and apply inter-panel assignment logic for cables connecting multiple panels. For inter-panel cables, the system must first check for equipment associations, then default to the lower panel number rule. This multi-level conditional logic, combined with the need to identify equipment from drawing text and evaluate spatial relationships, creates significant complexity that manual application handles inconsistently, leading to the error sources identified below.

The HV and MV panels use voltage-level-based patterns P5NP, P3NP, P2NP, P1NP, while the LV panels use fixed alphanumeric codes. This distinction reflects the different naming require-

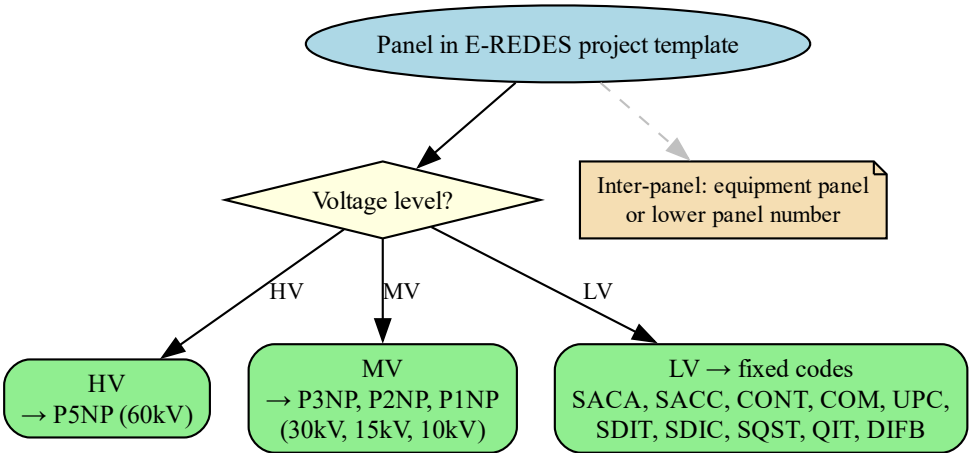


Figure 3.1: Panel Classification Decision Tree

ments: HV & MV cables encode the voltage level in the name, while LV cables encode panel type. These panel classification rules directly inform cable naming patterns, as panel identifiers are encoded in LV cable names.

The next section presents how panel identifiers are used in cable naming patterns.

3.2 E-REDES Cable Naming Rules

D00-C13-500 specifies naming patterns that encode voltage level, phase association, sequential numbering, and panel identifiers [1]. These patterns constitute formal grammars that the automation system must implement to generate compliant cable names.

Table 3.3 illustrates the structured naming pattern for HV and MV cables. The pattern encodes voltage level—1=10kV, 2=15kV, 3=30kV, 5=60kV—phase association L1, L2, L3, and sequential numbering within each phase. The formal grammar structure W + voltage_code + phase + number

Table 3.3: HV & MV Cable Naming Pattern Components

Component	Format
Cable identifier	W
Voltage code	1, 2, 3, 5
Phase identifier	L1, L2, L3
Sequential number	Integer
Example names	W2L12 (15kV, L1, #2), W5L23 (60kV, L2, #3)

Source: D00-C13-500 Section 8.1 [1]

Table 3.4: LV Cable Naming Pattern Components

Component	Format
Cable identifier	W
LV voltage code	9
Two-digit number	01–99
Panel identifier	From Table 3.2
Example names	W907P501, W912SACA

Source: D00-C13-500 Section 8.3 [1]

requires algorithmic implementation: systems must identify voltage level from circuit characteristics or drawing information, determine phase association from three-phase system analysis, and assign sequential numbers within each phase.

The LV cables follow the pattern W9 + two-digit_number + panel_identifier, see Table 3.4. The fixed voltage code 9 identifies all LV cables, while the two-digit number provides sequential identification within each panel. The panel identifier links cables to their associated panels, enabling traceability for maintenance and operations.

D00-C13-500 Section 4 defines voltage thresholds [1]. HV voltages are defined as higher than 60 kV. MV voltages range from 10 to 30 kV. LV voltages are below 1 kV. Table 3.5 summarizes the voltage level codes used in the naming patterns.

The following section documents the terminal identification rules needed for the generation of the connection list.

3.3 Terminal Strip and Conductor Identification

DRE-C13-510 specifies the assignments of terminal strip function and conductor color coding [26], while D00-C13-500 Section 9.1 specifies the terminal block numbering formats [1]. Table 3.6 summarizes all assignments and numbering formats of terminal strip functions.

Table 3.6 shows the hierarchical organization of the terminal strips. X0.x sub-rulers handle supply and auxiliary functions with low-numbered terminals 1–99. The main rulers X1–X3 handle phase terminals. X5.x sub-rulers handle inter-panel connections 501–599. X6 and X7 handle auxiliary services 601–799.

Table 3.5: Voltage Level Codes and Thresholds

Voltage Level	Code
10 kV	1
15 kV	2
30 kV	3
60 kV	5
LV	9

Source: D00-C13-500 Section 4 [1]

Table 3.6: Terminal Strip Functions and Numbering Formats

Terminal Function	Purpose	Numbering Format
X0.1	Direct Current (DC) polarities supply	X0.1:1–99
X0.2	Alternating Current (AC) supply	X0.2:1–99
X0.3	Panel DC polarities	X0.3:1–99
X0.4	Panel AC circuits	X0.4:1–99
X0.5	ACCP door DC polarities	X0.5:1–99
X0.6	Power quality	X0.6:1–99
X0.7	Earth terminals	X0.7:1–99
X1	Phase terminals (current circuits)	X1:101–199
X2	Phase terminals (voltage circuits)	X2:201–299
X3	Phase terminals (control/signaling)	X3:301–499
X5.1	Inter-panel terminals (current circuits)	X5.1:501–599
X5.2	Inter-panel terminals (voltage circuits)	X5.2:501–599
X5.3 – X5.6	Inter-panel terminals (control/signaling)	X5.x:501–599
X6	SACC	X6:601–699
X7	SACA	X7:701–799

Sources: DRE-C13-510 [26], D00-C13-500 Section 9.1 [1]

The complexity of terminal identification stems from the hierarchical structure that requires pattern recognition algorithms to extract terminal labels from drawing text. Systems must recognize three terminal categories: formatted terminals with explicit rulers such as "-X5.2:501", unformatted numbers requiring ruler inference such as "501", and special cases such as "?" or "-" placeholders. The identification challenge is compounded by the need to distinguish terminal labels from page numbers, equipment codes, and grid labels that may match terminal patterns.

DRE-C13-510 Section 5.2 defines conductor color assignments [26]. For protection and measurement circuits, use red conductors for currents and blue conductors for voltages. For control, signalization, and auxiliary services, AC circuits use black conductors, and DC, currents in mA and voltages in mV use brown conductors. DC auxiliary services that operate at 48 V use yellow conductors. Earth circuits use green/yellow conductors. This color coding enables the visual identification of circuit functions during installation and maintenance.

DRE-C13-510 specifies the mapping logic. The power circuit conductors connect to phase

Table 3.7: Conductor Color Coding by Circuit Type

Circuit Type	Conductor Color
Protection and measurement (currents)	Red
Protection and measurement (voltages)	Blue
Control, signaling and auxiliary services (AC)	Black
Control, signaling and auxiliary services (DC, mA, mV)	Brown
DC auxiliary services at 48 V	Yellow
Earth circuits	Green & Yellow

Source: DRE-C13-510 Section 5.2 [26]

Table 3.8: Circuit-to-Cable-Type Mapping

Circuit Function	Cable Type	Characteristics
Force, command & signaling	XAZ1	Steel armor, fire retardant, zero halogen
AC auxiliary services supply	XAZ1	Steel armor, fire retardant, zero halogen
PEA lighting and socket circuits	XAZ1	Steel armor, fire retardant, zero halogen
Current and voltage measurement circuits	XHZ1	Copper tape armor, electromagnetic isolation
Lighting circuits	XZ1	Fire retardant, zero halogen
Socket circuits	XZ1	Fire retardant, zero halogen
Communication circuits	SFTP Cat6A	Structured cabling

Source: DRE-C13-512 Section 3.2 [27], DRE-C13-511 Section 10 [28] for communication circuits

terminals X1, X2, and X3. The command conductors connect to the control terminals X3 or the auxiliary terminals X5.3 through X5.6. The measurement conductors connect to the current terminals X1 or voltage terminals X2. The inter-panel conductors connect to the auxiliary terminals X5. These circuit-to-terminal mapping rules enable automated association of extracted conductors with terminal functions for connection list generation. The mapping rules require systems to identify circuit functions from the drawing context, creating an automation challenge addressed through route-terminal matching algorithms.

The following section documents cable type selection rules needed for cable schedule generation.

3.4 Cable Type Selection and Section Requirements

DRE-C13-512 defines circuit-to-cable-type mapping and minimum section requirements [27]. DRE-C13-512 Section 3.2 defines three cable type categories [27]. Communication circuits use structured cabling according to DRE-C13-511 Section 10 [28], which specifies SFTP Cat6A cables following EN 50173-1 standards. Table 3.8 summarizes the circuit-to-cable-type mapping.

DRE-C13-512 Section 4 specifies the assignments of sections. Table 3.9 summarizes the minimum section requirements.

Auxiliary services require larger sections, 25–35 mm², due to higher current ratings, while control and measurement circuits use smaller sections, 2,5 mm², sufficient for signal transmission. All cable types must be fire retardant, flame retardant, and zero halogen per DRE-C13-512 Section 3.1. The rated voltage is 0.6kV/1kV for all LV cables. Metallic sheaths must be grounded at both ends when present.

The following section presents the documentation structure requirements needed for report generation.

Table 3.9: Minimum Cable Sections by Circuit Function

Circuit Function	Minimum Section (mm²)
SACA supply	35
SACC supply	25
Lighting and socket panel	16
Power circuits	2,5
Command and signaling circuits	2,5
Current measurement and protection	2,5
Voltage measurement and protection	2,5
Building lighting	2,5
Building sockets	4

Source: DRE-C13-512 Section 4 [27]

3.5 Documentation Requirements

The E-REDES standards specify the documentation deliverables and structural requirements for substation projects [1]. D00-C13-500 specifies the reference requirements: systems must format extracted and generated information into structured technical documentation that matches normative requirements. These structural requirements create an automation challenge for report generation, requiring template-based formatting that matches E-REDES documentation structures.

Cable schedules list cable names, conductor types, cross-sections, terminations, and routing information. Connection lists document terminal-to-terminal mappings using the terminal block formats described above. These lists include source and destination terminals, conductor identification through color coding or numbering, and cable associations linking conductors to cable names. Descriptive reports provide an overview of the project and equipment descriptions. The technical specifications define the characteristics of the equipment and the installation requirements.

The compliance of the name syntax verifies that the cable names follow the naming patterns. The validity of the panel identifier checks that the panel identifiers match the panel classifications. The correctness of the Voltage code ensures the voltage levels are in the same place. Documentation completeness verifies that all required fields are present and that all connections are documented. These validation requirements create automation opportunities: automated systems can verify compliance programmatically, reducing manual validation effort.

The E-REDES standards presented above define complex requirements that create significant manual effort when applied to cable list production. Panel classification requires evaluating 11 panel types from Table 3.2 with conditional logic for equipment associations and inter-panel rules. Cable naming implements two distinct grammars with voltage code mapping and phase identification. Terminal identification processes hierarchical structures with 15 ruler types from Table 3.6 and pattern recognition for formatted, unformatted and special cases. Applying these standards manually requires extensive cross-referencing of multi-page drawings, manual lookup of naming rules, and iterative validation procedures. The next section presents how Bragalux currently

Table 3.10: E-REDES Requirements and Automation Implications

E-REDES Requirement	Complexity	Automation Implication
Panel Classification	11 panel types, conditional logic, equipment associations, inter-panel rules	Multi-stage classification algorithm, equipment zone detection, lookup table-based mapping
Cable Naming Patterns	Two distinct grammars: HV/MV use W + voltage + phase + number; LV use W9 + number + panel; voltage code mapping, phase identification	Grammar implementation, pattern matching, voltage level detection, phase association logic
Terminal Identification	15 ruler types, hierarchical structures, formatted/unformatted/special cases, pattern recognition	Multi-stage validation, pattern matching, ruler inference, spatial validation, circle detection
Cable Type Selection	Circuit function classification, lookup table mapping, cross-section determination	Circuit function detection, lookup table implementation, cross-section calculation
Documentation Structure	Template-based formatting, validation requirements, structural compliance	Template engine, format validation, structural compliance checking

Source: E-REDES standards

performs these tasks manually, identifying the workflow challenges that motivate automation.

