

Automated government document analysis and classification using robotic process automation-ChatGPT model

Adisorn Phonsupharak¹, Rattanawadee Panthong²

¹Department of Technology and Digital Data Management, School of Information and Communication Technology, University of Phayao, Phayao, Thailand

²Department of Information Technology, School of Information and Communication Technology, University of Phayao, Phayao, Thailand

Article Info

Article history:

Received Jul 6, 2025

Revised Jul 31, 2026

Accepted Aug 12, 2026

Keywords:

ChatGPT API

Document classification

Intelligent document processing

Optical character recognition

Robotic process automation

ABSTRACT

Government offices still process most official documents manually, and the cost is familiar: slow turnaround, inconsistent filing, and data-entry mistakes that are easy to make when every document must be read, classified, and re-keyed. This study tackles that problem with an automated document-processing system for the University of Phayao Document Management System (UP-DMS) that brings together robotic process automation (RPA), optical character recognition (OCR), and the chat generative pre-trained transformer application programming interface (ChatGPT API). A software bot retrieves each document, OCR pulls out the text, ChatGPT classifies the document and summarizes its contents, and the bot writes the result back into UP-DMS. On the OCR pilot, the system reached 89.00% accuracy, enough to support the later stages reliably. Tested on 105 authentic administrative documents, it classified the three main document types at 95.24% accuracy (macro-F1 = 0.952) and the fifteen sub-categories at 82.86%. Automation also cut the average handling time from roughly 4.5 minutes per document to 0.38 minutes, a 91.57% reduction. When 24 administrative staff trialed the system, they rated it highly for usability ($\bar{x} = 4.49$), performance ($\bar{x} = 4.49$), and practical impact ($\bar{x} = 4.60$). The findings point to a workable way of reducing workload while improving both the speed and the accuracy of document handling in public and academic institutions.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Rattanawadee Panthong

Department of Information Technology, School of Information and Communication Technology

University of Phayao

Phayao, Thailand

Email: rattanawadee.pa@up.ac.th

1. INTRODUCTION

Modern organizations rely heavily on document-based workflows, and the volume of digital content requiring management continues to grow. This is why both public agencies and private firms have adopted document management systems (DMS) to facilitate efficient storage, retrieval, and processing of information. A well-designed DMS enhances productivity, improves data accuracy, reduces human error, and supports timely decision-making [1], [2]. Conventional paper-based document handling is deficient in each of these respects. Data ends up stored without structure, retrieval drags, and security is often weak—problems that lead to lost files, delayed access, and workflows that quietly waste time [3]. As the amount of information grows, the need for intelligent, automated document management becomes increasingly compelling.

Three technologies have now matured enough to change this picture. Robotic process automation (RPA) handles repetitive, rule-based steps and takes routine work off people's hands [4], [5]. Optical character recognition (OCR) turns scanned or image-based pages into machine-readable, searchable text [6]–[8]. The chat generative pre-trained transformer application programming interface (ChatGPT API), built on a large language model, adds what the other two lack: it understands language, interprets content, summarizes, and generates structured responses [9], [10].

What remains absent is a unified framework that integrates these technologies into a coherent operational pipeline. Most existing studies look at these tools in isolation, or apply them to general business automation, software testing, OCR alone, or chatbots [11], [12]. Government document workflows have drawn far less attention, even though they are exactly the hard case: the documents are semi-structured, written in Thai, noisy after scanning, and bound by procedure [13]. While intelligent document processing pipelines that combine OCR, classification, and workflow automation are well established for structured documents such as invoices and financial forms, this specific combination of rule-based automation, large language model-based semantic understanding, and fuzzy-logic uncertainty routing has not, to the authors' knowledge, been applied to and evaluated on Thai government administrative documents in a real operational setting.

This study proposes such a framework, built specifically for Thai government documents. It is more than a chain of RPA, OCR, and ChatGPT calls: it couples rule-based automation and OCR extraction with language-model semantic reasoning, a fuzzy-logic layer for uncertain cases, and automated workflow execution. This combination enables the system to handle the semi-structured, noise-prone Thai-language documents that challenge conventional rule-based approaches.

The work makes three contributions. It introduces a hybrid intelligent-automation framework for government document workflows. It evaluates that framework on real administrative documents across OCR quality, classification accuracy, processing speed, and user satisfaction. It extends the design with uncertainty-aware mechanisms and a roadmap toward more scalable artificial intelligence (AI)—fuzzy-logic classification, knowledge representation, reinforcement learning, and workflow optimization.

2. RELATED WORKS

Documents at the University of Phayao come in three kinds. External letters are official correspondence with outside agencies and carry the university format and emblem. Internal letters are interdepartmental memos that keep coordination fast and systematic. Forms collect data or requests—job applications, leave requests, authorizations, and the like.

2.1. Robotic process automation

RPA is a software technology that builds "bots" to carry out repetitive tasks in place of people, using rule-based decisions and image recognition to work faster, make fewer errors, and stay competitive [4]. It pairs naturally with AI and machine learning (ML), which let it take on the high-volume, repetitive work that people used to do by hand. In banking and finance, for instance, bots reach into existing applications to process transactions, move data, and produce statements, improving efficiency, security, and customer satisfaction [14], [15]. As one of the main drivers of digital transformation, RPA connects with technologies such as AI and the internet of things (IoT) and automates workflows inside hybrid cloud-based ERP systems, although how it fits with Industry 4.0 still needs more study [4]. Speed and accuracy come with risks, though. RPA can open the door to cyber fraud and data breaches, so careful implementation, governance, and rule configuration are needed to keep those in check while also addressing data privacy, security, and workforce concerns [14], [16].

The gains are well documented. One bot, named Gul, handled email and updated Google AdWords campaigns, cutting a 30-minute task to under three minutes [17]. Another project integrated RPA with systems, applications, and products in data processing (SAP) in the order-to-cash process, shortening execution time and cutting errors [18]. A notable limitation, however, is the maintenance overhead: such systems require ongoing monitoring and updates. RPA also makes automation accessible to non-technical users. Tasks such as moving spreadsheet data into a database become straightforward, and the time saved is substantial compared with writing the same logic in Java or Python [5].

2.2. ChatGPT API

The ChatGPT API enables developers to integrate conversational AI into their applications for real-time, human-like exchanges. It supports multi-turn conversations, keeps track of context, gives control over system and user messages, and exposes parameters such as temperature and maximum tokens. That makes it a good fit for intelligent chatbots, virtual assistants, and automation workflows that hinge on understanding language [19].

Because it is built on the generative pre-trained transformer (GPT) architecture, ChatGPT handles text generation, summarization, question answering, translation, and interpretation, and it has been used for code generation, documentation, and troubleshooting—though accuracy, data security, ethics, domain knowledge, and training-data bias remain open concerns [9], [20]. In education, it has helped draft announcements, prepared materials, and support academic writing, even if over-reliance is a worry [21]. In automation testing, pairing it with Selenium and Pytest supports self-healing scripts that cut maintenance [12]. Bringing AI, ML, and natural language processing (NLP) together sharpens the decisions an RPA bot can make, and earlier work on RPA process selection backs the idea of using language-model interpretation inside RPA workflows [16].

2.3. Optical character recognition

OCR turns printed, scanned, or image-based text into a machine-readable form. It stays difficult because document images differ so much—in language, font, layout, resolution, noise, skew, stamps, signatures, and handwriting—which is why it draws on image processing, pattern recognition, and language-specific methods [8]. How well it works depends on the script, the language, the quality of the document, and the preprocessing applied [7] and earlier studies have tackled text detection in cluttered scenes and post-processing under noise [6], [22].

