

Received: 9 ธ.ค. 2568

Revised: 26 ม.ค. 2569

Accepted: 28 ม.ค. 2569

ระบบถอดข้อมูลใบเสร็จโอนเงินอัตโนมัติ โดยใช้เทคโนโลยี Tesseract OCR

ร่วมกับการถอดรหัส QR Code

Automatic Bank Slip Data Extraction System

via Tesseract OCR and QR Code Decoding

สรวิชัย บุญเสริม¹, ทิพญาดา แก้วมะคำ¹, พาสน์ ปราโมกษ์ชน¹,อรรณวิทย์ ชังคมานนท์¹ และ กองกาญจน์ ดุลยไชย^{1*}¹สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้Sorawit Bunset¹, Tipyada Kaewmakam¹, Part Pramokchon¹,Attawit Changkamanon¹ and Kongkarn Dullayachai^{1*}¹Computer Science Department, Faculty of Science, Maejo University

*Corresponding author: kongkarn@maejo.mju.ac.th

Abstract

This research aims to develop and compare the efficiency of an automated bank transfer receipt data extraction system using Tesseract OCR technology combined with QR Code decoding. The system was developed using Go 1.22 and integrated with a LINE Official Account via Webhook to provide near real-time services and store data in a PostgreSQL database.

The evaluation was conducted using 102 sample receipts from three banks, with 34 samples from each. The transaction reference numbers extracted by OCR were compared with the data decoded from the QR Code using the Windowed Levenshtein Distance technique, with a similarity threshold of 70 percent. The results, categorized by bank, showed that Kasikornbank achieved 100% accuracy with an average processing time of 1.65 seconds; Krungthai Bank achieved 97.06% accuracy with an average processing time of 1.78 seconds; and Siam Commercial Bank achieved 100% accuracy with an average processing time of 1.84 seconds. In

summary, the system achieved an overall average accuracy of 99.02% and a total average processing time of 1.76 seconds (S.D. 0.50 seconds) per receipt. These findings demonstrate the system's high efficiency, making it suitable for application in automated payment verification workflows for organizations.

Keywords: Bank Transfer Receipt; Tesseract OCR; QR Code; Windowed Levenshtein Distance; LINE Official Account

บทคัดย่อ

งานวิจัยนี้มุ่งพัฒนาและประเมินประสิทธิภาพระบบถอดข้อมูลใบเสร็จโอนเงินอัตโนมัติ โดยใช้เทคโนโลยี เทสเซอร์แรค (Tesseract OCR) ร่วมกับการถอดรหัส QR Code ระบบพัฒนาด้วยภาษา Go 1.22 เชื่อมต่อกับ LINE Official Account ผ่าน Webhook เพื่อให้บริการแบบกึ่งเรียลไทม์และจัดเก็บข้อมูลลง PostgreSQL

การประเมินผลใช้กลุ่มตัวอย่างใบเสร็จ 102 ใบ จาก 3 ธนาคาร ธนาคารละ 34 ใบ โดยเปรียบเทียบรหัสอ้างอิงที่อ่านได้จาก OCR กับข้อมูลที่ถอดจาก QR Code ด้วยเทคนิค Windowed Levenshtein Distance ภายใต้เกณฑ์ความคลาดคลึงร้อยละ 70 ผลการทดสอบจำแนกรายธนาคารพบว่า ธนาคารกสิกรไทย มีความถูกต้องร้อยละ 100 เวลาประมวลผลเฉลี่ย 1.65 วินาที ธนาคารกรุงไทย มีความถูกต้องร้อยละ 97.06 เวลาประมวลผลเฉลี่ย 1.78 วินาที และ ธนาคารไทยพาณิชย์ มีความถูกต้องร้อยละ 100 เวลาประมวลผลเฉลี่ย 1.84 วินาที โดยสรุปภาพรวมระบบมีอัตราความถูกต้องเฉลี่ยร้อยละ 99.02 และเวลาในการประมวลผลเฉลี่ยรวม 1.76 วินาที (S.D. 0.50 วินาที) ต่อใบเสร็จ แสดงให้เห็นว่าระบบมีประสิทธิภาพสูง เหมาะสำหรับการประยุกต์ใช้ในระบบตรวจสอบการชำระเงินอัตโนมัติขององค์กร

คำสำคัญ: การถอดข้อมูลใบเสร็จโอนเงิน; เทสเซอร์แรค โอซีอาร์; รหัสคิวอาร์; ระยะห่างเลเวนชไตน์แบบหน้าต่าง; ปัญหาลิไน์ทางการ

1. บทนำ

ในปัจจุบัน เทคโนโลยีทางการเงิน (FinTech) และระบบธนาคารผ่านโทรศัพท์มือถือ (Mobile Banking) ได้เข้ามามีบทบาทสำคัญในการขับเคลื่อนระบบเศรษฐกิจดิจิทัลของประเทศไทย ส่งผลให้ปริมาณธุรกรรมทางการเงินแบบไร้เงินสดเพิ่มสูงขึ้นอย่างรวดเร็ว โดยหลักฐานสำคัญที่ใช้

ยืนยันการทำธุรกรรมคือ "ใบเสร็จโอนเงินอิเล็กทรอนิกส์" หรือ "สลิปธนาคาร" (E-Slip) ซึ่งนิยมใช้ส่งยืนยันการชำระเงินระหว่างผู้ซื้อและผู้ขาย อย่างไรก็ตาม กระบวนการตรวจสอบความถูกต้องของใบเสร็จเหล่านี้ยังคงเป็นปัญหาคอขวด (Bottleneck) สำหรับผู้ประกอบการและองค์กร เนื่องจากส่วนใหญ่ยังต้องอาศัยการตรวจสอบด้วยสายตามนุษย์ (Manual Verification) ซึ่งมีความล่าช้า มีโอกาสเกิดความผิดพลาด (Human Error) สูง และไม่สามารถตรวจสอบใบเสร็จปลอมที่มีการดัดต่อภาพได้อย่างมีประสิทธิภาพ ท่ามกลางปริมาณธุรกรรมจำนวนมากที่เกิดขึ้นในแต่ละวัน

เพื่อแก้ไขปัญหาดังกล่าว การนำเทคโนโลยี การรู้จำอักขระด้วยแสง (Optical Character Recognition: OCR) มาประยุกต์ใช้จึงเป็นแนวทางที่ได้รับความนิยม โดยเฉพาะการใช้เครื่องมือโอเพนซอร์สอย่าง เทสเซอแรค (Tesseract OCR) อย่างไรก็ตาม การประยุกต์ใช้ OCR กับใบเสร็จโอนเงินของธนาคารไทยประสบปัญหาท้าทายทางเทคนิคหลายประการ โดยเฉพาะปัญหาด้าน "สัญญาณรบกวนบนพื้นหลัง" (Background Noise) เนื่องจากใบเสร็จของธนาคารไทยมักมีการออกแบบลดลายกราฟิกที่มีสีสันทับซ้อน หรือการใช้สีพื้นหลังตามเทศกาล ซึ่งส่งผลกระทบต่อความแม่นยำ (Accuracy) ในการแยกแยะตัวอักษรของระบบ OCR ดังที่การศึกษาของ Sayallar และคณะ (2023) ได้ชี้ให้เห็นว่า แม้ Tesseract OCR จะมีความแม่นยำสูงในเอกสารทั่วไป แต่มักมีข้อจำกัดในการอ่านคำ (Recall) ที่ต่ำลงเมื่อต้องจัดการกับเอกสารที่มีสัญญาณรบกวนหรือพื้นหลังซับซ้อน นอกจากนี้ ลักษณะของภาษาไทยที่มีสระและวรรณยุกต์อยู่ในระดับบรรทัดที่ซ้อนกัน ทำให้การอ่านข้อมูลสำคัญ เช่น ชื่อผู้โอน หรือหมายเลขอ้างอิง มักเกิดความคลาดเคลื่อนเมื่อเทียบกับการอ่านเอกสารภาษาอังกฤษทั่วไป ซึ่งสอดคล้องกับงานวิจัยของ Thammarak และคณะ (2022) และ วิสรุต เหล่าดารา (2565) ที่พบว่า Tesseract OCR มักประสบปัญหาความคลาดเคลื่อนในการอ่านสระ วรรณยุกต์ และพยัญชนะไทยที่มีลักษณะคล้ายคลึงกัน หากไม่มีกระบวนการประมวลผลภาพก่อน (Pre-processing) ที่เหมาะสม