Table 3.10 summarizes E-REDES requirements and their automation implications, providing a synthesis of standards analysis and establishing clear connections between requirements and system design needs.

3.6 Current Manual Workflow in Bragalux

Based on Bragalux operational data, manual cable list production typically requires approximately 2 to 4 hours for new substation installations, depending on project complexity, and one to two hours for integrated project interventions involving equipment additions or modifications within existing substations, depending on the depth of intervention.[1]

The most time-consuming steps are name assignment, documentation compilation, and validation. Name assignment requires manual application of E-REDES naming rules, referencing standards documents and evaluating equipment associations. Documentation compilation creates cable schedules and connection lists using spreadsheet software or word processors. Validation cross-references multi-page drawings to verify completeness and correctness.

[1] Exact figures vary widely by project complexity and remain proprietary.

Table 3.11: Common Error Types in Manual Workflow

Error Type	Description
Incorrect cable naming	Naming rules applied incorrectly or equipment associations misidentified
Missing connections	Incomplete drawing analysis or overlooked conductor paths
Terminal misidentification	Terminal labels misread or terminal functions misunderstood
Inconsistent panel identifier assignment	Panel classification rules applied differently across projects

Source: Bragalux operational data

Workflow bottleneck analysis reveals that the 2–4 hour duration stems from three primary causes. **Cross-referencing overhead**: E-REDES drawings span multiple pages, ranging from 6 to 26 pages per panel in the project template (Table 3.1), requiring manual navigation between pages and between panels to verify connections, identify equipment associations, and confirm terminal assignments. Each cable connection may refer to terminals on different pages, requiring iterative page navigation that accumulates time overhead. **Rule lookup complexity**: E-REDES naming rules require evaluating multiple conditions—voltage level, phase association, panel classification, equipment associations—that are not immediately obvious from drawing geometry. The manual application requires referencing standards documents to verify panel identifiers and naming patterns, creating cognitive overhead that slows processing. **Validation iteration**: Manual validation requires cross-checking cable names against naming rules, verifying terminal assignments against connection lists, and ensuring documentation completeness. Errors discovered during validation require correction and re-validation, creating iterative cycles that extend processing time. Automation addresses these bottlenecks through programmatic cross-referencing via spatial indexing for fast terminal lookup, algorithmic rule application with naming rules implemented as code to eliminate lookup overhead, and automated validation through systematic comparison against reference data for objective verification without iteration.

Common errors include incorrect cable naming, missing connections, terminal misidentification, and inconsistent panel identifier assignment. Table 3.11 summarizes these types of errors.

Incorrect cable naming occurs when the naming rules are applied incorrectly or when equipment associations are misidentified. Missing connections result from incomplete drawing analysis or overlooked conductor paths. Terminal misidentification occurs when terminal labels are misread or when terminal functions are misunderstood.

Inconsistent panel identifier assignment occurs when panel classification rules are applied differently across projects or when inter-panel assignment rules are misinterpreted.

These dependencies compound the time requirements and increase the probability of errors, creating economic motivation for automation.

The final section of this chapter synthesizes automation opportunities into explicit automation requirements.

3.7 Automation Requirements Derivation

The E-REDES standards presented above, combined with the workflow challenges identified in the previous section, define the automation requirements presented below.

High-priority requirements address the most time-consuming manual tasks:

- Terminal block extraction enables connection list generation.
- The panel identifier assignment logic, including the evaluation of the equipment association and inter-panel assignment rules, enables the LV cable naming.
- Implementation of the HV & MV and LV naming patterns enables automated cable naming.

Medium-priority requirements enhance automation completeness:

- Conductor color identification supports validation and connection verification.
- Cable type selection based on circuit function completes cable schedule generation.
- Documentation formatting according to structural requirements ensures compliance with Bragalux project templates.
- Validation of naming syntax compliance, panel identifier validity, and voltage code correctness enables automated quality checking.

The next chapter presents the system architecture and methodology addressing these requirements through PDF extraction, name generation algorithms, validation frameworks, and report generation templates.

Chapter 4

System Architecture and Methodology

This chapter presents the system architecture that addresses the requirements through PDF-based extraction, multi-stage validation algorithms, naming rule implementation, and formatted report generation. The architecture translates the E-REDES standards into algorithmic components, enabling automated terminal extraction, route detection, and cable naming while maintaining compliance with utility-specific rules documented in the previous chapter.

The system processes CAD drawings through a PDF-based workflow: terminal extraction produces E-REDES-compliant terminal lists; route detection and cable naming implement E-REDES grammars and route-terminal matching.

The architecture of the system consists of two core components: terminal extraction and CAD to PDF conversion. Terminal extraction addresses the high-priority requirement for terminal block extraction, enabling the generation of connection lists. Automatic conversion of CAD drawings to PDF allows automated processing without CAD software dependencies. Route detection and cable naming implement geometric path extraction and E-REDES naming rules.

4.1 System Overview

The system processes CAD drawings through five sequential stages: format conversion enables PDF-based processing; text extraction provides raw data from drawing annotations; terminal identification applies multi-stage validation to extract terminal blocks; route extraction performs geometric path identification; and output generation creates formatted documentation. This pipeline architecture enables modular processing where each stage operates on standardized data structures, allowing independent optimization and error isolation.

Figure 4.1 illustrates the entire system architecture, showing component relationships, data flow, and success indicators.

Drawing (DWG) files require conversion to PDF using COM interface automation. PDF files are processed directly. The unified interface routes files to appropriate processors according to file extension.

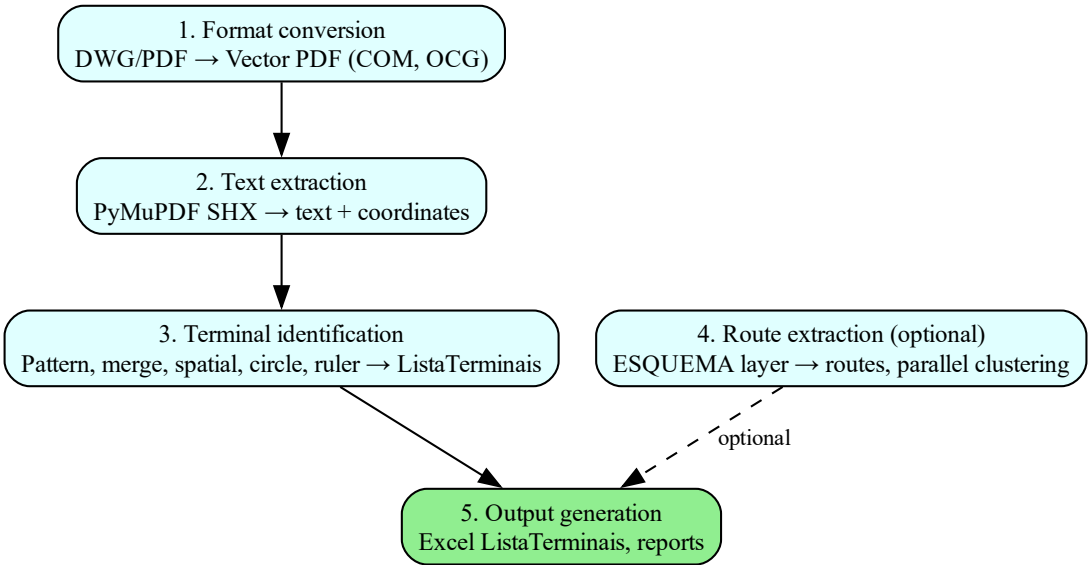


Figure 4.1: System Architecture Overview

The primary components include terminal extraction and conversion from CAD to PDF. Terminal extraction identifies and catalogs terminal blocks from drawing text through multi-stage validation. The conversion of CAD to PDF enables batch processing of DWG files without manual CAD software interaction, preserving the vector information required for automated analysis.

Secondary components include route detection and cable naming. Route detection extracts geometric paths representing conductor routes from drawing layers. Cable naming implements E-REDES grammars and route-terminal matching.

Terminal extraction operates independently, generating terminal lists without requiring route or cable information. CAD to PDF conversion functions as a pre-processing stage, allowing downstream components to process vector PDF. Route detection and cable naming form an integrated pipeline that implements route-terminal matching for cable identification.

4.1.1 Design Rationale

The architecture employs a modular pipeline design to address the automation challenges identified in the previous chapter. Multi-stage validation was chosen for terminal extraction over single-stage pattern matching because not every text matching terminal pattern is actually a terminal—page numbers, equipment codes, and grid labels produce false positives in simple pattern matching. Alternative approaches include single-stage pattern matching, which would be faster but would have higher false positive rates, and machine learning classification, which requires training data and may not generalize across drawing styles [2].

The terminal extraction architecture is grounded in pattern recognition theory, where multi-stage validation implements a hierarchical classification strategy: pattern matching provides initial candidate identification, content filtering removes false positives through negative pattern matching, spatial validation applies geometric constraints, and circle detection provides geometric confirmation. This hierarchical approach follows established pattern recognition principles where multiple weak classifiers combine to achieve strong classification performance, reducing false positive rates while maintaining high recall. Spatial indexing optimization employs grid-based spatial data structures, enabling efficient proximity queries through cell-based partitioning that reduces search complexity from O(n×m) to O(n×log m) for n terminals and m circles. Route-terminal matching applies graph theory concepts, where routes and terminals form a bipartite graph with distance-weighted edges, and matching algorithms identify optimal associations through distance minimization and bundle clustering. These theoretical foundations justify the architectural choices: pattern recognition theory supports multi-stage validation effectiveness, spatial indexing theory enables performance optimization, and graph theory provides the mathematical framework for route-terminal association. Multi-stage validation was chosen because it enables independent optimization of each validation criterion while maintaining interpretability and avoiding training data requirements, addressing the limitations of ML approaches identified in the literature review. Multi-stage validation separates concerns into independent stages (pattern recognition, content filtering, spatial validation, ruler assignment, and circle detection), enabling independent optimization of each stage and reducing false positive rates.

PDF-based processing was selected over CAD-native approaches to enable batch processing without requiring CAD software to be frozen during processing. Alternative approaches include native CAD processing, which requires the CAD software to be frozen during processing and prevents technicians from working simultaneously [8], and raster image processing, which loses vector precision as image-based approaches [3], [17] lose vector geometric information when rasterizing drawings. PDF-based processing was chosen because it preserves vector information while enabling batch processing of multiple drawings simultaneously without freezing CAD software, addressing PDF-based workflow integration. Vector PDF preservation maintains coordinate precision required for spatial validation while enabling the parallel processing of multiple drawings. The tolerance values in Tables 4.1 and 4.2 were calibrated through iterative testing on the SE Pracana dataset (see Section 5.5), balancing coverage (minimizing false negatives) against accuracy (minimizing false positives), as detailed in the calibration methodology.

The modular architecture enables flexible pipeline execution: terminal extraction can be completed without route extraction or cable naming, addressing the terminal identification challenge identified earlier. This design choice prioritizes ensuring all terminals are identified for technician review over requiring complete pipeline success.

4.2 CAD to PDF Conversion Pipeline

The CAD to PDF conversion pipeline enables automated processing of DWG files. The manual conversion of hundreds of drawings per project creates significant time overhead, consuming 30 minutes to 1 hour per integrated project intervention. The conversion pipeline automates this process using COM interface automation through Python's `win32com` library, enabling programmatic control of CAD software. While the conversion process requires CAD software to be frozen, once PDF are generated, subsequent processing—terminal extraction, route detection, cable naming—operates on PDF files without requiring CAD software to be frozen, allowing technicians to continue working in CAD during processing. Success depends on reliable COM interface access, proper layer preservation, vector information retention, and drawings with consistent dimensions and layer organization.