Deep learning now underpins most modern OCR. PaddleOCR chains text detection, direction classification, and recognition for multilingual work [23]. EasyOCR reads text from both documents and natural scenes across languages [24] and cloud services such as Google Vision OCR offer API-based extraction that slots neatly into automation [25]. These options are what motivate the comparison of Google Vision with PaddleOCR and EasyOCR on Thai government documents in this study.

3. METHOD

This study builds an automated model for classifying and analyzing documents by combining UiPath Studio (version 2026.0.198 STS), Google Cloud OCR, and ChatGPT. The workflow runs in five stages: i) designing and implementing the RPA workflow, ii) extracting text from scanned documents with OCR, iii) prompt engineering and integration with the ChatGPT API for content understanding and summarization, iv) classifying documents into predefined categories, and v) automated data entry and storage into the University of Phayao Document Management System (UP-DMS). Figure 1 shows the complete process.

3.1. Dataset

The evaluation corpus is made up of 105 authentic Thai-language administrative documents drawn from the UP-DMS. To keep the sample representative of everyday government work, the documents were taken from the high-volume routine operations that the central administration unit processes. The set is strictly balanced at the main-class level, with exactly 35 documents each for external letters, internal letters, and other forms (total $n = 105$). Across these, the documents span fifteen sub-categories in the proportions they naturally occur in daily tasks. All personally identifiable information (PII) was masked before analysis.

For the experimental setup, the proposed RPA-ChatGPT framework is a zero-shot classifier and therefore needs no training data. The comparative baselines in subsection 4.7 do, so a stratified split was used there: the traditional supervised models were trained on 74 documents (70%) and evaluated on a held-out set of 31 documents (30%), and the RPA-ChatGPT model was scored on that same test set so the comparison stays direct and fair. Ground-truth labels for document classification were assigned manually by two annotators: one administrative expert from the central division with over five years of experience in university document management, and one researcher with domain knowledge of Thai government document conventions. Each document was independently labeled by both annotators for main class and subcategory, using a standardized guideline derived from the UP-DMS document taxonomy and the Government Correspondence Regulation (2021). The guideline defined criteria for distinguishing each of the three main categories and the 15 subcategories, with illustrative examples for ambiguous cases such as combined-function forms and interdepartmental memoranda. In cases of disagreement, the final label was determined through discussion and consensus. The inter-annotator agreement rate was 95.24% (Cohen's kappa = 0.928), indicating almost perfect agreement.

3.2. Proposed method

The method follows a multi-stage design that ties together RPA, OCR, and the ChatGPT language model to automate document classification and summarization. It starts by retrieving documents

automatically, extracts their text with OCR, and then analyzes and summarizes them through the ChatGPT API. The processed data are finally written into the target system. Figure 1 shows the full architecture.

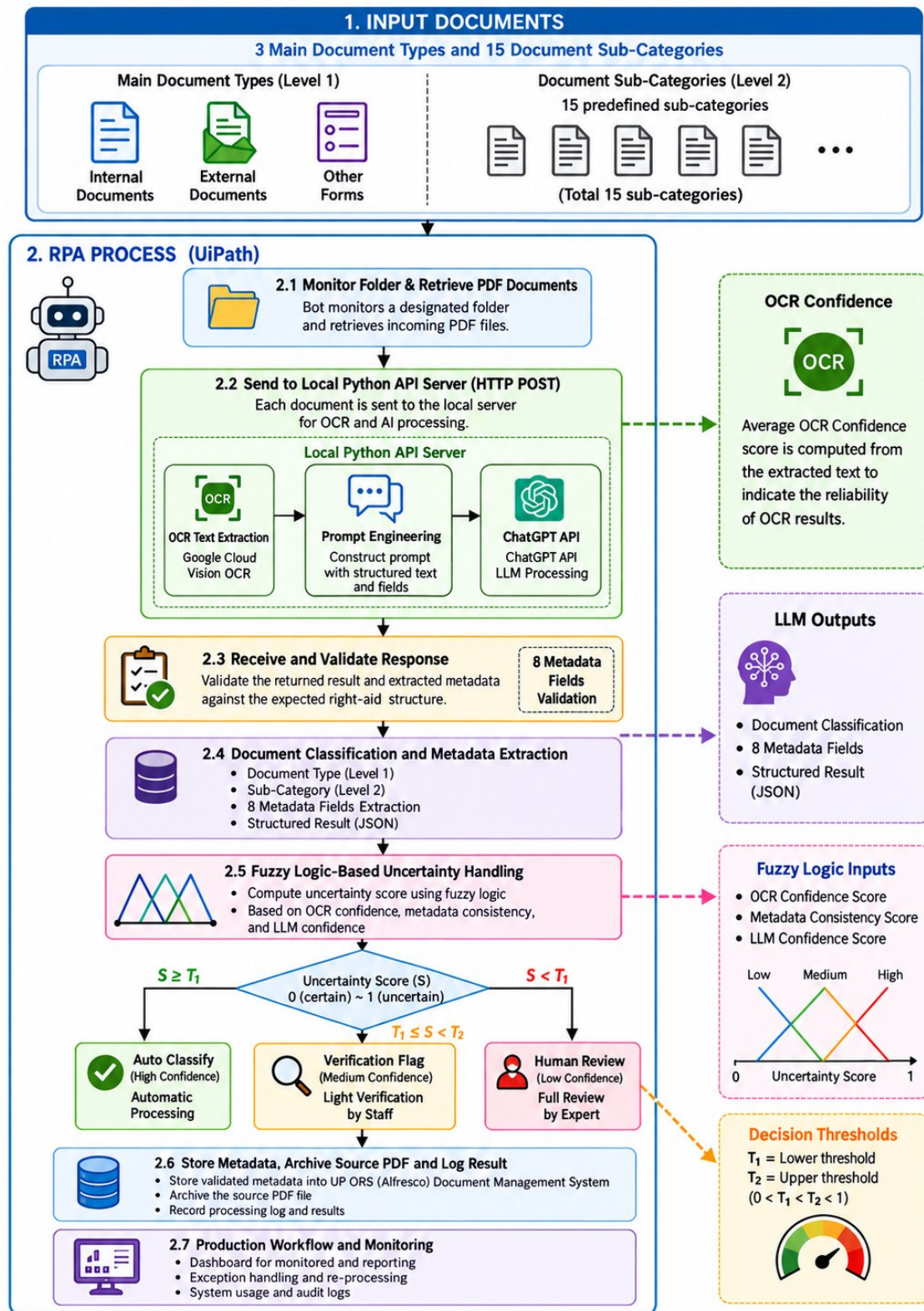


Figure 1. Proposed RPA-ChatGPT document classification framework

3.2.1. Design and implementation of the robotic process automation workflow

The automation is built in UiPath Studio (version 2026.0.198 STS) to execute rule-based steps. A software bot starts by retrieving documents from a designated folder and preparing them for subsequent stages. This process runs entirely without human involvement.

3.2.2. Optical character recognition text extraction

The system reads text with Google Cloud Vision OCR through the Get OCR text activity in UiPath, working on the first page only—the page where the features that drive classification, such as headers and addressees, actually sit. Since the UP-DMS documents are digitally generated PDFs rather than scanned physical copies, the ground truth was pulled straight from the portable document format (PDF) text layer with a parsing library. Taking the reference from the digital source keeps it high-fidelity, removes any human transcription error, and supports consistent ground-truth construction for the OCR pilot evaluation [7], [8].