นอกเหนือจากข้อมูลตัวอักษร ใบเสร็จโอนเงินมาตรฐานในประเทศไทยยังประกอบด้วย รหัสคิวอาร์ (QR Code) ตามมาตรฐาน PromptPay (EMVCo Standard) ซึ่งภายในบรรจุข้อมูลดิจิทัลที่สำคัญสำหรับการตรวจสอบ (Payload) ได้แก่ รหัสอ้างอิงรายการ (Transaction Ref), วันที่และเวลาที่โอน, และข้อมูลผู้ส่ง-ผู้รับ ซึ่งเป็นข้อมูลที่มีความน่าเชื่อถือสูงและไม่ขึ้นกับคุณภาพของกราฟิกพื้นหลัง ดังนั้น การใช้ข้อมูลจาก QR Code เข้ามาร่วมตรวจสอบ (Cross-Verification) จึงเป็นกุญแจสำคัญในการยกระดับความถูกต้องของระบบเพื่อชดเชยข้อจำกัดของ OCR เพียงลำพัง

งานวิจัยนี้จึงมุ่งเน้นพัฒนาระบบถอดข้อมูลใบเสร็จโอนเงินอัตโนมัติ โดยบูรณาการการทำงานระหว่าง Tesseract OCR และ การถอดรหัส QR Code เพื่อเปรียบเทียบและยืนยันความถูกต้องของข้อมูลผ่านเทคนิค ระยะห่างเลเวนชไตน์แบบหน้าต่าง (Windowed Levenshtein Distance) โดยระบบถูกพัฒนาด้วย ภาษา Go (Golang) เวอร์ชัน 1.22 ซึ่งมีจุดเด่นในเรื่องการจัดการการทำงาน

แบบพร้อมกัน (Concurrency) ผ่านกลไก Goroutines ทำให้สามารถรองรับการประมวลผลรูปภาพจำนวนมากได้รวดเร็วกว่าภาษาอื่น และมีการบริหารจัดการหน่วยความจำที่มีประสิทธิภาพ ตามแนวทางที่ ณรงค์เกียรติ นามห้วยทอง และคณะ (2568) ได้นำเสนอไว้ว่าภาษา Go สามารถช่วยลดเวลาในการประมวลผลธุรกรรมและเหมาะสำหรับการทำงานในระบบจริงที่ต้องตอบสนองแบบเรียลไทม์ ซึ่งสอดคล้องกับเป้าหมายของงานวิจัยนี้ในการให้บริการในรูปแบบกึ่งเรียลไทม์ (Near Real-time) ผ่านช่องทาง บัญชีไลน์ทางการ (LINE Official Account)

ผลลัพธ์จากงานวิจัยนี้ไม่เพียงแต่ช่วยแก้ปัญหาความผิดพลาดจากการอ่านข้อมูลบนพื้นหลังที่ซับซ้อนของใบเสร็จธนาคารไทย แต่ยังก่อให้เกิดองค์ความรู้ใหม่ในการประยุกต์ใช้เทคนิคการตรวจสอบข้อมูลแบบสองทาง (Two-way Verification) ระหว่าง OCR และ QR Code ซึ่งสามารถนำไปต่อยอดเป็นระบบตรวจสอบการชำระเงินอัตโนมัติ (Automated Payment Verification) ที่มีความแม่นยำและเสถียรภาพสูง ช่วยลดภาระงานและเพิ่มความเชื่อมั่นให้กับภาคธุรกิจและองค์กรในการรองรับธุรกรรมดิจิทัลในอนาคต

2. วัตถุประสงค์

1. เพื่อพัฒนาระบบตรวจสอบใบเสร็จโอนเงินอัตโนมัติ โดยใช้ Tesseract OCR ร่วมกับ QR Code และยืนยันความถูกต้องด้วยเทคนิค Windowed Levenshtein Distance
2. เพื่อพัฒนาช่องทางใช้งานผ่าน LINE Official Account ให้รองรับการตรวจสอบและแจ้งผลแบบกึ่งเรียลไทม์ พร้อมระบบจัดเก็บข้อมูลธุรกรรม
3. เพื่อประเมินประสิทธิภาพและความแม่นยำของระบบ โดยทดสอบกับกลุ่มตัวอย่างใบเสร็จจากธนาคารกรุงไทย กรุงไทย และไทยพาณิชย์

3. งานวิจัยที่เกี่ยวข้อง

ณรงค์เกียรติ นามห้วยทอง และคณะ (2568) ศึกษาประสิทธิภาพของ Tesseract OCR สำหรับการตรวจสอบธุรกรรมทางการเงิน โดยเปรียบเทียบชุดประมวลผลภาพ 5 รูปแบบ (IPPA, IPP2, IPP6, IPP7 และ IPP12) ที่ใช้เทคนิคต่าง ๆ เช่น Grayscale, Resize, Contrast Adjustment และ Sauvola Threshold เพื่อวิเคราะห์ผลต่อความแม่นยำของการรู้จำตัวอักษร ผลการวิจัยพบว่าชุด IPP12 ซึ่งใช้เทคนิค Sauvola Threshold ให้ประสิทธิภาพสูงสุด โดยมี Accuracy 81.03%, Precision 82.14%, Recall 90.79% และ F1-Score 86.26% สะท้อนถึงประสิทธิภาพของการทำ Adaptive Binarization ในการลดความผิดพลาดจากภาพที่มีความซับซ้อนหรือแสงไม่สม่ำเสมอ นอกจากนี้ยังทดสอบระบบด้วยภาษา Go (Golang) ซึ่งรองรับการประมวลผลแบบ Concurrency และใช้หน่วยความจำอย่างมีประสิทธิภาพ พบว่าสามารถประมวลผลใบเสร็จได้เฉลี่ย 5.313 วินาทีต่อ

ภาพ เหมาะสำหรับการใช้งานจริงแบบเรียลไทม์ ผลการศึกษานี้สามารถนำมาเป็นแนวทางในการพัฒนาระบบถอดข้อมูลจากใบเสร็จการโอนเงินของธนาคารที่ใช้ OCR ร่วมกับการถอดรหัส QR Code เพื่อเพิ่มความถูกต้องและความรวดเร็วในการยืนยันธุรกรรมทางการเงินผ่านช่องทางออนไลน์ เช่น LINE Official Account ซึ่งเป็นส่วนหนึ่งของงานวิจัยฉบับนี้

Sayallar และคณะ (2023) ได้พัฒนาและเปรียบเทียบประสิทธิภาพของระบบ OCR สำหรับการอ่านข้อมูลจากภาพใบเสร็จรับเงิน โดยทดสอบ Nacsoft OCR ร่วมกับเครื่องมือที่ใช้กันอย่างแพร่หลาย ได้แก่ Tesseract OCR, EasyOCR และ Google Vision API บนชุดข้อมูลภาษาเตอร์กิกและภาษาอังกฤษ ผลการวิจัยพบว่า Tesseract OCR มีความแม่นยำ (Precision) สูง แต่มีข้อจำกัดด้านความสามารถในการค้นหา (Recall) โดยเฉพาะเมื่อประมวลผลใบเสร็จที่มีพื้นหลังซับซ้อนหรือมีสัญญาณรบกวน ส่งผลให้ประสิทธิภาพโดยรวมยังเป็นรอง Google Vision API งานวิจัยดังกล่าวสะท้อนถึงข้อจำกัดของการใช้ Tesseract OCR เพียงลำพัง และสนับสนุนแนวคิดของงานวิจัยฉบับนี้ ที่นำการถอดรหัส QR Code มาใช้เป็นกลไกตรวจสอบความถูกต้องเพื่อชดเชยข้อจำกัดดังกล่าว

วิศรุต เหล่าดารา (2565) ได้ศึกษาเปรียบเทียบประสิทธิภาพของ Tesseract OCR และ EasyOCR ในการอ่านตัวอักษรภาษาไทยจากเอกสารที่มีลักษณะเป็นภาพ พบว่า EasyOCR ให้ความแม่นยำและการระบุตำแหน่งข้อความได้ดีกว่า อย่างไรก็ตาม งานวิจัยได้เสนอให้มีการประมวลผลภาพก่อน (Image Pre-processing) เช่น การแปลงภาพเป็นขาวดำ เพื่อลดสัญญาณรบกวนและเพิ่มความคมชัดของตัวอักษร ซึ่งช่วยปรับปรุงประสิทธิภาพของ OCR ได้อย่างมีนัยสำคัญ แนวทางดังกล่าวสอดคล้องกับความจำเป็นในการจัดการพื้นหลังของใบเสร็จโอนเงินในงานวิจัยนี้

