The Sound Demixing for thai traditional music

นาย ธนกร ปรารมย์ 6452300113
นาย สัณห์พิชัยกรณ์ แจ้งสว่าง 6452300407

การศึกษาค้นคว้าอิสระเสนอคณะวิศวกรรมศาสตร์และเทคโนโลยี
สถาบันการจัดการปัญญาภิวัฒน์
เพื่อเป็นส่วนหนึ่งของการศึกษาหลักสูตรวิศกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์
พ.ศ.๒๕๖๖ ลิขสิทธิ์เป็นของการจัดการปัญญาภิวัฒน์

The Sound Demixing for thai traditional music

Mister Tanakorn Prarom 6452300113
Mister Sanpishaikorn Jaensawang 6452300407

A Senior Project Submitted in Partial Fulfilment of the
Requirements For the Degree of Bachelor of Computer Engineering
Faculty of Engineering and Technology Academic Year 2023
Copyright of Panyapiwat Institute of Management

เรื่อง	The Sound Demixing for thai traditional music
โดย	**นาย ธนกร ปรารมย์ 6452300113**
	นาย สัณห์พิชัยกรณ์ แจ้งสว่าง 6452300407
คณะ	**วิศวกรรมศาสตร์และเทคโนโลยี**
สาขาวิชา	**วิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์**
อาจารย์ที่ปรึกษา	**รศ.ดร.ปริญญา สงวนสัตย์**

ได้รับการอนุมัติเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิตสาขาวิชาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์

.....คณบดีคณะ วิศวกรรมศาสตร์และเทคโนโลยี

(รศ.ดร.พิสิษฐ์ ชาญเกียรติก้อง)

.....ประธานกรรมการ

(รศ.ดร.ปริญญา สงวนสัตย์)

.....กรรมการ

(ผศ.ดร.อดิศร แขกซอง)

.....กรรมการ

(ดร.ติณณภพ ดินดำ)

.....หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์

(รศ.ดร.ปริญญา สงวนสัตย์)

ชื่อเรื่อง The Sound Demixing for thai traditional music
ชื่อผู้วิจัย นาย ธนกร ปรารมย์
นาย สัณห์พิชัยกรณ์ แจ้งสว่าง
ชื่ออาจารย์ที่ปรึกษา รศ.ดร.ปริญญา สงวนสัตย์
ชื่อปริญญา วิศวกรรมศาสตร์และเทคโนโลยีสาขาวิศวกรรมคอมพิวเตอร์
ปีการศึกษา พ.ศ.2567

บทคัดย่อ

เพลงไทยเดิมเปรียบเสมือนมรดกทางวัฒนธรรมอันล้ำค่าของประเทศที่ควรค่าแก่การอนุรักษ์ เทคโนโลยี Sound Demixing เข้ามาตอบโจทย์ความท้าทายนี้ผ่านการแยกเสียงเครื่องดนตรีไทยออกจากเพลงไทยเดิม ช่วยให้การศึกษา วิเคราะห์ และอนุรักษ์เพลงไทยเดิมมีประสิทธิภาพมากยิ่งขึ้น

Sound Demixing อาศัยเทคนิค DSP และ AI แยกเสียงต่างๆ ออกจากส่วนผสมเสียง วิธีการที่นิยมได้แก่CNN, ICA, NMF และ DNNs งานวิจัยชิ้นนี้มุ่งเน้นไปที่การพัฒนา Sound Demixing ของเพลงไทยเดิมโดยใช้เทคนิคเหล่านี้ เพื่อทำการแยกเสียงผ่านความท้าทาย เช่น เสียงรบกวนความซับซ้อนของเครื่องดนตรีไทยและข้อมูลการฝึกที่จำกัด

โดยสรุป Sound Demixing เปรียบเสมือนเครื่องมือทรงพลังสำหรับการศึกษาวิเคราะห์และอนุรักษ์เพลงไทยเดิมและ จะมีบทบาทสำคัญในการสืบสานมรดกทางวัฒนธรรมอันล้ำค่านี้ต่อไป

Title	The Sound Demixing for thai traditional music
Author's Name	Mister Tanakorn Prarom
	Mister Sanpishaikorn Jaensawang
Advisor	Assoc. Prof. Dr. Parinya Sanguansat
Degree	Bachelor of Engineering
	Computer Engineering and Artificial Intelligence
Academic Year	2023

Abstract

Traditional Thai music is like a valuable cultural heritage that is worth preserving. Sound Demixing technology comes in to answer this challenge. Through separating the sounds of Thai musical instruments from traditional Thai songs. Helps to study, analyze and preserve traditional Thai songs more efficiently.

Sound Demixing uses DSP and AI techniques to separate different sounds. out of the sound mix Popular methods include CNN, ICA, NMF and DNNs. This research focuses on the development of Sound Demixing of traditional Thai songs. Using these techniques To separate sounds through challenges such as noise and the complexity of Thai musical instruments and limited training data

In summary, Sound Demixing is a powerful tool for studying, analyzing, and preserving traditional Thai music and It will play an important role in continuing this valuable cultural heritage.

กิตติกรรมประกาศ

โปรเจกต์ The Sound Demixing for thai traditional music นี้สำเร็จลงได้ด้วยดี เนื่องจากการให้ความช่วยเหลือและคำแนะนำของรศ.ดร.ปริญญา สงวนสัตย์ อาจารย์ที่ปรึกษางานวิจัย ซึ่งเป็นผู้ให้ความรู้เกี่ยวกับเนื้อหาโครงงานและการตรวจสอบพร้อมทั้งให้คำแนะนำ ชี้แนะแนวทางการศึกษาตลอดจนให้ความกรุณาในการตรวจทานแก้ไขโครงงานฉบับนี้จนทำให้ประสบความสำเร็จ จึงต้องขอขอบพระคุณอย่างสูงมา ณ โอกาศนี้

สุดท้ายนี้ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่า โครงงานเล่มนี้จะเป็นประโยชน์ต่อผู้ที่สนใจไม่มากก็น้อย หากมีส่วนใดที่บกพร่องหรือมีข้อผิดพลาดประการใด ทางผู้จัดทำต้องขออภัยไว้ ณ ที่นี้ ด้วย

คณะผู้จัดทำ

(ธนกร,สัณห์พิชัยกรณ์)

สารบัญ

หน้า

บทคัดย่อ 3

บทคัดย่อ (ภาษาอังกฤษ) 4

กิตติกรรมประกาศ 5

สารบัญ 6

สารบัญรูป 8

สารบัญกราฟ 10

บทที่ 1 11

1.1 ที่มาและความสำคัญ 11

1.2 วัตถุประสงค์ของการวิจัย 12

1.3 ขอบเขตการวิจัย 12

1.4 ประโยชน์ที่คาดว่าจะได้รับ 12

1.5 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินการ 13

1.6 นิยามคำศัพท์เฉพาะ 14

บทที่ 2 15

2.1 ทฤษฎีที่เกี่ยวข้อง 15

2.2 เอกสารงานวิจัยที่เกี่ยวข้อง 19

บทที่ 3 26

3.1 เครื่องมือที่ใช้ในการพัฒนา 26

3.2 เตรียมชุดข้อมูล 27

3.3 สร้าง และ ฝึกโมเดล 33

3.4 ทดสอบประสิทธิภาพโมเดล 38

บทที่ 4 44

4.1 การทดสอบและการตรวจสอบการวิจัย 44

4.2 สรุปผลการทดสอบ 49

สารบัญ (ต่อ)

หน้า

บทที่ 5 52
5.1 สรุปผลการวิจัย 52
5.2 การอภิปราย 52
5.3 วิเคราะห์ปัญหาจากการดำเนินการวิจัย 53
5.4 แนวทางและข้อเสนอแนะในการพัฒนาต่อในอนาคต 53
บรรณานุกรม 54

สารบัญรูป

หน้า

ภาพที่ 2.1.1.3.1 แสดงการจำแนกตระกูลเครื่องดนตรีตามประเภทของวงดนตรี 16
ภาพที่ 2.1.2.6.1 แสดงความแตกต่างระหว่างการสุ่มสัญญาณ (sampling rate) 17
ที่ 8 kHz และ 44.1 kHz
ภาพที่ 2.1.3.1 รูปแบบของเซลล์ประสาท (Neuron model) 18
ภาพที่ 2.2.2.1 ตารางแสดงผลการเปรียบเทียบ model ด้วย SDR metric 19
ภาพที่ 2.2.3.1 ตารางแสดงผลการเปรียบเทียบ model ด้วย SDR metric 20
ทั้งแบบเสียงร้องและเสียงเครื่องดนตรี
ภาพที่ 2.2.4.1 ตารางแสดงผลการเปรียบเทียบ model BSRNN กับ model 21
ใน state of the art ด้วย SDR metric
ภาพที่ 2.2.5.1 อธิบายโครงสร้างโมเดล Feed-forward และ โมเดล BLSTM 21
ภาพที่ 2.2.5.2 เปรียบเทียบประสิทธิภาพโมเดลแยกเสียงดนตรี 22
โดยใช้ชุดข้อมูล DSD100
ภาพที่ 2.2.6.1 โครงสร้างของโมเดล TFC-TDF-U-Net v2 22
ภาพที่ 2.2.6.2 การเปรียบเทียบประสิทธิภาพของโมเดล 23
ภาพที่ 2.2.1 การเปรียบเทียบโมเดล 25
ภาพที่ 3.2.1.1 แสดงรูปแบบไฟล์ในชุดข้อมูล MUSDB18 28
ภาพที่ 3.2.5.1.1 แสดงฟังก์ชั่นตัดเสียงเครื่องดนตรีทั้ง 3 30
ภาพที่ 3.2.5.3.1 แสดงฟังก์ชั่น Augmentation ทั้ง 3 30
ภาพที่ 3.2.5.4.1 แสดงฟังก์ชั่นการจัดการโฟลเดอร์ DSTHAI และ ไฟล์เสียงผสม 31
ภาพที่ 3.2.5.4.2 แสดงรูปแบบโฟลเดอร์ DSTHAI 32
ที่เก็บชุดข้อมูลสำหรับฝึกโมเดลไว้
ภาพที่ 3.2.6.2.1 แสดงฟังก์ชันการโหลดข้อมูลและสร้างลำดับ 32
ภาพที่ 3.2.6.3.1 แสดงฟังก์ชันการแบ่งชุดข้อมูล 33
ภาพที่ 3.2.6.4.1 แสดงฟังก์ชันสร้างชุดข้อมูลแบบกำหนด 33
ขนาดของชุดข้อมูล (batch)

สารบัญรูป (ต่อ)

หน้า

ภาพที่ 3.3.1.1 แสดงโครงสร้างของสถาปัตยกรรม Wave-U-Net 34
ภาพที่ 3.3.1.4.1 แสดงฟังก์ชันการสร้าง Wave-U-Net โมเดล 36
ภาพที่ 3.3.2.1 แสดงฟังก์ชั่นการ Compile Wave-U-Net โมเดล 37
ภาพที่ 3.3.3.1 แสดงฟังก์ชันสำหรับการ Callbacks 38
ภาพที่ 3.3.4.1 แสดงการฝึกฝนโมเดลด้วยคำสั่ง fit 38
ภาพที่ 3.4.1.1 แสดงการนำเข้าไลบารีที่ใช้ 39
ภาพที่ 3.4.2.1 แสดงการโหลดโมเดล 39
ภาพที่ 3.4.3.1 แสดงฟังก์ชั่นการสร้างโมเดล 39
ภาพที่ 3.4.3.2 แสดงฟังก์ชั่นการแยกเสียงโมเดล 40
ภาพที่ 3.4.4.1 แสดงฟังก์ชั่นการประเมินด้วยค่า SDR 41
ภาพที่ 3.4.5.1.1 แสดงแบบฟอร์มการประเมินการแยกเสียง 42
เครื่องดนตรีไทยเดิมส่วนความคุ้นเคย
ภาพที่ 3.4.5.2.1 แสดงแบบฟอร์มการประเมินการแยกเสียง 43
เครื่องดนตรีไทยเดิมส่วนความชัดเจนในการแยกเสียง
ภาพที่ 3.4.5.3.1 แสดงแบบฟอร์มการประเมินการแยกเสียงเครื่องดนตรีไทย 43
เดิมส่วนการแสดงความคิดเห็น

สารบัญกราฟ

หน้า

กราฟที่ 4.1.2.1 การเปรียบเทียบการใช้ CPU ของแต่ละโมเดล 45
กราฟที่ 4.1.2.2 การเปรียบเทียบการใช้ RAM ของแต่ละโมเดล 45
กราฟที่ 4.1.3.1 การเปรียบเทียบค่า learning rate ของแต่ละโมเดล 46
กราฟที่ 4.1.4.1 การเปรียบเทียบค่า Training loss ของแต่ละโมเดล 46
กราฟที่ 4.1.5.1 การเปรียบเทียบค่า MAE ของแต่ละโมเดล 47
กราฟที่ 4.1.6.1 การเปรียบเทียบ เวลา ของแต่ละโมเดล 47
กราฟที่ 4.1.7.1 การเปรียบเทียบค่า Validate Loss ของแต่ละโมเดล 48
กราฟที่ 4.1.7.2 การเปรียบเทียบค่า Validate MAE ของแต่ละโมเดล 48
กราฟที่ 4.2.1.1 กราฟแท่งแสดงความคมชัดของเสียงซออู้ 49
กราฟที่ 4.2.1.2 กราฟแท่งแสดงความคมชัดของเสียงขลุ่ย 49
กราฟที่ 4.2.1.3 กราฟแท่งแสดงความคมชัดของเสียงระนาดเอก 50

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

เพลงไทยเดิมเป็นศิลปะดนตรีที่สะท้อนวัฒนธรรมและวิถีชีวิตของคนไทยมานานนับศตวรรษมีบทบาทสำคัญในการสร้างสรรค์อัตลักษณ์และเสริมสร้างความรู้สึกภาคภูมิใจในวัฒนธรรมท้องถิ่น อย่างไรก็ตามการอนุรักษ์และสืบสานเพลงไทยเดิมกลับมีความท้าทายเนื่องจากการเปลี่ยนแปลงทางสังคมและเทคโนโลยีที่รวดเร็วทำให้เกิดความเสี่ยงที่วัฒนธรรมดนตรีนี้จะเลือนหายไปการอนุรักษ์เพลงไทยเดิมจึงเป็นภารกิจที่สำคัญเพื่อให้อนุชนรุ่นหลังได้เรียนรู้และสืบทอดต่อไป

ในยุคดิจิทัลการนำเทคโนโลยีเข้ามามีบทบาทในการอนุรักษ์และศึกษาเพลงไทยเดิมซึ่งเป็นแนวทางที่มีศักยภาพมาก เทคโนโลยีการแยกเสียง (Sound Demixing) ก็ได้มีประวัติศาสตร์งานวิจัยมาอย่างยาวนาน หนึ่งในแรงจูงใจหลักมาจากประโยชน์ที่หลากหลายที่เทคโนโลยีนี้มอบให้ ด้วยเทคโนโลยีการแยกสัญญาณผู้ผลิตและศิลปินดนตรีสามารถนำดนตรีเก่ามาใช้ประโยชน์ได้สตูดิโอภาพยนตร์สามารถนำภาพยนตร์คลาสสิกเก่าที่บันทึกมาในแบบเสียงโมโนมาฟื้นฟูใหม่และนำกลับมาฉายในโรงภาพยนตร์โดยใช้ประโยชน์จากนวัตกรรมใหม่ ๆ ในปัจจุบันเทคโนโลยีการแยกเสียงก็ยังคงพัฒนาอย่างต่อเนื่องดั่งเช่นThe Sound Demixing Challenge 2023 [5] ที่ได้รวบรวมผู้คนมาพัฒนาโมเดลเพื่อใช้ในการแยกเสียง ทำให้ Sound demixing เป็นเครื่องมือที่สามารถช่วยในการอนุรักษ์เพลงไทยเดิมได้อย่างมีประสิทธิภาพด้วยความสามารถในการแยกเสียงเครื่องดนตรีไทยออกจากเพลงไทยเดิม ทำให้สามารถวิเคราะห์องค์ประกอบดนตรีได้อย่างละเอียดลึกซึ้งและสามารถจัดเก็บข้อมูลเสียงในรูปแบบที่บริสุทธิ์และชัดเจน

เทคโนโลยี Sound Demixing ใช้เทคนิคการประมวลผลสัญญาณดิจิทัล (DSP) และปัญญาประดิษฐ์ (AI) ในการแยกเสียงจากส่วนผสมเสียงรวม วิธีการที่นิยมในการแยกเสียงประกอบด้วย Deep Convolutional Neural Networks (CNN) การวิเคราะห์องค์ประกอบอิสระ (ICA) , การแยกเมทริกซ์ที่ไม่เป็นลบ (NMF), และเครือข่ายประสาทเทียมเชิงลึก(DNNs)การนำเทคโนโลยีเหล่านี้มาประยุกต์ใช้กับเพลงไทยเดิมจะช่วยในการศึกษาวิจัย วิเคราะห์เชิงลึก และการฝึกสอน รวมถึงการสร้างฐานข้อมูลเสียงที่มีคุณภาพ

ดังนั้นการพัฒนาเทคโนโลยี AudioDemixing
สำหรับเพลงไทยเดิมจึงมีความสำคัญอย่างยิ่งทั้งในแง่ของการอนุรักษ์วัฒนธรรมการศึกษาวิจัยและการสร้างทรัพยากรดนตรีที่สามารถใช้ในการเรียนการสอนการแยกเสียงเพลงไทยเดิมออกมาอย่างชัดเจนและถูกต้องจะช่วยให้การศึกษาและอนุรักษ์เพลงไทยเดิมมีประสิทธิภาพยิ่งขึ้นและเป็นการสืบสานมรดกทางวัฒนธรรมอันล้ำค่านี้ให้คงอยู่ต่อไปในอนาคต

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อพัฒนาเทคนิคการแยกเสียงเครื่องดนตรีไทยออกจากเพลงที่มีการนำเครื่องดนตรีไทยมาใช้โดยใช้เทคโนโลยีการแยกเสียง (Audio Demixing) ที่มีประสิทธิภาพ

1.2.2 เพื่อสำรวจและแก้ไขปัญหาที่เกิดขึ้นในกระบวนการแยกเสียง เช่น เสียงรบกวน ความซับซ้อนของเสียงเครื่องดนตรีไทย และข้อมูลการฝึกที่จำกัด

1.2.3 เพื่อพัฒนาระบบและเครื่องมือที่สามารถช่วยในการอนุรักษ์และศึกษาวิเคราะห์เพลงไทยเดิมโดยสามารถใช้งานได้ง่ายและมีประสิทธิภาพสูง

1.3 ขอบเขตการวิจัย

1.3.1 การพัฒนาโมเดล AI จะใช้ภาษา Python ร่วมกับ Pytorch
ซึ่งเป็นซอฟต์แวร์ที่มีความสามารถในการสร้างและฝึกโมเดล Deep Learning

1.3.2 การเตรียมข้อมูล สำหรับการสร้างโมเดล AI โดยจะแบ่งเป็นชุดข้อมูลสำหรับฝึกโมเดลและทดสอบโมเดล โดยข้อมูลดังกล่าวจะประกอบไปด้วยเสียงเครื่องดนตรีไทยทั้งหมด 3 เสียงซึ่งได้มาจากเพลงเดี่ยว ได้แก่ ขลุ่ยเพียงออ ระนาดเอก และ ซออู้

1.3.3 การสร้างโมเดล สำหรับ Sound Demixingโดยใช้ CNN (Convolutional Neural Network) เป็น Algorithm ของ model นี้

1.3.4 การทดสอบและประเมินผล ใช้ SDR (Signal-to-Distortion Ratio) เป็นตัวชี้วัดที่ใช้ประเมินคุณภาพของสัญญาณที่ถูกแยกหรือปรับปรุงจากสัญญาณต้นฉบับ คำนวณจากอัตราส่วนระหว่างพลังงานของสัญญาณที่ต้องการ (target signal) กับพลังงานของสัญญาณที่ผิดพลาดหรือความผิดเพี้ยน (distortion/error signal)

1.3.5 การวิเคราะห์ผล การวิเคราะห์ผลการทดสอบและประเมินผลเพื่อนำมานำเสนอ

1.4 ประโยชน์ที่คาดว่าจะได้รับจากการวิจัย

1.4.1 การอนุรักษ์และสืบสานวัฒนธรรมไทย: เทคโนโลยี Sound Demixing ยังสามารถนำไปใช้ในการสร้างสรรค์ผลงานเพลงไทยใหม่ หรือการ Remastering เพลงไทยเดิม
ช่วยให้เพลงไทยเป็นที่รู้จักและได้รับความนิยมมากขึ้น

1.4.2 การศึกษาและวิจัย: เทคโนโลยี Sound Demixing ช่วยให้นักวิจัยสามารถศึกษาโครงสร้างเสียงของเครื่องดนตรีไทยและเสียงดนตรีประกอบช่วยให้เข้าใจกลไกการสร้างเสียงและทฤษฎีดนตรีไทยได้ดียิ่งขึ้นและยังสามารถนำไปใช้ในการวิเคราะห์เพลงไทยช่วยให้เข้าใจเนื้อหาอารมณ์และความหมายของเพลงได้ดียิ่งขึ้น