For a fair, standardized benchmark, every document was rendered at 300 DPI and sent to Google Cloud Vision, PaddleOCR, and EasyOCR with no extra image-level preprocessing such as manual noise filtering or skew correction. The consistently high typographic quality of the digital source justifies this, as reflected in an average OCR confidence score (OCS) of 0.922 over the 105 documents. Performance was measured with the character error rate (CER)—the substitutions, deletions, and insertions relative to the PDF-derived ground truth [6], [7]. The OCR accuracy was calculated using (1). Google Cloud Vision came out ahead with the lowest CER (0.110), beating PaddleOCR (0.123) and EasyOCR (0.175), which is why it was adopted for the main pipeline in its default configuration [25].

$$OCR\ Accuracy\ (\%) = \left(1 - \frac{S+D+I}{N}\right) \times 100 \quad (1)$$

Where S denotes the number of substituted characters, D denotes the number of deleted characters, I denotes the number of inserted characters, and N represents the total number of characters in the ground-truth text.

3.2.3. Prompt engineering and integration with ChatGPT API

The extracted OCR text is dropped into a structured Thai-language instruction template that steers the model's reasoning. The strategy here is zero-shot: the full instruction set lives in the user message rather than in a separate system prompt. The template fixes a response layout of eight ordered fields—(1) subject, (2) sender organization, (3) document number (Arabic numerals only), (4) document date, (5) recipient, (6) executive-style summary, (7) main document type, and (8) functional sub-category. All experiments used OpenAI's GPT-4o model accessed through the ChatGPT API. To keep classification consistent, fields 7 and 8 are given explicit label sets. For the document type, the model must pick "internal document" (signaled by the term 'memo'), "forms," or "external document." For the sub-category, it must choose from a fixed list of 15 labels taken from the University's workflow records. The full template reads:

"[OCR-extracted text] Please respond in Thai only. Do not use English. Respond using the numbered fields 1-8 provided below and do not alter the structure.

1) Subject: ...; 2) Sender: ...; 3) Document Type: [Internal | Form | External]; 4) Document Category: [select from 15 predefined labels]. Respond to all items. Do not change the format."

Reproducibility and reliability are enforced by setting the temperature to zero. This buys three things: structural consistency, so every response fills the 8-field template with nothing missing; label adherence, so no out-of-vocabulary category names slip in; and valid consistency checking. Because the system runs a ChatGPT consistency check (CCC) that relies on two inference calls, a temperature of zero is what guarantees that any difference between outputs reflects real model uncertainty rather than random sampling.

3.2.4. ChatGPT: classify document type and summarize key information

ChatGPT reads the document, distills its core information, and sorts it into a predefined category—internal memo, external letter, or form template. This takes the manual reading and sorting off the table. This speeds the whole process up.

3.2.5. Fuzzy logic-based classification for document uncertainty

The fuzzy logic layer exists to handle classification uncertainty, and it does so by combining three signals: the OCS, the keyword match score (KMS), and the CCC [14], [15]. For the sake of transparency and reproducibility, the membership functions and rule sets are given in full. Each input is mapped to linguistic levels (low, medium, and high) with a mix of trapezoidal and triangular functions, listed in Table 1. The

inference engine runs a Mamdani-style rule base [26] of nine rules that fix the final fuzzy confidence score (FCS), shown in Table 2. The FCS (on a 0–100 scale) then drives a three-tier routing decision: documents at or above 70.79 (the 30th percentile of the observed FCS distribution) are processed automatically, and anything under is sent for human verification or full manual review, so errors are contained through uncertainty-aware delegation [14], [19].

Table 1. Membership function parameters for fuzzy inference

Variable	Level	Parameters [a, b, c, d]	Type
OCS/KMS	Low/medium/high	[0,0,0.2,0.4]/[0.2,0.5,0.7]/[0.5,0.7,1,1]	Trap/tri/trap
CCC	Low/high	[0,0,0.3,0.5]/[0.5,0.7,1,1]	Trap/trap
Output (FCS)	Low/medium/high	[0,0,25,45]/[30,50,70]/[55,75,100,100]	Trap/tri/trap

Table 2. Fuzzy rule base for routing decision

Rule	OCS	KMS	CCC	Output (FCS)
R1	High	High	High	High
R2–R5	Medium/high	Medium/high	High	Medium/high
R6–R9	Low/medium	Low	Low	Low

3.2.6. Data entry and storage

Once the response has passed validation, the summarized and categorized output is handed back to the RPA bot for the final stage of the pipeline. The bot signs in to the UP-DMS web interface and writes the eight extracted fields—subject, sender organization, document number, document date, recipient, executive-style summary, main document type, and functional sub-category—directly into the corresponding form fields, so that no manual keying is required at any point. The source PDF is then archived and linked to the newly created record, and a processing log containing the OCR confidence score, the fuzzy confidence score, and the routing decision taken for that document is written to the audit trail. Together these steps close the automation loop, in which a document moves from folder retrieval to a stored, fully indexed DMS record without human intervention, as shown in Figure 2. Figure 2(a) presents a source document retrieved by the RPA bot, and Figure 2(b) shows extracted metadata automatically entered into UP-DMS.

3.3. System testing and evaluation

3.3.1. Evaluation of processing speed performance

To gauge processing speed, human performance was compared directly with the RPA-based model. The point was to see how much time automation actually saves. This meant measuring the average time to process one document by hand versus with the RPA system, and then computing the improvement with the formula, as in (2).

$$\text{Processing Speed Improvement (\%)} = \frac{T_{\text{human}} - T_{\text{robot}}}{T_{\text{human}}} \times 100 \quad (2)$$

Where T_{human} is the average time taken by a human to process one document, and T_{robot} is the average time taken by the RPA bot to process one document. In (2), this helps quantify the time efficiency gained through automation.

3.3.2. Classification accuracy

Working from a stratified sample of 105 documents (35 internal, 35 external, and 35 forms), the model's output was checked against the manually labeled categories. Accuracy follows the formula, as in (3).

$$\text{Accuracy} = \frac{\text{Number of Correct Classifications}}{\text{Total Number of Documents}} \times 100 \quad (3)$$

This measure is a standard one for classification tasks in ML and NLP, where it is used to judge how well automated systems perform.

3.4. User satisfaction evaluation

To judge how usable and effective the system felt in practice, a structured questionnaire was run. It covered four areas—ease of use, perceived usefulness, accuracy of results, and overall satisfaction. Each statement was rated on a five-point Likert scale [27] from 1 (strongly disagree) to 5 (strongly agree), a scale long used in social-science research to measure attitudes and opinions.



บันทึกข้อความ

กองการเจ้าหน้าที่ มหาวิทยาลัยพะเยา

วันที่ ๗5

ที่ - 9 ม.ค. 2567

เวลา ๑๔.๕๕ น.

หน่วยงาน คณะวิทยาศาสตร์การแพทย์ โทร. 3811-3812

ที่ อว 7334/3564 วันที่ 27 ธันวาคม 2566

เรื่อง ขี้แจงการขอรับรองจริยธรรมการวิจัย

เรียน เลขาธิการคณะกรรมการพิจารณาตำแหน่งทางวิชาการ

ตามที่ ข้าพเจ้า อาจารย์ประจำหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา โภชนาการและการกำหนดอาหาร คณะวิทยาศาสตร์การแพทย์ ได้เสนอผลงานทางวิชาการเพื่อขอกำหนด ตำแหน่งทางวิชาการ ตำแหน่ง ผู้ช่วยศาสตราจารย์ ในสาขาวิชา โภชนศาสตร์ (2191) โดยได้ยื่นผลงานทาง วิชาการ ประเภทงานวิจัยทั้งหมด 2 ผลงาน ซึ่งมีผลงานที่ไม่ได้ขอรับรองจริยธรรมการวิจัย จำนวน 1 ผลงาน ข้าพเจ้าจึงขอชี้แจงผลงานวิจัยที่ไม่ได้ขอรับรองจริยธรรมการวิจัย ดังนี้

1. ผลงานวิจัย เรื่อง "A comparative study regading traditional cooking processes in Northern Thailand influence phytochemical content, antioxidant capacity and inhibition of key enzyme activity in glutinous rice."