Thammarak และคณะ (2022) ได้เปรียบเทียบประสิทธิภาพระหว่าง Tesseract OCR และ Google Cloud Vision API ในการอ่านเอกสารภาษาไทยที่มีลวดลายพื้นหลังซับซ้อน ผลการศึกษพบว่า Google Cloud Vision API มีความถูกต้องสูงกว่าอย่างชัดเจน ขณะที่ Tesseract OCR มีข้อจำกัดในการอ่านสระและวรรณยุกต์ภาษาไทย อย่างไรก็ตาม การประยุกต์ใช้เทคนิคการประมวลผลภาพก่อน เช่น การปรับความคมชัดและความสว่าง สามารถช่วยเพิ่มความแม่นยำในการรู้จำตัวอักษรได้ ซึ่งเป็นแนวทางสำคัญในการปรับปรุงคุณภาพการอ่านข้อมูลจากภาพใบเสร็จโอนเงิน

4. วิธีการดำเนินการ

งานวิจัยนี้ดำเนินการตามวงจรการพัฒนาาระบบ (System Development Life Cycle: SDLC) เพื่อให้การพัฒนาเป็นไปอย่างมีขั้นตอนและมีประสิทธิภาพ ดังรายละเอียดต่อไปนี้

4.1 ระยะการวางแผนและวิเคราะห์ความต้องการ (Planning and Analysis)

4.1.1 การศึกษาเครื่องมือและเทคโนโลยี: ผู้วิจัยได้เลือกใช้เครื่องมือและเทคโนโลยีหลัก ได้แก่

1. ภาษาโก (Go Programming Language) เวอร์ชัน 1.22: เป็นภาษาหลักในการพัฒนาระบบส่วนหลัง (Backend) ร่วมกับ เว็บเฟรมเวิร์กไฟเบอร์ (Fiber Web Framework) เพื่อจัดการการรับส่งข้อมูลผ่านเว็บฮุก (Webhook)

2. เทคโนโลยีการรู้จำอักขระด้วยแสง (Optical Character Recognition: OCR): โดยเลือกใช้ เทสเซอร์แรค โอซีอาร์ (Tesseract OCR) สำหรับการแปลงภาพใบเสร็จเป็นข้อความ

3. การถอดรหัสภาพ (Image Decoding): ใช้ไลบรารี gozxing สำหรับอ่านข้อมูลจาก รหัสคิวอาร์ (QR Code) ตามมาตรฐานพร้อมเพย์

4. อัลกอริทึมการตรวจสอบความถูกต้อง: ประยุกต์ใช้เทคนิค ระยะห่างเลเวนชไตน์แบบหน้าต่าง (Windowed Levenshtein Distance) เพื่อเปรียบเทียบความคล้ายคลึงของชุดข้อมูล

5. ส่วนติดต่อผู้ใช้งาน (User Interface): ให้บริการผ่าน บัญชีไลน์ทางการ (LINE Official Account) เพื่อความสะดวกในการใช้งาน

6. ระบบจัดการฐานข้อมูล (Database Management System): เลือกใช้ โพสต์เกรสคิวเอล (PostgreSQL) ในการจัดเก็บประวัติธุรกรรมและผลการตรวจสอบ

4.2.2 ชุดข้อมูลสำหรับการทดสอบ (Test Dataset): ผู้วิจัยได้รวบรวมกลุ่มตัวอย่าง "ใบเสร็จโอนเงินอิเล็กทรอนิกส์" ที่ได้จากการทำธุรกรรมจริงในช่วงปี 2567 ถึง 2568 โดยใช้วิธีการเลือกแบบเจาะจง (Purposive Sampling) จำนวนรวมทั้งสิ้น 102 ใบ เพื่อใช้เป็นตัวแทนของความหลากหลายด้านกราฟิกและโครงสร้างข้อมูล ประกอบด้วย:

1. ธนาคารกสิกรไทย (KBANK) มี 2 รูปแบบ ได้แก่

รูปแบบที่ 1: พื้นหลังเป็นทุ่งหญ้าสายรุ้งสีพาสเทลและตัวละครสวมชุดเจ้าหญิงนั่งบนไม้

รูปแบบที่ 2: พื้นหลังเป็นลายการ์ตูนเงินและตัวละครนั่งบนกองเงิน

2. ธนาคารกรุงไทย (KTB) มี 2 รูปแบบ ได้แก่

รูปแบบที่ 1: พื้นหลังส่วนหัวเป็นโทนสีฟ้า พร้อมลวดลายร่มสีต่าง ๆ

รูปแบบที่ 2: พื้นหลังส่วนหัวไล่เฉดสีฟ้าไปส้มแบบโค้ง

3. ธนาคารไทยพาณิชย์ (SCB) มี 1 รูปแบบ ได้แก่

รูปแบบที่ 1: พื้นหลังสีม่วง ส้ม มีภาพท้องฟ้าและก้อนเมฆ และทุ่งลาเวนเดอร์

แต่ละรูปแบบใช้จำนวน 14 ถึง 34 ใบเสร็จ รวมทั้งหมด 102 ใบเสร็จ เพื่อประเมินประสิทธิภาพของระบบในเชิงโครงสร้างภาพและการตรวจสอบความถูกต้องจาก OCR และ QR Code

4.2 ระบบการออกแบบระบบ (System Design) ผู้วิจัยได้ออกแบบสถาปัตยกรรมระบบให้รองรับการทำงานแบบกึ่งเรียลไทม์ โดยมีการทำงานร่วมกันระหว่างส่วนหน้า (Frontend) คือ LINE Platform และส่วนหลัง (Backend) โดยออกแบบฐานข้อมูล PostgreSQL เพื่อจัดเก็บประวัติธุรกรรมและผลการตรวจสอบ

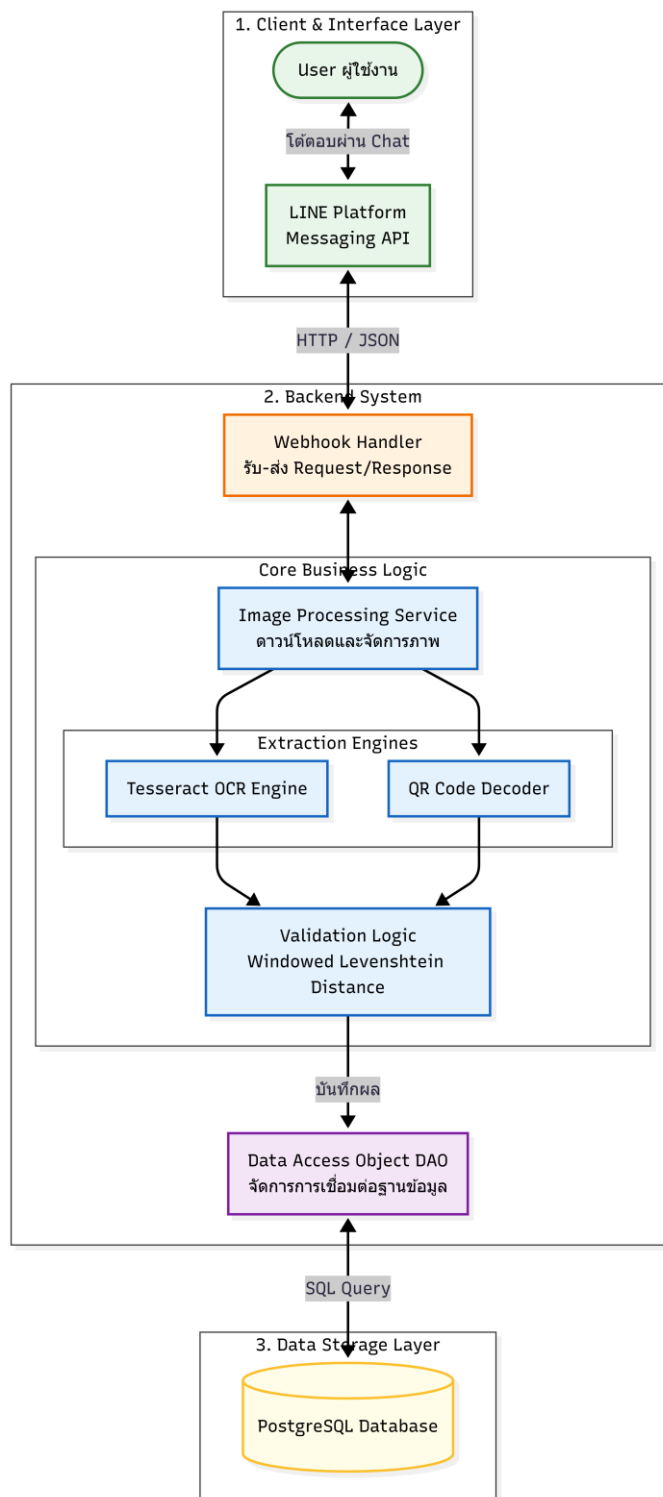
4.2.1 การออกแบบฐานข้อมูล (Database Design) ในการพัฒนาระบบนี้ ผู้วิจัยได้เลือกใช้ระบบจัดการฐานข้อมูล PostgreSQL ซึ่งเป็นฐานข้อมูลเชิงสัมพันธ์ (Relational Database) ที่มีความเสถียรสูง โดยได้ออกแบบโครงสร้างการจัดเก็บข้อมูลให้รองรับการตรวจสอบธุรกรรมแบบกึ่งเรียลไทม์และสามารถตรวจสอบย้อนหลังได้ (Audit Trail) โดยมีรายละเอียดการจัดเก็บข้อมูลหลัก 3 ส่วนสำคัญ ดังนี้:

1. ข้อมูลการระบุตัวตนและช่องทาง (Identity & Channel): ระบบจะจัดเก็บข้อมูลรหัสผู้ใช้งาน (User ID) และข้อมูลการเชื่อมต่อผ่าน LINE Platform (LINE User ID) เพื่อใช้ในการตอบกลับผลการตรวจสอบไปยังผู้ใช้ที่ถูกต้อง

2. ข้อมูลรายละเอียดธุรกรรม (Transaction Details): ประกอบด้วยข้อมูลสำคัญที่ได้จากการถอดรหัสและ OCR ได้แก่ รหัสอ้างอิงรายการ (Transaction Reference Number), ข้อมูลดิบจาก QR Code, จำนวนเงิน, วันที่และเวลาที่โอน รวมถึงข้อมูลธนาคารและเลขบัญชีของผู้ส่งและผู้รับโอน

3. สถานะและผลการตรวจสอบ (Status & Verification Logs): ใช้สำหรับบันทึกผลลัพธ์จากการประมวลผล เช่น ค่าความคล้ายคลึง (Similarity Score) ที่คำนวณได้, สถานะความถูกต้องของใบเสร็จ (Valid/Invalid), ข้อความแจ้งเตือน (Message), และวันที่เวลาที่ทำการบันทึกข้อมูล (Timestamp)

การออกแบบนี้มุ่งเน้นความถูกต้องครบถ้วนของข้อมูล (Data Integrity) และประสิทธิภาพในการเรียกดูประวัติการตรวจสอบย้อนหลัง



ภาพที่ 1 สถาปัตยกรรมของระบบการถอดข้อมูลใบเสร็จการโอนเงินของธนาคาร

โครงสร้างสถาปัตยกรรมของระบบการถอดข้อมูลใบเสร็จการโอนเงินของธนาคารมีองค์ประกอบการทำงานออกเป็น 3 ส่วนหลัก ดังนี้:

Client & Interface Layer (ส่วนติดต่อผู้ใช้และภายนอก):

1. User: ผู้ใช้งานที่ส่งรูปสลิปผ่านแอปพลิเคชัน LINE
2. LINE Platform: ทำหน้าที่เป็นตัวกลาง (Interface) ในการรับรูปภาพจากผู้ใช้งานและส่งข้อมูลต่อมายังระบบหลังบ้านผ่าน Webhook API

Backend System (ส่วนประมวลผลหลัก - พัฒนาด้วย Go 1.22):

1. Webhook Handler: เป็นโมดูลด่านหน้าสำหรับรับ HTTP Request จาก LINE และส่ง Response ตอบกลับ
2. Image Processing Service: โมดูลสำหรับดาวน์โหลดไฟล์ภาพและเตรียมไฟล์ให้พร้อมสำหรับกระบวนการถอดข้อมูล
3. Extraction Engines: หน่วยประมวลผลย่อย ประกอบด้วย Tesseract OCR (สำหรับอ่านตัวหนังสือ) และ QR Decoder (สำหรับถอดรหัสภาพ) ทำงานแยกกันเพื่อให้ได้ข้อมูลมาเปรียบเทียบ
4. Validation Logic: โมดูลตัดสินใจความถูกต้อง โดยใช้อัลกอริทึม Windowed Levenshtein Distance คำนวณความคล้ายคลึงของข้อมูลทั้งสองแหล่ง
5. Data Access Object (DAO): โมดูลกลางสำหรับจัดการคำสั่ง SQL เพื่อบันทึกหรือดึงข้อมูล เพื่อแยก Logic การเขียนโค้ดออกจากฐานข้อมูล

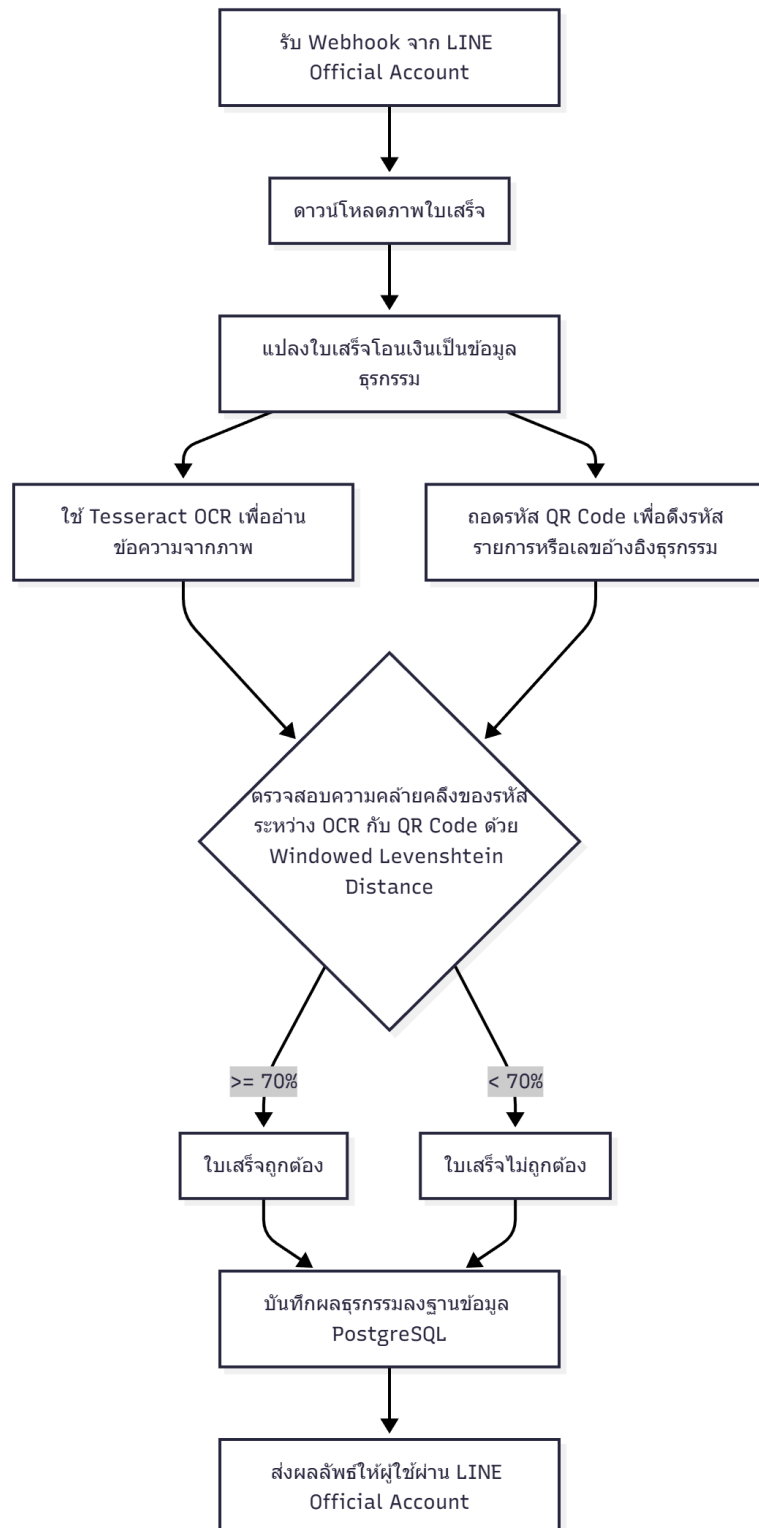
Data Storage Layer (ส่วนจัดเก็บข้อมูล):

1. PostgreSQL: ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) สำหรับจัดเก็บประวัติธุรกรรม ผลการตรวจสอบ

4.3 ระยะเวลาพัฒนาระบบ (Implementation)

