

Article

Not peer-reviewed version

Form-Based Test Modeling, Analysis, and Complexity Evaluation for Smart OCR Apps

[Jerry Gao](#), [Radhika Agarwal](#)^{*}, [Rohit Vardam](#), [Jasleen Narang](#)

Posted Date: 16 January 2026

doi: 10.20944/preprints202601.1280.v1

Keywords: computer vision; smart OCR apps; test modeling; text complexity; quality assurance; automation



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Form-Based Test Modeling, Analysis, and Complexity Evaluation for Smart OCR Apps

Jerry Gao , Radhika Agarwal * , Rohit Vardam and Jasleen Narang

Department of Computer Engineering, College of Engineering, San Jose State University, San Jose, CA, USA

* Correspondence: radhika6696@gmail.com

Abstract

OCR systems are increasingly being used in high-stakes and form-based applications, such as employment onboarding and voter registration, where accuracy is crucial to ensure system reliability. The paper introduces a test modeling and evaluation framework for Smart OCR applications that combines decision tables, context trees, and component-level complexity analysis to permit complete and automated validation. The proposed approach models document structure, environment (lighting, angle, and quality of images), and completeness of the input used to produce representative test cases of different operational situations. Accuracy and coverage measures are also used together to measure fidelity of recognition and structural completeness of multiple form components at the multimodal level, such as text fields, tables, checkboxes, radio buttons, and signatures. The framework is confirmed with an empirical case study of the application of employment and voter registration forms with three commercial OCR tools, such as Microsoft Lens, Amazon Textract, and Parsio.io. The experimental findings show the apparent trade-offs between accuracy and coverage and indicate considerable differences in symbolic and contextual extraction abilities by tools. Parsio.io scores the best on the criteria of balanced performance, as it has a high coverage and multimodal recognition strength among the tested systems.

Keywords: computer vision; smart OCR apps; test modeling; text complexity; quality assurance; automation

1. Introduction

Optical Character Recognition (OCR) is a fundamental component of contemporary document processing systems, facilitating the automated transformation of scanned documents, images, and photographs into structured, machine-readable data. OCR technologies are widely implemented in various domains, including employment onboarding, voter registration, healthcare records management, financial services, and government operations. Within these contexts, OCR systems are often integrated into downstream decision-making processes, where inaccuracies or incomplete data extraction can lead to operational errors, compliance issues, or misinterpretation of information. Consequently, maintaining the reliability and robustness of OCR systems in real-world environments represents a significant quality assurance challenge.

Recent advances in computer vision and deep learning have significantly improved OCR recognition accuracy, particularly for clean and well-aligned text. However, real-world documents often contain complex layouts such as hierarchical sections, nested tables, checkbox and radio-button panels, conditional logic, and handwritten or semi-structured fields. Traditional OCR evaluation methods, which focus on character- or word-level accuracy, do not capture these structural and semantic challenges. As a result, a system may achieve high textual accuracy but still fail to associate labels with values, detect checkbox selections, or recognize signatures, leading to incomplete or misleading results.

OCR performance is greatly impacted by contextual and environmental elements, such as illumination, picture quality, capture angle, and camera distance, in addition to structural complexity.

Though they are frequently examined in an ad hoc or unsystematic manner, variations in these settings can considerably influence recognition outcomes. The majority of OCR benchmarks and datasets currently in use place a strong emphasis on static picture collections and lack a systematic method for determining test coverage across contextual dimensions. As a result, many OCR systems are implemented without an adequate understanding of their failure modes or robustness boundaries.

This research suggests a methodical test modeling and evaluation strategy for Smart OCR systems in order to overcome these constraints. To facilitate the creation of organized and repeatable test cases, the framework incorporates decision tables, context trees, and hierarchical input-output modeling. To ensure controlled variation among test scenarios, combinations of document attributes, field completion levels, and environmental factors are enumerated using decision tables. Input and output trees depict the hierarchical structure of document fields both before and after OCR processing, whereas context trees model external elements like illumination, shot angle, and image quality. When combined, these models enable fine-grained validation of OCR behavior in both contextual and structural dimensions.

Beyond test modeling, this paper proposes a dual-metric evaluation technique based on accuracy and coverage. Accuracy assesses the correctness of recognized elements in relation to ground truth, whereas coverage measures the extent to which all structural components of a document—such as tables, checkboxes, value-pair lists, and signature regions—are tested and validated. This distinction is critical since great accuracy over a narrow set of features does not imply robustness in real-world applications. The suggested framework assesses OCR system quality more comprehensively by combining accuracy and coverage analyses.

The suggested methodology is validated using empirical case studies of two representative structured documents: employment application forms and voter registration forms. These texts were chosen for their moderate to high structural complexity, multimodal input aspects, and real-world applicability. Three popular OCR applications, Microsoft Lens, Amazon Textract, and Parsio.io, are evaluated using automated test execution and validation pipelines in a variety of context-driven testing scenarios. The results show significant trade-offs between recognition accuracy, structural coverage, and resilience, emphasizing the significance of rigorous test modeling in OCR evaluations.

The remaining sections of this work are organized as follows. Section 2 discusses related work on optical character recognition systems, with a focus on current assessment, validation, and testing methods. Section 3 presents case studies of structured documents, such as employment applications and voter registration forms, and explains the AI-based test modeling using input trees, output trees, context trees, and decision tables. Section 4 analyzes the test results for the AI-based model and provides the statistical comparison. Section 5 explains the evaluation approach, introduces accuracy and coverage metrics, and provides an empirical test case analysis. Section 6 presents the automated execution results and comparative findings across multiple OCR tools. Finally, Section 7 concludes the paper by outlining potential avenues for further research in intelligent OCR testing.

2. Related Works

In recent years, artificial intelligence (AI) has undergone significant growth and evolution. Software systems are progressively incorporating AI features, including identification, recognition, prediction, and recommendation. To ensure that AI features function accurately, consistently, and reliably, testing AI software [29], which encompasses a range of quality assurance tasks for AI-based software systems, is crucial. However, AI software testing is different from traditional software testing because AI features include special qualities, including large-scale unstructured input data, unexpected scenarios, ambiguity in system outputs, reactions, or actions, and data-driven training features [30]. The interpretability of ML and DL models, the absence of precise definitions and requirements, the creation of test inputs, the definition of test oracles, and the management of dynamic settings are some of the difficulties this presents for AI software testing [31]. A wide range of AI testing subjects, including

test modeling [30,32–34], test case generation [35–37], test framework, and automation [34,35,38,39], have been thoroughly studied to overcome these problems.

Unit testing, integration testing, function testing, and system testing are the typical components of software testing. The method of testing AI functions mainly consists of test modeling, test case development, test execution, and test quality evaluation, as noted by Gao et al. (2021) [32]. For testing AI software, a number of standard validation techniques for conventional software have been used, such as metamorphic testing. A metamorphic testing-based testing methodology is proposed by C. Tao et al. (2019) [30], particularly for facial age recognition systems. Expert systems, computer vision, speech recognition, natural language processing, and artificial intelligence for business are some of the categories into which AI software systems may be divided according to their functionality.

Optical Character Recognition (OCR) is a significant computer vision technique used to identify text in scanned documents or images. [40]. There are six phases typically involved in text recognition in contemporary OCR systems: picture capture, preprocessing, segmentation, feature extraction, classification, and post-processing [41]. The Conventional Neural Network (CNN), one of the most widely used OCR approaches, is one of the methods used at every stage to increase the system's overall accuracy [42,43]. In a PhD dissertation, [44] presented several performance metrics for the page-reading system, such as character accuracy, word accuracy, non-stopword accuracy, and phrase accuracy.

OCR has been applied in various domains, and studies have discussed its applications, limitations, and improvements achieved to achieve better results. Prior research demonstrates certain ways OCR can be implemented, such as translating specific documents or extracting particular information from documents. For example, image processing and OCR applications have been explored for the extraction of input information from prescription documents [45], along with improving OCR accuracy, focusing on Thai documents using the Tesseract Python package [46]. A demonstration of how the Batbayin script was translated into the original Filipino format using OCR was also studied [47], illustrating that OCR has been used to translate and recover information for various languages.

Other research emphasizes advancing OCR systems in different ways, such as dataset creation, resource integration, and domain-based applications. An OCR dataset was built using forms from high school students to evaluate system accuracy through human verification [48]. Several additional studies focus on improving OCR accuracy and efficiency through the use of deep learning and advanced modeling techniques. For example, research has been done on deep neural network-based models for mixed language OCR [49]. Other studies focus on developing new analytic schemes for offline cursive handwriting, using segmentation and other tools to improve OCR reading [50]. There have been improvements made on Quranic text recognition [51] and OCR with neural networks and microscopic imaging [52]. In another study, a modular OCR pipeline was built to integrate third-party OCR engines [53].

These research papers focus on the shift from rule-based systems to more advanced test modeling. Each of these papers reflects on how OCR can be used and improved to develop and translate the different scriptures, forms, and more. Although these advancements have been made, there are many challenges that are faced in terms of OCR. The main focus of this work is to analyze how well each of these OCR tools performs through automated testing, validation, and different augmentation techniques.

Review Table 1 shows that prior works focused on accuracy benchmarking and dataset development, with few addressing systematic test modeling and automated test generation for OCR systems. This paper builds upon these studies by integrating decision tables and context trees for structured test case generation, enabling comprehensive evaluation of OCR performance under variable conditions.

Table 1. Summary of Related Works on OCR Testing and Validation.