1.4.3 การพัฒนาเครื่องดนตรีไทย: ช่วยให้นักพัฒนาเครื่องดนตรีไทยสามารถออกแบบและสร้างเครื่องดนตรีใหม่ที่มีเสียงที่ไพเราะและสมจริงมากยิ่งขึ้น

1.4.4 อุตสาหกรรมดนตรี: ช่วยให้นักดนตรีไทยสามารถแยกเสียงเครื่องดนตรีของตัวเองออกจากเพลงช่วยให้นักดนตรีสามารถฝึกฝนและพัฒนาฝีมือการบรรเลงได้ดียิ่งขึ้น

1.5 อุปกรณ์และเครื่องมือที่ใช้ในการดำเนินงาน

1.5.1 Hardware

เครื่องคอมพิวเตอร์ 3 เครื่อง ที่มีคุณสมบัติ ดังนี้

1. Notebook (Lenovo)
   - AMD Ryzen 5 5600H with Radeon Graphics
   - RAM 32 GB DDR4 3200 MHz
   - 64 bit Operating System
2. Notebook (Acer)
   - AMD Ryzen 7 3750
   - RAM 8GB DDR4 3200 MHz
   - 64 bit Operating System
3. PC
   - AMD Ryzen 5 5600G with Radeon Graphics
   - RAM 32 GB DDR4 3200 MHz
   - 64 bit Operating System

1.5.2 Software

1. Colab ใช้ในการ train model

   TPU hight ram:
   - RAM 334.6 GB
   - Disk 225.3 GB
2. google drive เก็บ dataset และ model
3. จัดทำเอกสารโดยใช้ซอฟต์แวร์ Microsoft 365: Word

1.6 นิยามคำศัพท์เฉพาะ

1.6.1 Audio Demixing คือเทคนิคการแยกเสียงต้นฉบับออกจากส่วนผสมของเสียงหลายเสียง เปรียบเสมือนการแยกชิ้นส่วนของวงดนตรีออกจากเพลงที่บันทึกไว้

1.6.2 Digital Signal Processing (DSP) คือการวิเคราะห์และประมวลผลสัญญาณต่างๆ โดยใช้เทคนิคทางคณิตศาสตร์บนคอมพิวเตอร์หรืออุปกรณ์ดิจิทัลอื่นๆ แทนการใช้ระบบแอนะล็อกแบบดั้งเดิม

1.6.3 Independent Component Analysis (ICA)
คือเทคนิคทางคณิตศาสตร์ที่ใช้แยกสัญญาณผสมออกเป็นส่วนประกอบย่อยที่เป็นอิสระ

1.6.4 Non-negative Matrix Factorization (NMF)
คือเทคนิคการแยกตัวประกอบเมทริกซ์ที่ใช้สำหรับวิเคราะห์ข้อมูลที่ไม่เป็นลบ เช่น ข้อมูลภาพ เสียง และข้อความ โดยแยกเมทริกซ์ที่มีค่าเป็นจำนวนบวกออกเป็นสองเมทริกซ์ย่อย
1. เมทริกซ์ตัวประกอบ (W) : เมทริกซ์นี้มีขนาด (m x n) โดยที่ m คือจำนวนแถวในเมทริกซ์ดั้งเดิม และ n คือจำนวนหลักฐาน
2. เมทริกซ์การเปิดใช้งาน (H) : เมทริกซ์นี้มีขนาด (n x k) โดยที่ k คือจำนวนสาเหตุ

1.6.5 Deep Neural Networks (DNNs)คือส่วนหนึ่งของการเรียนรู้ของเครื่อง (Machine Learning) และการประมวลผลข้อมูลที่ถูกออกแบบมาโดยจำลองโครงสร้างของระบบประสาทเทียมในสมองมนุษย์ โดยมีการประยุกต์ใช้หลายชั้น (layers) ของโหนด (nodes) ในแต่ละชั้น เพื่อทำงานกับข้อมูลขนาดใหญ่และซับซ้อนได้อย่างมีประสิทธิภาพ

1.6.6 Remastering คือกระบวนการนำเอาเพลงที่บันทึกไว้แล้ว มาทำการปรับแต่งเสียงใหม่ให้ดียิ่งขึ้น โดยใช้เทคโนโลยีและเครื่องมือที่ทันสมัย

บทที่ 2
วรรณกรรมที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ดนตรีไทย

บทวิยานิพนธ์นายโอภาส แก้วต่าย [3] ได้หยิบยกคำอธิบายเกี่ยวกับดนตรีไทยที่เฉลิมศักดิ์ พิกุลศรี (2530) ได้กล่าวถึง ประเภทของเครื่องดนตรีไทยและวงดนตรีไทยไว้ดังต่อไปนี้

1. ประเภทของเครื่องดนตรีไทย

ในคัมภีร์ที่ชื่อว่า นาฏยศาสตร์ (Natayasastra) ได้จำแนกเครื่องดนตรีออกเป็น 4 ประเภท คือ

- ตะตะ คือ เครื่องดนตรีประเภทมีสายสำหรับดีดหรือสีเป็นเสียง
- สุษิระ คือ เครื่องดนตรีประเถทที่เป่าเป็นเสียง
- อะวะนัทธะ คือ เครื่องดนตรีหุ้มหนังสีเป็นเสียง
- ฆะนะ คือ เครื่องดนตรีที่กระทบเป็นเสียง

ดนตรีทางตะวันตกนั้นการจำแนกเครื่องดนตรี

จำแนกตามลักษณะของเครื่องดนตรีแบ่งออกเป็น 4 ประเภทดังนี้

- เครื่องสาย (String Instruments)
- เครื่องลมไม้ (Wood wind Instruments)
- เครื่องทองเหลือง (Brass Instruments)
- เครื่องเพอร์คัชชัน (Percussion Instruments)

สำหรับเครื่องดนตรีไทยนั้น การจำแนกเราอาศัยจำแนกกิริยาอาการปฏิบัติของผู้เล่น ซึ่งสามารถแบ่ง ออกได้เป็น 4 ประเภทดังนี้

- เครื่องดนตรีประเภทดีด (Plucked String Instruments)
- เครื่องดนตรีประเภทสี (Bowed String Instruments)
- เครื่องดนตรีประเภทตี (Percussion Instruments)
- เครื่องดนตรีประเภทเป่า (Wind Instruments)

2. วงดนตรีไทย วงดนตรีไทยในปัจจุบันแบ่งออกเป็น 3 ประเภท คือ

1. วงเครื่องสาย หมายถึง วงดนตรีที่ประกอบด้วยดนตรีประเภทเครื่องสาย เป็นหลัก และมีเครื่องดนตรีประเภทเครื่องเป่าเป็นส่วนประกอบ โดยมีฉิ่ง ฉาบ กรับ เป็นเครื่องประกอบจังหวะ เครื่องดนตรีประเภททำนองประกอบด้วย จะเข้ ซออู้ ซอด้วง โทน-รำมะนา และ ฉิ่ง ฉาบ กรับ โหม่ง ความมาน้อยเครื่องดนตรีขึ้นอยู่กับขนาดของวง

2. วงปี่พาทย์ หมายถึง วงดนตรีที่เกิดจากการประสมกันระหว่างเครื่องดนตรีประเภทเป่า และเครื่องดนตรีประเภทตี

3. วงมโหรี เป็นวงดนตรีที่เกิดจากการประสมกันระหว่างวงปี่พาทย์และวง เครื่องสาย โดยติดเครื่องดนตรีที่มีเสียงดังออก เช่น ปี่ออก วงมโหรีเป็นวงดนตรีที่มีความสมบูรณ์ ที่สุด กล่าวคือ มีเครื่องดนตรีที่ประสมอยู่ในวงครบทุกตระกูล คือ ดีด สี ตีและเป่า

	ดีด	สี	ตี	เป่า
เครื่องสาย	จะเข้	ซอด้วง ซออู้	-	ขลุ่ย
ปี่พาทย์	-	ไม้นวม ซออู้	ระนาดเอก ระนาดทุ้ม ระนาดเอกเหล็ก ฆ้องวงใหญ่และเล็ก	ไม้แข็ง ปี่ใน นอก กลาง ขลุ่ยหลีบ, เพียงออ, อู้
มโหรี	จะเข้	ซอสามสาย ซออู้ ซอด้วง	ระนาดเอก-ทุ้ม ระนาดเอกเหล็ก-ทุ้ม ฆ้องวงใหญ่-เล็ก	ขลุ่ยอู้ ขลุ่ยเพียงออ ขลุ่ยหลีบ

ภาพที่ 2.1.1.3.1 แสดงการจำแนกตระกูลเครื่องดนตรีตามประเภทของวงดนตรี

ที่มา: โอภาส แก้วต่าย, การจำแนกกลุ่มเพลงไทยเดิม, มหาวิทยาลัยศิลปากร, 2552, หน้า 6 [2]

2.1.2 เสียง (Sound) (อานนท์ นามสนิท 2549: 4-5 [2])

เสียงเกิดจากการสั่นของโมเลกุลของวัตถุ แล้วถ่ายเทพลังงานให้กับโมเลกุลของ อากาศที่วิ่งมาชน ทำให้โมเลกุลของอากาศมีความเร็วสูงขึ้นแล้วเกิดการชนกันของโมเลกุล เมื่อเกิดการชนกันแล้วถ่ายเทพลังงานออกมาเป็นทอดๆ ก็เหมือนคลื่นของความดันที่แผ่ออกไป เมื่อมากระทบแผ่นไดอะแฟรมที่หูของมนุษย์ก็จะเปลี่ยนให้เป็นสัญญาณไฟฟ้าส่งไปยังสมอง ทำให้เราสามารถรับรู้และแยกแยะเสียงต่างๆ ได้

1. ระดับเสียง (Pitch) ระดับเสียงเกิดจากการสั่นสะเทือนของวัตถุวัตถุที่สั่นสะเทือนเร็วจะทำให้เกิดระดับเสียงที่สูงกว่า ในขณะเดียวกันวัตถุที่สั่นสะเทือนช้าก็จะทำให้เกิด ระดับเสียงที่ต่ำกว่า โดยระดับเสียงจะมีหน่วยเป็นรอบต่อนาทีวัตถุที่สั่นสะเทือนมากกว่าจะมีความถี่มากกว่าทำให้เกิดระดับเสียงที่สูงกว่าถ้าความถี่มากขึ้นเท่าตัว ระดับเสียงจะสูงขึ้นหนึ่งช่วงคู่ แปด (Octave) เช่น เสียงที่มีความถี่ 220 รอบต่อนาที จะมีความถี่เป็นช่วงคู่แปดกับเสียงที่มีความถี่ 110 รอบต่อวินาที

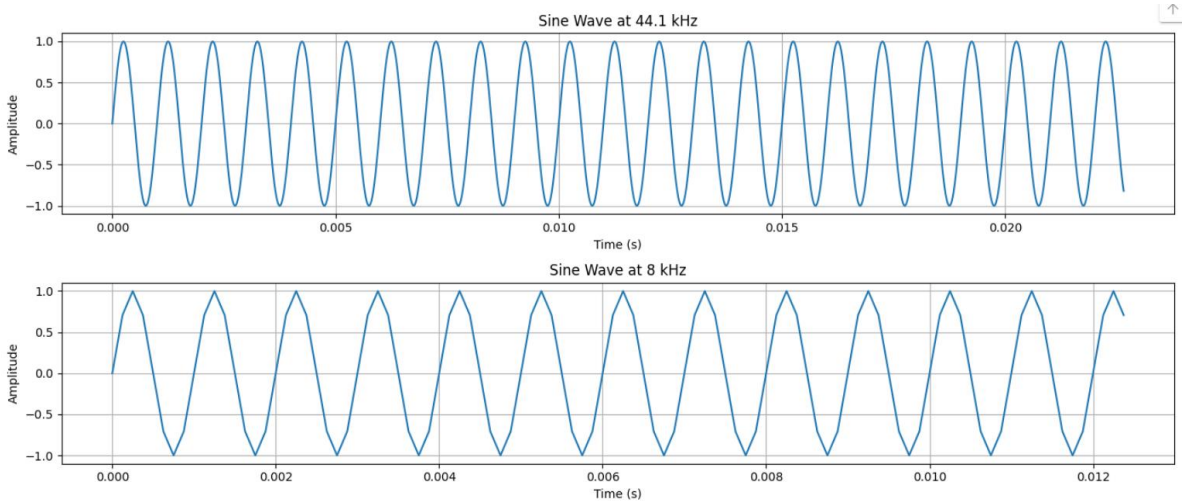
2. ความเข้มของเสียง (Intensity) หมายถึงเสียงเบาเสียงดังเกิดจากแรงสั่นสะเทือนของวัตถุที่เป็นแหล่งกำเนิดเสียงสะเทือนมากจึงเกิดเสียงดังมากในทางตรงกันข้ามหากวัตถุต้นกำเนิดเสียงสั่นสะเทือนน้อยก็จะเกิดเสียงเบาดังนั้นความเข้มของเสียงจึงขึ้นอยู่กับความแรงที่ส่งจากแหล่งกำเนิดเสียงไปยังหูโดยสามารถวัดได้จากความสูงของคลื่นเสียง (Amplitude) ปัจจัยที่มีผลต่อความเข้มเสียงก็คือระยะทางเพราะเมื่อเสียงเดินทางผ่านบรรยากาศความเข้าเสียงจะน้อยลงตามลำดับ

3. สีสันเสียง (Timbre หรือ Tone color) คือลักษณะที่แตกต่างกันของเสียงแต่ละ โดยคลื่นเสียงที่ออกมาจากแหล่งกำเนิดเสียงที่ต่างชนิดกันก็จะมีลักษณะรูปร่างที่แตกต่างกัน
เช่น เมื่อเราเล่นโน้ตเดียวกันโดยใช้เครื่องดนตรีที่ต่างกันสองชนิด รูปคลื่นที่ได้จากเครื่องดนตรีทั้งสองก็จะแตกต่างกันทำให้เราสามารถแยกแยะได้ว่างเสียงที่ได้ยินเป็นเสียงของเครื่องดนตรีชนิดใดบ้าง

4. คุณภาพเสียง (Tone quality) หมายถึงเสียงที่ได้ออกมาจากแหล่งกำเนิดเสียง มีคุณภาพที่ดีมากหรือดีน้อยเช่นเมื่อเราเล่นโน้ตเดียวกันจากเปียโนคนละตัวคุณภาพเสียงที่ได้ก็จะต่างกัน

5. ความยาวเสียง (Duration) เป็นพื้นฐานของดนตรีซึ่งต้องเกี่ยวข้องกับเวลา เสียงแต่ละเสียงที่เกิดขึ้นต้องมีระยะเวลาซึ่งทำให้เกิดเสียงสั้นเสียงยาวไม่ว่าเสียงจะมีระดับเสียงที่ แน่นอนหรือไม่ก็จะต้องมีระยะเวลาเข้ามาเกี่ยวข้องเสมอ ความยาวของเสียงจึงเป็นที่มาของจังหวะในดนตรี ซึ่งรวมไปถึงความยาวของความเงียบด้วยเนื่องจากดนตรีเป็นผลของกระบวนการเกิด เสียงสลับกับความเงียบซึ่งต่างก็ต้องมีระยะเวลาทั้งนั้น

6. การสุ่มหน้าคลื่น (Sampling) เนื่องจากเสียงนั้นเป็นสัญญาณต่อเนื่อง เมื่อต้องการแปลงให้อยู่ในรูปของสัญญาณดิจิตอลก็จะต้องทำการสุ่มหน้าคลื่นด้วยการดูว่าขณะนี้มี Amplitude เท่าไหร่ โดยอัตราในการสุ่มหน้าคลื่นนี้เรียกว่า Sampling Rate เช่น ถ้าสุ่ม 8000 ครั้งต่อวินาที เรียกว่า 8 kHz sampling หมายความว่าในเวลา 1 วินาที จะได้ข้อมูล amplitude จำนวน 8000 จุด



ภาพที่ 2.1.2.6.1 แสดงความแตกต่างระหว่างการสุ่มสัญญาณ (sampling rate) ที่ 8 kHz และ 44.1 kHz

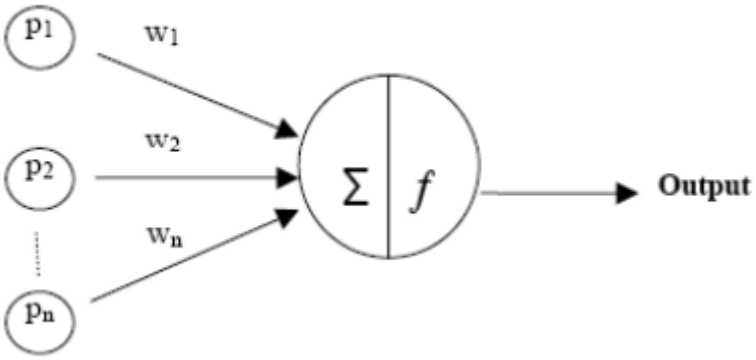
7. ความสูงของคลื่น (amplitude) Sampling rate เป็นตัวกำหนดความมากน้อยของจุดที่ใช้แทนหน้าคลื่น โดยจุดที่บันทึก นั่นก็คือ ความสูงของคลื่น ซึ่งจะใช้ตัวเลขแค่ 0 กับ 1 แทน
amplitude ของคลื่น เรียกการบันทึกแบบนี้ว่ามี quantization แบบ 2 bit ถ้าเพิ่มจำนวน bit มา เป็น 4 bit จะสามารถกำหนดระดับ amplitude ได้เพิ่มขึ้นเป็น 16 ระดับ

2.1.3 เครือข่ายใยประสาทเทียม (Artificial neural network-ANN) (จารวี ฉันทสิทธิพร 2548: 16 [1])

เครือข่ายใยประสาทเทียม (Artificial neural network)

เป็นการจำลองการทำงานบางส่วนของสมองมนุษย์ ซึ่งประกอบด้วยเซลล์ประสาท (neuron) เป็นจำนวนมาก โดยในแต่ละเซลล์จะประกอบด้วยนิวเคลียส (nucleus) ตัวเซลล์ (cell body) ใยประสาทนำเข้า (dendrite) แกนประสาทนำออก (axon)

โดยใยประสาทนำเข้าจะมีหน้าที่รับสัญญาณไฟฟ้าเคมีซึ่งส่งมาจากเซลล์ประสาทที่อยู่ใกล้เคียง เมื่อสัญญาณไฟฟ้าเคมีที่ได้รับเข้ามาเกินค่าค่าหนึ่งเซลล์ประสาทจะถูกกระตุ้นและส่งสัญญาณไปทางแกนประสาทนำออกต่อไป เมื่อปรับโครงสร้างและรูปแบบมาใช้กับเครือข่ายใยประสาทเทียมจะใช้โหนด (nodes) ทำหน้าที่คล้ายกับตัวเซลล์ประสาท (neurons) ดังภาพที่ 3 เซลล์ประสาทแต่ละโหนดจะรับค่าอินพุตได้หลายค่า (P1, P2, ---. Ps) แต่ผลการกระตุ้นหรือค่าเอาต์พุตที่ได้มีเพียงหนึ่งค่า ซึ่งคำนวณได้จากการใช้ transfer function (f) กับผลรวมเชิงเส้นแบบถ่วงน้ำหนัก (w, W2, ... , w.) ของอินพุต



ภาพที่ 2.1.3.1 รูปแบบของเซลล์ประสาท (Neuron model)

ที่มา : จารวี ฉันทสิทธิพร, "การจำแนกชนิดยาเม็ดจากภาพถ่าย โดยใช้เทคนิคเครือข่ายใยประสาท" (วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร, 2548), 16.

