

1. Title + Author Information

เนื้อหาที่ต้องมี:

- ชื่อเรื่อง (Title)
 - ชื่อผู้เขียน
 - สังกัด/ที่อยู่หน่วยงาน
 - Corresponding Author
-

2. Abstract (150–250 words)

ต้องสรุป:

- ปัญหา (Problem)
 - วิธีการ (Methods) เช่น YOLO + CNN + Regression
 - ระบบที่พัฒนา (Proposed system)
 - ผลลัพธ์สำคัญ (Key results)
 - ข้อดีของระบบ (Impact)
-

3. Index Terms (Keywords)

ตัวอย่างที่ควรมี:

- Water Meter Reading
 - Analog Gauge
 - YOLO Detection
 - CNN Digit Recognition
 - Pointer Regression
 - Computer Vision
 - OCR
-

4. Introduction

ควรอธิบาย:

- ความสำคัญของการอ่านมิเตอร์น้ำ
- ปัญหาในการอ่านแบบ manual
- ข้อจำกัดของงานก่อนหน้า
- เหตุผลที่ต้องใช้ YOLO + CNN + Regression
- Contributions ของงาน (4–5 ข้อ)

ตัวอย่าง Contributions:

- พัฒนาระบบตรวจจับช่องตัวเลขและหน้าปัดด้วย YOLO
- เสนอ CNN สำหรับรู้จำตัวเลขเฉพาะแบบของมิเตอร์น้ำ
- เสนอโมเดล Regression สำหรับประมาณมุมเข็มชี้
- สร้าง dataset ภาพมิเตอร์จริง
- สร้างระบบใช้งานผ่านมือถือพร้อม backend

5. Related Work

แบ่งเป็นหมวดย่อยได้ เช่น:

- งาน OCR และ Digit Recognition (CNN, CRNN, MNIST-based)
- งาน Detection ด้วย YOLO/Deep Learning
- งานอ่านมิเตอร์หรือ gauge reading
- ข้อจำกัดของงานก่อนหน้า (เช่น อ่านเฉพาะตัวเลข, ไม่รองรับ pointer)
- ช่องว่างที่งานนี้เกิดขึ้น (gap)

6. Methodology

6.1 System Overview

- Workflow ของระบบทั้งหมด

- ภาพ Block Diagram / Pipeline

6.2 Dataset Collection & Annotation

- จำนวนภาพ
- ประเภทของมิเตอร์
- วิธี Annotation (Digit window, Dial, Pointer)
- การทํา Preprocessing
- Augmentation

6.3 YOLO Detection for Digit Window & Dial

ต้องบอก:

- YOLO รุ่นที่ใช้ (เช่น YOLOv8)
- Classes ที่ตรวจพบ
- Training config (resolution, epochs, loss, optimizer)
- ตัวอย่าง bounding box

6.4 Digit Segmentation & Preprocessing

- การครอบตัวเลข
- การ threshold
- การแยกตัวเลขแต่ละหลัก

6.5 CNN Digit Recognition

ความ:

- โครงสร้าง CNN
- จำนวนชั้น (layers)
- Activation function
- Batch size, optimizer, learning rate

6.6 Pointer Angle Estimation

อธิบาย:

- การหาศูนย์กลาง (Center) ของมิเตอร์
- การหาปลายเข็ม (Pointer Tip)

- การคำนวณมุมด้วย atan2
- โมเดล Regression ที่ใช้

6.7 Mobile/Web System

- ใช้ WebView
 - API Django
 - Publishing flow (capture → detect → read → store)
-

7. Experimental Setup

ต้องระบุ:

- Hardware (GPU/CPU)
 - Software (PyTorch, OpenCV, Django)
 - Train/Test Split
 - Metrics: mAP, Accuracy, MAE, F1-score
 - วิธีประเมิน end-to-end system
-

8. Results

ควรมีอย่างน้อย 4 ส่วน:

8.1 YOLO Detection Results

- ค่า mAP
- Precision / Recall
- ตัวอย่างภาพ Detection

8.2 Digit Recognition Results

- Accuracy เฉลี่ย
- Confusion Matrix
- ตัวอย่างภาพ Predict ถูก/ผิด

8.3 Pointer Angle Regression Results

- ค่าคลาดเคลื่อน (MAE)
- กราฟเปรียบเทียบค่าจริง/ค่าทำนาย
- ตัวอย่าง pointer detection

8.4 End-to-End Meter Reading

- ความแม่นยำของทั้งระบบ
- เวลาเฉลี่ยต่อภาพ
- ตัวอย่างการอ่านจริง

9. Discussion

ควรพูดถึง:

- จุดแข็งของระบบ
- จุดที่ยังพิศพาด (เช่น glare, blur, dirt)
- สิ่งที่ระบบทำได้ในโลกจริง
- ความเป็นไปได้ในการใช้งานจริง

10. Conclusion

เนื้อหาที่ควรมี:

- สรุปความสำเร็จของระบบ
- ข้อความแม่นยำ
- แผนพัฒนาต่อ เช่น
 - เพิ่มชุดข้อมูล
 - ใช้ Transformer-based OCR
 - Edge inference บนมือถือ

11. Acknowledgment (ຄໍາສົ່ງ)

ເຫັນ:

- ອາຈານທີ່ປະກາດ
 - ຜ່ານວ່າຍັນທີ່ໄໝກາພຫຼືອອຸປະກອບ
-

12. References

- ຮູບແບບ IEEE (ເລຂ [1], [2], ...)
- ຈານ YOLO, CNN, OCR
- ຈານເກີ່ມວັດທະນາທີ່ກຳນົດໃຫຍ້ກຳນົດ analog gauge

ໂຄຣງເປີເປົອຮ້ IEEE

1. Introduction

- ປຶ້ມຫາຂອງການອ່ານມີເຕອັນນຳ
- ຊິ້ວຍກັດຂອງຈານເຄີມ
- Motivation ຂອງຮະບບທີ່ຄູນພັດນາ
- Contribution ຂອງຮະບບຄູນ (3–5 bullet)

2. Related Work

- ຈານອ່ານມີເຕອັນນຳ
- ຈານ Vision + YOLO
- CNN Digit Recognition
- Transition Model ຂອງ pointer
- ຈານ OCR ອື່ນ ๆ

3. Methodology

- Architecture ຂອງຮະບບ (ຕາມ pipeline ໃນໄຟລ໌ຂອງຄູນ)
- YOLO Detection (ກວດອັນຕົວເລີບ + pointer)
- Digit Recognition ຕ້ວຍ CNN
- Angle Regression Model
- Transition Model
- Preprocessing
- ຖັນກາພປະກອນຈາກໄຟລ໌ເຄີມ

4. Experimental Setup

- Dataset (ຈຳນວນກາພ, ວິຊີເກີບ, labeling)
- Train/Test Split
- Evaluation Metric
- Environment

5. Results

- ຕາരາງຜດ YOLO
- ຕາරາງຜດ CNN
- ຕາරາງ pointer regression

– เปรียบเทียบแบบมีภาพประกอบ

– Discussion

6. Conclusion

– สรุปผล

– ข้อจำกัด

– งานที่จะพัฒนาต่อ

References

– นำรายการในไฟล์ของคุณมาจัดตาม IEEE style

งานของ Nguyen et al. (2025) มีข้อจำกัดหลายอย่าง ซึ่งงานวิจัยของเรานั้นเพื่อแก้ปัญหาดังกล่าว

[Ref] B. N. Van et al., "Water Meter Reading Based on Text Recognition Techniques and Deep Learning," *IEEE Access*, 2025.

