การวิเคราะห์ข้อมูลพอยต์คลาวด์ไลดาร์เพื่อการตรวจจับวัตถุแบบสามมิติในรถยนต์ขับเคลื่อนอัตโนมัติ

LiDAR point cloud analysis for 3D Object Detection in autonomous vehicles

ศิรัส ทองอรุณ 63010914

Sirus Thongarun 63010914

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2566

การวิเคราะห์ข้อมูลพอยต์คลาวด์ไลดาร์เพื่อการตรวจจับวัตถุแบบสามมิติในรถยนต์ขับเคลื่อนอัตโนมัติ

LiDAR point cloud analysis for 3D Object Detection in autonomous vehicles

จัดทำโดย

ศิรัส ทองอรุณ 63010914

Sirus Thongarun 63010914

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2566

รายงานวิชา โครงงาน 2 ปีการศึกษา 2566

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์ข้อมูลพอยต์คลาวด์ไลดาร์เพื่อการตรวจจับวัตถุแบบสามมิติในรถยนต์ขับเคลื่อนอัตโนมัติ

LiDAR point cloud analysis for 3D Object Detection in autonomous vehicles

ผู้จัดทำ นายศิรัส ทองอรุณ รหัสนักศึกษา 63010914

รายงานนี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว

Sumet Wiatchai

(ผศ.ดร.สุเมฆ วิศยทักษิณ)

อาจารย์ที่ปรึกษา

หัวข้อโครงงาน การวิเคราะห์ข้อมูลพอยต์คลาวด์ไลดาร์เพื่อการตรวจจับวัตถุแบบสามมิติในรถยนต์ขับเคลื่อนอัตโนมัติ

นักศึกษา นายศิรัส ทองอรุณ รหัสนักศึกษา 63010914

ปริญญา วิศวกรรมศาสตร์บัณฑิต

ภาควิชา วิศวกรรมอิเล็กทรอนิกส์

ปีการศึกษา 2566

อาจารย์ที่ปรึกษาโครงงาน ผศ.ดร.สุเมฆ วิศยทักษิณ

บทคัดย่อ

ในปัจจุบัน ไลดาร์เป็นเซนเซอร์ชนิดหนึ่งซึ่งเข้ามามีบทบาทสำคัญในระบบยานยนต์ขับเคลื่อนอัตโนมัติเพื่อใช้ตรวจจับวัตถุ โดยไลดาร์นั้นจะใช้การวัดระยะทางด้วยเลเซอร์ในย่านใกล้อินฟราเรดซึ่งทำให้มีความละเอียดของข้อมูลที่มากกว่าการใช้เรดาร์ในการตรวจจับระยะ และ ข้อมูลรูปภาพจากกล้อง และด้วยการประยุกต์ใช้ไลดาร์เพื่อตรวจจับวัตถุแบบสามมิติ ทำให้การตรวจจับมีความแม่นยำมากขึ้น อย่างไรก็ตาม ข้อมูลที่ได้จากไลดาร์นั้นกระจัดกระจายมาก กล่าวคือ ไม่มีโครงสร้างของข้อมูลที่ชัดเจน โดยเปรียบเทียบกับข้อมูลรูปภาพที่มีโครงสร้างของพิกเซลที่ชัดเจน มากไปกว่านั้น ข้อมูลไลดาร์ยังมีปริมาณที่ค่อนข้างมากเมื่อเทียบกับข้อมูลรูปภาพจากกล้อง ซึ่งอาจทำให้การคำนวณต่างๆต้องใช้ระยะเวลาพอสมควรจนอาจจะไม่สามารถทำงานในเวลาจริงได้ ดังนั้น ในโครงงานฉบับนี้จึงมุ่งเน้นไปที่การศึกษาระเบียบวิธีการประมวลผลข้อมูลไลดาร์แบบต่างๆ รวมถึงการใช้โครงข่ายประสาทเทียมในการสร้างตัวตรวจจับวัตถุแบบสามมิติโดยใช้ข้อมูลไลดาร์

Project title	LiDAR point cloud analysis for 3D Object Detection in autonomous vehicles
Students	Mr. Sirus Thongarun Student ID 63010914
Degree	Bachelor of Engineering
Program	Electronics Engineering
Academic Year	2022
Project Advisor	Asst. Prof. Sumek Wisayataksin

ABSTRACT

Currently, LiDAR is a type of sensor that plays an important role in autonomous vehicle systems for detecting objects. LiDAR uses laser to measure the distance in the near-infrared region, which provides higher resolution than radar and camera data. and by applying lidar to detect objects in three dimensions. This makes the detection more accurate. However, the information obtained from LiDAR is very sparse, that is, there is no clear data structure. By comparing with image data that has a clear pixel structure. more than that the amount of LiDAR data is still quite large, which can cause various calculations to take a considerable amount of time. Therefore, this project focuses on studying different LiDAR data processing methods. This includes using neural networks to create 3D object detectors using LiDAR data.

กิตติกรรมประกาศ

โครงงานครั้งนี้ สามารถสำเร็จลุล่วงได้จาก ผศ.ดร.สุเมฆ วิศยทักษิณ อาจารย์ที่ปรึกษาโครงงานที่ได้สละเวลาอันมีค่า เพื่อแนะนำ คำสอนต่างๆ เกี่ยวกับความรู้ทั้งด้านวงจรดิจิทัล การประมวลผลดิจิทัล ตลอดจนการตรวจทาน และ แก้ไขข้อบกพร่องต่างๆด้วยความเอาใจใส่จนโครงงานชิ้นนี้สำเร็จ ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณรุ่นพี่ทั้งปริญญาตรี และ ปริญญาโทในห้องแลบ DICEs ที่คอยให้คำแนะนำในการจัดทำโครงงานชิ้นนี้

สุดท้ายนี้ ขอขอบคุณครอบครัว และผู้เกี่ยวข้องต่างๆ ที่สนับสนุนการทำโครงงานนี้จนสำเร็จลุล่วงไปได้ด้วยดี

ศิรัส ทองอรุณ

สารบัญ

สารบัญรูป

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในระบบยานยนต์สมัยใหม่ ระบบขับเคลื่อนอัตโนมัติเริ่มมีส่วนเข้ามาเกี่ยวข้องในชีวิตประจำวัน ซึ่งทำให้เกิดความสะดวกสบาย และ ความปลอดภัยมากยิ่งขึ้นสำหรับผู้ใช้โดยเฉพาะในรถยนต์ หนึ่งในองค์ประกอบของระบบขับเคลื่อนอัตโนมัติก็คงจะเป็นอะไรไปไม่ได้นอกจากการตรวจรู้วัตถุ และ สภาพแวดล้อมโดยรอบของตัวยานยนต์นั้นๆ ซึ่งในปัจจุบันก็มีวิธีมากมายในการตรวจรู้ ไม่ว่าจะเป็นการใช้กล้องเดี่ยว (Monoscopic) หรือ กล้องรอบคันรถ (Surround Camera) และเซนเซอร์อย่างเรดาร์ (Radio Detection and Ranging) หรือ ไลดาร์ (Light Detection and Ranging) อย่างไรก็ตาม เซนเซอร์แต่ละชนิดนั้นก็มีคุณสมบัติเฉพาะตัวที่ต่างกันออกไป และ เหมาะสมกับในสถานการณ์ที่ต่างกันออกไป เช่น ในกล้องเดี่ยวหน้ารถ อาจเหมาะกับการใช้งานในช่วงเวลากลางวัน เพราะมีระดับแสงที่เหมาะสม แต่ในช่วงเวลากลางคืนอาจไม่เหมาะสมมากนัก เพราะปริมาณแสงที่ไม่เพียงพอ (กรณีที่มีไฟทางไม่สว่างพอ) จึงทำให้ประสิทธิภาพการทำงานของกล้องเดี่ยวลดลง อย่างไรก็ตาม ในโครงงานฉบับนี้จะมุ่งเน้นไปที่การศึกษาเซนเซอร์ไลดาร์ ซึ่งใช้หลักการของการสะท้อนกลับของแสงโดยเทียบกับระยะเวลาที่แสงเดินทางไปตกกระทบกับวัตถุและสะท้อนกลับมา เพื่อใช้คำนวณเป็นระยะทาง แต่สิ่งที่ชุดข้อมูลจากไลดาร์มีความแตกต่างจากข้อมูลรูปเลยนั่นคือ ข้อมูลไลดาร์มีโครงสร้างที่ไม่แน่นอน และมีการกระจายตัวของข้อมูลอย่างมาก หากเปรียบเทียบกับรูปภาพนั้น รูปภาพมีโครงสร้างที่แน่นอนกว่า นั่นคือ มีการเรียงตัวของพิกเซลอย่างแน่นอน แต่สำหรับไลดาร์นั้นไม่ได้มีการเรียงตัวเป็นพิกเซลอย่างในรูปภาพ แต่เป็นจุดที่เรียกว่าพอยต์คลาวด์ (Point Cloud) กระจายตัวอยู่ทั่วปริภูมิ 3 มิติ ดังนั้น การประมวลผลรูปภาพต่างๆ ซึ่งทำในปริภูมิ 2 มิติ ไม่สามารถปรับใช้กับชุดข้อมูลพอยต์คลาวด์ได้โดยตรง เพียงเพราะเป็นข้อมูลแบบ 3 มิติ แต่เป็นข้อมูลที่ไม่มีโครงสร้างแน่นอนอีกด้วย แต่ในปัจจุบัน มีการพัฒนาโครงสร้างประสาทเทียมหลายชนิดเพื่อนำมาใช้ในการจำแนกประเภทวัตถุแบบ 3 มิติโดยใช้ไลดาร์ได้เป็นจำนวนมาก ซึ่งอาจแบ่งประเภทของโครงสร้างได้เป็น 3 แบบหลักๆคือ 1) โครงสร้างที่ประมวลผลพอยต์คลาวด์โดยตรง และ 2) โดยสร้างที่ประมวลผลพอยต์คลาวด์ทางอ้อม เช่นการทำ Voxelization เพื่อแปลงพอยต์คลาวด์ให้เป็นกล่อง 3 มิติและประมวลผลต่อจากนั้น หรือแปลงให้เป็นรูป 2 มิติ (Bird’s Eye View) และประมวลผลต่อ ซึ่งในแต่ละแบบก็จะมีข้อดีและข้อเสียที่ต่างกันออกไป อย่างไรก็ตาม ในแต่ละโครงสร้างนั้น ไม่ว่าจะเป็นการประมวลผลทางตรงหรือทางอ้อมก็จะต้องประกอบไปด้วยองค์ประกอบของโครงข่ายประสาทเทียมประเภทเดียวกัน โครงงานนี้จึงจะทำการศึกษาโครงสร้างของประสาทเทียมประเภทต่างๆ และพัฒนาโครงสร้างที่มีอยู่ ให้ดียิ่งขึ้นต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาการจำแนกวัตถุแบบสามมิติโดยใช้ไลดาร์

1.2.2 เพื่อศึกษาความเป็นไปได้ในการพัฒนาตัวเร่งประมวลผล และ อัลกอริทึม

1.2.3 เพื่อศึกษาแนวโน้มความเป็นไปได้ในการพัฒนายานยนต์ขับเคลื่อนอัตโนมัติ

1.3 ขอบเขตของการศึกษา

1.3.1 ศึกษาการจำแนกประเภทวัตถุแบบสามมิติโดยใช้ข้อมูลจากไลดาร์

1.3.2 ศึกษาโครงข่ายประสาทเทียมที่จำเป็นสำหรับการจำแนกประเภทวัตถุแบบสามมิติ

1.4 ระยะเวลาในการทำโครงงาน

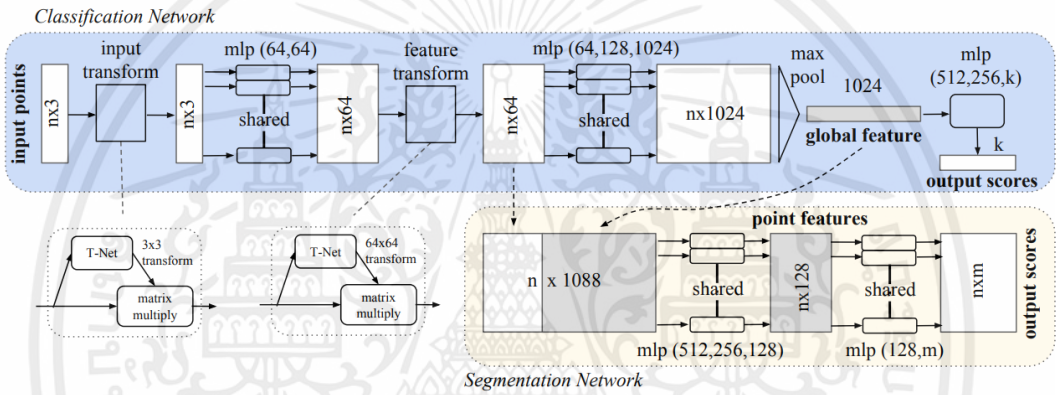
6 ธ.ค. 2566 – 6 มี.ค. 2567

บทที่ 2
ข้อมูลพื้นฐาน

2.1 งานวิจัยที่เกี่ยวข้อง (Related Work)

