บทความวิจัย

การตรวจจับสิ่งของภายในบ้านด้วยการเรียนรู้เชิงลึก

Household Detection Using Deep Learning

กิตติธัช ด่านชัยภูมิพัฒน์, พัชรพล โพธิคำ, พรทิวา ปะวะระ
สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
63011212026@msu.ac.th, 63011212167@msu.ac.th, pornntiwa.p@msu.ac.th

บทคัดย่อ

ในปัจจุบันมีผู้พิการทางสายตาที่มีปัญหาในการหาสิ่งของภายในบ้านและอาจจะไม่มีคนค่อยช่วยเหลือในการหาสิ่งของ ดังนั้นการระบุตำแหน่งสิ่งของภายในบ้านจึงมีประโยชน์ต่อผู้พิการทางสายตาที่ต้องการหาสิ่งของภายในบ้าน ยกตัวอย่างเช่น ต้องการหาสิ่งของภายในบ้านและต้องการหาสิ่งของที่มีลักษณะคล้ายกัน เป็นต้น

โครงงานฉบับนี้นำเสนอการสร้างแอปพลิเคชันระบุสิ่งของภายในบ้านเพื่อช่วยเหลือผู้พิการทางสายตาในการช่วยหาสิ่งของสะดวกมากขึ้น โดยในงานนี้จะอยู่ในลักษณะของการระบุสิ่งของภายในบ้าน โดยอาศัยการการตรวจจับวัตถุ(Object Detection) กับ YOLO(You only look once) และใช้ Speech Recognition ในการวิเคราะห์เสียงพูดของผู้ใช้ในการทำงาน

1. บทนำ

ในปัจจุบันมีผู้พิการทางสายตาตั้งแต่กำเนิดและจากอุบัติเหตุ การสูญเสียการมองเห็นนั้นเป็นอุปสรรคในการดำเนินชีวิตอย่างมาก เช่น การต้องการหาสิ่งของภายในบ้าน หรือสิ่งของที่อยู่บนชั้นวาง แต่ไม่สามารถระบุได้ว่าอยู่ส่วนไหน และถ้าหากต้องการหาสิ่งของที่อยู่รวมกับของชิ้นอื่นภายในกล่องอาจจะต้องใช้เวลานาน ดังนั้นการระบุสิ่งของภายในบ้านจึงเป็นประโยชน์ต่อผู้พิการทางสายตาที่อาจจะไม่มีคนค่อยช่วยเหลือในการหาสิ่งของภายในบ้าน

เนื่องจากของแต่ละชิ้นมีลักษณะเฉพาะ แต่มีบางชิ้นที่ลักษณะเหมือนหรือคล้ายกันแต่คุณสมบัติเฉพาะ เช่น ด้ายถักที่มีสีต่างกัน จึงเป็นเรื่องยากที่ผู้พิการทางสายตาจะแยกด้วยการสัมผัส เราจึงเล็งเห็นปัญหานี้จึงอยากนำปัญญาประดิษฐ์ (Artificial Intelligence) และการเรียนรู้แบบอัตโนมัติ (Deep Learning) ด้วยอัลกอริทึมYOLO(You only look once) มาจำแนกประเภทสิ่งของใช้ภายในบ้านและคุณสมบัติเฉพาะสิ่งของ พร้อมยังนำ Speech Recognition มาช่วยสั่งการผ่านทางเสียงเพื่อความสะดวกในการใช้งาน

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Objective Detection

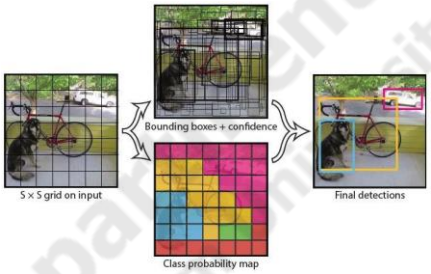
"เทคโนโลยีตรวจจับวัตถุ" (Object detection) คือ หนึ่งในฟีเจอร์หลักของ AI (Artificial Intelligence) ที่ใช้กับกล้องวงจรปิด สามารถค้นหาสิ่งของโดยใช้ AI มาวิเคราะห์ข้อมูล จากการมองเห็นของคอมพิวเตอร์ (Computer Vision) และการประมวลผลภาพ Image Processing) เพื่อตรวจจับวัตถุที่อยู่ในรูปหรือวิดีโอ เช่น มนุษย์ สัตว์ สิ่งของ รถยนต์ อาคาร และวัตถุอื่น ๆ ที่อยู่ในรูปภาพ หรือวิดีโอ

โดยตามหลักแล้วก่อนที่จะพัฒนามาเป็นเทคโนโลยีตรวจจับวัตถุ (Object detection) จะต้องผ่านการจัดหมวดของวัตถุ (Object Classification) มาก่อน โดยที่การจัดหมวดของวัตถุจะเป็นการจัดหมวดหมู่ของรูปภาพว่ารูปภาพนั้นคือภาพอะไร แต่เทคโนโลยีตรวจจับวัตถุจะเป็นการระบุเลยว่า ในรูปภาพนั้นมีวัตถุอะไรบ้าง ซึ่งจุดนี้จะต้องอาศัยการทำงานของ AI เข้ามาช่วยในการวิเคราะห์ข้อมูลด้วยเช่นกัน

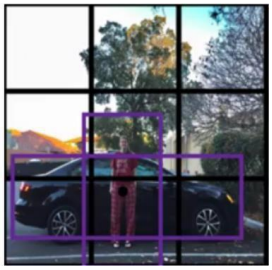
2.1.2 YOLO (You Only Look Once)