ข้อจำกัดของ Nguyen et al. (2025)

ข้อจำกัด 1: อ่านไม่ได้เมื่อภาพเบลอ, มีฝุ่น, มีคราบน้ำ, หรือแสงไม่ดี

ในหน้า Conclusion เขียนว่า

"challenges remain when handling blurry, dusty, or poorly captured images"

ข้อจำกัด 2: ใช้ OCR บนตัวเลขยาว → ล้มเหลวใน half-digit transition

ดู Table 2 / Table 3 บนหน้า 11 จะเห็นว่า OCR หล่ายตัวอ่านผิดในกรณี:

- มุมเอียง
- ครึ่งตัวเลข (เช่นการเปลี่ยนจาก 0 → 1)
- ตัวเลขแดง-ดำปะปน

ข้อจำกัด 3: ไม่รองรับมิเตอร์แบบ Pointer Dial เลย

งานนี้อ่าน เฉพาะตัวเลข ไม่มีการอ่านเข็ม ทำให้ไม่รองรับมิเตอร์แบบหน้าปัดหมุน (pointer)
ซึ่งในหน้า 3-4 ของงานมีแค่ตัวเลขในกรอบ ไม่พูดถึง pointer ตรงไหนเลย

ข้อจำกัด 4: Detection → OCR แบบสองขั้นตอน ทำให้อาจ fail เมื่อ detection ผิดแม้เพียงเล็กน้อย

ดู Figure 2 – Architecture หน้า 5

เข้าใช้ flow = YOLOv8 → แล้ว OCR เลย

ถ้า YOLO crop ไม่ดี OCR จะพัง

ข้อจำกัด 5: ยังไม่มีระบบใช้งานภาคสนาม (Mobile App / Real Deployment)

งานนี้ไม่มี mobile app interaction แบบ real-time

เรา用 WebView + Django Backend ซึ่งเป็นข้อได้เปรียบสำคัญ

งานเรา แก้ปัญหา และข้อจำกัดเน้นได้ยังไงบ้าง

การแก้ข้อจำกัด 1: ภาพเบลอ/มีฝุ่น/สภาพแสงแปรปรวน

งานเรามี preprocessing แบบเฉพาะ เช่น

- adaptive threshold
- denoise
- custom crop pipeline

และ YOLO ของตัวเองที่ดูแล dataset ที่มีภาพจริงจากมิเตอร์ที่ใช้ทั่วไป

การแก้ข้อจำกัด 2: Half-digit transition

มี Digit segmentation แยกหลัก ทำให้รับมือได้ดีกว่า OCR sequence ของงานเดิม

การแก้ข้อจำกัด 3: รองรับ pointer dial

- pointer tip detection
- angle regression
- meter value synthesis (digit + dial)

ซึ่งงานในไฟล์ [ไม่มีไฟล์ชั้นนี้เลย](#) → จุดเด่นชัดที่สุดของงานเรา

การแก้ข้อจำกัด 4: ลดปัญหา YOLO crop ผิด

- multi-stage crop
- additional digit-centering preprocessing
- เงื่อนไขตรวจสอบตัวเลขผิดปกติ (sanity check)

การแก้ข้อจำกัด 5: ใช้งานจริง (deployment)

- Mobile WebView app
- Django backend
- User login / household record
- ส่งผลการอ่านแบบ real-time

ซึ่งงานนี้ไม่มีส่วน “deployment” เลย

(1) Abstract (150–250 คำ)

งานวิจัยนี้นำเสนอระบบอ่านค่ามิเตอร์น้ำอ่านมาโดยอัตโนมัติที่ออกแบบมาเพื่อแก้ไขข้อจำกัดที่พบในวิธีการเดิม ซึ่งมักประสบปัญหาในการประมวลผลภาพที่เบลอ มีผุ่น คราบน้ำ หรือสภาพแสงไม่เพียงพอ อีกทั้งยังกำหนดตำแหน่งตัวเลขได้ไม่แม่นยำ ทำให้เกิดข้อผิดพลาดต่อเนื่องในขั้นตอนรูจัดตัวเลข โดยเฉพาะกรณีตัวเลขเปลี่ยนผ่าน (half-digit transition) นอกจากนี้ วิธีการเดิมส่วนใหญ่ยังไม่รองรับมิเตอร์แบบเข็มซึ่ง แล้วไม่สามารถใช้งานได้จริงในภาคสนาม