2.1.1 PointNet

PointNet คือ โครงสร้างของโครงข่ายประสาทเทียมชุดแรกๆ โดย Charles Qi ที่มีการนำข้อมูลพอยต์คลาวด์มาประมวลผล เช่น การจำแนกวัตถุ (Classification) และ การแบ่งส่วนวัตถุ (Segmentation) โดยใช้ข้อมูลพอยต์คลาวด์ตรงๆ โดยโครงสร้างของ PointNet จะมีลักษณะดังแสดงด้านล่าง

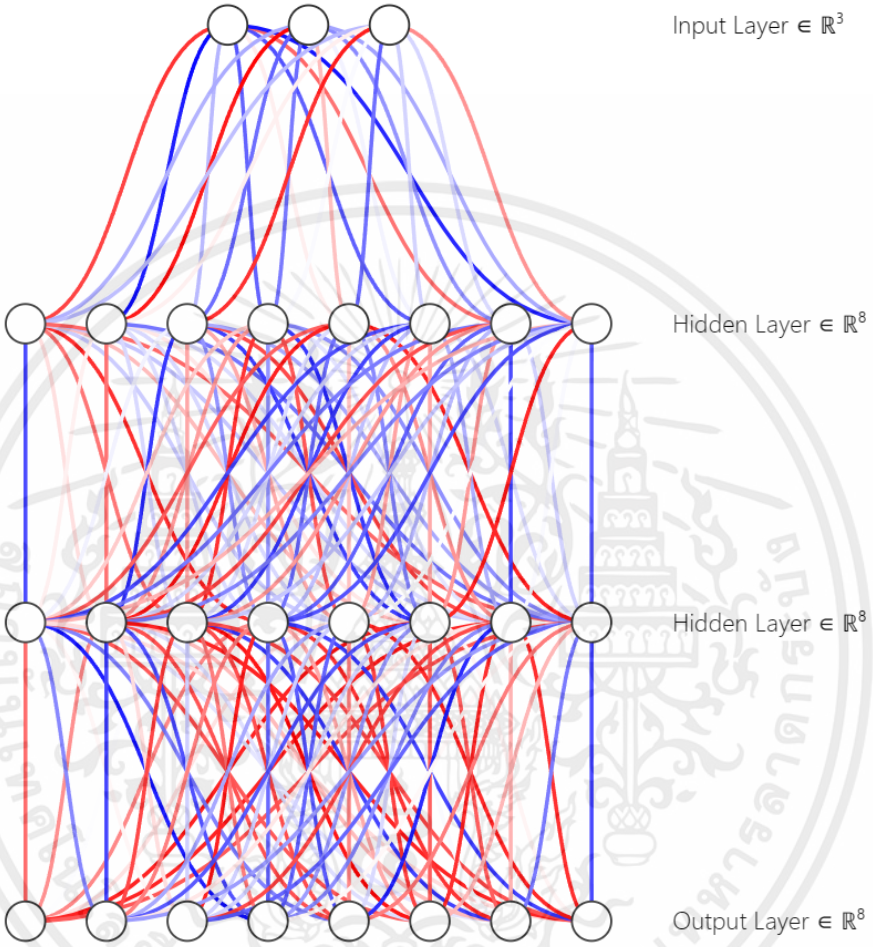


รูปที่ 1 โครงสร้าง PointNet แหล่งที่มา : https://arxiv.org/abs/1612.00593

เนื่องจากโครงสร้างของข้อมูลพอยต์คลาวด์นั้นเป็นโครงสร้างที่ไม่เป็นระเบียบ กล่าวคือ ไม่สามารถคาดเดาตำแหน่งที่แน่นอนของข้อมูลได้ เพราะข้อมูลที่ได้จากไลดาร์นั้นจะมีลักษณะที่บันทึกตำแหน่งทางปริภูมิสามมิติจากการสะท้อนกลับของแสงที่ได้ส่องออกไปตลอดเวลาที่ตัวอุปกรณ์มีการหมุน นั้นทำให้ไม่สามารถคาดเดาตำแหน่งที่แน่นอนของข้อมูลชุดต่อไปที่จะมาถึงได้เลย ขึ้นกับว่าจะมีการกวาดเลเซอร์ส่องแสงออกไปในตำแหน่งไหน ดังนั้น เมื่อมีการบันทึกข้อมูลในการทดลองที่ 1 ข้อมูลชุดที่ 1 อาจมาจากการตรวจจับการสะท้อนกลับเจอที่ตำแหน่ง (0,0,2) ในขณะที่การทดลองที่ 2 ข้อมูลชุดที่ 1 อาจจะมาจากตำแหน่ง (1,0,2) ก็เป็นได้ แม้จะมาจากการทดลองที่วัตถุเดียวกัน ดังนั้น ในโครงข่ายประสาทเทียมที่สร้างขึ้นจำเป็นที่จะต้องมีความเข้าใจแม้ข้อมูลจะมีการสับเปลี่ยนตำแหน่ง (Index Invariance) จึงเป็นเหตุผลที่ตัวโครงสร้างนั้นประกอบไปด้วยส่วนหัวใจของ PointNet นั่นคือ เพอร์เซ็ปตรอนหลายชั้นร่วม (Shared Multi-layer perceptron)

2.1.1.1 Shared multi-layer perceptron

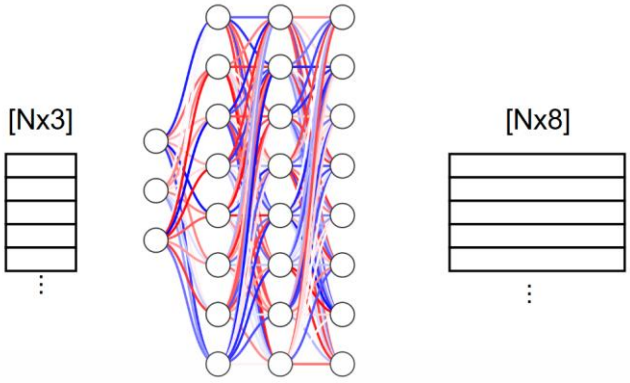
เพอร์เซ็ปตรอนหลายชั้นร่วม ในที่นี้จะย่อว่า Shared MLP สามารถอธิบายได้ด้วยโครงสร้างของชั้นเชื่อมโยงแบบสมบูรณ์ (Fully-connected Layer, FCN) ที่ข้อมูลขาเข้าทุกชุดจะผ่านชั้นนี้โดยใช้พารามิเตอร์เดียวกัน ซึ่งสามารถแสดงได้ดังรูปด้านล่าง



รูปที่ 2 ตัวอย่างโครงสร้าง Shared-MLP (8,8,8)

ด้วยวิธีการนี้เอง จึงทำให้ไม่ว่าลักษณะโครงสร้างการเรียงตัวของข้อมูลจะเป็นอย่างไร ตัวโครงข่ายประสาทเทียมก็ยังสามารถเข้าใจได้ โดยในโครงสร้างนี้จะทำการเรียนรู้ลักษณะของข้อมูลในระดับองค์รวม (Global Features)

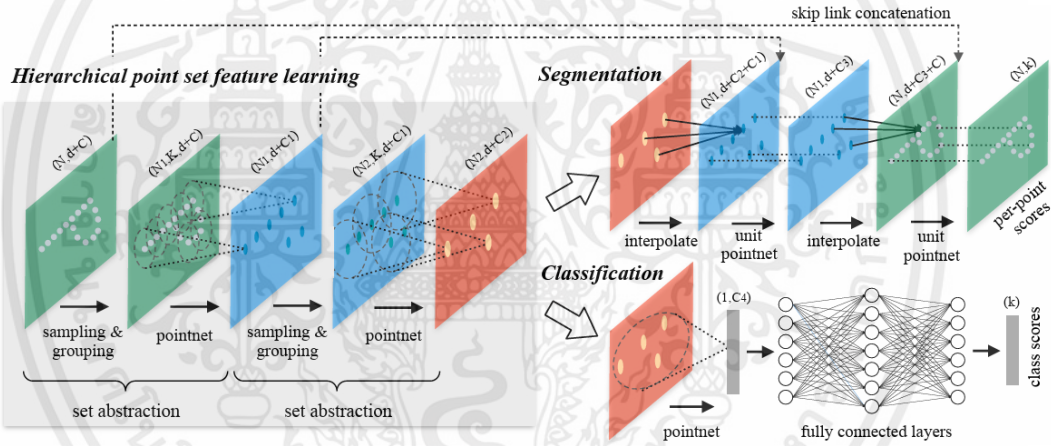
สำหรับการใช้งานเพื่อจำแนกวัตถุ เมื่อมีการคำนวณผ่าน Shared MLP ในหลายๆชั้นแล้ว จะทำการ Max Pooling เพื่อดึงลักษณะเด่นของคุณลักษณะทั้งหมดออกมาเป็นเวกเตอร์ของคุณลักษณะที่ไม่ขึ้นกับตำแหน่งของข้อมูล และนำไปสู่การใช้โครงสร้าง FCN เพื่อจำแนกชนิดของวัตถุต่อไป



รูปที่ 3 มิติของข้อมูลก่อน และ หลังผ่าน Shared-MLP

2.1.2 PointNet++

ในปีเดียวกันกับการเผยแพร่ผลงานของ PointNet Charles Qi ได้มีการเผยแพร่งานวิจัยที่เป็นส่วนขยายของ PointNet นั่นคือ PointNet++ ซึ่งมีโครงสร้างโดยรวมดังแสดงด้านล่าง



รูปที่ 4 โครงสร้าง PointNet++ แหล่งที่มา : https://arxiv.org/abs/1706.02413

โดยเหตุผลที่ว่าโครงสร้างของ PointNet เรียนรู้คุณลักษณะแบบองค์รวมอย่างเดียว ซึ่งทำให้คุณลักษณะแบบเจาะจง (Local Features) ที่อาจช่วยให้การจำแนกวัตถุ หรือ การแบ่งส่วนวัตถุสามารถทำได้ดีขึ้นถูกละเลยไป จึงมีการพัฒนาต่อโดยเพิ่มในเรื่องของการเรียนรู้คุณลักษณะแบบเจาะจงเข้าไปด้วย โดยสร้างชั้นที่เรียกว่า Set Abstraction (SA) ขึ้นมาโดยประกอบไปด้วยส่วนของการสุ่มเลือกข้อมูลที่ใช้หลักการสุ่มจุดห่างไกลกันที่สุด (Furthest Point Sampling), ส่วนของการจับกลุ่ม (Grouping) และ ส่วนของโครงสร้าง PointNet แบบเล็ก (Mini-PointNet) รวมไว้ในชั้นเดียวกัน ทำให้โครงสร้างนี้มีความแม่นยำในการจำแนกวัตถุ และ การแบ่งส่วนวัตถุได้ดีกว่า ซึ่งในแต่ละส่วนสามารถอธิบายได้ดังแสดงด้านล่าง

2.1.2.1 Furthest Point Sampling

การสุ่มจุดห่างไกลกันที่สุดคือการสุ่มจุดบนปริภูมิสามมิติเป็นจำนวน N จุดที่มีระยะทางห่างกันมากที่สุดขึ้นมา เพื่อรับรองว่าจุดทั้งหมดที่สุ่มนั้น จะกระจายตัวอยู่บนปริภูมิสามมิติมากที่สุด ซึ่งในที่นี้ จะได้เป็นข้อมูลมิติ [N, 3] ขึ้นมา โดย 3 คือตำแหน่งของจุดเหล่านั้น

2.1.2.2 Grouping

เมื่อมีการสุ่มจุดที่ห่างไกลกันที่สุดขึ้นมาเป็นจำนวน N จุดแล้ว จะมองจุดเหล่านั้นเป็นจุดศูนย์กลาง และทำการจับกลุ่มจุดใดๆ ที่มีระยะจากจุดถึงศูนย์กลางนั้นๆ น้อยกว่าค่า r ใดๆ ซึ่งวิธีนี้จะเรียกว่า Ball Query แต่ด้วยวิธีนี้เอง จึงทำให้แต่ละจุดศูนย์กลางแต่ละจุดนั้นมีจำนวนของกลุ่มจุดที่จับกลุ่มไม่เท่ากัน อันเนื่องมาจากการกระจายตัวของพอยต์คลาวด์ที่แตกต่างกันอย่างมาก เช่น ในข้อมูลที่อยู่ใกล้ๆ อาจมีความหนาแน่นของพอยต์คลาวด์ที่มาก ซึ่งตรงข้ามกับข้อมูลที่อยู่ไกลๆ ดังนั้น จำนวนจุดที่อยู่ภายในรัศมีทรงกลม r ของจุดศูนย์กลางใดๆ อาจมีมากหรือน้อยแตกต่างกันออกไป อย่างไรก็ตาม จะมีการกำหนดจำนวนจุดพอยต์คลาวด์ที่จะจับกลุ่มสูงสุดได้ K จุด ซึ่งเป็นค่าสูงสุดที่จุดศูนย์กลางใดๆจะสามารถจับกลุ่มได้นั่นเอง และจะได้ข้อมูลที่มีมิติ [N, K, 3] ขึ้นมา