คือสถาปัตยกรรมที่ทาง ultralytics ได้ออกแบบไว้เพื่อทำ Image Detection ได้อย่างรวดเร็วและมีประสิทธิภาพ

ขั้นตอนการทำงานของ YOLO จะเป็นในลักษณะการแบ่งรูปภาพเป็นส่วนๆ หรือกริด grid) จากนั้นทำการเลื่อนการคำนวนไปทีละจุดตามที่แบ่งกริดไว้ (Sliding Windows) พร้อมกับคำนวณจำแนกหาว่าวัตถุจะมีอยู่จริงหรือไม่จากความน่าจะเป็นของวัตถุที่ปรากฏในพื้นที่นั้นๆด้วยกระบวนการ Intersection over Union (IoU) เป็นการวัดประสิทธิภาพของโมเดล



ภาพประกอบที่ 1 การทำงานของอัลกอริทึม YOLO



ภาพประกอบที่ 2 การทำงาน Anchor box

ภาพประกอบที่ 3 การทำงาน Non-max Suppression

ในการทำนายกรอบล้อมวัตถุจะได้ข้อมูลเป็นชุดข้อมูล ประเภทอาร์เรย์ ซึ่งประกอบด้วยข้อมูลการมีอยู่จริงของวัตถุ ตำแหน่งและขนาดของกรอบล้อมวัตถุ และ

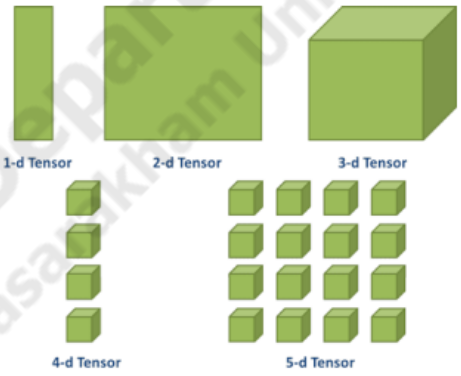
ชนิด ของวัตถุ กรณีถ้าในแต่ละกริดมีวัตถุมากกว่าหนึ่งอย่าง YOLO มีกระบวนการ Anchor box เพื่อ แก้ปัญหาดังกล่าว แนวคิดคือการสร้าง Anchor box ใน รูปทรงต่างๆ และคำนวณใหม่ เพื่อให้ได้ผลลัพธ์ที่ครอบคลุม และ แม่นยำมากขึ้น เมื่อระบบสร้างกรอบล้อมวัตถุจนหมดแล้ว จะเข้าสู่กระบวนการ Non-max Suppression ซึ่งขั้นตอนนี้จะเป็นการลดจำนวนกรอบล้อม วัตถุที่เป็นวัตถุเดียวกันถ้ากล่องใดมีค่านี้สูง (เกิน Threshold ของ iou ที่กำหนด) แสดงว่ามันคือ Object เดียวกัน

2.1.3 Flutter

Flutter คือ Framework ที่ใช้สร้าง UI สำหรับ mobile application ที่สามารถทำงานข้ามแพลตฟอร์มได้ทั้ง iOS และ Android ในเวลาเดียวกัน โดยภาษาที่ใช้ใน Flutter นั้นจะเป็นภาษา dart ซึ่งถูกพัฒนาโดย Google และที่สำคัญคือเป็น open source ที่สามารถใช้งานได้แบบ

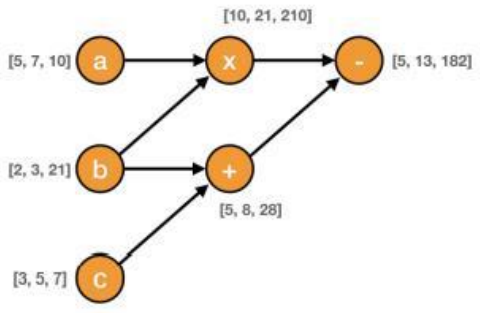
2.1.4 TensorFlow

TensorFlow เป็นไลบรารีโอเพนซอร์ซ (Library open source) ได้รับการ พัฒนาโดยทีม Google Brain ของบริษัท Google ได้ทำการเปิดตัวเมื่อวันที่ 11 กุมภาพันธ์ 2017 สามารถทำงานบน CPU และ GPUs รองรับระบบปฏิบัติการ Linux, macOS, Windows และ Android เป็นการทำงานสำหรับการคำนวณเชิงตัวเลข เป็นการสร้างกราฟกระแสข้อมูล และโครงสร้าง เพื่อกำหนดการทำงานของข้อมูลผ่านกราฟ โดยรับข้อมูลโหนดเข้ามาเป็นอาร์เรย์หลายมิติที่เรียกว่าเทนเซอร์ (tensor) ไหลผ่านการเชื่อมต่อการทำงานและแสดงผลข้อมูลแต่ละครั้ง 15 ส่วนประกอบเบื้องต้นของ TensorFlow คือ เทนเซอร์ เป็นเวกเตอร์(vector) หรือเมตริกซ์ (Metrix) ของมิติ n ที่แสดง ข้อมูลประเภทเดียวกัน เป็นการคำนวณทั้งหมดที่เกี่ยวข้องกับเมตริกซ์ ที่มีการดำเนินงาน ในกราฟ



ภาพประกอบที่ 4 รูปร่างเทนเซอร์

กราฟ เป็นการคำนวณตัวดำเนินการทางคณิตศาสตร์ (operator) เช่น บวก ลบ คูณ หาร เป็นต้น ทั้งหมดภายในกราฟที่เรียกว่าโหนด (node) เป็นการ เชื่อมต่อเทนเซอร์เข้าด้วยกัน