Ref.	Objective	Automated Test Validation	Test Modeling	Test Generation	Augmentation
[45]	Image processing and OCR to accurately extract prescription information	Yes (usage of OCR tools done, automatic)	Rule-based (Tesseract OCR)	Image provided manually	None
[46]	Improve the accuracy of OCR for Thai language	Yes (test validation through Tesseract Python package, automatic)	Comparison testing	Collected online from GitHub repo	None
[47]	OCR package to translate Baybayin to original Filipino format	Yes (test validation through MATLAB and OCR, automatic)	Comparison using white/black box testing	Collected online from public repository	None
[48]	Build OCR dataset to evaluate OCR systems with human verification	No (creation of dataset only)	Black box testing	Collected forms using high school student data	None
[49]	Develop and evaluate mixed-language OCR models using deep neural networks	Yes (test validation done through Python scripts and libraries)	Data-driven modeling	Collected text from Finnish newspaper and journal corpus	None
[50]	Develop new analytic scheme for cursive handwriting by segmentation and feature extraction to improve OCR reading	Yes (measured by ground truth comparison)	Black box testing	Tested on handwritten cursive image datasets	Offline cursive data
[51]	Evaluate a new OCR method for Quranic text recognition	Yes (comparison with ground truth)	Black box testing	Testing on Quranic text dataset	None
[52]	Evaluate OCR system using neural networks and microscopic imaging	Yes (comparison with ground truth)	Black box testing	Tested using microscopic images	None
[53]	Develop modular OCR pipeline integrating third-party OCR engines	Yes (validated using ground truth)	Black box testing	Tested on medical forms	None
This Paper	Analyze different AI-based OCR tools for intelligent test modeling and validation	automated validation via Python scripts (Tesseract)	3-D AI test modeling (context, input, output)	Image-based test case generation	Noise-based augmentation

3. AI Test Modeling on Forms

The term test modeling describes the method of systematically expressing the components of a system being tested (including inputs, situations, and responses) in a way that facilitates the systematic generation of test cases and the evaluation of their adequacy [54]. Decision tables, state-transition diagrams, and classification trees have been used as traditional models to model tests in software

engineering. There is a strong demand for well-defined and practical test models for black-box validation of AI-powered functions in modern intelligent systems [55]. So, 3D AI test modeling has recently been studied for smart AI chatbots and computer systems. It forms the basis for the context tree, which represents the environmental and situational factors that affect OCR performance. As shown in Figure 1, the context includes:

- Lighting Conditions: Normal lighting vs. insufficient lighting
- Distance of the image captured: close vs far
- Shot Angle: Direct overhead vs. angled capture
- Image Quality: Clear vs. blurred images
- Physical condition of paper: Good vs any stained, faded, or wrinkled.

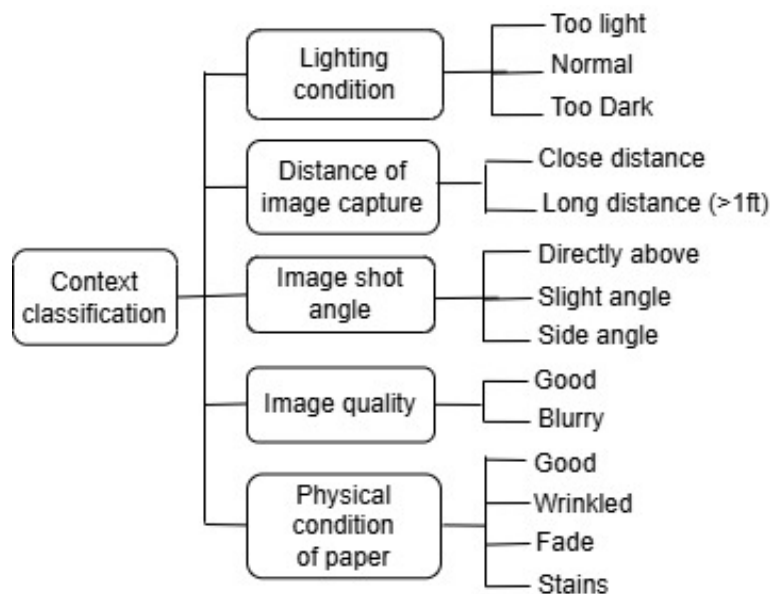


Figure 1. Context tree showing environmental factors affecting OCR performance.

This section aims to understand the test modeling for two different forms: the employment form and the voter registration form, which represent a standard format for most of the forms available, considering tables, value pair lists, checkbox panels, radio button panels, transaction information, and any form of signatures or uploads.

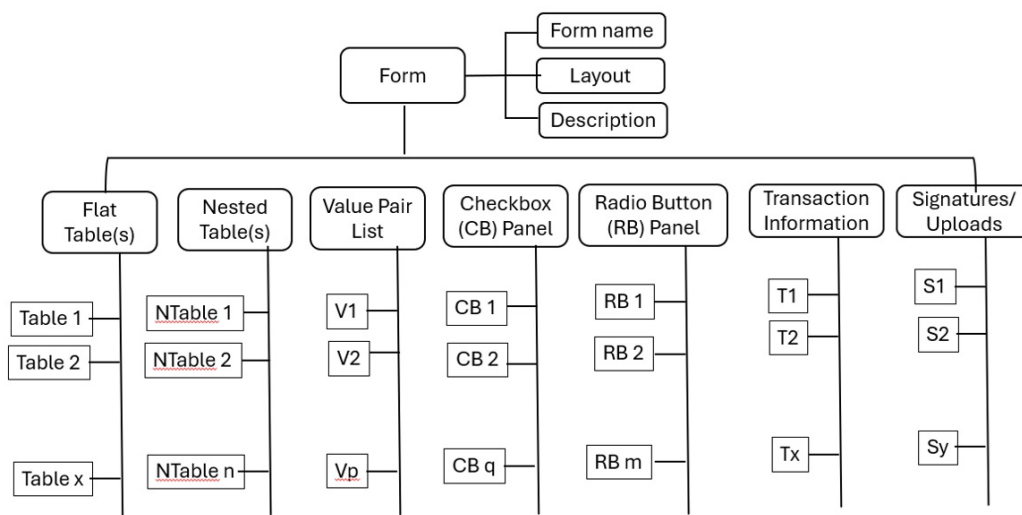


Figure 2. Possible elements in a form.

3.1. Employment Application

Considering the employment application from Figure 3, the employment application input tree organizes fields into logical categories:

- **Personal Data:** Name, address, phone numbers, work start date, salary expectations, education level
- **Position Information:** Work hours, shift preferences, employment status, work authorization, criminal background
- **Qualifications:** Educational background, special skills, references

Standard Application for Employment

It is our policy to comply with all applicable state and federal laws prohibiting discrimination in employment based on race, age, color, sex, religion, national origin, disability or other protected classifications.

Please carefully read and answer all questions. You will not be considered for employment if you fail to completely answer all the questions on this application. You may attach a résumé, but all questions **must** be answered.

"Employer"		Position applying for	
PERSONAL DATA			
Name (last, first, middle)			
Street Address and/or Mailing Address		City	State
Zip			
Home Telephone Number	Business Telephone Number	Cellular Telephone Number	
Date you can start work	Salary Desired	Do you have a High School Diploma or GED? Yes <input type="checkbox"/> No <input type="checkbox"/>	
POSITION INFORMATION Check all that you are willing to work			
Hours: Full Time <input type="checkbox"/>	Part Time <input type="checkbox"/>	Days Evenings <input type="checkbox"/>	Swing Graveyard Weekends <input type="checkbox"/>
Status: Regular <input type="checkbox"/>		Temporary <input type="checkbox"/>	
Are you authorized to work in the U.S. on an unrestricted basis?		Yes <input type="checkbox"/> No <input type="checkbox"/>	
Have you ever been convicted of a felony? (Convictions will not necessarily disqualify an applicant for employment.)		Yes <input type="checkbox"/> No <input type="checkbox"/>	
If yes, explain:			
Have you been told the essential functions of the job or have you been viewed a copy of the job description listing the essential functions of the job? Yes <input type="checkbox"/> No <input type="checkbox"/>			
Can you perform these essential functions of the job with or without reasonable accommodation? Yes <input type="checkbox"/> No <input type="checkbox"/>			
QUALIFICATIONS Please list any education or training you feel relates to the position applied for that would help you perform the work, such as schools, colleges, degrees, vocational or technical programs, and military training.			
	School Name	Degree	Address/City/State
School			
School			
Other			
SPECIAL SKILLS List any special skills or experience that you feel would help you in the position that you are applying for (leadership, organizations/teams, etc.)			
REFERENCES Please list three professional references not related to you, with full name, address, phone number, and relationship. If you don't have three professional references, then list personal, unrelated references.			
Name	Address/City/State	Phone	Relationship

Figure 3. Standard Application for Employment.

3.1.1. Input Tree

The input tree (Figure 5) structure represents the hierarchical organization of form fields for different document types.

Voter Registration Application

Before completing this form, review the General, Application, and State specific instructions.

Are you a citizen of the United States of America? <input type="checkbox"/> Yes <input type="checkbox"/> No Will you be 18 years old on or before election day? <input type="checkbox"/> Yes <input type="checkbox"/> No If you checked "No" in response to either of these questions, do not complete form. (Please see state-specific instructions for rules regarding eligibility to register prior to age 18.)		This space for office use only.	
1	<input type="checkbox"/> Mr. <input type="checkbox"/> Miss Last Name <input type="checkbox"/> Mrs. <input type="checkbox"/> Ms.	First Name	Middle Name(s) <input type="checkbox"/> Jr <input type="checkbox"/> II <input type="checkbox"/> Sr <input type="checkbox"/> III <input type="checkbox"/> IV
2	Home Address	Apt. or Lot #	City/Town State Zip Code
3	Address Where You Get Your Mail If Different From Above		City/Town State Zip Code
4	Date of Birth Month Day Year	5 Telephone Number (optional)	6 ID Number - (See item 6 in the instructions for your state)
7	Choice of Party (see item 7 in the instructions for your State)	8 Race or Ethnic Group (see item 8 in the instructions for your State)	
I have reviewed my state's instructions and I swear/affirm that: ■ I am a United States citizen ■ I meet the eligibility requirements of my state and subscribe to any oath required. ■ The information I have provided is true to the best of my knowledge under penalty of perjury. If I have provided false information, I may be fined, imprisoned, or (if not a U.S. citizen) deported from or refused entry to the United States.		Please sign full name (or put mark) ▲ Date: _____ Month Day Year	

If you are registering to vote for the first time: please refer to the application instructions for information on submitting copies of valid identification documents with this form.

