

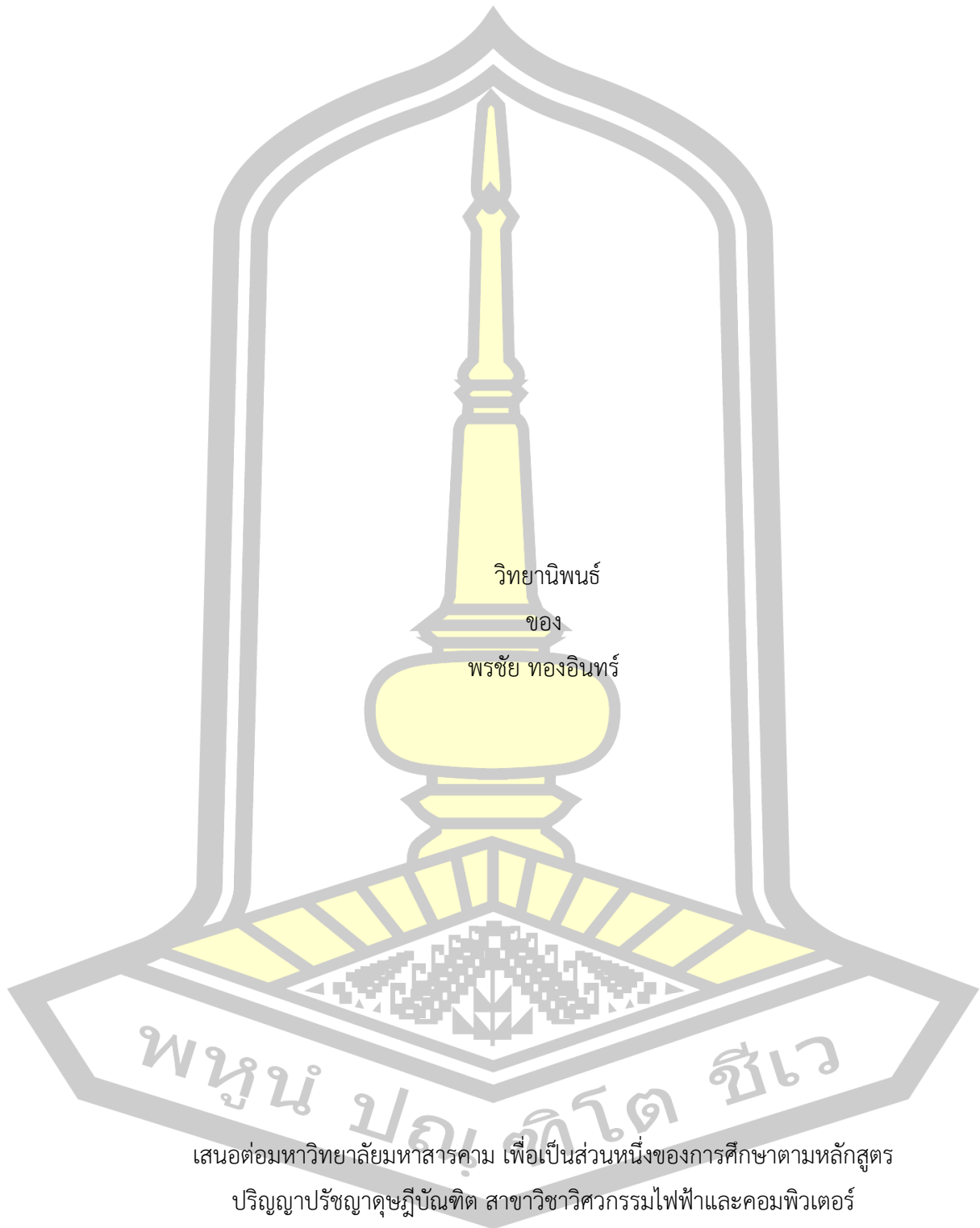
เทคนิคการตรวจจับการลัมด้วยอุปกรณ์ที่พียู

วิทยานิพนธ์
ของ
พรชัย ทองอินทร์

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
มกราคม 2567

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

เทคนิคการตรวจจับการลัดด้วยอุปกรณ์ที่พื้ญ

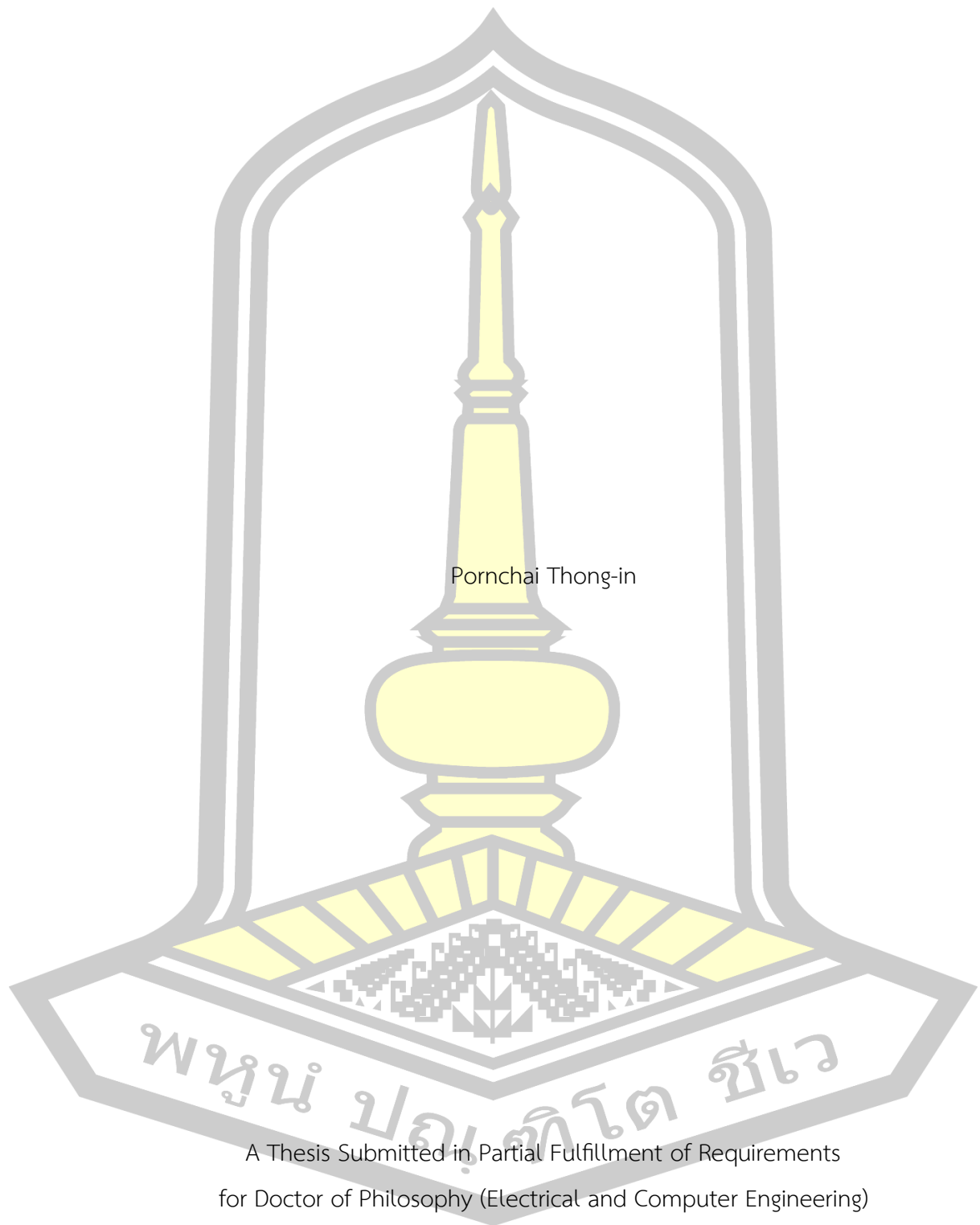


เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

มกราคม 2567

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Fall Detection Techniques Using TPU



Pornchai Thong-in

A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Electrical and Computer Engineering)

January 2024

Copyright of Mahasarakham University



คณะกรรมการสอบวิทยานิพนธ์ ได้พิจารณาวิทยานิพนธ์ของนายพรชัย ทองอินทร์ แล้ว เห็นสมควรรับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์ ของมหาวิทยาลัยมหาสารคาม

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รศ. ดร. อนันต์ เครือทรัพย์ถาวร)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผศ. ดร. ณัฐวุฒิ สุวรรณทา)

.....กรรมการ

(ศ. ดร. วรวัฒน์ เสงี่ยมวิบูล)

.....กรรมการ

(ผศ. ดร. นิวัตร อังควิเศษฐพันธ์)

.....กรรมการ

(ผศ. ดร. ชัยยงค์ เสริมผล)

.....กรรมการผู้ทรงคุณวุฒิภายนอก

(รศ. ดร. อนันต์ เครือทรัพย์ถาวร)

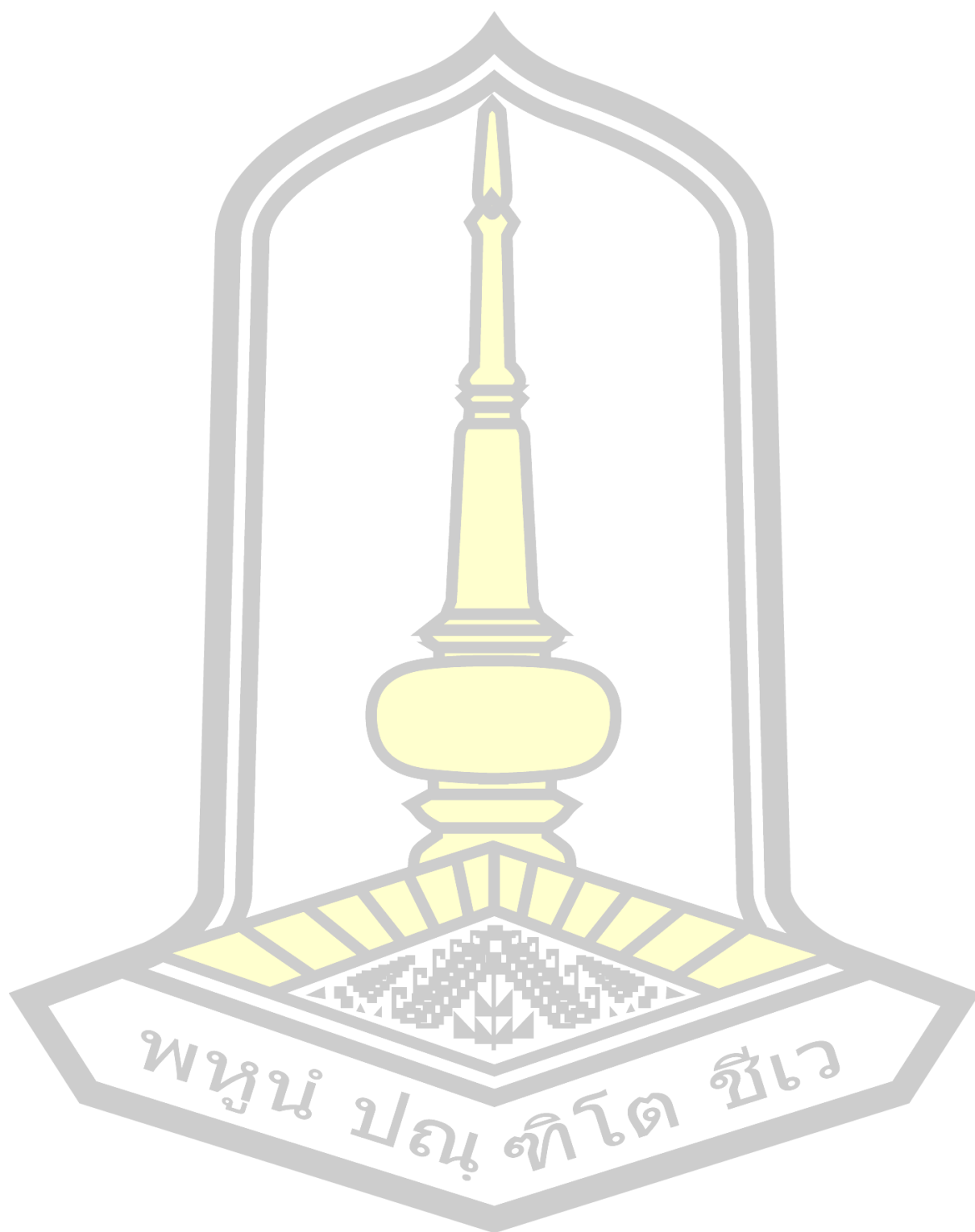
มหาวิทยาลัยขอนแก่นให้รับวิทยานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร ปริญญา ปรัชญาดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ ของมหาวิทยาลัย มหาสารคาม

.....
(รศ. ดร. เกียรติศักดิ์ ศรีประทีป)

คณบดีคณะวิศวกรรมศาสตร์

.....
(รศ. ดร. กริสน์ ชัยมูล)

คณบดีบัณฑิตวิทยาลัย



ชื่อเรื่อง เทคนิคการตรวจจับการล้มด้วยอุปกรณ์ที่พียู
 ผู้วิจัย พรชัย ทองอินทร์
 อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. ณัฐวุฒิ สุวรรณทา
 ปริญญา ปรัชญาดุษฎีบัณฑิต สาขาวิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์
 มหาวิทยาลัย มหาวิทยาลัยมหาสารคาม ปีที่พิมพ์ 2567

บทคัดย่อ

งานวิจัยฉบับนี้ได้นำเสนอการประยุกต์ใช้ Pose estimation โดยใช้รูปแบบ PoseNet เป็นตัวสร้างจุดต่อโครงกระดูกของมนุษย์จากภาพเคลื่อนไหว ซึ่งภาพจะได้มาจากกล้อง 2D RGB แล้วนำข้อมูลที่ได้เข้าสู่รูปแบบการตัดสินใจแบบ M1 และ M2 เพื่อนำไปทำนายการตรวจจับการล้มของมนุษย์ โดยประมวลผลบนอุปกรณ์ Edge วิธีการที่นำเสนอนี้ช่วยลดต้นทุนอุปกรณ์ และไม่ต้องการให้ผู้ใช้สวมเซ็นเซอร์หรือใช้อุปกรณ์ถ่ายภาพเฉพาะ โดยระบบจะทำงานบนเครื่องประมวลผลปลายทาง (Edge Device) ซึ่งใช้ Google coral dev board ผลจากการทดลองในการตรวจจับการหกล้มพบว่า วิธี M1 มีค่า 99.8% และวิธี M2 มีค่า 97%

คำสำคัญ : อุปกรณ์ประมวลผลปลายทาง, การตรวจจับการล้ม, อุปกรณ์ที่พียู, จุดต่อโครงกระดูก, โครงกระดูก, โฟสเน็ต, กูเกิลโครอลเดฟบอร์ด

พจนัน ปณ จิตโต ชีเว

TITLE	Fall Detection Techniques Using TPU		
AUTHOR	Pornchai Thong-in		
ADVISORS	Assistant Professor Nattawoot Suwannata , Ph.D.		
DEGREE	Doctor of Philosophy	MAJOR	Electrical and Computer Engineering
UNIVERSITY	Maharakham University	YEAR	2024

ABSTRACT

This research project has presented the application of Pose estimation using the PoseNet model to generate skeletal joint points of a human from motion images obtained from a 2D RGB camera. The data extracted from this process is then fed into decision-making models, namely M1 and M2, to predict human fall detection. The processing is done on an Edge device, specifically the Google Coral Dev Board. This approach helps reduce equipment costs and eliminates the need for users to wear sensors or use specialized imaging equipment. The experimental results for fall detection show an accuracy of 99.8% for method M1 and 97% for method M2.

Keyword : Edge Computing, Fall Detection, Edge TPU, Keypoint, skeleton, PoseNet, Google Coral dev board

พหุ ประทีป ชีวะ

กิตติกรรมประกาศ

ข้าพเจ้าขอแสดงความขอบคุณอย่างสูงต่ออาจารย์ที่ปรึกษา ผศ.ดร.ณัฐวุฒิ สุวรรณทา ที่ให้คำแนะนำและสนับสนุนข้าพเจ้าตลอดระยะเวลาการศึกษา อาจารย์ได้มอบความรู้ ประสบการณ์ และคำปรึกษาอันมีค่าแก่ข้าพเจ้า ซึ่งมีส่วนสำคัญที่ทำให้ข้าพเจ้าสามารถสำเร็จการศึกษาระดับปริญญาเอกได้

ข้าพเจ้าขอขอบพระคุณคณาจารย์ทุกท่านในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหาสารคาม ที่ได้กรุณาให้ความรู้และคำแนะนำแก่ข้าพเจ้าตลอดระยะเวลาการศึกษา

ข้าพเจ้าขอขอบพระคุณครอบครัวและญาติพี่น้องทุกท่านที่ให้ความรัก ความอบอุ่น และกำลังใจแก่ข้าพเจ้าเสมอมา

ข้าพเจ้าหวังเป็นอย่างยิ่งว่าความรู้และประสบการณ์ที่ได้รับจากการศึกษาครั้งนี้ จะเป็นพื้นฐานในการนำไปประยุกต์ใช้ในการทำงานและดำเนินชีวิตต่อไป

ขอแสดงความนับถือ

พรชัย ทองอินทร์

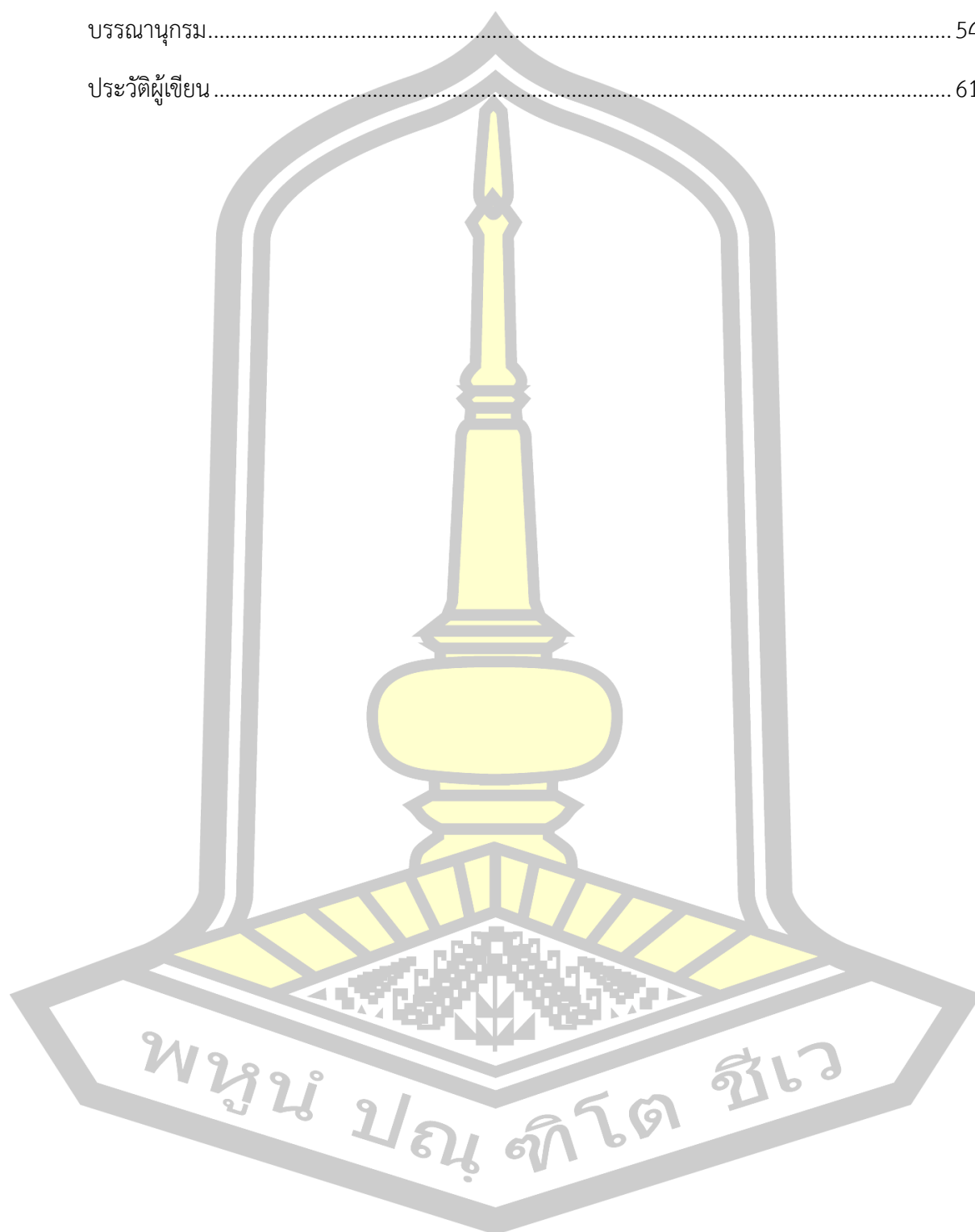


สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ญ
สารบัญภาพประกอบ.....	ฎ
บทที่ 1 บทนำ.....	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ความสำคัญของการวิจัย.....	2
1.4 ขอบเขตของการวิจัย.....	2
1.5 นิยามศัพท์เฉพาะ.....	2
บทที่ 2 ปรีทัศน์เอกสารข้อมูล	4
2.1 Pose estimation.....	4
2.2 เทคโนโลยี Edge Computing.....	6
2.3 การหกล้มของคน.....	7
2.4 งานวิจัยที่เกี่ยวข้อง.....	8
บทที่ 3 วิธีดำเนินการวิจัย.....	26
3.1 การเตรียมใช้งานอุปกรณ์ Edge	26
3.1.1 ความต้องการของระบบ.....	26
3.1.2 การโปรแกรมลงบอร์ด Coral Dev Board	26