ภาพประกอบที่ 5 กราฟที่มีการเชื่อมเทนเซอร์เข้าด้วยกัน

2.1.5 Flutter Text to Speech (TTS)

Flutter Text to Speech (TTS) คือเทคโนโลยีที่ช่วยแปลงข้อความเป็นเสียงในแอปพลิเคชันที่พัฒนาด้วย Flutter framework ซึ่งเป็นเครื่องมือสำหรับสร้างแอปพลิเคชันมัลติแพลตฟอร์มที่สามารถทำงานได้ทั้งบน Android และ iOS ด้วยโค้ดเดียวกัน การใช้งาน Flutter TTS ช่วยให้คุณสามารถให้แอปพลิเคชันของคุณอ่านข้อความออกเสียงหรือพูดในภาษาอื่น ๆ ได้อย่างง่ายดาย

2.1.6 Flutter Speech to Text (STT)

Flutter Speech to Text (STT) เป็นเทคโนโลยีที่อนุญาตให้แอปพลิเคชันใน Flutter รับข้อมูลเสียงจากผู้ใช้และแปลงเสียงนั้นเป็นข้อความ เพื่อให้แอปพลิเคชันสามารถเข้าใจและประมวลผลข้อมูลที่เสียงบอกได้

3.วิธีดำเนินงานวิจัย

3.1 กรอบการดำเนินงาน

3.1.1 Deep learning

ในส่วนของ Deep learning จะเป็นขั้นตอนในการสร้าง House model ที่ใช้ในการ classification ที่จะนำไปใช้ใน mobile application ก่อนที่จะสร้าง House model จะต้องมี dataset ของสิ่งของก่อน โดย dataset ที่ได้จะรวบรวมจากหลากหลายเว็บไซต์ หลังจากนั้นจะนำ dataset ที่ได้ไปใช้ในการ train model โดยใช้ pre-train model ของ YOLOv5 พอ train model ได้ตามที่ต้องการจากนั้นจะบันทึก model เป็นไฟล์ tflite เพื่อนำไปใช้บน moblie application

3.1.2 Mobile Application

จากภาพประกอบที่ ในส่วนของ Mobile application โดยขั้นตอนในการใช้งาน House model คือผู้ใช้ต้องนำมือถือส่องไปรอบๆ บริเวณห้อง จากนั้น mobile application จะรับภาพจากกล้องมือถือ model จะทำการ classification และส่งผลลัพธ์กลับมาให้ผู้ใช้

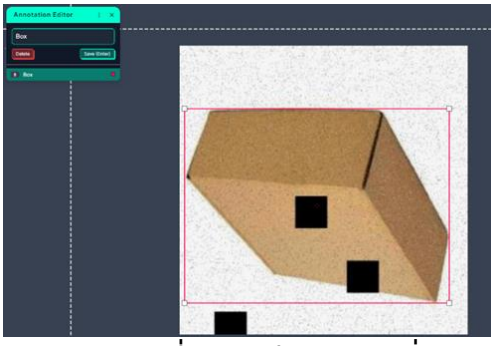
3.2 การเตรียมชุดข้อมูล (Dataset)

3.2.1 ชุดข้อมูล

โครงงานนี้ได้ใช้ชุดข้อมูลสิ่งของ 24 ชนิด ในการสร้างโมเดล ซึ่งจะมีรูปภาพทั้งหมด 8,959 รูป ประกอบด้วยชุดข้อมูลสำหรับการ Train 6,268 รูป Valid 1,788 รูป และ Test 903 รูป ดังตารางที่ 3.1 รูปภาพ Input ทั้งหมดจะถูกปรับขนาด 640*640 รูปภาพทั้งหมดได้ทำการรวบรวมมาจาก Roboflow

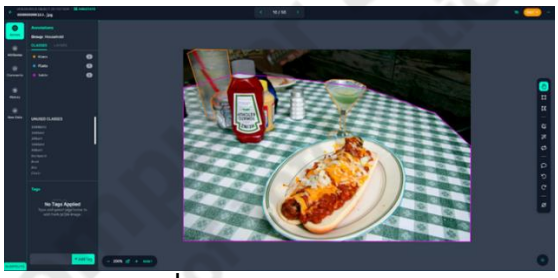
3.2.2 Roboflow

Roboflow เป็น Computer Vision Developer Framework สำหรับใช้จัดเก็บเตรียมชุดข้อมูล และสร้างแบบจำลองต่างๆ ที่สามารถใช้งานผ่าน web browser ได้



ภาพประกอบที่ 6 การตีกรอบภาพสิ่งของ

การตีกรอบแบบ Polygon annotation ช่วยให้ระบบตรวจจับวัตถุได้อย่างแม่นยำมากขึ้น เนื่องจากสามารถระบุรูปร่างและขอบเขตของวัตถุได้อย่างละเอียด โดยการนำตำแหน่ง x และ y แต่ละจุดมาเชื่อมกันตามลำดับในการกำหนดขอบเขตของวัตถุในภาพ เช่น x1y1, x2y2, x3y3



ภาพประกอบที่ 7 การตีกรอบแบบ Polygon annotation

dataset ที่เตรียมใน Roboflow มาใช้เมื่อเรา generate รูปภาพเสร็จทำการ Export ไฟล์เราจะได้สองไฟล์คือ images และ labels