2.1.2.3 Mini-PointNet

หลังจากที่ได้กลุ่มของพอยต์คลาวด์ในแต่ละจุดศูนย์กลางขึ้นมาแล้ว จะนำกลุ่มของพอยต์คลาวด์ในแต่ละจุดศูนย์กลางนั้นไปผ่านโครงสร้างของ PointNet แบบเล็ก นั่นคือในส่วนของ Shared-MLP และ Max Pooling เท่านั้น เพื่อสกัดคุณลักษณะเด่นแบบเจาะจงออกมา ทำให้จะได้ข้อมูลสุดท้ายมีมิติ [N, 3 + C] โดย C คือจำนวนคุณลักษณะที่ได้จากโครงสร้าง PointNet แบบย่อนั้นเอง

2.1.2.4 โครงสร้างของ PointNet++ สำหรับการจำแนกวัตถุ

หลังจากที่ผ่านชั้น SA (โดยอาจมีมากกว่า 1 ชั้นก็ได้) มาแล้ว ก็จะนำข้อมูลเหล่านั้นมาเข้าโครงสร้าง PointNet แบบเล็กเพื่อสกัดเป็นคุณสมบัติแบบองค์รวมออกมา และนำไปเข้าสู่ชั้นของ FCN เพื่อสกัดเป็นความน่าจะเป็นของชนิดวัตถุนั้นๆต่อไป

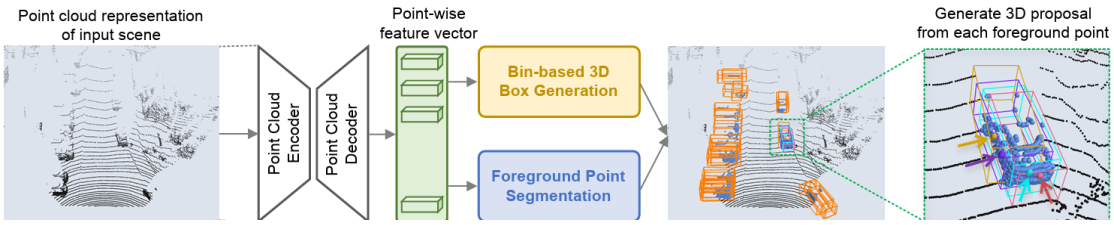
2.1.3 PointRCNN

PointRCNN เป็นส่วนต่อขยายที่พัฒนาโดยใช้โครงสร้างของ PointNet++ Semantic Segmantation มาเพิ่มเติมในส่วนของ Second Stage เพื่อใช้สำหรับการตรวจจับวัตถุ

2.1.3.1 PointRCNN first stage

ในส่วนแรกของโมเดล PointRCNN จะเป็นการใช้ PointNet++ ในส่วนของ Semantic Segmentation เพื่อใช้สกัดคุณสมบัติของแต่ละจุด และนำไปเข้า FCN เพื่อใช้สกัดหาจุดพื้นหน้า

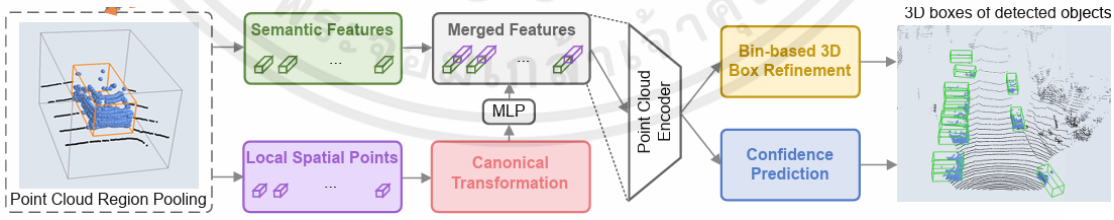
(Foreground Point) ทั้งหมด และ สกัดหา Bounding Box แบบคร่าวๆ โดยจะถือว่าจุดใดๆที่อยู่ภายใต้ Bounding Box จริงๆของตัววัตถุนั้น เป็นจุดพื้นหน้าทั้งหมด



รูปที่ 5 โครงสร้าง PointRCNN ในส่วนแรก แหล่งที่มา : https://arxiv.org/abs/1812.04244

2.1.3.2 PointRCNN second stage

ในส่วนที่สองของโมเดลนั้น จะทำการ Pool จุดต่างๆที่อยู่ภายใต้ Bounding Box ที่ทำนายออกมา แล้วทำการแปลงพิกัดให้อยู่ในรูปของ Canonical ก่อน กล่าวคือ แปลงให้มีตำแหน่งของจุดที่ใช้ทำนายเป็นตำแหน่งเริ่มต้น (Origin Point) จากนั้นนำไปเข้า PointNet-Mini เพื่อสกัดเป็นตำแหน่งที่ศูนย์กลางของตัววัตถุ และ ขนาดของตัววัตถุ รวมถึง ชนิดของวัตถุนั้นๆ โดยจะใช้การทำนายตำแหน่งแบบ bin-based กล่าวคือ ตัวโครงข่ายจะทำนายตำแหน่งในลักษณะของช่องที่ตัววัตถุนั้นอยู่ ซึ่งเป็นค่าแบบไม่ต่อเนื่อง (Discrete) และจะทำนายส่วนเหลือ (Residual) ซึ่งเป็นการเพิ่มเติมความแม่นยำของการทำนายศูนย์กลางของวัตถุ และด้วยเหตุผลที่ใช้การทำนายศูนย์กลางแบบไม่ต่อเนื่องนั้น ก็เพื่อที่จะสามารถใช้ฟังก์ชันสูญเสีย (Loss Function) แบบ Cross Entropy Loss ได้ ซึ่งพบว่ามีความสามารถในการทำนายตำแหน่งได้ดีกว่าแบบการใช้ค่าแบบต่อเนื่อง ในโมเดลนี้ยังใช้แนวคิดของ Anchor เพื่อทำนายวัตถุอีกด้วย ซึ่งทำให้ความสามารถในการทำนายขนาดของวัตถุสามารถทำได้ดี แต่อย่างไรก็ตาม การใช้ Anchor ก็อยากทำให้เกิดปัญหาซึ่งจะอธิบายต่อไปในโครงงานฉบับนี้

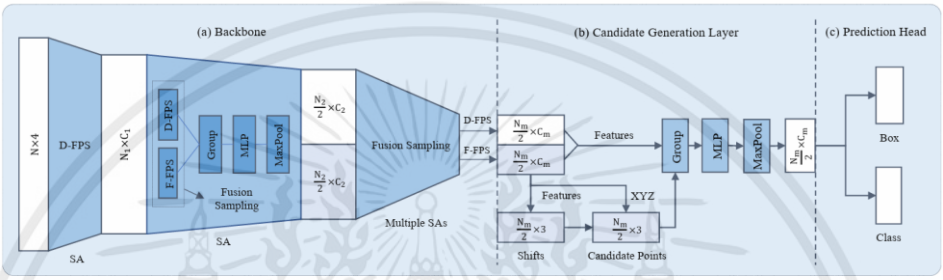


รูปที่ 6 โครงสร้าง PointRCNN ในส่วนที่สอง แหล่งที่มา : https://arxiv.org/abs/1812.04244

2.1.4 3D Single Stage Object Detector

เนื่องจาก PointRCNN เป็นโครงข่ายแบบสองตอน ซึ่งทำให้มีการประมวลผลที่ซ้ำ และ ใช้ทรัพยากรค่อนข้างมาก จึงอาจไม่เหมาะสำหรับการใช้งานแบบเวลาจริง (Real time) ดังนั้น Zetong Yang, et al. จึงนำแนวคิดบางส่วนของ PointRCNN มาพัฒนาต่อให้เป็นโครงข่ายแบบตอนเดียว โดย

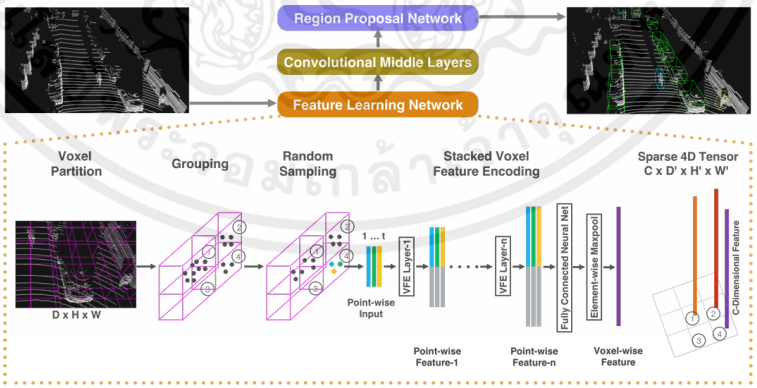
ในโครงข่ายนี้ จะนำส่วนของ SA เข้ามาใช้ แต่จะทำการเพิ่ม F-FPS ซึ่งคือการทำ Farthest Point Sampling แต่จะไม่ใช้ระยะทางในปริภูมิสามมิติมาคำนวณ แต่จะใช้ระยะทางในปริภูมิคุณสมบัติเข้ามาช่วย ซึ่งพบว่า การใช้ F-FPS จะทำให้ได้การสุ่มจุดสำหรับเซนทรอยด์มีโอกาสสุ่มโดยจุดภายใน Bounding Box จริงๆของตัววัตถุมากกว่าซึ่งให้ผลดีสำหรับการระบุตำแหน่งของตัววัตถุ จากนั้นจะทำการใช้จุดจาก F-FPS มาเป็นจุดตัวแทนศูนย์กลางวัตถุ เรียกว่า Candidate Point และจะใช้ SA มาประมวลผลอีกครั้ง ซึ่งในครั้งนี้จะไม่ใช้ FPS ในการสุ่มจุด แต่จะใช้ Candidate Point เหล่านั้นมาเป็นจุดศูนย์กลางแทน และทำการทำนายตำแหน่งของวัตถุ และ Bounding Box ออกมา



รูปที่ 7 โครงสร้างของ 3DSSD แหล่งที่มา : https://arxiv.org/abs/2002.10187

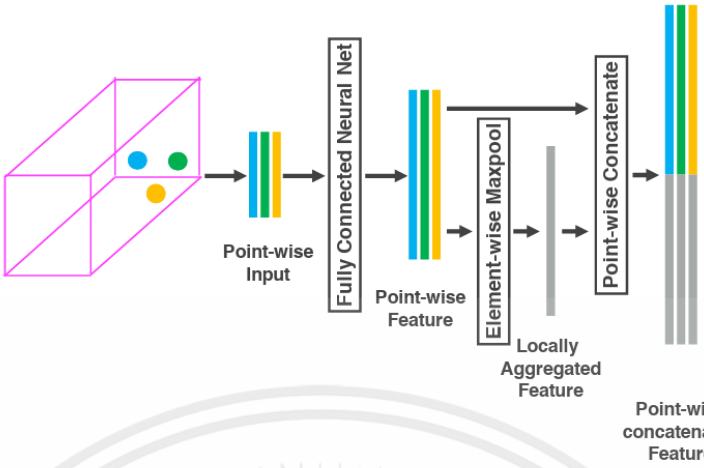
2.1.5 VoxelNet

VoxelNet เป็นการพัฒนาขั้นตอนประมวลผลก่อน (Pre-processing) ซึ่งมีจุดมุ่งหมายในการแปลงพอยต์คลาวด์ให้เป็นกล่องสามมิติ เรียกว่า ว็อกเซล (Voxel) ก่อน เพื่อที่จะสามารถใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันได้ ซึ่งนับว่าเป็นงานชิ้นแรกๆที่ได้มีการนำพอยต์คลาวด์มาใช้กับโครงข่ายแบบคอนโวลูชัน โดยมีโครงข่ายเข้ารหัสว็อกเซลเป็นหัวใจสำคัญสำหรับงานชิ้นนี้



รูปที่ 8 โครงสร้างของ VoxeNet แหล่งที่มา : https://arxiv.org/abs/1711.06396

2.1.5.1 Voxel Features Encoder

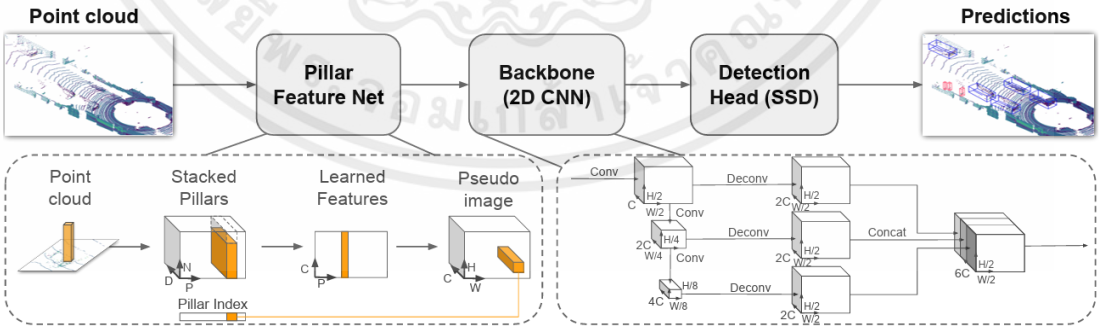


รูปที่ 9 โครงสร้างของ Voxel Features Encoder แหล่งที่มา : https://arxiv.org/abs/1711.06396

