

GOLF TRACER — แผนการเทรนรอบใหม่

แผนการเทรนรอบใหม่ ตั้งแต่สร้างชุดข้อมูลจนถึงตัดสินใจขึ้นใช้งาน

สร้าง dataset ใหม่จาก label ล่าสุด · fine-tune · ประเมิน · เกณฑ์ตัดสิน — พร้อม
ตัวเฝ้าดูอัตโนมัติทุก 30 นาที

จัดทำโดยทีมพัฒนา Golf Tracer

เครื่องเทรน: go-ai (Mac Studio) · Ultralytics 8.4.106 · PyTorch 2.13.0 · imgsz 960 · device mps

ฐานข้อมูล label: labelhub.db · 11,741 / 11,803 เฟรมบันทึกแล้ว (99.47%)

31 ก.ค. 2569 · v1.0 · เวลาที่ประมาณไว้คำนวณจากรอบที่วัดได้จริง ไม่ใช่การเดา

บทคัดย่อ.

เอกสารนี้เป็นแผนปฏิบัติการสำหรับการเทรนรอบใหม่ ครอบคลุมตั้งแต่การสร้างชุดข้อมูลขึ้นใหม่จากป้ายกำกับล่าสุด ไปจนถึงเกณฑ์ตัดสินว่าจะนำโมเดลขึ้นใช้งานหรือไม่ พร้อมตัวเฝ้าดูอัตโนมัติที่ตรวจสอบสถานะการเทรนทุก 30 นาทีโดยไม่กินโควตาการใช้งาน สามเรื่องที่ต้องรู้ก่อนเริ่ม เรื่องแรก มีงานที่ต้องทำก่อนสร้างชุดข้อมูล ไม่ใช่หลังจากนั้น คือฐานข้อมูลมีคลิปซ้ำสองคู่ และคู่หนึ่งถูกกำหนดให้อยู่คนละฝั่งของการแบ่งชุด ถ้าสร้างชุดข้อมูลตามสภาพปัจจุบัน วิดีโอเดียวกันจะไปอยู่ทั้ง train และ val ทำให้ตัวเลขประเมินวัดการจำแนกรู้จำแนกรายละเอียดและคำสั่งแก้อยู่ในขั้นที่ 1 และต้องได้รับการยืนยันก่อนเพราะเป็นการแก้ไขข้อมูลบนเครื่องที่ให้บริการอยู่ เรื่องที่สอง เวลาที่ต้องใช้ยาวกว่าที่คาด รอบล่าสุดใช้เวลา 313 นาทีสำหรับ 120 รอบบนชุดข้อมูล 4,511 ภาพ ชุดใหม่มีภาพมากกว่าประมาณ 1.5 เท่า จึงประมาณได้ราว 7 ถึง 8 ชั่วโมง แผนนี้จึงออกแบบให้การเทรนทำงานบนเครื่องเทรนโดยตรง ไม่ผูกกับ session ของผู้ควบคุม เรื่องที่สาม ตัวเลขที่ใช้ตัดสินคือ recall ของลูกกอล์ฟ ฐานปัจจุบันอยู่ที่ 0.6648 ไม่ใช่ mAP และไม่ใช่ค่าเฉลี่ยรวม

คำสำคัญ— *training plan, dataset rebuild, data leakage, fine-tuning, model promotion*

สารบัญ

- 1 ภาพรวมและลำดับงาน
- 2 ขั้นที่ 0 — ตรวจสอบความพร้อม
- 3 ขั้นที่ 1 — ลบคลิปซ้ำ (ต้องยืนยันก่อน)
- 4 ขั้นที่ 2 — สร้างชุดข้อมูลใหม่
- 5 ขั้นที่ 3 — ตรวจสอบชุดข้อมูลก่อนเทรน
- 6 ขั้นที่ 4 — fine-tune
- 7 ขั้นที่ 5 — ประเมินผลหลังเทรน
- 8 ขั้นที่ 6 — เกณฑ์ตัดสินว่าจะขึ้นใช้งาน
- 9 ขั้นที่ 7 — ส่งโมเดลเข้าแอป
- 10 ตัวเฝ้าดูอัตโนมัติ
- 11 ความเสี่ยงและสิ่งที่ยังไม่ตัดสิน