2.2 เอกสารงานวิจัยที่เกี่ยวข้อง

งานวิจัยเกี่ยวข้องกับการแยกเสียง (Sound Demixing) นั้นจะแบ่งเนื้อหาหรือวิธีทำการเป็นสองส่วน คือส่วนของการเลือกและการสกัดคุณลักษณะ และ ก็ส่วนของการแยกเสียง งานวิจัยนี้จึงได้ทำการรวบรวมและศึกษางานวิจัยต่าง ๆ ที่เกี่ยวข้อง เพื่อทำการศึกษาวิธีการ รวมถึงเทคนิคในการแยกเสียงดังต่อไปนี้

1. J. Smith A. Johnson R. Doe และ P. Brown [4]
ได้เสนอวิธีการแยกเสียงดนตรีจากสัญญาณเสียงโมโน โดยใช้เครือข่ายประสาทเทียมแบบ convolutional (CNN) โดยใช้ DSD100 เป็นชุดข้อมูลเสียงเพลงที่ประกอบด้วยเพลงความยาวเต็มรูปแบบ 100 เพลง แยกตามประเภทดนตรี เพลงแต่ละเพลงถูกแยกออกเป็นสี่แทร็ก: เสียงกลอง เสียงเบส เสียงร้อง และแทร็กอื่นๆ
โดยสร้างจำนวนพารามิเตอร์ของเครือข่ายเป็นสัดส่วนโดยตรงกับเวลาในการประมวลผลที่เครือข่ายต้องกา ร เนื่องจากเป้าหมายของงานวิจัยนี้คือการออกแบบเวลาแฝงต่ำ (low-latency)
ในอัลกอริธึมการแยกแหล่งที่มา (source separation algorithm)
ของงานวิจัยนี้ได้พยายามลดพารามิเตอร์ของเครือข่ายให้เหลือน้อยที่สุดโดยการปรับตัวแปร เวลา รูปร่างตัวกรอง (t1, f1) และ(t2, f2), จำนวนตัวกรอง, N1 และ N2 และจำนวนโหนดในคอขวด, NN โดยที่ไม่กระทบต่อประสิทธิภาพการทำงาน ตัวแปรเหล่านี้ถูกกำหนดให้เป็น 25 (290 ms), (1, 513), (12, 1), 50, 30 และ 128 ตามลำดับ

โดยผลลัพท์ของงานวิจัยนี้แสดงให้เห็นว่าเวลาในการประมวลผลที่ MLP กำหนดบนระบบคอมพิวเตอร์ที่ใช้นั้นสูงกว่าเวลาประมวลผลที่จำเป็นสำหรับ CNN ถึง 4 เท่า สำหรับการป้อนข้อมูลเวลาจาก 290 มิลลิวินาที CNN ใช้เวลาเพียง 161 มิลลิวินาทีในการประมวลผล ในขณะที่ MLP ใช้เวลาเฉลี่ย 654 มิลลิวินาที ทำให้ CNN ใช้เวลาน้อยกว่าวิธีการแบบเดิม (MLP)
และสามารถเรียนรู้ข้อมูลที่ซับซ่อนได้จากการลดมิติของข้อมูล

2. H. Kim J. Park และ S. Lee [1] ได้เสนอวิธีการแยกเสียงร้องโดยใช้ U-Net ร่วมกับ spectrogram แบบ complex-valued โดยใช้ MUSDB ชุดข้อมูลเสียงดนตรีที่ออกแบบมาสำหรับงานแยกแหล่งดนตรี (Music Source Separation - MSS) โดยมีจำนวนเพลง 1500 เพลง หลากหลายประเภท แบ่งออกเป็น เสียงร้อง เสียงเครื่องดนตรี และ เสียงรบกวน

model	# parameters	SDR (vocals)
DGRU-DGConv	more than 1.9M	6.99
TAK1	1.22M	6.60
UMX	8.89M	6.32
TFC-TDF (small)	0.99M	**7.07** ±.08
TFC-TDF (large)	2.24M	**7.98** ±.07

ภาพที่ 2.2.2.1 ตารางแสดงผลการเปรียบเทียบ model ด้วย SDR metric

ที่มา H. Kim J. Park และ S. Lee, Investigating U-Nets with various Intermediate Blocks for Spectrogram-based Singing Voice Separation, Cornell University, 2020, หน้า 6

ผลการวิจัยพบว่า บล็อกที่ประกอบด้วยชั้น Convolutional และ Fully-connected (TFC-TDF)สามารถแยกเสียงร้องได้มีประสิทธิภาพดีที่สุดตามค่า SDR metric

3. Daniel Stoller, Sebastian Ewert และ Simon Dixon [9] ได้นำเสนอ Wave-U-Net โมเดลเครือข่ายประสาทเทียมแบบ multi-scale สำหรับงานแยกแหล่งเสียงแบบ end-to-end โดยใช้ MUSDB เช่นเดียวกับงานวิจัยที่ใช้แยกเสียงร้องด้วย U-Net
แต่ที่แตกต่างกันคืองานวิจัยนี้สามารถทำงานบน raw waveform โดยไม่ต้องแปลงเป็น spectrogram ได้

		M1	M2	M3	M4	M5	M7	U7	U7a
Voc.	Med.	3.90	3.92	3.96	4.46	**4.58**	**3.49**	2.76	2.74
	MAD	3.04	3.01	3.00	3.21	3.28	2.71	2.46	2.54
	Mean	-0.12	0.05	0.31	**0.65**	0.55	-0.23	-0.66	**0.51**
	SD	14.00	13.63	13.25	13.67	13.84	13.00	12.38	10.82
Acc.	Med.	7.45	7.46	7.53	**10.69**	10.66	**7.12**	6.76	6.68
	MAD	2.08	2.10	2.11	3.15	3.10	2.04	2.00	2.04
	Mean	7.62	7.68	7.66	**11.85**	11.74	**7.15**	6.90	6.85
	SD	3.93	3.84	3.90	7.03	7.05	4.10	3.67	3.60

Table 2. Test set performance metrics (SDR statistics, in dB) for each singing voice separation model. Best performances overall and among comparison models are shown in bold.

	Vocals				Other			
	Med.	MAD	Mean	SD	Med.	MAD	Mean	SD
M6	3.0	2.76	-2.10	15.41	2.03	1.64	1.68	6.14
	Bass				Drums			
	Med.	MAD	Mean	SD	Med.	MAD	Mean	SD
M6	2.91	2.47	-0.30	13.50	4.15	1.99	2.88	7.68

Table 3. Test performance metrics (SDR statistics, in dB) for our multi-instrument model

ภาพที่ 2.2.3.1 ตารางแสดงผลการเปรียบเทียบ model ด้วย SDR metric ทั้งแบบเสียงร้องและเสียงเครื่องดนตรี
ที่มา Daniel Stoller, Sebastian Ewert และ Simon Dixon, Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation, Cornell University, 2018, หน้า 6

ผลการวิจัยพบว่าตัวโมเดล M6 ซึ่งเกิดจากการปรับขนาดโมเดลให้รองรับสัญญาณเสียงที่มี Cมิติที่สอง (C) ถูกพิจารณาเป็นช่องฟีเจอร์ช่องสัญญาณ ข้อมูลอินพุต (M) จะถูกปรับเปลี่ยนจากเมทริกซ์ขนาด Lm x 1 เป็น Lm x C โดยใช้ K ชั้น Convolution แต่ละชั้นมีขนาดฟิลเตอร์ 1 และมี C ฟิลเตอร์ย่อย นั้นสามารถแยกเสียงเครื่องดนตรีได้ดีกว่าการแยกเสียงร้องเมื่อเทียบกับ model ตัวอื่นๆ ในขณะที่การแยกเสียงเดี่ยวที่เป็นเสียงร้องโมเดล M5 จะทำงานได้ดีกว่า

4. Yi Luo และ Jianwei Yu [8] ได้นำเสนอโมเดล Band-split RNN (BSRNN) สำหรับงานแยกแหล่งเสียงดนตรี (Music Source Separation - MSS) โมเดลนี้ทำงานในโดเมนความถี่ โดยแบ่ง spectrogram ของสัญญาณเสียงผสมออกเป็นย่านความถี่ (subbands) แต่ละย่านความถี่จะถูกประมวลผลด้วย RNN แยกต่างหาก โดยงานวิจัยนี้ได้ใช้ MUSDB และ private dataset จำนวน 1750 เพลง ในการสร้างโมเดล

Model	Vocals				Bass				Drum				Other				All			
	uSDR		cSDR		uSDR		cSDR		uSDR		cSDR		uSDR		cSDR		uSDR		cSDR	
	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ	HQ	nHQ
ResUNetDecouple+ [25]	–	–	–	8.98	–	–	–	6.04	–	–	–	6.62	–	–	–	5.29	–	–	–	6.73
CWS-PResUNet [26]	–	–	8.92	–	–	–	5.93	–	–	–	6.38	–	–	–	5.84	–	–	–	6.77	–
KUIELab-MDX-Net [32]	–	–	8.97	9.00	–	–	7.83	7.86	–	–	7.20	7.33	–	–	5.90	5.95	–	–	7.47	7.54
Hybrid Demucs [31]	–	–	8.13	8.04	–	–	**8.76**	**8.67**	–	–	8.24	8.58	–	–	5.59	5.59	–	–	7.68	7.72
BSRNN	10.04	9.92	10.01	10.21	6.80	6.77	7.22	7.51	8.92	8.68	9.01	8.58	6.01	5.97	6.70	6.62	7.94	7.84	8.24	8.23
+ finetuning	**10.47**	**10.36**	**10.47**	**10.53**	7.20	7.17	8.16	8.30	**9.66**	**9.46**	**10.15**	**9.65**	**6.33**	**6.27**	**7.08**	**7.00**	**8.42**	**8.32**	**8.97**	**8.87**

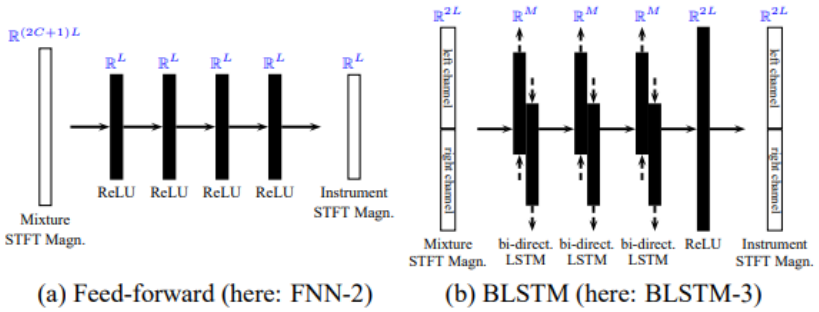
ภาพที่ 2.2.4.1 ตารางแสดงผลการเปรียบเทียบ model BSRNN กับ model ใน state of the art ด้วย SDR metric

ที่มา Yi Luo และ Jianwei Yu, Music Source Separation with Band-split RNN, Cornell University, 2020, หน้า 7

ผลการวิจัยแสดงให้เห็นว่าโมเดล BSRNN แสดงประสิทธิภาพการแยกแหล่งเสียงที่ดีกว่าโมเดล MSS ทั่วไป เช่น Deep Res-UNet และ Conv-TasNet บนชุดข้อมูล MUSDB18 และ MUSDB18-HQ แต่การแยกเสียงเบสยังด้อยกว่าโมเดลอื่นๆ เล็กน้อย

5. Stefan Uhlich , Marcello Porcu , Franck Giron , Michael Enenkl , Thomas Kemp , Naoya Takahashi และ Yuki Mitsufuji [10]

ได้นำเสนอวิธีการแยกเสียงดนตรีออกเป็นแทร็กของเครื่องดนตรีแต่ละชิ้นโดยใช้เครือข่ายประสาทเทียมแบบผสม โดยวิธีนี้มีประสิทธิภาพที่ดีกว่าวิธีการแยกเสียงแบบดั้งเดิม (การวิเคราะห์ความถี่ (Spectral Analysis), การวิเคราะห์เวลา (Temporal Analysis) และโมเดลทางสถิติ) การวิจัยนี้ใช้วิธีการเพิ่มข้อมูล (data augmentations) เพื่อปรับปรุงการเรียนรู้ของเครือข่ายประสาทเทียม



ภาพที่ 2.2.5.1 อธิบายโครงสร้างโมเดล Feed-forward และ โมเดล BLSTM

ที่มา Stefan Uhlich , Marcello Porcu , Franck Giron , Michael Enenkl , Thomas Kemp , Naoya Takahashi and Yuki Mitsufuji, Improving music source separation based on deep neural networks through data augmentation and network blending, 2017 , หน้า 3

	Approach	SDR in dB Bass	Drums	Other	Vocals	Acco.	Comments
Single-channel methods	BLEND (SWF)	2.76	3.93	3.37	5.13	11.53	$\lambda = 0.25$
	sNMF [24,25]	-0.84	1.12	1.82	2.17	8.58	$Q = 25$
	dNMF [26]	0.91	1.87	2.43	2.56	8.88	$Q = 25$
	DeepNMF [27]	1.88	2.11	2.64	2.75	8.90	$Q = 25$
Multi-channel methods	BLEND (MWF)	2.98	4.13	3.52	5.23	11.70	$\lambda = 0.25$
	NUG [9]	2.72	3.89	3.18	4.55	10.29	

ภาพที่ 2.2.5.2 เปรียบเทียบประสิทธิภาพโมเดลแยกเสียงดนตรีโดยใช้ชุดข้อมูล DSD100

ที่มา Stefan Uhlich , Marcello Porcu , Franck Giron , Michael Enenkl , Thomas Kemp , Naoya Takahashi and Yuki Mitsufuji, Improving music source separation based on deep neural networks through data augmentation and network blending, 2017 , หน้า 5

ผลการวิจัยนี้แสดงให้เห็นถึงผลลัพธ์ที่ดีที่สุดซึ่งได้มาจากการผสมผสานเชิงเส้น (linear blending) ของเอาต์พุตจากเครือข่ายประสาทเทียมแบบฟีดฟอร์เวิร์ด (feed-forward) และเครือข่ายประสาทเทียม LSTM แบบวนรอบทิศทางสองทาง (recurrent bi-directional)

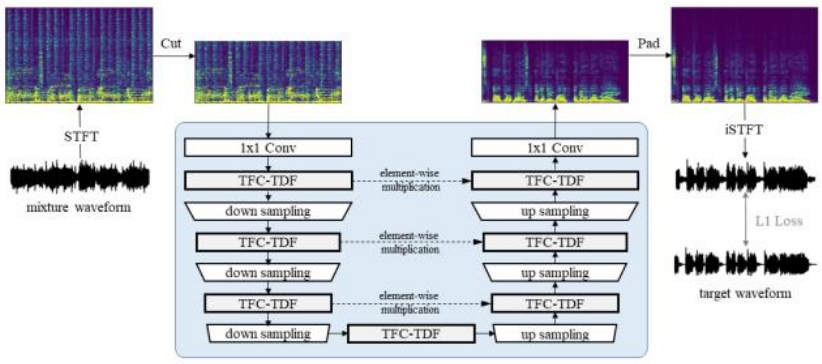
6. Kim, M., Lee, J., & Lee, H. [7]

ได้นำเสนอวิธีการแยกดนตรีออกจากสัญญาณเสียงผสมโดยใช้เครือข่ายประสาทเทียมโดยมีขั้นตอนสำคัญ 2 ขั้นตอน ได้แก่ คือ

1. วิเคราะห์สัญญาณเสียงผสม

จะทำการวิเคราะห์สัญญาณเสียงผสมเพื่อแยกคุณสมบัติที่เกี่ยวข้องกับแหล่งกำเนิดเสียงแต่ละแหล่ง เปรียบเสมือนการแยกส่วนประกอบต่างๆ ของเสียงผสม เช่น เสียงร้อง (ความถี่ รูปแบบคลื่น และความดัง) และเสียงเครื่องดนตรี (เสียงกลอง เสียงกีตาร์ และเสียงเบส)

2. การสร้างสัญญาณแยก จะใช้คุณสมบัติที่แยกออกมาจากขั้นตอนที่ 1 เพื่อสร้างสัญญาณแยกสำหรับแต่ละแหล่งกำเนิดเสียง เปรียบเสมือนการประกอบชิ้นส่วนที่แยกไว้กลับมาเป็นเสียงต้นฉบับ



ภาพที่ 2.2.6.1 โครงสร้างของโมเดล TFC-TDF-U-Net v2

ที่มา Kim, M., Lee, J., & Lee, H. (2021). KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing. In Proceedings of the 25th International Conference on Digital Signal Processing., หน้า 4

	vocals	drums	bass	other
Hybrid Demucs (defossez)	8.04	**8.58**	**8.67**	5.59
KUIELab-MDX-Net (kuielab)	**9.00**	7.33	7.86	**5.95**
Danna-Sep (KazaneRyonoDanna)	7.63	7.20	7.05	5.20

ภาพที่ 2.2.6.2 การเปรียบเทียบประสิทธิภาพของโมเดล

ที่มา Kim, M., Lee, J., & Lee, H. (2021). KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing. In Proceedings of the 25th International Conference on Digital Signal Processing., หน้า 6

ผลการวิจัยนี้แสดงให้เห็นว่าโมเดลที่ใช้สามารถเสียงดนตรีออกจากสัญญานเสียงผสมได้อย่างแม่นยำ และจากการเปรียบเทียบโมเดลทั้ง 3 ทำให้แสดงถึงการแยกเสียงร้องออกจากสัณญานเสียงได้ อย่างมีประสิทธิภาพมากกว่าโมเดลอีกทั้ง 2 โมเดล

7. Alex Favaro Aaron Lewis และ Garrett Schlesinger [6] ได้นำเสนอวิธีการแยกแหล่งเสียงดนตรี (Music Source Separation - MSS) โดยใช้อัลกอริทึม Independent Component Analysis (ICA) ซึ่งได้นำเสนอ 3 วิธีในการใช้งานซึ่งให้ผลลัพทธ์ที่แตกต่างกันออกไป
โดยใช้ชุดข้อมูลเป็นการบันทึกเสียงเครื่องดนตรี 4 ชิ้น ได้แก่ กีตาร์ไฟฟ้า เปียโน แซ็กโซโฟน เทรนเนอร์ และกลองสแนร์

วิธีการที่ 1 คือ FastICA สำหรับข้อมูลเชิงซ้อน
(CFastICA)วิธีนี้แปลงสัญญาณต้นฉบับด้วยฟังก์ชันฟูริเยร์ (Fourier Transform)
และนำไปใช้กับอัลกอริทึม CFastICA ผลลัพทธ์ที่ได้นั้นประสบความสำเร็จเพียงเล็กน้อย
แยกเสียงกลองออกได้ แต่เครื่องดนตรีอื่นๆ ยังผสมกันอยู่
สาเหตุมาจากการที่สัญญาณต้นฉบับมีความสัมพันธ์กันในโดเมนความถี่ CFastICA
ไม่สามารถแยกสัญญาณที่สัมพันธ์กันได้

วิธีการที่ 2 คือ FastICA สำหรับขนาด (magnitude)
ของสเปกตรัมวิธีนี้แปลงสัญญาณต้นฉบับด้วยฟังก์ชันฟูริเยร์ และนำขนาด (magnitude)
ของผลลัพธ์ไปใช้กับอัลกอริทึม FastICA โดยผลลัพทธ์สามารถแยกเสียงเปียโน (piano) และกลองสแนร์ (snare drum) ได้ค่อนข้างดี สาเหตุเป็นเพราะ ความถี่ของกลองสแนร์มีความเป็นอิสระมากที่สุด

วิธีการที่ 3 ICA with Linear Regression
ในอัลกอริทึม FastICA
ใช้การแก้ปัญหาแบบแยกสัญญาณทีละตัวซึ่งในขั้นตอนการคำนวณสัญญาณต้นฉบับทางผู้วิจัยได้เลือใช้วิธีที่ 3
ในการกำจัดความสัมพันธ์กับความถี่ออกจากโมเดลโดยการคำนวณค่าที่แต่เดิมเป็นค่าเฉลี่ยของการคำนวณสัญญาณต้นฉบับเองด้วยการใช้ Linear Regression
โดยผลลัพทธ์ของวิธีนี้ยังมีปัญหาในเรื่องของการเพิ่มองศาอิสระ (degrees of freedom)

ใหม่ทำให้การไล่ระดับความชัน (gradient descent) ของ FastICA
ไม่สามารถบรรจบได้ภายในระยะเวลาที่เหมาะสม

โดยสรุปแล้วงานวิจัยนี้แสดงให้เห็นว่า ICA
สามารถแยกแหล่งเสียงดนตรีได้แม้จะมีข้อจำกัดในด้านชุดข้อมูลแต่โมเดลนี้ไม่มีประสิทธิภาพมากนักเมื่อน
ำไปใช้กับเสียงที่มีความถี่ใกล้เคียงกัน

ชื่อผู้วิจัย	โมเดล	ชุดข้อมูล	SDR(Bass, Drums, Vocals, Others)	ทรัพยาการ	ข้อดีข้อเสีย
J. Smith A. Johnson R. Doe และ P. Brown	CNN-(CONV)	DSD100	(0.9 +- 2.7, 2.4+-2.0, 1.3+-2.4, 0.8+-1.5)	GeForce GTX TITAN X GPU, Intel Core i7-5820K 3.3GHz 6-Core Processor, X99 gaming 5 x99 ATX DDR44 motherboard	ข้อดี คือ Processing Time เร็วกว่า MLP, low-latency, ข้อเสีย คือ ใช้ข้อมูลจำนวนมาก, อาจเกิดปัญหา Overfitting
H. Kim J. Park และ S. Lee	TFC-TDF(U-Net)	MUSDB	(-, -, 7.07+-.08, -)	-	ข้อดี คือ ประหยัดพารามิเตอร์,รองรับเสียงรบกวน ข้อเสีย คือ ใช้ชุดข้อมูลมาก, ปรับแต่งได้ยาก
Daniel Stoller, Sebastian Ewert และ Simon Dixon	M6(Wave-U-Net)	MUSDB	(2.91, 4.15, 3.0, 2.03)	-	ข้อดี คือ รองรับเสียงรบกวน ข้อเสีย คือ ใช้ข้อมูลมาก, อาจเกิดปัญหา Overfitting
Yi Luo และ Jianwei Yu	Band-split RNN (BSRNN)	MUSDB18 และ MUSDB18-HQ	MUSDB18:(7.51, 8.58, 10.21, 6.62) MUSDB18-HQ:(7.22, 9.01, 10.01, 6.70)	-	ข้อดี คือประสิทธิภาพการแยกแหล่งเสียงที่ดีกว่าโมเดล MSS ทั่วไป ข้อเสีย คือการแยกเสียงเบสยังด้อยกว่าโมเดลอื่นๆ เล็กน้อย