คำชี้แจง : งานวิจัยนี้เป็นการศึกษาวิธีการเตรียมและวิธีการปรุงสุกข้าวเหนียวด้วยกรรมวิธีที่ แตกต่างกัน เพื่อเก็บรักษาและกระตุ้นสารสำคัญที่ออกฤทธิ์ทางชีวภาพในการยับยั้งการทำงานของ เอนไซม์ที่เกี่ยวข้องกับการย่อยสลายอาหารประเภทคาร์โบไฮเดรต ได้แก่ แอลฟา-อะไมเลส และ แอลฟา-กลูโคซิเดส ซึ่งการทดลองดังกล่าวเป็นการทดลองที่ทำในหลอดทดลอง (in vitro study) จึงไม่มีการขอจริยธรรมในสัตว์หรือในมนุษย์

จึงเรียนมาเพื่อโปรดพิจารณา


 (นายณัฐพล ทั่นสุวรรณ)
 อาจารย์

(a)

โมเดล RPA-CHATGPT

แบบฟอร์มบันทึกข้อมูลอัตโนมัติ

โมเดล RPA-ChatGPT สำหรับการวิเคราะห์และจำแนกเอกสารราชการอัตโนมัติกรณีศึกษา มหาวิทยาลัยพะเยา

รับโมเดล RPA-ChatGPT

EXAMPLE

ชื่อเรื่อง

ขี้แจงการขอรับรองจริยธรรมการวิจัย

ผู้ส่ง เลขที่หนังสือ

คณะวิทยาศาสตร์การแพทย์ 43

วันที่หนังสือ

27 ธันวาคม 2566

เรียน

เลขาธิการคณะกรรมการพิจารณาตำแหน่งทางวิชาการ

สรุปใจความหนังสือ

ยูเบกเกอร์ rice. เพื่อยางวาง ณ วยชนบทการศารวกรเพรอมเสวกรณรุ่งสุกข้าวเหนียวผยกรวมวบทเพรคพจน เพื่อเก็บรักษาและกระตุ้นสารสำคัญที่ออกฤทธิ์ทางชีวภาพในการยับยั้งการทำงานของเอนไซม์ที่เกี่ยวข้องกับการย่อยสลายอาหาร ประเภทคาร์โบไฮเดรต โดยมีการทดลองในหลอดทดลอง (in vitro study) และไม่มีการขอจริยธรรมในสัตว์หรือมนุษย์

การจำแนกเอกสาร หมวดหมู่เอกสาร

หนังสือชี้แจง หนังสือภายใน

บันทึก

(b)

Figure 2. Data entry and storage: (a) source document retrieved by the RPA bot and (b) extracted metadata automatically entered into UP-DMS

4. RESULTS AND DISCUSSION

This section reports the experiments and the performance of the proposed RPA-ChatGPT system. Earlier work has shown that folding ChatGPT into automation pipelines makes systems more adaptable and easier to maintain [12]. The results that follow build on that idea and test how well large language models handle document processing and classification in a real administrative setting.

4.1. Optical character recognition accuracy evaluation

To assess the OCR component, three engines were compared: Google Vision OCR, PaddleOCR, and EasyOCR—a mix of cloud-based and deep-learning approaches that turn up in practical document workflows [24]. The test used 20 representative Thai government documents, with CER, OCR accuracy, and average processing time as the metrics. Table 3 gives the results. CER was chosen because character-level errors matter especially in Thai, where word boundaries are not always marked; character-level evaluation is the usual way to judge OCR under noisy, multilingual, or otherwise complex conditions [6]–[8]. As Table 3 shows, over the 20 test documents Google Vision OCR came out on top, with the lowest CER (0.110), the highest average accuracy (89.00%), and the fastest processing time (3.80 seconds per document).

Table 3. OCR performance comparison of three OCR engines

OCR engine	Model type	Language	Avg. CER	Avg. Accuracy (%)	Avg. Time (sec)
Google Vision OCR	Cloud OCR	Thai	0.110	89.00	3.80
PaddleOCR	Deep learning (CNN/CRNN-based)	Thai	0.123	87.70	12.31
EasyOCR	Deep learning (CRAFT + CRNN-based)	Thai/Thai-Eng	0.175	61.25	17.72

4.2. Qualitative analysis of model responses

To probe the model's semantic understanding and structural reliability, its generated responses were examined qualitatively, with a focus on how it turns noisy OCR text into structured administrative data. Even with minor character-level errors in the raw Thai OCR, the model reasoned well enough from context to normalize entity names and hold a consistent format. Table 4 shows its output for an "approval request letter," where it extracts the key administrative entities and keeps to the prescribed 8-field template. As Table 4 shows, the model pulled the core content into a coherent summary (Field 6) and correctly picked the document category (Field 8) from the predefined list. Across all 105 documents, it kept 100% structural adherence, confirming that deterministic prompting (temperature = 0) is a dependable way to produce machine-readable data for downstream RPA work.

Table 4. Example of ChatGPT-generated response for an administrative document

Number of fields	Field	Model output (translated from Thai)
1	Subject	Request for approval of the digital skills development project for support staff.
2	Sender	School of Information and Communication Technology, University of Phayao.
3	Document no.	0001/2567.
4	Date	March 19, 2024.
5	Recipient	President of the University of Phayao.
6	Summary	The School of ICT requests approval to conduct a digital skills development project for 40 support staff with a budget of 85,000 THB for the 2024 fiscal year.
7	Doc type	External document.
8	Doc category	Approval request letter.

4.3. Fuzzy logic-based classification evaluation

The fuzzy confidence layer aggregates three signals—OCR confidence, keyword-matching score, and ChatGPT classification consistency—into a single FCS that estimates how reliable each automatic classification is. Fuzzy scoring is well suited to this task because it permits partial membership and graded confidence rather than a hard accept-or-reject decision, a principle long applied in uncertainty handling and recognition tasks [26], [28]. To evaluate the layer, the FCS was computed for all 105 documents and used to route each document either to automatic processing or to human review according to a confidence threshold. Because the threshold governs a trade-off between coverage—the proportion of documents processed automatically—and accuracy within the automated zone, three operating points were examined, as summarized in Table 5.

Table 5. Fuzzy confidence routing: coverage–accuracy trade-off (n = 105)

Operating point	FCS threshold	Auto-processed (n)	Coverage (%)	Auto-zone accuracy (%)	Human review (n)
Conservative	77.50 (50th pct)	53	50.5	96.2	52
Default	70.79 (30th pct)	73	69.5	97.3	32
Aggressive	58.50 (10th pct)	93	88.6	91.4	12

At the default operating point ($\text{FCS} \geq 70.79$, the 30th percentile of the FCS distribution), 69.5% of documents were processed automatically at 97.3% accuracy, while the remaining 30.5% were routed to human review; the combined accuracy including human-reviewed documents was 96.2%. Lowering the threshold increased coverage at the cost of auto-zone accuracy (88.6% coverage at 91.4% accuracy), whereas raising it yielded near-perfect automation over a smaller set of documents. These results confirm that the fuzzy layer provides an effective and tunable mechanism for delegating uncertain documents to human oversight. The routing analysis was calibrated on the three-run majority-vote prediction set, for which the overall accuracy is 83.81%.