4.2.1 Format Detection and Routing

The system supports two input formats: DWG and PDF. Format detection occurs through file extension analysis. The DWG files are routed to the conversion pipeline. PDF files are routed directly to format-specific readers.

DWG files require conversion because reliable libraries for direct text extraction from this format are unavailable, and native CAD processing prevents parallel work. The PDF format enables SHX text extraction through PyMuPDF [29], which reads text content from PDF annotations embedded by AutoCAD during export. SHX, compiled shape font format, refers to the font format used by AutoCAD-based software for technical drawings, stored as PDF annotations rather than standard text objects. The unified reader interface presents a consistent data structure to downstream components.

4.2.2 Vector Extraction Methods

Conversion of CAD to PDF preserves vector information critical for automated analysis. Vector preservation enables accurate coordinate extraction for terminal positions, route geometries, and spatial relationships. Rasterization would lose coordinate precision and reduce spatial validation effectiveness.

The conversion pipeline uses COM interface automation to programmatically control CAD software. The architecture supports multiple CAD software platforms through programmatic interfaces. Implementation details, including version detection and connection management, are documented in the next chapter.

Vector preservation occurs through CAD software plot-to-PDF functionality. The COM interface accesses the plot configuration objects, enabling programmatic control of the plot area, scale, and output settings. Plot area selection supports multiple modes: display extents, drawing limits, named views, window selection, and layout-based plotting. Window selection enables user-defined plot regions for multi-page drawings requiring specific page extraction.

4.2.3 Layer Preservation Techniques

Preservation of the layer structure maintains the drawing organization required for component-specific processing. E-REDES drawings organize content into layers: "TEXTO" for text elements, "ESQUEMA" for route geometries, "Formato+leg" for drawing annotations and borders. Layer preservation enables format-specific readers to filter content by layer, improving extraction accuracy.

The conversion pipeline preserves layer information through the CAD software layer export settings. The COM interface configuration maintains the layer visibility during PDF generation. The PDF format represents the layers as Optional Content Groups, OCG—PDF structures representing CAD layers that enable programmatic layer access in downstream processing. Format-specific readers detect layer names through OCG enumeration, returning to content-based detection when explicit layers are unavailable.

4.2.4 Output Generation

Batch processing capabilities enable conversion of entire project folders without manual intervention. The conversion pipeline processes all DWG files in a specified directory, generating the corresponding PDF files in an output subfolder.

Error handling enables robust batch processing: individual file conversion issues do not abort batch processing. Conversion issues log details while successful conversions continue. Validation verifies PDF generation success through file existence and size checks, ensuring output quality automatically.

4.3 Terminal Extraction Architecture

Terminal extraction forms the core component of the system, addressing the high-priority automation requirement for terminal block extraction that enables connection list generation. The architecture employs multi-stage validation combining pattern recognition, spatial analysis, and rule-based inference. Terminal identification follows the E-REDES terminal strip functions and numbering formats documented in the previous chapter.

Terminal extraction processes drawings through six sequential stages, employing a multi-stage validation strategy that separates concerns and enables independent optimization of each stage. Stage 1, text extraction, captures all text elements from PDF format. Stage 2, pattern recognition, identifies terminal candidates using E-REDES formats. Stage 3, text merging, combines fragmented labels that export as separate text elements. Stage 4, spatial validation, confirms geometric relationships through circle detection, reducing false positives from annotation text. Stage 5, ruler assignment, ensures complete terminal data through spatial matching or inference. Stage 6, terminal list generation, creates structured Excel output.

This multi-stage approach addresses the challenge that not every text matching terminal patterns is actually a terminal—page numbers, equipment codes, and grid labels can produce false

positives in simple pattern matching. Multi-stage validation was chosen over single-stage pattern matching because it enables independent optimization of each validation criterion, reducing false positive rates while maintaining high recall. Success depends on accurate pattern recognition, effective text merging, reliable spatial validation, and correct ruler assignment. The tolerance parameters in Table 4.1 control the spatial matching, text merging, and validation thresholds. These tolerances were calibrated through iterative testing to balance coverage against accuracy, as detailed in the implementation chapter. Success requires drawings following E-REDES layer conventions and terminal labeling standards.

4.3.1 Text Extraction from PDF

PDF text extraction uses PyMuPDF to read SHX text annotations. AutoCAD exports SHX fonts as PDF annotations with the title "AutoCAD SHX Text" when the PDFSHX system variable is enabled, PDFSHX=1. Each annotation contains text content in its metadata along with bounding box coordinates for spatial positioning. This approach extracts text directly from embedded metadata in CAD software rather than parsing vector geometry.

The annotation iteration filters for "AutoCAD SHX Text" titles, extracting text content and position coordinates from annotation metadata. This method preserves the exact text strings as defined in the source CAD drawing, eliminating OCR errors and font interpretation issues.

PDF extraction generates raw text lists that contain all text elements before filtering. Raw extraction maintains equipment label detection separately from terminal extraction. Terminal filtering removes equipment labels as non-terminals, but equipment detection requires access to unfiltered text. The architecture maintains dual text streams: raw text for equipment detection and filtered text for terminal identification.

4.3.2 Terminal Pattern Recognition

Regular expression matching was chosen over machine learning classification for two reasons. First, E-REDES terminal formats follow deterministic patterns defined by standards and in Table 3.6, making rule-based matching more appropriate than statistical learning. Second, regular expressions provide exact pattern matching with near 100% precision for valid patterns, while ML approaches require training data and may misclassify edge cases. Pattern recognition occurs before spatial validation to filter terminal candidates early in the pipeline, reducing computational complexity of subsequent validation stages.

Algorithm 1 formalizes the terminal pattern recognition process. The algorithm implements a hierarchical pattern matching strategy that processes text elements through three sequential stages: formatted terminal patterns containing explicit ruler codes and terminal numbers such as “-X5.2:501”; unformatted number patterns requiring ruler inference such as “501”; and special case patterns such as placeholder terminals e.g. “?”. This hierarchical approach separates terminal labels from page numbers, equipment labels, and other drawing annotations by prioritizing explicit patterns over inferred ones.

The algorithm processes each text element once, performing pattern matching operations with time complexity O(1) per element using regular expressions. The overall time complexity is O(n) where n is the number of text elements. The spatial complexity is O(n) to store the classification results.

Algorithm 1 Terminal Pattern Recognition

Require: Text element t, Configuration c
Ensure: Terminal classification or $\emptyset$

```
Extract ruler pattern: r ← MatchRulerPattern(t, c)
if r ≠ ∅ then
    Extract terminal number: n ← MatchTerminalNumber(t, r, c)
    if n ≠ ∅ then
        return Formatted terminal (r, n)
    end if
else
    Check unformatted number: n ← MatchUnformattedNumber(t, c)
    if n ≠ ∅ then
        return Unformatted terminal requiring inference
    else
        Check special cases: s ← MatchSpecialCase(t, c)
        if s ≠ ∅ then
            return Special case terminal s
        end if
    end if
end if
return ∅ (not a terminal)
```

Formatted terminal patterns match ruler-terminal combinations following E-REDES conventions. Patterns recognize X0.x sub-rulers X0.1 through X0.7, main rulers X1, X2, X3, X5, X6, X7, and X5.x sub-rulers X5.1 through X5.6. Terminal numbers follow ruler-specific ranges: X0.x terminals use 1–99, X1 uses 101–199, X2 uses 201–299, X3 uses 301–499, X5.x uses 501–599, X6 uses 601–699, and X7 uses 701–799.

Unformatted number patterns match digits without ruler prefixes. These require ruler inference through spatial matching or number-based rules. Single- and double-digit numbers 1–99 map to X0.x sub-rulers through spatial proximity matching. Three-digit numbers apply first-digit rules: 1XX maps to X1, 2XX maps to X2, 3XX and 4XX map to X3, 5XX maps to X5.x sub-rulers, 6XX maps to X6, and 7XX maps to X7.

Special case patterns handle edge cases: "?" represents unknown terminals in external connections, "-X?" represents ruler labels without terminal numbers, and formatted patterns with missing components require validation. Pattern recognition classifies text elements but does not validate terminal validity; spatial validation and ruler assignment occur in subsequent stages.

4.3.3 Text Merging and Validation

Fragmented terminal labels split across multiple text elements require merging. Terminal labels sometimes export as separate text strings: ruler labels such as “X5.2” and terminal numbers such as “501” appear as distinct elements despite representing a single terminal. For example, when "X5.2" and "501" appear as separate text elements within 30 drawing units, the algorithm merges them into "-X5.2:501". Distance-based clustering identifies spatially proximate text elements that match terminal patterns and combines them into complete terminal labels.

Merging employs distance-based clustering with tolerance thresholds. Text elements within the merging tolerance undergo pattern compatibility testing. Compatible patterns include ruler-terminal pairs, number-number pairs that require ruler inference, and formatted-unformatted combinations. Merging preserves coordinate information from the primary text element, typically the ruler label when present.

Validation rules exclude component labels such as relay identifiers and equipment codes, title block text, and annotation elements. Spatial filtering removes text outside of the schematic drawing areas, excluding borders, headers, and footers. Content filtering uses negative patterns to reject common non-terminal text: equipment identifiers, panel codes, and measurement units.

Ruler assignment validation ensures that all terminals receive correct ruler associations. Formatted terminals extract rulers directly from text patterns. Unformatted numbers require spatial matching to nearby ruler labels or inference through number-based rules. Ruler assignments are required before inclusion in output lists.

Algorithm 2 formalizes the text merging process. The algorithm clusters spatially proximate text elements within tolerance thresholds from Table 4.1, then applies pattern compatibility rules to determine valid merges. Compatible patterns include ruler-terminal pairs e.g. “X5.3” + “502” $\rightarrow$ “X5.3:502”, number-number pairs requiring ruler inference, and formatted-unformatted combinations. Distance-based clustering was chosen over semantic analysis because terminal labels lack semantic context, making geometric proximity the primary signal for identifying fragmented labels. Alternative approaches include semantic analysis, which would require training data and may not generalize across different drawing styles, and rule-based inference, which is fragile to layout variations. Distance-based clustering was chosen because it operates on geometric relationships that are consistent across E-REDES drawing styles. The 30-unit merging tolerance was calibrated through iterative testing on the SE Pracana dataset, starting with conservative values and incrementing until false positive rates exceeded acceptable thresholds, as detailed in the implementation chapter.

The algorithm uses distance-based clustering with the worst-case complexity of $O(n^2)$ for n text elements, as each element must be compared with all others within tolerance. However, spatial indexing reduces this to O(n log n) average case. The space complexity is O(n) for storing merged labels and processed element sets.

Algorithm 2 Text Merging Algorithm

Require: Text elements $T = \{t_1, t_2, ..., t_n\}$, Merging tolerance τ
Ensure: Merged terminal labels M
Initialize merged labels $M \leftarrow \emptyset$
Initialize processed set $P \leftarrow \emptyset$
for each text element $t_i \in T$ **do**
  if $t_i \notin P$ **then**
    Find nearby elements: $N \leftarrow \{t_j : \text{distance}(t_i, t_j) \leq \tau, t_j \notin P\}$
    if $N \neq \emptyset$ **then**
      Check pattern compatibility: $compatible \leftarrow \text{PatternCompatible}(t_i, N)$
      Pattern compatibility rules: (1) ruler-terminal pairs: one element matches ruler pattern (e.g., "X5.2"), another matches terminal number pattern (e.g., "501"); (2) number-number pairs: both elements match unformatted number patterns requiring ruler inference (e.g., "501" + "502"); (3) formatted-unformatted: one element matches formatted pattern (e.g., "-X5.2:501"), another matches unformatted pattern (e.g., "502")
      if $compatible = \text{true}$ **then**
        Merge t_i and compatible elements into merged label m
        Add m to M
        Mark t_i and merged elements as processed: $P \leftarrow P \cup \{t_i\} \cup N$
      else
        Add t_i as standalone element to M
        $P \leftarrow P \cup \{t_i\}$
      end if
    else
      Add t_i as standalone element to M
      $P \leftarrow P \cup \{t_i\}$
    end if
  end if
end for
return Merged labels M

4.3.4 Spatial Validation and Circle Detection

Geometric confirmation of terminal validity occurs through the detection of circle symbols. E-REDES drawings represent terminals with circle symbols surrounding terminal labels. PDF processing uses path analysis to identify these symbols and validate the proximity of the terminal text to the circle centers. This validation reduces false positives from annotation text matching terminal patterns.