4.3.1 เครื่องมือและสภาพแวดล้อมในการพัฒนา (Development Tools and Environment) ระบบถูกพัฒนาบนระบบปฏิบัติการ Linux Mint 22.2 โดยใช้เครื่องมือและไลบรารีหลักดังนี้:

1. ภาษาโปรแกรม: ใช้ภาษา Go (Golang) เวอร์ชัน 1.22
2. เว็บเฟรมเวิร์ก: Fiber (v2)
3. ระบบจัดการฐานข้อมูล: PostgreSQL เวอร์ชัน 18
4. เอนจินการรู้จำอักขระ: Tesseract OCR เวอร์ชัน 5
5. ไลบรารี: gozxing (สำหรับ QR Code) และ gosseract (สำหรับเชื่อมต่อ OCR)



ภาพที่ 2 ผังงานการทำงานของระบบ (System Workflow)

การพัฒนากระบวนการถอดข้อมูลใบเสร็จโอนเงินอัตโนมัติ ได้ดำเนินการพัฒนาตามขั้นตอนการทำงานที่ออกแบบไว้ ดังแสดงใน ภาพที่ 2 โดยมีรายละเอียดกระบวนการทำงานของระบบ (System Workflow) ดังนี้

1. การรับข้อมูล (Input Reception): ระบบเริ่มต้นการทำงานเมื่อได้รับสัญญาณ Webhook จาก LINE Official Account ซึ่งเกิดจากการที่ผู้ใช้งานอัปโหลดรูปภาพใบเสร็จโอนเงินเข้ามาในระบบ
2. การเตรียมข้อมูล (Data Preparation): ส่วนประมวลผลหลัก (Backend) จะทำการดาวน์โหลดไฟล์ภาพต้นฉบับจากเซิร์ฟเวอร์ของ LINE มาพักไว้เพื่อเตรียมเข้าสู่กระบวนการถอดข้อมูล
3. การประมวลผลขนาน (Parallel Processing): ระบบประมวลผลข้อมูลพร้อมกันสองส่วน ได้แก่ การอ่านข้อมูลตัวอักษรจากภาพด้วย Tesseract OCR และการถอดรหัส QR Code เพื่อดึงข้อมูลธุรกรรมดิจิทัล
4. การตรวจสอบความถูกต้อง (Validation Logic): ระบบนำรหัสอ้างอิงที่ได้จาก OCR และ QR Code มาเปรียบเทียบกับอัลกอริทึม Windowed Levenshtein Distance เพื่อคำนวณค่าความคล้ายคลึง โดยกำหนดเกณฑ์ร้อยละ 70 สำหรับการตัดสินใจความถูกต้องของใบเสร็จ
5. การจัดเก็บและแสดงผล (Storage & Output): ระบบบันทึกข้อมูลธุรกรรมและผลการตรวจสอบลงในฐานข้อมูล PostgreSQL และส่งผลลัพธ์กลับไปยังผู้ใช้งานผ่าน LINE Official Account ในรูปแบบกึ่งเรียลไทม์

4.4 ระยะเวลาทดสอบระบบและเกณฑ์การประเมิน (System Testing and Evaluation Criteria) ผู้วิจัยได้ทดสอบประสิทธิภาพของระบบโดยนำภาพใบเสร็จจากกลุ่มตัวอย่างเข้าสู่กระบวนการประมวลผลจริง และเปรียบเทียบผลลัพธ์กับข้อมูลอ้างอิงที่ถูกต้อง เพื่อประเมินความแม่นยำและประสิทธิภาพของระบบ

4.4.1 ขั้นตอนการทดสอบ (Testing Procedures)

1. นำเข้าไฟล์ภาพใบเสร็จโอนเงินจากกลุ่มตัวอย่างจำนวน 102 ใบ ผ่านช่องทาง LINE Official Account เข้าสู่ระบบ
2. ระบบถอดรหัสอ้างอิงจากภาพด้วย Tesseract OCR และถอดข้อมูลจาก QR Code
3. เปรียบเทียบความคล้ายคลึงของข้อมูลทั้งสองชุดด้วยอัลกอริทึม Windowed Levenshtein Distance
4. บันทึกค่าความคล้ายคลึง สถานะความถูกต้อง และเวลาในการประมวลผลลงในฐานข้อมูล เพื่อนำไปวิเคราะห์ทางสถิติ

4.4.2 เกณฑ์การประเมินและดัชนีชี้วัด (Evaluation Criteria and Metrics) ผู้วิจัยได้กำหนดดัชนีชี้วัดประสิทธิภาพของระบบใน 3 ด้านหลัก ดังนี้

(1) เกณฑ์การยอมรับความคล้ายคลึง (Similarity Threshold)

กำหนดค่าขีดแบ่งความคล้ายคลึงที่ ร้อยละ 70 (70%) สำหรับการตัดสินว่าใบเสร็จถูกต้อง (Valid) การกำหนดค่านี้มีเหตุผลรองรับทางเทคนิคเพื่อจัดการกับข้อจำกัดของ OCR ภาษาไทยและสัญญาณรบกวนบนภาพ (Background Noise) ดังนี้:

1. การจัดการความคลาดเคลื่อนของอักขระ (Character Error Tolerance): Tesseract OCR มักเกิดความคลาดเคลื่อนในการอ่านสระ วรรณยุกต์ภาษาไทย หรืออักขระที่มีลักษณะใกล้เคียงกันจากทลายพื้นหลัง เกณฑ์ร้อยละ 70 จึงช่วยยอมรับความผิดพลาดเล็กน้อยโดยไม่ปฏิเสธข้อมูลที่ถูกต้อง

2. การยืนยันข้อมูล (Verification Focus): ระบบเน้นการตรวจสอบความสอดคล้องระหว่างข้อมูลจาก OCR และ QR Code มากกว่าการถอดข้อความอย่างสมบูรณ์ ค่าความคล้ายคลึงตั้งแต่ร้อยละ 70 ขึ้นไปจึงเพียงพอในการยืนยันว่าใบเสร็จไม่มีการตัดต่อหรือสวมสิทธิ์ QR Code จากรายการอื่น

(2) การประเมินความแม่นยำ (Accuracy Assessment) วัดประสิทธิภาพความถูกต้องของระบบโดยรวม ใช้สูตรคำนวณดังนี้:

$$Accuracy(\%) = \frac{\text{จำนวนใบเสร็จที่ตรวจสอบถูกต้อง}}{\text{จำนวนใบเสร็จทั้งหมด}} \times 100$$

(3) การประเมินความเสถียร (Stability Assessment) วัดความสม่ำเสมอของเวลาในการประมวลผล (Processing Time) เพื่อยืนยันว่าระบบสามารถทำงานได้ในเวลาที่ใกล้เคียงกันแม้ภาพจะมีลักษณะต่างกัน โดยใช้ค่า ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: S.D.) คำนวณจากสูตร:

$$S.D. = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

5. ผลการศึกษา

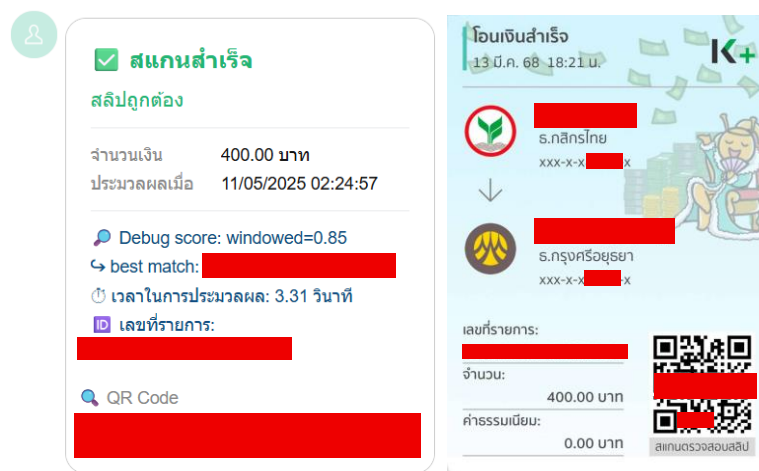
การทดลองในงานวิจัยนี้มุ่งประเมินความถูกต้องและประสิทธิภาพของระบบถอดข้อมูลใบเสร็จอิเล็กทรอนิกส์ โดยใช้กลุ่มตัวอย่างใบเสร็จจาก ธนาคารกรุงไทย ธนาคารไทยพาณิชย์ และ ธนาคารกสิกรไทย ธนาคารละ 34 ใบเสร็จ รวมทั้งหมด 102 ใบเสร็จ ผลการทดสอบแบ่งออกเป็น 2 ส่วน ได้แก่ ผลการทดสอบความถูกต้องของระบบ และผลการทดสอบเวลาเฉลี่ยในการประมวลผล