4.4. Performance analysis: processing speed efficiency

The first part of the analysis compares the RPA-ChatGPT model with human handling on processing speed. The key figures are the mean processing time and its standard deviation (SD), which together show how consistent the performance is. This mirrors earlier RPA studies that measure automation benefits through time reduction and process efficiency [3], [5], [17], [18]. Table 6 summarizes the comparison.

Table 6. Comparison of processing efficiency between humans and robots

Type	Sample size	Mean (seconds)	SD (seconds)	Improvement (%)
Human	10	270.00	7.06	-
Robot	10	22.75	1.47	91.57

The performance differential shown in Table 6 is substantial. A human needed on average 270.00 seconds per document, while the automated workflow needed just 22.75 seconds—a 91.57% cut. That confirms the efficiency of the pipeline, and it's fit for everyday administrative document handling. The difference between manual and automated processing time was statistically significant by Welch's t-test ($t = 108.5$, $df \approx 9.8$, $p < 0.0001$), confirming that the reduction is not attributable to sampling variation; human times ranged from 258 to 281 s (mean 270.00 s, SD 7.06 s).

To see whether that advantage held across document types or varied with them, the automated per-document time was examined over all 105 documents (35 per class), as Figure 3 reports. The mean end-to-end time was almost the same across the three types—internal 22.64 s, external 23.03 s, and form 22.56 s—so the pipeline is stable with respect to category. The spread, however, was not: internal letters were the most consistent (SD 1.66 s), external letters sat in between (SD 2.06 s), and forms varied most (SD 5.60 s). Forms are bimodal, which explains it—short leave-request forms went through in about 16–18 seconds, while multi-field job-application forms took roughly 32–35 seconds because of the larger text volume and the greater number of fields. The overall mean of 22.74 seconds lines up with the aggregate robot time in Table 6.

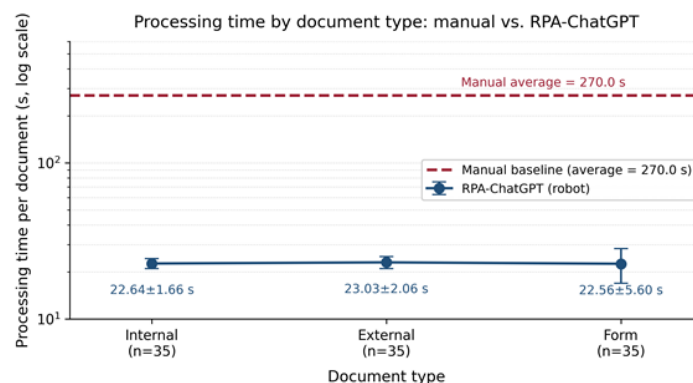


Figure 3. Processing time per document by type: manual baseline versus RPA-ChatGPT

Notably, the end-to-end processing time is dominated by the RPA orchestration and OCR stages rather than the language-model inference step. To isolate the classification step, the per-document ChatGPT API response time was measured for each main class on the held-out test set ($n = 31$): external documents averaged 0.93 s (SD = 0.26 s), internal documents 0.77 s (SD = 0.05 s), and forms 1.37 s (SD = 1.84 s). The form class's large variance traced back to a single 6.61 s outlier caused by transient network latency rather than the document itself; drop that case and forms averaged 0.79 s (SD = 0.07 s), essentially the same as the others. In short, the classification step is fast and steady across document types, and the 91.57% improvement over manual handling holds uniformly for external documents, internal documents, and forms.

4.5. Performance analysis: accuracy in document classification

The proposed model's classification was assessed at two levels of detail: the three main document types (external letters, internal letters, and forms) and the fifteen fine-grained sub-categories. Since the ChatGPT-based classifier is zero-shot and needs no task-specific training, it was run on the full set of 105 documents, each treated as unseen. For the benchmark against the fine-tuned baselines in subsection 4.7, which do need training data, a stratified 74/31 train-test split was used as well, so every model was judged on the same 31-document test set. All runs used a deterministic setup (temperature = 0), and the ChatGPT figures were checked for stability across three independent runs.

At the main-category level, the model got 100 of the 105 documents right—an overall accuracy of 95.24%, a macro-averaged F1-score of 0.952, and a 95% confidence interval of [89.3%, 97.9%] from the Wilson score interval [29]. At the sub-category level, it reached 82.86% (87 of 105 documents) with a macro-averaged F1-score of 0.839. Accuracy dropped as the task went from three classes to fifteen, which is what one would expect: the finer the subtypes, the harder they are to separate. Table 7 gives the per-class figures for the three main categories, and Figure 4 shows the confusion matrix.

Table 7. Per-class classification performance for the three main document categories ($n = 105$)

Category	Precision	Recall	F1-score	Recall 95% CI
External letter	0.972	1.000	0.986	[0.901, 1.000]
Internal letter	0.895	0.971	0.932	[0.855, 0.995]
Form	1.000	0.886	0.939	[0.740, 0.955]
Macro average	0.956	0.952	0.952	—

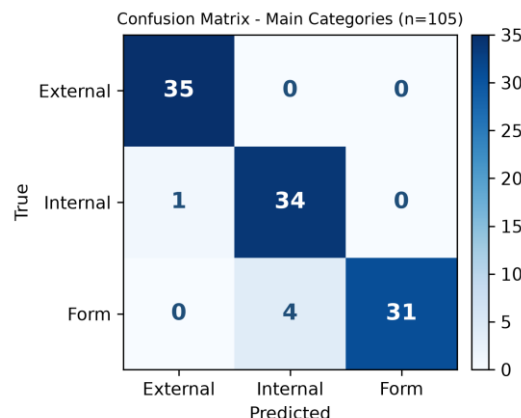


Figure 4. Confusion matrix heatmap of the three main document categories ($n = 105$)

External letters were read with perfect recall (1.000), a reflection of how standardized official outgoing correspondence is—something the OCR and prompt-based classifier handle very consistently. Forms reached perfect precision (1.000): nothing from another category was ever mislabeled as a form. Their slightly lower recall (0.886) came from four form documents that were tagged as internal letters. Internal letters had the highest recall of the more variable classes (0.971) but the lowest precision (0.895), since those four forms plus one external letter landed in this category. Figure 4 lays out the pattern.

As Figure 4 shows, only five documents were misclassified, and each error sat on an adjacent category boundary—one internal-to-external and four form-to-internal—with no confusion at all

between the most dissimilar external-letter and form classes. This indicates that the remaining errors arise from genuinely ambiguous documents, such as structured forms written in memorandum style, rather than any systematic bias in the classifier. At the sub-category level, the model reached an overall accuracy of 82.86% (87 of 105 documents), with macro-averaged precision, recall, and F1-score of 0.866, 0.860, and 0.839. The full per-sub-category breakdown is in Table 8, and Figure 5 shows how accuracy was distributed.