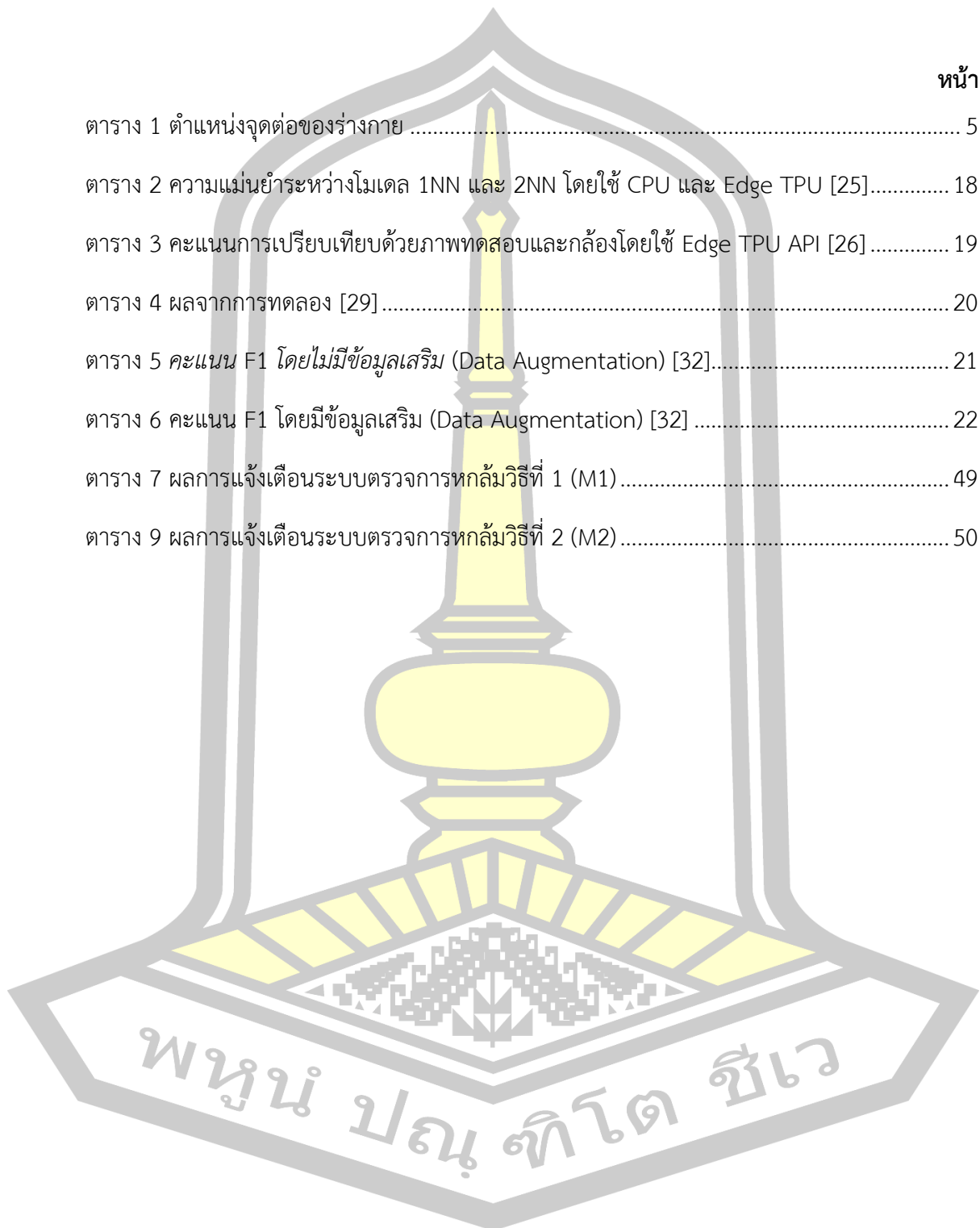
3.1.3 การติดตั้ง MDT	29
3.1.4 การอัปเดต Mendel บนบอร์ด	30
3.1.5 การทดสอบการทำงาน Machine Learning บนบอร์ด.....	30
3.2 ติดตั้งกล้องวิดีโอเข้ากับ Coral Dev Board	31
3.3 ออกแบบขั้นตอนการตรวจจับการล้มของมนุษย์.....	32
3.3.1 ขั้นตอนการตรวจจับการล้มวิธีที่ 1 (M1).....	33
3.3.2 ขั้นตอนการตรวจจับการล้มวิธีที่ 2 (M2).....	37
3.3.3 ความเร็วของส่วนหัว.....	38
3.3.4 การคำนวณระยะทางของ Keypoint.....	40
3.3.6 อัตราส่วนความกว้างและความสูงของร่างกาย	42
3.4 แปลงโมเดลให้ทำงานบน Coral Dev Board.....	45
บทที่ 4 ผลการทดลองและการอภิปรายผล.....	46
4.1 ผลการทดลอง.....	46
4.1.1 ผลการใช้โมเดล PoseNet.....	46
4.1.2 ผลการคำนวณหาค่าความเร็วส่วนหัว.....	47
4.1.3 ผลการคำนวณหาค่าระยะทางของ Keypoint	48
ผลการทดสอบประสิทธิภาพ 4.2การตรวจจับการล้มด้วยวิธีการประมวลผลบนอุปกรณ์ที่พ็ญ	49
4.3 การอภิปรายผล.....	50
4.3.1 การออกแบบการตรวจจับการล้มโดยตัดสินใจจากการประเมินท่าทางของมนุษย์	51
4.3.2 การใช้อุปกรณ์ปลายทางในการประมวลผล.....	51
บทที่ 5 สรุปผล และข้อเสนอแนะ.....	52
5.1 สรุปผล.....	52
5.1.1 ผลการทดสอบประสิทธิภาพ วิธีที่ 1 (M1).....	52
5.1.2 ผลการทดสอบประสิทธิภาพ วิธีที่ 2 (M2).....	52

5.2 ข้อเสนอแนะ	53
บรรณานุกรม.....	54
ประวัติผู้เขียน	61



สารบัญตาราง

	หน้า
ตาราง 1 ตำแหน่งจุดต่อของร่างกาย	5
ตาราง 2 ความแม่นยำระหว่างโมเดล 1NN และ 2NN โดยใช้ CPU และ Edge TPU [25]	18
ตาราง 3 คะแนนการเปรียบเทียบด้วยภาพทดสอบและกล้องโดยใช้ Edge TPU API [26]	19
ตาราง 4 ผลจากการทดลอง [29]	20
ตาราง 5 คะแนน F1 โดยไม่มีข้อมูลเสริม (Data Augmentation) [32]	21
ตาราง 6 คะแนน F1 โดยมีข้อมูลเสริม (Data Augmentation) [32]	22
ตาราง 7 ผลการแจ้งเตือนระบบตรวจการหลวมวิธีที่ 1 (M1)	49
ตาราง 9 ผลการแจ้งเตือนระบบตรวจการหลวมวิธีที่ 2 (M2)	50



สารบัญภาพประกอบ

	หน้า
ภาพประกอบ 1 แสดงตำแหน่งข้อต่อของร่างกาย	4
ภาพประกอบ 2 ตำแหน่งของอุปกรณ์ edge	6
ภาพประกอบ 3 แสดงวงสวิงกอล์ฟ 10 ตำแหน่ง [1]	8
ภาพประกอบ 4 Pipeline การตรวจจับการล้มด้วยท่าทางของมนุษย์ [5]	10
ภาพประกอบ 5 การสกัดเอาจุดต่อโครงกระดูก 7 จุด [10]	12
ภาพประกอบ 6 การตั้งค่าสวิตช์บูตเป็นโหมด SD การ์ด	27
ภาพประกอบ 7 ช่องเสียบการ์ด microSD อยู่ที่ยี่ด้านล่างของบอร์ด	27
ภาพประกอบ 8 สาย USB-C ที่เชื่อมต่อกับบอร์ด	28
ภาพประกอบ 9 สวิตช์บูตตั้งค่าเป็นโหมด eMMC	28
ภาพประกอบ 10 การเชื่อมต่อข้อมูลและการจ่ายไฟโดยใช้สาย USB-C	29
ภาพประกอบ 11 การทดสอบ Machine Learning บนบอร์ด	31
ภาพประกอบ 12 ตำแหน่งของ CSI Camera Connector	31
ภาพประกอบ 13 การต่อสายดินเข้ากับ CSI Camera Connector	32
ภาพประกอบ 14 การต่อสายดินเข้ากับกล้อง	32
ภาพประกอบ 15 ข้อมูลโครงกระดูก 17 จุด โดยได้จาก PoseNet	33
ภาพประกอบ 16 ขั้นตอนการตรวจจับการล้มวิธีที่ 1 (M1)	34
ภาพประกอบ 17 เมื่อส่วนหัวมีความเร็วในการเคลื่อนที่มากกว่า 20 จะแสดง Warning	35
ภาพประกอบ 18 กรณีไม่ล้มจะแสดงคำว่า Not Fall	36
ภาพประกอบ 19 กรณีล้มจะแสดงคำว่า Fall	36
ภาพประกอบ 20 ขั้นตอนการตรวจจับการล้มวิธีที่ 2 (M2)	38
ภาพประกอบ 21 การตรวจจับความเร็วส่วนหัว	39

ภาพประกอบ 22 การวัดระยะทางจะหัวไปยังข้อเท้า	40
ภาพประกอบ 23 ระยะทางจากหัวไปยังพื้นที่ด้านล่างของเฟรม	41
ภาพประกอบ 24 ระยะทางจากสะโพกไปยังพื้นที่ด้านล่างของเฟรม	42
ภาพประกอบ 25 ท่าเดินหรือยืน	43
ภาพประกอบ 26 ท่าล้มหรือนั่ง	43
ภาพประกอบ 27 ท่านอนหรือล้มลง	44
ภาพประกอบ 28 การแปลงโมเดลสำหรับ Coral Dev Board	45
ภาพประกอบ 29 ตัวอย่างผลที่ได้จากการหาตำแหน่งพิกัดในแต่ละเฟรมจากการใช้ PoseNet	47
ภาพประกอบ 30 ผลลัพธ์ค่าความเร็วส่วนหัว	47
ภาพประกอบ 31 ผลที่ได้จากการหาค่าระยะทางในแต่ละเฟรม	48
ภาพประกอบ 32 ผลที่ได้จากการประมวลผล FPS และ Inference time	48



บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

การหกล้มเป็นสาเหตุหนึ่งที่ทำให้ผู้สูงอายุที่มีอายุ 65 ปีขึ้นไปมีโอกาสเสียชีวิตโดยไม่ได้ตั้งใจ หากตรวจไม่พบเหตุการณ์การหกล้มอย่างทันท่วงที อาจนำไปสู่การบาดเจ็บสาหัสหรือถึงแก่ชีวิตได้ วิธีการตรวจจับการล้มทั่วไปจะกำหนดให้ผู้สวมเซ็นเซอร์ ซึ่งอาจยุ่งยากสำหรับผู้ใช้ในการสวมใส่ และการจัดตำแหน่งเซ็นเซอร์ที่ไม่ถูกต้องอาจทำให้อ่านค่าผิดพลาดได้ จากการศึกษาพบว่าผู้สูงอายุที่ได้รับความช่วยเหลือทันทีหลังจากการหกล้ม และสามารถลดความเสี่ยงของการเสียชีวิตได้อย่างมีประสิทธิภาพ และลดความเสี่ยงของการรักษาในโรงพยาบาลที่ต้องได้รับการรักษาในระยะยาว ดังนั้น การตรวจจับการหกล้มอย่างทันท่วงทีจึงเป็นสิ่งสำคัญ

การศึกษาในปัจจุบันเกี่ยวกับการตรวจจับการล้ม สามารถจัดกลุ่มได้เป็น 3 ประเภท ประเภทแรกใช้เซ็นเซอร์ที่สามารถสวมใส่ได้เพื่อตรวจจับการล้มและนำข้อมูลมาใช้ เช่น มาตรวัดความเร่งแบบสามแกนเพื่อตรวจจับท่าทางของบุคคล ประเภทที่สองใช้เซ็นเซอร์เพื่อตรวจจับการหกล้ม ผ่านการสัมผัสเตือน เซ็นเซอร์อินฟราเรด หรือเสียง ประเภทที่สามใช้คอมพิวเตอร์วิทัศน์และใช้ลำดับวิดีโอหรือภาพที่ถ่ายโดยกล้องวงจรปิดเพื่อตรวจจับการหกล้ม วิธีนี้ใช้ภาพที่ถ่ายมาเพื่อคำนวณรูปร่าง ท่าทาง หรือตำแหน่งศีรษะที่เปลี่ยนไป เพื่อตรวจจับการเปลี่ยนแปลงในลักษณะเชิงเวลาของการเคลื่อนไหว และท่าทางของร่างกายมนุษย์

วิธีตรวจจับการล้มโดยเซ็นเซอร์ที่สวมใส่ ส่วนใหญ่ใช้หน่วยวัดแรงเฉื่อย เช่น มาตรวัดความเร่งและไจโรสโคป เซ็นเซอร์วัดอัตราการเต้นของหัวใจ เครื่องวัดสัญญาณชีวการแพทย์ที่สวมใส่ เพื่อระบุลักษณะและตรวจจับการล้ม ระบบดังกล่าวใช้อัลกอริธึม การเรียนรู้ของเครื่อง (Machine Learning) เช่น Decision Tree (DT), K-Nearest Neighbor (KNN) และ Support Vector Machine (SVM) เพื่อตรวจจับการล้ม วิธีการนี้อาจให้เวลาตอบสนองที่รวดเร็วและความแม่นยำสูง แต่อย่างไรก็ตาม ผู้ใช้จำเป็นต้องสวมเซ็นเซอร์ และอาจทำให้ระบบใช้งานไม่ได้เมื่อผู้ใช้ล้มสวมใส่ อุปกรณ์ได้รับความเสียหาย หรือวางเซ็นเซอร์ไว้ไม่ถูกต้อง นอกจากนี้ เมื่อเกิดการหกล้ม ขนาดของตัวเครื่องอาจทำให้ผู้ใช้ได้รับบาดเจ็บสาหัสได้ ดังนั้นจึงจำเป็นต้องมีวิธีการตรวจจับการล้มที่ไม่ต้องสวมเซ็นเซอร์

วิธีการตรวจจับการล้มจากการมองเห็นด้วยคอมพิวเตอร์ (Computer Vision) จะใช้การประมวลผลภาพและแบบจำลองความน่าจะเป็น เพื่อตรวจจับรูปร่างของมนุษย์และการเรียนรู้ด้วยเครื่องจักร (Machine Learning) ที่ใช้โครงข่ายประสาทเทียม (Neural Network) สำหรับการจำแนกการเคลื่อนไหว วิธีการดังกล่าวจะดึงภาพเบื้องหน้าโดยมีเพียงวัตถุที่เคลื่อนไหวจากพื้นหลัง

ผ่านแบบจำลองผสมแบบเกาส์เซียน (GMM) เพื่อสร้างรูปร่างของร่างกายมนุษย์ วิธีนี้ช่วยขจัดปัญหาเซ็นเซอร์ที่สวมใส่ได้และไม่จำเป็นต้องใช้อุปกรณ์ราคาแพง แต่มีความอ่อนไหวต่อการรบกวนจากสิ่งแวดล้อม เช่น แสง เสื้อผ้า หรือภาพบุคคลที่ทับซ้อนกัน วิธีการที่ใช้คอมพิวเตอร์วิทัศน์สำหรับการตรวจจับการล้มอีกประเภทหนึ่ง คือ การใช้จุดต่อโครงกระดูกมนุษย์จาก Kinect และ OpenPose หรือ Deepcut เพื่อให้ได้จุดต่อโครงกระดูกเพื่อระบุและทำนายการล้ม ตามการเรียนรู้ของเครื่องหรือการเรียนรู้เชิงลึก (Deep Learning) การจดจำโครงกระดูกมีแนวโน้มที่จะถูกรบกวนจากสิ่งแวดล้อมน้อยกว่าการจดจำรูปร่างของมนุษย์ เนื่องจากข้อมูลการฝึกจะอิงตามจุดต่อของโครงกระดูกซึ่งไม่ใช่รูปภาพ เวลาในการฝึกจึงสั้นลง อย่างไรก็ตาม วิธีการนี้ได้รับผลกระทบจากคุณภาพของคุณลักษณะที่ไม่ดีและจุดร่วมที่ขาดหายไป ซึ่งส่งผลให้ความแม่นยำในการตรวจจับการล้มลดลง

งานวิจัยฉบับนี้ ได้นำเสนอการประยุกต์ใช้ Pose estimation โดยใช้รูปแบบ PoseNet เป็นตัวสร้างจุดต่อโครงกระดูกของคนจากภาพเคลื่อนไหว แล้วนำข้อมูลที่ได้นำเข้าสู่รูปแบบการตัดสินใจแบบ M1 และ M2 เพื่อนำไปทำนายการตรวจจับการล้มของคนโดยประมวลผลบนอุปกรณ์ Edge วิธีการที่นำเสนอนี้ช่วยลดต้นทุนอุปกรณ์ และไม่ต้องการให้ผู้สวมเซ็นเซอร์หรือใช้อุปกรณ์ถ่ายภาพเฉพาะ ใช้เวลาน้อยในการเทรนระบบ โดยระบบจะทำงานบนเครื่องประมวลผลปลายทาง (Edge Device) ซึ่งใช้ Google coral dev board

1.2 วัตถุประสงค์ของการวิจัย

เพื่อหารูปแบบตรวจจับการล้มของคนโดยทำงานบนอุปกรณ์ Edge

1.3 ความสำคัญของการวิจัย

งานวิจัยนี้จะทำให้ได้รูปแบบของการตรวจจับการล้มของคน โดยการใช้การประมวลผลจากอุปกรณ์ Edge (Google coral dev board) และสามารถนำไปประยุกต์ใช้งานจริงได้

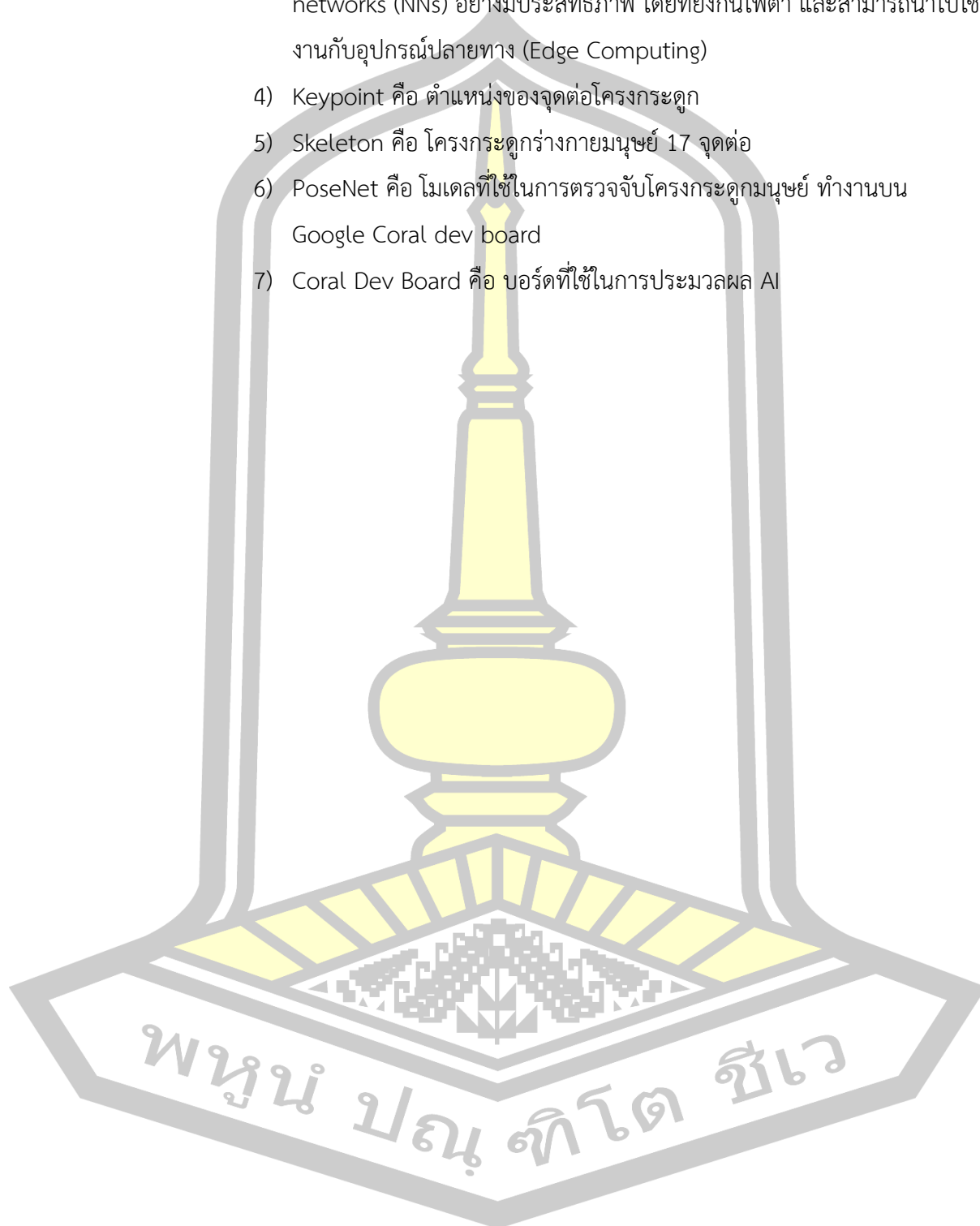
1.4 ขอบเขตของการวิจัย

ใช้รูปแบบ PoseNet ในการสร้างภาพจุดต่อโครงกระดูกมนุษย์จากภาพเคลื่อนไหว และ ใช้ Google coral dev board เป็นอุปกรณ์ Edge ประมวลผลการตรวจจับการล้ม

1.5 นิยามศัพท์เฉพาะ

- 1) Edge Computing คือ การประมวลผลโดยอุปกรณ์ปลายทาง
- 2) Fall Detection คือ ระบบตรวจจับการล้ม

- 3) Edge TPU คือ ชิป ASIC ที่ Google ออกแบบเองให้ประมวลผล neural networks (NNs) อย่างมีประสิทธิภาพ โดยที่ยังกินไฟต่ำ และสามารถนำไปใช้งานกับอุปกรณ์ปลายทาง (Edge Computing)
- 4) Keypoint คือ ตำแหน่งของจุดต่อโครงกระดูก
- 5) Skeleton คือ โครงกระดูกร่างกายมนุษย์ 17 จุดต่อ
- 6) PoseNet คือ โมเดลที่ใช้ในการตรวจจับโครงกระดูกมนุษย์ ทำงานบน Google Coral dev board
- 7) Coral Dev Board คือ บอร์ดที่ใช้ในการประมวลผล AI