5.1 ผลการทดสอบความแม่นยำของระบบ

ผลการทดสอบแสดงในตาราง โดยเปรียบเทียบจำนวนใบเสร็จที่ระบบสามารถถอดข้อมูลได้ถูกต้องตามเกณฑ์ความคล้ยคลึงของรหัสอ้างอิง ($\geq 70\%$)

ตารางที่ 1 ผลการทดสอบความแม่นยำของระบบ (ธนาคารกสิกรไทย)

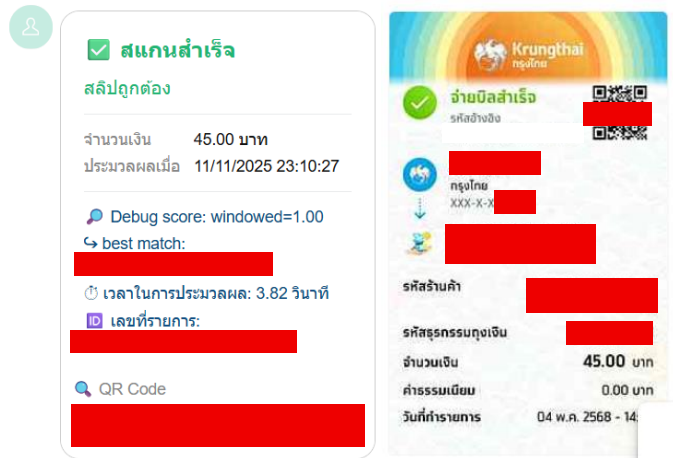
รูปแบบใบเสร็จ ธนาคารกสิกรไทย	จำนวนใบเสร็จที่ ทดสอบ (ใบ)	ถูกต้อง (ใบ)	ผิดพลาด (ใบ)	ร้อยละความ ถูกต้อง (%)
รูปแบบที่ 1	17	17	0	100
รูปแบบที่ 2	17	17	0	100
เฉลี่ยรวมทั้งหมด	34	34	0	100



ภาพที่ 3 ตัวอย่างผลลัพธ์การทดสอบใบเสร็จของธนาคารกสิกรไทย

ตารางที่ 2 ผลการทดสอบความแม่นยำของระบบ (ธนาคารกรุงไทย)

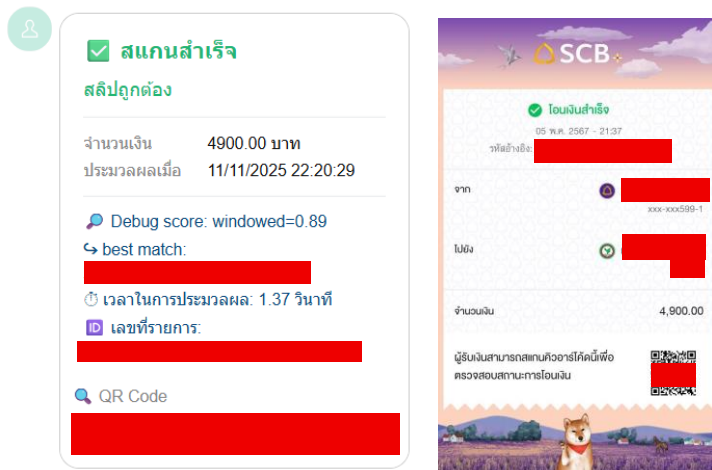
รูปแบบใบเสร็จ ธนาคารกรุงไทย	จำนวนใบเสร็จที่ ทดสอบ (ใบ)	ถูกต้อง (ใบ)	ผิดพลาด (ใบ)	ร้อยละความ ถูกต้อง (%)
รูปแบบที่ 1	17	17	0	100
รูปแบบที่ 2	16	16	1	94.12
เฉลี่ยรวมทั้งหมด	34	33	1	97.06



ภาพที่ 4 ตัวอย่างผลลัพธ์การทดสอบใบเสร็จของธนาคารกรุงไทย

ตารางที่ 3 ผลการทดสอบความแม่นยำของระบบ (ธนาคารไทยพาณิชย์)

รูปแบบใบเสร็จ ธนาคารไทยพาณิชย์	จำนวนใบเสร็จที่ ทดสอบ (ใบ)	ถูกต้อง (ใบ)	ผิดพลาด (ใบ)	ร้อยละความ ถูกต้อง (%)
รูปแบบที่ 1	34	34	0	100
เฉลี่ยรวมทั้งหมด	34	34	0	100



ภาพที่ 5 ตัวอย่างผลลัพธ์การทดสอบใบเสร็จของธนาคารไทยพาณิชย์

ตารางที่ 4 สรุปผลการทดสอบความแม่นยำรวมทุกธนาคาร

ธนาคาร	จำนวนทั้งหมด (ใบ)	ถูกต้อง (ใบ)	ผิดพลาด (ใบ)	ร้อยละความ ถูกต้อง (%)
กสิกรไทย	34	34	0	100
กรุงไทย	34	33	1	97.06
ไทยพาณิชย์	34	34	0	100
เฉลี่ยรวมทั้งหมด	102	101	1	99.02

จากตารางที่ 1-4 ระบบสามารถถอดข้อมูลและตรวจสอบใบเสร็จได้ถูกต้อง 101 จาก 102 ใบ คิดเป็นอัตราความถูกต้องเฉลี่ยร้อยละ 99.02 โดยธนาคารกสิกรไทยและธนาคารไทยพาณิชย์มีความถูกต้องร้อยละ 100 ขณะที่ธนาคารกรุงไทยมีความถูกต้องร้อยละ 97.06 ผลลัพธ์ดังกล่าวแสดงให้เห็นว่าระบบสามารถรองรับความแตกต่างของรูปแบบใบเสร็จในแต่ละธนาคารได้อย่างมีประสิทธิภาพภายใต้เกณฑ์การเปรียบเทียบรหัสอ้างอิงที่กำหนดไว้ ($\geq 70\%$) ทั้งนี้ การแยกรูปแบบย่อยของใบเสร็จ โดยเฉพาะในกรณีของธนาคารกสิกรไทยและธนาคารกรุงไทย ช่วยให้ระบบเลือกวิธีประมวลผลที่เหมาะสมกับลักษณะภาพได้ดียิ่งขึ้น และส่งผลให้ความแม่นยำในการถอดข้อมูลเพิ่มขึ้นอย่างชัดเจน

5.2 ผลการทดสอบเวลาเฉลี่ยในการประมวลผล

แสดงผลตารางการทดสอบด้านเวลาเฉลี่ยในการประมวลผลของแต่ละธนาคาร โดยวัดระยะเวลาตั้งแต่ระบบได้รับข้อมูลภาพจาก LINE Official Account จนถึงการส่งข้อความตอบกลับผลลัพธ์ให้ผู้ใช้

ตารางที่ 5 ผลการทดสอบเวลาเฉลี่ย (ธนาคารกสิกรไทย)

รูปแบบใบเสร็จ ธนาคารกสิกรไทย	จำนวนใบเสร็จที่ ทดสอบ (ใบ)	เวลา ประมวลผล เฉลี่ย (วินาที)	เวลา ประมวลผล รวม (วินาที)	ส่วนเบี่ยงเบน มาตรฐาน (S.D.)
รูปแบบที่ 1	17	1.69	28.79	0.25
รูปแบบที่ 2	17	1.61	27.42	0.62
เฉลี่ยรวมทั้งหมด	34	1.65	28.105	0.44

ตารางที่ 6 ผลการทดสอบเวลาเฉลี่ย (ธนาคารกรุงไทย)

รูปแบบใบเสร็จ ธนาคารกรุงไทย	จำนวนใบเสร็จที่ ทดสอบ (ใบ)	เวลา ประมวลผล เฉลี่ย (วินาที)	เวลา ประมวลผล รวม (วินาที)	ส่วนเบี่ยงเบน มาตรฐาน (S.D.)
รูปแบบที่ 1	17	2.01	34.19	0.60
รูปแบบที่ 2	17	1.55	26.36	0.61
เฉลี่ยรวมทั้งหมด	34	1.78	30.28	0.60

ตารางที่ 7 ผลการทดสอบเวลาเฉลี่ย (ธนาคารไทยพาณิชย์)

รูปแบบใบเสร็จ ธนาคารไทยพาณิชย์	จำนวนใบเสร็จที่ ทดสอบ (ใบ)	เวลา ประมวลผล เฉลี่ย (วินาที)	เวลา ประมวลผล รวม (วินาที)	ส่วนเบี่ยงเบน มาตรฐาน (S.D.)
รูปแบบที่ 1	34	1.84	62.71	0.46