Table 8. Precision, recall, and F1-score by document sub-category (n = 105)

Main	Sub-category	Precision	Recall	F1	n	Recall 95% CI
External	Information request to external org.	0.727	1.000	0.842	8	[0.68, 1.00]
External	Event/activity invitation letter†	1.000	0.909	0.952	11	[0.62, 0.98]
External	Meeting invitation letter†	1.000	1.000	1.000	6	[0.61, 1.00]
External	Result notification letter	1.000	1.000	1.000	7	[0.65, 1.00]
External	General support request letter†	0.875	0.583	0.700	12	[0.32, 0.81]
Internal	Explanation letter	1.000	1.000	1.000	4	[0.51, 1.00]
Internal	Signature request letter	1.000	1.000	1.000	5	[0.57, 1.00]
Internal	Report request letter	0.857	1.000	0.923	6	[0.61, 1.00]
Internal	Approval request letter	0.400	0.500	0.444	4	[0.15, 0.85]
Internal	Schedule change notification	0.800	1.000	0.889	4	[0.51, 1.00]
Internal	Meeting postponement request letter	1.000	1.000	1.000	3	[0.44, 1.00]
Form	Approval request form	0.800	0.615	0.696	13	[0.36, 0.82]
Form	General petition form	1.000	0.286	0.444	7	[0.08, 0.64]
Form	Job application form	1.000	1.000	1.000	7	[0.65, 1.00]
Form	Leave request form	0.533	1.000	0.696	8	[0.68, 1.00]
Overall	Macro average	0.866	0.860	0.839	105	—

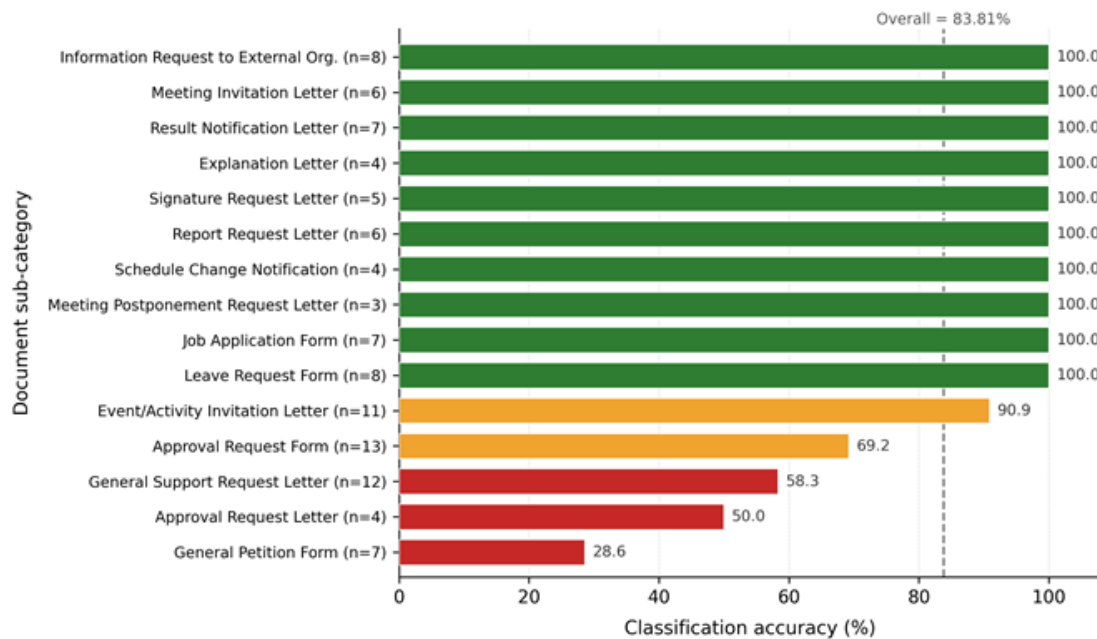


Figure 5. Classification accuracy by document sub-category (n = 105)

4.6. Bias and limitations analysis

A few biases and limitations are worth spelling out. First, the dataset is Thai-only and tied to one institution, which may limit how far the findings carry to other administrative settings or languages. Second, the 15 sub-categories cover the high-frequency workflows but not the full taxonomy of government documentation. Third, because ChatGPT is proprietary, how much formal Thai administrative language it saw during training is unknown, so it may lean on surface-level semantic cues rather than deep procedural knowledge. The framework offsets these with an uncertainty-aware fuzzy routing layer [14]. Documents that are linguistically ambiguous or fall outside the model's high-confidence zone are flagged for human review, which keeps institutional auditability and error control intact [21], [22].

4.7. Artificial intelligence model benchmark for document classification

To dig deeper into sub-category performance, macro-average precision, recall, and F1-score were used alongside accuracy, giving a fuller picture across all 15 sub-categories. For a stronger comparison, the proposed RPA-ChatGPT model was set against transformer-based NLP classifiers—cross-lingual language model-robustly optimized bidirectional encoder representations from transformers approach (XLM-RoBERTa), a multilingual model for cross-lingual representation learning, and WangchanBERTa, a transformer pretrained specifically for Thai [30], [31]. All models were tested on the same 105 documents across 15 sub-categories. Table 9 collects the results.

Table 9. Benchmark comparison of AI models for document classification

Model	Training docs	Test docs	Accuracy (%)	Macro F1
ChatGPT (proposed, zero-shot)	None	31	82.80 ± 1.52	0.793 ± 0.008
Random forest (TF-IDF)	74	31	74.19	0.682
SVM (TF-IDF)	74	31	51.61	0.378
WangchanBERTa (fine-tuned)	74	31	58.10	0.506 ± 0.117
XLM-RoBERTa (fine-tuned)	74	31	39.80	0.298 ± 0.164
Keyword-based rules	None	31	32.26	0.266

As Table 9 shows, the zero-shot ChatGPT classifier led on 15-class accuracy ($82.80\% \pm 1.52\%$) and macro-average F1 (0.793 ± 0.008), ahead of random forest (74.19%, macro-F1 = 0.682), support vector machine (SVM) (51.61%, macro-F1 = 0.378), and a keyword-based rule baseline (32.26%, macro-F1 = 0.266)—and it did so without a single labeled training example. A McNemar test between ChatGPT and random forest did not reach significance ($b = 7$, $c = 4$, and $p = 0.549$), which is unsurprising on so small a test set, but the difference points consistently in the same direction and backs the practical edge of the proposed approach. The fine-tuned transformers performed less well at this fine granularity (WangchanBERTa: 58.10%, macro-F1 = 0.506; XLM-RoBERTa: 39.80%, macro-F1 = 0.298), a sign that roughly five training documents per class is simply too few for transformer fine-tuning. The proposed system's full-corpus result (82.86% over all 105 documents, subsection 4.5) matches its held-out test score here. These results indicate that the system's principal contribution is not only its fine-grained accuracy but its zero-shot operation—no per-class labeled training—which matters in government settings where labeled documents are scarce and new sub-categories keep appearing. This comparison should therefore be read as evidence of data efficiency rather than architectural superiority: under the severe data scarcity of this setting (approximately five labeled documents per class), the zero-shot approach is the more practical choice, and fine-tuned transformers would likely match or surpass it given sufficient labeled data.

4.8. Assessment of satisfaction

Participants were chosen by purposive sampling for their direct experience with the UP-DMS. Twenty-four administrative staff from the Central Division of the University of Phayao took part, working through a structured questionnaire on a five-point Likert scale across three dimensions: usability, system performance, and practical impact. Ratings were high on all three dimensions. Usability scored $\bar{x} = 4.49$ ($SD = 0.53$), with ease of use and everyday convenience rated highest ($\bar{x} = 4.58$ and 4.54). System performance also scored $\bar{x} = 4.49$ ($SD = 0.54$), where users singled out timeliness and processing speed. Practical impact came out on top at $\bar{x} = 4.60$ ($SD = 0.52$), reflecting the time saved and the gain in work efficiency.

The questionnaire's internal consistency was checked with Cronbach's alpha, computed from the item-level answers of all 24 participants. The full 10-item instrument was excellent ($\alpha = 0.951$), and the dimension-level values held up too: $\alpha = 0.901$ for usability (4 items), $\alpha = 0.826$ for system performance (3 items), and $\alpha = 0.902$ for practical impact (3 items), all well past the usual 0.70 threshold. The score distribution tells the same story: of the 240 individual item ratings, 57.5% were 5, 37.1% were 4, and 5.4% were 3, with an item SD of 0.53–0.72. No rating fell under 3, and the lowest item mean was 4.33—consistently positive, without extreme spread. Alongside the numbers, users offered qualitative feedback during the evaluation. They noted that the system cut down on repetitive data entry and made document routing simpler.