PDF processing analyzes drawing paths, identifying closed circular paths within terminal symbol size ranges. Detection caches circle coordinates in a spatial index per page, enabling O(log n) lookup during terminal validation and avoiding redundant processing.

Distance calculations associate terminals with circles through spatial matching. Conditional validation distinguishes terminal types: X0.x terminals require nearby X0.x ruler labels for validation, while other terminals use circle detection when available but do not require it for validity. Only these terminal rulers have this conditional validation due to the frequent appearance of false

positives, as the usage of single- and double digit numbers is much more common than triple digit from the other rulers.

A multi-stage validation strategy separates concerns into independent validation stages to address the challenge that not every text matching terminal patterns is actually a terminal, as identified in the problem statement. Page numbers, equipment codes, and grid labels can produce false positives in simple pattern matching, requiring additional validation beyond pattern recognition alone. This multi-stage approach was chosen over single-stage pattern matching because it enables independent optimization of each validation criterion, as detailed in the design rationale above.

Algorithm 3 implements a five-stage filter system that processes terminal candidates sequentially. The algorithm applies five validation stages: pattern format compliance verification against E-REDES terminal formats; content filtering to exclude equipment labels and annotation text; spatial location validation within schematic drawing areas; ruler assignment through proximity matching or inference; and geometric confirmation through circle detection when available. Only terminals passing all applicable stages are accepted. This sequential filtering approach enables independent optimization of each validation criterion, significantly reducing false positive rates while maintaining high recall, as validated in the results chapter.

Each validation stage processes terminals sequentially with O(1) operations per terminal for pattern and content validation. Spatial validation and ruler assignment use spatial indexing, achieving O(log m) complexity where m is the number of rulers or circles. Circle detection uses cached results, which require an O(1) lookup. The overall time complexity is O(n log m) for n terminals and m geometric elements. The space complexity is O(n + m) for storing terminals and geometric validation data.

Table 4.1 lists exact tolerance values from the system configuration, calibrated through iterative testing on the SE Pracana dataset These parameters control the spatial matching, text merging, and validation thresholds. The calibration process measured the distances between the terminal text labels and their associated circle symbols, with the 12-unit circle detection tolerance accommodating 98% of valid terminals while excluding annotation text that matches terminal patterns but lacks geometric validation. These tolerance values were selected through systematic range testing: starting with conservative values (50% of measured distances), incrementing until false positive rates exceeded 5% and then selecting values that balanced coverage (minimizing false negatives) against accuracy (minimizing false positives). The 12-unit circle detection tolerance, for example, was chosen because it accommodates 98% of valid terminals while maintaining a false positive rate below 2%.

Table 4.2 documents tolerance parameters for route detection and matching. These parameters control route extraction, parallel route identification, and route-terminal association.

4.3.5 Terminal List Generation

The generation process aggregates terminals on all drawing pages, groups by ruler for organization, and exports to Excel format. The output includes terminal identifiers, ruler assignments,

Table 4.1: Terminal Extraction Tolerance Parameters

Parameter	Value (drawing units)	Purpose
Terminal merging tolerance	30	Text element merging distance
Circle detection tolerance	12	Terminal circle proximity validation
Circle max radius	3	Terminal symbol size validation
Spatial matching tolerance	12	Terminal-circle association
PDF circle detection tolerance	5	PDF-specific circle detection
Schematic area Y minimum	100	Exclude page headers
Schematic area Y maximum	700	Exclude page footers
Terminal route tolerance	100	Terminal-route association

Source: Configuration file `autocable.yaml`. Drawing units correspond to millimeters in E-REDES standard drawings.

Table 4.2: Route Detection Tolerance Parameters

Parameter	Value (drawing units)	Purpose
Route-terminal snap tolerance	150	Route-terminal matching distance
External snap tolerance	500	External cable route-terminal matching
Parallel route tolerance	5	Parallel route grouping distance
Minimum route length	5	Filter noise routes (short segments)
Maximum route length	500	Filter border elements (long segments)
Equipment zone radius	500	Base radius for equipment zone detection

Source: Configuration file `autocable.yaml`. Drawing units correspond to millimeters in E-REDES standard drawings.

Algorithm 3 Multi-Stage Terminal Validation

Require: Merged terminals $M = \{m_1, m_2, ..., m_n\}$, Configuration c
Ensure: Validated terminals V

```
Initialize validated terminals V ← ∅
for each terminal m_i ∈ M do
    Stage 1: Pattern validation
    if PatternValid(m_i, c) = false then
        continue
    end if
    Stage 2: Content validation
    if ContentValid(m_i, c) = false then
        continue
    end if
    Stage 3: Spatial validation
    if SpatialValid(m_i, c) = false then
        continue
    end if
    Stage 4: Ruler assignment
    r ← AssignRuler(m_i, c)
    if r = ∅ then
        continue
    end if
    Stage 5: Circle detection (optional for X0.x, required for others when available)
    if RequiresCircle(m_i) then
        if CircleDetected(m_i, c) = false then
            continue
        end if
    end if
    Add validated terminal: V ← V ∪ {(m_i, r)}
end for
return Validated terminals V
```

page locations, and coordinates.

Terminal aggregation collects all validated terminals from multi-page drawings. Page numbering follows zero-indexed convention matching PDF page enumeration. Terminal data structures include text content, position coordinates, ruler assignments, and page numbers.

Linear grouping organizes terminals for human review and validation. Grouping simplifies manual verification workflows by organizing terminals according to the E-REDES ruler hierarchy. Within groups, terminals are classified by terminal number for sequential review.

Export includes terminal identifiers in E-REDES format (ruler:number, e.g., "X5.2:501"), page references for drawing traceability and coordinate data for spatial validation.

Terminal extraction provides terminal block identification, but complete automation requires route detection to identify cable paths connecting terminal blocks.

4.4 Route Detection Architecture

Route detection extracts geometric paths representing conductor routes from drawing layers, addressing the requirement for cable path identification connecting terminal blocks. This component enables downstream cable naming automation. Geometric analysis was chosen over semantic understanding because E-REDES drawings encode cable connections through spatial relationships (routes connect different terminals) rather than explicit semantic annotations, making geometric proximity the primary signal for route-terminal association.

4.4.1 Route Extraction Methods

Entity filtering isolates route geometries from annotation elements, text labels, and border graphics. Route extraction processes LINE and LWPOLYLINE entities from ESQUEMA layers in PDF format. Filtering applies two criteria: layer name matching and geometric characteristics (length validation).

Algorithm 4 formalizes the route extraction filtering process, addressing the challenge of identifying conductor routes among thousands of line entities in E-REDES drawings. The algorithm applies layer and length filters to identify valid conductor routes, as E-REDES drawings organize routes into ESQUEMA layers separate from text and annotation layers Drawings contain thousands of line entities, but only a subset represent actual cable routes. Other lines represent borders, dimension lines, or annotation elements, requiring filtering to isolate valid routes. The algorithm implements a two-stage filtering strategy based on E-REDES drawing conventions: First, it filters entities by layer name, selecting only those from the route layer (ESQUEMA). Second, it validates geometric characteristics, specifically route length, using thresholds from Table 4.2. The length filter removes very short segments (likely annotation fragments) and excessively long segments (likely border elements), calibrated through testing. This filtering strategy isolates valid conductor routes from drawing noise, enabling a more accurate route-terminal association for cable naming (Section 4.5).

The algorithm processes each entity once, performing O(1) layer name comparison and O(1) length calculation per entity. The overall time complexity is O(n) for n entities. The spatial complexity is O(k), where k is the number of filtered routes, typically much smaller than n.

PDF route extraction analyzes path objects within specified layers. PyMuPDF provides path iteration with coordinate extraction, enabling geometric analysis of route segments. Path analysis identifies *polyline* structures representing multi-segment routes. Coordinate extraction preserves vector precision, maintaining the spatial relationships required for terminal matching.

Length filtering removes noise routes: very short segments likely represent annotation elements rather than conductor paths. The length thresholds exclude segments under 5 drawing units (Table 4.2). This filtering reduces false positive routes, but may exclude valid short connection segments in dense terminal areas.

Algorithm 4 Route Extraction Filtering

Require: Entities $E = \{e_1, e_2, ..., e_n\}$, Layer name L, Configuration c
Ensure: Filtered routes R
Initialize filtered routes $R \leftarrow \emptyset$
Extract thresholds: $\tau_{min} \leftarrow c$.min_route_length, $\tau_{max} \leftarrow c$.max_route_length
for each entity $e_i \in E$ **do**
  if e_i.layer $\neq L$ **then**
    continue
  end if
  if e_i.type $\notin \{\text{LINE}, \text{LWPOLYLINE}\}$ **then**
    continue
  end if
  Calculate length: $\ell \leftarrow \text{CalculateLength}(e_i)$
  if $\tau_{min} \leq \ell \leq \tau_{max}$ **then**
    Add to routes: $R \leftarrow R \cup \{e_i\}$
  end if
end for
return Filtered routes R

4.4.2 Layer-Based Filtering

E-REDES drawings organize routes into ESQUEMA layers, separate from text (TEXTO) and annotation layers. Layer detection employs name matching against common layer naming conventions: "ESQUEMA" or "Esquema". When explicit layers are unavailable, the system falls back to content-based filtering.

Drawings following E-REDES layer standards enable reliable filtering. Drawings with non-standard layer organization or merged layers employ content-based analysis for route identification.

4.4.3 Parallel Route Identification

Grouping routes representing multi-conductor cables requires parallel route identification, as E-REDES cables contain multiple conductors bundled together E-REDES cables contain multiple conductors, typically 3–4 for power circuits and 2–4 for control circuits, and parallel routes with similar start and end regions likely represent conductors within the same cable bundle. Algorithm 5 formalizes the parallel route clustering process. For example, a three-phase power cable contains three parallel conductors represented as separate route entities in the drawings, requiring grouping to identify cable bundles. The algorithm identifies routes that run parallel and close together, grouping them into cable bundles representing multi-conductor cables. The algorithm uses dual-criteria matching: spatial proximity (perpendicular distance within tolerance from Table 4.2) and directional similarity (angle difference within tolerance). This approach prevents false grouping of perpendicular routes that happen to intersect, addressing geometric ambiguity in route identification. Routes identified as parallel are grouped using transitive closure, ensuring that all routes in a bundle are grouped together even if not all directly parallel to each other. Dual-criteria

matching was chosen over single-criteria (distance-only) because angle validation prevents false grouping of perpendicular routes that happen to be spatially close, significantly reducing false positive bundle assignments in dense route networks.

The algorithm compares each route with all the others, requiring $O(n^2)$ distance and angle calculations for n routes. However, spatial indexing can reduce this to O(n log n) average case. The transitive closure grouping adds $O(n^2)$ worst-case complexity for dense route networks. The overall time complexity is $O(n^2)$ in the worst case, O(n log n) with spatial indexing. The space complexity is O(n) for storing route clusters.

The proximity of the endpoint within tolerance thresholds enables clustering algorithms to group routes. Tight tolerances miss valid parallel groups with endpoint variation. Loose tolerances group unrelated routes sharing general regions. Optimal tolerance selection is calibrated for E-REDES standardized project types, ensuring consistent performance across projects following the same standards.

Parallel identification enables conductor count estimation for cable grouping. Route groups with N parallel routes suggest N-conductor cables. Accuracy depends on route extraction completeness—specifically, identifying all route segments within a cable bundle.

4.4.4 Route Validation and Geometric Analysis

Validation employs multiple matching strategies to associate routes with terminals. Multiple strategies were chosen over a single matching approach because E-REDES drawings following standardized project types may exhibit different geometric patterns: some use explicit stub lines, others use route curvature, and some employ alternative connection indicators. Multiple strategies provide robustness across these pattern variations within E-REDES standardized project types.