1. ภาพรวมและลำดับงาน

แผนนี้มีเจ็ดขั้น ขั้นที่ 1 ต้องได้รับการยืนยันจากผู้มีสิทธิ์ตัดสินใจก่อน เพราะเป็นการแก้ไขข้อมูลบนเครื่องที่ให้บริการอยู่ ขั้นที่ 2 ถึง 7 ทำต่อเนื่องได้เอง

ขั้น	งาน	เวลาประมาณ	สถานะ
0	ตรวจสอบความพร้อมเครื่อง สำรองฐานข้อมูล	5 นาที	ทำได้เลย
1	ลบคลิปป้าออกจากฐานข้อมูล	5 นาที	รอการยืนยัน
2	สร้างชุดข้อมูลใหม่จากป้ายกำกับล่าสุด	10-20 นาที	รอขั้นที่ 1
3	ตรวจสอบชุดข้อมูลก่อนเทรน	1 นาที	รอขั้นที่ 2
4	fine-tune ต่อจากโมเดลที่ใช้งานอยู่	4-8 ชั่วโมง	รอขั้นที่ 3
5	ประเมินผลหลังเทรน	20-30 นาที	ตัวเฝ้าดูจะแจ้งเอง
6	ตัดสินใจว่าจะขึ้นใช้งานหรือไม่	ใช้วิจารณญาณ	รอขั้นที่ 5
7	ส่งโมเดลเข้าแอป	30 นาที	เมื่อผ่านขั้นที่ 6

ตารางที่ 1 ลำดับงานและเวลาที่ประมาณไว้

เวลาในขั้นที่ 4 ไม่ใช้การเดา รอบล่าสุดบนเครื่องเดียวกันใช้ 313 นาทีสำหรับ 120 รอบ บนชุดข้อมูล 4,511 ภาพ ชุดใหม่มีภาพฝึกประมาณ 6,700 ภาพขึ้นไป คิดเป็น 1.49 เท่า จึงประมาณได้ราว 470 นาทีที่ 120 รอบ หรือราว 235 นาทีถ้าใช้ 60 รอบ

2. ขั้นที่ 0 — ตรวจสอบความพร้อม

ตรวจสอบอย่างก่อนแต่ละอะไรทั้งสิ้น ตรวจสอบแล้ว ณ เวลาจัดทำเอกสาร ผลเป็นดังนี้

- ไม่มีการเทรนค้างอยู่ — ตรวจสอบแล้ว ไม่มีกระบวนการ `train.py` ทำงาน และค่าภาระงานเครื่องอยู่ที่ 2.08
- เนื้อที่ว่างเพียงพอ — ตรวจสอบแล้ว เหลือ 1.6 เทราไบต์ ชุดข้อมูลใช้เนื้อที่จริงเพียงเล็กน้อย เพราะเก็บภาพเป็นทางลัด
- สำรองฐานข้อมูลก่อนแก้ — ยังไม่ได้ทำ ต้องทำในขั้นที่ 1 ก่อนลบอะไร

การสำรองฐานข้อมูลไม่ใช่พิธีกรรม ขั้นที่ 1 เป็นการลบแถวข้อมูลแบบย้อนกลับไม่ได้ และฐานข้อมูลนี้คือแหล่งข้อมูลจริงของงาน label ทั้งหมด 9,462 แถวที่คนวาดด้วยมือ

3. ขั้นที่ 1 — ลบคลิปซ้ำ (ต้องยืนยันก่อน)