4.9. Discussion

The proposed RPA-ChatGPT framework sharply improved processing efficiency without giving up classification quality. Against manual handling, it cut the average processing time by 91.57% (22.75 s versus 270.00 s per document) and reached an overall classification accuracy above 85%, with several categories clearing 92%. This echoes earlier findings that RPA lightens repetitive

administrative loads and that AI-based document understanding strengthens workflow automation and decision support [7], [15], [16].

It helps to read these results against earlier OCR-RPA work in government and finance. Most of that work centered on structured or semi-structured documents—invoices, financial forms, standardized records—and typically reported processing-time cuts of roughly 80–90% for fixed-layout material [3], [5], [17], [18]. This study pushes into harder territory: free-form Thai administrative correspondence, which is far more demanding to understand. Thai has no spaces between words, stacks diacritics and tone marks that trip up OCR, and relies on highly standardized official phrasing that recurs across different document types.

Two mechanisms account for the gains. First, OCR pulls out the text but cannot read intent, so keyword-based classifiers stumble when the same administrative phrase—"request for approval," say—shows up across several categories. ChatGPT instead reasons over context: who is writing to whom, where the key phrases sit, and what surrounds them. That contextual read is what puts it ahead of keyword-feature classifiers like SVM and random forest on the fine-grained 15-category task (Table 9). Second, fuzzy logic cushions OCR uncertainty. Rather than fixing OCR errors, it weighs OCR confidence against keyword consistency and other signals to judge how trustworthy an automatic classification is; low-confidence documents are routed to a human instead of being classified on their own, which lifts accuracy inside the automated zone to 97.3%, against 83.81% across the whole dataset (this routing analysis was calibrated on the three-run majority-vote prediction set).

Beyond the technical results, the framework has practical implications for digital government. Its modular design means it can sit alongside existing DMS through standard web interfaces, with no need to replace legacy infrastructure. Privacy does need attention, since the current setup sends OCR-extracted text to an external commercial language model; for broader public-sector use, PII should be anonymized before any external API call, in line with Thailand's Personal Data Protection Act (PDPA) [32]. And because the fuzzy routing records confidence scores, decision variables, and routing outcomes for every document, it builds in the kind of auditability that transparent, accountable automation depends on.

These characteristics align the framework with Thailand's national digital government agenda. Under the digital government development plan (2023–2027), government agencies are directed toward interoperability, standardized data exchange, and citizen-centric service delivery. The proposed architecture supports these objectives directly: it produces structured metadata compatible with national document standards, and requires only lightweight customization—prompt template and label set adaptation—to extend across agencies and ministry levels. For national-scale deployment, three governance prerequisites would need to be established: i) standardized prompt templates aligned with the government correspondence regulation [13], enabling consistent classification across agencies; ii) inter-agency API governance agreements to manage OpenAI access, cost allocation, and model version control; and iii) centralized fuzzy threshold calibration to account for document-quality variation across regional and central government offices. Agencies with strict data sovereignty requirements could additionally adopt on-premises Thai-language models such as Typhoon, eliminating dependency on external cloud services while preserving the framework's modular architecture.

The framework is also easier to adopt for organizations short on technical resources. Where transformer-based approaches demand graphics processing unit (GPU) infrastructure, large labeled datasets, and machine-learning specialists, this system leans on low-code RPA, cloud OCR, and pay-per-use language-model services. Staff only step in through the familiar routine of document review when an uncertain case is flagged, so little technical training is required. That accessibility shows up in the user study, where perceived practical impact drew the highest rating ($\bar{x} = 4.60$), and it may help narrow the digital-capability gap between well-resourced central agencies and smaller regional offices.

For all that, some limitations remain. Most errors clustered among categories that share administrative language and structure, especially semi-structured forms and general request documents. OCR slips from low-quality scans, handwritten notes, and missing context added to the misclassifications. The framework also reads only the OCR text from the first page and makes no explicit use of layout or multi-page context—choices that keep the implementation simple but can weaken discrimination on complex documents. And because classification is prompt-based rather than fine-tuned, the most ambiguous cases may still be out of reach.

4.10. Bias control and generalizability

The system was built for the Thai government administrative context, but its architecture is not inherently language-bound: the only language-specific pieces are the category label list in the prompt and the keyword dictionary, and both adapt to other languages without touching the core RPA workflow or the fuzzy inference engine. The 15 sub-categories reflect the high-frequency routine documents set out in the Thai government correspondence regulation, and adding a new category takes only a prompt update, not

retraining. GPT-4o also aligned well with the Thai administrative register; an error analysis of the 18 misclassified cases showed that the failures came from ambiguous category boundaries rather than any gap in linguistic understanding.

4.11. Future research directions

Four directions stand out for future work. The first is scale: extending the evaluation to larger, multi-institutional corpora that cover the full taxonomy of Thai government correspondence. The second is language: testing the zero-shot approach beyond Thai, especially in English and neighboring ASEAN languages. The third is adaptivity—using reinforcement learning and human-in-the-loop feedback to tune routing thresholds and fuzzy rule weights from verification outcomes, while stronger OCR (deep-learning and handwriting-aware) and knowledge-graph integration of the extracted metadata push the framework toward semantic search and decision support. The fourth is time: longitudinal studies over 12–24 months to see whether the efficiency gains last at scale and whether accuracy drifts as document conventions and upstream models change.

5. CONCLUSION

This study set out an intelligent document automation system that unites RPA, OCR, and the ChatGPT API for government document processing. In testing, it cut average handling time by 91.57% against manual work and reached a sub-category classification accuracy of 82.86% (95.24% at the main-class level) without any labeled training data. The user study backed up its practical value, with strong ratings for usability, system performance, and practical impact. The novelty is in the hybrid design: by weaving together RPA, OCR, ChatGPT, and fuzzy-logic uncertainty handling, the system delivers not just automation but semantic interpretation, confidence-aware classification, and workflow-level decision support—a practical, scalable route to modernizing document work in public and academic institutions.

ACKNOWLEDGMENTS

The authors would like to thank the administrative staff of the University of Phayao for their cooperation and feedback during system testing.

FUNDING INFORMATION

This research received no external funding.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Adisorn Phonsupharak	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Rattanawadee	✓			✓	✓					✓		✓	✓	
Panthong														

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**ding

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

INFORMED CONSENT

Informed consent was obtained from all participants involved in the study.

ETHICAL APPROVAL

This study was approved by the Human Research Ethics Committee of the University of Phayao (Approval No. HREC-UP-HSST 1.1/014/68). All procedures involving human participants were conducted in accordance with the approved ethical standards.

DATA AVAILABILITY

The data used in this study are not publicly available due to confidentiality and privacy restrictions related to administrative documents. Aggregated or anonymized data may be made available from the corresponding author, [RP], upon reasonable request.