Figure 4. Voter Registration Application.

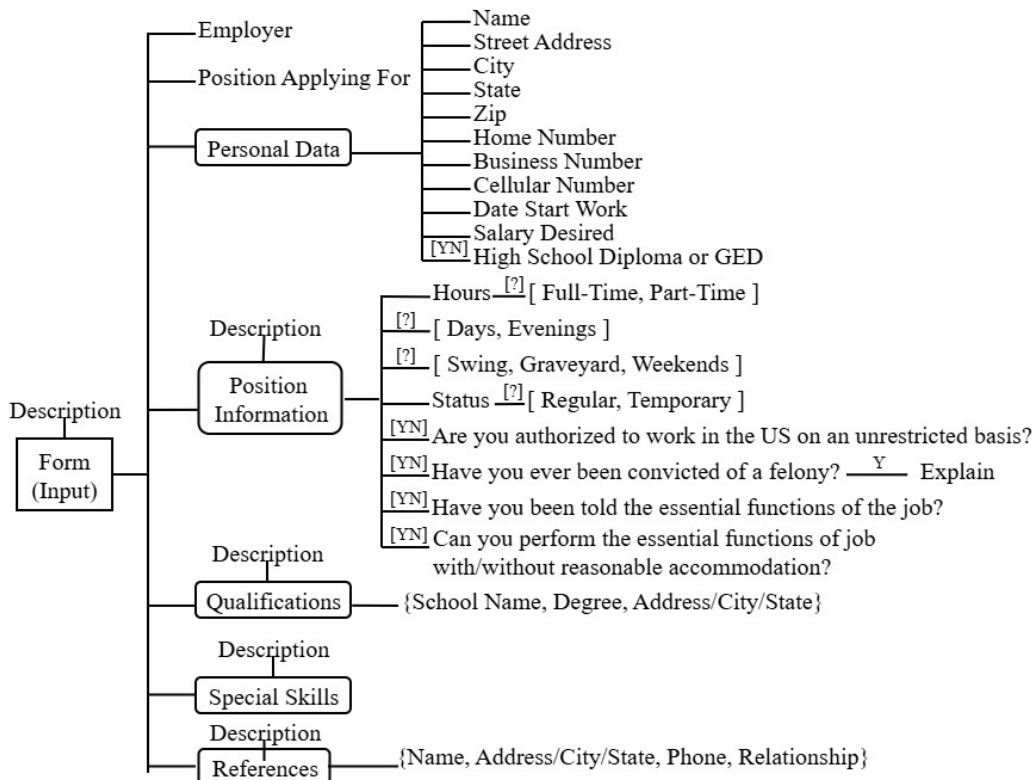


Figure 5. Employment application input tree structure.

3.1.2. Output Tree

The output tree (Figure 6) represents the hierarchical structure of the extracted fields after processing.

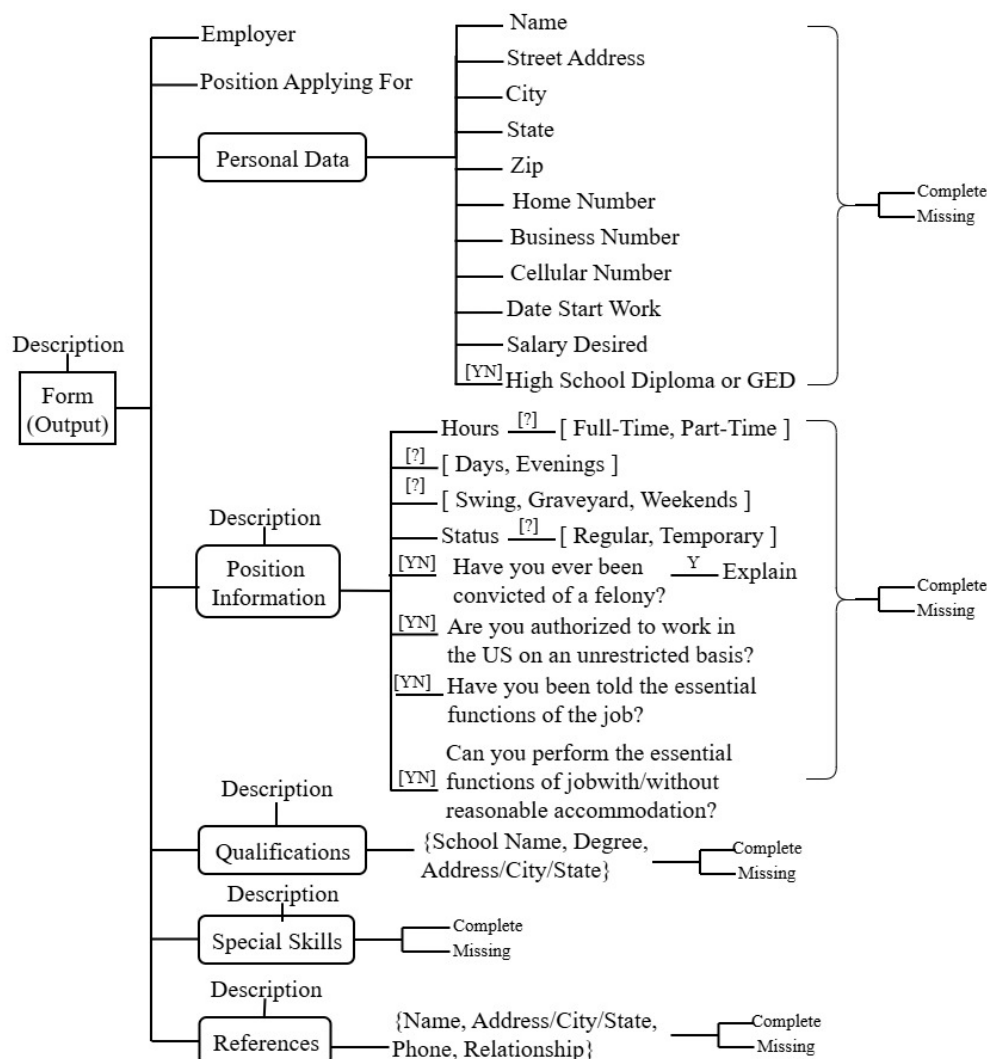


Figure 6. Employment application output tree structure.

Table 2 represents the test case in the job application form evaluated under normal lighting conditions using three applications, namely, Microsoft Lens, Amazon Textract, and Parsio.io.

Table 2. Example Test Case — Lighting (Normal Lighting: Job Application).

Application	Expected Output	Actual Output	Result
Microsoft Lens	Should be able to extract the majority, if not all, of the text.	Extracted approximately 90% of the text.	Pass
Amazon Textract	Same as expected above.	Accurately extracted most of the text, including key-value pairs, checkboxes, and table headers.	Pass
Parsio.io	Same as expected above.	Extracted all of the text and tables.	Pass

3.2. Voter Registration

Referring to the voter registration form from Figure 4, the voter registration input tree categorizes fields as:

- **Eligibility Requirements:** Citizenship status, age verification

- **Personal Information:** Name, address, date of birth, contact information
- **Registration Details:** ID number, party affiliation, demographic information
- **Declaration:** Signature and date

3.2.1. Input Tree

The input tree (Figure 7) structure represents the hierarchical organization of form fields for different document types.

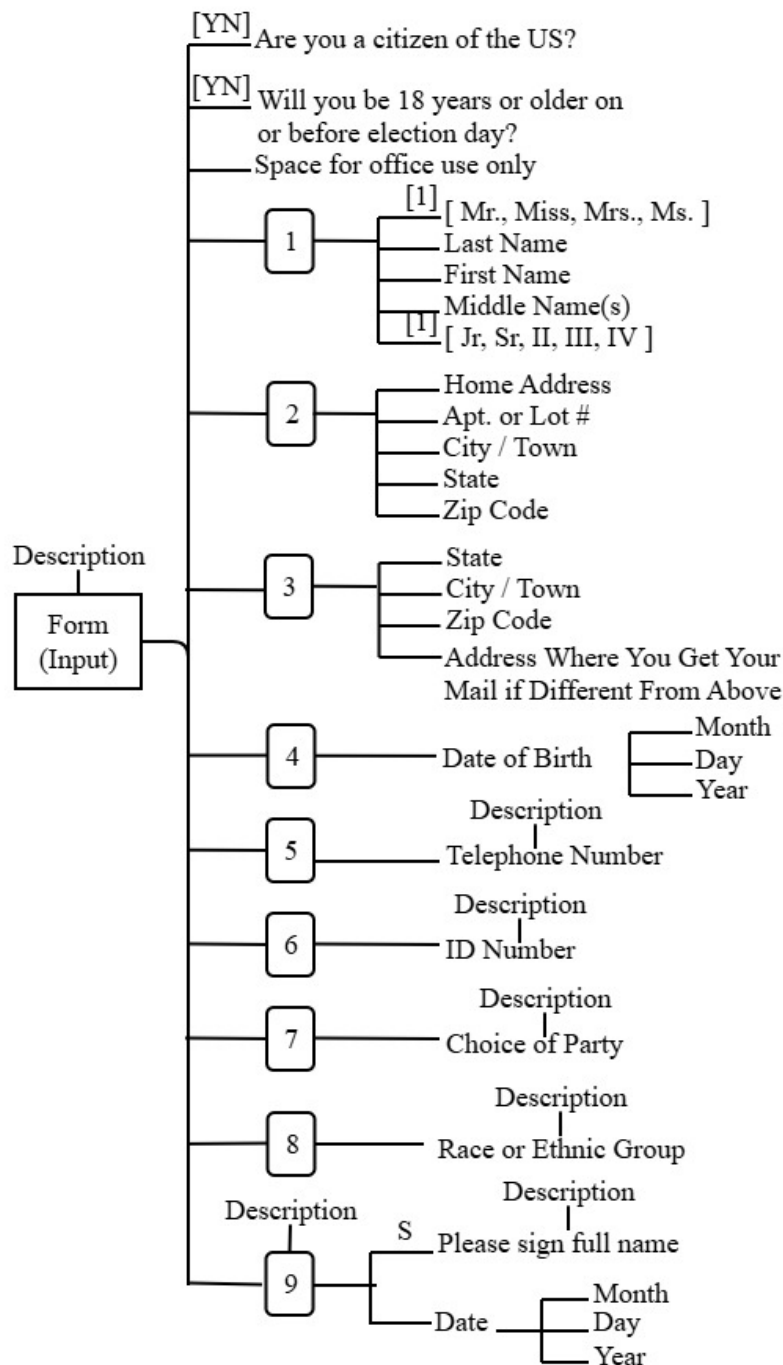


Figure 7. Voter registration input tree structure.