ในส่วนของการสกัดคุณสมบัติจากว็อกเซล จะเป็นการนำข้อมูลของแต่ละจุดภายในว็อกเซลนั้นๆมาผ่าน Shared-MLP และจากนั้นจะทำการ Max Pool ของทุกๆจุดภายในทุกว็อกเซลและนำผลลัพธ์ที่ได้เหล่านั้นไปต่อกับข้อมูลเดิมต่อไป

2.1.6 PointPillar

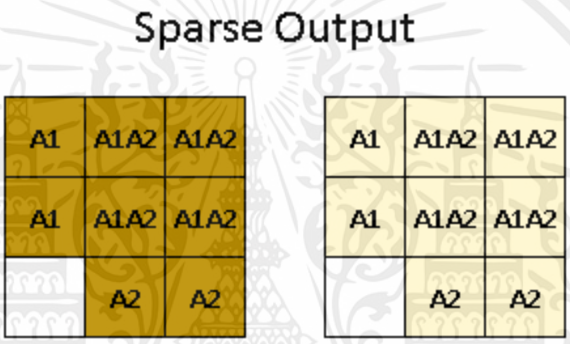
เนื่องจากปัญหาของ VoxelNet ที่ใช้การคำนวณที่สูงมากอันเนื่องมาจากการคอนโวลูชันแบบสามมิติ PointPillar โดย A.H. Lang จึงถือกำเนิดขึ้นโดยการแก้ไขจากการทำ Voxelization เป็น Pillarization ซึ่งคือการทำ Voxelization โดยกำหนดให้ขนาดของ Voxel ทางแกน z เป็นอนันต์ ซึ่งด้วยวิธีการนี้เอง ทำให้ข้อมูลผลลัพธ์ที่ได้นั้นเป็นแบบสองมิติ และสามารถใช้การคอนโวลูชันแบบสองมิติได้ ซึ่งทำให้ระยะเวลาในการประมวลผลน้อยลงอย่างเห็นได้ชัด



รูปที่ 10 โครงสร้างของ PointPillar แหล่งที่มา : https://arxiv.org/abs/1812.05784

2.1.7 Sparsely Embedded Convolution Detection

เนื่องจากปัญหาของ VoxelNet ที่ใช้การคำนวณที่สูงนั้น ทำให้ไม่สามารถประมวลผลในเวลาจริงได้ แต่ก็แลกมากับความแม่นยำในการตรวจจับที่สูง งานวิจัยนี้จึงถือกำเนิดขึ้นด้วยการแก้ไขปัญหาความช้าของการคอนโวลูชันแบบสามมิติ โดยสังเกตุจาก Voxel ส่วนมากที่ได้จากการ Voxelization นั้นเป็น Voxel ที่ว่างเปล่า กล่าวคือไม่มีข้อมูลใดๆอยู่เลย ซึ่งเป็นเพราะข้อมูลพอยต์คลาวด์ที่มีการกระจายตัวอย่างมาก ทำให้ในบางบริเวณของปริภูมิเป็นบริเวณว่างๆ แต่ในการคอนโวลูชันนั้น พื้นที่ว่างๆดังกล่าวจะถูกแทนที่ด้วยการเติมศูนย์ (Zero Padding) เพื่อให้การคอนโวลูชันสามารถทำได้ ดังนั้นจึงมีแนวคิดที่จะทำการคำนวณคอนโวลูชันในบริเวณที่มีข้อมูลอยู่เท่านั้น จึงเกิดเป็น Sparse Convolutional Neural Network ขึ้นมา ซึ่งสามารถแสดงได้ดังรูปด้านล่าง



รูปที่ 11 ผลลัพธ์การคอนโวลูชันแบบเบาบาง แหล่งที่มา : https://towardsdatascience.com/how-does-sparse-convolution-work-3257a0a8fd1

จะเห็นว่าการเลือกใช้งานคอนโวลูชันแบบเบาบาง (Sparse Convolution) ช่วยลดการประมวลผลไปค่อนข้างมาก จึงทำให้การคำนวณมีความเร็วมากขึ้นอย่างมีนัยสำคัญ

2.1.8 CenterPoint

จากทั้งหมดที่กล่าวไปนั้น มักจะใช้ส่วนทำนายชั้นสุดท้ายแบบมี Anchor ซึ่งปัญหาหลักของ Anchor คือการจัดวางตัวของ Bounding Box กับ ความเป็นจริงนั้นไม่ค่อยแม่นยำมากนัก ดังนั้นจึงใช้การทำนายแบบ Anchor free ซึ่งก็คือจะเป็นการทำนายกล่องแบบตรงๆเลย

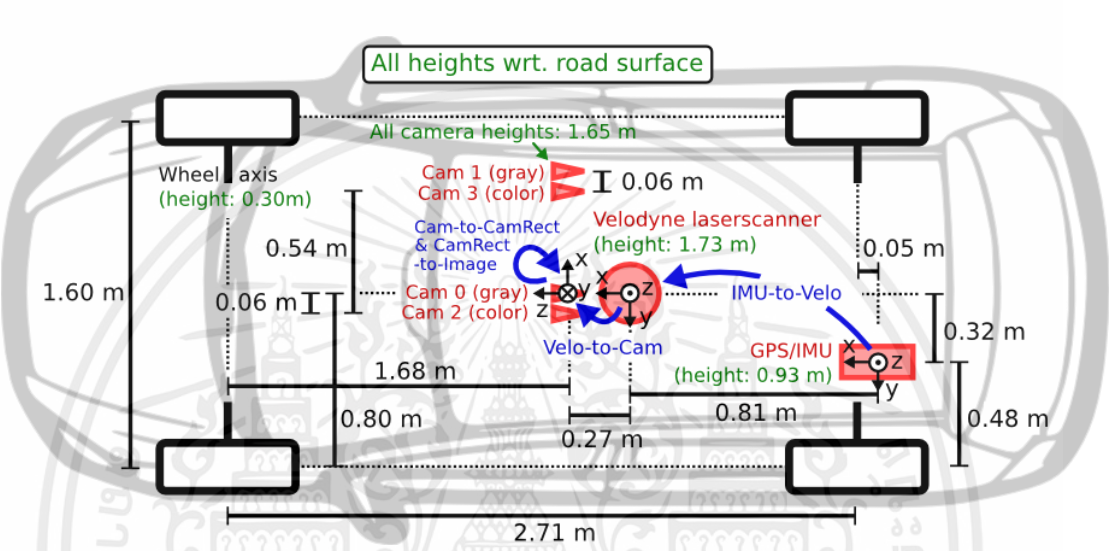
2.1.9 PillarNet

จากข้อสังเกตของ PointPillar ซึ่งมีความเร็วที่สูง แต่ยังขาดส่วนที่ใช้สกัดข้อมูล รวมถึงมีจำนวนของเสาหลักที่ว่างเป็นจำนวนมาก ในวิจัยฉบับนี้จึงยกโครงสร้าง PointPillar เดิมเพียงแต่เปลี่ยนหลักการคอนโวลูชันให้เป็นแบบเบาบาง (Sparse Convolution) แทน และยังเพิ่มชั้นของการคอนโวลูชันให้มากขึ้นกว่าเดิม จึงทำให้มีความแม่นยำในการทำนายมากขึ้นไปอีก

2.2 ข้อมูลพื้นฐานเกี่ยวกับชุดข้อมูล

2.2.1 ชุดข้อมูล KITTI

ชุดข้อมูล KITTI เป็นโปรเจกต์ความร่วมมือระหว่าง Toyota Technological Institute at Chicaco และ Karlsruhe Institute of Technology ได้จัดทำชุดข้อมูลขึ้นมาเพื่อใช้ในการสร้างพื้นฐานระบบยานยนต์ขับเคลื่อนอัตโนมัติ ซึ่งภายในประกอบไปด้วยชุดข้อมูลย่อยหลายชุดโดยมีเซ็นเซอร์ดังนี้



รูปที่ 12 การจัดวางเซนเซอร์ของชุดข้อมูล KITTI แหล่งที่มา : https://www.cvlibs.net/datasets/kitti/setup.php

1) ระบบนำทางเฉื่อย OXTS RT 3003

2) ไลดาร์ Velodyne HDL-64E ทำการสแกนบริเวณโดยรอบที่ความถี่ 10 Hz ให้ข้อมูลประมาณหนึ่งแสนจุดต่อเฟรม

3) กล้องสี (FL2-14S3C-C) และ กล้องขาวดำ (FL2-14S3M-C) จำนวนอย่างละ 2 กล้องโดยมีมุมมองภาพทางด้านหน้ารถอย่างเดียว ความละเอียด 1.4 ล้านพิกเซล มีขนาดของภาพหลังการประมวลผลก่อนอยู่ที่ 1382x512 โดยกล้องจะทำการจับภาพเมื่อไลดาร์ที่การหมุนอยู่ที่ตำแหน่งหน้ารถพอดี

4) เลนส์กล้องแบบเปลี่ยนความยาวโฟกัสได้ 4-8 mm (Edmund Optics NT59-917)

ในโครงงานฉบับนี้ ชุดข้อมูลที่ใช้คือ 3D Object Detection ซึ่งประกอบไปด้วยชุดข้อมูลไลดาร์และกล้องทั้งหมด 7481 เฟรมสำหรับการฝึกสอน และ 7518 เฟรมสำหรับการวัดผลซึ่งทั้งหมดประกอบไปด้วยวัตถุจำนวน 80256 วัตถุ ซึ่งมีการแบ่งความยากของการตรวจจับวัตถุเป็นสามระดับ นั่นคือ

1) ระดับง่าย โดยมีขนาดของ Bounding Box น้อยที่สุดอยู่ที่ 40 พิกเซล และ สามารถมองเห็นได้ง่ายผ่านกล้อง

2) ระดับปานกลาง โดยมีขนาดของ Bounding Box น้อยที่สุดอยู่ที่ 25 พิกเซล และสามารถมองเห็นได้แต่ไม่ชัดเจน อาจถูกบังบางส่วน

3) ระดับยาก โดยมีขนาดของ Bounding Box น้อยที่สุดอยู่ที่ 25 พิกเซล และ มองเห็นได้ยาก หรือ ถูกบังเกือบทั้งหมด

2.2.2 ชุดข้อมูล nuscenes

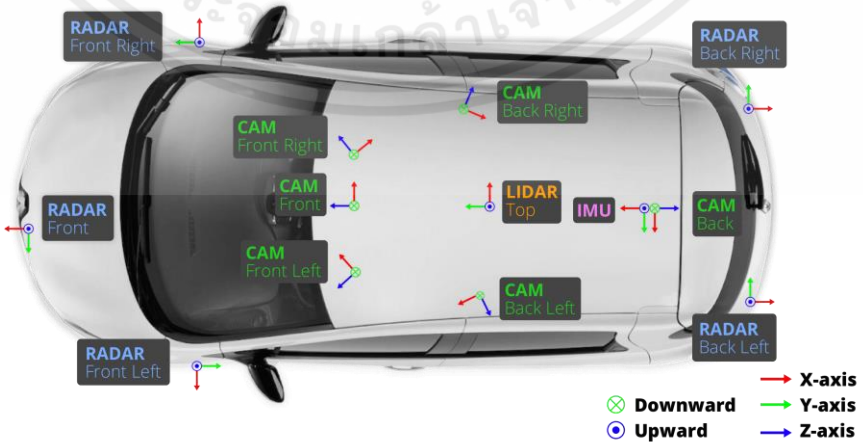
ชุดข้อมูล nuscenes เป็นชุดข้อมูลโดย Motional ที่เผยแพร่เมื่อเดือนมีนาคม 2562 โดยมีจุดมุ่งหมายเพื่อรองรับงานวิจัยที่เกี่ยวข้องกับการของเห็นของคอมพิวเตอร์ และ ยานยนต์ขับเคลื่อนอัตโนมัติ โดยมีชุดเซ็นเซอร์ดังนี้

1) ระบบนำทางเฉื่อย

2) ไลดาร์ Velodyne HDL-32E ทำการสแกนบริเวณโดยรอบที่ความถี่ 20 Hz ให้ข้อมูลประมาณ 1.39 ล้านจุดต่อวินาที มีระยะทางสูงสุดที่ตรวจจับได้ 100m และใช้งานได้จริงสูงสุดที่ 70 m

3) กล้องสี (Basler acA1600-60gc) รอบคันรถจำนวน 6 ตัว มีขนาดของภาพ 1600x900 โดยมีความถี่การจับภาพที่ 12 Hz

4) เลนส์กล้องแบบความยาวโฟกัสคงที่ 5.5 mm (Evetar Lens N118B05518W F1.8 f5.5mm 1/1.8")



รูปที่ 13 การจัดวางของเซนเซอร์ชุดข้อมูล nuscenes แหล่งที่มา : https://www.nuscenes.org/nuscenes#data-collection