นี่คือขั้นที่สำคัญที่สุดของแผน และเป็นเหตุผลที่ต้องทำก่อนสร้างชุดข้อมูล ไม่ใช่หลัง ฐานข้อมูลมีคลิปซ้ำสองคู่ ทั้งคู่มีค่าเฟรมอิมแพกต์เท่ากันและมีเฟรมที่บันทึกแล้ว จึงเข้าเงื่อนไขที่เครื่องมือสร้างชุดข้อมูลจะดึงเข้ามาใช้ทั้งคู่ คู่แรกเป็นปัญหาชัดเจน H8G1 ถูกกำหนดให้อยู่ฝั่ง train และ H8G1 copy อยู่ฝั่ง val ทั้งสองมาจากวิดีโอเดียวกันและมีเฟรมอิมแพกต์เดียวกันคือ 60 สำหรับคลิปที่มาจากระบบ label เครื่องมือสร้างชุดข้อมูลใช้ค่าฝั่งที่บันทึกไว้ในฐานข้อมูลโดยตรง ไม่ได้คำนวณใหม่ ผลคือถ้าสร้างชุดข้อมูลตามสภาพนี้ เฟรมที่แทบจะเหมือนกันจะไปอยู่ทั้งสองฝั่ง และตัวเลข val จะสูงขึ้นอย่างไม่มีความหมาย เพราะมันวัดว่าโมเดลจำภาพที่เคยเห็นได้หรือไม่ คู่ที่สองไม่ทำให้ข้อมูลร่วข้ามชุด เพราะอยู่ฝั่ง train ทั้งคู่ แต่ทำให้คลิปนั้นมีน้ำหนักเป็นสองเท่าในการฝึกโดยไม่มีเหตุผล ผลพลอยได้ที่น่าสนใจ เฟรมที่ยังไม่ได้ label ทั้ง 62 เฟรมอยู่ใน H8G1 copy ทั้งหมด การลบสำเนาออกจึงทำให้ความคืบหน้าเป็น 100 เปอร์เซนต์ทันที โดยไม่ต้องลงแรง label เพิ่มแม้เฟรมเดียว และไม่สูญเสียเนื้อหาใด เพราะต้นฉบับยังอยู่ครบ

id	ชื่อคลิป	ฝั่งที่กำหนดไว้	เฟรมอิมแพกต์	เฟรม	บันทึกแล้ว
26	H8G1	train	60	226	226
27	H8G1 copy	val	60	226	164
30	h6g1_3	train	314	61	61
42	h6g1_3 copy	train	314	61	61

ตารางที่ 2 คลิปซ้ำที่ตรวจพบในฐานข้อมูล

คำสั่งที่จะใช้ ยังไม่ได้รัน รออนุมัติ: สำรองฐานข้อมูลด้วย `sqlite3 labelhub.db ".backup labelhub-before-dedupe.db"` แล้วจึงลบด้วย `DELETE FROM clips WHERE id IN (27, 42);` ตารางเฟรมและกล่องผูกไว้ด้วย `ON DELETE CASCADE` จึงถูกลบตามไปเอง ผลที่คาดคือเฟรมหายไป 287 แถว (226 ของ id 27 และ 61 ของ id 42) ซึ่งเป็นเนื้อหาซ้ำทั้งหมด

4. ขั้นที่ 2 — สร้างชุดข้อมูลใหม่

หลังจากข้อมูลสะอาดแล้วจึงสร้างชุดข้อมูล เครื่องมือจะอ่านฐานข้อมูล คัดเฉพาะเฟรมที่มีคนยืนยืน แบ่งชุด ตรวจสอบจำนวนลูกในชุดประเมิน แล้วเขียนโฟลเดอร์ใหม่พร้อมใบสรุปตัวเลข แนะนำให้เขียนลงโฟลเดอร์ใหม่ที่มีวันที่กำกับ ไม่เขียนทับของเดิม เพื่อให้ย้อนกลับได้และเปรียบเทียบได้ว่าชุดข้อมูลเปลี่ยนไปอย่างไร

- ดูผลก่อนเขียนจริงด้วยโหมด `--dry-run`
- เขียนจริงลงโฟลเดอร์ใหม่ด้วย `--out data/datasets/clean-<วันที่>`
- เครื่องมือจะปฏิเสธการสร้างเองถ้าชุด `val` หรือ `test` มีกล่องลูกกอล์ฟไม่ถึง 60 กล่อง ซึ่งเป็นค่าที่ควรปล่อยให้ทำงาน ไม่ควรปิด