REFERENCES

- [1] R. Mehta and R. Chaher, "Implementation of robotic process automation (RPA) in digital marketing," in *2022 3rd International Conference for Emerging Technology (INCET)*, May 2022, pp. 1–4, doi: 10.1109/INCET54531.2022.9824263.
- [2] Y. Ratchatawetchakul, B. Bansri, K. Ratchatawetchakul, N. Satchawatee, W. Ratchatawetchakul, and I. Eampoonga, "Integrating robotic process automation (RPA) with hybrid cloud enterprise resource planning (ERP)," in *2024 5th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, Jun. 2024, pp. 1–4, doi: 10.1109/TIMES-iCON61890.2024.10630776.
- [3] W. M. P. van der Aalst, M. Bichler, and A. Heinzl, "Robotic process automation," *Business & Information Systems Engineering*, vol. 60, no. 4, pp. 269–272, Aug. 2018, doi: 10.1007/s12599-018-0542-4.
- [4] D. A. Nabilla M, R. W. Witjaksono, and T. F. Kusumasari, "Implementation of automated testing: order to cash business processes on SAP using Uipath software," in *2024 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, Jul. 2024, pp. 148–153, doi: 10.1109/ISITIA63062.2024.10667810.
- [5] N. K. Gupta, A. Chaudhary, R. Singh, and R. Singh, "ChatGPT: exploring the capabilities and limitations of a large language model for conversational AI," in *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICIT)*, Nov. 2023, pp. 139–142, doi: 10.1109/ICAICIT60255.2023.10465811.
- [6] C. M. G. Paredes, C. Machuca, and Y. M. S. Claudio, "ChatGPT API: brief overview and integration in software development," *International Journal of Engineering Insights*, vol. 1, no. 1, pp. 25–29, Nov. 2023, doi: 10.61961/injei.v1i1.7.
- [7] J. Ribeiro, R. Lima, T. Eckhardt, and S. Paiva, "Robotic process automation and artificial intelligence in Industry 4.0 – a literature review," *Procedia Computer Science*, vol. 181, pp. 51–58, 2021, doi: 10.1016/j.procs.2021.01.104.
- [8] R. Smith, "An overview of the Tesseract OCR engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, Sep. 2007, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.
- [9] N. Islam, Z. Islam, and N. Noor, "A survey on optical character recognition system," *Journal of Information & Communication Technology*, vol. 10, no. 2, pp. 1–4, 2016.
- [10] L. Chen *et al.*, "OCR text region detection based on Chinese character features in natural scenes," in *2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP)*, Apr. 2023, pp. 1538–1541, doi: 10.1109/ICSP58490.2023.10248635.
- [11] J. Li, "Research on English automatic recognition system based on OCR technology," in *2023 7th Asian Conference on Artificial Intelligence Technology (ACAIT)*, Nov. 2023, pp. 1061–1066, doi: 10.1109/ACAIT60137.2023.10528556.
- [12] A. Z. Kalo and M. L. Sipos, "Key-value pair searching system via Tesseract OCR and post-processing," in *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Jan. 2021, pp. 000461–000464, doi: 10.1109/SAMI50585.2021.9378680.
- [13] Office of the Prime Minister, "Rabiat Samnak Nayok Raththamontri wa duai Ngan Sarabun (Chabap thi 4), Pho. So. 2564 [Prime Minister's Office Regulation on Correspondence Work (No. 4), B.E. 2564]," *Royal Thai Government Gazette*. Thailand, 2021. [Online]. Available: https://icead.kku.ac.th/wp-content/uploads/2021/05/T_0001.pdf
- [14] B.-H. Chen and C.-C. Chen, "Invention of Line-ChatBot: an innovative application of ChatGPT API and LINE Bot for enhanced student learning," in *2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII)*, Aug. 2023, pp. 452–455, doi: 10.1109/ICKII58656.2023.10332623.
- [15] Y. Kim and R. Kouatly, "Self-healing automation testing with Selenium and ChatGPT API integration to improve the efficiency of the maintenance," in *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, Nov. 2024, pp. 339–344, doi: 10.1109/FLLM63129.2024.10852490.
- [16] N. Yadav and S. P. Panda, "A path forward for automation in robotic process automation projects: potential process selection strategies," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, May 2022, pp. 801–805, doi: 10.1109/COM-IT-CON54601.2022.9850739.
- [17] R. Gomathi, "Automating the task of transferring data from spreadsheet to database using robotic process automation," in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, Nov. 2022, pp. 1257–1260, doi: 10.1109/ICAISS55157.2022.10010983.
- [18] C. Vijai, S. M. Suriyalakshmi, and M. Elayaraja, "The future of robotic process automation (RPA) in the banking sector for better customer experience," *Shanlax International Journal of Commerce*, vol. 8, no. 2, pp. 61–65, Apr. 2020, doi: 10.34293/commerce.v8i2.1709.
- [19] H. Valgaeren, "Robotic process automation in financial and accounting processes in the banking sector-a qualitative study with experts in the field," Master's thesis, Faculty of Economics and Business, KU Leuven, Leuven, Belgium, 2019.
- [20] K. Shi, L. Li, H. Liu, J. He, N. Zhang, and W. Song, "An improved KNN text classification algorithm based on density," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, Sep. 2011, pp. 113–117, doi: 10.1109/CCIS.2011.6045043.
- [21] P. Patri, "Robotic process automation: challenges and solutions for the banking sector," *International Journal of Management*, vol. 11, no. 12, Dec. 2020, doi: 10.34218/IJM.11.12.2020.031.
- [22] D. Holmberg and M. H.-Nilsson, "Managing robotics process automation (RPA) - a qualitative case study of the adoption and implementation process in the banking sector." Master's Thesis, Master's Programme in International Strategic Management, Department of Business Administration, Lund University, Lund, Sweden, 2020.

- [23] Y. Du *et al.*, “PP-OCR: a practical ultra lightweight OCR system,” Oct. 2020, *arXiv: 2009.09941*.
- [24] Jaied AI, “EasyOCR,” *github.com*, 2024, Accessed date: Sep. 13, 2026. [Online]. Available: <https://github.com/JaiedAI/EasyOCR>
- [25] Google Cloud, “Cloud Vision API documentation,” *docs.cloud.google.com*, 2024, Accessed date: Sep. 13, 2026. [Online]. Available: <https://docs.cloud.google.com/vision/docs>
- [26] E. H. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, Jan. 1975, doi: 10.1016/S0020-7373(75)80002-2.
- [27] R. Likert, “A technique for the measurement of attitudes,” *Archives of Psychology*, vol. 22, no. 140, pp. 1–55, 1932.
- [28] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.
- [29] E. B. Wilson, “Probable inference, the law of succession, and statistical inference,” *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, Jun. 1927, doi: 10.1080/01621459.1927.10502953.
- [30] A. Conneau *et al.*, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451, doi: 10.18653/v1/2020.acl-main.747.
- [31] L. Lowphansirikul, C. Polpanumas, N. Jantrakulchai, and S. Nutanong, “WangchanBERTa: pretraining transformer-based Thai language models,” Mar. 2021, *arXiv: 2101.09635*.
- [32] Kingdom of Thailand, *Personal Data Protection Act, B.E. 2562*. Thailand, 2019. [Online]. Available: https://data.thailand.opendevelopmentmekong.net/en/laws_record/2562/resource/ec616be5-9fbf-4071-b4b5-cb1f3e46e826

BIOGRAPHIES OF AUTHORS



Adisorn Phonsupharak    received his B.Sc. degree in Industrial Computer from Chiang Rai Rajabhat University, Thailand. He currently works as a system developer and academic officer in the Division of Central Administration at the University of Phayao. His research interests include robotic process automation (RPA), optical character recognition (OCR), artificial intelligence applications in public administration, and web-based system development. He has played a key role in the development of the University of Phayao Document Management System and has been actively involved in knowledge transfer initiatives. He can be contacted at email: adisorn.po@up.ac.th.



Rattanawadee Panthong    received the Ph.D. (Computer Science) from Kasetsart University, Thailand in 2021, with the dissertation “Hybrid feature selection based on class-dependent technique for liver cancer classification model of central region in Thailand”. She is currently an instructor at the School of Information and Communication Technology, University of Phayao, Thailand. In addition, she is serving as assistant dean in the School of Information and Communication Technology (2019–2024). Her research interests are in machine learning, data mining, data analytics, and data warehousing. She can be contacted at email: rattanawadee.pa@up.ac.th.