Stefan Uhlich , Marcello Porcu , Franck Giron , Michael Enenkl , Thomas Kemp , Naoya Takahashi และ Yuki Mitsufuji	แบบผสมผสาน (hybrid neural network) ได้แก่ CNN และ RNN	DSD100	BLEND (SWF):(2.76, 3.93, 5.13, 3.37) BLEND (MWF):(2.98, 4.13, 5.23, 3.52)	-	ข้อดี คือเมื่อมีระดับเสียงรบกวนที่พื้นหลังและประเภทของเพลงจะส่งผลต่อโมเดลได้น้อย ข้อเสีย คือการใช้ชุดข้อมูลที่มีขนาดใหญ่และความซับซ้อนของโมเดลจะส่งผลต่อการใช้งานทำให้ใช้งานได้ยากขึ้น
Kim, M., Lee, J., & Lee, H.	TFC-TDF-U-Net v2	MUSDB18	(7.86, 7.33, 9.00, 5.95)	Ubuntu 20.04 at least four cuda-able GPUs (each >= 2080ti) 1.5 TB disk storage for data augmentation wandb for logging	ข้อดี คือมีประสิทธิภาพในการแยกเสียงดนตรีออกจากสัญญาณเสียงผสม ข้อเสีย คือสัญญาณเสียงรบกวนจะส่งผลต่อการแยกเสียงอย่างมาก
Alex Favaro Aaron Lewis และ Garrett Schlesinger	ICA	private dataset	-	-	ข้อดี คือ ไม่ใช้ชุดข้อมูลในการฝึกมาก ข้อเสีย คือ การเกิด Underfitting, ไวต่อสัญญาณรบกวน

ภาพที่ 2.2.1 การเปรียบเทียบโมเดล

บทที่ 3
วิธีดำเนินการวิจัย

บทนี้กล่าวถึงวิธีการดำเนินการวิจัย ซึ่งมีเป้าหมายที่จะพัฒนาโมเดลแยกเสียงเครื่องดนตรีไทยจากเพลงไทยเดิม โดยนำสถาปัตยกรรมที่เกี่ยวกับ Convolutional Neural Network ชื่อว่า Wave-U-Net ซึ่งใช้ในการแยกแหล่งกำเนิดเสียง มาใช้ในในการพัฒนาโมเดลที่จะแยกเสียงเครื่องดนตรีไทยออกจากเพลงไทยเดิมได้อย่างมีประสิทธิภาพ นอกจากนี้ยังใช้เทคนิคอื่นๆ เช่น Data augmentation, Padding, Dropout, Adam Optimizer, Kernel Initializers, Learning rate schedules เพื่อเพิ่มประสิทธิภาพและความแม่นยำในการแยกเสียงเครื่องดนตรีไทย
โดยมีขั้นตอนในการดำเนินการดังกล่าว ต่อไปนี้

3.1 เครื่องมือที่ใช้ในการพัฒนา

ในการพัฒนาโมเดลแยกเสียงเครื่องดนตรีไทย การเตรียมเครื่องมือที่เหมาะสมเป็นสิ่งสำคัญ คณะผู้จัดทำได้เลือกใช้เครื่องมือและซอฟต์แวร์ต่าง ๆ ดังนี้

3.1.1 ฮาร์ดแวร์

คณะผู้จัดทำได้เลือกใช้เครื่องมือพัฒนาการวิจัย ได้แก่ คอมพิวเตอร์ระบบปฏิบัติการ Windows ซึ่งประกอบไปด้วย คอมพิวเตอร์แบบพกพา จำนวน 2 เครื่อง และ คอมพิวเตอร์แบบตั้งโต๊ะ จำนวน 1 เครื่อง เพื่อใช้ในการนำเข้าวิดีโอเสียงเพลงไทยเดิม และ ตัดต่อเสียงเพื่อใช้ในการสร้าง Dataset

3.1.2 ซอฟต์แวร์

คณะผู้จัดทำได้เลือกใช้ซอฟต์แวร์สำหรับการสร้าง Dataset และ พัฒนาโมเดลเป็น Google Colaboratory หรือ Colab ซึ่งเป็นซอฟแวร์ที่สามารถเขียนและเรียกใช้ Python ในเบราว์เซอร์ ช่วยให้คณะผู้จัดทำสามารถทำการสร้างโมเดลโดยใช้ทรัพยากรบน Cloud ได้

3.1.3 ภาษาโปรแกรม

คณะผู้จัดทำได้เลือกใช้ Python เป็นภาษาที่ใช้ในการพัฒนา เนื่องจากเป็นภาษาที่มีความนิยมในการพัฒนาโมเดล เพราะ ประกอบด้วยไลบรารีที่หลากหลาย และ เอื้ออำนวยต่อการทำ Machine Learning

3.1.4 ไลบรารีและเครื่องมือ

คณะผู้จัดทำได้เลือกใช้ไลบรารีและเครื่องมือที่สำคัญดังต่อไปนี้

1. TensorFlow Keras คือ ไลบรารีสำหรับการสร้างและฝึกอบรมโมเดลของ deep learning มีฟังก์ชันสำหรับการสร้างโมเดล Input, Conv1D, MaxPooling1D, Conv1DTranspose, Concatenate, Dropout, Model และ he_normal เพื่อเพิ่มประสิทธิภาพในการแยกเสียงเครื่องดนตรีไทย

2. Pandas คือ ไลบรารีสำหรับการจัดการและวิเคราะห์ข้อมูลในรูปแบบของ DataFrame ซึ่งคล้ายกับตารางในฐานข้อมูล สามารถใช้สำหรับการประมวลผลข้อมูลที่มีโครงสร้าง เช่น csv ไฟล์เป็นต้น

3 NumPy คือ ไลบรารีสำหรับการคำนวณเชิงตัวเลข สามารถจัดการกับอาร์เรย์หลายมิติ และให้ฟังก์ชันทางคณิตศาสตร์สำหรับจัดการข้อมูลได้

4. Time คือ ไลบารีสำหรับการจัดการกับเวลาต่าง ๆ เช่น การหยุดเวลาชั่วคราว หรือ การจับเวลาการทำงานของโค้ด

5. OS คือ ไลบารีสำหรับการโต้ตอบกับระบบปฏิบัติการ เช่น การจัดการไฟล์และไดเร็กทอรี

6. Random คือ ไลบารีสำหรับการสุ่ม เช่น การสุ่มเลือกจากลิสต์ เป็นต้น

7. Librosa คือ ลบรารีสำหรับการวิเคราะห์และจัดการกับเสียงในงานประมวลผลเสียงดิจิทัล สามารถใช้โหลดไฟล์เสียง คำนวณฟีเจอร์ และแสดงผลข้อมูลเสียง

8. Matplotlib คือ ไลบารีสำหรับการสร้างกราฟและการแสดงผลข้อมูลต่างๆ

9. PyTube คือ ไลบรารีสำหรับการดาวน์โหลดวิดีโอจาก YouTube

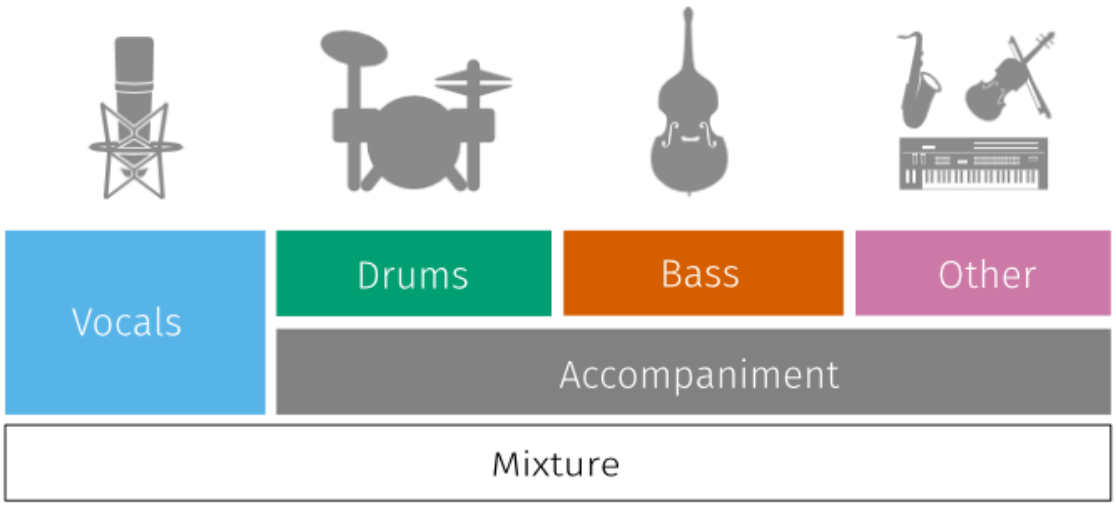
10. Pydub คือ ไลบรารีสำหรับการจัดการและประมวลผลไฟล์เสียง เช่น การแปลงไฟล์ การตัดต่อ และการรวมไฟล์เสียง

11. Sklearn คือ ไลบรารีภาษา ที่ใช้สำหรับการทำ machine learning และ data mining มีฟังก์ชันและเครื่องมือที่หลากหลายเพื่อการสร้างและฝึกอบรมโมเดลทางสถิติ และ machine learning เช่น การแบ่งข้อมูลเป็นชุดฝึก (training set) เป็นต้น

3.2 เตรียมชุดข้อมูล

ในการพัฒนาโมเดลแยกเสียงเครื่องดนตรีไทยจากเพลงไทยเดิม ทางคณะผู้จัดทำพยามหาชุดข้อมูลที่มีอยู่แล้ว แต่ไม่พบชุดข้อมูลในรูปแบบเพลงไทยเดิมที่มีการแยกเสียงเครื่องดนตรีไทยซึ่งตรงกับความต้องการของโปรเจคที่พบมีเพียงชุดข้อมูลในลักษณะที่ต้องการแต่เป็นเพลง และ เครื่องดนตรีสากล หรือภาษาอื่นๆเท่านั้น ดังนั้นคณะผู้จัดทำจึงตัดสินใจที่จะสร้างชุดข้อมูลของตัวเองขึ้นมา โดยมีขั้นตอนการสร้างชุดข้อมูลดังนี้

3.2.1 ทำการเลียนแบบชุดข้อมูล MUSDB18



ภาพที่ 3.2.1.1 แสดงรูปแบบไฟล์ในชุดข้อมูล MUSDB18

ที่มา : https://sigsep.github.io/datasets/musdb.html

3.2.2 เลือกเครื่องดนตรีไทยเดิม 3 ลักษณะ ได้แก่

1. เครื่องสี คือ ซอ
2. เครื่องตี คือ ระนาด
3. เครื่องเป่า คือ ขลุ่ย

3.2.3 ทำการนำเข้าไลบารีที่ใช้ในการจัดเตรียมชุดข้อมูลก่อนฝึก

1. pytube ใช้ในการดาวน์โหลดเสียงเพลงเดี่ยวไทยเดิมจากเว็บไซต์ Youtube
2. pydub ใช้ในการแบ่งเสียง และ จัดเก็บไฟล์เสียงในรูปแบบ mp3
3. os ใช้ในการสร้างโฟลเดอร์ และ ควรคุมไดเร็กทอรี
4. Numpy ใช้ในการประมวลผลทางคณิตศาสตร์
5. Random ใช้ในการสุ่มค่าตัวเลข
6. Shutil ใช้ในการลบโฟลเดอร์

3.2.4 ดาวน์โหลดเสียงเพลงเดี่ยวไทยเดิมจากเว็บไซต์ Youtube ตามเครื่องดนตรีที่เลือกโดยให้เสียงอยู่ในรูปแบบของ mp3

1. เครื่องดนตรี : ซอ

   ชื่อคลิป : เดี่ยวซออู้ ไพเราะเพราะพริ้ง-อ.สมบูรณ์ บุญวงศ์

   อดีตผช.อธิการบดีสถาบันบัณฑิตพัฒนศิลป์

   โดย : Siammelody

   จำนวนเพลง: 11 เพลง

   ระยะเวลา : 1 ชั่วโมง 18 วินาที

   ลิงค์วิดีโอ : https://youtu.be/3yw6ZS8m7H8

2. เครื่องดนตรี : ระนาด

   ชื่อคลิป : เดี่ยวระนาด ไพเราะ เสนาะหู โดยชัยยุทธ โตสง่า (ป๋อม บอยไทย) ศิลปิน ศิลปาธร EP.1 (Thai Classical Music)

   โดย : Siammelody

   จำนวนเพลง: 10 เพลง

   ระยะเวลา : 1 ชั่วโมง 4 วินาที

   ลิงค์วิดีโอ : https://youtu.be/CDxQT5xPPUA

3. เครื่องดนตรี : ขลุ่ย

   ชื่อคลิป : เพลงไทยเดิม EP.10 เดี่ยวขลุ่ย Thai Classical Music เพลงกล่อมนอนหลับสบาย คลายกังวล

   โดย : Siammelody

   จำนวนเพลง: 14 เพลง

   ระยะเวลา : 1 ชั่วโมง 7 นาที 45 วินาที

   ลิงค์วิดีโอ : https://youtu.be/8jmqG2aOgxg

3.2.5 แปลงชุดข้อมูลก่อนฝึกโมเดล

โดยคณะผู้จัดทำได้ใช้วิธีการประมวลเสียง
เพื่อทำการเตรียมชุดข้อมูลให้อยู่ในรูปแบบที่สามารถนำไปใช้วิเคราะห์หรือการเรียนรู้โมเดล

3.2.5.1 จัดการเสียงโดยตัดความยาวของเสียง

โดยในขั้นตอนนี้คณะผู้จัดทำได้ทำการตัดเสียงเป็นชุดละ 30
วิโดยที่เสียงที่มีความยาวไม่ถึง 30 วินาทีจะทำการเพิ่มความยาวเสียงด้วยความเงียบ

```python
def split_audio(audio_name, audio_file_path, segment_duration=30 * 1000):
    audio = AudioSegment.from_file(audio_file_path)
    segment_duration_ms = segment_duration
    for i in range(0, len(audio), segment_duration_ms):
        split_path = f"./DSTHAI/tack{(i // segment_duration) + 1}"
        if not os.path.exists(split_path):
          os.makedirs(split_path)
        segment = audio[i:i + segment_duration_ms]
        if len(segment) < segment_duration_ms:
            silence_duration = segment_duration_ms - len(segment)
            segment += AudioSegment.silent(duration=silence_duration)
        segment.export(f"{split_path}/{audio_name}.mp3", format="mp3")
```

ภาพที่ 3.2.5.1.1 แสดงฟังก์ชั่นตัดเสียงเครื่องดนตรีทั้ง 3

3.2.5.2 ทำการผสมเสียงของเครื่องดนตรีทั้ง 3 เข้าด้วยกัน

3.2.5.3 ทำการเพิ่มขนาดของชุดด้วยการทำ Data Augmentation ดังนี้

1. พิ่มเสียงรบกวนโดยการใช้ค่า normal distribution กำหนดให้ค่าเฉลี่ย(mean) เป็น 0 และ ส่วนเบี่ยงเบนมาตรฐาน (standard deviation) เป็น 0.02 * 32767

2. เปลี่ยนระดับเสียง เป็นค่าสุ่มระหว่าง -2 ถึง 2 เท่าของเสียงเดิม

3. เปลี่ยนความเร็วเสียงเป็นค่าสุ่มระหว่าง 0.9 ถึง 1.1 เท่าของเสียงเดิม

```python
def add_noise(audio_segment, noise_level=0.05):
    noise = AudioSegment(
        np.random.normal(0, noise_level * 32767, len(audio_segment.get_array_of_samples())).astype(np.int16).tobytes(),
        frame_rate=audio_segment.frame_rate,
        sample_width=audio_segment.sample_width,
        channels=audio_segment.channels
    )
    return audio_segment.overlay(noise)

def change_pitch(audio_segment, semitones=2):
    new_sample_rate = int(audio_segment.frame_rate * (2.0 ** (semitones / 12.0)))
    return audio_segment._spawn(audio_segment.raw_data, overrides={'frame_rate': new_sample_rate}).set_frame_rate(audio_segment.frame_rate)

def change_speed(audio_segment, speed=1.2):
    return audio_segment._spawn(audio_segment.raw_data, overrides={'frame_rate': int(audio_segment.frame_rate * speed)}).set_frame_rate(audio_segment.frame_rate)
```

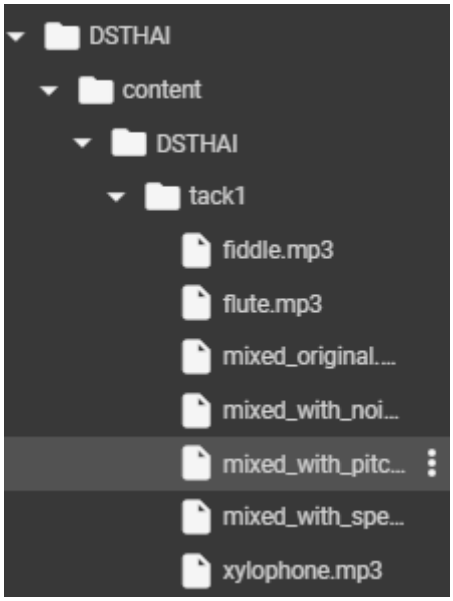
ภาพที่ 3.2.5.3.1 แสดงฟังก์ชั่น Augmentation ทั้ง 3
(เพิ่มเสียงรบกวน, การเพิ่มระดับเสียง และ การเพิ่มความเร็วเสียง)

3.2.5.4 แยกไฟล์เสียงลงในโฟลเดอร์ DSTHAI

โดยในขั้นตอนนี้คณะผู้จัดทำได้แยกย่อยไฟล์เสียงเป็น track โฟลเดอร์โดยให้ชื่อไฟล์เป็นชื่อของเครื่องดนตรี ชื่อของเสียงที่ผสมเป็น mixed_original และชื่อของเสียง ผสมที่ผ่านการทำData Augmentation จะแยกออกเป็น mixed_with_noise, mixed_with_pitch และ mixed_with_speed ตามลำดับที่ได้อธิบายไว้ใน 3.2.5.3 ซึ่งหากโฟลเดอร์ track ไหนที่ไม่มีเครื่องดนตรีครบทั้ง 3 จะถูกลบทิ้ง

```python
# Loop through each track folder
for track_folder in track_folders:
    print(f'Processing {track_folder}...')
    # List all audio files in the current track folder
    audio_files = [f.path for f in os.scandir(track_folder) if f.is_file() and f.name.endswith('.mp3')]

    # Check if there are more than one audio file
    if len(audio_files) > 1:
        # Load the audio files
        audio_segments = [AudioSegment.from_file(audio_file) for audio_file in audio_files]

        # Mix the audio files together
        mixed_audio = audio_segments[0]
        for segment in audio_segments[1:]:
            mixed_audio = mixed_audio.overlay(segment)

        # Define the output folder for the mixed audio files
        output_folder = os.path.join("/content/DSTHAI", os.path.basename(track_folder))
        if not os.path.exists(output_folder):
            os.makedirs(output_folder)

        # Save the original mixed audio
        mixed_audio.export(os.path.join(output_folder, 'mixed_original.mp3'), format='mp3')

        # Apply and save different augmentations
        augmented_audio = add_noise(mixed_audio, noise_level=0.02)
        augmented_audio.export(os.path.join(output_folder, 'mixed_with_noise.mp3'), format='mp3')

        augmented_audio = change_pitch(mixed_audio, semitones=random.randint(-2, 2))
        augmented_audio.export(os.path.join(output_folder, 'mixed_with_pitch_shift.mp3'), format='mp3')

        augmented_audio = change_speed(mixed_audio, speed=random.uniform(0.9, 1.1))
        augmented_audio.export(os.path.join(output_folder, 'mixed_with_speed_change.mp3'), format='mp3')

    else:
        print(f'Deleting {track_folder} as it does not contain more than 1 instrument')
        shutil.rmtree(track_folder)

print('Processing complete.')
```

ภาพที่ 3.2.5.4.1 *แสดงฟังก์ชั่นการจัดการโฟลเดอร์ DSTHAI และ ไฟล์เสียงผสม*



ภาพที่ 3.2.5.4.2 *แสดงรูปแบบโฟลเดอร์ DSTHAI ที่เก็บชุดข้อมูลสำหรับฝึกโมเดลไว้*

3.2.6 เตรียมพร้อมชุดข้อมูล

หลังจากที่ทำการแปลง จัดวางชุดข้อมูลให้อยู่ในรูปแบบที่เหมาะสมโดยเลียนแบบชุดข้อมูล MUSDB18 แล้ว ขั้นตอนต่อไปจะเป็นการเตรียม และ แบ่งชุดข้อมูลที่ใช้ในการฝึกฝนโมเดล