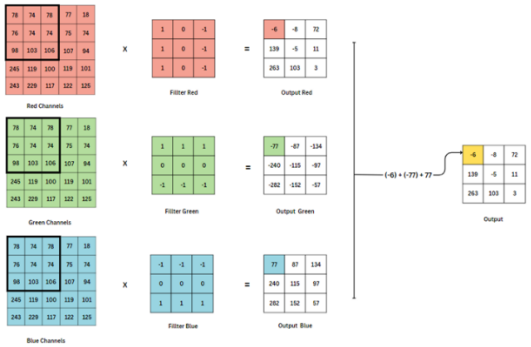
3.3 ขั้นตอนการทำงานของ YOLO (You Only Look Once)

3.3.1 การเตรียมอินพุต

อัลกอริธึมจะใช้รูปภาพเป็นอินพุต รูปภาพจะถูกปรับขนาดให้เป็นขนาดคงที่โดยยังคงอัตราส่วนไว้ รูปภาพที่ปรับขนาดแล้วเหล่านี้จะถูกป้อนเข้าสู่โมเดล YOLOv5 โดยรูปภาพของเรามีอินพุตมีขนาด 640 x 640 x 3

3.3.2 การทำงานของ Convolution Layer

คือการสกัดคุณลักษณะของรูปภาพอินพุต โดยรูปภาพจะถูกแบ่งเป็นแมทริกซ์ อีกเมทริกซ์คือเซตของพารามิเตอร์ที่สามารถเรียนรู้ได้ที่เรียกว่า Filter โดยการทำ convolution จะใช้ filter มาสแกนภาพเพื่อทำการแยกองค์ประกอบ โดยวิธีดำเนินการ เริ่มจากนำ filter สแกนไปบนภาพ และทำการคำนวณโดยการนำเอาตำแหน่งที่ตรงกันของภาพต้นฉบับและ filter มาคูณกัน แล้วจึงนำผลรวมของทุกตำแหน่งมาบวกกัน (ตำแหน่งที่ 1 + ตำแหน่งที่ 2 + ... + ตำแหน่งที่ n)



ภาพประกอบที่ 8 การทำ Convolution ขนาด 5 x 5 และ filter ขนาด 3 x 3

3.3.3 ReLU (Rectified Linear Unit)

ผลลัพธ์ที่ได้จากการทำ Convolution ในแต่ละตำแหน่งจะแปลงค่าด้วยฟังก์ชัน ReLU ที่เป็นการแปลงแบบไม่เป็นเชิงเส้น เพื่อความง่ายในการคำนวณและประเมิน ประสิทธิภาพผลลัพธ์คือการลบค่าลบทั้งหมดออกจาก Convolution ค่าบวกทั้งหมด ยังคงเหมือนเดิม แต่ค่าลบทั้งหมดจะเปลี่ยนเป็นศูนย์ เนื่องจากค่าลบอาจบ่งบอกถึงข้อมูลที่ไม่เกี่ยวข้องหรือไม่สนใจในการแยกแยะคุณสมบัตินั้น ๆ

3.3.4 Max Pooling Layer

Max Pooling ช่วยให้โมเดลความสามารถในการระบุคุณสมบัติที่สำคัญมากยิ่งขึ้น เนื่องจากมันจะเลือกค่าที่มากที่สุดในบริเวณของ filter ซึ่งหมายความว่ามันจะเน้นการตอบสนองต่อลักษณะที่สำคัญ และช่วยให้โมเดลมีความเป็นอิสระต่อขนาดของข้อมูลนำเข้า เนื่องจากการขยับ filter ด้วยค่า stride ทำให้ขนาดข้อมูลผ่านไปอาจลดลง แต่ Max Pooling ยังคงรักษาข้อมูลที่สำคัญและลดขนาดข้อมูลอย่างมีประสิทธิภาพ

หลักการทำงานของ Max Pooling Layer ทำการกำหนดขนาดของ filter และกำหนดค่า stride จากนั้นวางฟิลเตอร์ที่มุมซ้ายสุดแล้วนำค่าที่มากที่สุดที่อยู่ในบริเวณของ filter มาเป็นค่าของรูปใหม่ จากนั้นขยับ filter ตามค่า stride เช่นค่า stride = 1 ก็ให้ขยับ filter ทีละ 1 ช่อง

3.3.5 Flatten

Flatten คือการทำให้ค่าข้อมูลที่เป็นภาพประกอบที่ได้จาก Max pooling 2 มิติ กลายเป็นข้อมูล Vector แบบ 1 มิติ เพื่อเป็นการเตรียมข้อมูลก่อนส่งเข้า Fully-Connected Layer เนื่องจาก Neuron ใน Fully-Connected Layer

3.3.7 Softmax

Softmax function เป็นฟังก์ชันที่จะแปลงคะแนนผลลัพธ์ของชั้นสุดท้ายในแต่ละโหนดของโครงข่ายประสาทเทียมเป็นค่าความน่าจะเป็นตามสัดส่วนของคลาส คลาสไหนที่มีค่าความน่าจะเป็นมากที่สุดโมเดลจะเลือกคลาสนั้นเป็นคำตอบ