3.2.2. Output Tree

The output tree (Figure 8) structure represents the hierarchical organization of extracted fields after processing.

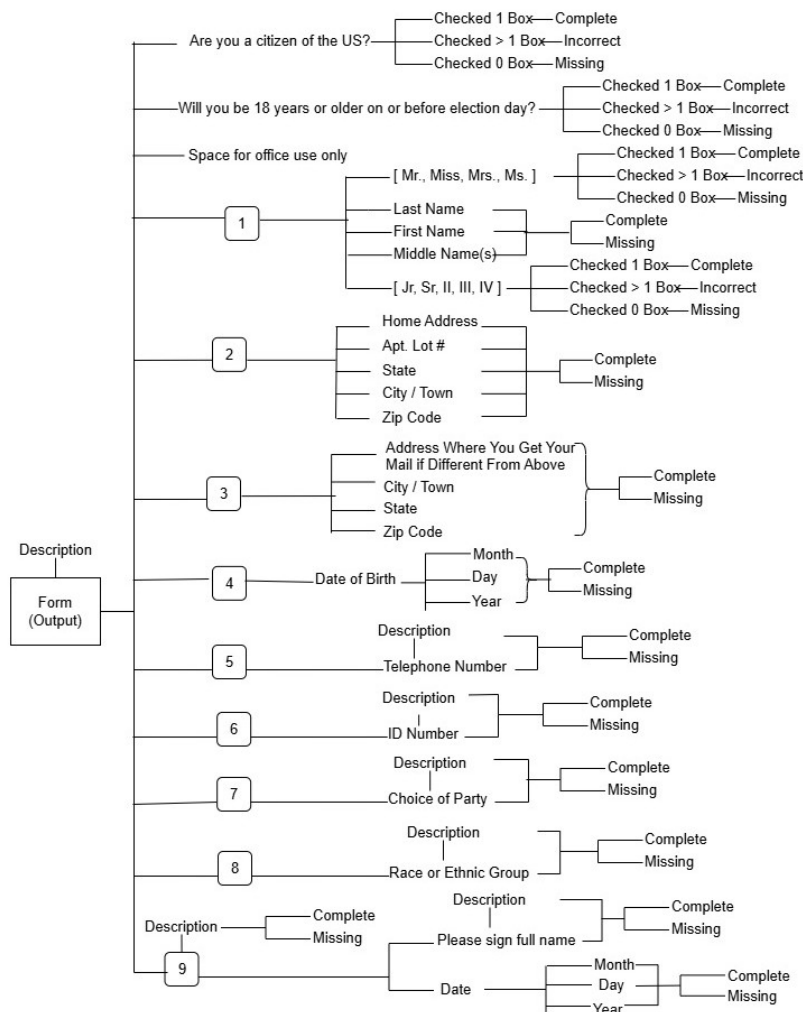


Figure 8. Voter registration output tree showing validation results.

Table 3 represents the test case in the voter registration form evaluated under normal lighting conditions using three applications, namely, Microsoft Lens, Amazon Textract, and Parsio.io.

Table 3. Example Test Case — Lighting (Light Lighting: Voter Form).

Application	Expected Output	Actual Output	Result
Microsoft Lens	The data should be mostly legible.	Failed to recognize telephone number, ID month or year, ZIP code, and name. Extracted only partial text.	Fail
Amazon Textract	Same as expected above.	Extracted text with moderate accuracy. Table alignment is mostly correct. Checkboxes are correctly detected. Signature not detected. Extracted most information correctly, with only the current date's month missing.	Pass
Parsio.io	Same as expected above.	Produced a more accurate table for checkboxes and name compared to the normal image.	Pass

3.3. Decision Table Implementation

The decision table provides a systematic approach to test case generation by mapping input conditions to expected outcomes. Table 4 presents the comprehensive decision table for OCR testing scenarios. The marking 'X' indicates that a field is present in the test case, and 'O' indicates the

observed outcome. A result is considered a pass when at least ninety percent of the total required fields are present and a fail when fewer than ninety percent of the required fields are present. "Form Complete" indicates that all required fields are present, representing one hundred percent completion, while "Missing Fields Detected" indicates that one or more required fields are missing.

Table 4. OCR Test Decision Table.

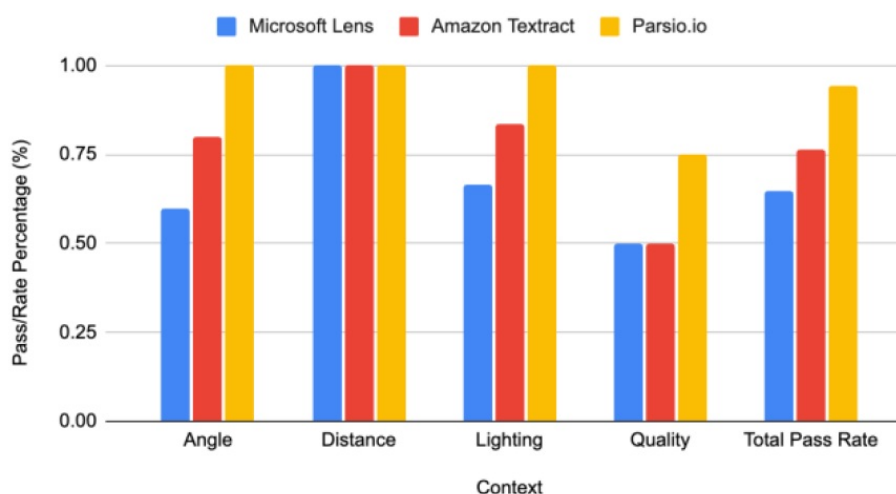
Category	Attribute	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Context											
Lighting Condition	Normal	X	X		X			X	X		X
	Too dark			X		X				X	
Shot Angle	Directly above	X			X				X		X
	Slight angle		X			X				X	
Image Quality	Good	X	X		X	X		X			
	Blurry			X			X				X
Employment Application – Input Fields											
Personal Data	Name	X	X	X				X	X	X	X
	Address/City/State/Zip	X	X					X		X	X
	Phone Numbers	X		X					X		
	Date Start Work	X	X					X			X
	Salary Desired	X	X			X		X			X
	Diploma/GED (Y/N)	X	X					X	X		
	Position Info	Work Hours (FT/PT)	X	X					X		
Shift Preference		X	X								
Status (Regular/Temp)		X									X
Authorized to Work (Y/N)		X	X							X	
Convicted Felony (Y/N)		X									X
Essential Functions (Y/N)		X	X							X	
Qualifications		School/Degree/Address	X	X					X		
Special Skills	Description	X							X		
References	Name/Address/Phone	X									X
Voter Registration – Input Fields											
Eligibility	Citizenship (Y/N)			X	X	X		X		X	
	Age ≥ 18 (Y/N)				X	X		X	X		
Name Info	Last/First/Middle Name				X			X			X
Address	Home/City/State/Zip			X	X					X	
	Mailing Address (if diff)				X						
Other Info	Date of Birth				X	X			X		
	Telephone Number		X		X						
	ID Number				X					X	
	Party Choice				X						
	Race/Ethnic Group				X						
Declaration	Signature				X	X					
	Date				X				X		
Output Validation											
Employment Application	Pass (Fields present $\geq 90\%$)	O	O					O	O	O	O
	Fail (Fields present $< 90\%$)			O							
	Form Complete (100% fields)	O									
	Missing Fields Detected		O	O				O	O	O	O
Voter Registration	Pass (Fields present $\geq 90\%$)				O	O		O	O	O	
	Fail (Fields present $< 90\%$)						O				O
	Form Complete (100% fields)				O						
	Missing Fields Detected					O	O				O

The decision table approach enables systematic generation of test cases that cover the complete space of possible OCR scenarios. Each test case (T1-T10) represents a specific combination of context factors and field completeness levels, allowing for comprehensive evaluation of OCR system performance.