Route endpoint analysis considers geometric relationships between routes and terminal blocks. Route-terminal matching uses distance-based association with snap tolerance from Table 4.2; when multiple terminals fall within that range, zone-based and bundle-first strategies provide additional context. Stub line detection identifies perpendicular connection segments, with stub geometry that varies between drawings. Some drawings use explicit stub lines, others use route curvature, and some employ alternative connection indicators. Distance-based matching selects the nearest terminals within tolerance, with zone-based and bundle-first matching providing additional context when there are multiple candidates.

Route-terminal matching employs geometric analysis for route-terminal association. Multiple matching strategies address geometric relationships: distance-based matching for proximity, zone-based matching around ruler labels for spatial context, and bundle-first matching for parallel route groups. Bundle-first matching is prioritized when parallel routes are detected because multi-conductor cables (represented as parallel routes) should be matched as groups rather than individually, improving accuracy for cable bundle identification. Matching strategies include route endpoint analysis, stub line detection, and geometric interpretation of E-REDES drawing conventions.

Algorithm 5 Parallel Route Clustering

Require: Routes $R = \{r_1, r_2, ..., r_n\}$, Configuration c
Ensure: Route clusters $C = \{C_1, C_2, ..., C_m\}$ where each C_i contains parallel routes
Initialize clusters $C \leftarrow \emptyset$
Extract thresholds: $\tau_{angle} \leftarrow c$.angle_tolerance, $\tau_{prox} \leftarrow c$.parallel_route_tolerance, $\tau_{endpoint} \leftarrow c$.max_endpoint_distance
for each route $r_i \in R$ **do**
    Calculate route angle: $\theta_i \leftarrow \text{CalculateAngle}(r_i.\text{start}, r_i.\text{end})$
    Find compatible cluster: $cluster \leftarrow \text{FindCompatibleCluster}(r_i, C, \tau_{angle}, \tau_{prox}, \tau_{endpoint})$
    if $cluster \neq \emptyset$ **then**
        Add route to cluster: $cluster \leftarrow cluster \cup \{r_i\}$
    else
        Create new cluster: $C \leftarrow C \cup \{\{r_i\}\}$
    end if
end for
function FindCompatibleCluster$(r, C, \tau_{angle}, \tau_{prox}, \tau_{endpoint})$
for each cluster $C_j \in C$ **do**
    if $\forall r_k \in C_j$: AreParallel$(r, r_k, \tau_{angle}, \tau_{prox}, \tau_{endpoint})$ **then**
        return C_j
    end if
end for
return $\emptyset$
function AreParallel$(r_1, r_2, \tau_{angle}, \tau_{prox}, \tau_{endpoint})$
Calculate angle difference: $\Delta\theta \leftarrow |\text{Angle}(r_1) - \text{Angle}(r_2)|$
if $\Delta\theta > \tau_{angle}$ **then**
    return false
end if
Calculate perpendicular distance: $d_{perp} \leftarrow \text{PerpendicularDistance}(r_1, r_2)$
if $d_{perp} > \tau_{prox}$ **then**
    return false
end if
Calculate endpoint distance: $d_{endpoint} \leftarrow \min(\text{Distance}(r_1.\text{start}, r_2.\text{start}), \text{Distance}(r_1.\text{end}, r_2.\text{end}))$
if $d_{endpoint} > \tau_{endpoint}$ **then**
    return false
end if
return true
return Clusters C

Table 4.3: Equipment-to-Panel Mapping Rules

Panel Type	Equipment Association	Panel Identifier Pattern
60kV Panel	AT equipment in panel	P5NP (NP = 2-digit panel number)
30kV Panel	MT equipment in panel	P3NP (NP = 2-digit panel number)
15kV Panel	MT equipment in panel	P2NP (NP = 2-digit panel number)
10kV Panel	MT equipment in panel	P1NP (NP = 2-digit panel number)
AC Auxiliary Services	SACA equipment	SACA
DC Auxiliary Services	SACC equipment	SACC
Counting	CONT equipment	CONT
Communications	COM equipment	COM
Central Processing Unit	UPC equipment	UPC
Intrusion Detection	SDIT equipment	SDIT
Fire Detection	SDIC equipment	SDIC
Technical Service Quality	SQST equipment	SQST

Source: E-REDES standards

4.5 Cable Naming Architecture

This section describes how cable names are generated from routes, terminals, and equipment The implementation achieves insufficient accuracy for production; causes and metrics are in the results chapter, implications in the conclusions. Below: route-terminal matching, equipment and panel logic, naming rules, and design limits.

4.5.1 Route-Terminal Matching

Cable names need origin and destination terminals; the drawing has no explicit links, only geometry. E-REDES encodes connections by proximity, so association is by distance: a route is linked to a terminal if its endpoint lies within snap tolerance (Table 4.2). When several terminals fall inside that range, the match is underdetermined; zone-based (ruler regions) and bundle-first (parallel route groups) add context but cannot replace semantic disambiguation.

4.5.2 Equipment and Panel Logic

The equipment identifiers are taken from the drawing text; panel classification uses the E-REDES rules to assign panel identifiers for LV names, W9 + number + panel id. Equipment zones are built by clustering text positions with radius 500 units from Table 4.2. The mapping of equipment-to-panel is given in Table 4.3; Algorithm 6 implements the search. Zone size is project-dependent and may need calibration per drawing set.

Algorithm 6 implements the mapping: equipment identifiers from text → equipment association → panel id via Table 4.3 Complexity: O(n) pattern matching, O(n log n) zone clustering, O(1) per lookup.

Routes near equipment zones get panel associations for LV naming; accuracy depends on zone and route-endpoint detection.

Algorithm 6 Panel Classification from Equipment Associations

Require: Equipment identifiers $E = \{e_1, e_2, ..., e_n\}$, Equipment zones Z, Mapping table M
Ensure: Panel identifiers P
Initialize panel identifiers $P \leftarrow \emptyset$
for each equipment identifier $e_i \in E$ **do**
  Find equipment zone: $z_i \leftarrow \text{FindZone}(e_i, Z)$
  if $z_i = \emptyset$ **then**
    continue
  end if
  Determine equipment association: $association \leftarrow \text{DetermineEquipmentAssociation}(e_i)$
  if $association = \emptyset$ **then**
    continue
  end if
  Apply panel classification rule: $panel \leftarrow \text{ClassifyPanel}(association, M)$
  if $panel \neq \emptyset$ **then**
    Add panel identifier: $P \leftarrow P \cup \{panel\}$
  end if
end for
return Panel identifiers P

4.5.3 Limitations of PDF-Based Connection Line Extraction

In PDFs, connection lines are vector geometry but are *fragmented* on export: continuous CAD paths become disconnected segments, so which segment belongs to which cable is unknown.

4.5.3.1 Empirical Evidence

Algorithm 7 measures how many route endpoints fall within tolerance of ground-truth terminal positions. On this dataset the resulting connection rate is zero: routes exist but cannot be attributed to terminal pairs.

Failure persists even with tolerances much larger than typical spacing (10–50 units): the bottleneck is fragmentation, not calibration. CAD keeps continuous paths with terminal references; PDF export turns them into segments without segment-relationship metadata, so geometric-only matching cannot reconstruct terminal-to-terminal paths.

4.5.3.2 Fragmentation and PDF Rendering

In export, continuous CAD paths become polylines, disconnected segments, and composite paths; the PDF keeps geometry but loses which segment belongs to which cable. The extraction correctly reads ESQUEMA entities; the limit is the lack of structural metadata in the PDF, not the extractor. Unlike label-extraction work (e.g. [2]), here names must be generated from geometry alone, so fragmentation is decisive.

Algorithm 7 Route-Terminal Connection Analysis

Require: Routes $R = \{r_1, r_2, ..., r_n\}$, Terminals $T = \{t_1, t_2, ..., t_m\}$, Tolerance τ
Ensure: Connection rate ρ, Routes near terminals $R_{connected}$

Initialize connected routes: $R_{connected} \leftarrow \emptyset$
Initialize endpoint count: $N_{endpoints} \leftarrow 0$
for each route $r_i \in R$ **do**
    Extract endpoints: $(p_{start}, p_{end}) \leftarrow \text{GetEndpoints}(r_i)$
    $N_{endpoints} \leftarrow N_{endpoints} + 2$
    Find terminals on same page: $T_{page} \leftarrow \{t_j \in T : \text{Page}(t_j) = \text{Page}(r_i)\}$
    Check start endpoint connection
    for each terminal $t_j \in T_{page}$ **do**
        $d \leftarrow \text{Distance}(p_{start}, \text{Position}(t_j))$
        if $d \leq \tau$ **then**
            $R_{connected} \leftarrow R_{connected} \cup \{r_i\}$
            break
        end if
    end for
    Check end endpoint connection (if not already connected)
    if $r_i \notin R_{connected}$ **then**
        for each terminal $t_j \in T_{page}$ **do**
            $d \leftarrow \text{Distance}(p_{end}, \text{Position}(t_j))$
            if $d \leq \tau$ **then**
                $R_{connected} \leftarrow R_{connected} \cup \{r_i\}$
                break
            end if
        end for
    end if
end for
Calculate connection rate: $\rho \leftarrow |R_{connected}|/N_{endpoints}$
return Connection rate ρ, Connected routes $R_{connected}$

4.5.3.3 Implications and CAD-Native Alternative

Terminal and route extraction work on the data; cable naming fails because the paths cannot be reconstructed. Production-grade naming would need either CAD-native connection data or non-purely geometric methods. CAD-native access would avoid fragmentation, but requires that CAD be reserved during processing the current design favors batch PDF processing and accepts this limit.

4.6 System Integration

A single reader exposes terminal and route data in the same structures for PDF and DWG Components call each other synchronously; terminals and routes are passed as structured data: positions, layers, metadata. No shared state beyond the pipeline sequence.

4.6.1 Pipeline and Data Flow

Order: format conversion → text extraction → terminal validation → output; optionally route extraction and cable naming run after terminals and feed output as well. Terminal extraction does not depend on routes or cable naming, so terminal lists can be produced even when route/cable stages fail or are skipped.

4.6.2 Errors and Propagation

A failing stage logs and, where possible, lets the rest run (e.g. terminal list still generated if route extraction fails). Errors propagate downstream: bad or missing terminals hurt cable naming; bad or missing routes or equipment hurt cable naming; terminal extraction output is not affected by route or cable errors. Ambiguity in route-terminal or panel assignment cannot be resolved by the current architecture.

Chapter 5

Implementation, Datasets, and Validation

This chapter documents implementation choices for the architecture in the previous chapter: technology stack, mapping of Algorithms 1–3 to code, tolerance calibration, datasets, and validation methodology.

5.1 Terminal Extraction Implementation

5.1.1 Technology Stack

The implementation uses Python 3.10 with PyMuPDF (1.23.0) as the primary PDF library, pdfplumber (0.10.0) and pdfminer.six (20221105) as fallbacks, and openpyxl (3.0.0) for Excel output. PyMuPDF was chosen for SHX font support (AutoCAD SHX text in PDF annotations), coordinate precision (bounding boxes for spatial validation), and performance; the tiered fallback covers edge cases. The COM interface (`win32com`) was chosen for DWG-to-PDF over native libraries because it uses the same plot engine as manual CAD, preserves OCG layers, and enables batch control; dependency on CAD matches production workflows.

5.1.2 PDF Text Extraction

The `PDFDatabase` class provides a unified PDF interface. SHX text is stored as PDF annotations rather than standard text; the extraction iterates over annotations and reads content from `content`, `title`, or `subject`, with `get_text()` as a fallback. Layer filtering uses OCG when available.

5.1.3 Terminal Pattern Recognition

Pattern recognition implements Algorithm 1 and classifies rulers , standalone numbers, and the "?" placeholder. Classification is based on regex patterns described in the methodology.

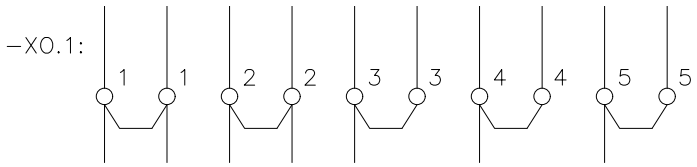


Figure 5.1: Row layout (schematic): ruler and terminal numbers on the same horizontal line.