สิ่งที่ต้องดูในผลลัพธ์ จำนวนภาพต่อชุด จำนวนกล่องแยกตามคลาส และจำนวนลูกช่วงลอยของชุด `val` และ `test` ถ้าจำนวนภาพรวมน้อยกว่าชุดเดิมอย่างมีนัยสำคัญ ให้หยุดตรวจสอบ เพราะแปลว่ามีเฟรมที่เคยผ่านเกณฑ์แล้วตกเกณฑ์ไป

5. ขั้นที่ 3 — ตรวจสอบชุดข้อมูลก่อนเทรน

รันเครื่องมือตรวจสอบ ใช้เวลาไม่ถึงนาที และตรวจเจ็ตรื่องที่เคยผิดพลาดมาแล้วจริงในโครงการนี้ได้แก่ ทางลัดที่เข้าไปในที่ที่ไม่มีอยู่ ตำแหน่งฐานของชุดข้อมูลที่ยังชี้ไปเครื่องอื่น จำนวนคลาสที่ยุบเหลือหนึ่ง ชุดประเมินที่ไม่มีลูก ภาพที่ไม่มีป้ายกำกับ ป้ายกำกับที่ไม่มีภาพ และค่าพิกัดที่ไม่อยู่ในช่วงศูนย์ถึงหนึ่ง ถ้าเครื่องมือคืนค่าไม่เป็นศูนย์ ห้ามเริ่มเทรน

จุดที่พลาดง่ายและตรวจไม่เจอด้วยตาเปล่า หลังสร้างชุดข้อมูลใหม่ บรรทัดตำแหน่งฐานในไฟล์กำหนดชุดข้อมูลจะชี้ไปที่โฟลเดอร์ที่สร้างมันขึ้นมา ถ้าย้ายชุดข้อมูลข้ามเครื่องต้องแก้บรรทัดนี้ก่อน เพราะ Ultralytics อ้างอิงตำแหน่งของ `train`, `val` และ `test` จากบรรทัดนี้ทั้งหมด

6. ขั้นที่ 4 — fine-tune

การเทรนเริ่มจากโมเดลที่ใช้งานอยู่ ไม่ได้เริ่มจากศูนย์ เหตุผลคือชุดข้อมูลที่มีป้ายกำกับยังเล็กเมื่อเทียบกับงานตรวจจับทั่วไป หน้าที่ของมันคือสอนกรณีโมเดลปัจจุบันพลาด ได้แก่ ลูกเบลอจากการเคลื่อนที่เร็ว ลูกขาวบนท้องฟ้าที่มีเมฆ และลูกที่อยู่ไกลจนเล็กมาก ไม่ใช่สอนใหม่ว่าลูก

กอล์ฟหน้าตาเป็นอย่างไร ต้องส่งตัวเลือกระบุชุดข้อมูลอย่างชัดเจนทุกครั้ง เพราะค่าตั้งต้นของเครื่องมือเทรนขึ้นไปทีโพล์กำหนดชุดข้อมูลที่ประกาศไว้เพียงคลาสเดียว ซึ่งจะย้ายกล่องลูกกอล์ฟทั้งหมดไปเป็นคลาสหัวไม้อย่างเงียบ ๆ ความผิดพลาดนี้เคยทำให้เสียเวลาเทรนไปหนึ่งชั่วโมงแล้วครั้งหนึ่ง

ตัวเลือก	ค่าที่จะใช้	เหตุผล
ชุดข้อมูล	โพลเดอร์ที่สร้างในชั้นที่ 2	ส่งชัดเจน ไม่พึ่งค่าตั้งต้น
โมเดลเริ่มต้น	yolo_birdiex_2c_1.pt	โมเดลที่ใช้งานอยู่ และเป็นตัวเปรียบเทียบเดียวกับที่เครื่องมือประเมินใช้
รอบการฝึก	120	รอบล่าสุดที่ให้ผลดีที่สุดใช้ค่านี้ ถ้าเวลาจำกัดให้ใช้ 60 แล้วดูแนวโน้ม
ขนาดภาพ	960	ต้องตรงกับที่แอปใช้ตอนอนุมาน ถ้าไม่ตรงผลจะเพี้ยน
หยุดเมื่อไม่ดีขึ้น	8 รอบ	กันการเทรนต่อไปเปล่า ๆ เมื่อผลนิ่งแล้ว
อุปกรณ์	mps	เร่งด้วยหน่วยประมวลผลกราฟิกของ Apple Silicon