บทที่ 2

ปริทัศน์เอกสารข้อมูล

2.1 Pose estimation

Pose Estimation คือ การประมาณท่าทางเป็นการใช้แบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) เพื่อประเมินท่าทางของบุคคลจากภาพหรือวิดีโอโดยการประเมินตำแหน่งเชิงพื้นที่ของข้อต่อที่สำคัญของร่างกาย เป็นเทคนิคการมองเห็นด้วยคอมพิวเตอร์ที่ตรวจจับร่างกายมนุษย์ในรูปภาพและวิดีโอ เพื่อให้สามารถระบุได้ ตัวอย่างเช่น ตำแหน่งที่ข้อศอกของใครบางคนปรากฏขึ้นในรูปภาพ สิ่งสำคัญคือต้องตระหนักถึงข้อเท็จจริงที่ว่า การประมาณค่าท่าทางเป็นเพียงการประมาณว่าข้อต่อหลักของร่างกายอยู่ที่ไหนและไม่รู้จักรว่าใครอยู่ในภาพหรือวิดีโอ แบบจำลองการประมาณค่าท่าทางจะใช้ภาพของกล้องที่ประมวลผลเป็นข้อมูลเข้าและส่งออกข้อมูลเกี่ยวกับจุดสำคัญ จุดสำคัญที่ตรวจพบจะถูกสร้างขึ้นโดยรหัสชิ้นส่วน โดยมีคะแนนความเชื่อมั่นระหว่าง 0.0 ถึง 1.0 คะแนนความเชื่อมั่นบ่งชี้ความน่าจะเป็นที่มีจุดสำคัญอยู่ในตำแหน่งนั้นแบบจำลองการประมาณค่าใน TensorFlow Lite ซึ่งจะใช้ในบอร์ด Coral Dev Board จะมี 2 แบบ คือ MoveNet และ PoseNet



ภาพประกอบ 1 แสดงตำแหน่งข้อต่อของร่างกาย

ที่มา : <https://github.com/google-coral/project-posenet>

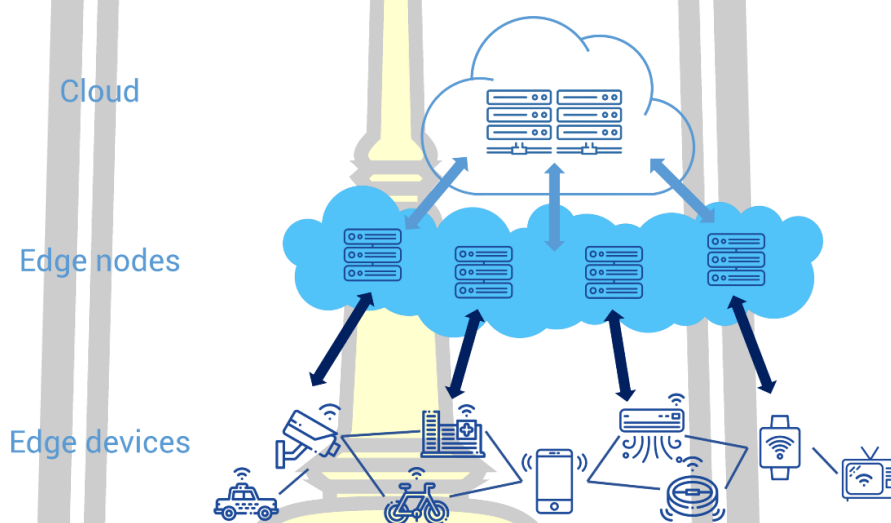
ตาราง 1 ตำแหน่งจุดต่อของร่างกาย

ตำแหน่ง	ส่วนของร่างกาย
0	จมูก
1	ตาซ้าย
2	ตาขวา
3	หูซ้าย
4	หูขวา
5	ไหล่ซ้าย
6	ไหล่ขวา
7	ข้อศอกซ้าย
8	ข้อศอกขวา
9	ข้อมือซ้าย
10	ข้อมือขวา
11	สะโพกซ้าย
12	สะโพกขวา
13	เข่าซ้าย
14	เข่าขวา
15	ข้อเท้าซ้าย
16	ข้อเท้าขวา

สำหรับแบบจำลอง PoseNet โดยใช้ MoveNet จะมี 2 รูปแบบคือ แบบที่ 1 MoveNet.Lighting มีขนาดเล็กกว่า เร็วกว่า แต่แม่นยำน้อยกว่ารุ่น Thunder สามารถทำงานแบบเรียลไทม์บนสมาร์ทโฟนสมัยใหม่ และแบบที่ 2 MoveNet.Thunder เป็นรุ่นที่แม่นยำกว่า แต่ยิ่งใหญ่กว่าและช้ากว่า Lightning มีประโยชน์สำหรับกรณีการใช้งานที่ต้องการความแม่นยำสูงกว่า MoveNet มีประสิทธิภาพเหนือกว่า PoseNet แบบอื่นในชุดข้อมูลที่หลากหลาย โดยเฉพาะอย่างยิ่งในรูปภาพที่มีภาพการออกกำลังกาย ข้อต่อต่างๆ ของร่างกายที่ตรวจพบโดยแบบจำลองการประมาณการทำท่าทาง แสดงในภาพประกอบ 2

2.2 เทคโนโลยี Edge Computing

Edge Computing คือ แนวคิดในการประมวลผลใกล้กับแหล่งกำเนิดข้อมูลมากที่สุด ซึ่งเป็นเรื่องของตำแหน่งการทำงานของอุปกรณ์ ซึ่งแทนที่การส่งข้อมูลทั้งหมดจะถูกส่งไปประมวลผลที่ Cloud Server อาจจะอยู่ในรูปแบบการวิเคราะห์ข้อมูล การประมวลผลข้อมูลเชิงสถิติ โดยตำแหน่งการประมวลผลของ Edge นั้นสามารถเป็นได้ตั้งแต่บนตัวอุปกรณ์ IoT, อุปกรณ์เกตเวย์, Server เป็นต้น



ภาพประกอบ 2 ตำแหน่งของอุปกรณ์ edge

ประโยชน์ของ Edge Computing สรุปได้ดังนี้

1. ความเร็วในการรับส่ง ความเร็วในการรับส่งข้อมูลจัดว่าเป็นข้อดีของ Edge Computing เพราะถ้าส่งข้อมูลขึ้นไปประมวลผลบน Cloud ทุกครั้งเพื่อหาคำตอบ จากบริการ API ต่างๆ ก็จะใช้เวลานานกว่า เนื่องจากต้องประมวลผลข้อมูลทั้งหมด ดังนั้น Edge Computing จึงมีความเร็ว เพราะไม่จำเป็นต้องส่งข้อมูลไปประมวลผลบน Cloud ทุกครั้ง
2. ความปลอดภัย ความเป็นส่วนตัวและความปลอดภัยเป็นประเด็นที่ทุกคนให้ความสำคัญ ประโยชน์ของการใช้ Edge computing จะช่วยเพิ่มความสามารถในการป้องกันจากการโจมตีของผู้ไม่หวังดี เพราะ การที่ไม่จำเป็นต้องเอาข้อมูลส่งออกไปเก็บหรือประมวลผลที่อื่น จะสามารถช่วยในเรื่องความปลอดภัยของข้อมูลได้
3. ประหยัด Bandwidth ในปัจจุบันทุกครอบครัวมีการใช้งานอุปกรณ์เทคโนโลยีกันเป็นจำนวนมาก และหากจะต้องส่งข้อมูลไปยังระบบ Cloud จะทำให้ต้องใช้ Bandwidth จำนวนมาก การใช้ Edge computing จะช่วยให้อุปกรณ์ต่างๆ ที่มีระบบ AI สามารถประมวลผลข้อมูลในตัวได้

เลย ทำให้การประมวลผลบน Cloud ลดลง และเมื่อการประมวลผลลดลง ก็จะทำให้จำนวนการใช้ Bandwidth ลดลงด้วยเช่นกัน

2.3 การหกล้มของคน

ความหมายของการหกล้ม จากการทบทวนวรรณกรรม พบว่ามีผู้ให้ความหมายของการหกล้มในผู้สูงอายุที่แตกต่างกันดังนี้

ดุษฎี ปลาฤทธิ์ (2544) อุบัติเหตุพลัดตกหกล้ม หมายถึง เหตุการณ์ที่เกิดขึ้นกับผู้สูงอายุ โดยไม่ได้คาดคิดมาก่อน ไม่ได้ตั้งใจ เกิดขึ้นโดยบังเอิญ ทำให้ร่างกายสูญเสียสมดุลและตกไปพื้น หรือในระดับที่ต่ำกว่า ซึ่งมีสาเหตุจากการสะดุด การลื่น การตกบันได ตกเก้าอี้ หรือเตียง

ลัดดา เกียมวงศ์ (2544) การหกล้ม หมายถึง บุคคลที่สูญเสียการทรงตัวโดยไม่ตั้งใจ และไม่ได้เกิดจากการแรงกระทำภายนอก โดยทำให้ส่วนใดส่วนหนึ่งของร่างกาย ได้แก่ มือ แขน เข่า ก้น หรือร่างกายทั้งตัวสัมผัสกับพื้น และต้องเกิดเหตุการณ์ดังกล่าวตั้งแต่ 2 ครั้งขึ้นไป

บุปผา จันทจรรัส (2546) การหกล้ม หมายถึง การเปลี่ยนแปลงตำแหน่งของร่างกายจากการลื่นถลาหรือตกลงไปสู่พื้นผิวอื่นที่ต่ำกว่าร่างกาย เป็นเหตุการณ์ที่เกิดขึ้นโดยไม่ตั้งใจ และไม่สามารถควบคุมได้ อาจส่งผลให้ร่างกายของผู้ป่วยได้รับการบาดเจ็บก็ได้

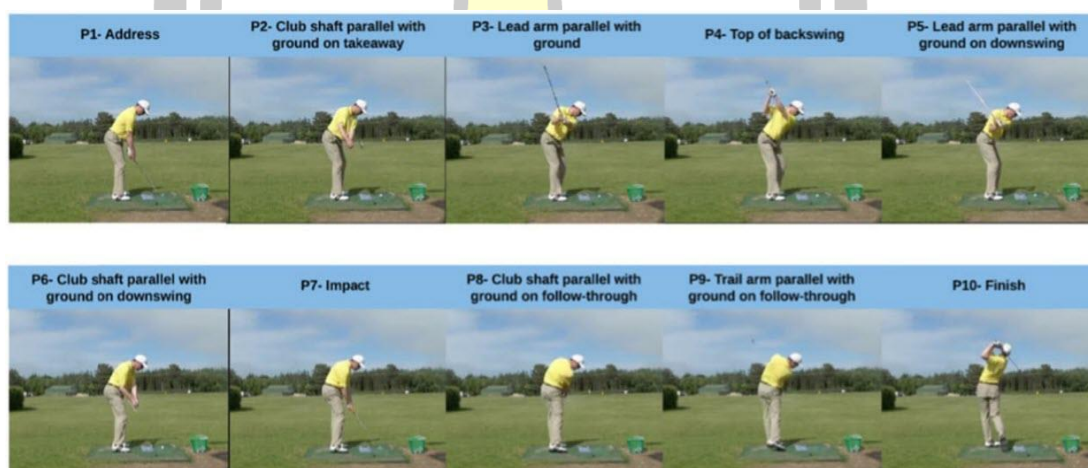
อารี ปรมัตถการ และคณะ (2553) คำจำกัดความของคำว่า “หกล้ม” ในทางวิจัย หมายถึง ภาวะที่ผู้สูงอายุล้มลงไปสู่พื้นหรือพบว่านอนอยู่ที่พื้น หรือเป็นภาวะที่ล้มลงไปกระทบกับวัสดุอุปกรณ์ที่อยู่ในบริเวณนั้น เช่น เก้าอี้เคาน์เตอร์ แล้วต้องพยายามดิ้นตัวกลับมาเพื่อการทรงตัว ความหมายของการหกล้มในผู้สูงอายุ คือ ภาวะที่บุคคลล้มลงไปกระทบกับวัสดุอุปกรณ์ที่อยู่ในบริเวณนั้น เช่น พื้น เก้าอี้ โต๊ะ บันได และการหกล้มนั้นอาจส่งผลให้ร่างกายของผู้สูงอายุได้รับบาดเจ็บหรือไม่ได้รับบาดเจ็บก็ได้

น้อมจิตต์ นวลเนตร์ (2543) การล้ม หมายถึง เหตุการณ์ที่บุคคลหนึ่งสูญเสียการทรงตัว อย่างไม่ได้ตั้งใจ ทำให้มือ แขน เข่า ก้น หรือร่างกายทั้งตัวต้องสัมผัสหรือกระทบกับพื้นโดย เหตุการณ์ที่เกิดขึ้นนี้ไม่ได้สาเหตุจากตัวบุคคลนั่นเอง เช่น เป็นลม หรือมีอาการกล้ามเนื้ออ่อนแรงอย่างกะทันหัน เป็นต้น

สุทธิชัย จิตะพันธ์กุล (2544) การหกล้ม หมายถึง การที่เกิดการเปลี่ยนท่าโดยไม่ตั้งใจและเป็นผลให้ร่างกายทรุดหรือลงนอนกับพื้น หรือปะทะสิ่งของต่างๆ เช่น โต๊ะ เตียง การเปลี่ยนท่าโดยไม่ตั้งใจเหล่านี้อาจเกิดจากอาการหน้ามืดเป็นลม ขาอ่อนแรง และจากการสะดุด เกยวติง ลื่นไหล โดยไม่รวมถึงการถูกชก หรือตี ทำร้าย

2.4 งานวิจัยที่เกี่ยวข้อง

ปี ค.ศ. 2020 THEODORE T. KIM และคณะ [1] ได้นำเสนอการประมาณการทำทาง เพื่อทำนายประสิทธิภาพของวงสวิงนักกอล์ฟสมัครเล่นโดยทำงานบนอุปกรณ์ edge processing โดยเลือกใช้บอร์ด coral dev board ติดกล้องวิดีโอทำการจับภาพวงสวิงนักกอล์ฟเป็นภาพวิดีโอ ในขณะที่กำลังตีกอล์ฟ ในงานวิจัยนี้ได้แยกวงสวิงกอล์ฟออกเป็น 10 ตำแหน่งวิธีการทดลอง จะเริ่ม จาก ข้อมูลวิดีโอของการสวิงจากตำแหน่งกล้องลงมาจะถูกเก็บรวบรวม ต่อจากนั้นวิดีโอจะถูกแปลง เป็น เฟรมเดี่ยว โดยโมเดลแมชชีนเลิร์นนิง PoseNet ที่ทำงานบน TPU รวบรวม ตำแหน่งท่าโพส แบบ เฟรมต่อเฟรมและทำให้ข้อมูลจะเรียงขึ้นโดยใช้ตัวกรอง Savitzky-Golay เพื่อช่วยลดการ เปลี่ยนแปลงกะทันหันที่เกิดจากความไม่ถูกต้องของ PoseNet จากนั้นก็สร้างโมเดล Key Frame Selector เพื่อเน้นเฟรมที่น่าสนใจ สร้างข้อมูลบางส่วนผ่านการรวบรวมวิดีโอที่สนามไตรฟ์ ซึ่งเน้นถึง ความจำเป็นในการกรองข้อมูลและสามารถทำงานบนบอร์ดได้ดีและใช้ KFS พล็อตเตอร์ส่วนที่สำคัญ เพื่อทำนายผลการสวิง



ภาพประกอบ 3 แสดงวงสวิงกอล์ฟ 10 ตำแหน่ง [1]

งานวิจัยนี้ทดลองกับบอร์ด Coral Dev Board เริ่มด้วยการใช้กล้องถ่ายภาพวิดีโอเพื่อดูว่า TPU จะสามารถประมวลผลภาพจากกล้องถ่ายทอดสดด้วยโมเดลได้เร็วเพียงใด มีการใช้ขนาดของภาพที่แตกต่างกันสองแบบซึ่งรวมอยู่ใน PoseNet Repository รุ่นหนึ่งรูปมีขนาด 641×481 และอีกรุ่นหนึ่ง รูปมีขนาด 1281×781 การเรียกใช้โมเดลความละเอียดประมาณ 30 FPS บน Coral DEV Board ซึ่งเป็นโมเดลที่มีความละเอียดสูงกว่านั้นมีประสิทธิภาพที่ต่ำกว่าที่ประมาณ 20 FPS ซึ่งจะใช้เป็น พื้นฐานในการพิจารณาว่า สามารถประมาณค่าตามเวลาจริงได้ด้วยตำแหน่ง Pose Estimate

หรือไม่ และความแม่นยำสำหรับการระบุวงสวิง P1-P10 นั้น สามารถเรียกใช้งานแบบเรียลไทม์ได้หรือไม่ในการเริ่มต้นเลือกที่จะตรวจสอบว่าข้อมูลจะปรากฏที่ 30 FPS ซึ่งมีความเหมาะสมกว่าการตรวจสอบ ตัวอย่างวงสวิงและการคำนวณเวลาโดยประมาณของวงสวิงกอล์ฟแบบปกติพบว่า 30 FPS จะยอมรับได้ในกรณีส่วนใหญ่แม้สวิงมีจุดที่สำคัญที่สุดที่ระบุว่าการสวิงจะสำเร็จหรือไม่ ตำแหน่ง เหล่านี้จะเป็นจุดเริ่มต้นที่ดีสำหรับคนที่ให้ความสำคัญกับการปรับปรุงวงสวิง ข้อมูลเพียงพอจะถูกบันทึกไว้เพื่อวิเคราะห์คุณลักษณะบางอย่าง แต่โมเดลจะต้องทำงานเร็วขึ้นเพื่อวิเคราะห์การเคลื่อนไหว ที่รวดเร็วอย่างแม่นยำ โดยทั่วไปการสวิงสามารถทำได้ภายใน 3 ถึง 4 วินาที ซึ่งแปลงได้เพียง 240 จุดเท่านั้นโดยการกระจายน้ำหนักไปที่แบ็คสวิงที่ช้ามาก ทำให้ข้อมูลมีความเที่ยงตรงสูงขึ้น

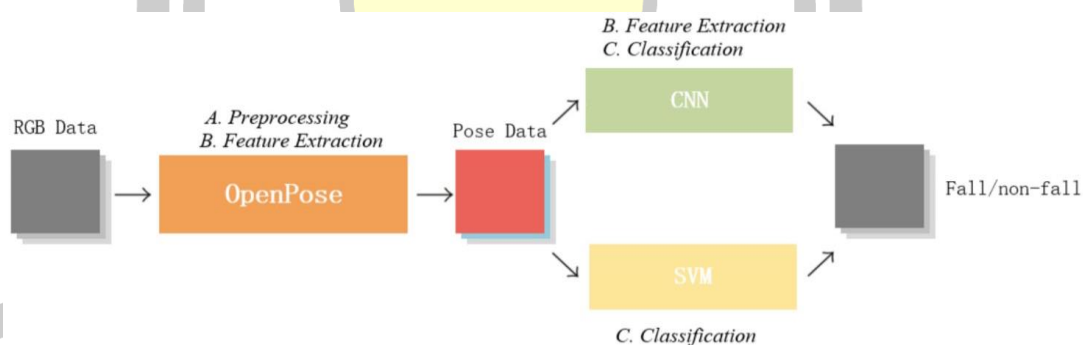
ปี ค.ศ. 2017 Markus D. Solbach และ John K. Tsotsos [2] ได้นำเสนอ การใช้ Vision-Based ในการตรวจจับการล้มของผู้สูงอายุ โดยตรวจจับจากการประมาณท่าทางของมนุษย์ (Human Pose Estimation) ในลักษณะ 2 มิติ มีสี่ตัวร่วมกับข้อมูลที่มีความลึกที่เป็นคุณลักษณะสำคัญของข้อมูลท่าทางมนุษย์แบบ 3 มิติ คำนวณระนาบพื้นในแบบ 3 มิติและใช้หลายค่า เพื่อตรวจสอบว่าบุคคลนั้นล้มหรือไม่ โดยมีการติดกล้องเพื่อสังเกตการณ์แบบติดกับที่และสามารถติดกล้องได้หลายๆตัว และมีการใช้กล้องสเตอริโอใช้ CNN ในการตรวจจับบุคคลแบบ 2 มิติ และเพื่อให้ รับรู้สภาพแวดล้อมและคำนวณความลึกของภาพ การประมวลผลแบบสเตอริโอจะแก้ไขภาพสี ด้านซ้ายและขวา และคำนวณความลึกของภาพ 2D Human Pose Estimation โดยใช้ Deep Learning เพื่อประเมินท่าทางของมนุษย์ในแบบ 2D ขั้นตอนนี้ทำให้ได้พิกัดพิกเซลสำหรับแต่ละจุด สำคัญ ที่ตรวจพบและระดับความมั่นใจสำหรับขั้นตอนนี้จะใช้ภาพสีด้านซ้ายการคำนวณ 3D Pose ใช้ ข้อมูลเมทริกซ์ของกล้อง depth image และข้อมูลจุดสำคัญของขั้นตอนก่อนหน้าเพื่อนำข้อมูล จุดสำคัญลงใน 3D World-Coordinates สำหรับ 3D Ground Plane Detection ใช้ความลึกภาพเช่นกัน เพื่อคำนวณระนาบพื้นในแบบ 3 มิติ ซึ่งเป็นสิ่งสำคัญ สำหรับขั้นตอนการทำนายว่าล้มหรือไม่ การตัดสินใจขึ้นอยู่กับพื้นระนาบท่า 3 มิติ และระดับ ความมั่นใจ ของแต่ละจุดสำคัญที่ตรวจพบ ขึ้นอยู่กับผลลัพธ์ของขั้นตอนการทำนาย ระบบจะทำการแจ้งเตือน หากตรวจพบการล้มหรือเริ่มใหม่ด้วยกล้องสเตอริโอ มีความแม่นยำที่สูงกว่า 91%

ปี ค.ศ. 2019 Sungil Jeong, Sungjoo Kang และ Ingeol Chun ได้นำเสนอวิธีการตรวจจับการล้มจากภาพโครงกระดูกมนุษย์โดยใช้ LSTM สำหรับอุตสาหกรรม [3] การผลิต โดยใช้กล้อง วงจรปิดชนิด 2D RGB เป็นสัญญาณวิดีโอในการตรวจจับบุคคล และจะเลือกเอาเฉพาะจุดต่อโครงกระดูกของมนุษย์ ซึ่งใช้ไลบรารี Open Pose สร้างข้อมูลจุดต่อโครงกระดูกจากวิดีโอ และทำการจำแนก แบ่งกลุ่มของภาพโดยใช้รูปแบบ ของ Long Short Time Memory (LSTM) เพื่ออนุมานว่าล้มหรือไม่ล้ม มีการใช้ชุดข้อมูลการล้มของคนจาก UR Fall Detection (URFD) Dataset และ

SDUFall Dataset ซึ่งทั้ง 2 ชุดข้อมูลนี้ถูกบันทึกด้วย Microsoft Kinect ในการสกัดคุณลักษณะ ที่สำคัญจะใช้รูปแบบของ RAW, HCLC และ SHCLC โดยมีค่าความแม่นยำ 95.28 เปอร์เซ็น