ตารางที่ 8 สรุปผลเวลาเฉลี่ยรวมทุกธนาคาร

ธนาคาร	จำนวนทั้งหมด (ใบ)	เวลาประมวลผล เฉลี่ย (วินาที)	เวลา ประมวลผลรวม (วินาที)	ส่วนเบี่ยงเบน มาตรฐาน (S.D.)
กสิกรไทย	34	1.65	28.105	0.44
กรุงไทย	34	1.78	30.28	0.60
ธนาคารไทยพาณิชย์	34	1.84	62.71	0.46
เฉลี่ยรวมทั้งหมด	102	1.76	121.09	0.50

จากการทดสอบการประมวลผลภาพใบเสร็จจำนวน 102 ใบ พบว่าระบบใช้เวลาเฉลี่ย 1.76 วินาทีต่อใบเสร็จ (S.D. 0.50) โดยรูปแบบของใบเสร็จมีผลต่อความเร็วในการประมวลผลอย่างชัดเจน ธนาคารกสิกรไทยมีเวลาเฉลี่ย 1.65 วินาที (S.D. 0.44) โดยรูปแบบที่ 2 เร็วกว่าแบบที่ 1 เล็กน้อย สะท้อนว่าความคมชัดระหว่างตัวอักษรกับพื้นหลังช่วยให้ OCR ทำงานได้ดี แม้มีลวดลาย ขณะที่ธนาคารกรุงไทยมีเวลาเฉลี่ย 1.78 วินาที (S.D. 0.60) และพบความแตกต่างของรูปแบบอย่างชัดเจน โดยรูปแบบที่มีพื้นหลังซับซ้อนใช้เวลานานกว่าอย่างมีนัยสำคัญ ส่วนธนาคารไทยพาณิชย์ซึ่งมีรูปแบบเดียว ใช้เวลาเฉลี่ย 1.84 วินาที (S.D. 0.46) แม้สูงกว่าเล็กน้อยแต่มีความเสถียรจากโครงสร้างที่คงที่

โดยรวมระบบมีประสิทธิภาพสูง รองรับการงานแบบกึ่งเรียลไทม์ และสามารถตอบสนองผู้ใช้งานได้ภายในเวลาไม่ถึง 2 วินาที เหมาะสำหรับการประยุกต์ใช้ในระบบตรวจสอบการชำระเงินอัตโนมัติ

6.อภิปรายผล

จากการทดลองระบบตรวจสอบความถูกต้องของใบเสร็จโอนเงินจำนวน 102 ใบ พบว่าระบบมีประสิทธิภาพสูงในระดับที่น่าพอใจ โดยมีอัตราความแม่นยำเฉลี่ยร้อยละ 99.02 และเวลาประมวลผลเฉลี่ย 1.76 วินาทีต่อใบเสร็จ องค์ความรู้ใหม่ที่ได้จากงานวิจัยนี้คือการพิสูจน์ประสิทธิภาพของ "กระบวนการตรวจสอบแบบไฮบริด (Hybrid Verification)" ที่บูรณาการระหว่างข้อมูลภาพ (OCR) และข้อมูลดิจิทัล (QR Code) ซึ่งผลการวิจัยแสดงให้เห็นว่าการใช้เทคนิค Windowed Levenshtein Distance ที่เกณฑ์ความคล้ายคลึงร้อยละ 70 เป็นจุดสมดุลที่เหมาะสมในการแก้ปัญหาความคลาดเคลื่อนทางอักขระภาษาไทยและปัญหาสระลอยที่มักพบใน Tesseract OCR ทำให้ระบบสามารถยอมรับความผิดพลาดเล็กน้อย (Typographical Tolerance) โดยยังคงความสามารถในการคัดกรองใบเสร็จที่ผิดปกติได้อย่างแม่นยำ ซึ่งช่วยลดข้อจำกัดของการใช้ OCR เพียงลำพังในบริบทของเอกสารที่มีความซับซ้อนสูงได้เป็นอย่างดี

ในด้านประสิทธิภาพและความเสถียรของระบบ เมื่อพิจารณาค่า ส่วนเบี่ยงเบนมาตรฐาน (Standard Deviation: S.D.) ของเวลาในการประมวลผล พบว่าในภาพรวมมีค่าเท่ากับ 0.50 วินาที อย่างไรก็ตาม เมื่อวิเคราะห์จำแนกตามรายธนาคารจะพบความแตกต่างของความเสถียรที่น่าสนใจ โดย ธนาคารกสิกรไทย มีความเสถียรสูงสุดด้วยค่า S.D. ต่ำที่สุดเพียง 0.44 วินาที รองลงมาคือ ธนาคารไทยพาณิชย์ ที่มีค่า S.D. อยู่ที่ 0.46 วินาที ซึ่งถือว่ามีความใกล้เคียงกันมาก แสดงให้เห็นว่าใบเสร็จที่มีโครงสร้างและพื้นหลังเป็นระเบียบจะช่วยให้ระบบประมวลผลได้ด้วยเวลาที่ใกล้เคียงกันสม่ำเสมอ ในขณะที่ ธนาคารกรุงไทย มีค่า S.D. สูงสุดที่ 0.60 วินาที สะท้อนให้เห็นถึงความแปรปรวนของระยะเวลาประมวลผลที่ได้รับผลกระทบจากความซับซ้อนของลวดลายกราฟิกพื้นหลังที่แตกต่างกันในแต่ละรูปแบบ

สำหรับการวิเคราะห์ข้อผิดพลาด พบความผิดพลาดจำนวน 1 รายการ (คิดเป็นร้อยละ 2.94 ของธนาคารกรุงไทย) ในใบเสร็จรูปแบบที่ 2 ซึ่งมีพื้นหลังไล่เฉดสี จากการตรวจสอบพบว่าสาเหตุหลักเกิดจากคุณภาพของไฟล์ภาพต้นฉบับที่มีความคมชัดต่ำ (Low Contrast) ประกอบกับตำแหน่งของรหัสอ้างอิงถูกวางทับอยู่บนรอยต่อของการไล่ระดับสีพื้นหลังพอดี ส่งผลให้ Tesseract OCR ไม่สามารถแยกแยะขอบของตัวอักษรออกจากพื้นหลังได้ชัดเจน จึงอ่านค่าผิดพลาดจนค่าความคล้ายคลึงต่ำกว่าเกณฑ์ที่กำหนด ปัญหานี้สอดคล้องกับงานวิจัยของ Sayallar และคณะ (2023) ที่ระบุว่า Tesseract OCR มักมีข้อจำกัดด้าน Recall เมื่อต้องจัดการกับเอกสารที่มีสัญญาณรบกวน

(Background Noise) สูง แต่ในงานวิจัยนี้ได้นำเสนอทางออกด้วยการใช้ QR Code มาช่วยสอบทานความถูกต้อง ทำให้ภาพรวมความแม่นยำยังคงอยู่ในระดับสูง

นอกจากนี้ ผลการวิจัยยังมีความสอดคล้องกับงานวิจัยของ ณรงค์เกียรติ นามห้วยทอง และคณะ (2568) ที่พบว่าการใช้ภาษา Go (Golang) ช่วยให้ระบบประมวลผลได้รวดเร็ว เหมาะสำหรับการทำงานแบบเรียลไทม์ และสอดคล้องกับ Thammarak และคณะ (2022) ในประเด็นความท้าทายของการอ่านสระและวรรณยุกต์ไทย ทั้งนี้ ระบบที่พัฒนาขึ้นยังมีจุดเด่นเรื่อง ความสามารถในการขยายผล (Scalability) โดยสามารถรองรับการตรวจสอบใบเสร็จของธนาคารอื่นๆ ในประเทศไทยได้ทันที เนื่องจากโครงสร้างข้อมูลหลักอ้างอิงตามมาตรฐาน QR Code PromptPay (EMVCo Standard) ซึ่งเป็นมาตรฐานกลางที่ทุกธนาคารใช้งานร่วมกัน หรือสามารถประยุกต์ใช้ร่วมกับ API ของธนาคารเพื่อการตรวจสอบที่แม่นยำยิ่งขึ้นในอนาคต