ระบบที่พัฒนาขึ้นพسانเทคนิคการตรวจจับด้วย YOLO ร่วมกับโมเดลรูจัดตัวเลขแบบ CNN ที่ประมวลผลแบบแยกตัวเลขรายหลัก เพิ่มความแม่นยำในการอ่านค่าตัวเลขจากภาพจริง พร้อมทั้งพัฒนามोเดล Regression สำหรับประมาณมูลของเข็มซึ่ง ทำให้รองรับมิเตอร์อน้าล็อกที่ใช้หั้งตัวเลขและเข็มในระบบเดียว นอกจากนี้ยังเสริมประสิทธิภาพด้วย preprocessing เนพาะทาง ได้แก่ adaptive thresholding, การลดสัญญาณรบกวน และ multi-stage cropping เพื่อรองรับสภาพภาพถ่ายที่หลากหลาย และผ่านเข้ากับแอปพลิเคชันมือถือแบบ WebView เพื่อใช้งานภาคสนามร่วมกับ Django backend แบบเรียลไทม์

ผลการทดลองแสดงให้เห็นว่าระบบสามารถตรวจจับช่องตัวเลข รูจัดตัวเลข และประเมินมูลของเข็มได้อย่างถูกต้อง แม่นยำในสภาพแวดล้อมจริง ลดข้อผิดพลาดในการอ่านค่าของมนุษย์ และมีศักยภาพสูงต่อการประยุกต์ใช้ในงานจัดเก็บค่าน้ำขององค์กรหรือหน่วยงานสาธารณูปโภค

(2) Related Work

งานวิจัยด้านการอ่านค่ามิเตอร์น้ำและมิเตอร์อ่านมาล็อกสามารถแบ่งออกได้เป็นสามกลุ่มหลัก ได้แก่ (1) งานประมวลผลภาพแบบดั้งเดิม (Traditional Image Processing), (2) งานที่ใช้ Deep Learning ในการตรวจจับและรูจัดตัวเลข, และ (3) งานประมาณการค่ามูลของเข็มมาตรฐาน (Analog Gauge Reading)

2.1 Traditional Image Processing for Meter Reading

แนวทางดั้งเดิมมักใช้วิธีการประมวลผลภาพ เช่น edge detection, thresholding, contour extraction และ morphological operations เพื่อแยกบริเวณตัวเลขหรือเข็มของมิเตอร์ แม้ว่าวิธีการเหล่านี้จะทำงานได้ดีภายใต้สภาพแสงที่เหมาะสม แต่มีข้อจำกัดรุนแรงเมื่อภาพเบลอ มีผุ่น ละอองน้ำ เกาสะท้อน หรือการบิดเบี้ยวจากเลนส์ ทำให้ไม่สามารถใช้งานในสภาพแวดล้อมภาคสนามได้อย่างเชื่อถือได้

2.2 Deep Learning-Based Detection and Digit Recognition

งานสมัยใหม่ก็ไม่ได้ตรวจจับวัตถุ เช่น YOLO, Faster R-CNN หรือ SSD มาใช้ในการระบุตำแหน่งของตัวเลข จากนั้นนำผลลัพธ์ที่ครอบคลุมไปตรวจจับเข้าสู่ OCR หรือ CNN เพื่อรู้จำตัวเลขแบบต่อเนื่อง (sequence-based recognition) อย่างไรก็ตาม การรู้จำตัวเลขแบบยาว (whole sequence OCR) มักให้ผลลัพธ์ไม่ดีในกรณีตัวเลขแค่บางส่วนกำลังเปลี่ยน (half-digit transition) หรือเมื่อช่องตัวเลขมีตำแหน่งเอียง หรือไม่สมมาตร อีกทั้งการทำงานแบบสองขั้นตอนนี้ไวต่อความผิดพลาดมาก หากขั้นตอนตรวจจับผิดเพี้ยง เล็กน้อย OCR จะอ่านผิดทั้งหมดทันที