5) เรดาร์ระยะไกล (Continental ARS 408-21) ความถี่การตรวจจับ 13 Hz โดยใช้คลื่นวิทยุความถี่ 77 GHz ระยะทางสูงสุดที่สามารถตรวจจับได้คือ 100 m และ สามารถใช้งานจริงได้สูงสุด 70 m

โดยในชุดข้อมูลดังกล่าวจะประกอบไปด้วยสถานการณ์ต่างๆถึง 1000 สถานการณ์ แบ่งออกเป็นสถานการณ์ที่ใช้ฝึกสอน 700 สถานการณ์, ใช้วัดผล 150 สถานการณ์, ใช้ทดสอบ 150 สถานการณ์ แบ่งออกเป็นชุดข้อมูลที่มีการระบุค่าความจริง (keyframe) ซึ่งประกอบไปด้วยชุดข้อมูลที่สุ่มมาที่ความถี่ 2 Hz และ ชุดข้อมูลทั่วไปประกอบไปด้วยชุดข้อมูลดิบทั้งหมด

2.3 การวัดผลการตรวจจับวัตถุ (Object detection evaluation)

ในการประมวลผลการตรวจจับวัตถุ จำเป็นที่จะต้องรู้จักกับ Precision และ Recall ก่อน ซึ่งก็ต้องมีความเข้าใจเกี่ยวกับ Confusion Matrix เบื้องต้น โดยใน Confusion Matrix จะเป็นเมทริกซ์ที่ประกอบไปด้วยตัวแปรทั้งหมด 4 ตัวนั่นคือ

1) True Positive หมายถึง การที่โมเดลทำนายว่าถูก/มี และ เป็นไปตามนั้นจริง

2) True Negative หมายถึง การที่โมเดลทำนายว่าไม่ถูก/ไม่มี และ เป็นไปตามนั้นจริง

3) False Positive หมายถึง การที่โมเดลทำนายว่าถูก/มี แต่ไม่เป็นไปตามนั้น

4) False Negative หมายถึง การที่โมเดลทำนายว่าไม่ถูก/ไม่มี แต่ไม่เป็นไปตามนั้น

จะสามารถเขียน Confusion matrix และ คำบรรยายได้ดังนี้

		ทำนาย	
		มี / ถูก	**ไม่มี / ไม่ถูก**
ความจริง	**มี / ถูก**	TP	FN
	ไม่มี / ไม่ถูก	FP	TN

และตัวแปรสองตัวที่ใช้วัดผลนั่นคือ

1) ความแม่นยำ (Precision) ซึ่งเป็นตัววัดว่า สิ่งที่ทำนายนั้นมีความถูกต้องกับความเป็นจริงมากแค่ไหน สามารถอธิบายเป็นสมการได้ดังนี้

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

2) ความระลึก *(Recall)* ซึ่งเป็นตัววัดว่า สามารถตรวจจับสิ่งที่ถูกหรือมีได้มากน้อยแค่ไหน ซึ่งถ้ายิ่งสูง หมายความว่ามีความสามารถในการตรวจสอบสิ่งที่สนใจมาก

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

บทที่ 3
การดำเนินการวิจัย

3.1 การประมวลผลก่อน (Data Pre-processing)

ในเบื้องต้น จะมีการคัดกรองจุดที่นอกเหนือจากช่วงที่ต้องการออก โดยมีช่วงที่ต้องการดังนี้

สำหรับชุดข้อมูล nuscenes ในแนวแกน x, y, z ของระบบพิกัดไลดาร์ [-51.2, 51,2], [-51.2, 51.2], [-5.0, 3.0] เมตร ตามลำดับ และ สำหรับชุดข้อมูล kitti ในแนวแกน x, y, z ของระบบพิกัดไลดาร์ [0, 69.12], [-39.68, 39.68], [-3, 1] เมตร ตามลำดับ

3.2 การเสริมชุดข้อมูล (Data Augmentation)

ในชุดข้อมูล nuscenes จะมีการสุ่มเสริมข้อมูลโดยมีวิธีการดังนี้

1) สุ่มกลับด้านพิกัดทั้งเฟรมตามแนวแกน x, y

2) สุ่มหมุนพิกัดทั้งเฟรมในช่วง [-45, 45] องศา

3) สุ่มเลื่อนพิกัดทั้งเฟรมในช่วง [-0.5, 0.5] ในทุกแนวแกน

4) สุ่มยืด หรือ หด พิกัดในทุกแนวแกนเป็นจำนวนเท่าในช่วง [0.9, 1.1] เท่า

และสำหรับชุดข้อมูล kitti จะมีการสุ่มเสริมข้อมูลโดยมีวิธีการดังนี้

1) สุ่มกลับด้านพิกัดทั้งเฟรมตามแนวแกน x เท่านั้น

2) สุ่มหมุนพิกัดทั้งเฟรมในช่วง [-45, 45] องศา

3) สุ่มยืด หรือ หด พิกัดในทุกแนวแกนเป็นจำนวนเท่าในช่วง [0.9, 1.05] เท่า

3.3 โครงสร้างโครงข่ายประสาทเทียมที่ใช้ (Neural Network Architecture)

3.3.1 โครงสร้างของ 3DSSD

ในส่วนของ 3D Backbone ประกอบไปด้วยส่วนของ SA ซึ่งมีรายละเอียดดังนี้

ตารางที่ 1 โครงสร้างของชั้น SA สำหรับ 3DSSD

ชั้นที่	ชนิดการสุ่ม	จำนวน Centroid	รัศมีการสุ่ม	จำนวนจุดสูงสุด	ลำดับชั้น MLP
1	D-FPS	4096	0.2 0.4	32 32	[16, 16, 32] [16, 16, 32]

			0.8	64	[32, 32, 64]
2	Fusion	อย่างละ 512	0.4 0.8 1.6	32 32 64	[64, 64, 128] [64, 64, 128] [64, 96, 128]
3	D-FPS และ F-FPS	256 และ 256	1.6 3.2 4.8	32 32 32	[128, 128, 256] [128, 192, 256] [128, 256, 256]
4	D-FPS Candidate Centroid	256	4.8 6.4	16 32	[256, 256, 512] [256, 512, 1024]

และในส่วนของ Dense Head จะประกอบไปด้วย

ตารางที่ 2 โครงสร้างของ Dense Head สำหรับ 3DSSD

ตัวแปรทำนายทำนาย	ชนิดของชั้น	ข้อมูลของชั้น	ฟังก์ชันสูญเสีย
ชนิดของวัตถุ (Class)	FCN	[256, 256, cls]	Cross Entropy Loss
ตำแหน่งของวัตถุแบบ bin-based	FCN	[256, 256, n]	Cross Entropy Loss
ตำแหน่งของวัตถุเศษเหลือ (Residual)			Weight Smooth L1
องศาเฉียง (Yaw Angle)			

3.3.2 โครงสร้างของ PointPillar

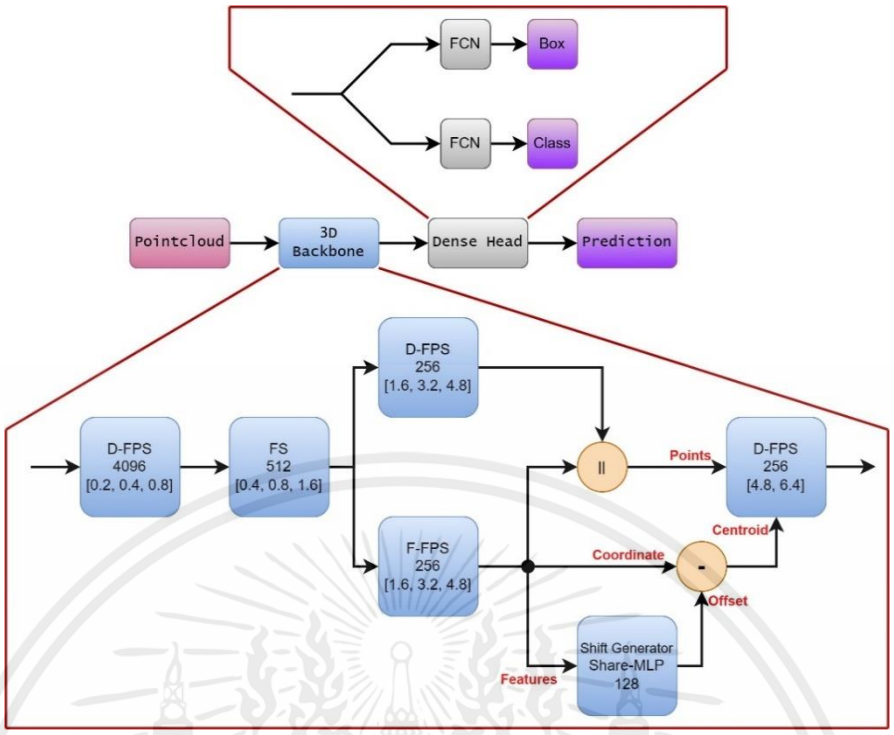
ในส่วนของ Pillar Encoder แต่ละจุดภายในเสาหลักจะมีคุณสมบัติดังนี้

$$p_i = \{x_i, y_i, z_i, (x_i - \overline{x_p}), (y_i - \overline{y_p}), (z_i - \overline{z_p}), (x_i - x_{pc}), (y_i - y_{pc}), (z_i - z_{pc})\}$$

โดย x_i คือ ตำแหน่งของจุดตามแนวแกน x

$\overline{x_p}$ คือ ค่าเฉลี่ยตำแหน่งในแนวแกน x ของทุกจุดภายในเสาหลักนั้น

x_{pc} คือ ตำแหน่งของศูนย์กลางเสาหลักในแนวแกน x



รูปที่ 14 โครงสร้างของ 3DSSD ที่ใช้

และสกัดคุณสมบัติด้วย FCN ขนาด 64 คุณสมบัติ จากนั้นก็แปลงเป็นข้อมูลสองมิติ โดยการนำคุณสมบัติในเสาหลักใดๆ ไปใส่ไว้ในตำแหน่งเสาหลัก ซึ่งจะกลายมาเป็นพิกเซลในภายหลัง แต่เสาหลักอื่นๆที่ไม่มีข้อมูลก็จะใช้การเติมศูนย์แทน

ในส่วนของ 2D Backbone ประกอบไปด้วยชั้นคอนโวลูชันซึ่งมีรายละเอียดดังนี้

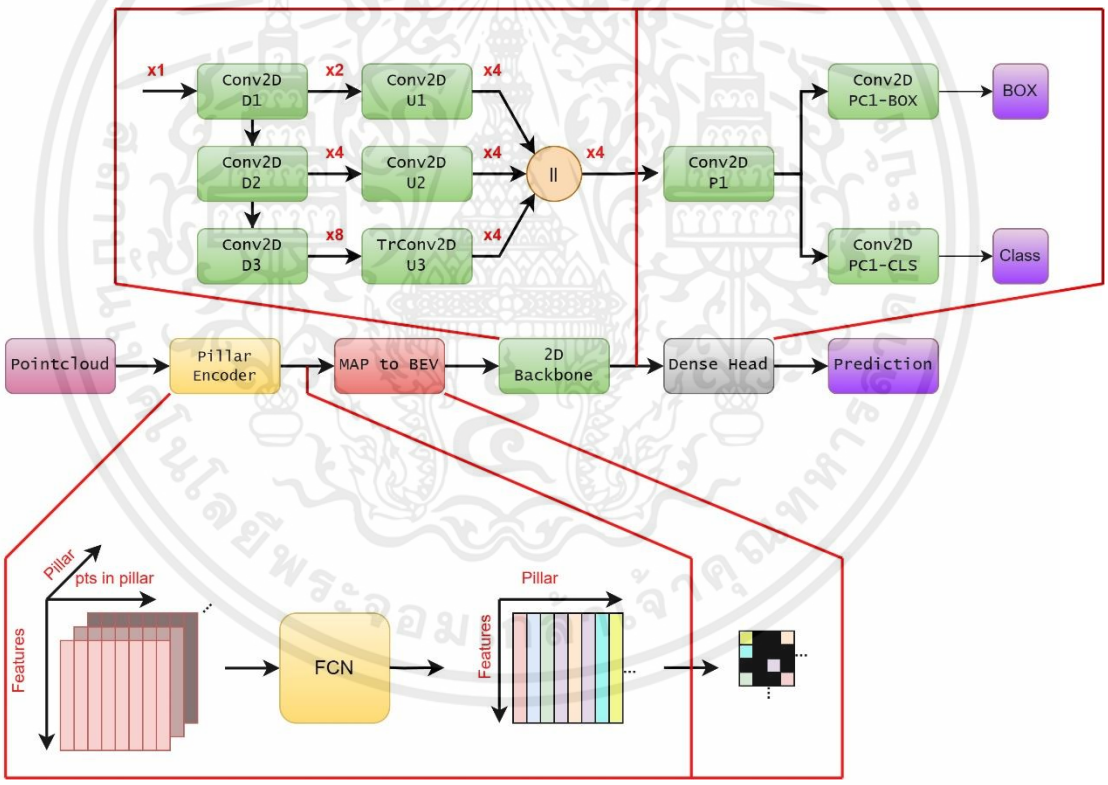
ตารางที่ 3 โครงสร้างของ 2D Backbone สำหรับ PointPillar