7. สรุปผลการวิจัย

จากการวิจัยเรื่อง “ระบบถอดข้อมูลใบเสร็จโอนเงินอัตโนมัติ โดยใช้เทคโนโลยี Tesseract OCR ร่วมกับการถอดรหัส QR Code” เพื่อพัฒนาระบบตรวจสอบความถูกต้องของข้อมูลใบเสร็จโอนเงินด้วยภาษา Go 1.22 ร่วมกับ Framework Fiber โดยใช้เทคโนโลยี Tesseract OCR ในการอ่านข้อความจากภาพ ร่วมกับการถอดรหัสข้อมูลธุรกรรมจาก QR Code เพื่อใช้เป็นเกณฑ์อ้างอิงในการตรวจสอบรหัสรายการ (Transaction Reference Number) ผลการวิจัยพบว่า ระบบที่พัฒนาขึ้นมีประสิทธิภาพสูงในการเปรียบเทียบและตรวจสอบความถูกต้อง โดยจากการทดสอบกับกลุ่มตัวอย่างใบเสร็จจำนวน 102 ใบ จาก 3 ธนาคารหลัก ได้แก่ ธนาคารกสิกรไทย ธนาคารกรุงไทย และธนาคารไทยพาณิชย์ (ธนาคารละ 34 ใบ) ระบบสามารถตรวจสอบความถูกต้องได้จำนวน 101 ใบ คิดเป็นค่าเฉลี่ยร้อยละ 99.02 และมีระยะเวลาในการประมวลผลเฉลี่ย 1.76 วินาทีต่อใบเสร็จ ส่วนเบี่ยงเบนมาตรฐาน 0.50 วินาที ซึ่งค่าความเบี่ยงเบนที่ต่ำสะท้อนให้เห็นถึงความสม่ำเสมอในการทำงาน และความพร้อมในการนำไปประยุกต์ใช้งานจริงในสภาพแวดล้อมแบบกึ่งเรียลไทม์ (Semi-Realtime)

นอกจากนี้ การแยกใบเสร็จตามรูปแบบของแต่ละธนาคารช่วยเพิ่มความแม่นยำในการระบุตำแหน่งข้อมูล โดยใบเสร็จที่มีโครงสร้างชัดเจนและพื้นหลังตัดกับตัวอักษรให้ผลลัพธ์ OCR ที่ดีกว่า ในขณะที่ความคลาดเคลื่อนส่วนใหญ่มาจากคุณภาพของภาพต้นทาง เช่น ความคมชัดต่ำหรือพื้นหลังที่ทำให้ตัวอักษรอ่านยาก

ในด้านองค์ความรู้ พบว่าความแม่นยำของระบบไม่ได้ขึ้นอยู่กับ OCR เพียงอย่างเดียว แต่ยังขึ้นกับการประยุกต์ใช้เทคนิค Windowed Levenshtein Distance โดยการกำหนดเกณฑ์ความคล้ายคลึงตั้งแต่ร้อยละ 70 ขึ้นไปช่วยให้ระบบสามารถจับคู่ข้อมูลระหว่าง OCR และ QR Code ได้

อย่างมีประสิทธิภาพแม้เกิดความผิดพลาดเล็กน้อย อีกทั้งลักษณะทางกายภาพของใบเสร็จมีผลต่อค่า Confidence Score ของ OCR อย่างมีนัยสำคัญ

โดยสรุป ระบบต้นแบบที่พัฒนาขึ้นมีความถูกต้อง รวดเร็ว และเสถียรในระดับที่ผู้ใช้งานยอมรับได้สำหรับการทำงานแบบ Near Real-time ผ่าน Chatbot และมีศักยภาพในการนำไปประยุกต์ใช้จริง รวมถึงต่อยอดสู่การพัฒนาเชิงพาณิชย์หรือระบบตรวจสอบธุรกรรมอัตโนมัติขององค์กร เพื่อช่วยลดภาระการตรวจสอบด้วยบุคคลและเพิ่มประสิทธิภาพการดำเนินงานในระยะยาว

8. ข้อเสนอแนะ

จากผลการวิจัยเรื่อง “ระบบถอดข้อมูลใบเสร็จโอนเงินอัตโนมัติ โดยใช้เทคโนโลยี Tesseract OCR ร่วมกับการถอดรหัส QR Code” พบว่าระบบที่พัฒนาขึ้นสามารถนำไปประยุกต์ใช้ในหน่วยงานที่ต้องตรวจสอบใบเสร็จการโอนเงินจำนวนมาก เช่น สถาบันการศึกษา หน่วยงานราชการ และธุรกิจออนไลน์ โดยช่วยลดภาระการตรวจสอบด้วยบุคคล ลดความคลาดเคลื่อนจากมนุษย์ และเพิ่มความรวดเร็วในการตรวจสอบในระดับกึ่งเรียลไทม์ อีกทั้งสามารถเชื่อมต่อกับช่องทางสื่อสาร เช่น LINE Official Account เพื่อให้ผู้ใช้ส่งใบเสร็จและรับผลการตรวจสอบได้โดยอัตโนมัติ ทั้งนี้ แนวทางการพัฒนาต่อยอดควรมุ่งเน้นการปรับปรุงขั้นตอนการเตรียมภาพก่อนประมวลผล เช่น การลดสัญญาณรบกวน การจัดแนวภาพ และการปรับความคมชัดอัตโนมัติ รวมถึงการเพิ่มความหลากหลายของชุดข้อมูลใบเสร็จจากหลายธนาคาร เพื่อเพิ่มความแม่นยำและความครอบคลุมของระบบ และในระยะยาวสามารถประยุกต์ใช้เทคนิคปัญญาประดิษฐ์หรือการเรียนรู้ของเครื่องในการจำแนกรูปแบบใบเสร็จและปรับพารามิเตอร์การถอดข้อมูลโดยอัตโนมัติ ซึ่งจะช่วยเพิ่มประสิทธิภาพ ความเสถียร และศักยภาพในการพัฒนาเป็นระบบตรวจสอบธุรกรรมทางการเงินอัจฉริยะในอนาคต

9. กิตติกรรมประกาศ

บทความนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความกรุณาอย่างสูงจาก อาจารย์ อรรถวิท ชังคมานนท์ ประธานที่ปรึกษาโครงการ ผู้ช่วยศาสตราจารย์ ก่องกาญจน์ ดุลยไชย และ ผู้ช่วยศาสตราจารย์ ดร.พาสน์ ปราโมกษ์ชน กรรมการที่ปรึกษาโครงการ ที่ให้คำแนะนำและข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์อย่างยิ่งต่องานวิจัย ขอขอบคุณบริษัท เพย์ โซลูชั่น จำกัด ที่ให้โอกาสและการสนับสนุนรวมถึงการให้คำปรึกษา ขอขอบคุณบุคลากรสาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้ ที่ให้ความรู้และคำปรึกษาด้านการจัดทำเอกสารโครงการ รวมถึงการสนับสนุนตลอดจนการปรับปรุง แก้ไขข้อบกพร่องต่าง ๆ จนทำให้งานวิจัยครั้งนี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้

10. เอกสารอ้างอิง

- ณรงค์เกียรติ นามห้วยทอง และคณะ. (2568). การศึกษาประสิทธิภาพของ Tesseract OCR สำหรับการประมวลผลภาพในการตรวจสอบธุรกรรมทางการเงิน. *วารสารแม่โจ้เทคโนโลยีสารสนเทศและนวัตกรรม*, 9(2), 51–60.
https://mitij.mju.ac.th/Search_Detail_Journal_MJU.aspx?Herb_ID=0209B
- วิศรุต เหล่าดารา. (2565). ระบบเซ็นเซอร์ข้อมูลคลออกจากเอกสารแนกคำพิพากษาด้วยปัญญาประดิษฐ์ (สารนิพนธ์ปริญญาโทบริหารธุรกิจ). มหาวิทยาลัยธุรกิจบัณฑิตย์
<https://libdoc.dpu.ac.th/thesis/Wisarut.Kang.pdf>
- Google. (n.d.). **Tesseract OCR**.
Retrieved from <https://github.com/tesseract-ocr/tesseract>
- Levenshtein, V. I. (1966). **Binary codes capable of correcting deletions, insertions, and reversals**. *Soviet Physics Doklady*, 10(8), 707–710.
<https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf>
- Sayallar, C., Sayar, A., & Babalik, N. (2023). **An OCR engine for printed receiptImages using deep learning techniques**. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(2).
<https://thesai.org/Publications/ViewPaper?Volume=14&Issue=2&Code=IJACSA&SerialNo=95>
- Thammarak, K., Kongkla, P., Sirisathitkul, Y., & Intakosum, S. (2022). **Comparative analysis of Tesseract and Google Cloud Vision for Thai vehicleregistration certificate**. *International Journal of Electrical and Computer Engineering (IJECE)*, 12(2), 1849-1858. <http://doi.org/10.11591/ijece.v12i2.pp1849-1858>