ปี ค.ศ. 2019 Hiroaki Kingetsu, Takeshi Konno, Shuji Awai, Daisuke Fukuda, Toshihiro Sonoda [4] ได้นำเสนอ ระบบตรวจจับความเสี่ยงจากการหกล้มโดยใช้วิดีโอสำหรับผู้สูงอายุ ใช้กล้องวิดีโอ SONY รุ่น HDR-CX470W เพื่อบันทึก 12 คน ทำการทดสอบการเดินระยะทาง 5 เมตรในบ้านพักคนชรา ในพื้นผิวเรียบไม่มีสิ่งกีดขวาง ตั้งค่ากล้องห่างจากจุดศูนย์กลางของพื้นที่ตรวจจับ 5 เมตร เพื่อให้มั่นใจว่าทุกช่วงอยู่ในระยะการมองเห็นของกล้องใช้ OpenPose ในการตรวจจับ 24 จุดสำคัญบนร่างกายและมือของมนุษย์ประมวลผลข้อมูลการถ่ายภาพเป็นภาพเดี่ยว งานวิจัยนี้ใช้ข้อมูลจากการบันทึกวิดีโอของอาสาสมัคร 8 คน เป็นข้อมูลการเทรนและใช้อีก 4 คนที่เหลือเป็นข้อมูลการทดสอบอัลกอริทึมการเรียนรู้ใช้ Support Vector Machine (SVM), Logistic regression และ Linear discriminant analysis โดยมีค่าความแม่นยำดังนี้ Support Vector Machine (SVM) มีค่าความแม่นยำ 100% ,Logistic regression มีค่าความแม่นยำ 100% และ Linear discriminant analysis มีค่าความแม่นยำ 75%

ปี ค.ศ. 2018 Zhanyuan Huang, Yang Liu, Yajun Fang และ Berthold K. P. Horn. [5] ได้นำเสนอ การตรวจจับการล้มด้วยวิดีโอสำหรับผู้สูงอายุด้วยการประเมินท่าทางของมนุษย์ โดยมีวิธีการดังภาพประกอบ 4



ภาพประกอบ 4 Pipeline การตรวจจับการล้มด้วยท่าทางของมนุษย์ [5]

งานวิจัยนี้เสนอ pipe line การตรวจจับการล้มใช้ภาพจากกล้องวิดีโอ 2 มิติ ที่มีการ ประเมิน ท่าทางของมนุษย์ โดยขั้นแรกใช้กล้องวิดีโอแบบ RGB จับภาพแล้วใช้ OpenPose เพื่อหา จุดสำคัญของร่างกายมนุษย์เป็นลักษณะรูปโครงกระดูกเป็นข้อมูล Pose Data ซึ่งได้ข้อมูลแบบ RGB ขั้นที่สอง ข้อมูลเหล่านี้จะกลายเป็นอินพุตของ Convolution Neural Network เพื่อทำการแยก คุณลักษณะ แบบหลายชั้น (multi-layered) อันดับที่สามทำการแบ่งกลุ่มข้อมูล โดยใช้ Binary

Classification ผ่านโครงข่ายประสาทเทียมสำหรับการเปรียบเทียบใช้ SVM เป็นตัวแยกประเภท ซึ่งผลที่ได้มีความแม่นยำค่อนข้างสูงเมื่อเปรียบเทียบผลลัพธ์กับแนวทางที่ทันสมัยแบบอื่น ๆ ใช้ชุดข้อมูลสามชุด คือ Multiple Cameras Fall Dataset, UR Fall Detection Dataset และ The Fall Detection Dataset

ปี ค.ศ. 2019 Pichayakul Jenpoomjai, Potsawat Wosri, Chih-Lin Hu และ Somchoke Ruengittinun [6] ได้นำเสนอ VA Algorithm สำหรับการตรวจจับการล้มของผู้สูงอายุ ด้วย 2D-Pose-Estimation ใช้ VA อัลกอริทึมซึ่งเป็นการตรวจสอบความเร็วและความเร่งในการตรวจจับการล้มของคน โดยการตรวจจับตำแหน่งของคอและสะโพกแล้วทำการคำนวณหาค่าความเร็วและความเร่ง โดยปกติแล้วความเร็วและความเร่งในการทำกิจกรรมประจำวันจะน้อยกว่าค่าการล้ม งานวิจัยนี้ใช้ Library Tensor Flow เพื่อเรียนรู้การประมาณท่าทางของคน ซึ่งจะเลือกค่าจุดบางจุดของร่างกายเพื่อนำไปวิเคราะห์พฤติกรรมจะประกอบไปด้วยค่าความเร็วและค่าความเร่ง แล้วนำไปคำนวณหาค่าเทียบกับค่าที่ตั้งไว้ ถ้ามีค่าน้อยกว่าค่าที่ตั้งไว้ก็จะทำนายว่ามีการล้มเกิดขึ้นซึ่งมีความแม่นยำอยู่ที่ 88 % โดยการตรวจจับบุคคลคนเดียว

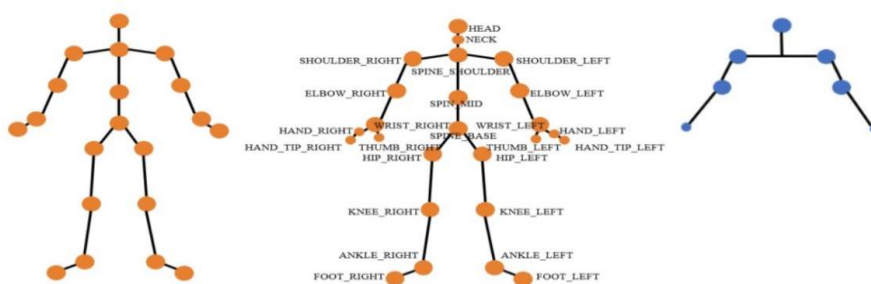
ปี ค.ศ. 2019 Md Mahedi Hasan, Md Shamimul Islam, Sohaib Abdullah [7] ได้นำเสนอการตรวจจับการล้มโดยอาศัย Robust Pose-Based Human โดยใช้โครงข่ายประสาทเทียม Recurrent Neural Network (RNN) แบบ Long Short Time Memory (LSTM) สำหรับการตรวจจับการล้มโดยใช้ข้อมูลจากวิดีโอเป็นข้อมูลแบบ 2D ในลักษณะที่เป็นจุดต่อโครงกระดูกโดยมีข้อมูลในตำแหน่งจุดต่างๆบนร่างกาย 25 จุด และพบว่าข้อไหล่ สะโพก และเข่า สามารถสร้างลักษณะ การล้มที่ชัดเจนและเป็นจุดดีที่สุดกว่าจุดอื่นๆ ดังนั้นจึงเลือกข้อต่อ 8 ข้อ เข่าขวา, เข่าซ้าย, สะโพกขวา, ตรงกลางสะโพก, สะโพกซ้าย, ไหล่ขวา, ไหล่ซ้าย และ คอ เป็นตัวอธิบายการล้ม โดยใช้ชุดข้อมูลของ FDD และ URFD ของการตรวจจับการล้ม ซึ่งวิธีการที่ เสนอมีความแม่นยำที่ 99.0% ทั้งคู่

ปี ค.ศ. 2019 Lehan Chen, Xiangbo Kong, Hiroyuki Tomiyama และ Lin meng. [8] ได้นำเสนอ การตรวจจับการล้มในหลายเมืองสำหรับประชากรผู้สูงอายุ โดยได้เลือก AlexNet กับภาพ pretreated ให้เป็น model ที่เหมาะสมที่สุดในการตรวจจับการล้มในการทดลองนี้ งานวิจัยนี้ได้สร้างชุดข้อมูลลักษณะของภาพของผู้คนโดยใช้กล้อง Realsense D435 จากนั้นทำการ ปรับภาพให้เป็นมาตรฐานและแก้ไขภาพที่มีข้อบกพร่องเพื่อปรับปรุงค่าที่ได้ จากการเทรนโมเดลให้มี ค่าที่เหมาะสมต่อจากนั้นได้ค้นหาสาเหตุของความแม่นยำที่มีค่าต่ำในการจดจำ ด้วยการปรับ ภาพให้เป็นมาตรฐานและสร้าง pretreatment layer เพื่อปรับปรุงอัตราการเรียนรู้สำหรับการตรวจ ตรวจจับการล้ม ซึ่งมีความแม่นยำอยู่ที่ 99.9% ของแต่ละรัฐ ในการทดลองนี้มีปัญหามากมาย ประการแรก สภาพแวดล้อม ต้องไม่รบกวนจากวัตถุอื่นและไม่กีดขวางมุมมองจากกล้องไปยังบุคคลที่ตรวจพบ

ประการที่สองอัลกอริธึมในตรวจจับร่างกายมีข้อบกพร่องไม่ถึงประสิทธิภาพที่คาดหวัง นอกจากนี้ข้อจำกัดของมุมมองในกล้อง RealSense D435 ยังเป็น ปัญหาในการใช้งาน จริงอีกด้วย

ปี ค.ศ. 2020 Jiawei Li, Shu-Tao Xia, และ Qianggang Ding [9] ได้นำเสนอ การรู้จำหลายระดับเกี่ยวกับการหกล้มจากกิจกรรมในชีวิตประจำวัน ในงานวิจัยนี้ ได้กำหนดให้ computer vision ทำงานหลายระดับสำหรับปัญหาการรู้จำการล้มจากการสังเกตด้วยสายตา และศึกษาวิธีการไปป์ไลน์ (pipe line) มีการจัดทำ frame level labels สำหรับท่าทาง ชุดข้อมูลการล้มในชีวิตประจำวันหลายชุด เพื่อทดสอบ วิธีการและสนับสนุนการวิเคราะห์การล้ม มีการใช้การจับภาพ การเคลื่อนไหวท่าทางแบบ Dynamic Pose Motion (DPM) ซึ่งสามารถ จับภาพได้โดยการ สกัดคุณลักษณะที่สำคัญของการเคลื่อนไหว นอกจากนี้ยังมีการนำเสนอไปป์ไลน์ การรู้จำลำดับระดับ การล้ม (sequence-level fall recognition pipeline) ซึ่งมีโครงสร้าง สองรูปแบบที่ชัดเจน คือ ลักษณะที่ปรากฏคุณสมบัติการเคลื่อนที่และมีรูปแบบ LSTM เพื่อสร้างแบบ จำลองชั่วคราว และการทำนายการล้มการทำงานในทางเทคนิค อันดับแรกจะแยกคุณลักษณะของ ท่าทาง (Pose) และคุณลักษณะของ Dynamic Pose Motion (DPM) จากลำดับของภาพแบบ 2D RGB จากนั้น ก็ทำการเทรนข้อมูลโดยใช้โมเดล CNN-LSTM สำหรับการทำนาย ซึ่งจะทำการ สุ่มตัวอย่างจาก วิดีโอ อินพุต ที่ยังไม่ได้ตัด จากนั้นจะแยกคุณลักษณะ ท่าทางและการเคลื่อนไหวที่ซ้อนกัน หรือทำ Normalized Partial Feature (NPF) หลังจากนั้นใช้รูปแบบ CNN-LSTM เพื่อสร้างชุดตัวอย่างข้อมูลทั้งหมด สุดท้าย หากมีล้ม ก็จะส่งข้อมูลที่เป็นตำแหน่งจุดเริ่มต้นของ การล้มและตำแหน่งสุดท้าย ของ วิดีโอออกไป

ปี ค.ศ. 2019 TSUNG-HAN TSAI และ CHIN-WEI HSU [10] ได้นำเสนอ การใช้ระบบตรวจจับการล้มจากภาพโครงกระดูกแบบ 3 มิติสำหรับเทคนิคการเรียนรู้เชิงลึก (Deep Learning) งานวิจัยนี้ได้ นำเสนอ ระบบตรวจจับการล้มจากภาพวิดีโอจากกล้อง Microsoft's Kinect V2 แล้วนำข้อมูลเข้าสู่ระบบ ประมวลผล เพื่อสกัดเอาคุณลักษณะที่เป็นโครงกระดูกจำนวน 7 จุด



ภาพประกอบ 5 การสกัดเอาจุดต่อโครงกระดูก 7 จุด [10]

งานวิจัยนี้ได้เลือกใช้ชุดข้อมูลการล้มของ NTU RGB+D เทรนข้อมูลการล้มโดยโมเดล AlexNet และ Mynet2D ทำงานบนบอร์ด NVIDIA Jetson Tx2 ซึ่งเป็นตัวประมวลผลแบบเรียลไทม์ ซึ่งพบว่าการใช้ชุดข้อมูล MyNet1D-D ได้ค่าความแม่นยำอยู่ที่ 99.2 เปอร์เซ็นต์

ปี ค.ศ. 2020 Wen-Yu Cai, Jia-Hao Guo, Mei-Yan Zhang, Zhi-Xiang Ruan, Xue-Chen Zheng และ Shuai-Shuai Lv [11] ได้นำเสนอ การตรวจจับการล้มโดยใช้รูปแบบ GBDT พร้อมกับข้อมูลที่มาจากการเซ็นเซอร์ท่าทางและการสกัดข้อมูลโครงกระดูกมนุษย์ งานวิจัยนี้ได้นำเสนอระบบตรวจจับการล้มที่ใช้เซ็นเซอร์การเร่งความเร็วแบบสามแกนแบบสวมใส่ได้ในแนวดิ่ง ใช้เพื่อตรวจจับท่าทางของร่างกายและกล้องตรวจสอบเพื่อตรวจจับอุบัติเหตุจากการหกล้ม โดยกล้อง จะทำการตรวจสอบเพื่อดึงข้อมูลจุดสำคัญของโครงกระดูกมนุษย์ใช้อัลกอริทึม GBDT ในการแบ่งกลุ่ม เพื่อตรวจจับการล้ม

ปี ค.ศ. 2021 Xiao L และ Wanmi Chen [12] ได้นำเสนอ อัลกอริทึมการรู้จำ การล้ม สำหรับผู้สูงอายุโดยอาศัยหุ่นยนต์บริการที่บ้าน งานวิจัยนี้เสนอวิธีการตรวจจับการล้ม โดยไม่ได้ตั้งใจ ตามหลักการสมมาตรของหุ่นยนต์บริการที่บ้านโดยจะใช้กล้อง Kinect เป็นตัวถ่ายวิดีโอ แล้วใช้ไลบรารี Open Pose ในการดึงข้อมูลร่างกายมนุษย์ออกมาในลักษณะโครงกระดูก และใช้พารามิเตอร์หลักสี่ตัวเพื่อระบุการล้ม คือ ความเร็วของการล้มที่ศูนย์กลางของข้อต่อสะโพก มุมกึ่งกลางของร่างกายมนุษย์กับพื้นและอัตราส่วนกว้างยาว ลักษณะเป็นกรอบรูปสี่เหลี่ยมผืนผ้า ภายนอกมนุษย์ซึ่งขั้นตอนการทำงานเฉพาะของอัลกอริทึมมีดังนี้ อันดับแรกจะรับข้อมูลจุดต่อโครงกระดูกมนุษย์ผ่าน OpenPose อันดับที่สองคำนวณมุมระหว่างเส้นกึ่งกลางของมนุษย์กับพื้น อันดับที่สองคำนวณมุมระหว่างเส้นกึ่งกลางของมนุษย์กับพื้น อันดับสามคำนวณมุมระหว่างเส้น กึ่งกลางของมนุษย์กับพื้น อันดับสี่คำนวณอัตราส่วนความกว้างต่อความสูงของร่างกายคน อันดับสุดท้ายตรวจสอบว่าบุคคลสามารถยืนขึ้นได้หรือไม่ หุ่นยนต์บริการที่บ้านสามารถยืนยันได้ว่าจำเป็น ต้องมีการโต้ตอบด้วยเสียงวิธีนี้มีอัตรา ความแม่นยำอยู่ที่ 97.3% ในการรับรู้เหตุการณ์การล้ม

ปี ค.ศ. 2021 Divya R, Riya T B , Rona Johns P และ Sreelakshmi T J [13] ได้นำเสนอการตรวจจับการล้มโดยใช้ OpenPose ตรวจจับจุดสำคัญของร่างกายมนุษย์จากเฟรมภาพวิดีโอที่รวบรวมโดยใช้โมดูล TensorFlow OpenPose ในการแยกโครงกระดูกมนุษย์และตรวจจับการหกล้ม โดยใช้ข้อมูลจุดต่อโครงกระดูกแบบจำลองการตรวจจับการล้ม ที่นำเสนอในงานวิจัยนี้ จะเรียนรู้การเปลี่ยนแปลงจุดเชื่อมต่อของโครงกระดูกมนุษย์เมื่อเวลาผ่านไปเมื่อเกิดการล้ม ระบบจะรับรู้ได้โดยการกำหนดเงื่อนไขการเปลี่ยนแปลงค่าสามประการ คือ ความเร็วของการล้มที่ศูนย์กลางของข้อต่อสะโพก มุมระหว่างเส้นกึ่งกลางของร่างกายมนุษย์กับพื้นและค่าอัตราส่วนความกว้างต่อความสูงในลักษณะของกรอบรูปสี่เหลี่ยมผืนผ้าภายนอกของร่างกายคน ผลการทดลองได้แสดงให้เห็นว่าขั้นตอนการทำงานได้ผลและให้ผลลัพธ์ตามที่ต้องการรวมทั้งการแจ้งเตือนที่เหมาะสม

ปี ค.ศ. 2021 Yangsen Chen, Rongxi Du, Kaitao Luo และ Yuheng Xiao [14] ได้นำเสนอระบบตรวจจับการล้มโดยการประเมินท่าทางแบบเรียลไทม์และ SVM งานวิจัยนี้ได้ใช้วิธีการประเมินท่าทางและใช้ yoloV5 เป็นตัวช่วยในการตรวจจับ โดยในขั้นแรกจะทำการแยกเฟรม ภาพวิดีโอการล้มตามลำดับที่มีความแตกต่างกันเพื่อทำการสร้างชุดข้อมูล จากนั้นจะทำการนำข้อมูลตัวอย่างการเทรนเข้าไปยังระบบเพื่อทำการเทรนจนกว่าจะได้ข้อมูลที่เหมาะสม ในขั้นตอนสุดท้าย จะทำการทดสอบข้อมูลจากวิดีโอโมเดลการเรียนรู้ว่าตรงตามเป้าหมายหรือไม่ จากผลการทดลอง แสดงให้เห็นว่า อัลกอริธึมที่ปรับปรุงแล้วสามารถตรวจจับการหกล้มหรือกิจกรรมในชีวิตประจำวัน (ADL) ได้อย่างมีประสิทธิภาพ

ปี ค.ศ. 2019 Guilherme Vieira Leite, Gabriel Pellegrino da Silva และ Helio Pedrini [15] ได้นำเสนอ การตรวจจับการล้มตามลำดับภาพในวิดีโอโดยใช้ Three-Stream Convolutional Neural Network งานวิจัยนี้ได้ใช้วิธีการ multi-stream เพื่อตรวจจับการล้ม ซึ่งแต่ละสตรีมจะมี คุณลักษณะสามประการ คือ optical flow, saliency map, และข้อมูล RGB ที่ส่งไปยังแต่ละสตรีม ของ VGG-16 และแบ่งกลุ่มตามรูปแบบของ Support Vector Machine (SVM) ว่ามีเหตุการณ์การ ล้มหรือไม่ ซึ่งจะทดลองดำเนินการกับชุดข้อมูลสองชุด คือ URFD และ FDD โดยได้ อัตราความแม่นยำ ที่ 98.84% และ 99.51% ตามลำดับ

ปี ค.ศ. 2020 Vincenzo Dentamaro, Donato Impedovo และ Giuseppe Pirlo [16] ได้นำเสนอการตรวจจับการล้มโดยการประมาณการท่าทางของมนุษย์และทฤษฎี Kinematic งานวิจัย นี้ได้นำเสนอระบบสนับสนุนการตัดสินใจ (Decision Support System) ซึ่งไปป์ไลน์ ได้รับการออกแบบมาเพื่อหาคุณลักษณะที่สำคัญและคำนวณสิ่งที่มองเห็นจากกล้องวิดีโอ ใช้ชุดข้อมูล การตรวจจับการล้มของ Le2i และ UR ใช้ทฤษฎี Kinematic Theory ในการคำนวณค่า การเคลื่อน ไหวอย่างรวดเร็วของมนุษย์และใช้โมเดล sigma-lognormal ร่วมกับลักษณะทางกายภาพ แบบ คลาสสิก (classic physical features) ในการตรวจจับการล้มอัตโนมัติในชุดข้อมูล Le2i และ URFD ซึ่งมีความแม่นยำ 98% และ 99% ตามลำดับ

ปี ค.ศ. 2020 Yoon-Kyu Kang, Hee-Yong Kang และ Dal-Soo Weon [17] ได้นำเสนอ การตรวจจับการล้มตามจุดสำคัญของโครงกระดูกมนุษย์โดยใช้ GRU งานวิจัยนี้ ได้ปรับปรุงวิธีการ ในการตรวจหาและ ประเมินตำแหน่งจุดสำคัญที่ศีรษะและไหล่และการเร่งความเร็วของการเปลี่ยนตำแหน่ง โดยใช้ข้อมูลจุดสำคัญของโครงกระดูกที่ดึงออกมาโดยใช้ PoseNet จากภาพที่ได้จากกล้อง 2D RGB และ เพื่อเพิ่มความแม่นยำ ในการตัดสินใจล้มได้เสนอวิธีการตรวจจับการล้มตามลักษณะของท่าทางหลังการล้มในวิธีการวิเคราะห์การเคลื่อนไหวของการล้มและความเร็วของการเปลี่ยนแปลงจุดสำคัญของโครงกระดูกของร่างกายมนุษย์ ตลอดจนการเปลี่ยนแปลงอัตราส่วนของความกว้างและความสูงของกล้องล้อมรอบร่างกายโดยใช้ชุดข้อมูล การตรวจจับการล้มแบบสาธารณะ สองชุดได้แก่

AI HUB และ URFD และเทคนิคเครือข่ายประสาทเทียม GRU สำหรับการจำแนก และการตรวจจับ การล้มในการตรวจจับการเคลื่อนไหวตามวิดีโอ และค้นหาวิธีการ แยกคุณลักษณะ การจำแนก ประเภทความสูงโดยใช้ PoseNet งานวิจัยนี้มีความแม่นยำในการตรวจจับอยู่ที่ 99.8%