ตารางที่ 3 ค่าที่จะใช้ และเหตุผล

ต้องรันแบบที่ไม่ตายเมื่อปิดการเชื่อมต่อ การเทรนใช้เวลา 4 ถึง 8 ชั่วโมง ถ้ารันผ่าน ssh ตรง ๆ แล้วการเชื่อมต่อหลุด กระบวนการจะถูกฆ่าไปด้วย ให้รันภายใต้ nohup หรือ tmux เสมอ และให้ตัวเฝ้าดูเป็นผู้อย่างงานความคืบหน้าแทนการเปิดหน้าต่างค้างไว้

7. ขั้นที่ 5 — ประเมินผลหลังเทรน

มีการประเมินสามขั้น และควรรันทั้งสามก่อนตัดสินใจ ตัวเฝ้าดูอัตโนมัติจะแจ้งเมื่อการเทรนจบพร้อมบอกไฟล์ผลที่เกิดขึ้นใหม่

- ตัวเลขบนชุด val เทียบกับโมเดลเดิมบนชุดแบ่งเดียวกัน เครื่องมือพิมพ์ให้เองทั้งสองฝั่ง
- ผลบนคลิปที่ไม่เคยอยู่ในชุดฝึก ค่าตั้งต้นคือหลุม H2G1, H4G1, H4G2 และ H8G1 ตัวเลข val ดีขึ้นแต่คลิปจริงแย่งคือสัญญาณให้หยุด
- คะแนนบนเฟรมที่คนวาดไว้ เทียบระหว่างโมเดลเก่ากับใหม่

ระวางการอ่านส่วนต่างที่เล็กเกินกว่าชุดประเมินจะรองรับ รอบก่อนหน้าใช้ชุด val ขนาด 53 ภาพจาก 2 คลิป และรายงานส่วนต่างระดับ 0.0004 ซึ่งเป็นความผันผวน ไม่ใช่ผลของการเทรน ชุดใหม่มี val หลายร้อยภาพ ทำให้ส่วนต่างเชื่อถือได้มากขึ้น แต่หลักการเดิมยังใช้ คือส่วนต่างที่เล็กกว่าความผันผวนของชุดประเมินไม่ใช่หลักฐาน

8. ขั้นที่ 6 — เกณฑ์ตัดสินว่าจะขึ้นใช้งาน

ไม่มีการเลื่อนโมเดลขึ้นใช้งานอัตโนมัติ และไม่ควรมี ตัวเลขฐานที่ต้องเอาชนะคือของโมเดลล่าสุดที่ใช้งานอยู่ เหตุผลที่ recall ของลูกเป็นตัวตัดสิน ตรงไปตรงมา ลูกที่โมเดลมองไม่เห็นในเฟรมหนึ่งคือรอยขาดหนึ่งจุดบนเส้นวิถีที่ผู้ใช้เห็นบนวิดีโอ ขณะที่ผลบวกถ่วงมักถูกตัวติดตามกรองออกด้วยเกณฑ์การเคลื่อนที่ ช่องว่างระหว่าง recall ของหัวไม้ที่ 0.819 กับของลูกที่ 0.665 คือประมาณ 15 จุด และเป็นเป้าหมายที่ชัดที่สุดของรอบนี้

ตัวชี้วัด	ฐานปัจจุบัน	เกณฑ์ผ่าน
recall ลูกกอล์ฟ	0.6648	ต้องสูงขึ้นอย่างมีนัยสำคัญ — นี่เป็นตัวตัดสินหลัก
precision ลูกกอล์ฟ	0.8924	ลดลงเล็กน้อยยอมรับได้ ตัวติดตามกรองผลบวกถ่วงได้บางส่วน
recall หัวไม้	0.8191	ไม่ควรถอยลงอย่างเห็นได้ชัด
mAP50	0.7956	ใช้ดูภาพรวม ไม่ใช่ตัดสินเดี่ยว
ผลบนคลิปที่ไม่เคยเห็น	ผลของโมเดลเดิม	ต้องไม่ถอยลง แม้ตัวเลข val จะดีขึ้น