2.3 Analog Gauge and Pointer Reading

งานในกลุ่มนี้มุ่งเน้นการตรวจจับหน้าปัดและประเมินมุมของเข็ม โดยมักใช้เทคนิค edge detection, Hough Circle Transform และเรขาคณิตพื้นฐานร่วมกับการถ่ายภาพที่มีความชัดเจนสูง อย่างไรก็ตาม งานเหล่านี้ ส่วนใหญ่ยังไม่ถูกประยุกต์ใช้อย่างสมบูรณ์ในระบบมิเตอร์น้ำทั่วไปที่มีทั้งหน้าต่างตัวเลขและเข็มในมิเตอร์เดียว

(3). Research Gap

จากการบทวนงานวิจัยที่ผ่านมา พบร่องว่างที่สำคัญดังนี้

ช่องว่างที่ 1: ความไม่ไวต่อคุณภาพภาพถ่ายของวิธีเดิม

วิธีตรวจจับ-รู้จำแบบเดิมล้มเหลวอย่างเมื่อภาพมีปัญหา เช่น เปลอ แสงน้อย ผุนสะสม หรือแสงสะท้อน ซึ่งพบได้ทั่วไปในมิเตอร์น้ำภาคสนาม

ช่องว่างที่ 2: การรู้จำตัวเลขแบบ sequence OCR ไม่รองรับ half-digit transition

ตัวเลขบนมิเตอร์น้ำจะค่อย ๆ เปลี่ยน ทำให้ตัวเลขบางหลักปรากฏครึ่งหนึ่ง การอ่านแบบ OCR ต่อเนื่องมักอ่านผิด ทำให้ค่ามิเตอร์ผิดทั้งชุด

ช่องว่างที่ 3: การไม่รองรับมิเตอร์แบบเข็มชี้ (Pointer Dial)

งานจำนวนมากเน้นเฉพาะการอ่านตัวเลข แต่มิเตอร์น้ำจำนวนมากยังใช้หน้าปัดเข็ม ระบบจึงต้องรองรับทั้งสองแบบหากจะใช้งานจริง

ช่องว่างที่ 4: ความผิดพลาดจากการครอบของโมเดลตรวจจับ

ในระบบที่พึ่งพา OCR แบบ sequence หาก YOLO ครอบไม่แม่นยำ OCR จะผิดทั้งหมด ระบบที่แข็งแกร่งต้องมี multi-stage crop และ sanity check เพื่อแก้ปัญหานี้

ช่องว่างที่ 5: การขาดระบบที่ใช้งานได้จริงภาคสนาม

งานจำนวนมากยังไม่มีระบบ mobile หรือ backend สำหรับทดสอบใช้งานจริงภายในได้ข้อจำกัดด้านเวลา และสภาพแวดล้อมจริงของเจ้าหน้าที่ตรวจมิเตอร์

*****งานวิจัยนี้จึงเสนอระบบที่มีคุณลักษณะดังนี้

1. **YOLO-based multi-stage detection** รองรับภาพจริงที่มีแสงไม่สตีบร ผ่น และความเบลอ
2. **Digit-wise CNN recognition** ช่วยขจัดปัญหา half-digit transition
3. **Pointer regression model** รองรับมิต่อร์แบบเข็มพร้อมตัวเลข
4. **Preprocessing** เฉพาะทาง เช่น adaptive threshold, denoise, crop alignment
5. **Field-ready deployment** ผ่าน Mobile WebView และ Django backend ให้ความสามารถใช้
งานจริงแบบเรียลไทม์