ปี ค.ศ. 2020 Bo-Hua Wang, Jie Yu, Kuo Wang, Xuan-Yu Bao และ Ke-Ming Mao [18] ได้นำเสนอการตรวจจับการล้มโดยการรวมคุณสมบัติสองช่องสัญญาณ งานวิจัยนี้ได้นำเสนอ แนวทางการตรวจจับการล้มด้วยภาพแบบใหม่โดยการรวมคุณลักษณะแบบช่องสัญญาณคู่ แนวทางที่ เสนอจะแบ่งเหตุการณ์ล้มออกเป็นสองส่วนคือ falling state และ fallen state ซึ่งจะอธิบาย เหตุการณ์การล้มจากมุมมองแบบเคลื่อนที่และแบบคงที่ ขั้นตอนแรกทำการตรวจจับวัตถุโดยใช้โมเดล Yolo และโมเดลการตรวจจับท่าทางของมนุษย์จะใช้โมเดล OpenPose ใช้สำหรับการประมวลผล เพื่อให้ได้จุดสำคัญและข้อมูลตำแหน่งของร่างกายมนุษย์ จากนั้น Dual Channel เลื่อนตำแหน่งช่อง ของภาพออกไปเรื่อยๆ เพื่อแยกคุณลักษณะการเคลื่อนที่ของร่างกายมนุษย์ (centroid speed, upper limb velocity) และคุณลักษณะคงที่ (human external ellipse) หลังจากนั้น MLP (Multilayer Perceptron) และ Random Forest จะถูกนำไปใช้เพื่อแยกประเภทข้อมูลคุณสมบัติ การเคลื่อนที่และคงที่แยกกันผลการจำแนกประเภทจะถูกรวมไว้สำหรับการตรวจจับการล้ม ผลการ ทดลองแสดงให้เห็นว่าวิธีการที่เสนอมีความแม่นยำ 97.33% และ 96.91% เมื่อทดสอบด้วยชุดข้อมูล การตรวจจับการล้ม UR และชุดข้อมูลการตรวจจับการล้ม Le2i

ปี ค.ศ. 2019 Yin Zheng, Dongping Zhang, Li Yanga และ Zhihong Zhou [19] ได้ นำเสนอการตรวจจับการล้มและการรู้จำด้วย GCN และ 2D Pose งานวิจัยนี้จะทำงาน บนคุณสมบัติ การเคลื่อนที่ของโครงกระดูกมนุษย์ โครงข่ายประสาทเทียมแบบ graph convolution network (GCN) และข้อมูลท่าทาง แบบ 2D ถูกนำมาใช้เพื่อดำเนินการทางด้าน deep learning และการ แบ่งกลุ่มตามเหตุการณ์เป้าหมายเพื่อตรวจจับการหกล้มใช้ GPU การเร่งความเร็วสำหรับ การทดสอบ บนแพลตฟอร์ม ubuntu ด้วยความเร็วเฉลี่ยที่ 25 fps ขั้นแรกจะนำภาพวิดีโอ หรือประมาณการ ท่าทางของมนุษย์เข้าสู่ระบบ ประการที่สองสกัดคุณลักษณะท่าทางของมนุษย์ ประการที่สาม คุณลักษณะท่าทางของมนุษย์ที่สกัดออกมาจะถูกป้อนลงใน โครงข่ายประสาทเทียม GCN เพื่อแยก คุณลักษณะ อันดับสุดท้ายทำการแบ่งกลุ่มเพื่อดำเนินการตรวจสอบว่าเป็นการล้ม หรือไม่ตามผลลัพธ์ เมื่อเทียบกับวิธีอื่นๆ วิธีการประมาณท่าทางจากข้อมูลโครงกระดูก 2 มิติ รวมกับ วิธี GCN ค่อนข้าง ดีที่สุดมีความแม่นยำ 100 เปอร์เซนต์

ปี ค.ศ. 2020 Guangmin Sun และ Zhongqi Wang [20] ได้นำเสนออัลกอริธึม การ ตรวจจับการล้มสำหรับผู้สูงอายุตามการประเมินท่าทางของมนุษย์ งานวิจัยนี้มีการเสนอรูปแบบ การ ตรวจจับการล้มด้วยอัลกอริธึมการประเมินท่าทางของมนุษย์ด้วย OpenPose โดยใช้ computer visoin การตรวจจับ จุดสำคัญของมนุษย์ของ OpenPose เมื่อรวมกับการตรวจจับวัตถุด้วย SSD

MobileNet สามารถลบจุดสำคัญ ที่ไม่ใช่ของมนุษย์ที่ตรวจพบโดยอัลกอริทึม OpenPose ลดอัตรา การตรวจจับที่ผิดพลาดของอัลกอริทึม ปรับปรุงความแข็งแกร่งของอัลกอริทึมในสภาพ แวด ล้อมที่ ซับซ้อนจากนั้นทำการสกัดเอาคุณสมบัติที่สำคัญของจุดต่างๆของมนุษย์ ใช้อัลกอริทึมการแบ่ง กลุ่ม SVDD การทดลองแสดงให้เห็นว่าวิธีนี้สามารถตรวจจับการล้มได้อย่างมีประสิทธิภาพและอัตรา ความ แม่นยำ 93 เปอร์เซ็นต์

ปี ค.ศ. 2020 Xiangbo Konga, Takeshi Kumakia, Lin Menga และ Hiroyuki Tomiyamaa [21] ได้นำเสนอวิธีการตรวจจับการล้มด้วยการวิเคราะห์โครงกระดูกโดยใช้กล้อง ToF งานวิจัยนี้ใช้กล้อง ToF, Microsoft Kinect V2 ,ใช้ภาษา C++ Visual Studio 2019 ในการเขียน โปรแกรม, OpenCV 4.4 และ Kinect SDK เวอร์ชัน 1409, กล้องเชื่อมต่อ กับคอมพิวเตอร์โน้ตบุ๊ก CPU โดยใช้ Intel(R) Core i7-10510U หน่วยความจำ 16GB และระบบปฏิบัติการ Windows 10 Home อัตราเฟรมอยู่ที่ 30-60 fps โดยไม่ใช้ GPU งานวิจัยนี้ได้นำเสนอการแยกแยะระหว่างการ หกล้มและการนอน จะใช้วิธีการคำนวณระยะทางจาก ศีรษะถึงพื้น จากนั้นก็คำนวณความเร็วของ ศีรษะ เมื่อเกิดอุบัติเหตุจากการล้มและการนอนบนพื้น ใน ขณะที่บุคคลประสบอุบัติเหตุจากการหกล้ม หัว กระแทกพื้นภายใน 50 เฟรม และเมื่อบุคคลไปนอน กับพื้นจะใช้เวลามากกว่า 200 เฟรม เมื่อศีรษะ อยู่ใกล้กับพื้น นอกจากนี้ความเร็วในการเคลื่อนที่ ช่วงขณะจะมากกว่า 1,000 มม./วินาที ในช่วงของ การล้ม ดังนั้นค่า1,000 จึงถูกกำหนดเป็นเกณฑ์ ในระบบนี้เมื่อศีรษะของบุคคลอยู่ใกล้กับ พื้นและจาก การทดลองจะสังเกตเห็นว่าเมื่อเกิดการล้มความเร็วที่ได้จะเกิน 1,000 มม./วินาที ต่อจากนั้นก็จะมี การแจ้งเตือนไปยังแพทย์หรือสมาชิกในครอบครัว เพื่อแจ้งให้ทราบว่าผู้สูงอายุอาจ อยู่ในอันตราย

ปี 2021 Cuiwei Liu, Jianxiong Lv, Xiaoxue Zhao, Zhaokui Li, Zhuo Yan และ Xiangbin Shi [22] ได้นำเสนอการตรวจจับการล้มจากวิดีโอ RGB-D ด้วยโมเดล Key Point Trajectory งานวิจัยนี้เสนอรูปแบบ Key Point Trajectory Model (KPTM) ที่แสดงลักษณะเฉพาะ ของภาพ RGB-D ด้วยชุดของตัวอธิบาย trajectory และทำการตัดสินใจว่าล้มหรือไม่h,โดยใช้โมเดล random forest classifier เป็นตัวแบ่งกลุ่มพร้อมกับกฎการรวม KPTM ที่นำเสนอประกอบด้วย 3 องค์ประกอบหลักคือ การแยกประเด็นสำคัญ การสร้างตัวอธิบาย trajectory และ ตัวแบ่งกลุ่ม ผล การตรวจจับการล้มโดยใช้ ชุดข้อมูล UR Fall Dataset ได้ความแม่นยำที่ 100 เปอร์เซ็นต์ ส่วนชุด ข้อมูลของ SDU Fall Dataset ได้ความแม่นยำที่ 98.30 เปอร์เซ็นต์

ปี ค.ศ. 2021 Yves M. Galvao, Janderson Ferreira, Vinicius A. Albuquerque, Pablo Barros และ Bruno J.T. Fernandes [23] ได้นำเสนอวิธีการหลายรูปแบบ (Multimodal) โดยใช้ การเรียนรู้เชิงลึกสำหรับการตรวจจับการล้ม เอกสารนี้แสดงวิธีแก้ปัญหาที่เป็นไปได้ในการ ตรวจจับ การหกล้ม โดยทดสอบโมเดลการเรียนรู้เชิงลึกที่แตกต่างกันในชุดข้อมูล UR Fall Detection และชุด

ข้อมูลการตรวจจับ UP-Fall โมเดลที่ได้รับผลลัพธ์ที่ดีที่สุดคือ CNN เพื่อจำแนกการล้ม ในชุดข้อมูลหลายรูปแบบผลลัพธ์ที่ได้แสดงให้เห็นว่าวิธีการหลายรูปแบบ (Multimodal) ที่นำเสนอ มีความถูกต้องถึง 99.87% สำหรับ UR Fall และ 99.99% สำหรับชุดข้อมูล UP-Fall โดยมีแหล่งข้อมูลการได้มาสองแห่ง (รูปภาพและมาตรความเร่ง) ที่มีเครือข่ายสองเครือข่ายที่แตกต่างกัน ได้แก่ CNN2D และ CNN1D หรือ LSTM แล้วสกัดคุณลักษณะแยกออกมา (ค่าของเอาต์พุตของเลเยอร์สุดท้ายของ CNN2D และ CNN1D หรือ LSTM) จะถูกแบ่งกลุ่มโดยเครือข่าย CNN1D เพื่อระบุว่าเกิดการล้มหรือไม่

ปี ค.ศ. 2014 Zhen-Peng Bian, Junhui Hou, Lap-Pui Chau, Senior และ Nadia Magnenat-Thalmann [24] ได้นำเสนอ การตรวจจับการล้มตามการติดตามส่วนของร่างกาย โดยใช้กล้อง Depth Camera ที่ใช้อินฟราเรด กล้องสามารถวัดการเคลื่อนไหวของร่างกายมนุษย์ และความสัมพันธ์ระหว่างร่างกายกับสิ่งแวดล้อม รูปแบบที่เสนอนั้นไม่ขึ้นอยู่กับแสงสว่างของไฟ และสามารถทำงานได้แม้ในห้องมืด มีการนำเสนออัลกอริธึม Randomized Decision Tree (RDT) สำหรับการสกัดคุณลักษณะที่สำคัญของจุดข้อต่อ ซึ่งใช้การคำนวณต่ำในระหว่างการเทรนและการทดสอบ จากนั้นจึงโมเดล Support Vector Machine (SVM) เพื่อพิจารณาว่ามีการล้มหรือไม่ ซึ่งมีความแม่นยำอยู่ที่ 97.79 เปอร์เซ็นต์

ปี ค.ศ. 2020 Keondo Park, Wonyoung Jang, Woochul Lee, Kisung Nam, Kihong Seong, Kyuwook Chai และ Wen-Syan Li [25] ได้นำเสนอการตรวจจับหน้ากาก แบบเรียลไทม์บน Google Edge TPU งานวิจัยนี้ ได้นำเสนอโมเดลขนาดเล็กสำหรับการตรวจจับว่า ผู้คนในพื้นที่ได้สวมหน้ากากหรือไม่ซึ่งสามารถนำไปใช้กับ Coral Dev Board ซึ่งเป็นบอร์ดพัฒนา ที่มีจำหน่ายทั่วไป ซึ่งมี Google Edge TPU ได้รวมการตรวจจับอ็อบเจกต์โดยใช้ MobileNetV2plus SSD และข้อมูลที่เป็นค่าจำนวนเต็มเท่านั้นสำหรับ ฮาร์ดแวร์ เป็นผลทำให้ lighter โมเดลใน Edge TPU มีเวลาทำงานที่ต่ำกว่ามาก ซึ่งเหมาะสำหรับการใช้งาน แบบเรียลไทม์ ในขณะที่ยังคงความ แม่นยำเทียบเท่ากับอุปกรณ์ที่คำนวณตัวเลขแบบจุดทศนิยม งานวิจัยนี้ใช้โครงสร้างโครงข่ายประสาทเทียม 2 รูปแบบ คือ โมเดล 2NN โมเดลนี้ใช้โครงข่ายประสาทเทียมสองโครง ข่าย โครงข่ายหนึ่งสำหรับการตรวจจับและอีกโครงข่ายสำหรับการจำแนกประเภท โดยใช้ MobileNetV2 plus SSD เพื่อตรวจจับใบหน้า และ MobileNetV2 เพื่อจำแนกใบหน้าที่ตรวจพบว่ามีหน้ากาก หรือไม่มีหน้ากาก งานวิจัยนี้ได้เทรนโมเดลการตรวจจับโดยใช้แบบจำลองที่ได้รับเทรนมาแล้วจาก tensorflow model zoo และได้ทำการเทรนแบ่งกลุ่มโดยวิธีการ transfer learning จาก MobileNetV2 ในการปรับปรุงประสิทธิภาพของโมเดลจึงได้รวมโมเดลการจัดหมวดหมู่ไว้ ในโมเดล การตรวจจับซึ่งโมเดลอันแรกคือ 1NN โดยใช้ MobileNetV2 plus SSD ซึ่งชุดข้อมูลที่ใช้ที่มีจำนวน ภาพค่อนข้างน้อย ดังนั้นจึงใช้วิธีการ Transfer Learning จากโมเดลการตรวจจับใบหน้า ที่ได้รับการ เทรนมาก่อนแล้ว โดยได้นำโมเดล

การตรวจจับใบหน้าจากTensorflow 1 Object Detection Model Zoo และใช้โมเดลนี้เป็นโมเดลพื้นฐานก่อนที่จะเริ่มต้นการเทรน ได้ทำการเปลี่ยนแปลงบาง อย่างบนโมเดลพื้นฐานเพื่อการทำ transfer learning ที่ดีขึ้นเนื่องจากโมเดลการตรวจจับใบหน้า มีเพียงหนึ่งคลาส (face) แต่โมเดลในงานวิจัยนี้จำเป็นต้องมีสองคลาส (Mask, NoMask) จึงได้ ขยายเลเยอร์การทำนายคลาสให้มี 2 แบบ สำหรับโมเดลที่สองคือ โมเดล 2NN ประกอบด้วย MobileNet v2 สำหรับการแบ่งกลุ่ม รวมกับค่า Average Pooling, Flatten และ Dense Layers การทดลองนี้ได้ทำการเปรียบเทียบความแม่นยำ ทั้งสองโมเดลโดยได้ผลตามตาราง 2

ตาราง 2 ความแม่นยำระหว่างโมเดล 1NN และ 2NN โดยใช้ CPU และ Edge TPU [25]

Environment	Data Type	1NN	2NN
CPU	32 Bit	64.7%	-
CPU	8 Bit	58.8%	55.0%
Edge TPU	8 Bit	58.4%	54.8%

ปี 2020 Ahmad Ammar Asyraf Jainuddin, Yew Cheong Hou, Mohd Zafri Baharuddin และ Salman Yussof [26] ได้นำเสนอ การวิเคราะห์ประสิทธิภาพของ Deep Neural Networks สำหรับ การจำแนกวัตถุด้วย Edge TPU งานวิจัยนี้มีจุดมุ่งหมายเพื่อให้การวิเคราะห์ ประสิทธิภาพบนฮาร์ดแวร์การเรียนรู้ของ Google ที่เรียกว่า Edge TPU ซึ่งถูกสร้างขึ้นสำหรับ อุปกรณ์ Edge โดยเฉพาะ นอกจากนี้ยังตรวจสอบโมเดล Deep Neural Network ประเภทต่างๆ เนื่องจากเกณฑ์มาตรฐานในปัจจุบันในการเรียนรู้แบบ Deep learning ได้รับการทดสอบด้วย ฮาร์ดแวร์ต่างๆ ที่ใช้ในแอปพลิเคชันบนอุปกรณ์ปลายทางและ ยังทำการเปรียบเทียบประสิทธิภาพ ของอุปกรณ์ Edge โดยใช้ Deep Neural Network ใน Tensorflow จากผลลัพธ์ที่ได้รับ ประสิทธิภาพของอุปกรณ์ Edge ที่มี Edge TPU นั้นเร็วกว่าอุปกรณ์ที่ไม่มี Edge TPU

งานวิจัยนี้ได้โมเดล Deep Learning เท่านั้นในการทดสอบ คือ MobileNet V1, MobileNet V2, EfficientNet-EdgeTPU (S), Inception V1, Inception V4 ซึ่งทุกโมเดลจะมีภาพ ด้วยขนาดอินพุต 224×224 ยกเว้น Inception V4 ที่มีขนาดอินพุต 299×299 เท่านั้น โมเดลเหล่านี้ ได้รับการเทรนด้วย ชุดข้อมูลของ ILSVRC2012 เพื่อจำแนกวัตถุ 1,000 ประเภทอุปกรณ์ Edge ที่ เลือก ได้แก่ Raspberry Pi 3 B+, Raspberry Pi 4 B พร้อม ram 1GB, Raspberry Pi 4 B พร้อม ram 4 GB และ Intel NUC พร้อมโปรเซสเซอร์ I7 ฮาร์ดแวร์ ที่อยู่ในรายการทั้งหมดมีการเชื่อมต่อ

พอร์ต USB 3.0 ยกเว้น Raspberry Pi 3 B+ ได้ใช้การถ่ายโอน ข้อมูลจากฮาร์ดแวร์ไปยัง Edge TPU เป็น ผลิตภัณฑ์ของ Google ที่เผยแพร่ภายใต้ชื่อ Coral เป็นฮาร์ดแวร์ที่สามารถใช้ Edge TPU ได้ อย่าง อิสระ ซึ่งผลจากการทดลองได้ค่าคะแนน ดังตาราง 3

ตาราง 3 คะแนนการเปรียบเทียบกับภาพทดสอบและกล้องโดยใช้ Edge TPU API [26]

Hardware	I7NUC	Pi 4B(4GB)	Pi 4B(1GB)	Pi 3B+
Edge TPU Image	11.87	9.23	9.40	4.35
Edge TPU Camera	625.56	515.85	518.05	115.91

ปี 2020 Jonah Sengupta, Rajkumar Kubendran, Emre Neftci และ Andreas Andreo [27] ได้นำเสนอ การใช้ Google Edge TPU ทดสอบ High-Speed, Real-Time, Spike-Based ObjectTracking and Path Prediction งานวิจัยนี้ได้ปรับปรุงความซ้ำซ้อนของข้อมูล ขนาดใหญ่ที่พบในแอปพลิเคชัน Machine Learning โดยการสาธิตการเชื่อมโยงเทคโนโลยี Dynamic Vision Sensor (DVS) และ Google Edge TPU ทั้งสองนี้เข้ากับเกณฑ์มาตรฐานของ Coral Edge TPU Board ในแอปพลิเคชันการติดตามและการทำนายวัตถุความเร็วสูงเมื่อเปรียบ เทียบกับ สถาปัตยกรรมแบบ floating point โดยใช้ Intel Compute Stick นั้น Edge TPU มีประสิทธิภาพเหนือกว่าในแง่ของค่าเวลา latency และประสิทธิภาพในการคำนวณ

ปี ค.ศ. 2018 Y. Hsieh และ Y. Jeng [28] ได้นำเสนอการพัฒนาระบบ IoT ตรวจสอบการล้มแบบอัจฉริยะภายในบ้านโดยใช้โมเดล Feedback Optical Flow Convolutional Neural Network งานวิจัยนี้ได้แนะนำโครงข่ายประสาทเทียมแบบ optical flow feedback convolutional neural network โดยใช้กล้องแบบ RGB ทำการสตรีมวิดีโอในสภาพแวดล้อม ที่บ้านใช้โมเดล rule-based ทำการ กรองภาพก่อนที่จะนำข้อมูลเข้าไปยังชั้นของ convolutional layer และบันทึกการตำแหน่งการเคลื่อนไหว ของภาพ (optical flow) ในหลายๆ ตำแหน่งที่มี การเปลี่ยนแปลง ในการทดลองนี้สามารถตรวจจับท่าอนปกติ และการนอนหลังจากล้ม ได้อย่างชัดเจน วิธีการที่เสนอนี้สามารถตรวจจับการเคลื่อนไหวของการกระทำได้อย่างมีประสิทธิภาพ และรับรู้ท่าทางการกระทำซึ่งได้ทำการเปรียบเทียบประสิทธิภาพกับชุดข้อมูลมาตรฐานอื่นๆ และปรับใช้กับโมเดลที่ได้สร้างขึ้นมาเพื่อจำลองสถานการณ์ในบ้าน เมื่อเปรียบเทียบประสิทธิภาพกับชุดข้อมูลมาตรฐานอื่นๆและใช้แบบจำลองจากงานวิจัยนี้เพื่อจำลองสถานการณ์ในบ้าน ซึ่งมีค่าความแม่นยำอยู่ที่ 92.65 เปอร์เซ็นต์

ปี 2020 A. Dragomir, G. Simion และ I. Ermai [29] ได้นำเสนอโครงข่ายประสาทเทียม สำหรับ การระบุประเภทผิว โดยทำงานบน Raspberry Pi 4 model B เชื่อมต่อกับ Coral USB Accelerator งานวิจัยนี้ได้รวบรวมรูปภาพจำนวน 140 รูปจาก Google จำนวน 35 รูปสำหรับแต่ละ Class ขนาดของรูปภาพแตกต่างกันทั้งภายในคลาสและระหว่างคลาสมิการใช้รูปภาพทั้งหมด 20 ภาพ ในการทดสอบในขณะที่ภาพที่เหลือใช้สำหรับการเทรนค่าสำหรับ epoch ตั้งค่าเป็น 100, ขนาด batch size ตั้งค่าเป็น 16, ส่วนค่าอัตราการเรียนรู้ (Learning Rate) เป็น 0.001 โดยการเทรน ข้อมูลจะใช้ Teachable Machine จากเว็บไซต์ ต่อจากนั้นก็ทำการ โหลดโมเดลและทดสอบ บน บอร์ด Raspberry Pi 4 ซึ่งทำงานร่วมกับ Edge TPU ในการทดสอบโมเดลนั้นสามารถใช้รูปภาพ จากไฟล์หรือจากเว็บแคมได้หรือพิมพ์ภาพแล้วแสดงทางกล้อง

ตาราง 4 ผลจากการทดลอง [29]