3.2.6.1 ทำการนำเข้าไลบารีที่ใช้ในการเตรียมพร้อมชุดข้อมูล

1. train_test_split ใช้ในการแบ่งชุดข้อมูลออกเป็น ชุดที่ใช้ฝึก และ ทดสอบประสิทธิภาพของโมเดล

3.2.6.2 การโหลดข้อมูลและสร้างลำดับ

โหลดข้อมูลและสร้างลำดับไว้ในตัวแปรลิสต์ที่ชื่อ tracks

```
def load_data(dataset_path, duration=30):
    tracks = []
    required_files = ['mixed_original.mp3', 'mixed_with_noise.mp3', 'mixed_with_pitch_shift.mp3', 'mixed_with_speed_change.mp3', 'fiddle.mp3', 'flute.mp3', 'xylophone.mp3']

    for folder in os.listdir(dataset_path):
      for mixed_file in ['mixed_original.mp3', 'mixed_with_noise.mp3', 'mixed_with_pitch_shift.mp3', 'mixed_with_speed_change.mp3']:
        folder_path = os.path.join(dataset_path, folder)
        if os.path.isdir(folder_path):
            files_in_folder = os.listdir(folder_path)
            if all(file in files_in_folder for file in required_files):
                mixture_path = os.path.join(folder_path, mixed_file)
                fiddle_path = os.path.join(folder_path, 'fiddle.mp3')
                flute_path = os.path.join(folder_path, 'flute.mp3')
                xylophone_path = os.path.join(folder_path, 'xylophone.mp3')

                mixture, sr = librosa.load(mixture_path, sr=None, duration=duration)
                fiddle, _ = librosa.load(fiddle_path, sr=sr, duration=duration)
                flute, _ = librosa.load(flute_path, sr=sr, duration=duration)
                xylophone, _ = librosa.load(xylophone_path, sr=sr, duration=duration)

                if len(mixture) == duration * sr and len(fiddle) == duration * sr and len(flute) == duration * sr and len(xylophone) == duration * sr:
                    mixture = mixture.reshape(-1, 1)
                    instruments = np.stack([fiddle, flute, xylophone], axis=-1)
                    tracks.append((mixture, instruments))
    return tracks, sr

dataset_path = '/content/DSTHAI/content/DSTHAI'
tracks, sample_rate = load_data(dataset_path)
```

ภาพที่ 3.2.6.2.1 *แสดงฟังก์ชันการโหลดข้อมูลและสร้างลำดับ*

3.2.6.3 การแบ่งชุดข้อมูล

ในขั้นตอนนี้คณะผู้จัดทำแบ่งได้แบ่งชุดข้อมูลออกเป็น ชุดฝึกโมเดล และ ชุดทดสอบโมเดล โดยแบ่งชุดฝึก 80 เปอร์เซ็นต์ และ ชุดทดสอบ 20 เปอร์เซ็นต์

```
def split_dataset(tracks, split_ratio=0.8):
    train_tracks, val_tracks = train_test_split(tracks, train_size=split_ratio, random_state=42)
    return train_tracks, val_tracks
train_tracks, val_tracks = split_dataset(tracks, split_ratio=0.8)
```

ภาพที่ 3.2.6.3.1 *แสดงฟังก์ชันการแบ่งชุดข้อมูล*

3.2.6.4 การสร้างชุดข้อมูลแบบกำหนดขนาดของชุดข้อมูล (batch)

ในขั้นตอนนี้คณะผู้จัดทำกำหนดให้ขนาดของชุดข้อมูลเป็น 12 ชุดสำหรับการฝึกแต่ละครั้ง

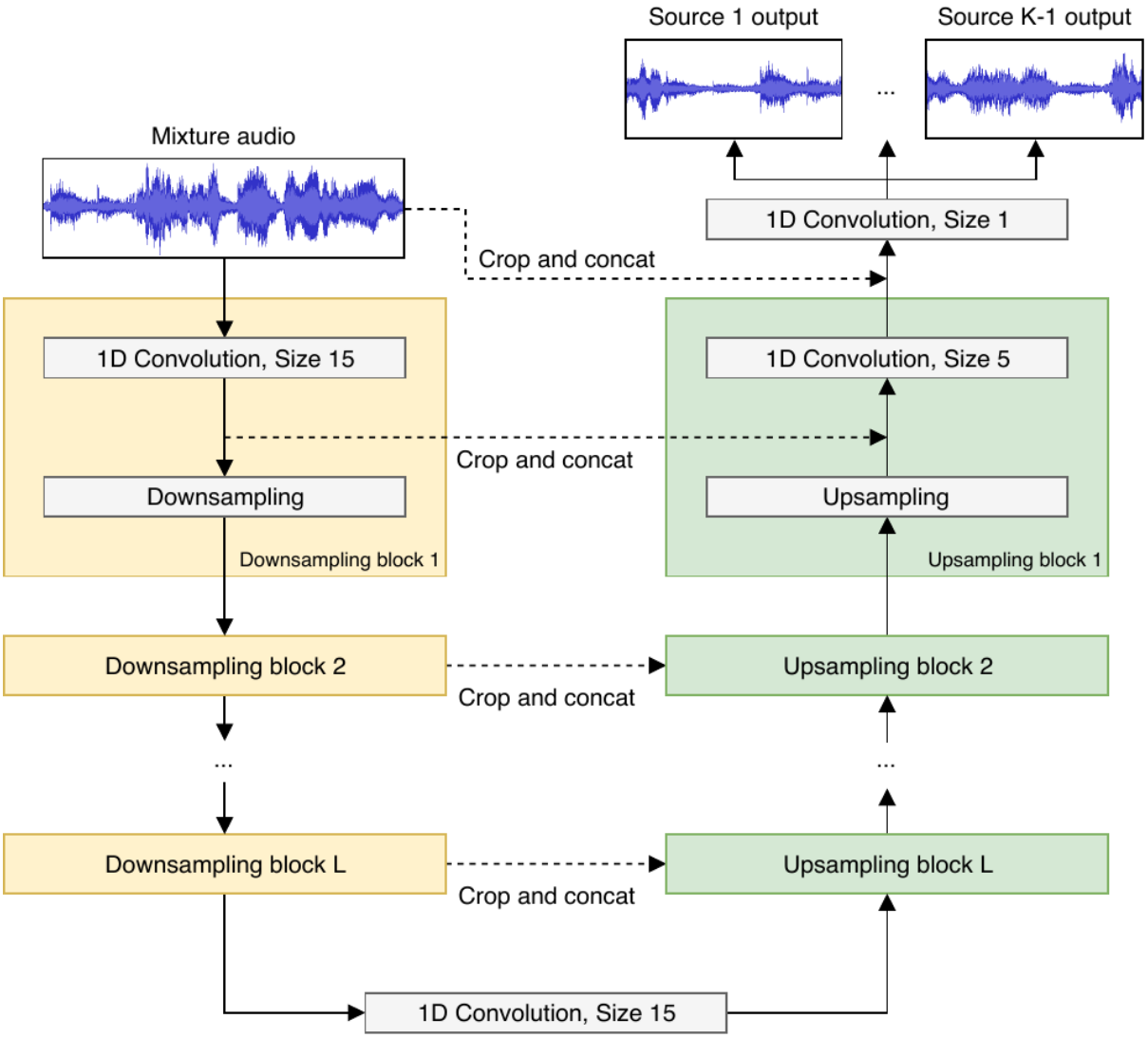
```
def data_generator(tracks, batch_size):
    while True:
        for start in range(0, len(tracks), batch_size):
            end = min(start + batch_size, len(tracks))
            batch_tracks = tracks[start:end]

            mixtures = np.array([track[0] for track in batch_tracks])
            instruments = np.array([track[1] for track in batch_tracks])

            yield mixtures, instruments

batch_size = 12
train_generator = data_generator(train_tracks, batch_size)
val_generator = data_generator(val_tracks, batch_size)
```

ภาพที่ 3.2.6.4.1 *แสดงฟังก์ชันสร้างชุดข้อมูลแบบกำหนดขนาดของชุดข้อมูล (batch)*

3.3 สร้าง และ ฝึกโมเดล

3.3.1 การสร้างโมเดล

ในส่วนนี้จะอธิบายการสร้างโมเดลอย่างละเอียด รวมถึงแต่ละส่วนประกอบของโมเดลที่ใช้ในการแยกเสียงเครื่องดนตรีไทยจากเพลงไทยเดิม โดยรูปแบบโมเดลหลักจะอ้างอิงจาก สถาปัตยกรรมชื่อว่า Wave-U-Net เป็นหลัก



ภาพที่ 3.3.1.1 แสดงโครงสร้างของสถาปัตยกรรม Wave-U-Net

ที่มา: https://github.com/f90/Wave-U-Net

3.3.1.1 การนำเข้าไลบารีที่จำเป็น

1. Input ใช้เพื่อกำหนดชั้น input ของโมเดล เป็นจุดเริ่มต้นของโมเดล neural network ซึ่งใช้ในการกำหนดรูปร่างของข้อมูลที่จะป้อนเข้าโมเดล
2. Conv1D ใช้เพื่อกำหนดชั้น input ของโมเดล เป็นจุดเริ่มต้นของโมเดล neural network ซึ่งใช้ในการกำหนดรูปร่างของข้อมูลที่จะป้อนเข้าโมเดล
3. MaxPooling1D ใช้สำหรับการลดขนาดของข้อมูลในมิติหนึ่งมิติ โดยการใช้ฟังก์ชัน max pooling ซึ่งจะเลือกค่าสูงสุดในแต่ละหน้าต่างย่อยของข้อมูล
4. Conv1DTranspose ใช้สำหรับการ upsampling ข้อมูลในมิติหนึ่งมิติ ซึ่งทำหน้าที่ตรงข้ามกับ Conv1D
5. Concatenate ใช้สำหรับการรวมข้อมูลจากหลาย ๆ ชั้นเข้าด้วยกันตามแกนที่กำหนด

6. Dropout ใช้เพื่อป้องกันการ overfitting โดยการสุ่มปิดหน่วยประมวลผล (neurons) บางส่วนในระหว่างการฝึกโมเดล

7. he_normal ใช้สำหรับการกำหนดค่าเริ่มต้นของน้ำหนักในชั้นคอนโวลูชั่น ซึ่งช่วยให้การฝึกโมเดลมีความเสถียรและรวดเร็วมากขึ้น

8. csv ใช้สำหรับการเขียน และ สร้างไฟล์ csv

9. resource ใช้สำหรับเก็บข้อมูลทรัพยากรที่ใช้ใน colab

3.3.1.2 การสร้าง Downsampling Block

ในขั้นตอนนี้คณะผู้จัดทำได้สร้าง Downsampling block เพื่อทำให้ข้อมูลลดมิติลง โดยภายใน block นี้ จะมีการใช้เลเยอร์ Convolutional 1D (Conv1D) เพื่อกรองข้อมูล และทำการส่งออกค่านั้นเป็น skip connection จากนั้นจะตามด้วยเลเยอร์ MaxPooling 1D เพื่อลดมิติของข้อมูล ก่อนกจะส่งค่านั้นออกมาเช่นกัน

3.3.1.3 การสร้าง Upsampling Block

ในขั้นตอนนี้คณะผู้จัดทำได้สร้าง Upsampling block เพื่อเพิ่มมิติของข้อมูลกลับ มา โดยภายใน block นี้ จะใช้เลเยอร์ Convolutional Transpose 1D (Conv1DTranspose) เพื่อเพิ่มมิติของข้อมูล และ Concatenate กับ skip connection ที่เก็บไว้จากขั้นตอนการ downsampling

3.3.1.4 การกำหนดขนาดของเลเยอร์

ในขั้นตอนนี้คณะผู้จัดทำได้ทำการกำหนดขนาดในแต่ละเลเยอร์ ซึ่งจะมีการเพิ่มลดบางส่วนในการสรุปผลโดยเป็นการสร้างโมเดลหลายรูปแบบเพื่อเปรียบเทียบผลลัพธ์

1. Input Layer : (None, 1)
2. Downsampling Block 1 :
   Conv1D: (None, 64) ⟶ 64 ฟิลเตอร์, kernel_size=15
   MaxPooling1D: (None, 32) ⟶ pool_size=2
3. Downsampling Block 2 :
   Conv1D: (None, 128) ⟶ 128 ฟิลเตอร์, kernel_size=15
   MaxPooling1D: (None, 64) ⟶ pool_size=2
4. Downsampling Block 3 :
   Conv1D: (None, 256) ⟶ 256 ฟิลเตอร์, kernel_size=15
   MaxPooling1D: (None, 128) ⟶ pool_size=2
5. Bottleneck :
   Conv1D: (None, 512) ⟶ 512 ฟิลเตอร์, kernel_size=15

6. Upsampling Block 1 :

Conv1DTranspose: (None, 256) ⟶ 256 ฟิลเตอร์, kernel_size=5, strides=2

Concatenate: (None, 512) ⟶ รวมค่า output กับ skip connection

7. Upsampling Block 2:

   Conv1DTranspose: (None, 128) ⟶ 128 ฟิลเตอร์, kernel_size=5, strides=2

   Concatenate: (None, 256) ⟶ รวมค่า output กับ skip connection

8. Upsampling Block 3:

   Conv1DTranspose: (None, 64) ⟶ 64 ฟิลเตอร์, kernel_size=5, strides=2

   Concatenate: (None, 128) ⟶ รวมค่า output กับ skip connection

9. Output Layer:

   Conv1D: (None, 3) ⟶ 3 ฟิลเตอร์, kernel_size=1, activation='linear'

```python
def create_wave_unet_model(input_shape):
    def downsampling_block(x, filters, kernel_size, pool_size):
        conv = Conv1D(filters, kernel_size, activation='relu', padding='same')(x)
        downsampled = MaxPooling1D(pool_size)(conv)
        return downsampled, conv

    def upsampling_block(x, skip_connection, filters, kernel_size, upsample_size):
        upsampled = Conv1DTranspose(filters, kernel_size, strides=upsample_size, activation='relu', padding='same', kernel_initializer=he_normal())(x)
        concat = Concatenate()([upsampled, skip_connection])
        return concat

    inputs = Input(shape=input_shape)
    skip_connections = []
    x = inputs

    # Downsampling blocks
    for filters in [64, 128, 256]:
        x, skip = downsampling_block(x, filters, 15, 2)
        skip_connections.append(skip)

    # Bottleneck
    x = Conv1D(512, 15, activation='relu', padding='same')(x)

    # Upsampling blocks
    for filters, skip in zip([256, 128, 64], reversed(skip_connections)):
        x = upsampling_block(x, skip, filters, 5, 2)

    outputs = Conv1D(3, 1, activation='linear', padding='same')(x)

    model = Model(inputs, outputs)
    return model
```

ภาพที่ 3.3.1.4.1 แสดงฟังก์ชันการสร้าง Wave-U-Net โมเดล

3.3.2 การคอมไพล์โมเดล

คณะผู้จัดทำได้ทำการคอมไพล์โมเดลโดยใช้ Optimizer เป็น Adam optimize โดยได้กำหนดพารามิเตอร์ดังนี้

1. Learining rate มีการเปลี่ยนแปลงในแต่ละโมเดล
2. Bata_1 = 0.9
3. Beta_2 = 0.999
4. Epsilonn = 1e-07
5. Loss function = Mean Squared Error (MSE)
6. Metrics = Mean Absolute Error (MAE) มีการเปลี่ยนแปลงในแต่ละโมเดล

```
adam_optimizer = tf.keras.optimizers.Adam(learning_rate=0.004, beta_1=0.9, beta_2=0.999, epsilon=1e-07)
model.compile(optimizer=adam_optimizer, loss='mse', metrics=[tf.keras.metrics.MeanAbsoluteError()])
```

ภาพที่ 3.3.2.1 แสดงฟังก์ชันการ Compile Wave-U-Net โมเดล

3.3.3 การตั้งค่า Callbacks

ในขั้นตอนนี้คณะผู้วิจัยได้ทำการสร้าง Callback สำหรับควบคุมการฝึกโมเดล และ การเก็บรวบรวบข้อมูลระหว่างการฝึก โดยมีการทำงานดังนี้

1. Early stopping คือ การหยุดฝึกโมเดลเมื่อค่า val_loss ไม่มีความพัฒนาใน 6 epoch
2. Learning rate scheduler คือ การลดอัตราการเรียนรู้ลงเมื่อค่า val_loss ไม่มีความพัฒนา ซึ่งมีการเปลี่ยนแปลงวิธีการที่ใช้ในแต่ละโมเดล
3. Checkpoint คือ การจัดเก็บโมเดลโดยกำหนดให้เก็บทุกๆ 5 epoch
4. keepDataCSV คือ Callback ที่ทางคณะผู้จัดทำสร้างขึ้นเพื่อเก็บข้อมูลในระหว่างการฝึกโมเดล ได้แก่ Epoch Number, Training Loss, Validate Loss, MAE, Validate MAE, Learning Rate, Elapsed Time, RAM, CPU

```python
class keepDataCSV(tf.keras.callbacks.Callback):
    def __init__(self, save_path):
        self.save_path = save_path
        self.csv_file = open(os.path.join(save_path, 'training_stats.csv'), 'w')
        self.csv_writer = csv.writer(self.csv_file)
        self.csv_writer.writerow(['Epoch Number', 'Training Loss', 'Validate Loss', 'MAE', 'Validate MAE', 'Learning Rate', 'Elapsed Time', 'RAM', 'CPU'])

    def on_train_begin(self, logs=None):
        # Record the start time when training begins
        self.start_time = time.time()

    def on_epoch_end(self, epoch, logs={}):
        # Calculate RAM and CPU usage
        ram_usage = resource.getrusage(resource.RUSAGE_SELF).ru_maxrss / 1024 ** 2
        cpu_usage = resource.getrusage(resource.RUSAGE_SELF).ru_utime + resource.getrusage(resource.RUSAGE_SELF).ru_stime

        # Write epoch data to CSV
        self.csv_writer.writerow([epoch + 1, logs.get('loss'), logs.get('val_loss'), logs.get('mean_absolute_error'), logs.get('val_mean_absolute_error'),
                                  logs.get('lr'), time.time() - self.start_time, ram_usage, cpu_usage])
        self.csv_file.flush()

# Create the callback object
save_path = '/content/ModelComplete'
keepDataCSV = keepDataCSV(save_path)
# Callbacks


# Define callbacks
checkpoint = ModelCheckpoint('/content/ModelComplete/model_{epoch:02d}.h5', save_freq='epoch', period=5)

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.25, patience=3, min_lr=1e-6)

early_stopping = EarlyStopping(monitor='val_loss', patience=6, restore_best_weights=True)
```

ภาพที่ 3.3.3.1 แสดงฟังก์ชันสำหรับการ Callbacks

3.3.4 การฝึกโมเดล

การฝึกโมเดลโดยใช้ข้อมูลที่เตรียมเอาไว้โดยตั้งค่า epochs จำนวน 1000 รอบเพื่อให้โมเดลมีโอกาสเรียนรู้จากข้อมูลมากเพียงพอ ใช้ข้อมูลที่ทำการแยกเอาไว้มา Validation เพื่อติดตามประสิทธิภาพของโมเดล

```python
model.fit(train_generator, epochs=epochs, callbacks=[early_stopping, lr_scheduler, checkpoint, keepDataCSV],
          steps_per_epoch=steps_per_epoch, validation_data=val_generator, validation_steps=validation_steps)
```

ภาพที่ 3.3.4.1 แสดงการฝึกฝนโมเดลด้วยคำสั่ง fit

3.4 ทดสอบประสิทธิภาพโมเดล

ในส่วนนี้จะอธิบายขั้นตอนการทดสอบประสิทธิภาพของโมเดล โดยวัดชี้วัดที่ใช้ในการวัดคุณภาพของการแยกเสียง Signal-to-Distortion Ratio (SDR) และ การทดสอบด้วย Subjective ซึ่งคือการประเมินจากประสบการณ์ของผู้ฟัง

3.4.1 การนำเข้าไลบรารีที่จำเป็น

ไลบารีที่ใช้ในการดาวโหลดไฟล์เสียง, จัดการโมเดล, และการคำควณทางคณิตศาสตร์ การใช้ mir_eval ในการทดสอบประสิทธิภาพ

```
import soundfile as sf
import os
import tensorflow as tf
import numpy as np
from pytube import YouTube
from pydub import AudioSegment
import librosa
```

ภาพที่ 3.4.1.1 แสดงการนำเข้าไลบารีที่ใช้

3.4.2 การโหลดโมเดล

ทำการโหลดโมเดลที่ได้ผ่านการฝึก

```
# Specify the path to your file in Google Drive
file_path = '/content/TrainComplete50/content/ModelComplete/model_50.h5'

# Load the model
model = tf.keras.models.load_model(file_path)
```