5.1.4 Text Merging

Algorithm 2 is implemented as distance-based clustering in two passes. In *row layout* (X5.x, dy 5, dx 700) the ruler label (e.g. X5.3) lies on the left and the terminal numbers (501, 502, 503, ...) sit on the same horizontal line, so the algorithm merges them into X5.3:501, X5.3:502, etc. In *column layout* (X0.x, dx 50) the ruler is above or below with numbers stacked vertically (Figure 5.2). The algorithm uses greedy matching (closest compatible ruler per number); time complexity is O(n²) in the worst case and O(n log n) with spatial indexing. Table 5.1 gives the tolerance values used.

5.1.5 Tolerance Calibration

Calibration used the SE Pracana dataset in three phases. Phase 1 established a baseline from text-element distances (row dx up to 700, column dx 50). Phase 2 applied range testing from conservative values until the false positive rate exceeded 5%, with dy=5.0 chosen to balance false negatives and positives. Phase 3 validated against reference data, achieving a 98% correct merge rate. Sensitivity analysis was not performed; the methodology (parameter range, P/R/F1 per value, interaction analysis) is documented for future calibration (Table 4.1).

Table 5.1: Text merging tolerance values

Parameter	Layout	Value
Vertical tolerance (dy)	Row layout	5.0 units
Horizontal tolerance (dx)	Row layout	700.0 units
Horizontal tolerance (dx)	Column layout	50.0 units
General merge distance	Both	30.0 units

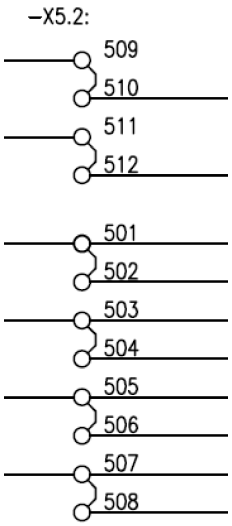


Figure 5.2: Column layout: ruler above or below vertically arranged terminal numbers, merged by proximity into ruler:number identifiers.

5.1.6 Spatial Validation

Algorithm 3 is implemented as a sequence of checks: pattern validation, content filter (excluding equipment labels), schematic-area filter, and then circle proximity (Table 4.1: 12 units detection, 3 units max radius). Circles are detected per page and cached; a grid spatial index (50-unit cells) reduces matching complexity from O(n×m) to O(n log m). Calibration on the SE Pracana dataset showed that 12 units accommodate 98% of valid terminals and that a 3-unit radius separates terminal symbols from equipment circles (5+ units). Ruler proximity matching uses a 400-unit tolerance for rulers positioned between terminal groups.

In the implementation used for the validation reported in Chapter 6, circle extraction and circle-proximity validation were disabled by configuration to reduce processing time (on the order of tens of seconds per multi-page document in batch runs). Terminal validation in those runs therefore relied only on pattern, content, schematic-area, and ruler-proximity. The algorithm is present in the codebase and can be re-enabled when higher precision is prioritised over runtime.

5.2 CAD to PDF Conversion

Conversion uses the COM interface `win32com` for AutoCAD; plot-to-PDF preserves vectors and OCG layers. The default plot mode is Layout 5, suited to E-REDES page layouts; the code enforces PlotType 5 before plotting to avoid blank output. Blank PDFs are detected by file size,

under 10 KB versus typically 100 KB or more; invalid files are removed and an exception is raised. Per-file failures are logged without aborting the batch.

5.3 Route Detection Implementation

Route detection implements Algorithms 4 and 5. Extraction uses LINE and LWPOLYLINE entities from the ESQUEMA layer, with layer selection by name and content-based fallback when OCG is missing, and a length filter of 5–500 units from Table 4.2. For parallel clustering, two routes are considered parallel if their midpoint distance is at most 5 units and their angle difference is at most 10°; transitive closure forms bundles. Route-terminal matching uses the snap tolerance from Table 4.2 for internal terminals and 500 units for external X? cables; proximity is queried via the spatial index, and when multiple terminals fall within range, ruler-type preference e.g. X5.x is applied. Results and limitations are presented in the next chapter.

5.4 Cable Naming Implementation

Cable naming implements Algorithm 6. Equipment zones use a 500-unit radius from Table 4.2. Naming follows E-REDES grammars, D00-C13-500 [1]: HV and MV cables use W+ voltage + phase + number; LV cables use W9 + number + panel, see Tables 3.3 and 3.4. The implementation faces three main challenges: geometric ambiguity when several terminals fall within snap tolerance, where zone and bundle strategies only partly help; context-dependent rules such as panel identifier and equipment association that require semantics beyond geometry; and error propagation from the terminal and route stages. Failure modes include ambiguous route-terminal association, external X? cables without equipment context, and inter-panel rules that are not derivable from coordinates. Quantitative results and implications are discussed in the results and conclusions chapters.

5.5 Datasets and Test Cases

Validation uses two datasets. The first is SE Pracana, pilot: 6 pages, 52 terminals, and rulers X0.1, X0.2, and X5.3; it is used for calibration and detailed analysis. The second is the full substation project, Project Template: 279 pages across 19 plan types including SACA, SACC, protection, control, and others; it is used for full pipeline and CAD-to-PDF validation and is what achieves implementation-ready results for terminal extraction. Reference data was produced by manual extraction following E-REDES formats and ruler assignment from Table 3.6, and is stored in a structured Excel sheet (Figure 5.3). The sheet includes a *unified coordinate* column: each value gives the page index in the drawing set and the grid cell on that page (e.g. 3 B1, 7 D4), where the cell is the row–column position on the schematic grid. This supports traceability between the terminal list and the drawings. Figure 5.4 shows an example schematic excerpt with the grid that is the source of these coordinates. This setup supports completeness validation and Objective 3.

SUBESTAÇÃO: SE PRACANA 60/30kV
PAINEL: UGC
ISO: 5457-A3_ESQ
DATA: 21/01/2026

LISTA DE TERMINAIS

X0.1			X0.2			X5.3		
TERMINAL	PAGE	UNIFIED COORDINATE	TERMINAL	PAGE	UNIFIED COORDINATE	TERMINAL	PAGE	UNIFIED COORDINATE
1	3	B1	1	2	B2	501	7	C1
2	3	B1	2	2	B2	502	7	D1
3	3	B2	3	2	B6	503	7	D2
4	3	B2	4	2	B6	504	7	D3
5	3	D4	5	2	B2	505	7	D3
6	3	D4	6	2	B2	506	7	D4
7	6	D4	7	2	B6	507	7	D4
8	6	D4	8	2	B6	508	7	D5
9	6	B1				509	7	D5
10	6	B2				510	7	D5
						511	7	D6
						512	7	D6
						513	7	D6
						514	7	D7
						515	7	D7
						516	7	D8
						517	8	D1
						518	8	D2

Figure 5.3: Reference data Excel format: terminals by ruler, with page and unified coordinate (page and grid cell) for traceability.

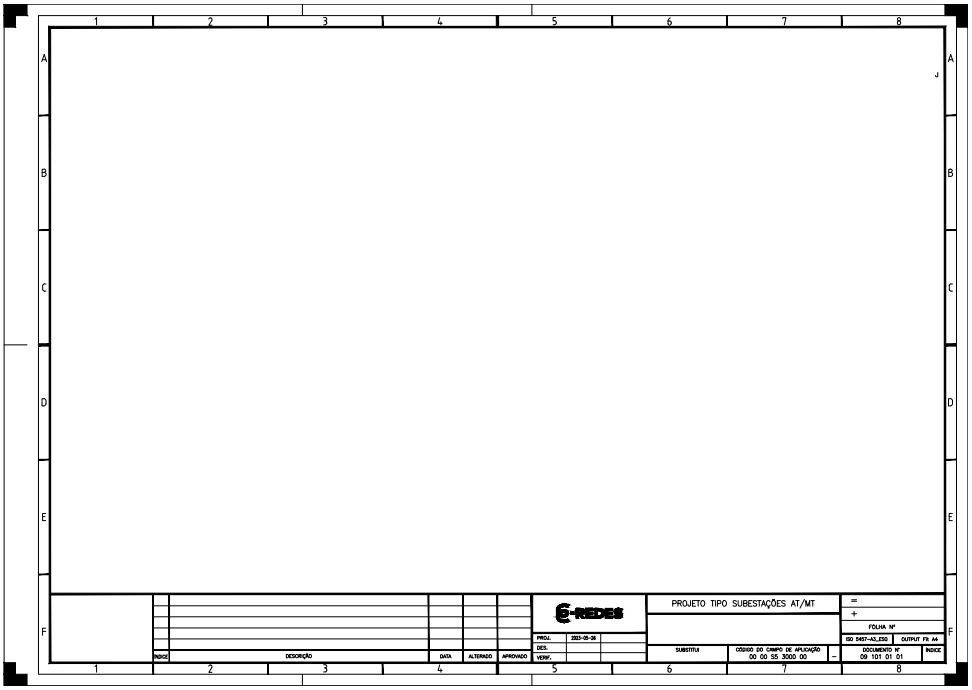


Figure 5.4: Example E-REDES schematic excerpt showing the grid; the unified coordinate in the reference data (page and cell) refers to this grid for traceability.

Table 5.2: Validation dataset characteristics

Dataset	Pages	Terminals	Rulers	Avg/Page	Terminal Distribution
SE Pracana (Pilot)	20	52	3	8.7	X0.1: 10 (19.2%), X0.2: 8 (15.4%), X5.3: 34 (65.4%)
Project Template (Full)	279	Ch. 6	Ch. 6	Ch. 6	Ch. 6

Full-dataset aggregates in the results chapter.

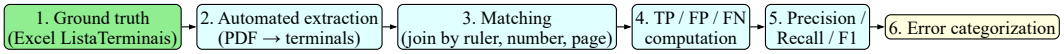


Figure 5.5: Validation workflow: reference data, extraction, matching, metrics, error analysis.

5.6 Validation Methodology

Automated output is compared to reference data on the ruler, terminal number, and page. The matching is exact in all three criteria; because the E-REDES formats are deterministic, exact matching is appropriate. TP, FP, FN and precision, recall and F1 from Table 5.3 are used because terminal extraction is an imbalanced classification problem: FP and FN have different costs, review versus addition, and precision and recall provide actionable diagnostics while F1 balances both. The validation procedure is: prepare the reference data in the format; run the pipeline; load and group the results by ruler and page; compute exact matches for TP; calculate P, R and F1; perform per-ruler analysis; categorize errors such as fragmentation, circle miss, ruler inference, annotation FP, OCG miss. Completeness validation compares automated lists to manual reference data; results are reported in the next chapter.

Table 5.3: Validation metric definitions

Metric	Definition
TP	Ruler, number, and page match
FP	Detected but not in reference data
FN	In reference data but not detected
Precision	$TP/(TP+FP)$
Recall	$TP/(TP+FN)$
F1	$2PR/(P+R)$

Chapter 6

Results and Discussion

This chapter reports validation results, metrics, and accuracy versus manual baselines. The results are given by component: terminal extraction is production-ready; CAD-to-PDF conversion is automated; route detection is partial; cable naming is insufficient. The discussion examines causes of success and limitations.

6.1 Terminal Extraction Results

The validation used a SE Pracana panel and the full E-REDES substation project template.

6.1.1 Pilot dataset (SE Pracana)

The pilot dataset is a single panel from the Pracana Substation with 20 pages, 52 terminals, 3 rulers, and was used for tolerance calibration and for per-ruler error analysis. Validation against manual reference data achieved 96.81% accuracy, 97.42% precision, 96.81% recall, and 97.11% F1 score. Table 6.1 presents detailed results by terminal type, ruler, showing expected counts, detected counts, correct matches, false positives, false negatives, precision, recall, and F1 score for each ruler category.

The pilot dataset achieved high precision and recall across all three ruler types; the calibrated tolerances handle the terminal patterns in this dataset well. Multi-stage validation filters out non-terminal text while keeping valid terminal identifiers. Overall metrics—98.04% precision, 96.15%

Table 6.1: Terminal extraction results by ruler type for SE Pracana pilot dataset.