Class	Accuracy	Trust Degree
Acne	1	>0.58
Blackheads	1	0.99
Freckles	1	0.98
Normal	1	0.99

ปี ค.ศ. 2021 S. Hosseininoorbin, S. Layeghy, B. Kusy, R. Jurdak และ M. Portmann [30] ได้นำเสนอ การแสดงข้อมูลสเปกโตรแกรมโดยใช้ Deep Learning ทำงานบน Edge TPU งานวิจัยนี้ได้เลือกใช้ Edge TPU ของ Google ซึ่งเป็น ASIC ที่สร้างขึ้นโดยเฉพาะซึ่งถูก ออกแบบมาเพื่อใช้งานทางด้านปัญญาประดิษฐ์ (AI) ที่ทำงานเป็นแบบอุปกรณ์ ประมวลผลปลายทาง (Edge Computing) ในการทดลองนี้ได้แบ่งประเภทกิจกรรมการเลี้ยงโคแบบ อัตโนมัติ ซึ่งต้องมีการ อนุมาณเพื่อทำงานบนอุปกรณ์ฝังตัวที่มีพลังงาน จำกัด ใช้ deep neural network ในการแยก ประเภท ซึ่งโดยปกติแล้วการทำงานบนอุปกรณ์ Edge จะมีข้อจำกัดด้าน ทรัพยากร อย่างมาก จาก ชุดข้อมูลกิจกรรมปศุสัตว์จริงและด้วยการใช้การแสดงผล joint-time-frequency (สเปกโตรแกรม) ได้สำรวจความแตกต่างที่แตกต่างกันระหว่างความถูกต้องของการ จำแนกประเภท และ ประสิทธิภาพการใช้พลังงานผลลัพธ์แสดงให้เห็นว่า Edge TPU สามารถให้ทั้ง ประสิทธิภาพ การ จำแนกประเภทที่ยอดเยี่ยม และประสิทธิภาพการใช้พลังงานสเปกโตรแกรมที่ใช้ ในงานวิจัยนี้ ได้

ปรับแต่งเพื่อ Edge TPU โดยเฉพาะ ซึ่งสามารถให้ความแม่นยำในการจำแนกประเภทมีค่าสูง และสิ้นเปลืองพลังงานน้อยที่สุด

ปี ค.ศ. 2021 Seyedehfaezeh Hosseininoorbin, Siamak Layeghya, Mohanad Sarhana, Raja Jurdakc และ Marius Portmanna [31] ได้นำเสนอการใช้ Edge TPU สำหรับการตรวจจับการบุกรุก (NIDS) ในเครือข่าย IoT โดยใช้ deep learning ที่อุปกรณ์ปลายทาง ของระบบ IoT ซึ่งได้ทำการทดลองหาประสิทธิภาพด้านการคำนวณและพลังงานของ Edge TPU ซึ่งเป็นของ Google เป็นแพลตฟอร์มฮาร์ดแวร์และพิจารณาตัวชี้วัดหลักสามตัวต่อไปนี้ คือ เวลาในการคำนวณ (การอนุมาน) ประสิทธิภาพการใช้พลังงานและประสิทธิภาพการแบ่งกลุ่มการรับส่งข้อมูล ประสิทธิภาพของการใช้งาน Edge TPU นั้นถูกนำมาเปรียบเทียบกับ CPU แบบฝังตัวที่ประหยัดพลังงาน (ARM Cortex A53) แพลตฟอร์ม Edge TPU เป็นตัวเร่ง ฮาร์ดแวร์ ASIC ที่สร้างขึ้น โดยเฉพาะซึ่งเพิ่งพัฒนาโดย Google เพื่อเรียกใช้การอนุมานที่อุปกรณ์ประมวลผล ปลายทาง (Edge) ตัวเร่งฮาร์ดแวร์ถูกสร้างขึ้นเพื่อเพิ่มความสามารถในการคำนวณของระบบและทำให้สามารถใช้ อัลกอริธึม Deep Learning บนอุปกรณ์ Edge อัลกอริธึม NIDS ที่ใช้ในการสำรวจนี้ได้เลือก deep neural network สองสถาปัตยกรรม ได้แก่ feed forward และ convolutional neural networks. การทดลองแสดงให้เห็นว่า CPU ARM Cortex A53 มีประสิทธิภาพเหนือกว่า Edge TPU อย่างมาก สำหรับโครงข่ายประสาทเทียมขนาดเล็ก

ปี 2020 Kraft, Dimitri, Karthik Srinivasan และ Gerald Bieber [32] ได้นำเสนอ อัลกอริธึมการตรวจจับการล้มจาก Deep Learning สำหรับระบบสมองกลฝังตัว สมาร์ทวอตช์ และ อุปกรณ์ IoT โดยใช้ มาตรวัดความเร่ง (Accelerometers) ในงานวิจัยนี้ ได้นำเสนอภาพรวม เกี่ยวกับฐานข้อมูลสาธารณะที่มีอยู่ ด้วยชุดข้อมูลการล้มจากเซ็นเซอร์และประสานชุดข้อมูลที่มีอยู่บน สายรัดข้อมือเพื่อการประเมินที่กว้างขึ้นและมีประสิทธิภาพและทำการวิเคราะห์อัตราการเรียนรู้ที่เป็นไปได้ในปัจจุบันของการตรวจจับการล้มโดยใช้อัลกอริธึม Deep Learning สำหรับมือถือ และระบบ สมองกลฝังตัว ผลลัพธ์และฐานข้อมูลที่น่าเสนอสามารถใช้สำหรับการวิจัยเพิ่มเติมและเพิ่มประสิทธิภาพ เพื่อเพิ่มอัตราการเรียนรู้การล้ม

ตาราง 5 คะแนน F1 โดยไม่มีข้อมูลเสริม (Data Augmentation) [32]

Dataset	Our				LSTM				Santos				ResNet			
	SMV		3 Axis		SMV		3 Axis		SMV		3 Axis		SMV		3 Axis	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Notch	0.98	0.01	0.97	0.03	1.00	0.00	0.95	0.01	0.92	0.04	0.94	0.01	0.77	0.14	0.91	0.05
MUMA	0.92	0.00	0.80	0.04	0.92	0.01	0.75	0.04	0.91	0.01	0.91	0.01	0.90	0.01	0.89	0.02
Sim Fall	0.89	0.00	0.83	0.04	0.88	0.02	0.80	0.12	0.88	0.01	0.89	0.00	0.90	0.01	0.89	0.01
SmartWatch	0.99	0.00	1.00	0.00	1.00	0.00	0.95	0.04	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
SmartFall	0.99	0.00	1.00	0.00	0.99	0.00	0.92	0.02	1.00	0.00	1.00	0.00	0.99	0.00	1.00	0.00
UP Fall	0.98	0.00	0.90	0.02	0.98	0.00	0.78	0.07	0.99	0.00	0.97	0.00	0.99	0.00	0.96	0.02
Average	0.96	0.00	0.91	0.02	0.96	0.01	0.86	0.05	0.95	0.01	0.95	0.00	0.93	0.03	0.94	0.02

ตาราง 6 คะแนน F1 โดยมีข้อมูลเสริม (Data Augmentation) [32]

Dataset	SMV		Our		LSTM		3 Axis		Santos		3 Axis		ResNet		3 Axis	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Notch	0.97	0.01	0.95	0.03	1.00	0.00	0.93	0.04	1.00	0.00	0.96	0.01	0.97	0.03	0.93	0.01
MUMA	0.93	0.01	0.94	0.01	0.91	0.01	0.84	0.06	0.92	0.00	0.92	0.00	0.92	0.01	0.94	0.01
Sim Fall	0.93	0.00	0.90	0.01	0.93	0.01	0.81	0.10	0.89	0.00	0.90	0.00	0.94	0.00	0.90	0.01
SmartWatch	0.99	0.00	1.00	0.00	0.99	0.00	0.95	0.03	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
SmartFall	0.99	0.00	1.00	0.00	0.98	0.00	0.92	0.03	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
UP Fall	0.98	0.00	0.99	0.00	0.97	0.00	0.91	0.05	0.98	0.00	0.99	0.00	0.98	0.00	0.99	0.00
Average	0.97	0.00	0.96	0.01	0.96	0.01	0.89	0.05	0.97	0.00	0.96	0.00	0.97	0.01	0.96	0.01

ปี 2019 Dharmitha Ajerla, Sazia Mahfuz และ Farhana Zulkernine [33] ได้นำเสนอ เฟรมเวิร์กการตรวจสอบผู้ป่วยตามเวลาจริงสำหรับการตรวจจับการล้ม ที่ใช้ Edge Computing ซึ่ง แทนที่ จะส่งข้อมูลไปยังคลาวด์ อุปกรณ์ที่สวมใส่จะส่งข้อมูลไปยังอุปกรณ์ Edge ที่อยู่ใกล้เคียง เช่น แล็ปท็อปหรืออุปกรณ์พกพาสำหรับการวิเคราะห์แบบเรียลไทม์ใช้อุปกรณ์เซ็นเซอร์แบบสวมใส่ จาก MbitentLab ซึ่งเป็นเอ็นจินการสตรีมโอเพ่นซอร์สที่เรียกว่า Apache Flink สำหรับการวิเคราะห์ ข้อมูลการสตรีมและโมเดล Long Short Term Memory (LSTM) สำหรับการวิเคราะห์การล้ม โมเดลได้รับการเทรนโดยใช้ชุดข้อมูลที่เผยแพร่ชื่อ "MobiAct" จะวิเคราะห์อัตราการล้มตัวอย่าง ที่ เหมาะสมการจัดวางเซ็นเซอร์และการแก้ไขข้อมูลแบบหลายสตรีมการประมวลผลโดยอุปกรณ์ ปลายทาง สามารถวิเคราะห์ข้อมูลการสตรีมแบบเรียลไทม์เพื่อตรวจจับการล้มโดยมีความแม่นยำอยู่ที่ 95.8 เปอร์เซ็นต์

ปี ค.ศ. 2019 Jun Xu, Zunwen He และ Yan Zhang [34] ได้นำเสนอ การรวมโครงข่าย CNN-LSTM เข้าด้วยกันสำหรับการตรวจจับการล้มในระบบ IoT ในบทความนี้มีการเสนออัลกอริธึม การตรวจจับการล้ม Convolutional Neural Network (CNN) ร่วมกับ Long Short Term Memory (LSTM) ข้อมูลจะได้จากจากเซ็นเซอร์การเร่งความเร็วแบบสามแกน (three-axis acceleration sensor) ผลการทดลองแสดงให้เห็นว่าเมื่อเปรียบเทียบกับอัลกอริธึมที่ใช้ Support Vector Machine (SVM) และ CNN แล้วอัลกอริธึมที่นำเสนอมีความแม่นยำในการตรวจจับ สูงกว่า ด้วยปริมาณข้อมูลเพียงเล็กน้อยซึ่งเหมาะมากสำหรับการใช้งานตรวจจับการล้ม ทางด้าน Internet of Things (IoT)

ปี ค.ศ. 2017 A. H. Fakhrulddin และ X. Fei and H. Li [35] ได้นำเสนอ การใช้ Convolutional Neural Networks (CNN) ตรวจจับการล้มของมนุษย์โดยใช้การสวมใส่ตัวตรวจจับ สัญญาณที่ร่างกาย (Body Sensor Networks : BSN) งานวิจัยนี้มีจุดมุ่งหมายเพื่อใช้รูปแบบของ Convolutional Neural Networks (CNN) ในการตรวจจับการล้มของมนุษย์ โดยการใช้วิธีการ CNN กับการสตรีมข้อมูลเซ็นเซอร์ที่รวบรวมจาก Body Sensor Networks (BSN) เพื่อปรับปรุงความ

แม่นยำในการตรวจจับการล้มเมื่อได้ชุดข้อมูลสตรีมเป็นอินพุตแล้ว ต่อจากนั้นก็แปลงเป็นรูปภาพก่อนใช้ CNN ความแม่นยำเมื่อเทียบกับวิธีการอื่นได้เท่ากับ 92.3 เปอร์เซนต์

ปี ค.ศ. 2018 Na Lu, Yidan Wu, Li Feng, Jinbo Song [36] ได้นำเสนอการเรียนรู้เชิงลึกสำหรับการตรวจจับการล้มโดยใช้รูปแบบ 3D-CNN ร่วมกับ LSTM บน Video Kinematic Data งานวิจัยนี้ได้แนะนำวิธีการ three-dimensional convolutional neural network (3D CNN) สำหรับการตรวจจับการล้มซึ่งได้รับการพัฒนาจากข้อมูลจลนศาสตร์ของวิดีโอเพื่อทราบการ สกัทยาคุณลักษณะที่สำคัญโอดและสามารถหลีกเลี่ยงการใช้ชุดข้อมูลการล้มขนาดใหญ่ของ deep learning ได้สำหรับ 2D CNN สามารถเข้ารหัสข้อมูลเชิงพื้นที่ได้เท่านั้นแต่ 3D CNN ที่ใช้สามารถ ดึงคุณลักษณะการเคลื่อนไหวออกจากลำดับเวลาซึ่งเป็นสิ่งสำคัญสำหรับการตรวจจับ การล้มเพื่อระบุตำแหน่งที่น่าสนใจในแต่ละเฟรมจึงมีการรวม LSTM (Long Short-Term Memory) กับการมองเห็นเชิงพื้นที่ (spatial visual attention scheme) ทำงานร่วมกัน โดยได้ใช้ชุดข้อมูล Sports-1M ที่ไม่มีตัวอย่างการล้มนำมาใช้ในการฝึก 3D CNN จากนั้นจึงรวมเข้ากับ LSTM เพื่อสกัดคุณลักษณะ ที่สำคัญด้วยชุดข้อมูลการล้ม การทดลองมีความแม่นยำสูงถึง 100 เปอร์เซนต์ นอกจากนี้ยังได้รับประสิทธิภาพที่เหนือกว่าในฐานะข้อมูลกิจกรรมอื่นๆ

ปี ค.ศ. 2021 Julien Maitre, Kevin Bouchard และ Sebastien Gaboury [37] ได้นำเสนอการตรวจจับการล้มด้วยเรดาร์ UWB ด้วยสถาปัตยกรรม CNN-LSTM งานวิจัยนี้ได้ใช้วิธีตรวจจับการหกล้มในอพาร์ทเมนต์จริงที่มีพื้นที่ 40 ตารางเมตร โดยใช้เรดาร์อัลตราไวด์แบนด์ สามตัวและโมเดล deep neural network ประกอบด้วย convolutional neural network stacked และ long-short term memory network ผ่านไปยังชั้น fully connected neural network เพื่อระบุว่าล้มหรือไม่ ซึ่งในกรณีจริง สามารถ สังเกตได้ว่าการล้มมีรูปแบบที่แตกต่างกัน ดังนั้น ข้อมูลในการเทรนและทดสอบรูปแบบการแบ่งกลุ่มการล้มจึงถูกแบ่งออกเป็น 4 ประเภท จำลองโดยผู้เข้าร่วม 10 คน ใน 3 อพาร์ทเมนต์ ขั้นตอนการทดสอบมีความแม่นยำเกือบ 90%

ปี ค.ศ. 2021 Jiang Wu, Jiale Wang, Ao Zhan และ Chengyu Wu [38] ได้นำเสนอ การตรวจจับการล้มด้วย CNN-Casual LSTM Network ทำงานบนเครื่องตรวจจับการเร่งความเร็ว แบบสามแกน (three-axis acceleration sensor) และเครื่องตรวจจับความเร็วเชิงมุม แบบหมุนสามแกน (three-axis rotation angular velocity sensor) โครงข่ายประสาท เทียมในระบบนี้ประกอบด้วย ชั้น การเข้ารหัส (Encoding Layer) ชั้นถอดรหัส (Decoding Layer) และตัวแบ่งกลุ่ม ResNet18 โดยที่ Encoding Layer ประกอบด้วย CNN 3 ชั้นและ Casual LSTM 3 ชั้น ส่วน Decoding Layer ประกอบด้วย deconvolution 3 ชั้นและ Casual LSTM 3 ชั้น ใช้ชุดข้อมูล SisFall เพื่อประเมินประสิทธิภาพของอัลกอริทึมผลการทดลองแสดงให้เห็นว่าอัลกอริทึม มีความแม่นยำสูงถึง 99.79%

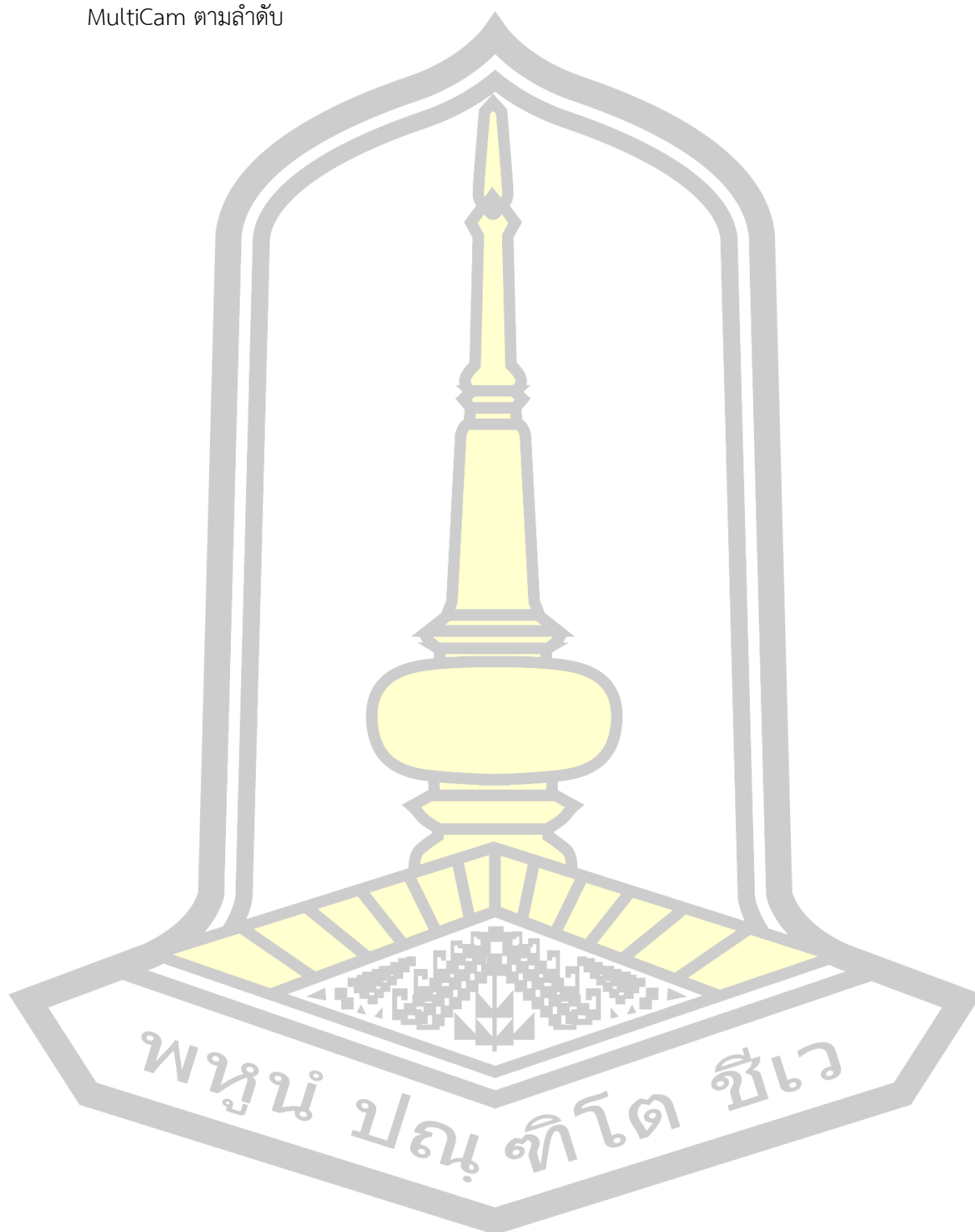
ปี ค.ศ. 2020 Yuxi Chen, Xiangbo Kong, Lin Meng, Hiroyuki Tomiyama [39] ได้นำเสนอระบบตรวจจับการล้มด้วยอุปกรณ์ปลายทาง (Edge Computing) สำหรับผู้สูงอายุ งานวิจัยนี้ได้ทำการศึกษาวิธีการเลือกแบบจำลองที่เหมาะสมที่สุดในการตรวจจับการล้ม นอกจากนี้ หากมีการตรวจพบการผิดพลาดเฟรมที่ตรวจพบการผิดพลาด จะถูกอัปโหลดไปยังคลาวด์เซิร์ฟเวอร์ และโมเดลจะได้รับเทรนใหม่และส่งไปยัง อุปกรณ์ปลายทางเพื่อตรวจจับการล้มและยังมีการใช้วิธีการ Deep Learning เพื่อทดสอบข้อมูลการเทรน ซึ่งโมเดล GoogLeNet ให้ประสิทธิภาพที่ดีที่สุด

ปี ค.ศ. 2020 YANG Yi-fei, HU Qing-chun, DAI Ming-ming, YAN Hong-ri และ LING Jin-yang [40] ได้นำเสนอการออกแบบและการใช้งานระบบตรวจจับการล้มด้วยวิดีโอ งานวิจัยนี้ ทำการปรับปรุงความแม่นยำและความเสถียรระบบ จึงได้ออกแบบระบบตรวจจับการล้มแบบ double-algorithm โดยใช้ไลบรารี OpenPose และมีการตั้งค่าไว้เพื่อการตัดสินใจใช้ในการ ตรวจจับการล้ม เป็นอัลกอริธึมตัวที่ 1 ในขณะที่การรวมกันของอีกโครงข่ายประสาทเทียมสองรูปแบบ นั้น ถูกใช้อย่างอิสระเป็นอัลกอริธึมตัวที่ 2 เมื่อทำการเปรียบเทียบโมเดล OpenPose หรือ 3DCNN เพียงอย่างเดียว ผลลัพธ์จะแสดงค่า F1 ที่สูงกว่า แม่นยำกว่าและมีความเหมาะสมในทางปฏิบัติ มากกว่าระบบ ตรวจจับการล้มแบบเรียลไทม์ได้รับการออกแบบ ตามการศึกษาอัลกอริทึมที่ 1 และ อัลกอริทึมที่ 2 ได้เพิ่มโมดูลการวิเคราะห์ข้อมูลประวัติเพื่อจัดเก็บและวิเคราะห์ภัยพิบัติของการหกล้ม