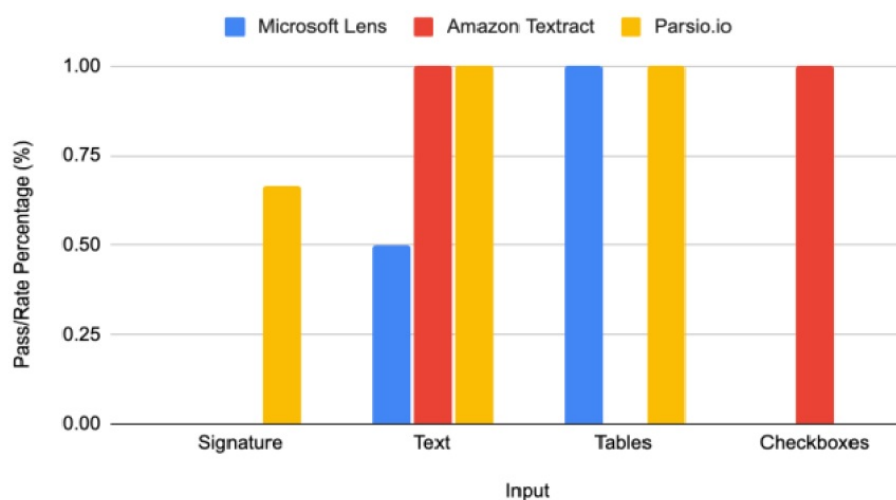
4. AI Test Model-Based Test Results Analysis

In the conventional testing aspect, three different black box testing methods were used, namely, boundary value testing, scenario testing, and category partitioning. In the section, each of the different applications and their testability was compared using category partition testing for AI testing. Each test case was organized into categories based on certain attributes; for example, the first half of the test cases were organized based on their context, and the second part in terms of input. The context was organized based on lighting, distance, angle, and quality. The number of test cases for the AI testing was a lot greater compared to the other black box testing methods. AI testing, along with category testing, generated nearly four times the number of test cases.

Figure 9 shows the text case results for context and input in (a) and (b), respectively. Microsoft Lens failed some test cases for angle, lighting, and quality. Amazon Textract failed comparatively fewer times, while Parsio.io successfully passed all the test cases except for the quality. All three applications successfully passed the distance-based test cases, demonstrating that camera distance variations from the form do not heavily impact results. In the case of input test cases, Amazon Textract failed table-related test cases due to its inability to accurately extract values from table cells. On the other hand, Microsoft Lens and Parsio.io performed better, each passing both test cases involving single and multiple table inputs. Parsio.io and Amazon Textract both passed text-related test cases, while Microsoft Lens failed primarily due to non-detection or misinterpretation of the text state.



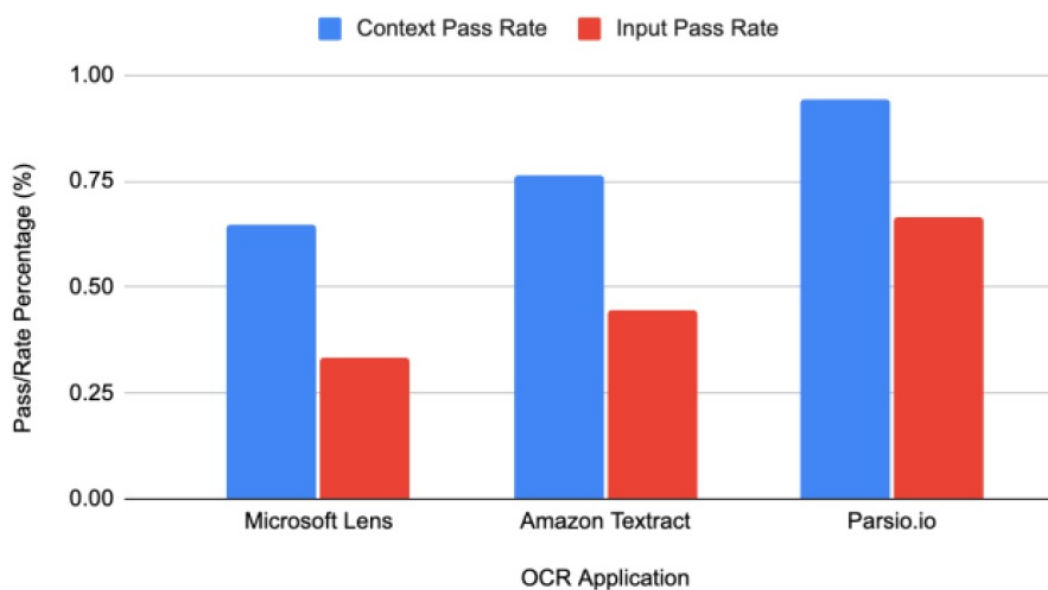
(a)



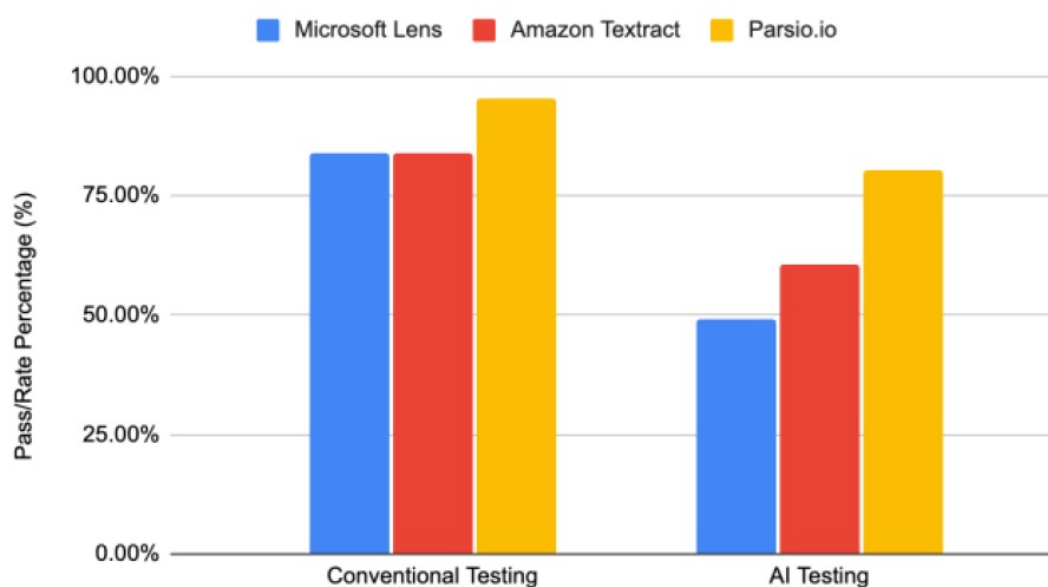
(b)

Figure 9. (a) Context Text Case Results. (b) Input Text Case Results.

Figure 10(a) shows that the context test case pass rate is higher than the input test case pass rate, while Figure 10(b) shows the comparative study between the conventional and AI testing across all three smart OCR apps.



(a)



(b)

Figure 10. (a) Comparison among context and input test result for OCR Smart Apps. (b) Conventional vs AI test result for OCR Smart Apps.

A total of 26 test cases were tested across three OCR applications: Microsoft Lens, Amazon Textract, and Parsio.io. Based on the observed failures seen in Figure 11, Microsoft Lens was the most error-prone, while Parsio.io demonstrated the highest overall accuracy. Amazon Textract showed intermediate results. The failures include both categories, i.e., context-related and input-related, exposing their limits in handling image quality, checkbox recognition, and signature detection.

Bug Analysis

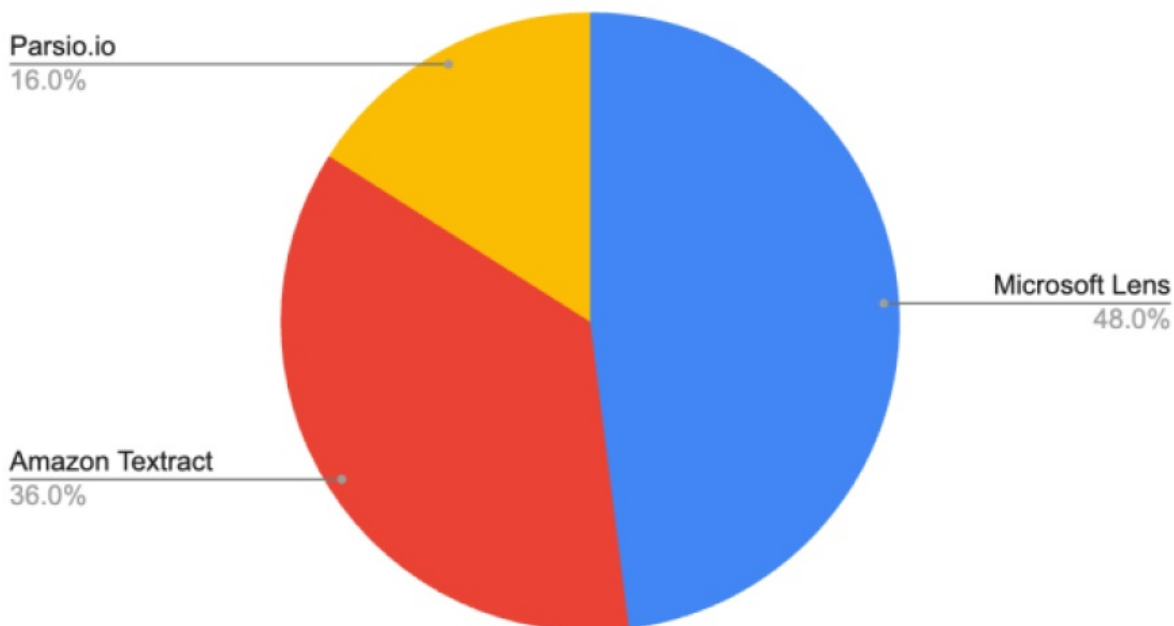


Figure 11. Overall Bug Comparison of all OCR Applications.

5. Evaluation Metrics and Test Case Analysis

In this section, the focus is on discussing several evaluation metrics, namely, accuracy, coverage criteria across the three different applications, and text complexity for both forms considered.

5.1. Framework for Accuracy and Coverage

As illustrated in Figure 2, a general-form document integrates multiple structural components such as flat tables, nested tables, value-pair lists, checkbox and radio-button panels, transaction metadata, and signature regions. The evaluation of OCR performance, therefore, extends beyond simple text recognition to capture both the *precision* and *completeness* of extraction. Two complementary indicators, accuracy and coverage, jointly characterize this quality.