Ruler	Expected	Detected	Correct	FP	FN	Precision (%)	Recall (%)	F1 (%)
X0.1	10	8	8	0	2	100	80	88,89
X0.2	8	8	8	0	0	100	100	100
X5.3	34	35	34	1	0	97,14	100	98,55
Total	**52**	**51**	**50**	**1**	**2**	**98,04**	**96,15**	**97,09**

FP = False Positives (detected but not in reference data); FN = False Negatives (in reference data but not detected).

Table 6.2: Terminal extraction validation results for full E-REDES project template panels.

Panel	Pages	Expected	Detected	Correct	Precision (%)	Recall (%)	F1 (%)
SACC	27	617	648	607	93,67	98,38	95,97
TP and Voltage Regulation	23	416	424	407	95,99	97,84	96,90
Inter-Busbars and HV Busbar Voltage	19	285	310	280	90,32	98,25	94,12
HV Line	19	251	265	245	92,45	97,61	94,96
TP	21	355	367	348	94,82	98,03	96,40
HV Busbar Voltage	9	106	117	104	88,89	98,11	93,27
TP Arrival	12	200	214	196	91,59	98,00	94,69
TSA, Neutral Reactance and MV Busbar Voltage	15	284	306	278	90,85	97,89	94,24
Capacitor Bank	17	253	252	248	98,41	98,02	98,22
MV Line	12	162	174	158	90,80	97,53	94,05
Metering	10	63	64	62	96,88	98,41	97,64
MV Line Metering	21	290	288	284	98,61	97,93	98,27
Communications	11	44	56	43	76,79	97,73	86,00
PCL UC	8	78	90	76	84,44	97,44	90,48
MV Inter-Busbars	10	121	140	118	84,29	97,52	90,42
Internal Arc	9	102	102	100	98,04	98,04	98,04
Power Quality	9	108	112	106	94,64	98,15	96,36
QIT	7	44	44	43	97,73	97,73	97,73
Average	–	**3779**	**3973**	**3703**	**93,20**	**97,99**	**95,54**

recall, 97.09% F1—are production-ready for terminal extraction, with minor errors from coordinate mismatches or text fragmentation during PDF export.

6.1.2 Full project template

Terminal extraction was validated on the full E-REDES project template (279 pages across 19 plan types; see Table 3.1). For eighteen panels from the template, reference data was prepared in the standard tabular Excel format used for terminal lists. Validation results for those eighteen panels are presented in Table 6.2.

The validated panels show production-ready recall (97.99% on average) with 270 false positives and 76 false negatives across 3779 expected terminals. Precision averages 93.20% and F1 95.54%. False positives occur where spatial matching assigns terminals to incorrect rulers or coordinates (e.g. X0.2, X0.3, X0.5) or when contact block numbers and revision labels match the terminal pattern without geometric validation. False negatives reflect coordinate mismatches or missing extraction. Table 6.2 summarises the per-panel metrics.

6.1.3 Error Analysis: False Positives and False Negatives

Figure 6.1 shows a schematic region where contact block numbers (e.g. 11, 12) match the terminal pattern; without circle-proximity validation such text can be incorrectly assigned to the nearest ruler, resulting in over-detection. Circle-proximity validation was not applied in the runs

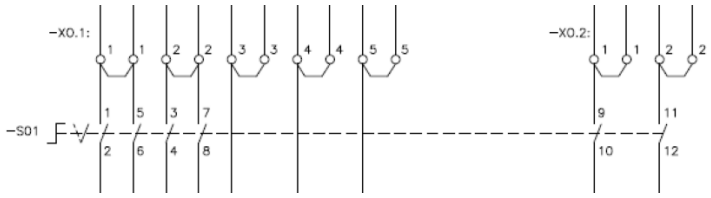


Figure 6.1: Schematic region where contact block numbers match the terminal pattern; without geometric validation such text can be incorrectly reported as terminals.

reported here (see Section 5.1); re-enabling it is expected to reduce false positives from pattern-matched text that is not near a terminal circle. Other causes include annotation text that matches terminal patterns, page or grid labels inside the schematic-area filter, ruler-assignment ambiguity when multiple rulers lie within proximity range, and the strict exact match (ruler, number, page): a detection that refers to the same physical terminal but with a different ruler or page is still counted as a false positive.

Most false negatives are due to adjacent-cell coordinate mismatch: the terminal is detected but assigned to a grid cell next to the reference (e.g. reference D6, detected D7), so it falls outside the tolerance for exact match. This happens when PDF export shifts coordinates slightly or when the terminal text is slightly offset from the circle centre. A smaller share of false negatives comes from terminals that are not detected at all, for example when the text lies in a non-standard OCG layer, when SHX or annotation extraction fails, or when the label is fragmented and not merged correctly.

The primary error sources are over-detection in rulers such as X0.2, X0.3, and X0.5, where terminal patterns (1–2 digit numbers) overlap with contact block numbers, revision numbers, and dimension labels in equipment tables. Under-detection occurs where coordinate mismatches or missing text extraction prevent exact matching. Across the eighteen validated panels, false positives (270) outnumber false negatives (76), so the main gain from improved geometric context (e.g. circle-proximity validation) would be a reduction in over-detection.

6.2 CAD to PDF Conversion Results

Batch conversion from DWG to PDF was run on the full E-REDES project template. The conversion pipeline (Section 5.1) was executed without manual intervention: each panel's DWG set was exported to a single multi-page PDF with vector geometry and content groups (Optional Content Groups (OCG)) preserved for layer-based extraction. The print area was set to coincide with the drawing limits so that the exported PDF matched the intended layout. Success was defined as completion without error and with the expected page count per panel. All panels converted successfully; vector entities (lines, polylines, circles, text) and layer structure were retained in the output PDFs, enabling downstream terminal extraction and route detection from the same files. The batch timing was recorded per panel; the total runtime for the full template was on the order

of minutes. No manual re-runs or corrections were required. The conversion step is therefore considered automated and production-ready for the project template scope.

6.3 Route Detection Results

Route extraction and parallel clustering (Algorithms 4, 5) were applied to the ESQUEMA layer on all validated drawings. LINE and LWPOLYLINE entities from the ESQUEMA layer were filtered by length (5–500 units) and grouped into parallel bundles where midpoint distance and angle difference lay within the tolerances of Table 4.2. Route counts per drawing and per panel were recorded during batch runs. Route-terminal matching was attempted using snap tolerance for internal terminals and a 500-unit range for external connections.

The conclusion that route detection was the failing component was reached as follows. Terminal extraction had already been validated against manual reference data (Section 6.1), so we had a reliable set of terminal positions and identities. For a subset of drawings we also had known cable connections (which terminals should be linked by which routes), either from reference documentation or from manual inspection. We ran route-terminal matching on the extracted routes and terminals: even after relaxing snap tolerance and distance thresholds, known cable connections could not be reliably recovered. Because the terminal side was validated, the mismatch had to come from the route side: either the extracted route geometry did not reach the terminal locations (e.g. segments ended short) or the routes were fragmented into disjoint pieces so that a single logical cable appeared as several unconnected segments. Inspection of the PDF-derived geometry confirmed that export breaks continuous CAD paths into disconnected segments, so the failure was attributed to route representation after PDF conversion (a structural limitation of the pipeline) rather than to tolerance tuning or to terminal extraction. A quantitative summary of route counts and match rates is omitted here; the implications for cable naming are taken up in the next section and in the conclusions.

6.4 Cable Naming Results

Cable naming was evaluated in relation to route-terminal matching accuracy and E-REDES grammar application. The system implements the full E-REDES naming grammars for HV, MV and LV cables and the panel classification logic described in Chapter 4; the naming is applied once route-terminal pairs are available. Because route-terminal matching did not achieve reliable associations on the validated drawings (see Section 6.3), cable naming could not be assessed quantitatively on the same basis as terminal extraction. Failure modes include ambiguous route-terminal association when several terminals fall within snap tolerance, missing context for external (X?) cables, and inter-panel rules that are not derivable from coordinates alone. Quantitative naming accuracy and the boundary between what can be automated with geometric extraction and what requires semantic or CAD-native input are discussed in the conclusions and in Chapter 7.

6.5 Discussion

Terminal extraction success relates to multi-stage validation and tolerance calibration. The SE Pracana pilot achieved 96.81% accuracy through iterative tolerance tuning. The full project template validation (eighteen panels) achieved 97.99% recall and 93.20% precision on average, indicating consistent performance across all panel types.

The primary limitation is over-detection in specific rulers (X0.x), where spatial matching assigns terminals to incorrect rulers or coordinates due to geometric ambiguity in PDF export. False positives occur when contact block numbers, revision labels, or dimension text match terminal patterns but lack the geometric context (circle proximity) that distinguishes actual terminals. False negatives result from coordinate mismatches introduced by PDF export variations or from missing text extraction when terminals appear in non-standard OCG layers.

CAD-to-PDF conversion runs in batch without manual intervention and preserves vector information and OCG layers for downstream extraction. Route-terminal matching and cable naming are limited by geometric ambiguity and context-dependent rules, as discussed in the conclusions chapter. Implications for partial versus full automation and for future work are taken up there.

Chapter 7

Conclusions

This work provided a technical assessment of automating cable documentation for Portuguese electrical substations, focusing on the compliance of E-REDES standards. The objectives were to evaluate automated terminal extraction for production-ready accuracy, to implement and evaluate cable naming within geometric limits, to validate on Bragalux project data, and to provide a PDF-based workflow.

Accomplishment of the Objectives

The objectives defined at the beginning of this work were systematically addressed. A summary of how each goal was approached is presented below.

The work began by examining the E-REDES standards and identifying terminal extraction as the highest-priority automation requirement. Manual terminal cataloging consumes 2 to 4 hours per project and introduces errors that compromise system safety. The multi-stage validation architecture was designed to address this challenge, combining pattern recognition, content filtering, spatial validation, ruler assignment, and circle detection to achieve production-ready accuracy.

Terminal extraction was implemented and validated. The multi-stage validation architecture was applied to real E-REDES project drawings and reached production-ready accuracy. Combining multiple validation stages that independently optimize the accuracy criteria improved on existing PDF-based extraction methods.

Cable naming automation was achieved only in part. The system implements E-REDES naming grammars for HV, MV, and LV cables, with full panel classification logic and equipment association algorithms. Route-terminal matching, however, hit fundamental limits: route fragmentation during PDF rendering decomposes continuous CAD paths into disconnected segments without segment relationship metadata.

The compliance with the E-REDES standards was fully achieved for the terminal extraction. Terminal extraction demonstrates complete compliance with E-REDES terminal identification formats across all terminal types in the validation dataset. Cable naming implements complete E-REDES naming grammars, but cannot achieve production-level accuracy due to route-terminal matching limitations.

Workflow integration was fully achieved. The PDF-based workflow runs in batch without freezing CAD software, removing the bottleneck where CAD-native processing blocks technicians. The CAD-to-PDF conversion pipeline automates manual conversion, cutting processing time from 1–2 hours to under 5 minutes per project. Excel output is E-REDES-compliant and ready for regulatory submission.

Validation and performance assessment were fully achieved. The validation methodology gives quantitative accuracy metrics, error analysis, and completeness checks. Terminal extraction lists all terminals for technician check and supports manual cable list production rather than replacing it. Reproducibility is supported by documented algorithms, full parameter tables, and step-by-step reproduction instructions.

Terminal Extraction Performance

The multi-stage validation architecture achieved production-ready accuracy for terminal extraction. The approach proved essential for reducing false positives from annotation text, equipment labels, and page numbers that initially matched terminal patterns. Content validation and spatial constraints filtered out these false positives. Circle detection provided geometric confirmation for most validated terminals, further reducing error rates.

Terminal extraction supports manual cable list production by ensuring that all terminals are identified and checked. Manual terminal cataloging requires 2–4 hours per project, depending on complexity, as technicians must verify terminal completeness and accuracy. Automated terminal extraction generates a complete terminal list that technicians review and validate, ensuring that no terminals are missed while maintaining documentation quality. The system assists technicians in the manual cable list production process rather than replacing it.