ชื่อชั้น	ชนิด	Filter Sz	# Filter	Stride	Padding	จำนวนชั้น
D1	Conv2D	[3,3]	64	2	1	1
	Conv2D	[3,3]	64	1	1	3
D2	Conv2D	[3,3]	128	2	1	1
	Conv2D	[3,3]	128	1	1	5
D3	Conv2D	[3,3]	256	2	1	1
	Conv2D	[3,3]	256	1	1	5
U1	Conv2D	[3, 3]	128	2	1	1
U2	Conv2D	[3, 3]	128	1	1	1
U3	TrConv2D	[3, 3]	128	2	1	1

ในส่วนของ Dense Head ประกอบด้วย Anchor Head หลายๆตัวรวมกัน ซึ่งในแต่ละอันนั้นประกอบไปด้วย

ตารางที่ 4 โครงสร้างของ Dense Head สำหรับ PointPillar

ตัวแปรทำนายทำนาย	ชนิดของชั้น	ข้อมูลของชั้น	ฟังก์ชันสูญเสีย
ชนิดของวัตถุ (Class)	Conv2D	[64, anchor]	Cross Entropy Loss
ตำแหน่งศูนย์กลางชดเชย (Center Offset)	Conv2D	[256, 256, 7]	Weighted Smooth L1
ขนาดของวัตถุ (Dimension)			
องศาเฉียง (Yaw Angle)			



รูปที่ 15 โครงสร้างของ PointPillar ที่ใช้

3.3.3 โครงสร้างของ SECOND

ในส่วนของ Voxel Features Encoder จะเป็นการตั้งค่าคุณสมบัติของแต่ละ Voxel ด้วยข้อมูลดังนี้

$$V_i = \{\bar{x}_i, \bar{y}_i, \bar{z}_i\}$$

โดย $\bar{x}_i$ คือ ค่าเฉลี่ยของพิกัดในแนวแกน *x* ของจุดภายใน *Voxel* นั้นๆ

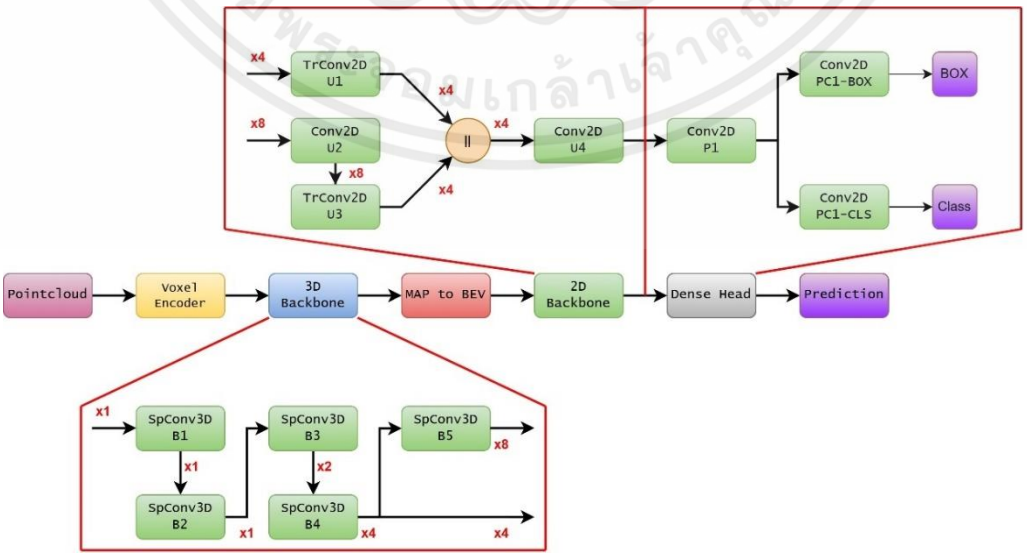
ในส่วนของ *3D Backbone* จะประกอบไปด้วนชั้นคอนโวลูชันดังนี้

ตารางที่ 5 โครงสร้างของ 3D Backbone สำหรับ SECOND

ชื่อชั้น	ชนิด	Filter Sz	# Filter	Stride	Padding	จำนวนชั้น
B1	SmConv3D	[3, 3, 3]	16	1	1	1
B2	SmConv3D	[3, 3, 3]	16	1	1	1
B3	SpConv3D	[3, 3, 3]	32	2	1	1
	SmConv3D	[3, 3, 3]	32	1	1	2
B4	SpConv3D	[3, 3, 3]	64	2	1	1
	SmConv3D	[3, 3, 3]	64	1	1	2
B5	SpConv3D	[3, 3, 3]	64	2	1	1
	SmConv3D	[3, 3, 3]	64	1	1	2

ในส่วนของ MAP 2 BEV จะเป็นการนำชุดข้อมูลที่ได้จาก 3D Backbone ซึ่งมีขนาดของคุณสมบัติเป็นลักษณะ 4 มิติ $[x, y, 2, C]$ *(3* มิติในเชิงพื้นที่ และ อีก 1 มิติในเชิงของช่อง*)* มาแปลงเป็นข้อมูล 3 มิติ $[x, y, 2C]$ *(2* มิติในเชิงพื้นที่ และ อีก 1 มิติในเชิงของช่อง*)* โดยการนำข้อมูลในมิติที่ 3 นับจากทางซ้ายเข้ามาต่อกันในตำแหน่งเชิงพื้นที่นั้นๆ (Concatenate)

ในส่วนของ Dense Head ประกอบด้วย Anchor Head หลายๆตัวรวมกัน ซึ่งเหมือนกับโครงสร้างด้านบน



รูปที่ 16 โครงสร้างของ SECOND ที่ใช้

3.3.4 โครงสร้างของ CenterPoint-PointPillar

ในโครงสร้างของ CenterPoint แบบ PointPillar จะเป็นโครงสร้างที่คล้ายกับ PointPillar เดิม เพียงแต่จะเปลี่ยนในส่วนของ Piilar Encoder เป็นแบบ Dynamic Encoder และในส่วนของ Dense Head ที่จากเดิม Anchor Multi Head ถูกเปลี่ยนเป็น Center Multi Head แทน

3.3.5 โครงสร้างของ PillarNet

ในส่วนของ 3D Backbone จะมีความคล้ายคลึงกับ SECOND เพียงแต่จะเปลี่ยนจากการคอนโวลูชันสามมิติเป็นสองมิติ และมีจำนวนฟิลเตอร์ในแต่ละชั้นที่ต่างกันออกไป ซึ่งกระบวนการทั้งหมดจะทำด้วย Sparse และ Sub Manifold Convolution

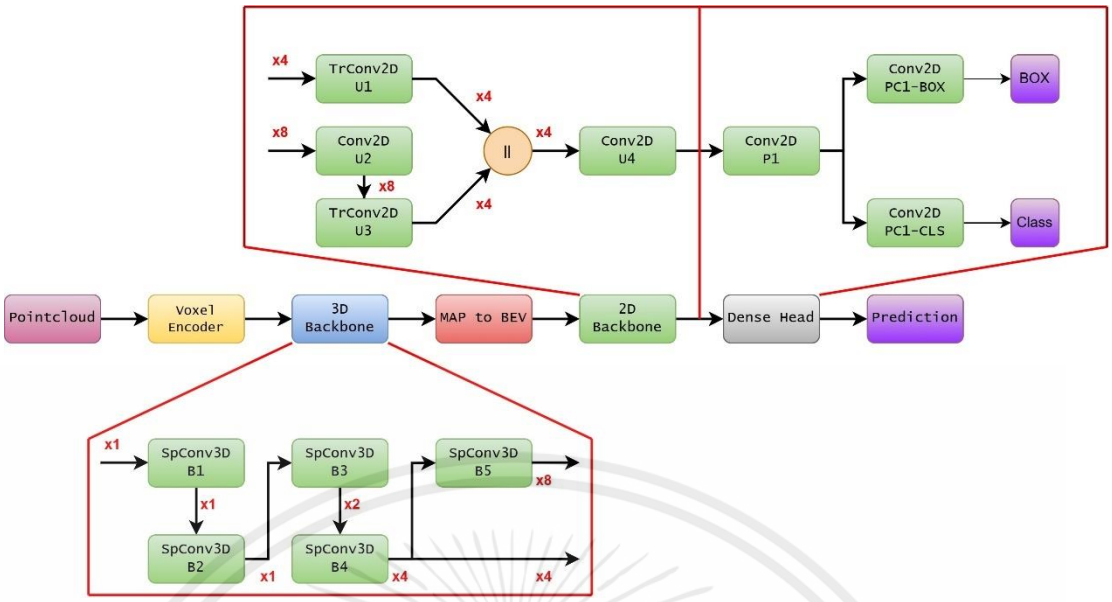
ตารางที่ 6 โครงสร้างของ 3D Backbone สำหรับ PillarNet

ชื่อชั้น	ชนิด	Filter Sz	# Filter	Stride	Padding	จำนวนชั้น
B1	SmConv2D	[3, 3]	32	1	1	2
B2	SpConv2D	[3, 3]	64	2	1	1
	SmConv2D	[3, 3]	64	1	1	2
B3	SpConv2D	[3, 3]	128	2	1	1
	SmConv2D	[3, 3]	128	1	1	2
B4	SpConv2D	[3, 3]	256	2	1	1
	SmConv2D	[3, 3]	256	1	1	2
B5	Conv2D	[3, 3]	256	2	1	1
	Conv2D	[3, 3]	256	1	1	2

ในส่วนของ 2D Backbone จะแตกต่างไปจากโมเดลที่กล่าวมาด้านบนในระดับหนึ่ง ซึ่งสามารถอธิบายได้ดังนี้

ตารางที่ 7 โครงสร้างของ 2D Backbone สำหรับ PillarNet

ชื่อชั้น	ชนิด	Filter Sz	# Filter	Stride	Padding	จำนวนชั้น
U1	TrConv2D	[3, 3]	256	1	1	1
U2	Conv2D	[3, 3]	256	1	1	6
U3	TrConv2D	[3, 3]	256	2	1	15
U4	Conv2D	[3, 3]	256	1	1	6



รูปที่ 17 โครงสร้างของ PillarNet ที่ใช้

3.4 การฝึกสอนโมเดล (Model Training)

ในโครงงานฉบับนี้ จะทำการฝึกสอนโมเดลในส่วนของ PillarNet ขึ้นมาเอง และในโมเดลอื่นๆ จะใช้ค่าน้ำหนักที่ผ่านการฝึกสอนมาแล้ว (Pre-trained Model) โดยมีรายละเอียดการฝึกสอนดังนี้

1) ใช้หน่วยประมวลผลกราฟฟิกสองชุดซึ่งคือ nVidia GTX 970 (VRAM 4 GB) และ nVidia GTX 1060Ti (VRAM 6 GB)

2) ใช้ขนาดของ Mini-Batch ที่ 2

3) ใช้อัตราการเรียนรู้ (Learning Rate) ที่ 0.01

4) ใช้ค่าโมเมนตัม 0.9

5) ใช้จำนวน Epoch ทั้งหมด 50 แต่ฝึกสอนจริงแค่ 20 Epoch

3.5 คะแนนการวัดผลชุดข้อมูล nuscenes (nuscenes evaluation score)

ในการวัดผลชุดข้อมูล nuscenes จะมีค่าคะแนนที่ทางผู้จัดทำชุดข้อมูลได้ให้ไว้เพิ่มเติมสำหรับวัดผลการตรวจจับวัตถุที่มีผลการตรวจจับเป็นแบบ True Positive ดังนี้

1) คะแนนความผิดพลาดการเคลื่อนตำแหน่งเฉลี่ย (Average Translation Error) ซึ่งคำนวณได้จากระยะห่างระหว่างศูนย์กลางวัตถุที่ตรวจจับได้กับความเป็นจริง

2) คะแนนความผิดพลาดขนาด (Average Scale Error) คำนวณได้จาก 1 – IoU โดยทำการจัดวางกล่องระบุตัววัตถุที่ได้จากการทำนายให้ตรงกับลักษณะการจัดวางที่แท้จริง ไม่คำนึงถึงคตวามผิดพลาดในการวางตำแหน่งและองศา

3) คะแนนความผิดพลาดการเรียงตัว (Average Orientation Error) คำนวณได้จากความผิดพลาดของมุมเฉียงสำหรับการจัดวางกล่องระบุวัตถุ