Accuracy (A) quantifies the proportion of correctly recognized textual or symbolic elements relative to the ground-truth dataset:

$$A = \frac{\sum \text{Correct Elements}}{\sum \text{Total Elements}} \times 100. \quad (1)$$

A Levenshtein-based edit-distance model computes per-element correctness, aggregated by component type to yield a unified form-level accuracy that reflects consistency across text, numeric fields, and symbolic marks.

Coverage (C) expresses completeness of testing and validation:

$$C(V_p, CB_q, RB_m, T_x, Sig_n) = \frac{\sum \text{Validated Components}}{\sum \text{Total Components}} \times 100. \quad (2)$$

It captures the extent to which all structural and contextual form components have been exercised under varying environmental and input conditions. While high coverage indicates broad validation of lighting, angle, and multilingual variability, low coverage reveals untested categories or unverified dependencies between layout elements.

A *coverage deficit* arises when a component such as a checkbox, radio-button group, or nested table appears in the layout but lacks an associated test case. For example, a five-column table with only three verified fields yields $C_{Table} = (3/5) \times 100 = 60\%$. Such omissions reduce overall coverage even if accuracy elsewhere remains high, leaving untested substructures—“Tax Details,” “Discount

Breakdown,” or signature panels—that compromise robustness. Integrating this deficit analysis with component-level formulations (Table V) creates a closed-loop evaluation model quantifying not only what the OCR system recognizes correctly but also what remains untested.

Table 5. Accuracy and Coverage Metrics for Form Components in OCR Evaluation.

Component	Symbolic Variables	Accuracy Formula	Accuracy Description	Coverage Formula	Coverage Description
Form (overall)	$F_1 \dots F_x$	$Acc_F = \frac{\sum_{i=1}^x Correct(F_i)}{x} \times 100$	Aggregates all correctly recognized form fields.	$Cov_F = \frac{\sum_{i=1}^x Covered(F_i)}{x} \times 100$	Aggregates the correctly covered form fields.
Flat Table(s)	$T_1 \dots T_s$	$Acc_T = \frac{\sum_{i=1}^s Correct(T_i)}{s} \times 100$	Checks for the accurately filled flat tables.	$Cov_T = \frac{\sum_{i=1}^s Covered(T_i)}{s} \times 100$	Evaluates cell-level coverage across flat tables.
Nested Table(s)	$NT_1 \dots NT_n$	$Acc_{NT} = \frac{\sum_{i=1}^n Correct(NT_i)}{n} \times 100$	Checks for the accurately filled nested tables.	$Cov_{NT} = \frac{\sum_{i=1}^n Covered(NT_i)}{n} \times 100$	Checks for the input coverage of nested tables.
Radio Button Panel	$RB_1 \dots RB_m$	$Acc_{RB} = \frac{\sum_{i=1}^m Correct(RB_i)}{m} \times 100$	Correct if both position and state (checked/unchecked) are recognized.	$Cov_{RB} = \frac{\sum_{i=1}^m Covered(RB_i)}{m} \times 100$	Number of radio panels tested or validated at least once.
Checkbox Panel	$CB_1 \dots CB_q$	$Acc_{CB} = \frac{\sum_{i=1}^q Correct(CB_i)}{q} \times 100$	Checks visual identification and mark-state recognition.	$Cov_{CB} = \frac{\sum_{i=1}^q Covered(CB_i)}{q} \times 100$	Number of individual checkboxes or selection combinations validated.
Value Pair List	$V_1 \dots V_p$	$Acc_{VP} = \frac{\sum_{i=1}^p Correct(VP_i)}{p} \times 100$	Key-value accuracy: both label and corresponding value must match.	$Cov_{VP} = \frac{\sum_{i=1}^p Covered(VP_i)}{p} \times 100$	The proportion of field-value pairs correctly validated in a form.
Transaction Information	$TI_1 \dots TI_y$	$Acc_{TI} = \frac{\sum_{i=1}^y Correct(TI_i)}{y} \times 100$	Includes merchant, total, tax, and item-line accuracy.	$Cov_{TI} = \frac{\sum_{i=1}^y Covered(TI_i)}{y} \times 100$	The proportion of transaction or contextual attributes validated.
Signature(s)	$Sig_1 \dots Sig_z$	$Acc_{Sig} = \frac{\sum_{i=1}^z Correct(Sig_i)}{z} \times 100$	Checks for accurately uploaded or electronic signatures.	$Cov_{Sig} = \frac{\sum_{i=1}^z Covered(Sig_i)}{z} \times 100$	Ensures authentication and consent verification elements are fully tested.

5.2. Empirical Test-Case Evaluation

The empirical analysis applied these dual metrics across 10 decision-table-derived scenarios that represent combinations of lighting, angle, distance, image quality, and form completeness. Context- and input-level pass rates were recorded for three OCR applications—**Microsoft Lens**, **Amazon Textract**, and **Parsio.io**—on both employment-application and voter-registration forms.

Contextual Pass Rates: Microsoft Lens achieved 11/17, Textract 13/17, and Parsio.io 16/17 successful detections as in the Table 6(1), showing progressive robustness with image variability.

Table 6. (1) Context Test Case Pass Rate. (2) Input Test Case Pass Rate.

(1) Context Test Case Pass Rate			
Context	Microsoft Lens	Amazon Textract	Parsio.io
Angle	3/5	4/5	5/5
Distance	2/2	2/2	2/2
Lighting	4/6	5/6	6/6
Quality	2/4	2/4	3/4
Total Pass Rate	11/17	13/17	16/17
(2) Input Test Case Pass Rate			
Input	Microsoft Lens	Amazon Textract	Parsio.io
Signature	0/3	0/3	2/3
Text	1/2	2/2	2/2
Tables	2/2	0/2	2/2
Checkboxes	0/2	2/2	0/2
Input Pass Rate	3/9	4/9	6/9

Input-Level Pass Rates: For input components such as text, tables, checkboxes, and signatures, Microsoft Lens (3/9), Textract (4/9), and Parsio.io (6/9) again reflected incremental improvement as seen in Table 6(2). Combining both layers, the comprehensive pass percentages were 54%, 65%, and 85%, respectively, indicating relative extraction reliability across heterogeneous test contexts.

These empirical pass rates were then cross-referenced with component-level coverage calculations defined in Table V to establish unified performance indices that capture not only recognition fidelity but also completeness of structural validation.

5.3. Comparative Coverage Criteria Across Applications

To align metric theory with observed results, coverage was quantitatively assessed for each OCR system using:

$$C = \frac{\sum \text{Validated Components}}{\sum \text{Total Components}} \times 100, \quad (3)$$

where validated components encompassed field blocks, table cells, checkbox and radio-button panels, transactional metadata, and signature regions.

Microsoft Lens exhibited moderate completeness with approximately 55% form-level coverage. Flat and nested tables reached ~60%, value-pair lists ~70%, and signature or checkbox regions <40%. The tool excelled in textual recognition but lacked structural and semantic coverage under skewed or dim lighting conditions.

Amazon Textract achieved ≈73% coverage, performing strongly on structured tables (~85%) and value-pair or transaction fields (~75%), yet demonstrating only partial validation of checkboxes and radio buttons (~60%) and missing most signatures. Its engine favored grid-based extraction but underrepresented non-tabular metadata.

Parsio.io reached ≈88% coverage, offering full validation of table structures (100%), high value-pair accuracy (90%), moderate checkbox/radio recognition (50–70%), and strong signature verification (80–100%). Its adaptability to handwritten or irregular regions enabled superior completeness across heterogeneous forms.

The integrated evaluation confirms the *accuracy–coverage trade-off*: broader coverage increases generalization but may expose the model to noisier inputs that momentarily lower accuracy, whereas limited coverage can inflate precision scores but narrow operational robustness. Among the tested systems, **Parsio.io** achieved the most balanced profile, combining high coverage with strong per-field accuracy, thereby offering a reproducible benchmark for evaluating AI-based OCR testing frameworks [56].

Table 7. Comparative Coverage Across OCR Applications.

Application	Coverage	Dominant Strength
Microsoft Lens	≈55%	Text field recognition
Amazon Textract	≈73%	Structured tables & metadata
Parsio.io	≈88%	Comprehensive field & table capture

5.4. Form-Based Text Complexity and Structural Evaluation

Form-based text complexity refers to the level of structural, lexical, and interactive difficulty present in a document that must be interpreted by an OCR or AI-based form-processing system. Unlike free-text OCR, form-based extraction requires the model to correctly segment regions, understand hierarchical layout, interpret multimodal components such as checkboxes or signature fields, and perform label-value association. The employment application and voter registration forms analyzed in this study exhibit varying degrees of structural richness, conditional logic, and multimodal input types that directly influence OCR accuracy and coverage.

To systematically evaluate their complexity, we employ a component-based taxonomy consisting of: (1) structural hierarchy, (2) lexical and instructional density, (3) multimodal input field diversity, (4) conditional or rule-based dependencies, and (5) geometric alignment and formatting constraints. These factors collectively determine the cognitive and computational load placed on an OCR system.

5.4.1. Qualitative Complexity Characteristics

Both forms exhibit nontrivial segmentation, multimodal input interpretation, and layout dependencies:

- **Hierarchical Field Organization:** Each form contains multiple top-level sections with nested fields, requiring vertical and horizontal region detection.
- **Multimodal Input Types:** Text boxes, checkboxes, numeric date blocks, multi-column tables, and handwritten signatures create diverse extraction requirements.
- **Lexical Variation:** Long legal statements and short field labels coexist, requiring flexible OCR text-line parsing.
- **Formatting Density:** Tight alignment, adjacent fields, and multi-column layouts necessitate high-precision bounding-box segmentation.
- **Conditional Logic and Dependencies:** “Yes/No” eligibility checks, “If yes, explain” prompts, and signed affirmations introduce semantic dependencies not present in ordinary text.