Cable Naming Limitations

Cable naming automation encountered fundamental limitations that prevent complete automation. Despite successful terminal detection and route extraction, route-terminal matching failed to establish reliable associations for reference cable connections. Empirical validation showed that known cable connections could not be matched even with extended tolerance thresholds, so the limitation is fundamental rather than a calibration issue.

The limitation stems from route fragmentation during PDF rendering. Continuous CAD paths connecting terminals are decomposed into multiple disconnected segments and *polylines* with intermediate vertices during PDF export. This fragmentation makes it ambiguous which segments form complete terminal-to-terminal paths. Geometric-only approaches cannot resolve this ambiguity without semantic metadata that identify segment relationships.

This finding quantifies the boundary conditions of PDF-based extraction for cable naming automation. CAD systems store connections as continuous paths with explicit terminal references, where the routes connect directly to terminal positions. PDF rendering decomposes these continuous paths into fragmented segments without preserving segment relationship metadata. This

represents a fundamental difference between CAD logical models and PDF geometric rendering.

The limitation does not invalidate the geometric approach but quantifies its boundary conditions. Terminal extraction shows that geometric extraction can reach high accuracy when data is available. However, cable naming requires route-terminal matching that depends on structural continuity, which is lost during PDF rendering.

Limitations and Validation Constraints

Validation uses both the SE Pracana pilot and the full E-REDES substation project template; validation against the full project is what achieves implementation-ready results for terminal extraction.

Technical constraints include dependency on the COM interface for CAD-to-PDF conversion, which requires the Windows operating system. Drawing format requirements, such as consistent layer organization and ESQUEMA layer presence, follow E-REDES standardized project types, ensuring consistent structure across projects.

Practical Implications and Deployment

Terminal extraction provides immediate workflow improvement for Bragalux. The system reaches production-ready accuracy and lists all terminals for technician review. Deployment should prioritize terminal extraction as the primary automation benefit, supporting technicians in manual cable list production by providing complete terminal lists that must be verified and integrated into cable documentation. Cable naming remains a future research direction rather than an immediate production requirement.

Future Research Directions

Future research should address the limitations of route-terminal matching through semantic understanding approaches. Machine learning methods could learn segment relationship patterns from CAD-native training data, enabling reconstruction of complete terminal-to-terminal paths from fragmented PDF segments. Knowledge-based systems could encode E-REDES drawing conventions to guide segment association.

Alternative matching strategies include template matching, pattern recognition, and context-aware matching using equipment associations and panel relationships. Hybrid geometric-semantic approaches that combine distance-based matching with semantic constraints could improve matching accuracy while maintaining PDF-based workflow independence.

Cable naming automation advances require context-aware rule application and knowledge-based systems. Context-aware rules would evaluate equipment associations, panel hierarchies, and drawing context to resolve ambiguous naming decisions. Integration of external knowledge bases could provide semantic metadata that is not available in geometric-only extraction.

System enhancements include real-time processing capabilities, interactive validation interfaces, and error correction mechanisms. Performance optimization would reduce processing time through parallel processing and algorithmic improvements. Scalability enhancements would support larger projects through distributed processing and database integration.

Research extensions include application to other utility standards, extension to other CAD drawing types, and integration with other automation systems. Multi-utility validation would assess performance across different naming conventions and utility standards. Industry deployment studies would quantify real-world benefits beyond laboratory validation.

Contributions

This work contributes to automation research in electrical substation documentation as follows. It presents a multi-stage terminal extraction architecture combining pattern recognition, spatial validation, circle detection, and ruler assignment that achieves production-ready accuracy on E-REDES drawings. It demonstrates PDF-based workflow integration that allows batch processing without freezing CAD software, addressing a critical workflow bottleneck. It establishes a validation methodology using real project data. It quantifies the boundary conditions of geometric-only extraction: terminal extraction succeeds; cable naming is limited by route fragmentation in PDF rendering, providing empirical evidence for where semantic or CAD-native approaches are required.

Final Remarks

Automated terminal extraction from E-REDES substation drawings reaches production-ready accuracy through multi-stage validation. The system exceeds production thresholds and lists all terminals for technician review. Complete cable naming automation remains out of reach because of route fragmentation during PDF rendering.

Key findings include:

- Terminal extraction achieves production-ready accuracy through multi-stage validation, ensuring all terminals are identified and checked by technicians;
- Route-terminal matching encounters fundamental limitations due to PDF rendering fragmentation, preventing complete cable naming automation;
- Geometric-only approaches can achieve high accuracy for terminal extraction but cannot resolve structural ambiguity for route-terminal matching;
- PDF-based workflow runs in batch without freezing CAD, removing critical workflow bottlenecks;
- The multi-stage validation architecture provides a foundation for reproducible deployment across E-REDES projects.

Although complete cable naming automation was not achieved, the terminal extraction system provides an immediate workflow improvement for Bragalux by ensuring that all terminals are identified for technician review. The system supports manual cable list production rather than replacing it, as technicians must still verify terminal completeness and produce cable documentation manually. The architecture, validation methodology, and documented limitations provide a valuable foundation for future research. With advances in semantic understanding and machine learning, route-terminal matching may overcome current geometric limitations, enabling complete cable naming automation in future systems.

The work contributes to automation research by quantifying the boundary conditions of PDF-based extraction, demonstrating what can be achieved through geometric approaches, and identifying where semantic understanding is required. This finding informs future architecture design decisions and provides empirical evidence for the limitations of geometric-only approaches in complex electrical drawing automation.

References

[1] E-REDES, "Referenciação - generalidades," E-REDES – Distribuição de Eletricidade, S.A., Portugal, Technical Document D00-C13-500/N, Mar. 2024, Rev. 1.

[2] A. R. Putra, S. Ha, and K.-P. Park, "Automatic extraction of cable connection information from 2D drawings for electrical outfittings design in shipyards," *International Journal of Naval Architecture and Ocean Engineering*, vol. 16, p. 100 630, 2024, ISSN: 20926782. DOI: 10.1016/j.ijnaoe.2024.100630. Accessed: Nov. 19, 2025.

[3] L. Yang et al., "Practical single-line diagram recognition based on digital image processing and deep vision models," *Expert Systems with Applications*, vol. 230, p. 120 891, 2023. DOI: 10.1016/j.eswa.2023.120891.

[4] P. F. M. Dias, "Projecto estruturado de Sistemas de Automação em Subestações segundo a norma 61850,"

[5] D. F. R. Martins, "Implementação do Projecto-Tipo da EDP para Automação de Subestações segundo a norma IEC 61850: Configuração de IED's de dois fabricantes para painéis de saída em MT,"

[6] J. P. G. Couto, "ESTUDO DO PROJETO ELÉTRICO DE UMA SUBESTAÇÃO ELEVATÓRIA ALTA TENSÃO,"

[7] J. P. L. Gonçalves, "Estudo do Projeto Elétrico de uma Subestação Elevadora,"

[8] T. Humel and J. Weaver, "Automating substation design with autodesk® inventor®, autodesk® autocad® electrical, and substation design suite," in *Autodesk University*, 2017, Session UT1708.

[9] Z. Peng, G. Yan, Q. Zhongshan, L. Huiyong, L. Mouying, and L. Shengnan, "CIM/G graphics automatic generation in substation primary wiring diagram based on image recognition," *Journal of Physics: Conference Series*, vol. 1617, no. 1, p. 012 007, Aug. 2020, ISSN: 1742-6596. DOI: 10.1088/1742-6596/1617/1/012007. Accessed: Nov. 25, 2025.

[10] EPLAN Software & Service, *Eplan electric p8 - electrical engineering software*, 2024. Accessed: Dec. 4, 2025. [Online]. Available: https://www.eplan.com/en/products/eplan-electric-p8.

[11] Zuken Inc., *E3.series - electrical design software*, 2024. Accessed: Dec. 4, 2025. [Online]. Available: https://www.zuken.com/en/products/e3-series.

[12] Bentley Systems, *Openutilities substation - substation design software*, 2024. Accessed: Dec. 4, 2025. [Online]. Available: https://www.bentley.com/en/products/product-line/electrical-engineering-software/openutilities-substation.

[13] IGE+XAO Group, *See electrical - electrical cad software*, 2024. Accessed: Dec. 4, 2025. [Online]. Available: https://www.see-electrical.com.

[14] Smap3D GmbH, *Smap3d electrical - electrical engineering software*, 2024. Accessed: Nov. 30, 2025. [Online]. Available: https://www.smap3d.com/en/products/electrical-ecad.

[15] Scada Systems, *Elecdes design suite - electrical cad software*, 2024. Accessed: Dec. 4, 2025. [Online]. Available: https://elecdes.com.

[16] Siemens Digital Industries Software, *Capital suite - electrical systems engineering*, 2024. Accessed: Dec. 6, 2025. [Online]. Available: https://plm.sw.siemens.com/en-US/capital/ee-systems-electrical.

[17] I. Ekeke, C. F. Moreno-García, and E. Elyan, "Attention-based framework for automated symbol recognition and wiring design in electrical diagrams," *Applied Artificial Intelligence*, vol. 39, no. 1, e2548834, 2025. DOI: 10.1080/08839514.2025.2548834.

[18] D. Hemker, J. Maalouly, H. Mathis, R. Klos, and E. Ravanan, "From schematics to netlists – electrical circuit analysis using deep-learning methods," *Advances in Radio Science*, vol. 22, pp. 61–75, 2024. DOI: 10.5194/ars-22-61-2024.

[19] R. Yao, X. Cheng, Z. Hui, Z. Li, and Z. Min, "Intelligent reading for multi-scale engineering drawings based on adaptive object detection and optical character recognition," in *Proc. 9th Int. Conf. Industrial Engineering and Applications*, 2022.

[20] H. Bhanbhro, Y. K. Hooi, W. Kusakunniran, and Z. H. Amur, "Symbol Detection in a Multi-class Dataset Based on Single Line Diagrams using Deep Learning Models," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 8, 2023, ISSN: 21565570, 2158107X. DOI: 10.14569/IJACSA.2023.0140806. Accessed: Nov. 19, 2025.

[21] S. Elnagar, V. Yoon, and M. A. Thomas, "An automatic ontology generation framework with an organizational perspective," in *Proc. 53rd Hawaii Int. Conf. System Sciences*, 2020.

[22] A. Devanand et al., "Ontopowersys: A power systems ontology for cross domain interactions in an eco industrial park," Cambridge Centre for Computational Chemical Engineering, Preprint, 2019.

[23] O. Onabanjo, F. Ortega Fernandez, M. Díaz Piloñeta, G. Martínez Huerta, and C. Moreno-García, "Analysis of electrical projects from single-line diagrams using generative ai," in *Proc. 29th Int. Congress Project Management and Engineering*, 2025.

[24] J. Villena Toro, A. Wiberg, and M. Tarkian, "Optical character recognition on engineering drawings to achieve automation in production quality control," *Frontiers in Manufacturing Technology*, vol. 3, p. 1 154 132, Mar. 2023, ISSN: 2813-0359. DOI: 10.3389/fmtec.2023.1154132. Accessed: Nov. 19, 2025.

[25] E-REDES, "Subestação tipo exterior - projeto-tipo – memória descritiva," E-REDES – Distribuição de Eletricidade, S.A., Portugal, Technical Document DIT-C13-500/N, Apr. 2025, Rev. 1.

[26] E-REDES, "Tecnologias de eletrificação - regras de execução," E-REDES – Distribuição de Eletricidade, S.A., Portugal, Technical Document DRE-C13-510/N, Mar. 2024, Rev. 1.

[27] E-REDES, "Circuitos BT - Regras de execução," E-REDES, Portugal, Technical Document DRE-C13-512/N, Nov. 2013.

[28] E-REDES, "Sistemas de iluminação, tomadas interiores, carga de ves, extração de fumos, ar condicionado, centrais de intrusão e incêndio - regras de execução," E-REDES – Distribuição de Eletricidade, S.A., Portugal, Technical Document DRE-C13-511/N, Mar. 2024.

[29] Artifex Software, Inc., *PyMuPDF: Python bindings for MuPDF*, Python library for PDF processing with vector coordinate precision and Optional Content Group (OCG) layer support, 2025. Accessed: Jan. 8, 2026. [Online]. Available: https://github.com/pymupdf/PyMuPDF.