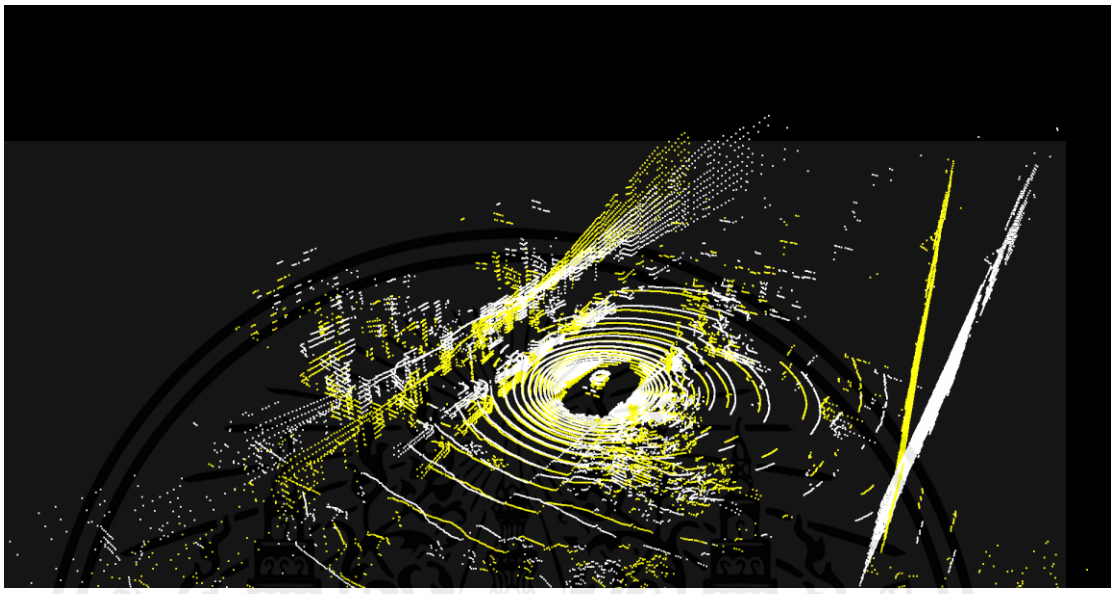
4) คะแนนความผิดพลาดอัตราเร็ว (Average Velocity Error) คำนวณได้จากความผิดพลาดของความเร็ววัตถุที่ทำนายกับความเป็นจริง

โดยในโครงงานฉบับนี้จะมุ่งเน้นไปที่คะแนนความผิดพลาดตำแหน่ง, คะแนนความผิดพลาดขนาด, และ คะแนนความผิดพลาดการเรียงตัวเป็นหลัก

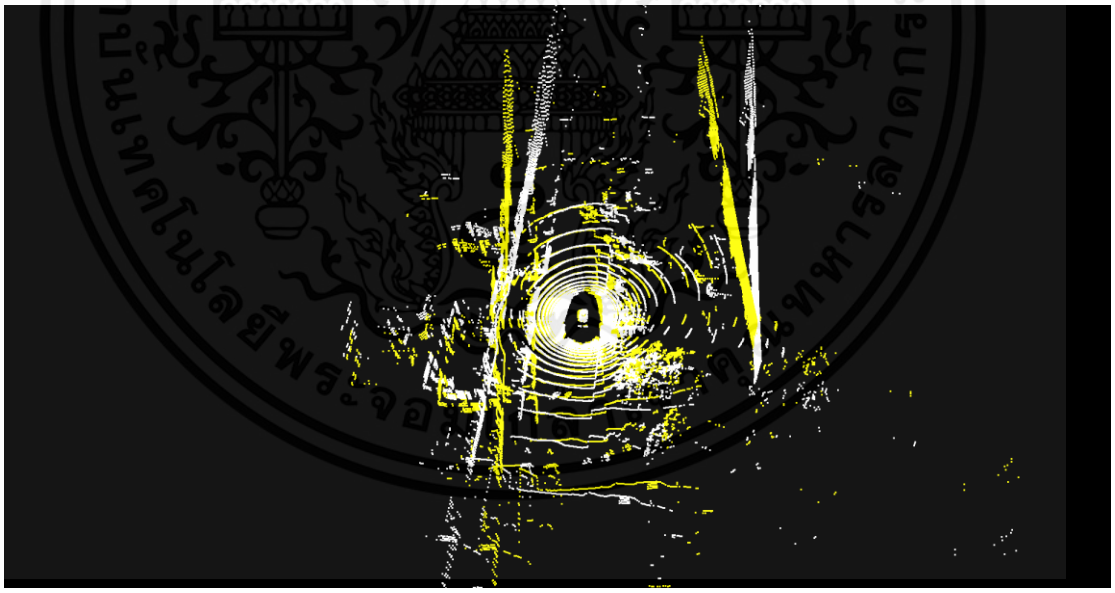
บทที่ 4

ผลการทดลอง

4.1 ผลลัพธ์จากการเสริมข้อมูล (Data Augmentation results)

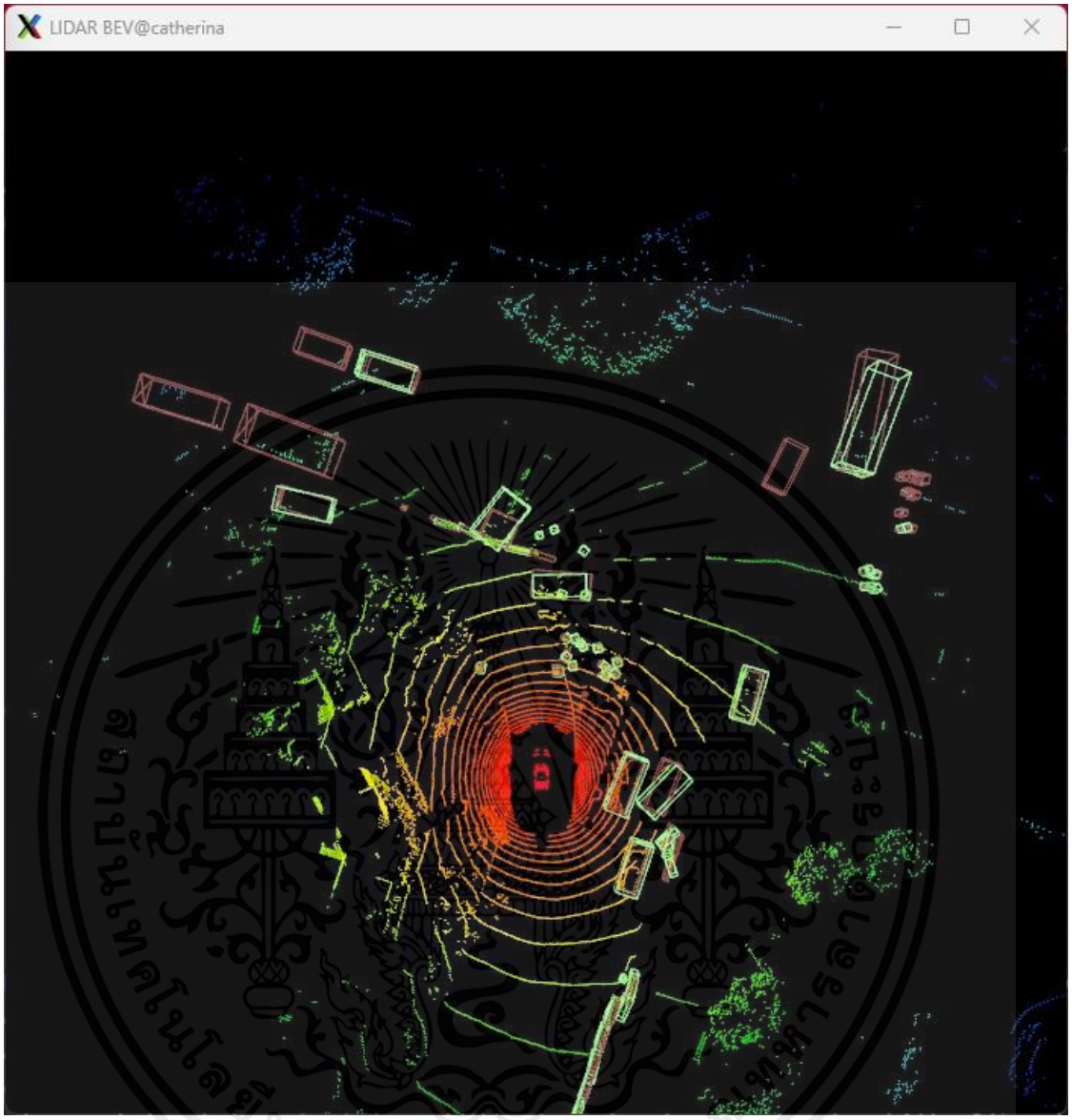


รูปที่ 18 ผลลัพธ์การเสริมข้อมูลในมุมเฉียง โดยสีขาวและเหลืองคือข้อมูลก่อนและหลังเสริมตามลำดับ

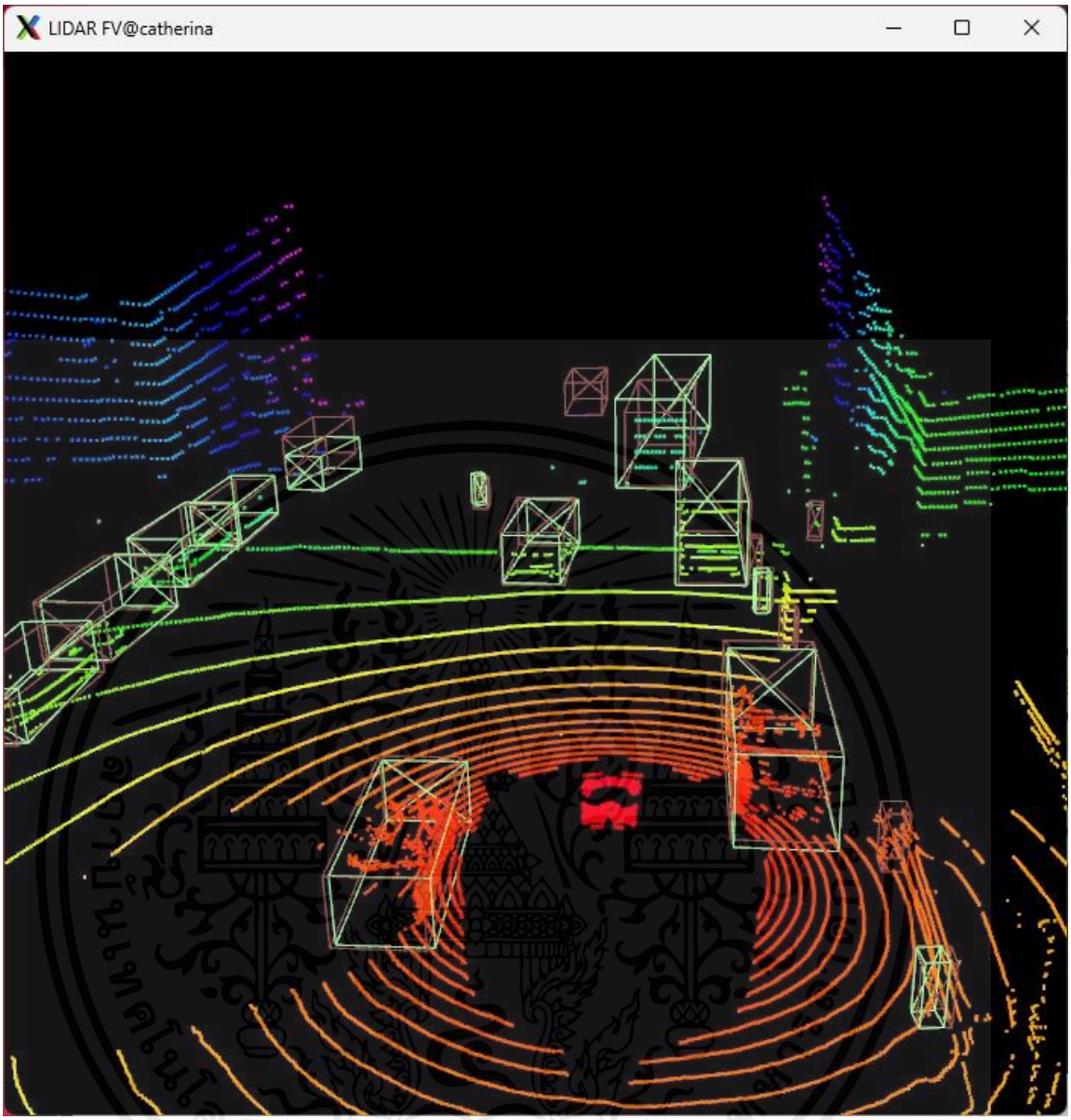


รูปที่ 19 ผลลัพธ์การเสริมข้อมูลในมุมบน โดยสีขาวและเหลืองคือข้อมูลก่อนและหลังเสริมตามลำดับ