Together, these characteristics place both forms within the *moderate-to-high complexity class*, with additional challenges arising from their respective structural designs.

5.4.2. Component-Level Complexity Evaluation

Using the evaluation metrics defined in the OCR framework, each form was decomposed into structural components and assigned weighted complexity scores (1–5 scale), indicating difficulty for OCR systems.

Employment Application Form Complexity

$$C_{\text{employment}} = \frac{(4 + 4 + 5 + 3 + 3 + 2)}{(6 \times 5)} \times 100 = \frac{21}{30} \times 100 \approx 70\%.$$

Thus, the employment application exhibits a 70% form-level complexity index, reflecting high multimodal density and challenging hierarchical segmentation.

Table 8. Component-Based Complexity Evaluation: Employment Application Form.

Component	Description	OCR Difficulty	Weight
Structural Hierarchy	Multiple sections with deeply nested sub-fields	Segmentation difficulty	4
Flat Tables	Qualifications and References multi-row tables	Cell-level alignment	4
Checkbox Panels	Work hours, shift, status, Y/N groups	Selection-state recognition	5
Value-Pair Lists	Names, address, phone, salary, etc.	Label-field association	3
Instructional Blocks	Long multi-line explanations	Dense text parsing	3
Numeric/Date Fields	Start date, salary, phone numbers	Structured numeric OCR	2

Voter Registration Form Complexity

$$C_{\text{voter}} = \frac{(5 + 5 + 4 + 3 + 4 + 3)}{(6 \times 5)} \times 100 = \frac{24}{30} \times 100 \approx 80\%.$$

Thus, the voter registration form yields an 80% form-level complexity index, indicating a greater structural difficulty relative to the employment form, primarily due to conditional logic and dense checkbox regions.

The voter registration form exhibits higher complexity due to its rule-based eligibility gates and high-density checkbox clusters, which significantly increase OCR burden. The employment application, though still complex, is more structured and predictable, resulting in a slightly lower overall complexity score.

These combined evaluations demonstrate that form-based OCR accuracy and coverage must account for diverse structural factors, and the dual-metric evaluation method provides a consistent framework for comparing heterogeneous document types.

Table 9. Component -Based Complexity Evaluation: Voter Registration Form.

Component	Description	OCR Difficulty	Weight
Eligibility Fields	Logic Conditional Yes/No gating questions	Semantic interpretation	5
Checkbox Panels	Titles, suffixes, race, party	High checkbox density	5
Value-Pair Fields	Name, address, DOB, ID number	Label alignment required	4
Numeric Blocks	DOB grid, phone blocks	Numeric segmentation	3
Signature/Oath Region	Dense legal text + signature/date	Mixed handwriting + text	4
Instructional Text	Multi-line state-specific instructions	Dense text OCR	3

6. Automation Results

The general strategy is to extract the inputs from various documents under different photo conditions. Photos of documents to extract text and other data were input by the web testing tool, Selenium. The expected output or correct answer is retrieved from a separate text file that has corresponding data extracted from the documents. Then, the AI-extracted data is compared to the ground truth. For every type of document feature, like text, tables, checkboxes, and signatures, the OCR-extracted data is compared with the expected ground truth using various software tools and techniques.

6.1. Microsoft Lens Execution Results

(a) Feature Based

Input	Data
Text	4/8
Tables	2/2
Checkboxes	0/2
Signature	0/1
Input Pass Rate	6/13

(b) Total Results

Metric	Count
Total Test Cases	13
Passed Test Cases	6
Bugs Detected	7

Microsoft Lens Observations:

- Microsoft Lens performs reliably with standard text fields and provides complete accuracy for table extraction; however, it fails to capture checkboxes and signatures, indicating limitations with symbolic or hand-drawn inputs.
- Detection quality drops under variable lighting and angle conditions, leading to multiple feature-level failures.
- With only 6 of 13 tests passed, Microsoft Lens shows limited structural understanding, making it better suited for basic text extraction rather than full form processing.

6.2. Amazon Textract Execution Results

(a) Feature Based

Input	Data
Text	6/8
Tables	2/2
Checkboxes	0/2
Signature	0/1
Input Pass Rate	8/13

(b) Total Results

Metric	Count
Total Test Cases	13
Passed Test Cases	8
Bugs Detected	5

Amazon Textract Observations:

- Amazon Textract provides strong structured-data extraction, with high text accuracy and reliable table recognition.
- It fails to interpret symbolic elements such as checkboxes and signatures, limiting usefulness for forms requiring user input.
- With 8 of 13 tests passed, Textract is more robust than Microsoft Lens, but its incomplete multi-modal coverage leads to moderate failure rates.

6.3. Parsio.io Execution Results

(a) Feature Based

Input	Data
Text	5/8
Tables	2/2
Checkboxes	1/2
Signature	1/1
Input Pass Rate	9/13

(b) Total Results

Metric	Count
Total Test Cases	13
Passed Test Cases	9
Bugs Detected	4

Parsio.io Observations:

- Parsio.io demonstrates the most balanced performance, handling text, tables, checkboxes, and signatures more effectively than the other tools.
- Its ability to extract both structured and symbolic data makes it the most reliable choice for real-world automated document processing.
- With 9 of 13 tests passed and the lowest bug count, Parsio.io maintains consistent accuracy even under challenging input or environmental variations.

7. Conclusions

In this paper, a systematic and structured method of testing and evaluation of Smart OCR applications has been proposed by means of including decision tables, context trees, and component-level complexity modeling. In contrast to traditional OCR tests that focus on individual accuracy measures or benchmark on single datasets, the suggested framework can be used to perform a holistic validation on document structure, environmental variability, and multimodal input scenarios. The explicit modeling of input, output, and contextual conditions allows the generation of the test cases to be automated and replicate the real-world usage conditions as opposed to the ideal laboratory settings.

Experimental analysis of employment application and voter registration forms showed that both the environmental context and complexity of the structure have a strong impact on the OCR performance. The findings demonstrated that there are still recurrent drawbacks in checkbox, radio-button, and signature recognition among popular OCR systems, where text and tabular recognition are still strong. The combination of accuracy and coverage measures was of essential importance in revealing these gaps because high recognition accuracy was frequently misleading about the unfinished validation of document parts. The complexity analysis also revealed that documents containing a large amount of conditional logic and multimodal fields present significantly greater cognitive and computational demands on OCR systems.

The results of comparisons of Microsoft Lens, Amazon Textract, and Parsio.io identified significant trade-offs between structured customer data extraction, multimodal robustness, and general coverage. Parsio.io was the most balanced, with higher coverage and flexibility with heterogeneous document elements, whereas other tools were strong in certain aspects but did not have the overall form knowledge. The results support the idea of transcending single-metric OCR evaluation to the multidimensional test models that capture the operational risk and deployment realities.

Overall, the proposed framework offers an expandable and reusable foundation for AI-based OCR testing, which aids in system selection, regression testing, and quality control of actual document pro-

cessing lines. To further improve the resilience and scalability of intelligent document understanding systems, future research will extend this technique to other document domains, incorporate adversarial and synthetic data augmentation, and investigate automated complexity-aware test optimization.

Author Contributions: J.G.: Conceptualization, formal analysis, supervision, review, and administration; R.A.: validation, data curation, methodology, writing-review, and editing; J.N.: resources, original draft preparation, methodology; R.V.: original draft preparation, case study, data curation, figure drafting; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: There is no conflict of interest.

References

29. Gao, J.; Tao, C.; Jie, D.; Lu, S. What Is AI Software Testing and Why? In Proceedings of the Proceedings of the IEEE International Conference on Service-Oriented System Engineering, 2019, pp. 27–2709.
30. Tao, C.; Gao, J.; Wang, T. Testing and Quality Validation for AI Software—Perspectives, Issues, and Practices. *IEEE Access* **2019**, *7*, 120164–120175.
31. Felderer, M.; Ramler, R. Quality Assurance for AI-Based Systems: Overview and Challenges. Cornell University Library, 2021.
32. Gao, J.; Patil, P.H.; Lu, S.; Cao, D.; Tao, C. Model-Based Test Modeling and Automation Tool for Intelligent Mobile Apps. In Proceedings of the Proceedings of the IEEE International Conference on Service-Oriented System Engineering, 2021, pp. 1–10.
33. Chen, T.Y.; Cheung, S.C.; Yiu, S.M. Metamorphic Testing: A New Approach for Generating Next Test Cases. Technical Report HKUST-CS98-01, Hong Kong University of Science and Technology, 1998.
34. He, Y.; Razi, M.; Gao, J.; Tao, C. A Framework for Autonomous Vehicle Testing Using Semantic Models. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, 2023, pp. 66–73.
35. Zhu, H.; Liu, D.; Bayley, I.; Harrison, R.; Cuzzolin, F. Datamorphic Testing: A Method for Testing Intelligent Applications. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, 2019, pp. 149–156.
36. Berend, D. Distribution Awareness for AI System Testing. In Proceedings of the Proceedings of the IEEE/ACM International Conference on Software Engineering, 2021, pp. 96–98.
37. Huang, J.; Zhang, J.; Wang, W.; He, P.; Su, Y.; Lyu, M.R. AEON: A Method for Automatic Evaluation of NLP Test Cases. *arXiv* **2022**. arXiv preprint.
38. Li, L.; Huang, Y.; Cui, X.; Cheng, X.; Liu, X. On Testing and Evaluation of Artificial Intelligence Models. In Proceedings of the Proceedings of the IEEE International Conference on Sensors, Electronics and Computer Engineering, 2023, pp. 92–97.
39. Zhu, H.; Bayley, I.; Liu, D.; Zheng, X. Automation of Datamorphic Testing. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, 2020, pp. 64–72.
40. Mori, S.; Suen, C.Y.; Yamamoto, K. Historical Review of OCR Research and Development. *Proceedings of the IEEE* **1992**, *80*, 1029–1058.
41. Mittal, R.; Garg, A. Text Extraction Using OCR: A Systematic Review. In Proceedings of the Proceedings of the International Conference on Inventive Research in Computing Applications, 2020, pp. 357–362.
42. Dessai, B.; Patil, A. A Deep Learning Approach for Optical Character Recognition of Handwritten Devanagari Script. In Proceedings of the Proceedings of the International Conference on Intelligent Computing, Instrumentation and Control Technologies, 2019, pp. 1160–1165.
43. Gan, J.; Wang, W.; Lu, K. Recognizing Online Handwritten Chinese Characters via One-Dimensional CNN. *Information Sciences* **2019**, *478*, 375–390.
44. Rice, S.V. Measuring the Accuracy of Page-Reading Systems, 1996.
45. Ponnuru, M.; Ponnmalar, P.S.; Likhitha, A.; Tanusree, B.; Chaitanya, G.G. Image-Based Extraction of Prescription Information Using OCR-Tesseract. *Procedia Computer Science* **2024**, *235*, 1077–1086.
46. Anakpluek, N.; Pasanta, W.; Chantharasukha, L.; Chokratansombat, P.; Kanjanakaew, P.; Siriborvornratanakul, T. Improved Tesseract Optical Character Recognition Performance on Thai Document Datasets. *Big Data Research* **2025**, *39*, 100508. <https://doi.org/10.1016/j.bdr.2025.100508>.