ตารางที่ 4 ฐานปัจจุบัน และเกณฑ์ตัดสิน

ถ้าผลไม่ผ่านเกณฑ์ อย่าฝืนเลื่อนขึ้นใช้งาน ทางเลือกที่มีเหตุผลรองรับคือเพิ่มข้อมูลที่เจาะกรณีที่พลาดสามกรณีคือลูกเบลอจากความเร็ว ลูกขวบนเมฆ และลูกไกลขนาดเล็ก การเพิ่มข้อมูลเจาะจุดให้ผลมากกว่าการเพิ่มเฟรมทั่วไปในจำนวนเท่ากัน

9. ขั้นที่ 7 — ส่งโมเดลเข้าแอป

เมื่อผ่านเกณฑ์แล้ว เครื่องมือเทรนส่งออกไฟล์ ONNX ให้เองพร้อมกับไฟล์น้ำหนัก การนำเข้าแอปต้องรักษาสองอย่างให้ตรงกับตอนเทรน คือขนาดภาพที่ 960 และลำดับคลาสที่ 0 คือหัวไม้ 1 คือลูกกอล์ฟ ถ้าอย่างใดอย่างหนึ่งไม่ตรง ผลบนแอปจะเพี้ยนโดยที่ตัวเลขตอนเทรนยังดูดีอยู่

ตรวจสอบก่อนปล่อย ให้ลองบนคลิปจริงของผู้ใช้อย่างน้อยหนึ่งคลิปที่ถ่ายกลางคืน เพราะกรณีไฟสปอตไลท์เป็นจุดที่ค่าความมั่นใจของโมเดลแยกลูกกับดวงไฟไม่ออก และเป็นข้อจำกัดที่ยังไม่ได้แก้ในระดับโมเดล

10. ตัวเฝ้าคุ้อัตโนมัติ

ติดตั้งแล้วและทำงานอยู่ ตรวจสอบทุก 30 นาที โดยไม่กินโควตาการใช้งานของผู้ควบคุมเลย เพราะการตรวจแต่ละรอบเป็นเพียงการเชื่อมต่อเข้าเครื่องเทรนแล้วอ่านสถานะ ไม่ได้เรียกใช้ผู้ช่วยปัญญาประดิษฐ์ สิ่งที่ทำแต่ละรอบคือ ตรวจสอบว่ากระบวนการเทรนยังทำงานอยู่หรือไม่ อ่านว่าถึงรอบที่เท่าไรแล้ว บันทึกลงแฟ้มบันทึก และอ่านสถานะโควตาที่เหลือ เมื่อพบว่าสถานะเปลี่ยนจากกำลังทำงานเป็นหยุดแล้ว มันจะเขียนบันทึกสรุปพร้อมตัวเลขผล แจ้งเตือนบนหน้าจอ และเรียกผู้ช่วยขึ้นมาทำงานต่อ หนึ่งครั้งเท่านั้น ไม่ใช่ทุกรอบ

ต้องการ	คำสั่ง
ดูสถานะและผลการตรวจล่าสุด	<code>bash scripts/install_watcher.sh --status</code>
ตรวจเดี๋ยวนี้นี้รอบ	<code>bash scripts/watch_training.sh --once</code>
ติดตั้งใหม่หลังแก้สคริปต์	<code>bash scripts/install_watcher.sh</code>
เลิกใช้	<code>bash scripts/install_watcher.sh --remove</code>

ตารางที่ 5 คำสั่งจัดการตัวเฝ้าดู

ข้อจำกัดที่ควรรู้ ตัวเฝ้าดูทำงานที่ระดับระบบปฏิบัติการ จึงอยู่รอดแม้ session ของผู้ควบคุมจบลง โควตาหมด เครื่องหลับ หรือรีสตาร์ท แต่ต้องเปิดเครื่องไว้ และไฟล์สคริปต์ที่ระบบเรียกใช้ต้องอยู่นอกโฟลเดอร์เดสก์ท็อป เพราะระบบปฏิบัติการปกป้องโฟลเดอร์นั้นไว้และงานเบื้องหลังจะถูกปฏิเสธการเข้าถึง