ภาพที่ 3.4.2.1 แสดงการโหลดโมเดล

3.4.3 การใช้โมเดลแยกเสียง

ทำการแยกเสียงโดยใช้ฟังก์ชั่นในการเรียกใช้งานโมเดล

```
def create_wave_unet_model(input_shape):
    def downsampling_block(x, filters, kernel_size, pool_size):
        conv = Conv1D(filters, kernel_size, activation='relu', padding='same')(x)
        downsampled = MaxPooling1D(pool_size)(conv)
        return downsampled, conv

    def upsampling_block(x, skip_connection, filters, kernel_size, upsample_size):
        upsampled = Conv1DTranspose(filters, kernel_size, strides=upsample_size, activation='relu', padding='same', kernel_initializer=he_normal())(x)
        concat = Concatenate()([upsampled, skip_connection])
        return concat

    inputs = Input(shape=input_shape)
    skip_connections = []
    x = inputs

    # Downsampling blocks
    for filters in [64, 128, 256]:
        x, skip = downsampling_block(x, filters, 15, 2)
        skip_connections.append(skip)

    # Bottleneck
    x = Conv1D(512, 15, activation='relu', padding='same')(x)

    # Upsampling blocks
    for filters, skip in zip([256, 128, 64], reversed(skip_connections)):
        x = upsampling_block(x, skip, filters, 5, 2)

    outputs = Conv1D(3, 1, activation='linear', padding='same')(x)

    model = Model(inputs, outputs)
    return model
```

ภาพที่ 3.4.3.1 แสดงฟังก์ชั่นการสร้างโมเดล

```python
def preprocess_audio(audio_path, duration, sr):
    audio, _ = librosa.load(audio_path, sr=sr, duration=duration)
    if len(audio) < duration * sr:
        # If the audio is shorter than the expected duration, pad it with zeros
        audio = np.pad(audio, (0, duration * sr - len(audio)))
    audio = audio.reshape(-1, 1)
    return audio

# Function to predict the separated sources
def predict_separation(model, mixed_audio):
    mixed_audio = mixed_audio[np.newaxis, ...]  # Add batch dimension
    predicted_sources = model.predict(mixed_audio)
    return predicted_sources[0]  # Remove batch dimension

# Function to save the separated sources
def save_audio(output_dir, sources, sr):
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    instrument_names = ['fiddle', 'flute', 'xylophone']
    for i, instrument in enumerate(instrument_names):
        output_path = os.path.join(output_dir, f'{instrument}.mp3')
        sf.write(output_path, sources[:, i], sr)

# Function to separate audio using the trained model
def separate_audio(model, input_audio_path, output_dir, duration, sr):
    # Load and preprocess the mixed audio
    mixed_audio = preprocess_audio(input_audio_path, duration, sr)

    # Predict the separated sources
    separated_sources = predict_separation(model, mixed_audio)

    # Save the separated sources
    save_audio(output_dir, separated_sources, sr)
```

ภาพที่ 3.4.3.2 แสดงฟังก์ชั่นการแยกเสียงโมเดล

3.4.4 การทดสอบประสิทธิภาพด้วย SDR

ในขั้นตอนนี้คณะผู้จัดทำได้ใช้ mir_eval ในการเรียกใช้ฟังก์ชั่นในการประเมินผลโดยมีผล sdr เป็นผลลัพธ์ของฟังก์ชัน

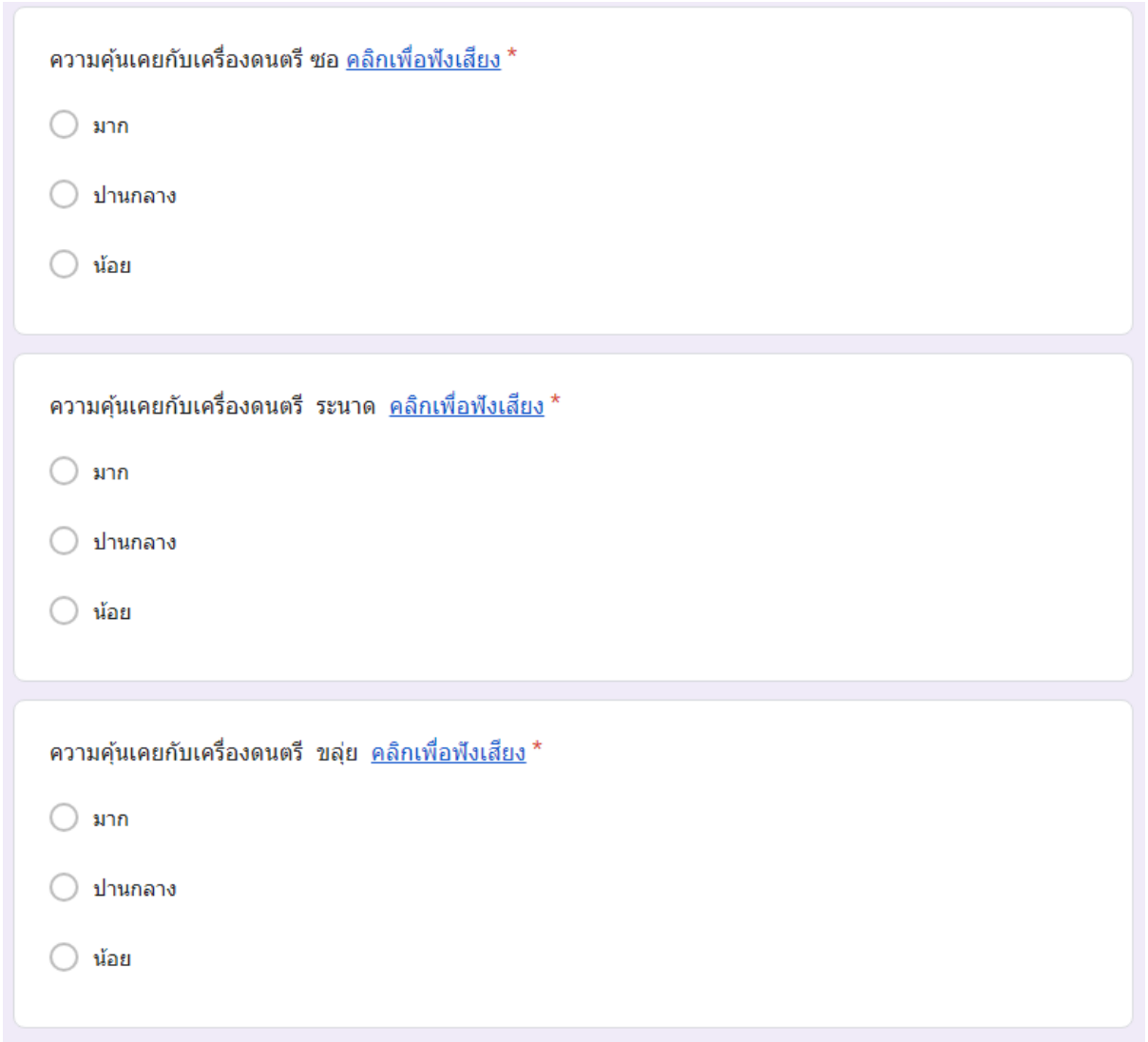
```python
# Function to load audio file
def load_audio(file_path):
    data, rate = librosa.load(file_path, sr=None) # Use librosa to load the audio file
    return rate, data

# Function to compute SDR
def compute_sdr(reference_sources, estimated_sources):
    sdr, _, _, _ = mir_eval.separation.bss_eval_sources(reference_sources, estimated_sources)
    return sdr

# Load your mixed audio and separated audio files
mixed_audio_path = '/content/DSTHAI/content/DSTHAI/tack1/mixed_with_noise.mp3'
source1_path = '/content/DSTHAI/content/DSTHAI/tack1/fiddle.mp3'
source2_path = '/content/DSTHAI/content/DSTHAI/tack1/flute.mp3'
source3_path = '/content/DSTHAI/content/DSTHAI/tack1/xylophone.mp3'

# Assuming the model's separated outputs are stored in these files
estimated_source1_path = '/content/separated_audio/fiddle.mp3'
estimated_source2_path = '/content/separated_audio/flute.mp3'
estimated_source3_path = '/content/separated_audio/xylophone.mp3'

# Load audio files
_, mixed_audio = load_audio(mixed_audio_path)
_, true_source1 = load_audio(source1_path)
_, true_source2 = load_audio(source2_path)
_, true_source3 = load_audio(source3_path)
_, estimated_source1 = load_audio(estimated_source1_path)
_, estimated_source2 = load_audio(estimated_source2_path)
_, estimated_source3 = load_audio(estimated_source3_path)

# Combine true and estimated sources into arrays
reference_sources = np.array([true_source1, true_source2, true_source3])
estimated_sources = np.array([estimated_source1, estimated_source2, estimated_source3])

# Compute SDR
sdr_values = compute_sdr(reference_sources, estimated_sources)
```

ภาพที่ 3.4.4.1 แสดงฟังก์ชั่นการประเมินด้วยค่า SDR

3.4.5 การทดสอบประสิทธิภาพด้วยการประเมินแบบ Subjective

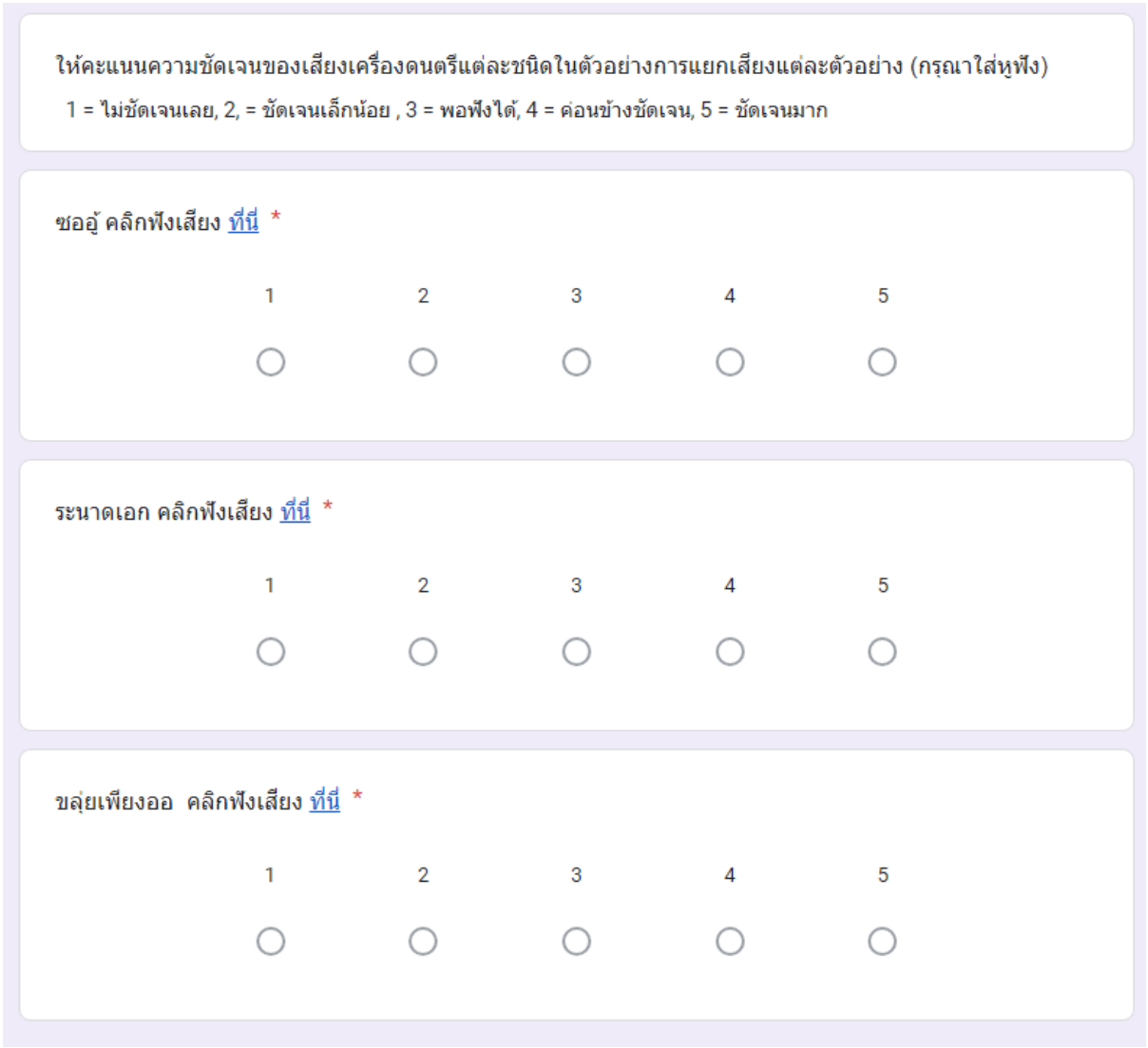
ในขั้นตอนนี้คณะผู้จัดทำได้สร้างแบบประเมินด้วย google form โดยให้ผู้เข้าทำเลือกระดับความชัดเจนของเสียงที่ผ่านการแยกด้วยโมเดลที่ผ่านการฝึก ซึ่งแบกออกเป็น 3 ส่วนได้แก่

3.4.5.1 ส่วนข้อมูลผู้ทำการประเมินและความคุ้นเคยในเครื่องดนตรีไทย



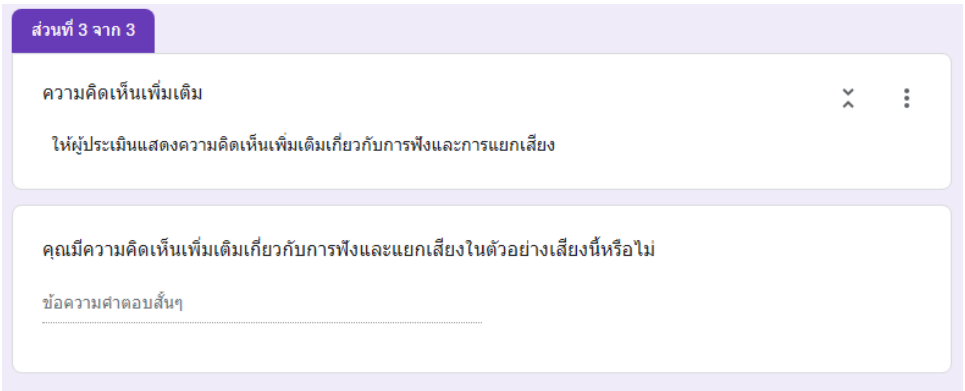
ภาพที่ 3.4.5.1.1 แสดงแบบฟอร์มการประเมินการแยกเสียงเครื่องดนตรีไทยเดิมส่วนความคุ้นเคย

3.4.5.2 ส่วนการฟังและการระบุเครื่องดนตรีที่ผ่านการแยกเสียง



ภาพที่ 3.4.5.2.1 แสดงแบบฟอร์มการประเมินการแยกเสียงเครื่องดนตรีไทยเดิมส่วนความชัดเจนในการแยกเสียง

3.4.5.3 ส่วนการแสดงความคิดเห็น



ภาพที่ 3.4.5.3.1 แสดงแบบฟอร์มการประเมินการแยกเสียงเครื่องดนตรีไทยเดิมส่วนการแสดงความคิดเห็น

บทที่ 4
ผลการวิจัย

ในการจัดทำงานวิจัยครั้งนี้คณะผู้จัดทำได้มีคความมุ่งมั่นที่จะเพิ่มความรู้สร้างและพัฒนาโมเดลการแยกเสียงเครื่องดนตรีไทยโดยในการสร้างและพัฒนาโมเดลในครั้งนี้เปรียเทียบประสิทธิภาพของการทดลองในรูปแบบต่างๆ ดังรายละเอียดผลวิจัยต่อไปนี้

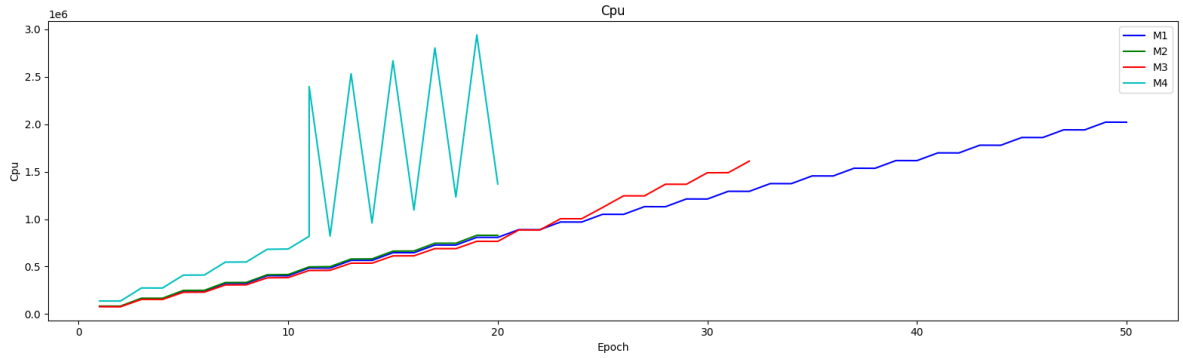
4.1 การทดสอบและการตรวจสอบการวิจัย

การทดสอบประสิทธิภาพของโมเดลการแยกเสียงเครื่องดนตรีไทยคณะผู้จัดทำได้ดำเนินการทดสอบกับเสียงเพลงเครื่องดนตรี 3 ชนิด ได้แก่ ขลุ่ยเพียงออ ระนาดเอก และ ซออู้ จากนั้นนำเสียงทั้ง 3 เสียงมารวมกันเพื่อให้โมเดลทำการแยกเสียงเครื่องดนตรีไทย โดยได้สร้างโมเดลไว้ทั้งหมด 5 โมเดล กำหนดให้ชื่อเป็น M1 ถึง M4 ตามลำดับ โมเดลทั้งหมดที่สร้างถูกปรับค่า และเลเยอร์บางส่วนเพื่อทดสอบประสิทธิภาพระหว่างโมเดล ซึ่งได้ผลการทดสอบโมเดลสรุปได้ดังนี้

4.1.1 รายละเอียดการปรับค่าของแต่ละโมเดล

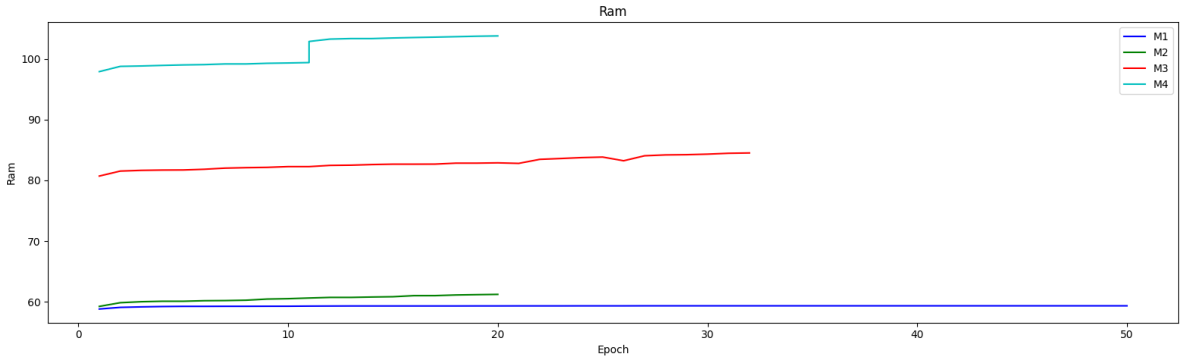
1. M1 ปรับให้ค่า learning rate เป็น 0.001 และ ทำการลด learning rate หลังจากผ่านไป 10 epoch เป็น learning rate คูณด้วย e กำลัง -0.1 และ ปรับขนาด batch เป็น 8 ใช้ dataset เป็นแบบไม่มี augmentation
2. M2 ปรับให้ทำการลด learning rate หลังจากผ่านไป 10 epoch เป็น learning rate คูณด้วย e กำลัง -0.01 และ ปรับขนาด batch เป็น 10 ใช้ dataset เป็นแบบเฉพาะที่ทำ augmentation
3. M3 ในช่วง 1 - 20 epoch ปรับให้ค่า learning rate เป็น 0.005 โมเดล เพิ่มการ ทำ he_normal ลงใน Conv1DTranspose ในช่วง 10 ถึง 20 epoch learning rate คูณด้วย e กำลัง -0.01 ใช้ dataset เป็นแบบเฉพาะที่ทำ augmentation และ ปรับขนาด batch เป็น 12 จาก นั้นในช่วง 20 - 30 epoch ได้ทำการเพิ่มขนาด dataset แล้ว ทำการลด epoch ลงเมื่อค่า val_loss ไม่มีการพัฒนาเป็น จำนวน 5 epoch เป็นครึ่งหนึ่งของค่าปัจจุบัน ใช้ dataset เป็นแบบทั้งที่ทำ และ ไม่ทำ augmentation
4. M4 ปรับให้ค่า learning rate เป็น 0.001, เพิ่มเลเยอร์ dropout (0.3) ทำการลด epoch ลง เมื่อค่า val_loss ไม่มีการพัฒนา เป็นจำนวน 5 epoch เป็นครึ่งหนึ่งของค่าปัจจุบัน ใช้ dataset เป็นแบบเฉพาะที่ทำ augmentation