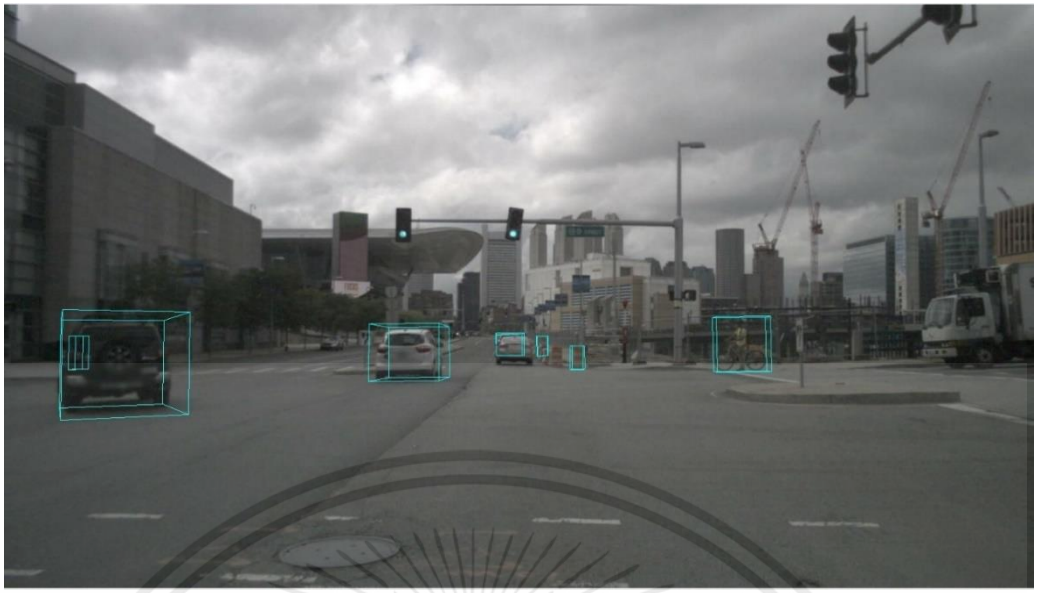
4.2 ผลลัพธ์การตรวจจับวัตถุ (Object detection result)



รูปที่ 20 ผลลัพธ์การตรวจจับวัตถุในมุมบน สีแดงและเขียว คือ กล่องความจริงและการทำนายตามลำดับ



รูปที่ 21 ผลลัพธ์การตรวจจับวัตถุมุมมองหน้ารถ สีแดงและเขียว คือ กล่องความจริงและการทำนายตามลำดับ

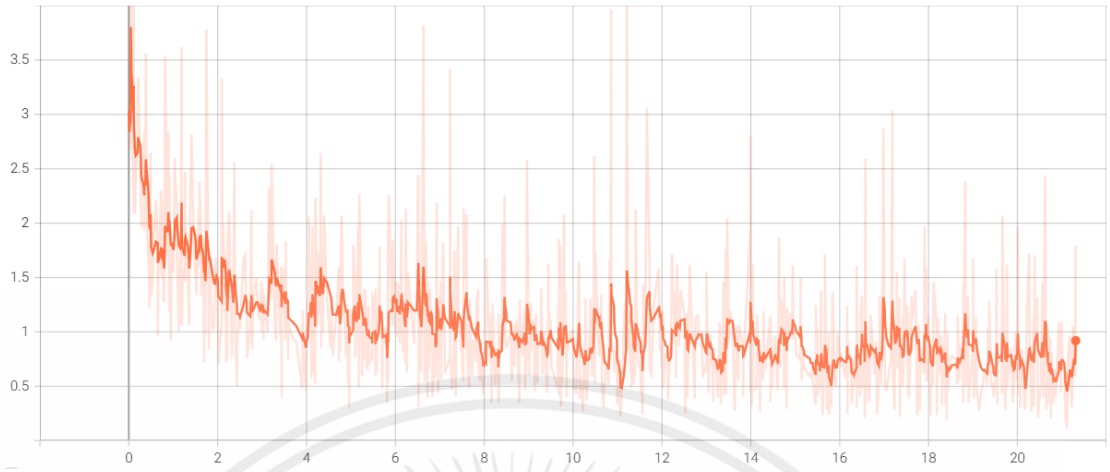


รูปที่ 22 มุมมองจากกล้องหน้ารถ และ กล่องที่ได้จากการทำนาย

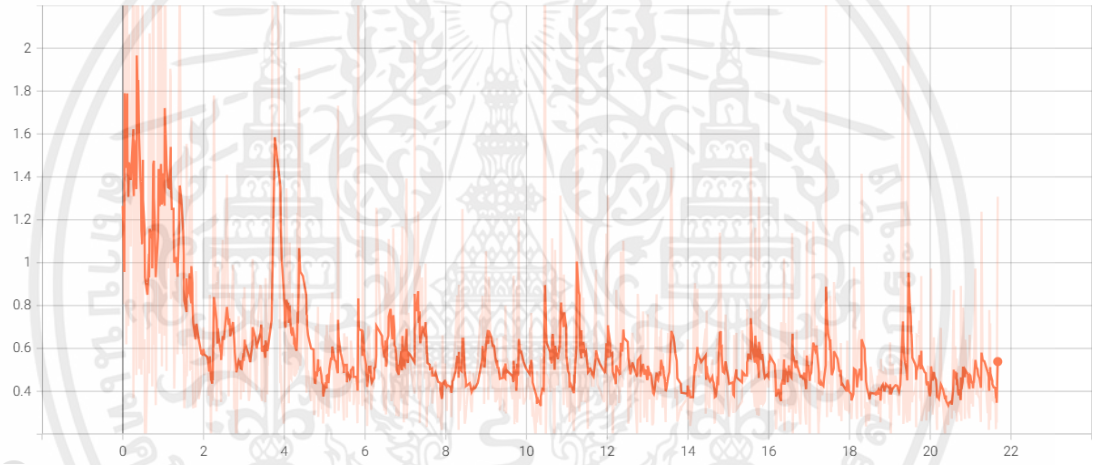
รูปที่ 23 มุมมองจากกล้องหน้ารถ และ กล่องที่ได้จากการทำนาย

ตารางที่ 8 คะแนนการตรวจจับวัตถุโดยใช้โมเดลต่างๆ

โมเดลที่ใช้	mATE	mASE	mAOE	mAP	# Param
3DSSD	37.45	28.15	40.69	30.25	2.81M
PointPillar	33.87	26.00	28.74	44.63	6.08M
SECOND	31.15	25.51	26.64	62.29	9.04M
CenterPoint-PP	31.13	26.04	42.92	60.70	5.99M
PillarNet	34.36	27.29	53.96	39.87	15.25M



รูปที่ 24 ค่าสูญเสียญ (Loss) จากการฝึกสอน PillarNet ของผังความร้อน (Heat Map) สำหรับการทำนายรถ



รูปที่ 25 ค่าสูญเสีย (Loss) จากการฝึกสอน PillarNet ของการระบุขนาดกล่องสำหรับการทำนายรถ

บทที่ 5
สรุป และข้อเสนอแนะ

จากโครงข่ายประสาทเทียมทั้งหมด พบว่า SECOND สามารถทำนายได้อย่างแม่นยำเมื่อเทียบกับโมเดลอื่นๆ อย่างไรก็ตาม การใช้งานทรัพยากรการคำนวณของ SECOND นั้นค่อนข้างสูงเป็นอย่างมาก จึงอาจไม่เหมาะสมกับการใช้งานบนระบบฝังตัว อย่างไรก็ตาม ในโมเดลกลุ่มเสาหลัก (Pillar) นั้นมีความเร็วในการประมวลผลค่อนข้างมาก และ มีความแม่นยำในระดับที่สูง แต่ก็ยังไม่สูงเพียงพอที่สามารถใช้ในระบบจริงๆได้ ในโครงงานฉบับนี้ จึงเสนอแนวทางในการพัฒนาดังนี้

1) การพิจารณาใช้ข้อมูลไลดาร์เชิงเวลาเข้ามาร่วมด้วย กล่าวคือ การพิจารณาข้อมูลที่ได้จากวงรอบการสแกนไลดาร์มากกว่า 1 วงรอบซึ่งจะทำให้จำนวนข้อมูลที่มากขึ้น อย่างไรก็ตามอาจมีผลของการคลาดเคลื่อนของตำแหน่งข้อมูลอันเนื่องมาจากการเคลื่อนที่ของวัตถุต่างๆ ซึ่งอาจต้องมีการชดเชยตำแหน่งที่คลาดเคลื่อนนี้ไปด้วย

2) การใช้การทำนายแบบความน่าจะเป็น โดยตำแหน่งของวัตถุ และ มิติของวัตถุอาจกำหนดได้โดยการใช้ฟังก์ชันความน่าจะเป็นซึ่งเป็นค่าที่ต่อเนื่อง และ เมื่อใช้ความน่าจะเป็นเหล่านี้มาพิจารณาในแต่ละเฟรมข้อมูลของไลดาร์ เราสามารถนำความน่าจะเป็นในแต่ละเฟรมมารวมกันเพื่อให้ได้ความแม่นยำของตำแหน่งที่เพิ่มขึ้นได้

3) การใช้ Graph Neural Network เข้ามาพิจารณาผลของการเกี่ยวข้องกันระหว่างจุด เนื่องจากในโมเดลที่กล่าวมาข้างต้นไม่ได้คำนึงถึงการเกี่ยวข้องกันระหว่างจุดแต่ละจุดมากนัก การนำ Graph Neural Network ในการสกัดความเกี่ยวข้องกันแต่ละจุดซึ่งจะช่วยเพิ่มคุณสมบัติเฉพาะของกลุ่มจุดนั้นๆเข้ามา

4) นำคุณสมบัติเชิงกายภาพอื่นๆเข้ามาร่วมด้วย เช่น สีของวัตถุที่ไลดาร์ตรวจจับได้ ด้วยเหตุผลที่ว่าวัตถุชนิดใดๆอาจมีคุณสมบัติเชิงกายภาพเช่นสีที่คล้ายๆกัน

ในส่วนของโมเดล PillarNet ที่ทำการฝึกสอนขึ้นมาเองซึ่งใช้ขนาดของ Mini-batch แค่ 2 อาจทำให้การฝึกสอนเป็นไปอย่างไม่ราบเรียบซึ่งสังเกตุได้จากกราฟฟังก์ชันสูญเสียที่มีค่าความสูญเสียที่แตกต่างกันมาก กล่าวคือ ตัวโครงสร้างประสาทเทียมได้เจอกับชุดข้อมูลที่น้อยเกินไปในแต่ละวงรอบการฝึกสอน ซึ่งเป็นข้อจำกัดที่มาจากปริมาณหน่วยความจำหน่วยประมวลผลกราฟิกที่ไม่พอ

เอกสารอ้างอิง

[1] Y. Zhou *et al.*, “End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds,” *Proc Mach Learn Res*, vol. 100, pp. 923–932, Oct. 2019, Accessed: Mar. 05, 2024. [Online]. Available: https://arxiv.org/abs/1910.06528v2

[2] T. Yin, X. Zhou, and P. Krähenbühl, “Center-based 3D Object Detection and Tracking,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 11779–11788, Jun. 2020, doi: 10.1109/CVPR46437.2021.01161.

[3] J. Li, C. Luo, and X. Yang, “PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2023-June, pp. 17567–17576, May 2023, doi: 10.1109/CVPR52729.2023.01685.

[4] W. Zheng, W. Tang, S. Chen, L. Jiang, and C. W. Fu, “CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud,” *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, vol. 4B, pp. 3555–3562, Dec. 2020, doi: 10.1609/aaai.v35i4.16470.

[5] Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, Nov. 2017, doi: 10.1109/CVPR.2018.00472.

[6] G. Shi, R. Li, and C. Ma, “PillarNet: Real-Time and High-Performance Pillar-based 3D Object Detection,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13670 LNCS, pp. 35–52, May 2022, doi: 10.1007/978-3-031-20080-9_3.

[7] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, pp. 3354–3361, 2012, doi: 10.1109/CVPR.2012.6248074.

[8] H. Caesar *et al.*, "nuScenes: A multimodal dataset for autonomous driving," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 11618–11628, Mar. 2019, doi: 10.1109/CVPR42600.2020.01164.

[9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 12689–12697, Dec. 2018, doi: 10.1109/CVPR.2019.01298.

[10] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D Single Stage Object Detector," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 11037–11045, Feb. 2020, doi: 10.1109/CVPR42600.2020.01105.

[11] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 770–779, Dec. 2018, doi: 10.1109/CVPR.2019.00086.

[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 77–85, Dec. 2016, doi: 10.1109/CVPR.2017.16.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," *Adv Neural Inf Process Syst*, vol. 2017-December, pp. 5100–5109, Jun. 2017, Accessed: Mar. 05, 2024. [Online]. Available: https://arxiv.org/abs/1706.02413v1

[14] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, and Y. Guo, “Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June, pp. 18931–18940, Mar. 2022, doi: 10.1109/CVPR52688.2022.01838.

[15] M. Simon, S. Milz, K. Amende, and H.-M. Gross, “Complex-YOLO: Real-time 3D Object Detection on Point Clouds,” Mar. 2018, Accessed: Mar. 05, 2024. [Online]. Available: https://arxiv.org/abs/1803.06199v2

[16] Y. Yan, Y. Mao, and B. Li, “SECOND: Sparsely Embedded Convolutional Detection,” *Sensors 2018, Vol. 18, Page 3337*, vol. 18, no. 10, p. 3337, Oct. 2018, doi: 10.3390/S18103337.

[17] W. Zheng, W. Tang, L. Jiang, and C. W. Fu, “SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 14489–14498, Apr. 2021, doi: 10.1109/CVPR46437.2021.01426.