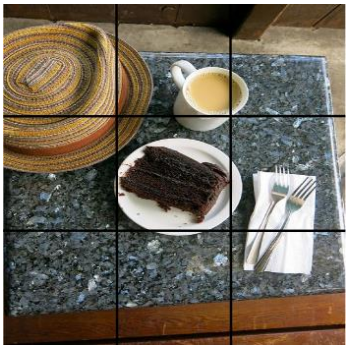
3.4 Detection

3.4.1 ตัวอย่างภาพที่มีวัตถุชนิดเดียวแสดงขั้นตอนการคำนวณ

กำหนดขนาดภาพ 640 x 640 และแบ่งภาพเป็นตาราง 3 x 3

3.4.1.1 กำหนดค่า y

เพื่อเก็บคำตอบ กำหนดค่า y เพื่อเก็บคำตอบ มีค่าเท่ากับ S x S x A x (5 + จำนวน class) โดยให้ A=2 ซึ่ง A คือ จำนวน anchor ภายในแต่ละ grid ในตัวอย่างกำหนดจำนวน class เท่ากับ 1 คือ Plate ดังนั้น y=3x3x2x(5+1) หรือ 3x3x2x6



ภาพประกอบที่ 9 ตัวอย่างการแบ่งตาราง 3 x 3

$$y = \begin{vmatrix} p(\text{Plate}) \\ bx \\ by \\ bh \\ bw \end{vmatrix}$$

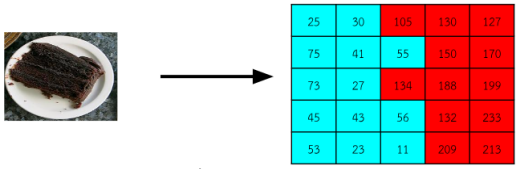
3.4.1.2 ขนาดของ anchor

ในแต่ละตารางจะหาได้จาก k-mean หากแต่ละตารางต้องการ 2 anchor

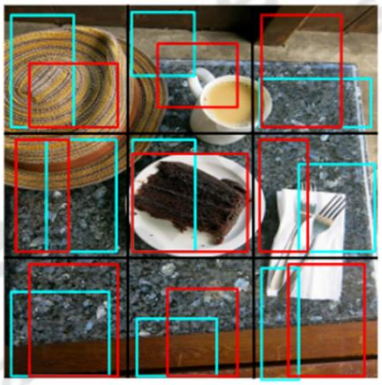
ภาพประกอบที่ 10 ตัวอย่างผลลัพธ์การกำหนดขนาด anchor โดยใช้ k-mean ผลลัพธ์ที่ได้ใน grid อื่นๆ

3.4.1.3 K-mean

K-Means เป็นวิธีที่นิยมใช้ในการแบ่งกลุ่มข้อมูล โดยเปรียบเทียบความคล้ายคลึง ของข้อมูล กับจุด ศูนย์กลางของแต่ละคลัสเตอร์ (Cluster) หรือค่าเฉลี่ย (Mean) เป็นการแบ่งแบบ Partitional clustering ด้วยการแบ่งข้อมูลออกเป็นส่วน ตามจำนวนกลุ่มที่ระบุ

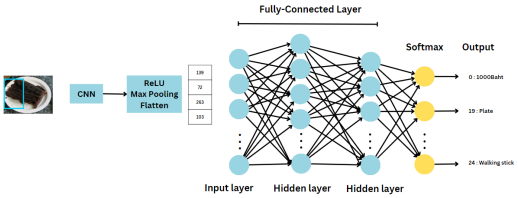


ภาพประกอบที่ 11 ตัวอย่างผลลัพธ์การจัดกลุ่มข้อมูล

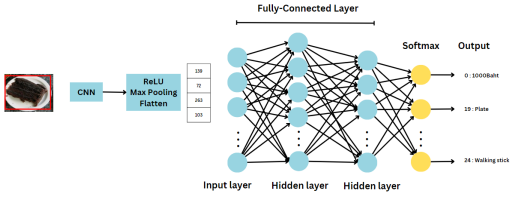


ภาพประกอบที่ 12 การแบ่งกลุ่มข้อมูลในแต่ละช่อง โดยแต่ละช่องจะมี 2 anchor

3.4.1.4 นำแต่ละ bounding box สกัด Features ด้วย CNN



ภาพประกอบที่ 13 นำแต่ละ bounding box สีน้ำเงินสกัด Features ด้วย CNN และเข้าโมเดล



ภาพประกอบที่ 14 นำแต่ละ bounding box สกัด Features ด้วย CNN และเข้าโมเดล

การเก็บค่า Y ของ grid ที่ 5 หลังจากเข้าโมเดลจะทราบว่าวัตถุที่อยู่ใน bounding box นั้นเป็น class อะไร

$$y = \begin{vmatrix} p(Class) \\ bx \\ by \\ bh \\ bw \end{vmatrix}$$

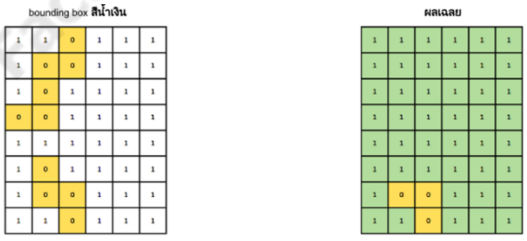
ใน grid อื่นๆก็ทำเช่นเดียวกันจนครบทุก grid จากนั้นต้องหาค่า probity โดยหาจากค่า IOU