ปี ค.ศ. 2020 Mohammed Farsi [41] ได้นำเสนอแอปพลิเคชันของชุดโมเดล RNN deep neural network ตรวจจบบการล้มจากชุดข้อมูลของ IoT งานวิจัยนี้ได้ศึกษาเพื่อแก้ปัญหาจำนวนชุดข้อมูลการเทรนที่ไม่เพียงพอในการตรวจจับการล้ม ทำได้โดยใช้ชุดเทคนิคต่าง ๆ เช่น Bagging และ Stacking ซึ่งชุดของการทดลองได้ดำเนินการบนชุดข้อมูลของ SmartFall และผลการทดลอง แสดงให้เห็นว่า อัลกอริธึมของ Random Forest มีประสิทธิภาพดีกว่าเมื่อเปรียบเทียบกับโมเดล LSTM เพียงตัวเดียวและต่างจากเทคนิคต่างๆแสดงให้เห็นว่าโมเดลการเรียนรู้ของเครื่องให้ ประสิทธิภาพที่ดีขึ้นเมื่อเปรียบเทียบกับโมเดล LSTM ในชุดข้อมูล SmartFall เนื่องจากแต่ละรุ่นของ LSTM ได้รับชุดย่อยของข้อมูลบางส่วน โมเดลจึงเกิด overfit อันเป็นผลมาจากข้อมูลการเทรน ไม่เพียงพอในอนาคต จะใช้วิธี SMOTE และเทคนิค class-weight เพื่อแก้ไขปัญหา ชุดข้อมูลที่ไม่สมดุล

ปี 2021 Sowmyayani S, Murugan V และ Kavitha J [42] ได้นำเสนอ การตรวจจับการล้มในระบบการดูแลผู้สูงอายุตามกลุ่มของภาพ งานวิจัยนี้ได้เสนอวิธีการใหม่ในการตรวจจับการหกล้มของผู้สูงอายุ โดยการเรียงลำดับวิดีโอซึ่งจะแบ่งออกเป็นกลุ่มของภาพ (GOP) ตามการเปลี่ยนของฉาก จากนั้นจะใช้ Optical Flow ระบุการเปลี่ยนแปลงที่เกิดขึ้นระหว่างกลุ่มของภาพ หากมีการเปลี่ยนแปลงอย่างมากในกลุ่มของภาพ หากขาดการเคลื่อนไหวจะถูกระบุอีกครั้งโดยใช้ Optical flow ในกลุ่มของภาพถัดไป ประสิทธิภาพของวิธีการที่เสนอจะถูกเปรียบเทียบกับวิธีการล่าสุดและพิสูจน์ได้ว่ามีความแม่นยำดีกว่า โดยได้รับความแม่นยำ 99% ใน 5 วินาทีสำหรับชุดข้อมูล UR FDD

นอกจากนี้ ความถูกต้องของวิธีการที่เสนอคือ 98.15% และ 99% สำหรับชุดข้อมูล FDD และ MultiCam ตามลำดับ



บทที่ 3

วิธีดำเนินการวิจัย

3.1 การเตรียมใช้งานอุปกรณ์ Edge

ในงานวิจัยนี้ได้เลือกใช้ Coral Dev Board เป็นอุปกรณ์ Edge ซึ่งถูกผลิตโดย google เป็นคอมพิวเตอร์บอร์ดเดียว (Single board) ที่สามารถทำงานเกี่ยวกับการเรียนรู้ของเครื่องจักร (Machine Learning) ที่ต้องการทำการอนุมาน (inference) อย่างรวดเร็ว ในรูปแบบของแผงวงจรเดี่ยวขนาดเล็ก สามารถนำมาใช้เพื่อสร้างต้นแบบของระบบปัญญาประดิษฐ์ (Artificial Intelligent) ในรูปแบบของระบบสมองกลฝังตัวโดยมีขั้นตอนการเตรียมใช้งานดังนี้

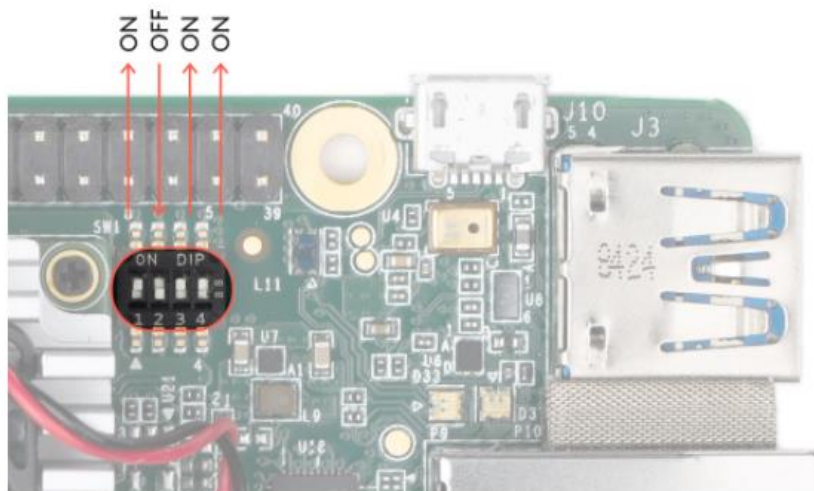
3.1.1 ความต้องการของระบบ

- 1) ติดตั้งระบบปฏิบัติการ Mendel Linux บนเครื่องคอมพิวเตอร์
- 2) ติดตั้ง Python version 3
- 3) การ์ด microSD ที่มีความจุอย่างน้อย 8 GB
- 4) แหล่งจ่ายไฟขนาด 2-3 A/5 V และสาย USB Type C เพื่อจ่ายไฟให้กับบอร์ด
- 5) สาย USB Type C ใช้เพื่อเชื่อมต่อบอร์ดกับคอมพิวเตอร์
- 6) ระบบอินเทอร์เน็ต

3.1.2 การโปรแกรมลงบอร์ด Coral Dev Board

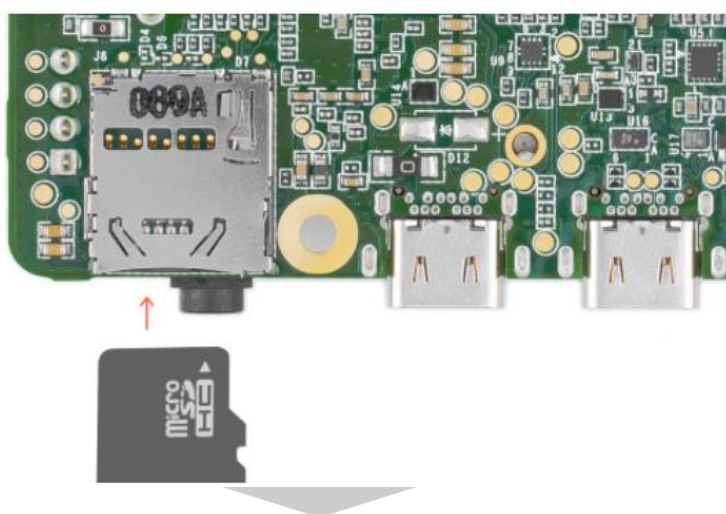
- 1) ดาวน์โหลดโปรแกรม enterprise-eagle-flashcard-20211117215217.zip จากลิงค์นี้ <https://mendel-linux.org/images/enterprise/eagle/enterprise-eagle-flashcard-20211117215217.zip> และแตกโปรแกรมลงในคอมพิวเตอร์จะได้ไฟล์ที่ชื่อ flashcard_arm64.img
- 2) ใช้โปรแกรม balenaEtcher จากลิงค์ <https://www.balena.io/etcher/> เพื่อเขียนไฟล์ flashcard_arm64.img ลงใน microSD ใช้เวลาประมาณ 5-10 นาที ขึ้นอยู่กับความเร็วของ microSD card และ adapter ของ microSD
- 3) เปลี่ยนสวิตช์โหมดบูตใน Coral Dev Board เพื่อบูตจากการ์ด SD ดังแสดงในภาพประกอบ 6

Boot mode	Switch 1	Switch 2	Switch 3	Switch 4
SD card	ON	OFF	ON	ON



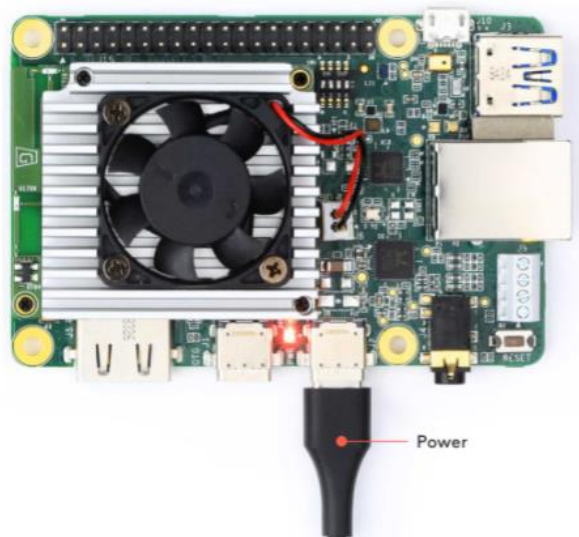
ภาพประกอบ 6 การตั้งค่าสวิตช์บูตเป็นโหมด SD การ์ด

4) เมื่อ microSD การ์ดถูกแฟลชแล้ว ให้นำการ์ดออกจากคอมพิวเตอร์อย่างแล้ว
เสียบลงใน Coral Dev Board (หมุดของการ์ดหันไปทางบอร์ด) และต้องปิดการทำงานของบอร์ดด้วย



ภาพประกอบ 7 ช่องเสียบการ์ด microSD อยู่ที่ด้านล่างของบอร์ด

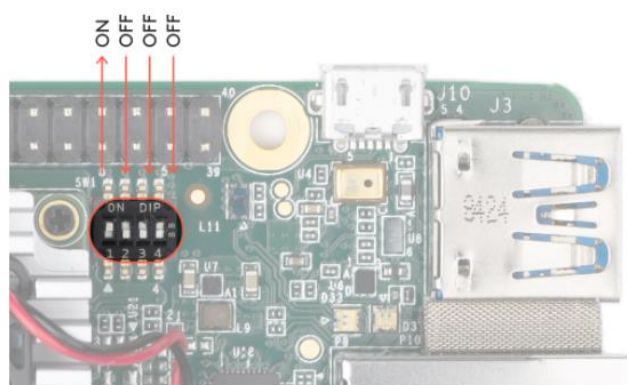
เมื่อบอร์ดบูตขึ้น บอร์ดจะโหลดอิมเมจจาก SD การ์ด และเริ่มแฟลชหน่วยความจำ eMMC ของบอร์ด ซึ่งการทำงานทั้งหมดควรเสร็จสิ้นใน 5-10 นาที ขึ้นอยู่กับความเร็วของ microSD การ์ด เราจะรู้ว่าระบบทำงานเสร็จแล้วเมื่อบอร์ดปิดและไฟ LED สีแดงดับลง



ภาพประกอบ 8 สาย USB-C ที่เชื่อมต่อกับบอร์ด

- 1) เมื่อไฟ LED สีแดงดับลง ให้ถอดปลั๊กและถอด microSD การ์ด ออกจากบอร์ด
- 2) เปลี่ยนสวิตช์โหมดบูตเป็นโหมด eMMC ดังแสดงในภาพประกอบ 9

Boot mode	Switch 1	Switch 2	Switch 3	Switch 4
eMMC	ON	OFF	OFF	OFF



ภาพประกอบ 9 สวิตช์บูตตั้งค่าเป็นโหมด eMMC

7) เชื่อมต่อบอร์ดกับแหล่งจ่ายไฟเพื่อเปิดการทำงานของบอร์ด ซึ่งจะบูตด้วย OS Mendel Linux การบูตเครื่องครั้งแรกหลังจากกระแสปะลอกจากแฟลชจะใช้เวลาประมาณ 3 นาที

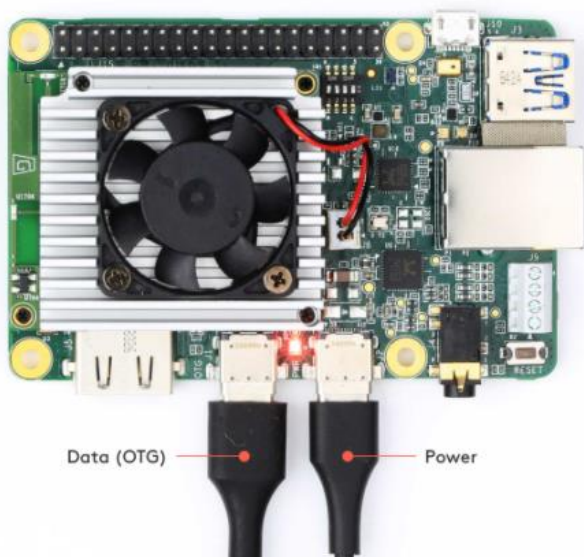
3.1.3 การติดตั้ง MDT

MDT เป็นเครื่องมือแบบ Command Line ที่ควบคุมการทำงานของ Coral Dev Board ซึ่งสามารถแสดงรายการอุปกรณ์ที่เชื่อมต่อกับบอร์ด ซึ่งคอมพิวเตอร์จะต้องติดตั้งโปรแกรมระบบปฏิบัติการ Debian และต้องเปิด shell terminal บนบอร์ดด้วย การติดตั้ง MDT บนคอมพิวเตอร์ ใช้คำสั่ง ดังนี้

```
python3 -m pip install --user mendel-development-tool
echo 'export PATH="$PATH:$HOME/.local/bin"' >> ~/.bash_profile
source ~/.bash_profile
```

ในขั้นต่อไปจะต้องทำการเชื่อมต่อบอร์ดกับ MDT เมื่อบอร์ดมีโปรแกรม Mendel Linux แล้ว บอร์ดก็สามารถเริ่มต้นสร้างเซสชันเซลล์ที่ปลอดภัยโดยใช้เครื่องมือ MDT ในการสร้าง OpenSSH เพื่อทำการติดต่อและควบคุมบอร์ดผ่านคอมพิวเตอร์โดยมีขั้นตอนดังนี้

- 1) เชื่อมต่อสาย USB-C จากคอมพิวเตอร์เข้ากับพอร์ต USB ของบอร์ด ที่มีข้อความว่า "OTG"



ภาพประกอบ 10 การเชื่อมต่อข้อมูลและการจ่ายไฟโดยใช้สาย USB-C

- 2) เปิดหน้าต่าง Terminal แล้วพิมพ์คำสั่ง

mdt devices

เมื่อประมวลผลแล้วจะเห็นชื่อของเครื่องคอมพิวเตอร์และไอพีแอดเดรส ซึ่งจะแตกต่างกันไปตามแต่ละเครื่อง เช่น

orange-horse (192.168.100.2)

ต่อจากนั้นพิมพ์คำสั่ง `mdt shell` เพื่อเข้าสู่การจัดการบอร์ด

3) เชื่อมต่ออินเทอร์เน็ตให้กับบอร์ด Coral Dev Board โดยพิมพ์คำสั่ง

nmtui

ต่อจากนั้นเลือก **Activate a connection** และเลือก network จากรายการ Wi-Fi (wlan0) หรือใช้คำสั่ง

`nmcli dev wifi connect <NETWORK_NAME> password <PASSWORD> ifname wlan0`

เพื่อเชื่อมต่อกับชื่อเครือข่ายที่รู้จัก ต่อจากนั้นทำการตรวจสอบการเชื่อมต่อด้วยคำสั่ง

nmcli connection show

ต่อจากนั้นก็จะเห็นข้อมูลของเครือข่ายของบอร์ด เช่น

NAME	UUID	TYPE	DEVICE
MyNetworkName	61f5d6b2-5f52-4256-83ae-7f148546575a	802-11-wireless	wlan0

DEVICE คือ wlan0 สำหรับการเชื่อมต่อ Wi-Fi หรือ eth0 สำหรับการเชื่อมต่ออีเทอร์เน็ต

3.1.4 การอัปเดต Mendel บนบอร์ด

ในการอัปเดต Mendel บนบอร์ด Coral Dev Board นั้นจะใช้คำสั่งดังนี้

`sudo apt-get update`

`sudo apt-get dist-upgrade`

3.1.5 การทดสอบการทำงาน Machine Learning บนบอร์ด

การทำงานจะผ่าน MDT ที่เครื่องคอมพิวเตอร์เพื่อควบคุมการทำงานของบอร์ดผ่านทางคอมพิวเตอร์ และทำการเชื่อมต่อบอร์ดกับคอมพิวเตอร์ผ่านทางพอร์ต OTG โดยใช้สาย USB-C ต่อจากนั้นพิมพ์คำสั่งที่คอมพิวเตอร์ดังนี้

`edgetpu_demo -stream`

จากนั้นบนเครื่องคอมพิวเตอร์ที่เชื่อมต่อกับ Dev Board หากเชื่อมต่อกับบอร์ดโดยใช้ MDT ผ่าน USB ให้เปิด 192.168.100.2:4664 ในเบราว์เซอร์ หากเชื่อมต่อกับบอร์ดด้วยวิธีอื่น

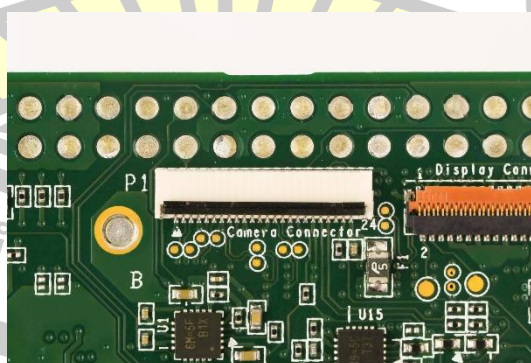
เช่น SSH ผ่าน LAN หรือด้วยสายอินเทอร์เน็ต ให้พิมพ์ที่อยู่ IP ที่ของบอร์ดลงในบราวเซอร์ด้วยพอร์ต 4664 เช่น <http://192.168.1.8:4664>



ภาพประกอบ 11 การทดสอบ Machine Learning บนบอร์ด

3.2 ติดตั้งกล้องวิดีโอเข้ากับ Coral Dev Board

กล้องจะเชื่อมต่อผ่านทาง Connector แบบ CSI ซึ่งจะอยู่ที่ด้านล่างของ Coral Dev Board ก่อนการเชื่อมต่อต้องตรวจให้แน่ใจก่อนว่าไม่ได้มีการจ่ายไฟเลี้ยงไปยังบอร์ด ที่ด้านล่างของ Dev Board บริเวณตำแหน่ง CSI "Camera Connector" ให้พลิกสล็อตขนาดเล็กให้หงายขึ้น ดังแสดงในภาพประกอบ 12



ภาพประกอบ 12 ตำแหน่งของ CSI Camera Connector

ต่อจากนั้นให้เลื่อนสายดินเข้าไปในขั้วต่อโดยให้หมดสัมผัสหันไปทางบอร์ด (แถบสีน้ำเงินหันออกจากบอร์ด) ดังแสดงในภาพประกอบ 13 แล้วปิดสลักดำ และเชื่อมต่อปลายอีกด้านของสายดินเข้ากับขั้วต่อที่ตรงกันบนกล่อง ดังแสดงในภาพที่ 14



ภาพประกอบ 13 การต่อสายดินเข้ากับ CSI Camera Connector



ภาพประกอบ 14 การต่อสายดินเข้ากับกล่อง

ต่อจากนั้นทำการจ่ายไฟเข้าบอร์ดและตรวจสอบว่าบอร์ดสามารถตรวจพบกล้องหรือไม่ โดยเรียกใช้คำสั่งนี้ `v4l2-ctl --list-devices` เมื่อรันคำสั่งแล้วจะต้องเห็นว่ากล้องอยู่ที่ตำแหน่ง `/dev/video0` ซึ่งจะแสดงข้อมูลออกมาดังนี้

```
i.MX6S_CSI (platform:30a90000.csi1_bridge):
/dev/video0
```

3.3 ออกแบบขั้นตอนการตรวจจับการล้มของมนุษย์

ในงานวิจัยนี้ได้ออกแบบการตรวจจับการล้มของมนุษย์ 2 วิธี โดยการใช้ข้อมูลตำแหน่งโครงกระดูก (Skeleton) คือ ค่าพิกัดตำแหน่งส่วนหัวซึ่งจะประกอบไปด้วย หูซ้าย หูขวา ตาซ้าย ตาขวา และ จมูก, ค่าพิกัดตำแหน่งส่วนกลาง ประกอบไปด้วย เอวซ้าย และ เอวขวา , ค่าพิกัดตำแหน่ง

ส่วนล่าง ประกอบไปด้วย ข้อเท้าซ้าย และ ข้อเท้าขวา ซึ่งถูกสกัดมาจาก PoseNet และ ความเร็วในการเคลื่อนที่ของส่วนหัว มาใช้ในการตรวจจับการล้ม



ภาพประกอบ 15 ข้อมูลโครงกระดูก 17 จุด โดยได้จาก PoseNet

3.3.1 ขั้นตอนการตรวจจับการล้มวิธีที่ 1 (M1)

วิธีที่ 1 ได้ออกแบบขั้นตอนการตรวจจับการล้มออกเป็น 7 ขั้นตอน คือ

ขั้นตอนที่ 1 รับข้อมูลวิดีโอ 2D RGB จากกล้อง

ขั้นตอนที่ 2 สกัดข้อมูลโครงกระดูก (Skeleton) จากวิดีโอ โดยใช้ PoseNet โมเดลซึ่งมี