11. ความเสี่ยงและสิ่งที่ยังไม่ตัดสินใจ

ส่วนนี้ระบุสิ่งที่อาจผิดพลาดและสิ่งที่ยังต้องการการตัดสินใจ เพื่อให้ผู้อ่านประเมินได้เองว่าแผนนี้พร้อมเดินหน้าเพียงใด สิ่งที่ยังไม่ตัดสินใจและต้องการคำยืนยัน มีข้อเดียวคือการลบคลิปซ้ำในชั้นที่ 1 เพราะเป็นการแก้ไขข้อมูลบนเครื่องที่ให้บริการอยู่ และย้อนกลับไม่ได้หากไม่ได้สำรองไว้ก่อนคำสั่งเตรียมไว้แล้วทั้งการสำรองและการลบ แต่ยังไม่ได้รับ สิ่งที่เอกสารนี้ยังไม่ได้ทำ ยังไม่ได้ทดลองเทรนเพื่อยืนยันว่าสภาพแวดล้อมบนเครื่องเทรนพร้อม แม้จะเป็นเครื่องเดิมที่เทรนรอบก่อนสำเร็จ และตัวเลขเวลาที่ประมาณไว้คำนวณจากสัดส่วนจำนวนภาพ ซึ่งเป็นการประมาณเชิงเส้นที่อาจไม่ตรงหากขนาดภาพหรือการโหลดข้อมูลต่างจากเดิม

ความเสี่ยง	ผลกระทบ	การรับมือ
สร้างชุดข้อมูลก่อนลบคลิปซ้ำ	วิดีโอเดียวกันอยู่ทั้ง train และ val ตัวเลขประเมินไร้ความหมาย	ทำชั้นที่ 1 ก่อนชั้นที่ 2 เสมอ
ลืมส่งตัวเลือกระบุชุดข้อมูล	เสียเวลาเทรนทั้งรอบ และรายงานเฉพาะตัวเลขหัวไม้	ใช้สคริปต์ห่อที่ส่งค่าให้เสมอ
การเชื่อมต่อหลุดระหว่างเทรน	กระบวนการถูกฆ่า เสียเวลาทั้งหมดที่ผ่านมา	รันภายใต้ nohup หรือ tmux
ตัดสินใจจากส่วนต่างที่เล็กเกินไป	เลื่อนโมเดลที่ไม่ได้ดีขึ้นจริงขึ้นใช้งาน	ดู recall ของลูกและผลบนคลิปที่ไม่เคยเห็นประกอบ
ป้ายกำกับผ่านสายตาคนเดียว	ความผิดพลาดของป้ายกำกับกลายเป็นเพดานของโมเดล	สุ่มรีวิวซ้ำในชุด val และ test ก่อน ปัจจุบันรีวิวเพียง 0.9 เปอร์เซนต์

ตารางที่ 6 ความเสี่ยงที่ระบุได้ พร้อมการรับมือ

เอกสารอ้างอิง

- [1] ฐานข้อมูล label: go-ai · ~/labelhub/data/labelhub.db · ตรวจสอบอ่านอย่างเดี่ยว 31 ก.ค. 2569
- [2] เครื่องมือสร้างชุดข้อมูล: build_dataset.py · เกณฑ์ MIN_BALLS_PER_EVAL_SPLIT = 60
- [3] เครื่องมือเทรน: train.py · Ultralytics 8.4.106 · PyTorch 2.13.0 · imgsz 960
- [4] ฐานเวลาที่ใช้ประมาณ: runs/ft-20260728-004149 ใช้ 313 นาที · 120 รอบ · 4,511 ภาพฝึก
- [5] ตัวเลขฐานของโมเดล: models/yolo_birdiex_2c_20260728-004149.json
- [6] คู่มือการเทรนฉบับเต็ม: docs/golf-tracer-training-handbook.th.pdf