3.4.1.5 IOU

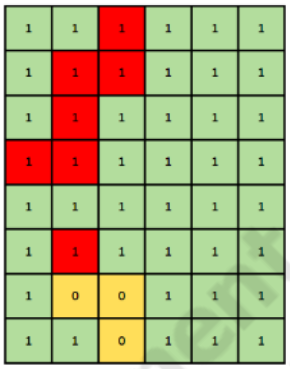
ในขั้นตอนนี้จะเป็นการนำแต่ละ bounding box ไปเปรียบเทียบกับผลเฉลย

$$IOU = \frac{area\_intersection}{areaUnion}$$

areaunio = area_box1 + area_box2 - area_intersection

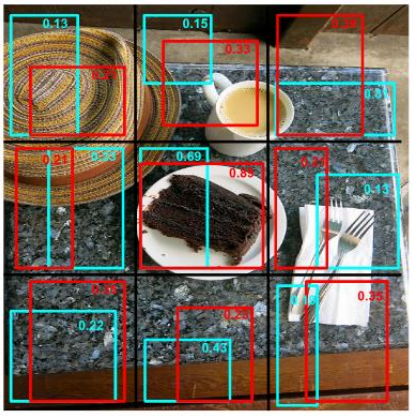


ภาพประกอบที่ 15 ตัวอย่างข้อมูลของกรอบสีน้ำเงินและผลเฉลย



ภาพประกอบที่ 16 areaOverlap

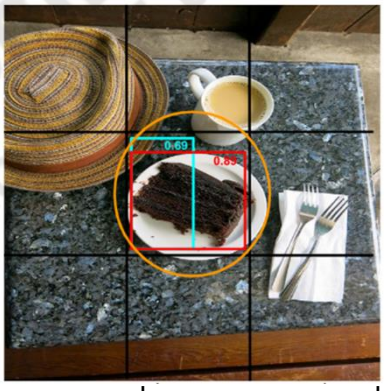
$$y = \begin{vmatrix} p(0.69) \\ 0.107 \\ 0.45 \\ 0.61 \\ 0.37 \\ p(0.81) \\ 0.133 \\ 0.41 \\ 0.72 \\ 0.45 \end{vmatrix}$$



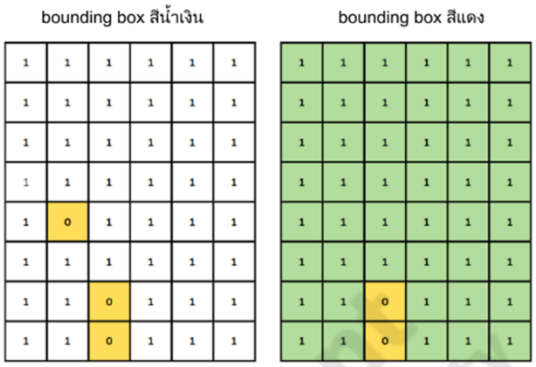
ภาพประกอบที่ 17 ตัวอย่างหลังจากการหาค่า probity ทั้งหมด

ภาพประกอบที่ 18 ตัดข้อมูลที่มีความน่าจะเป็นน้อยกว่าที่กำหนดออก

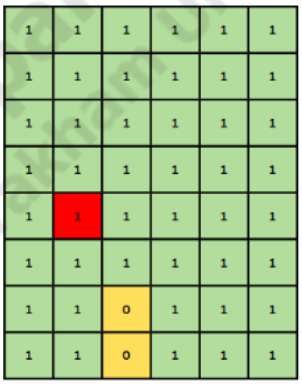
3.4.1.6 non-max suppression โดยวิธีการนี้จะทำการคำนวณค่า IOU ซึ่งเป็นการหาอัตราส่วน ของ Intersection area และ Union area ซึ่งเราจะกำหนดว่า ถ้าค่า IOU มากกว่า 0.5 เมื่อค่า IOU สูง แสดงว่ามี bounding box กำลังซ้อนทับกันอยู่ จะใช้วิธีการ non-max suppression ซึ่งจะสนใจ bounding box ที่มีความน่าจะเป็นสูงสุด ถ้าค่า IOU ต่ำกว่า 0.5 เราจะใช้คำตอบของทั้งสอง bounding box



ภาพประกอบที่ 19 ตัวอย่าง grid ที่ 5



ภาพประกอบที่ 20 ตัวอย่างข้อมูล bounding box สีน้ำเงิน และสีแดง



ภาพประกอบที่ 21 intersection_area ของ bounding box สีน้ำเงินและสีแดง

จากสูตรคำนวณ IOU คือ

$$IOU = \frac{area\_intersection}{areaUnion} \quad (1)$$

areaunio = area_box1 + area_box2 - area_intersection

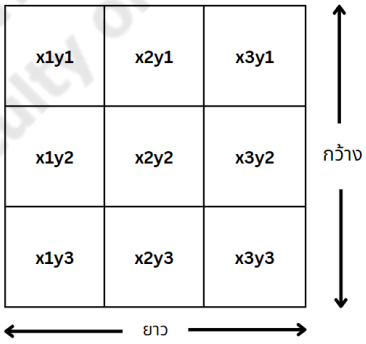
จากการคำนวณ IOU ของ grid ตัวอย่างได้ค่า IOU เท่ากับ 0.89 ซึ่งมากกว่าที่เรากำหนดไว้ แสดงว่า bounding box สีน้ำเงินและ

bounding box สีแดงกำลังครอบวัตถุเดียวกันอยู่ จึงใช้วิธีการ non-max suppression ซึ่งจะสนใจ bounding box ที่มีความน่าจะเป็นสูงสุด ใน gird ที่ 5 ก็ทำเช่นกัน จนครบทุก grid

ภาพประกอบที่ 22 ตัวอย่าง bounding box ที่มีความน่าจะเป็นสูงสุด

3.5 การระบุตำแหน่งวัตถุ

ในการระบุตำแหน่งวัตถุ จะใช้จุดตัด 9 ช่องในการระบุ ซึ่งในการระบุหลังจากที่ได้ค่า x และ y ของวัตถุ แล้วนำมาคำนวณกับความกว้าง และ ความยาว ของจอมือถือ