47. Pino, R.; Mendoza, R.; Sambayan, R. MaBaybay-OCR: A MATLAB-Based Baybayin Optical Character Recognition Package. *SoftwareX* **2025**, *29*, 102003.
48. Wilkinson, R.A.; Garris, M.D.; Geist, J.C. Machine-Assisted Human Classification of Segmented Characters for OCR Testing and Training. *Proceedings of SPIE* **1993**, *1906*, 208–217.
49. Drobac, S.; Lindén, K. Optical Character Recognition with Neural Networks and Post-Correction with Finite-State Methods. *International Journal on Document Analysis and Recognition* **2020**.
50. Arica, N.; Yarman-Vural, F.T. Optical Character Recognition for Cursive Handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2002**, *24*, 801–813.
51. Alotaibi, F.; Abdullah, M.T.; Abdullah, R.B.H.; Rahmat, R.W.O.K.; Hashem, I.A.T.; Sangaiah, A.K. Optical Character Recognition for Quranic Image Similarity Matching. *IEEE Access* **2018**, *6*, 554–562.
52. Wang, X.; et al. Intelligent Micron Optical Character Recognition of DFB Chips Using Deep Convolutional Neural Networks. *IEEE Transactions on Instrumentation and Measurement* **2022**, *71*, 1–9.
53. Rasmussen, L.V.; Peissig, P.L.; McCarty, C.A.; Starren, J. Development of an Optical Character Recognition Pipeline for Handwritten Form Fields from an Electronic Health Record. *Journal of the American Medical Informatics Association* **2012**, *19*, e90–e95.
54. Gao, J.; Agarwal, R. AI Test Modeling for Computer Vision Systems—A Case Study. *Computers* **2025**, *14*, 396. <https://doi.org/10.3390/computers14090396>.
55. Gao, J.; Agarwal, R.; Garsole, P. AI Testing for Intelligent Chatbots—A Case Study. *Software* **2025**, *4*, 12. <https://doi.org/10.3390/software4020012>.
56. Shu, J.; Miu, B.J.; Chang, E.; Gao, J.; Liu, J. Computer Vision Intelligence Test Modeling and Generation: A Case Study on Smart OCR. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, Shanghai, China, 2024; pp. 21–28. <https://doi.org/10.1109/AITest62860.2024.00011>.

References

29. Gao, J.; Tao, C.; Jie, D.; Lu, S. What Is AI Software Testing and Why? In Proceedings of the Proceedings of the IEEE International Conference on Service-Oriented System Engineering, 2019, pp. 27–2709.
30. Tao, C.; Gao, J.; Wang, T. Testing and Quality Validation for AI Software—Perspectives, Issues, and Practices. *IEEE Access* **2019**, *7*, 120164–120175.
31. Felderer, M.; Ramler, R. Quality Assurance for AI-Based Systems: Overview and Challenges. Cornell University Library, 2021.
32. Gao, J.; Patil, P.H.; Lu, S.; Cao, D.; Tao, C. Model-Based Test Modeling and Automation Tool for Intelligent Mobile Apps. In Proceedings of the Proceedings of the IEEE International Conference on Service-Oriented System Engineering, 2021, pp. 1–10.
33. Chen, T.Y.; Cheung, S.C.; Yiu, S.M. Metamorphic Testing: A New Approach for Generating Next Test Cases. Technical Report HKUST-CS98-01, Hong Kong University of Science and Technology, 1998.
34. He, Y.; Razi, M.; Gao, J.; Tao, C. A Framework for Autonomous Vehicle Testing Using Semantic Models. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, 2023, pp. 66–73.
35. Zhu, H.; Liu, D.; Bayley, I.; Harrison, R.; Cuzzolin, F. Datamorphic Testing: A Method for Testing Intelligent Applications. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, 2019, pp. 149–156.
36. Berend, D. Distribution Awareness for AI System Testing. In Proceedings of the Proceedings of the IEEE/ACM International Conference on Software Engineering, 2021, pp. 96–98.
37. Huang, J.; Zhang, J.; Wang, W.; He, P.; Su, Y.; Lyu, M.R. AEON: A Method for Automatic Evaluation of NLP Test Cases. *arXiv* **2022**. arXiv preprint.
38. Li, L.; Huang, Y.; Cui, X.; Cheng, X.; Liu, X. On Testing and Evaluation of Artificial Intelligence Models. In Proceedings of the Proceedings of the IEEE International Conference on Sensors, Electronics and Computer Engineering, 2023, pp. 92–97.
39. Zhu, H.; Bayley, I.; Liu, D.; Zheng, X. Automation of Datamorphic Testing. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, 2020, pp. 64–72.
40. Mori, S.; Suen, C.Y.; Yamamoto, K. Historical Review of OCR Research and Development. *Proceedings of the IEEE* **1992**, *80*, 1029–1058.
41. Mittal, R.; Garg, A. Text Extraction Using OCR: A Systematic Review. In Proceedings of the Proceedings of the International Conference on Inventive Research in Computing Applications, 2020, pp. 357–362.

42. Dessai, B.; Patil, A. A Deep Learning Approach for Optical Character Recognition of Handwritten Devanagari Script. In Proceedings of the Proceedings of the International Conference on Intelligent Computing, Instrumentation and Control Technologies, 2019, pp. 1160–1165.
43. Gan, J.; Wang, W.; Lu, K. Recognizing Online Handwritten Chinese Characters via One-Dimensional CNN. *Information Sciences* **2019**, *478*, 375–390.
44. Rice, S.V. Measuring the Accuracy of Page-Reading Systems, 1996.
45. Ponnuru, M.; Ponnmalar, P.S.; Likhitha, A.; Tanusree, B.; Chaitanya, G.G. Image-Based Extraction of Prescription Information Using OCR-Tesseract. *Procedia Computer Science* **2024**, *235*, 1077–1086.
46. Anakpluek, N.; Pasanta, W.; Chantharasukha, L.; Chokratansombat, P.; Kanjanakaew, P.; Siriborvornratanakul, T. Improved Tesseract Optical Character Recognition Performance on Thai Document Datasets. *Big Data Research* **2025**, *39*, 100508. <https://doi.org/10.1016/j.bdr.2025.100508>.
47. Pino, R.; Mendoza, R.; Sambayan, R. MaBaybay-OCR: A MATLAB-Based Baybayin Optical Character Recognition Package. *SoftwareX* **2025**, *29*, 102003.
48. Wilkinson, R.A.; Garris, M.D.; Geist, J.C. Machine-Assisted Human Classification of Segmented Characters for OCR Testing and Training. *Proceedings of SPIE* **1993**, *1906*, 208–217.
49. Drobac, S.; Lindén, K. Optical Character Recognition with Neural Networks and Post-Correction with Finite-State Methods. *International Journal on Document Analysis and Recognition* **2020**.
50. Arica, N.; Yarman-Vural, F.T. Optical Character Recognition for Cursive Handwriting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2002**, *24*, 801–813.
51. Alotaibi, F.; Abdullah, M.T.; Abdullah, R.B.H.; Rahmat, R.W.O.K.; Hashem, I.A.T.; Sangaiah, A.K. Optical Character Recognition for Quranic Image Similarity Matching. *IEEE Access* **2018**, *6*, 554–562.
52. Wang, X.; et al. Intelligent Micron Optical Character Recognition of DFB Chips Using Deep Convolutional Neural Networks. *IEEE Transactions on Instrumentation and Measurement* **2022**, *71*, 1–9.
53. Rasmussen, L.V.; Peissig, P.L.; McCarty, C.A.; Starren, J. Development of an Optical Character Recognition Pipeline for Handwritten Form Fields from an Electronic Health Record. *Journal of the American Medical Informatics Association* **2012**, *19*, e90–e95.
54. Gao, J.; Agarwal, R. AI Test Modeling for Computer Vision Systems—A Case Study. *Computers* **2025**, *14*, 396. <https://doi.org/10.3390/computers14090396>.
55. Gao, J.; Agarwal, R.; Garsole, P. AI Testing for Intelligent Chatbots—A Case Study. *Software* **2025**, *4*, 12. <https://doi.org/10.3390/software4020012>.
56. Shu, J.; Miu, B.J.; Chang, E.; Gao, J.; Liu, J. Computer Vision Intelligence Test Modeling and Generation: A Case Study on Smart OCR. In Proceedings of the Proceedings of the IEEE International Conference on Artificial Intelligence Testing, Shanghai, China, 2024; pp. 21–28. <https://doi.org/10.1109/AITest62860.2024.00011>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.