4.1.2 เปรียบเทียบการใช้ทรัพยากรในแต่ละโมเดล



กราฟที่ 4.1.2.1 การเปรียบเทียบการใช้ CPU ของแต่ละโมเดล

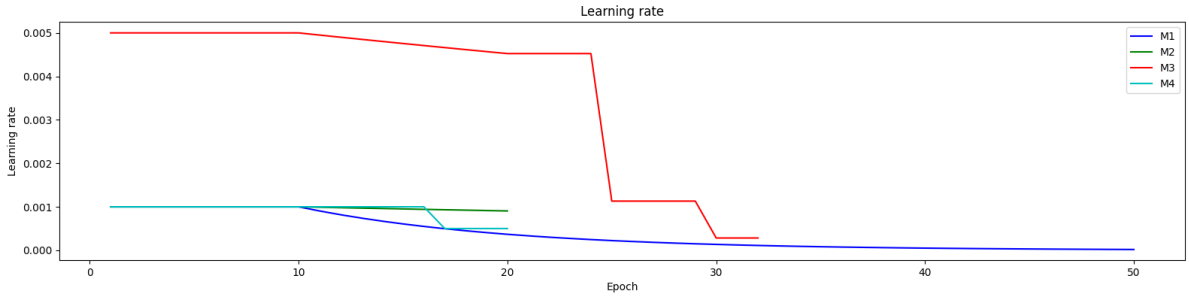
จากกราฟที่ 4.1.2.1 สามารถสรุปได้ว่าการใช้งาน CPU ของโมเดล M4 มีการใช้งานที่เยอะที่สุด ถ้าเปรียบเทียบกับโมเดลอื่นๆใน Epoch เดียวกัน รองลงมาคือ M2 ในขณะที่ M1 และ M3 มีการใช้งาน CPU ที่น้อยกว่า



กราฟที่ 4.1.2.2 การเปรียบเทียบการใช้ RAM ของแต่ละโมเดล

จากกราฟที่ 4.1.2.2 สามารถสรุปได้ว่า การใช้งาน RAM ของโมเดล M4 มีการใช้งานที่เยอะที่สุดถ้าเปรียบเทียบกับโมเดลอื่นๆใน Epoch เดียวกัน รองลงมาคือ M3 ในขณะที่ M2 และ M4 มีการใช้งาน RAM ที่น้อยกว่า

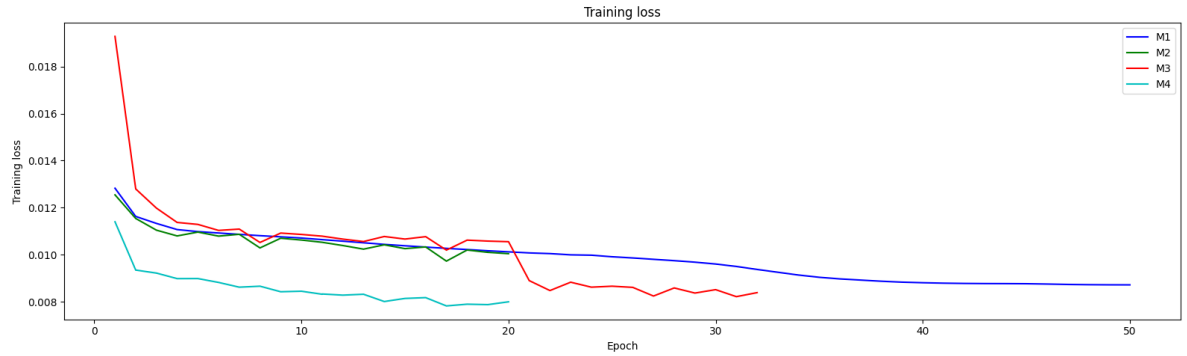
4.1.3 เปรียบเทียบการลดของ learning rate ในแต่ละโมเดล



กราฟที่ 4.1.3.1 การเปรียบเทียบค่า learning rate ของแต่ละโมเดล

จากกราฟที่ 4.1.3.1 สามารถสรุปได้ว่า การลดของ learning rate ของโมเดล M4 เมื่อมีการลดลงถึงจุดหนึ่ง learning rate จะไม่มีลดที่รวดเร็วมากนักหรือไม่มีการเปลี่ยนแปลงค่า learning rate ทำให้ช่วงท้ายของ M4 มีลักษณะที่อยู่ในแนวระนาบเมื่อเปรียบเทียบกับโมเดลอื่นทและไม่มีการคงตัวของค่า learning rate

4.1.4 เปรียบเทียบผล Training loss ของแต่ละโมเดล

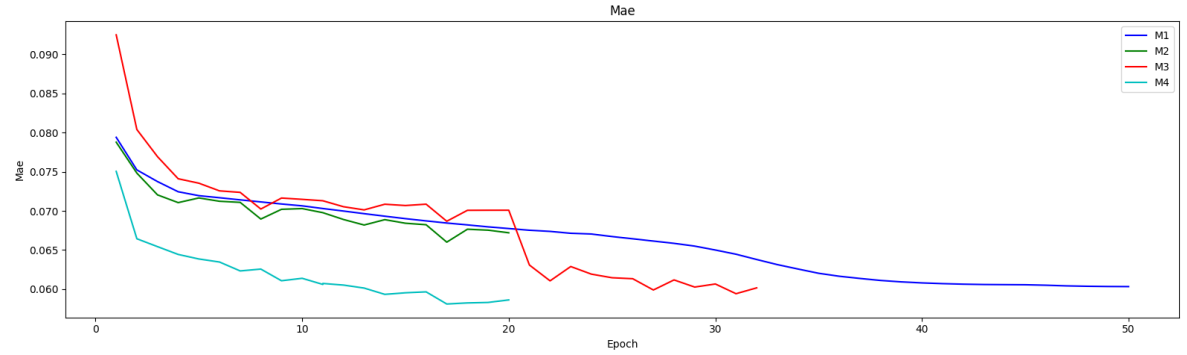


กราฟที่ 4.1.4.1 การเปรียบเทียบค่า Training loss *ของแต่ละโมเดล*

จากกราฟที่ 4.1.4.1 สามารถสรุปได้ว่าการค่า โมเดลกำลังเรียนรู้ได้ดีในช่วงแรก เมื่อถึงรอบประมาณ 10-15 การลดลงของการสูญเสียจะเริ่มช้าลงและมีความคงที่มากขึ้น โมเดล M4 มีประสิทธิภาพมาก ขณะที่ M3 ก็มีความสามารถที่ดีขึ้นหลังจากผ่านช่วง 20 epoch ไปซึ่งสันณิฐานว่ามีสาเหตุมาจากการปรับ learning rate

4.1.5 เปรียบเทียบผล Mean absolute error (MAE) ของแต่ละโมเดล

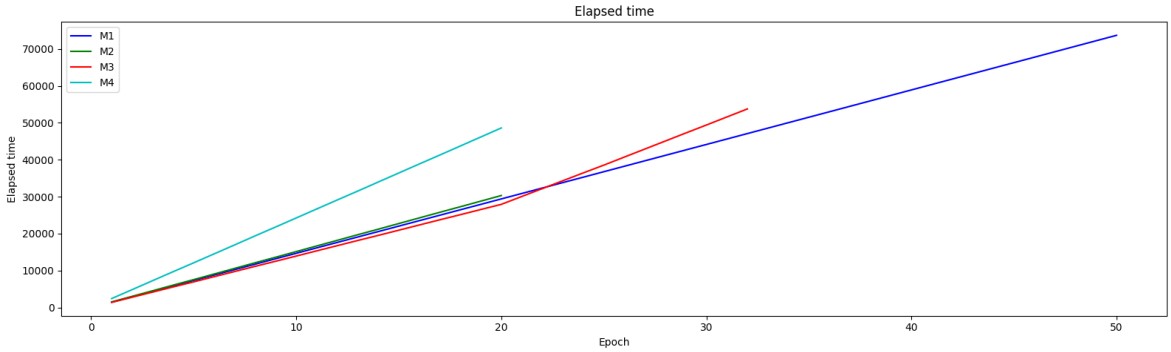
ในขั้นตอนนี้คณะผู้วิจัยได้เลือกเปรียบเทียบระหว่างโมเดล M3 กับ M4 ได้ผลดังน



กราฟที่ 4.1.5.1 การเปรียบเทียบค่า MAE ของแต่ละโมเดล

จากกราฟที่ 4.1.5.1 สามารถสรุปได้ว่าการค่า MAE ของโมเดล M4 มีแนวโน้มลดลงอย่างต่อเนื่อง ในขณะที่ค่า MAE ของ M3 มีการลดลงในช่วงแรกแต่คงที่ในช่วงท้าย และ ค่า MAE ของโมเดล M4 ยังมีค่าที่น้อยกว่า โมเดล M3 ซึ่งแสดงให้เห็นว่าโมเดลนี้มีประสิทธิภาพในการลดข้อผิดพลาดได้ดีกว่า

4.1.6 เปรียบเทียบระยะเวลาที่ใช้ในการฝึกของแต่ละโมเดล

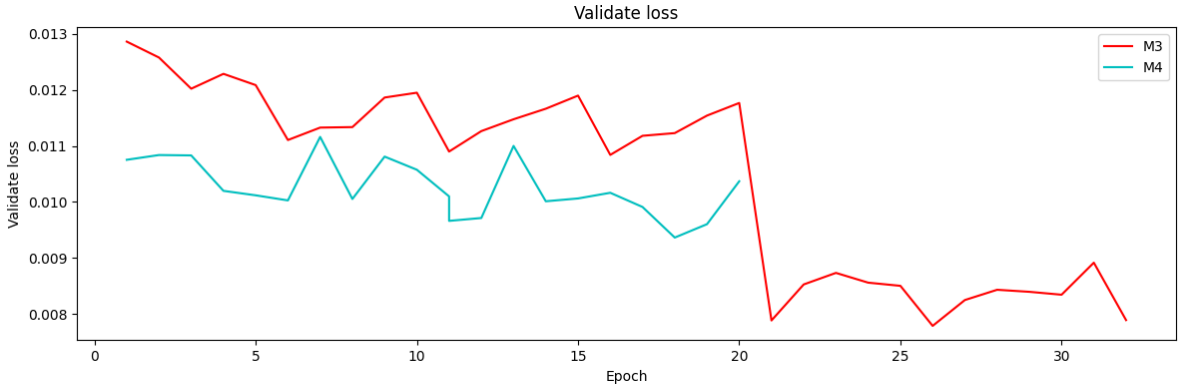


กราฟที่ 4.1.6.1 การเปรียบเทียบ เวลา ของแต่ละโมเดล

จากกราฟที่ 4.1.6.1 สามารถสรุปได้ว่า โมเดลที่ใช้ระยะเวลามากสุดคือ โมเดล M4 สันนิฐานว่าสาเหตุเกิดจากความซับซ้อนที่เพิ่มขึ้นในตัวของโมเดล ในขณะที่โมเดลที่เหลือมีระยะเวลาที่ใช้ใกล้เคียงกัน

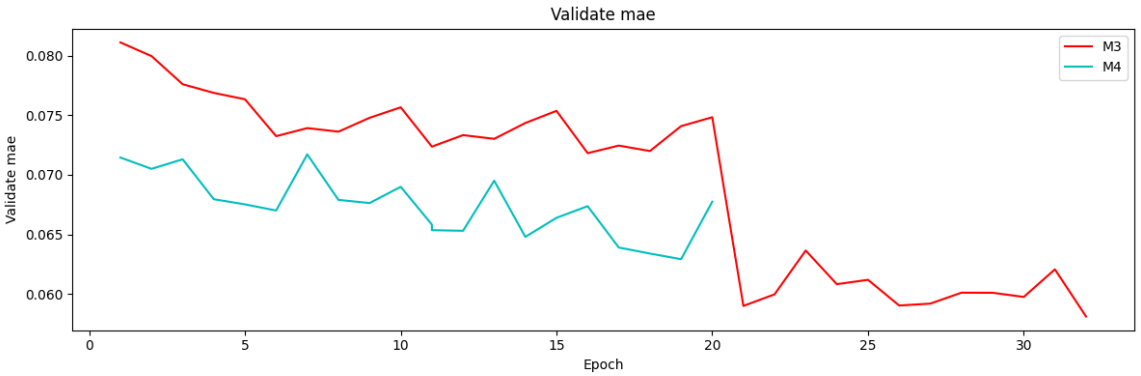
4.1.7 เปรียบเทียบผล Validate ของ Loss และ MAE ของแต่ละโมเดล

ในขั้นตอนนี้คณะผู้วิจัยได้เลือกเปรียบเทียบระหว่างโมเดล M3 กับ M4 ได้ผลดังนี้



กราฟที่ 4.1.7.1 การเปรียบเทียบค่า Validate Loss ของแต่ละโมเดล

จากกราฟที่ 4.1.7.1 สามารถสรุปได้ว่าการค่า Validate Loss ของโมเดล M3 ดูเหมือนจะมีประสิทธิภาพดีกว่าเมื่อเปรียบเทียบกับโมเดล M4 โดยที่ค่า Validation Loss ของ M3 มีค่าต่ำกว่าซึ่งบ่งบอกว่าอาจสามารถใช้งานกับข้อมูลที่ไม่เคยเห็นมาก่อนได้ดีกว่า M4



กราฟที่ 4.1.7.2 การเปรียบเทียบค่า Validate MAE ของแต่ละโมเดล

จากกราฟที่ 4.1.7.2 สามารถสรุปได้ว่าค่า Validate MAE ของโมเดล M3 ลดลงจาก 0.08 ในช่วงแรกของการฝึกอบรมและมีแนวโน้มลดลงอย่างต่อเนื่อง โดยเฉพาะในช่วงท้ายที่ลดลงอย่างมากค่า Validate MAE ของโมเดล M4 มีความผันผวนมากกว่าค่า Validate MAE ของ M3 แต่ยังคงมีค่าต่ำกว่า M3 ตลอดการฝึกอบรม แต่เมื่อผ่านช่วง 20 epoch เป็นต้นไปค่าของ M3 กลับลดต่ำกว่า M4 ได้แสดงให้เห็นว่าประสิทธิภาพของ M4 นั้นดีกว่า M3 ในช่วงต้นการฝึก แต่ในช่วงปลาย M3 นั้นมีประสิทธิภาพมากกว่า

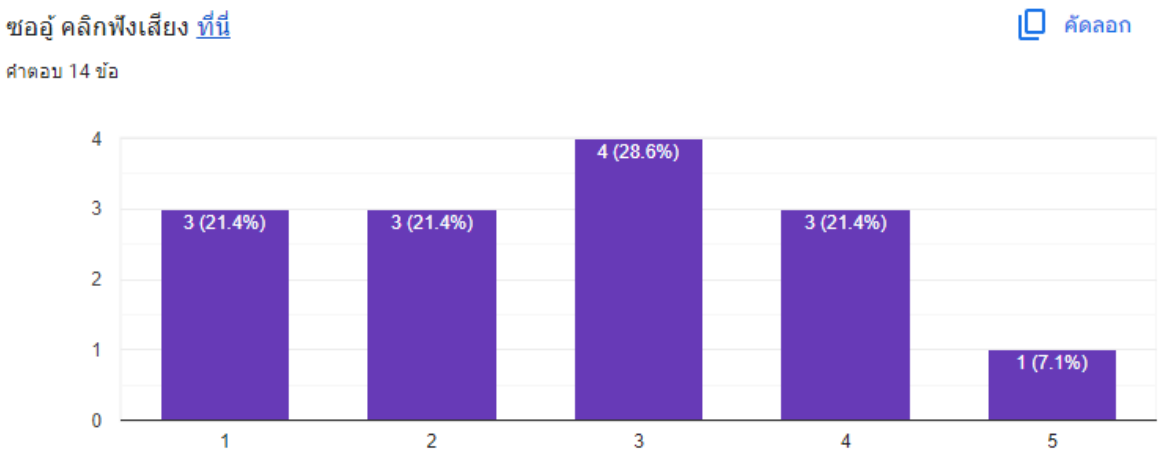
4.2 สรุปผลการทดสอบ

จากการทดสอบโมเดลการแยกเสียงเครื่องดนตรีไทยเดิมทำให้ สามารถสรุปผลได้ ดังนี้

4.2.1 การทดสอบโดยให้ผู้ทดสอบทำการฟังเสียงที่ผ่านการแยกเสียงโดยโมเดล

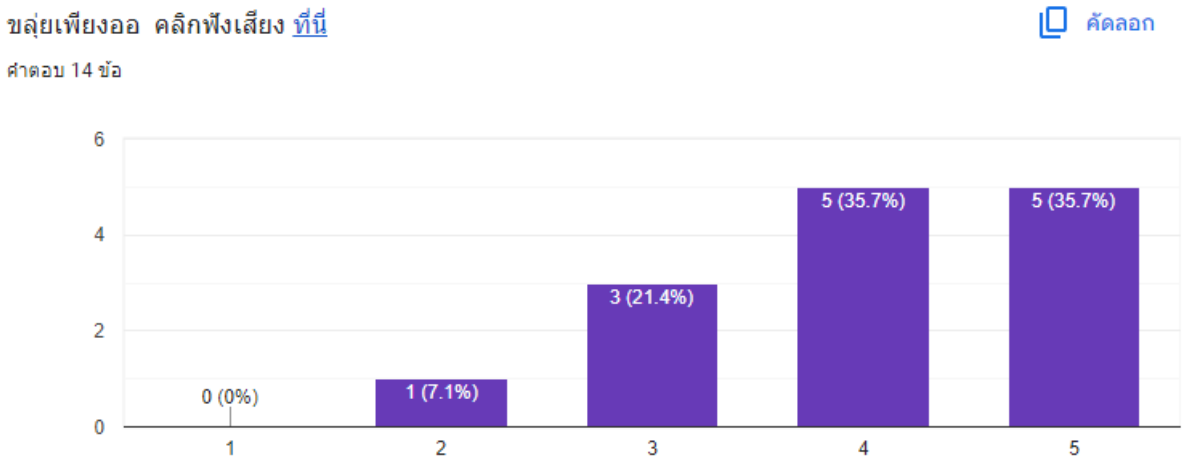
ในขั้นตอนนี้มีผู้เข้าร่วมประเมินจำนวนทั้งหมด 14 คน โดยทั้งหมดได้เลือกระดับความคมชัดของเสียได้แก่ 1 = ไม่ชัดเจนเลย, 2, = ชัดเจนเล็กน้อย , 3 = พอฟังได้, 4 = ค่อนข้างชัดเจน และ 5 = ชัดเจนมาก สรุปเป็นผลได้ดังนี้

1.ผลการทดสอบการแยกเสียง ซอ



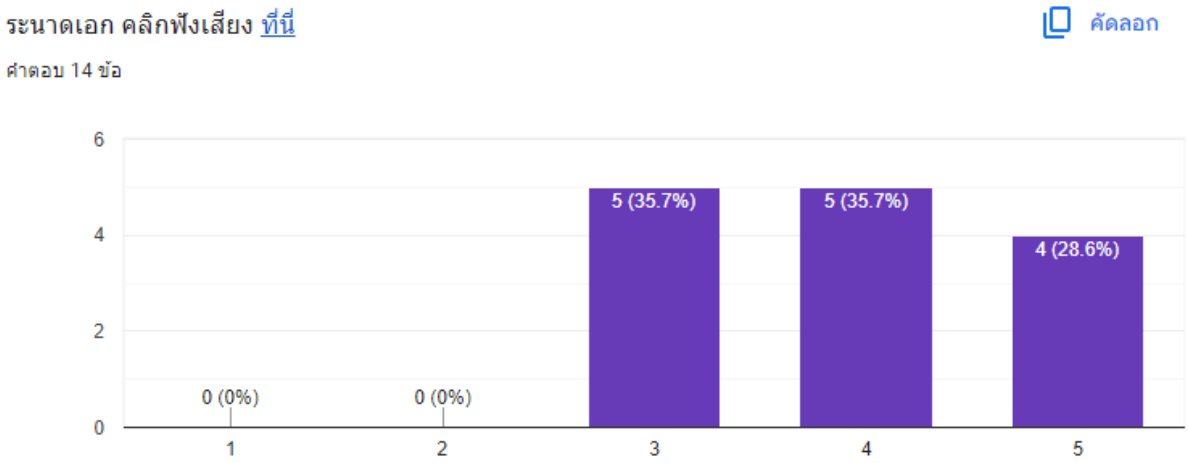
กราฟที่ 4.2.1.1 กราฟแท่งแสดงความคมชัดของเสียงซออู้

2.ผลการทดสอบการแยกเสียงขลุ่ย



กราฟที่ 4.2.1.2 กราฟแท่งแสดงความคมชัดของเสียงขลุ่ย

3. ผลการทดสอบการแยกเสียง ระนาด



กราฟที่ 4.2.1.3 กราฟแท่งแสดงความคมชัดของเสียงระนาดเอก

4.2.2 การทดสอบด้วย signal to distortion ratio (SDR)

ในขั้นตอนนี้คณะผู้จัดทำได้ทำการประสิทธิภาพของโมเดลทั้ง 4 โดยกำหนดให้โมเดลทั้ง 4 มีช่วงการฝึกอยู่ที่ 20 epoch เพื่อให้ค่ามีความเท่าเทียม