ภาพประกอบที่ 23 การระบุตำแหน่งวัตถุ

ตำแหน่งใน จุดตัด 9 ช่อง

1. Upper Left = x1y1
2. Upper Middle = x2y1
3. Upper Right = x3y1
4. Middle Left = x1y2
5. Middle = x2y2
6. Middle Right = x3y2
7. Lower Left = x1y3
8. Lower Middle = x2y3
9. Lower Right = x3y3

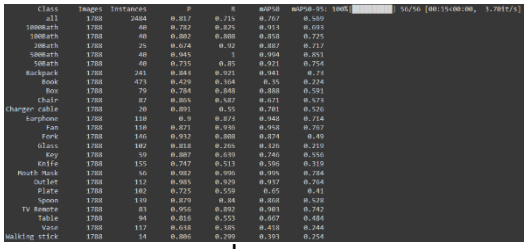
4.ผลการทดลอง

4.1 การเปรียบเทียบประสิทธิภาพการรู้จำจากสถาปัตยกรรม CNN

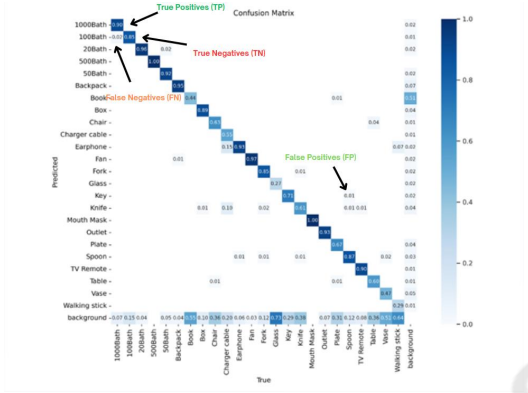
ในการจำแนกสิ่งของภายในบ้านได้ประยุกต์ใช้ร่วมกับสถาปัตยกรรม CNN โดยผู้จัดทำได้เลือกสถาปัตยกรรม YOLOv5n เป็นโมเดลที่มีประสิทธิภาพมากที่สุด โดยในการประเมินจะใช้ตาราง confusion matrix ในการทดลองครั้งนี้โดยใช้ชุดข้อมูลเดียวกัน

การประเมินประสิทธิภาพด้วย mAP50 (mean Average Precision at IoU 0.5) ในการวัดประสิทธิภาพ YOLOV5n จะใช้ชุดข้อมูลสำหรับการ Test ของของสิ่งภายในบ้านทั้งหมด 24 ชนิด คิดเป็น 10 เปอร์เซ็นต์ของทั้งหมด

ประสิทธิภาพ YOLOV5n มีค่า mAP50 = 0.767



ภาพประกอบที่ 24 ผลการประเมินประสิทธิภาพ YOLOv5n



ภาพประกอบที่ 25 การประเมินด้วย confusion matrix ของ YOLOv5n

ตารางที่ 1 ตัวอย่างผลการทำนาย

Class	ผลทำนายจากสถาปัตยกรรม		
	mAP50	YOLOv5n	ผลเฉลย
1000Baht	0.913	1000Baht	1000Baht
100Baht	0.858	100Baht	100Baht
20Baht	0.887	20Baht	20Baht
500Baht	0.994	500Baht	500Baht
50Baht	0.921	50Baht	50Baht
Backpack	0.941	Backpack	Backpack
Book	0.35	Book	Book
Box	0.888	Box	Box
Chair	0.671	Chair	Chair
Charger cable	0.701	Charger cable	Charger cable
Earphone	0.948	Earphone	Earphone
Fan	0.958	Fan	Fan
Fork	0.874	Fork	Fork
Glass	0.326	Glass	Glass
Key	0.746	Key	Key
Knife	0.596	Knife	Knife
Mouth mask	0.995	Mouth Mask	Mouth Mask
Outlet	0.937	Outlet	Outlet
Plate	0.65	Plate	Plate
Spoon	0.868	Spoon	Spoon
TV Remote	0.903	TV Remote	TV Remote
Table	0.667	Table	Table
Vase	0.418	Vase	Vase
Walking stick	0.393	ไม่ตรวจจับ	Walking stick

4.3 สรุปและวิเคราะห์ผลการทดลอง

จากการทดลองในการประเมินประสิทธิภาพโมเดล โดยการทดลองจากชุดข้อมูลรูป จากภาพประกอบที่ 4.1 และภาพประกอบที่ 4.2 พบว่า YOLOv5n มีค่า mAP50 เป็น 76

เปอร์เซ็นต์ ซึ่งมีค่าค่อนข้างดีแต่ก็ยังไม่ดีมากนัก เนื่องจากยังมีคลาสที่มีค่า mAP50 ต่ำกว่า 0.5

ตารางที่2 คลาสที่มีค่า mAP50 ต่ำกว่า 0.5

Class	mAP50
Book	0. 35
Glass	0.326
Vase	0.418
Walking stick	0.393

5. สรุปและอภิปรายผลการทดลอง

5.1 สรุปผลและอภิปรายผล

โครงงานปริญญานิพนธ์ฉบับนี้นำเสนอการตรวจจับสิ่งของภายในบ้านด้วยการเรียนรู้เชิงลึก (Household Detection Using Deep Learning) โดยใช้สถาปัตยกรรม YOLOV5 ในการเทรนรอบแรกไม่มีโมเดลใดที่ความแม่นยำเกิน 70 เปอร์เซ็นต์ จำเป็นจะต้องเทรนมากกว่า 1 รอบ เพื่อปรับพารามิเตอร์และน้ำหนักของโมเดลเพื่อให้มีความแม่นยำมากขึ้น นี่เป็นกระบวนการที่จำเป็นเพื่อเรียนรู้โมเดลให้สามารถตรวจจับวัตถุในภาพได้ดียิ่งขึ้น จนไม่มีการเปลี่ยนแปลง ผลลัพธ์ที่ได้คือค่าแม่นยำที่เพิ่มขึ้นเกิน 70 เปอร์เซ็นต์ อย่างไรก็ตามผลลัพธ์ที่กล่าวมาเกี่ยวกับตัวเลขเท่านั้น สิ่งที่สำคัญไม่ใช่เพียงแค่จำนวนรอบเทรนแต่ชุดข้อมูลที่มีจำนวนมากและมีความหลากหลายก็สำคัญเช่นกัน

5.2 ปัญหาและอุปสรรคในการดำเนินงาน

การหาชุดข้อมูล (dataset) ทำได้ยาก ปัญหาชุดข้อมูลที่มีจำนวนน้อยและช่องทางการหารูปภาพที่จำกัดทำให้การหารูปภาพที่มีคุณภาพและจำนวนที่เพียงพอในการเทรนโมเดล ทำให้ต้องทำการทดลองซ้ำหลายครั้ง และในหลายๆ ครั้งก็ได้ผลลัพธ์ที่ไม่ได้ตามที่คาดหวังเอาไว้

ปัญหาการ convert เป็น tflite แล้วนำมาใช้บน Mobile application เนื่องจาก flutter plugin บางตัวเก่าเกินไปอาจจะไม่รองรับการใช้งานสำหรับโมเดลใหม่ๆ

5.3 ข้อเสนอแนะ

1) การทำโครงงานเกี่ยวกับ Convolutional Neural Networks ในการเก็บชุดข้อมูล การเลือกรูปที่มีความคมชัดของลักษณะเด่นของสัตว์หรือสิ่งของนั้นๆ เป็นข้อสำคัญสำหรับการฝึกฝนและทดสอบโมเดล CNN เพราะจะช่วยให้โมเดลเรียนรู้และสกัดลักษณะเด่นได้ดีขึ้น

2) ปริมาณของชุดข้อมูลควรจะเพียงพอสำหรับการฝึกฝนและทดสอบ เพราะคุณภาพของโมเดลขึ้นอยู่กับปริมาณของชุดข้อมูลที่มีการกระจายในแต่ละกลุ่มหรือคลาสที่สมดุลกัน เพื่อป้องกันปัญหาการเรียนรู้ที่ไม่สมดุลในแต่ละคลาส

3) การนำโมเดลมาใช้บนแอปพลิเคชันมีปัญหาเนื่องจาก flutter plugin บางตัวเก่าเกินไปอาจจะไม่รองรับการใช้งานสำหรับโมเดลใหม่ๆ

เอกสารอ้างอิง

1. Intouch Kunakorntum, (2021), Image detection โดยใช้ YOLOv5 จากต้นจนจบ (ตอน 1), Retrieved 9 September 2022 from https://shorturl.asia/i4ynp
2. Intouch Kunakorntum, (2021), Image detection โดยใช้ YOLOv5 จากต้นจนจบ (ตอน 2 : Data Gathering and Collecting), Retrieved 11 September 2022 from https://shorturl.asia/nw5KN
3. Intouch Kunakorntum, (2021), Image detection โดยใช้ YOLOv5 จากต้นจนจบ (ตอน 3: Data Labeling and Image Augmentation), Retrieved 20 September 2023 from https://shorturl.asia/MJR2C
4. Intouch Kunakorntum, (2021), Image detection โดยใช้ YOLOv5 จากต้นจนจบ (ตอน 4: Model Training), Retrieved 14 April 2023 from https://shorturl.asia/mRATL
5. Liang Han Sheng, (2022), Yolov5 TFLite — Inferencing in Mobile Devices, Retrieved 17 August 2023 from https://shorturl.asia/Dy8jE
6. Mayank Mishra, (2020), Convolutional Neural Networks, Retrieved 10 September 2023 from https://shorturl.asia/vk6uR
7. ultralytics, (2022), YOLOv5 SOTA Realtime Instance Segmentation, Retrieved 10 September 2023, from https://github.com/ultralytics/yolov5/releases
8. Pagon Gatchalee, (2019), Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ใน Machine learning, Retrieved 14 September 2023 from https://shorturl.asia /4dXLO
9. ECoding, (2021), Softmax Function คืออะไร, Retrieved 13 August 2023 from https://shorturl.asia/SyE2W
10. Jens Tofte, (2022), Flutter TensorFlow lite | Basic Setup | Object Recognition | MobleNetSSD, Retrieved 28 July 2023 from https://shorturl.asia/8Sjby
11. Backslash Flutter, (2021), Flutter Text To Speech (TTS) | Flutter Tutorials For Beginners | 2021 , Retrieved 10 August 2023 from https://shorturl.asia/6kJOA

12. Marcus Ng, (2020), Flutter Speech to Text App Tutorial | Voice Recognition, Retrieved 10 August 2023 from https://shorturl.asia/nQlVo
13. Lookout, (2019), Lookout(Varies with device) [ซอฟต์แวร์แอปพลิเคชันมือถือ], Retrieved from https://shorturl.asia/VYRmN
14. Supersense, (2022), Supersense(1.4.13) [ซอฟต์แวร์แอปพลิเคชันมือถือ], Retrieved from https://play.google.com/store/apps/details?id=com.mediate.supersense&hl=en_US