1. การทดสอบ SDR โดยใช้โมเดล M1

1.ซอ: 11.24 dB

2.ขลุ่ย: 13.22 dB

3.ระนาด: 3.37 dB

ค่าเฉลี่ย: 9.27 dB

2. การทดสอบ SDR โดยใช้โมเดล M2

1.ซอ: 11.56 dB

2.ขลุ่ย: 12.63 dB

3.ระนาด: 4.80 dB

ค่าเฉลี่ย: 9.66 dB

3. การทดสอบ SDR โดยใช้โมเดล M3

1.ซอ: 11.20 dB

2.ขลุ่ย: 12.46 dB

3.ระนาด: 3.11 dB

ค่าเฉลี่ย: 8.92 dB

4. การทดสอบ SDR โดยใช้โมเดล M4

1.ซอ: 11.91 dB

2.ขลุ่ย: 13.78 dB

3.ระนาด: 5.03 dB

ค่าเฉลี่ย: 10.24 dB

จากค่า SDR โดยเฉลี่ยโมเดลที่มีประสิทธิภาพมากที่สุดคือโมเดล M4 และโมเดลที่มีประสิทธิภาพน้อยที่สุด คือ โมเดล M3 ซึ่งคณะผู้วิจัยสันณิฐานว่าเกิดจากความซับซ้อนของโมเดลไม่สูงพอ อ้างอิงจากการที่โมเดล M4 มีการเพิ่มเลเยอร์ dropout เข้าไปทำให้โมเดลมีความซับซ้อนมากขึ้น

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

การสรุปผลของการพัฒนาเทคนิคการแยกเสียงเครื่องดนตรีไทย The Sound Demixing for thai traditional music มีรายละเอียดดังต่อไปนี้

5.1 สรุปผลการวิจัย

จากผลการดำเนินงานวิจัยในบทที่ 4 สามารถสรุปผลการดำเนินงานวิจัยได้ดังต่อไปนี้

5.1.1 ทรัพยากรที่ใช้ในการฝึกโมเดล

จากการทดลองพบว่าเมื่อเพิ่มเลเยอร์ dropout ที่ใช้เพื่อลดโอกาสการเกิดสภาวะ overfit ลงในเลเยอร์ที่ 2 ถึง 8 ตามที่อธิบายไว้ใน 3.3.1.4 จะทำให้ระยะเวลาการฝึกโมเดลเพิ่มขึ้นจาก 5555 ต่อ epoch เป็น 555 รวมไปถึงการใช้งาน CPU ที่เพิ่มขึ้นจาก เป็น เมื่อเปรียบเทียบโมเดล M3 กับ M4 ในขณะที่ค่า Loss และ MAE ในแต่ละ epoch ของทั้ง 2 โมเดลมีความต่างกันค่อนข้างมาก เมื่อเปรียบเทียบกับโมเดล M1 และ M2 จึงสามารถสรุปได้ว่า การเพิ่มเลเยอร์ dropout มีส่วนให้ประสิทธิภาพของโมเดลดีขึ้นจริงแต่ก็ต้องแลกมากับการใช้งาน cpu และ เวลา ที่เพิ่มขึ้นอย่างมากเช่นกัน

5.1.2 ความสามารถในการแยกเสียงเครื่องดนตรี

จากการทดลองพบว่าค่า SDR ที่ได้จากการแยกเสียงระนาดนั้นมีค่าน้อยกว่าเครื่องดนตรีอื่นกลับกันในแบบประเมินกลับระบุให้เสียงระนาดเป็นเสียงที่ได้ยินค่อนข้างมากซึ่งคณะผู้วิจัยสันณิฐานว่าปัญหานี้เกิดจากความเป็นเอกลักษณ์ของเสียงระนาดทำให้แม้ค่า SDR จะมีค่าน้อยแต่ผู้ประเมินกลับสามารถได้ยินเสียงของระนาดได้ชัดเจน

5.2 การอภิปราย

จากการศึกษาค้นคว้าคณะวิจัยพบว่าในการแยกเสียงเครื่องดนตรีไทย ผู้วิจัยควรมี Dataset ที่เหมาะสมเช่น ความดัง ความชัดเจน คุณภาพ ความยาวของเสียงและ Dataset ที่ถูกทำขึ้นเพื่อการใช้งานในการ TrainModel ที่มีการแบ่งเสียงเครื่องดนตรีออกเป็นแต่ละชนิดและเสียงที่นำเสียงเครื่องดนตรีแต่ละชนิดมารวมกันสิ่งเหล่านี้ส่งผลถึงคุณภาพของโมเดลทำให้สามารถนำไปใช้งานได้ดียิ่งขึ้นและผู้วิจัยสามารถเลือกโมเดลอื่นๆตามที่ผู้วิจัยสนใจและปรับแต่งโมเดลก่อนทำการ TrainModel ให้เหมาะสมกับฮาร์ดแวร์ของผู้วิจัยซึ่งจะช่วยลดเวลาและการใช้ทรัพยากรให้ลดน้อยลงได้แต่ผลลัพธ์ที่ได้ยังมีประสิทธิภาพ

5.3 วิเคราะห์ปัญหาจากการดำเนินการวิจัย

5.3.1 การขาดชุดข้อมูลเฉพาะที่ใช้ในการฝึกฝนโมเดลการแยกเสียงเครื่องดนตรีไทย ซึ่งเป็นปัจจัยสำคัญในการสร้างโมเดลที่มีประสิทธิภาพ

5.3.2 ปัญหาระยะเวลาที่ใช้ในการฝึกโมเดล ซึ่งเกิดมาจากขนาดของชุดข้อมูลที่ใหญ่ทำให้ในการฝึกโมเดลแต่ละ epoch อยู่ที่ 12 ถึง 30 นาที

5.3.3 นำไปใช้งานจริงได้ยากเนื่องจากปัจจัยภายนอกอื่นๆ เช่น ความชัดของเสียง เสียงรบกวนที่เกิดขึ้น และ ความแตกต่างด้านเสียงที่แยกย่อยไปในแต่ละลักษณะของเครื่องดนตรีไทย

5.4 แนวทางและข้อเสนอแนะในการพัฒนาต่อในอนาคต

เพื่อแก้ปัญหาที่กล่าวถึงการพัฒนาเทคนิคการแยกเสียงเครื่องดนตรีไทย The Sound Demixing for thai traditional music ดังกล่าว

คณะผู้วิจัยได้มีข้อเสนอแนะและแนวทางที่อาจช่วยให้ประสิทธิภาพของงานวิจัยนี้เพิ่มขึ้น โดยมีดังต่อไปนี้

5.3.1 เนื่องจากการขาดแคลนชุดข้อมูลเฉพาะอาจสามารถใช้โปรแกรมการจำลองเสียงเครื่องดนตรีไทยในการสร้างชุดข้อมูลที่เฉพาะ เช่น เพลงไทยเดิมได้

5.3.2 ทดลองปรับค่า Parameter ของ Wave-U-Net โมเดลเพื่อผลการวิจัยที่ดีขึ้น

5.3.3 เปลี่ยนชนิดของเครื่องดนตรีไทย เพื่อขยายแนวทางการวิจัย

5.3.4 เปลี่ยนสถาปัตยกรรมของโมเดลการแยกเสียงไปเป็นแบบอื่น เช่น Band-split RNN (BSRNN)

5.3.5 เปลี่ยนการฝึกโมเดลเป็นแบ 1 เสียงหลายเครื่องดนตรีเป็น 1 เสียง 1 เครื่องดนตรี

บรรณานุกรม

[1] จารวี ฉันทสิทธิพร. **การจำแนกชนิดยาเม็ดจากภาพถ่าย โดยใช้เทคนิคเครือข่ายใยประสาท**. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร. 2548.

[2] อานนท์ นามสนิท. **การจำแนกกลุ่มเพลงไทยโดยใช้ซัพพอร์ตเวคเตอร์แมชชีนส์**. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาเทคโนโลยีคอมพิวเตอร์ ภาควิชาคอมพิวเตอร์ศึกษา บัณฑิตวิทยาลัย สถาบัน เทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2549

[3] โอภาส แก้วต่าย, **การจำแนกกลุ่มเพลงไทยเดิม**, *วิทยานิพนธ์ปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ ภาควิชาคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยศิลปากร*, 2552

[4] P. Chandna, M. Miron, J. Janer, and E. Gómez, **Monoaural Audio Source Separation Using Deep Convolutional Neural Networks**, in Lecture notes in computer science, 2017, doi: 10.1007/978-3-319-53547-0_25.

[5] G. Fabbro et al., **The Sound Demixing Challenge 2023 – Music Demixing Track**, arXiv (Cornell University), 2023, doi: 10.48550/arxiv.2308.06979.

[6] A. Favaro, A. Lewis, and G. Schlesinger, **ICA for Musical Signal Separation**, Stanford CS229: Machine Learning, 2017.

[7] M. Kim, W. Choi, J. Chung, D. Lee, and S. Jung, **KUIELab-MDX-Net: A Two-Stream Neural Network for Music Demixing**, arXiv (Cornell University), 2021,

doi: 10.48550/arxiv.2111.12203.

[8] Y. Luo and J. Yu, **Music Source Separation with Band-split RNN**, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2023, doi: 10.1109/taslp.2023.3271145.

[9] D. Stoller, S. Ewert, and S. Dixon, **Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation**, arXiv *(Cornell University),* 2018, doi: 10.48550/arxiv.1806.03185.

[10] S. Uhlich et al., **Improving music source separation based on deep neural networks through data augmentation and network blending**, 2017, doi: 10.1109/icassp.2017.7952158.

ภาคผนวก ก
แบบทดสอบ

แบบประเมินการแยกเสียงเครื่องดนตรีไทยเดิม

แบบประเมินการแยกเสียงเครื่องดนตรีไทยเดิม 3 ชนิด ได้แก่ ซออู้ ระนาดเอก และ ขลุ่ยเพียงออ
ส่วนที่ 1: ข้อมูลเบื้องต้นของผู้ประเมิน

ชื่อ-นามสกุล *

ข้อความคำตอบสั้นๆ

อายุ *

ข้อความคำตอบสั้นๆ

เคยฟังดนตรีไทยหรือไม่ *

- ใช่
- ไม่ใช่

ความคุ้นเคยกับเครื่องดนตรี ซอ คลิกเพื่อฟังเสียง *

- มาก
- ปานกลาง
- น้อย

ความคุ้นเคยกับเครื่องดนตรี ระนาด คลิกเพื่อฟังเสียง *

- มาก
- ปานกลาง
- น้อย

ความคุ้นเคยกับเครื่องดนตรี ขลุ่ย คลิกเพื่อฟังเสียง *

- มาก
- ปานกลาง
- น้อย

การฟังและการระบุเครื่องดนตรี

ฟังตัวอย่างเสียงแต่ละตัวอย่างแล้วตอบคำถามที่เกี่ยวข้อง
ตัวอย่างเสียง : https://drive.google.com/file/d/1-6oYB0yUmyk6Rl0z9j8LvTiuSfapda7f/view?usp=sharing

คุณได้ยินเสียงเครื่องดนตรีใดบ้างในตัวอย่างนี้ (เลือกได้มากกว่า 1 คำตอบ) *

- [ ] ซออู้
- [ ] ระนาดเอก
- [ ] ขลุ่ยเพียงออ
- [ ] ไม่มีเสียงเครื่องดนตรีไทยเดิม

ให้คะแนนความชัดเจนของเสียงเครื่องดนตรีแต่ละชนิดในตัวอย่างการแยกเสียงแต่ละตัวอย่าง (กรุณาใส่หูฟัง)
1 = ไม่ชัดเจนเลย, 2, = ชัดเจนเล็กน้อย , 3 = พอฟังได้, 4 = ค่อนข้างชัดเจน, 5 = ชัดเจนมาก

ซออู้ คลิกฟังเสียง ที่นี่ *

1	2	3	4	5
○	○	○	○	○

ระนาดเอก คลิกฟังเสียง ที่นี่ *

1	2	3	4	5
○	○	○	○	○

ขลุ่ยเพียงออ คลิกฟังเสียง ที่นี่ *

1	2	3	4	5
○	○	○	○	○

ความคิดเห็นเพิ่มเติม

ให้ผู้ประเมินแสดงความคิดเห็นเพิ่มเติมเกี่ยวกับการฟังและการแยกเสียง

คุณมีความคิดเห็นเพิ่มเติมเกี่ยวกับการฟังและแยกเสียงในตัวอย่างเสียงนี้หรือไม่

ข้อความคำตอบสั้นๆ

ภาคผนวก ข
ผลการประเมิน

ชื่อ-นามสกุล

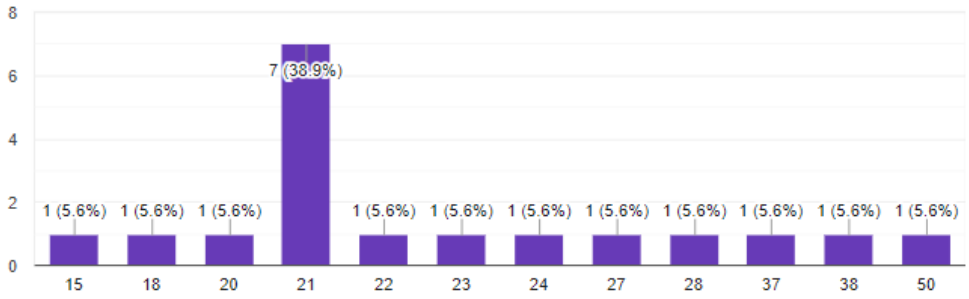
คำตอบ 18 ข้อ

- ธนกร ปรารมย์
- ณิชาริย์ ชำนาญแป้น
- สุทัตตา มังสระคู
- ร
- คนติดF
- ศุภพงศ์ หลวงพล
- อนุพงษ์ สงวนหงษ์
- ปาจรีย์ สาสกุล
- ธัญญ์นรี บุบผาชาติ

อายุ

คัดลอก

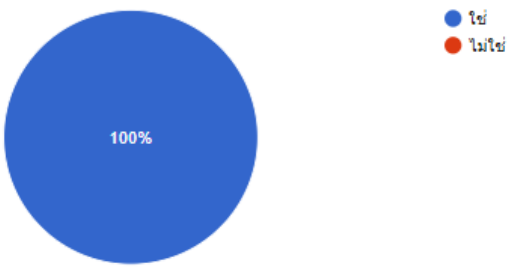
คำตอบ 18 ข้อ



เคยฟังดนตรีไทยหรือไม่

คัดลอก

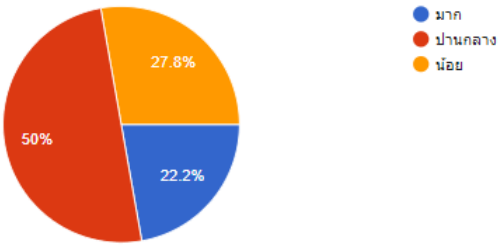
คำตอบ 18 ข้อ



ความคุ้นเคยกับเครื่องดนตรี ซอ คลิกเพื่อฟังเสียง

คัดลอก

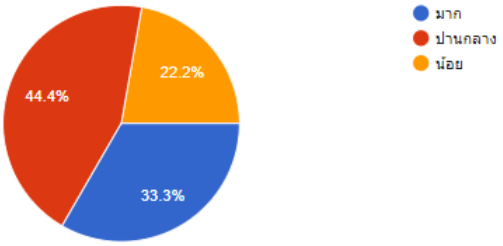
คำตอบ 18 ข้อ



ความคุ้นเคยกับเครื่องดนตรี ระนาด คลิกเพื่อฟังเสียง

คัดลอก

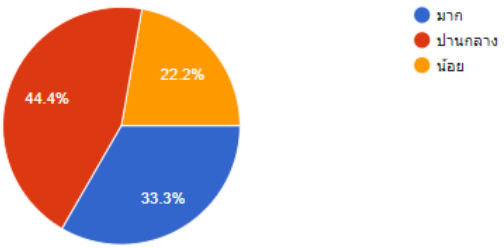
คำตอบ 18 ข้อ



ความคุ้นเคยกับเครื่องดนตรี ขลุ่ย คลิกเพื่อฟังเสียง

คัดลอก

คำตอบ 18 ข้อ

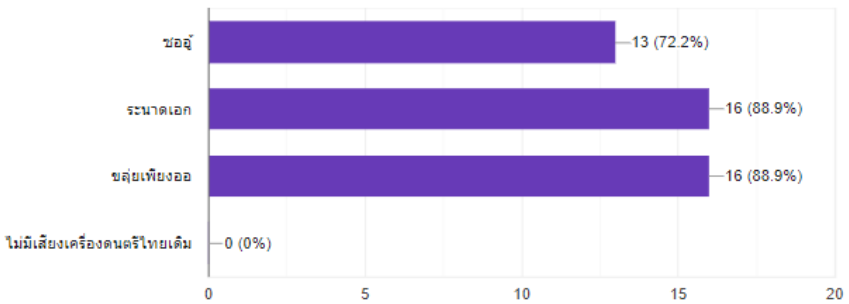


การฟังและการระบุเครื่องดนตรี

คุณได้ยินเสียงเครื่องดนตรีใดบ้างในตัวอย่างนี้ (เลือกได้มากกว่า 1 คำตอบ)

คัดลอก

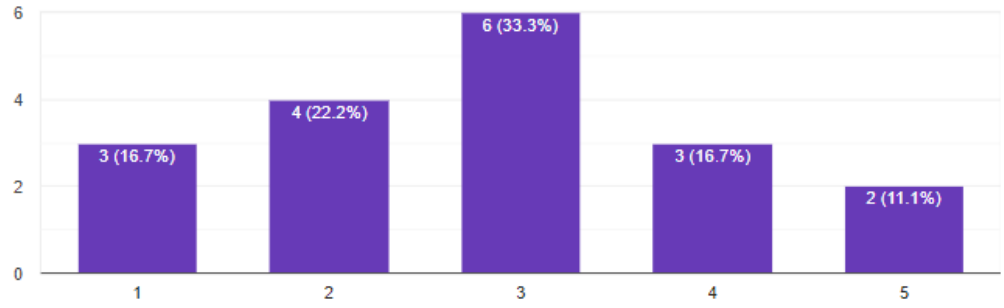
คำตอบ 18 ข้อ



ซออู้ คลิกฟังเสียง ที่นี่

คัดลอก

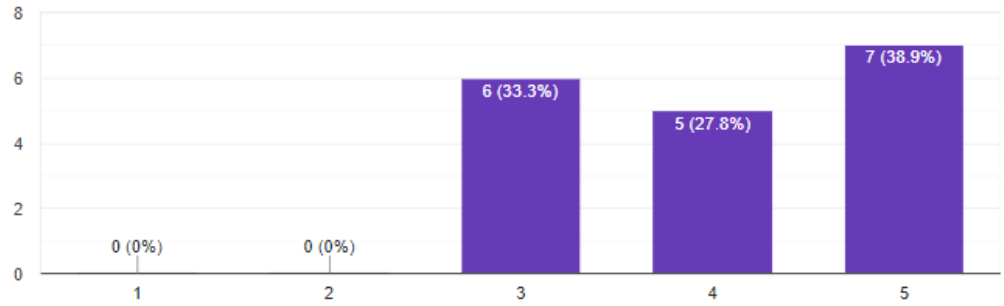
คำตอบ 18 ข้อ



ระนาดเอก คลิกฟังเสียง ที่นี่

คัดลอก

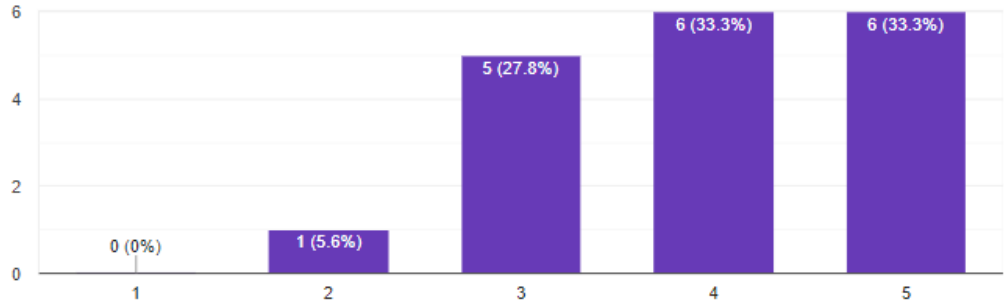
คำตอบ 18 ข้อ



ขลุ่ยเพียงออ คลิกฟังเสียง ที่นี่

คัดลอก

คำตอบ 18 ข้อ



ความคิดเห็นเพิ่มเติม

คุณมีความคิดเห็นเพิ่มเติมเกี่ยวกับการฟังและแยกเสียงในตัวอย่างเสียงนี้หรือไม่

คำตอบ 12 ข้อ

- เสียงบางเสียงยังเบา
- ยังค่อนข้างฟังยาก
- เสียงรวมซออู้มันทุ้มเบาเสียงมันเลยฟังยากแยกยากกว่าเสียงอื่น
- แยกไม่ค่อยออกว่าเสียงอะไร เนื่องไม่ชอบฟังดนตรีไทย
- มีเสียงรบกวนมาก
- ยังไม่สามารถแยกได้อย่างชัดเจนในเครื่องดนตรีบางชนิด
- -
- ไม่มีครับ
- เยี่ยมครับ