ทั้งหมด 17 จุดตั้งแต่ศีรษะจนถึงเท้า

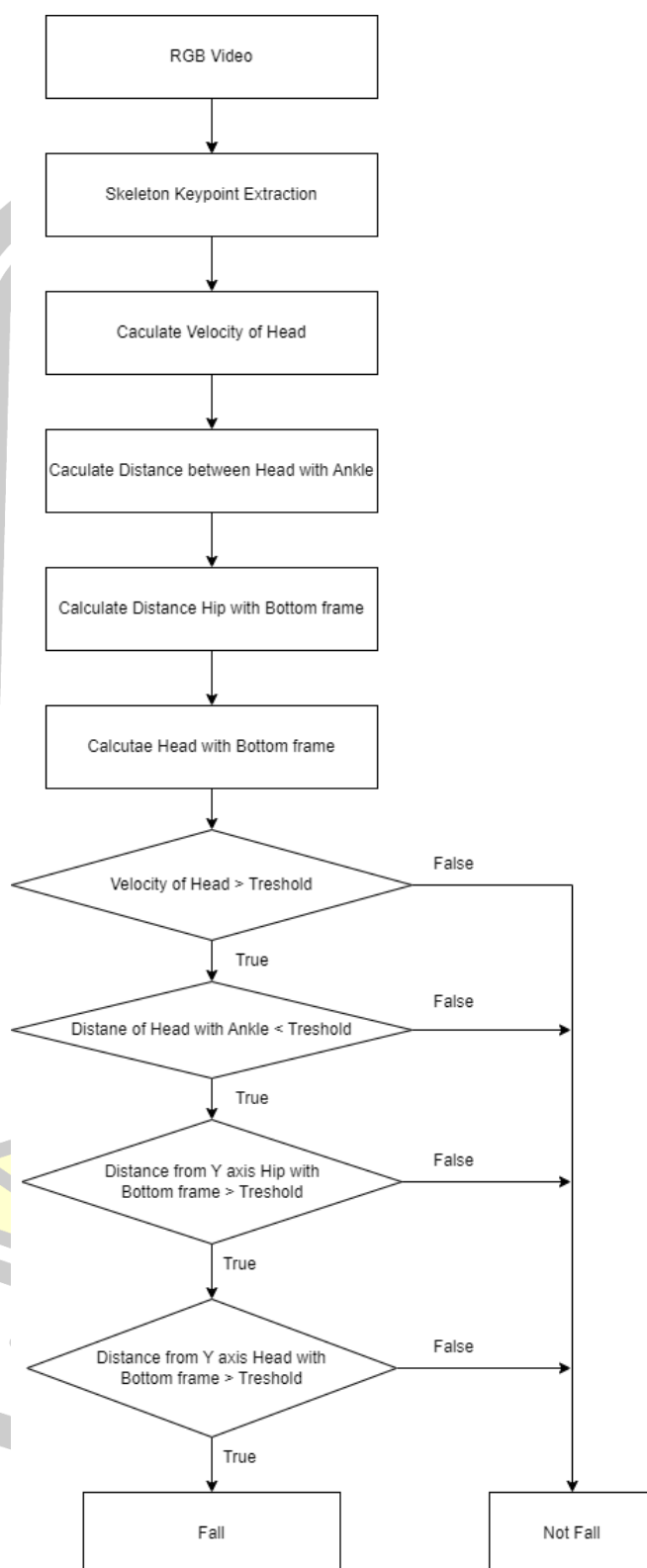
ขั้นตอนที่ 3 คำนวณค่าความเร็วการเคลื่อนที่ของส่วนหัว

ขั้นตอนที่ 4 คำนวณค่าระยะทางระหว่างส่วนหัวกับข้อเท้า

ขั้นตอนที่ 5 ตรวจสอบตำแหน่งสะโพกแกน Y กับพื้นที่ด้านล่างของเฟรม

ขั้นตอนที่ 6 ตรวจสอบตำแหน่งส่วนหัวแกน Y กับพื้นที่ด้านล่างของเฟรม

ขั้นตอนที่ 7 ทำนายการล้มจากข้อมูลที่ได้จากการคำนวณ



ภาพประกอบ 16 ขั้นตอนการตรวจจับการล้มวิธีที่ 1 (M1)

ขั้นตอนทำนายการล้มจากข้อมูลที่ได้จากการคำนวณ ในวิธีที่ 1 (M1) มีขั้นตอนดังนี้

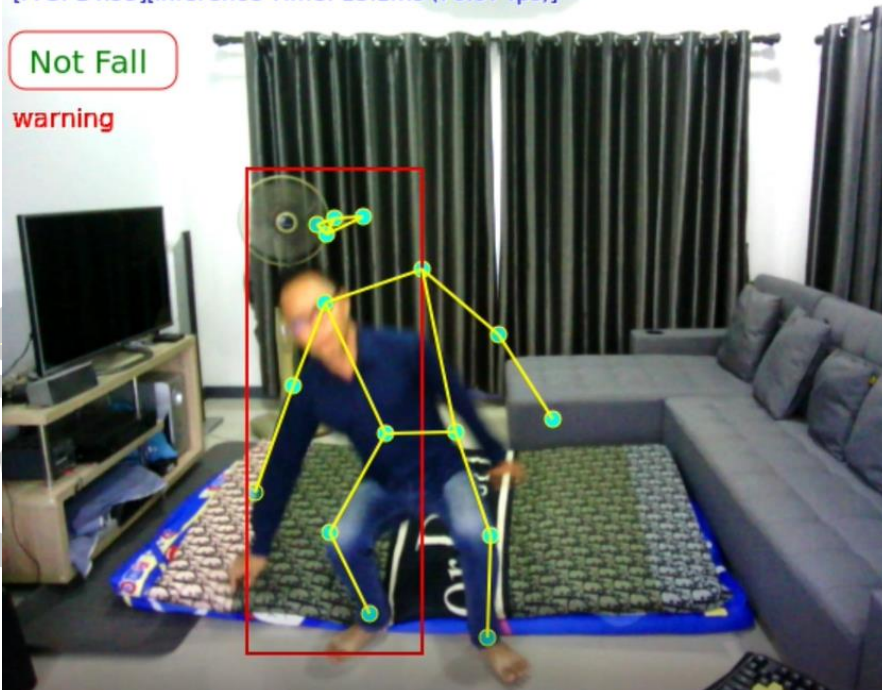
ขั้นตอนที่ 1 ตรวจสอบความเร็วส่วนหัวค่าตั้งแต่ 20 ขึ้นไปถือว่าต้องระวังเกิดการล้ม ถ้ามากกว่า 20 ให้ทำขั้นตอนที่ 2 ถ้าน้อยกว่าให้กลับไปตรวจสอบในขั้นตอนที่ 1 พร้อมกับแจ้งว่า Warning

ขั้นตอนที่ 2 ทำการตรวจสอบว่าระยะทางระหว่างหัวกับข้อเท้าไม่น้อยกว่า 200 หรือไม่ ถ้าน้อยกว่า ให้ไปทำขั้นตอนที่ 3 ถ้ามากกว่าให้กลับไปตรวจสอบในขั้นตอนที่ 1 พร้อมกับแจ้งว่า Not Fall

ขั้นตอนที่ 3 ทำการตรวจสอบว่า ระยะทางระหว่างจากเอวในแนวตั้ง (แกน Y) มายังบริเวณที่ด้านล่างของเฟรม มากกว่า 350 หรือไม่ ถ้ามากกว่าให้ไปทำขั้นตอนที่ 4 ถ้าน้อยกว่าให้กลับไปตรวจสอบในขั้นตอนที่ 1 พร้อมกับแจ้งว่า Not Fall

ขั้นตอนที่ 4 ทำการตรวจสอบว่า ส่วนหัวในแนวตั้ง (แกน Y) มายังบริเวณพื้นที่ด้านล่างของเฟรม มากกว่า 350 หรือไม่ ถ้ามากกว่าให้แจ้งว่า Fall หมายความว่า เกิดการล้ม ถ้าน้อยกว่าให้กลับไปตรวจสอบในขั้นตอนที่ 1 พร้อมกับแจ้งว่า Not Fall

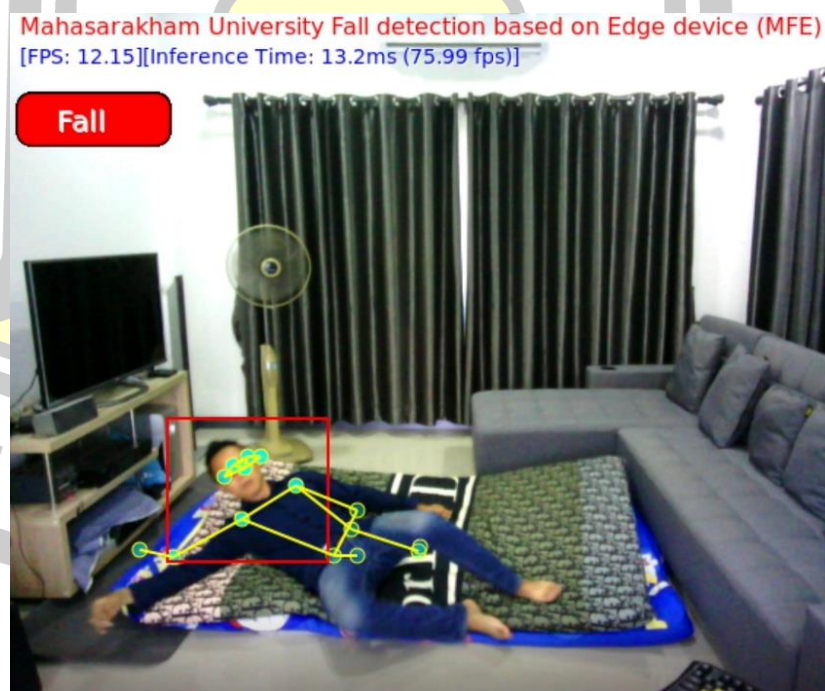
Mahasarakham University Fall detection based on Edge device (MFE)
[FPS: 14.55][Inference Time: 13.1ms (76.07 fps)]



ภาพประกอบ 17 เมื่อส่วนหัวมีความเร็วในการเคลื่อนที่มากกว่า 20 จะแสดง Warning



ภาพประกอบ 18 กรณีไม่ล้มจะแสดงคำว่า Not Fall



ภาพประกอบ 19 กรณีล้มจะแสดงคำว่า Fall

3.3.2 ขั้นตอนการตรวจจับการล้มวิธีที่ 2 (M2)

วิธีที่ 2 ได้ออกแบบขั้นตอนการตรวจจับการล้มออกเป็น 7 ขั้นตอน คือ

ขั้นตอนที่ 1 รับข้อมูลวิดีโอ 2D RGB จากกล้อง

ขั้นตอนที่ 2 สกัดข้อมูลโครงกระดูก (Skeleton) จากวิดีโอ โดยใช้ PoseNet

ขั้นตอนที่ 3 คำนวณหาค่าความเร็วการเคลื่อนที่ของส่วนหัว

ขั้นตอนที่ 4 คำนวณหาค่าแกน Y ส่วนหัว

ขั้นตอนที่ 5 คำนวณหาตำแหน่งของข้อเท้า

ขั้นตอนที่ 6 เปรียบเทียบระหว่างส่วนหัวกับข้อเท้าว่าอยู่ในตำแหน่งใด

ขั้นตอนที่ 7 ทำนายการล้มจากข้อมูลที่ได้จากการคำนวณ โดยคำนวณจากสูตร

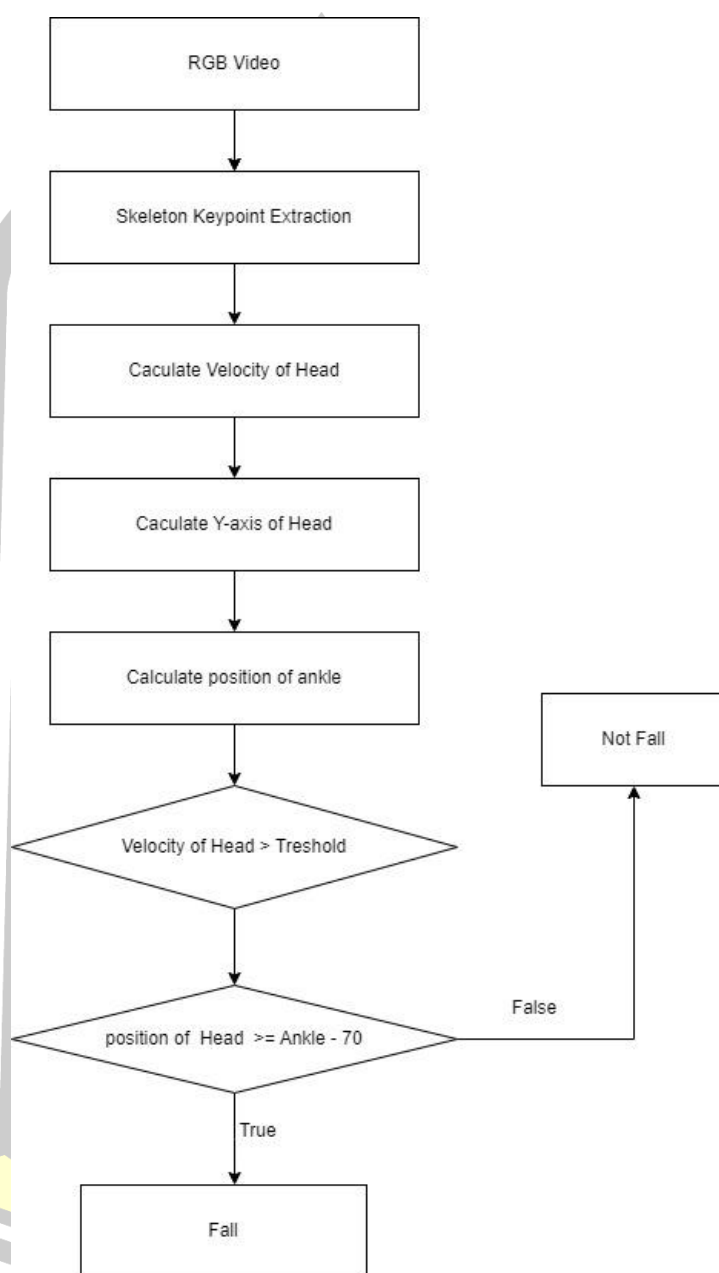
$$Fall = Head \geq (ankle - 70) \quad (1)$$

ขั้นตอนทำนายการล้มจากข้อมูลที่ได้จากการคำนวณ ในวิธีที่ 2 (M2) มีขั้นตอนดังนี้

ขั้นตอนที่ 1 ตรวจสอบความเร็วส่วนหัวค่าตั้งแต่ 20 ขึ้นไปถือว่าต้องระวังเกิดการล้ม ถ้ามากกว่า 20 ให้ทำขั้นตอนที่ 2 ถ้าน้อยกว่าให้กลับไปตรวจสอบในขั้นตอนที่ 1 พร้อมกับแจ้งว่า Warning

ขั้นตอนที่ 2 ตรวจสอบว่า ตำแหน่งส่วนหัว (Head) มีค่ามากกว่าหรือเท่ากับตำแหน่งของ ข้อเท้าลบด้วย 70 ถ้าเป็นไปตามเงื่อนไขให้แจ้งว่า Fall หมายความว่า เกิดการล้ม ถ้าน้อยกว่าให้ไปตรวจสอบขั้นตอนที่ 1 พร้อมกับแจ้งว่า Not Fall

พูน ปณ จิต ชีเว

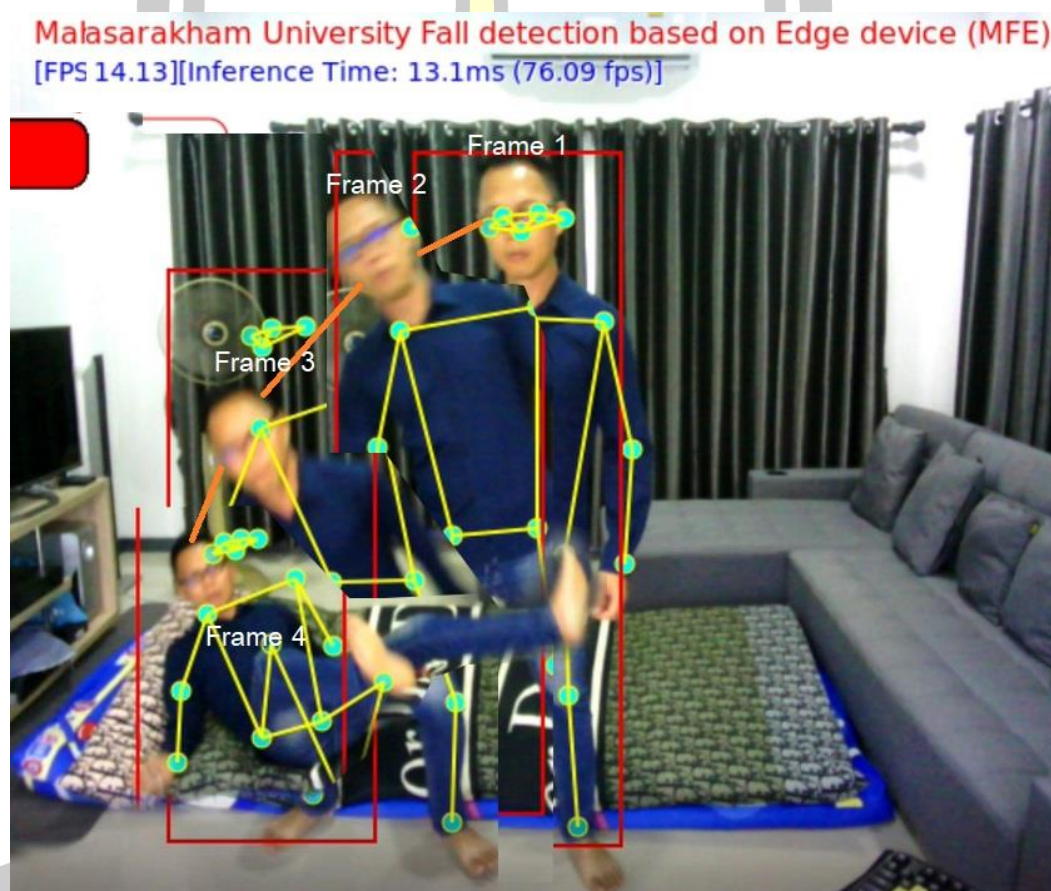


ภาพประกอบ 20 ขั้นตอนการตรวจจับการล้มวิธีที่ 2 (M2)

3.3.3 ความเร็วของส่วนหัว

เมื่อเกิดการล้มจุดศูนย์กลางของส่วนหัว ซึ่งจะเปลี่ยนไปในแนวตั้งอย่างกะทันหัน เกิดความเร็วมากกว่าปกติ ซึ่งจะใช้คุณลักษณะนี้ในการตรวจจับการล้มจากส่วนหัว โดยการประมวลผล

นั้น จะใช้ข้อมูลพิกัดส่วนหัวทั้ง 5 จุดคือ หูซ้าย หูขวา ตาซ้าย ตาขวา และ จมูก ที่ได้รับจากแต่ละเฟรมของภาพจาก PoseNet โดยใช้เวลาในการประมวลผลที่สั้นมากจากตำแหน่งยืนไปยังตำแหน่งที่ล้ม ดังนั้นจึงต้องมีการตรวจเช็คทุกเฟรมที่อยู่ติดกัน จากภาพประกอบ 18 แสดงจุดสำคัญของโครงกระดูก สำหรับวิธีการสกัดคุณลักษณะนั้น จะหาค่าความเร็วในการเคลื่อนที่ตำแหน่งของส่วนหัวลงไปยังพื้นที่ส่วนล่างของเฟรมภาพ และตรวจสอบว่าถึงค่าที่กำหนดหรือไม่ ซึ่งจะใช้เป็นเกณฑ์ในการตรวจจับการล้ม ในส่วนของการเคลื่อนที่ความเร็วส่วนหัว



ภาพประกอบ 21 การตรวจจับความเร็วส่วนหัว

$$s = |\sqrt{(x_{i-1} - x_i)^2} + \sqrt{(y_{i-1} - y_i)^2}| \quad (2)$$

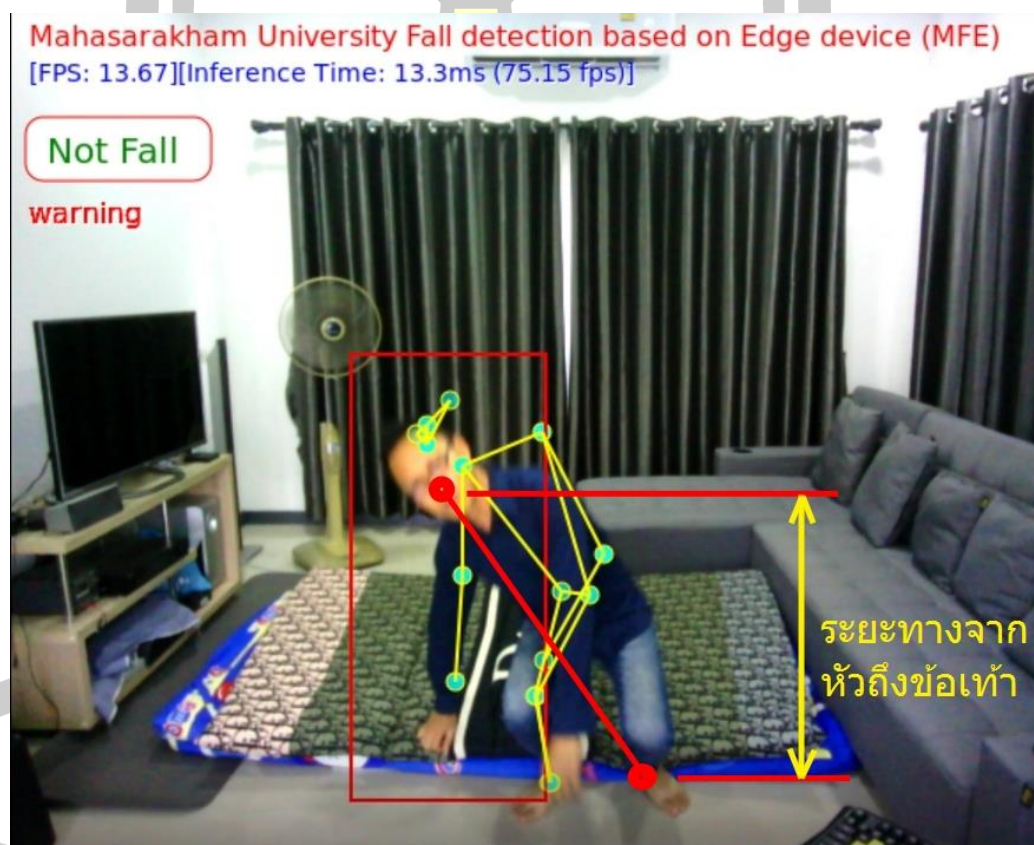
จากสมการที่ 2 ใช้วิธีวัดระยะทางยูคลิเดียน และ S คือความเร่งของตำแหน่งหัวของค่า X และ Y โดยจะคำนวณจากเฟรมต่อเฟรมตามลำดับ

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3)$$

จากสมการที่ 3 d คือ ค่าระยะทางจากจุดหนึ่งไปยังจุดหนึ่ง โดยจะคำนวณเฟรมต่อเฟรมตามลำดับ สำหรับ x และ y คือ ค่าพิกัดของแต่ละตำแหน่งที่ต้องการนำมาคำนวณ โดยจะได้จาก PoseNet

3.3.4 การคำนวณระยะทางของ Keypoint

ในการหาค่าระยะทางนั้น จะคำนวณหา จุด คือ จากจุดส่วนหัวไปยังข้อเท้า 1 ซึ่งจะกำหนดไว้ที่ ในการตัดสินใจล้มหรือไม่ในส่วนนั้นนั้น จะคำนวณระยะห่างระหว่าง หัวกับข้อเท้าว่า 200 ในการหาค่า 2 หรือไม่ ถ้าน้อยกว่าแสดงว่าเกิดการล้ม โดยใช้สมการที่ 200 น้อยกว่า

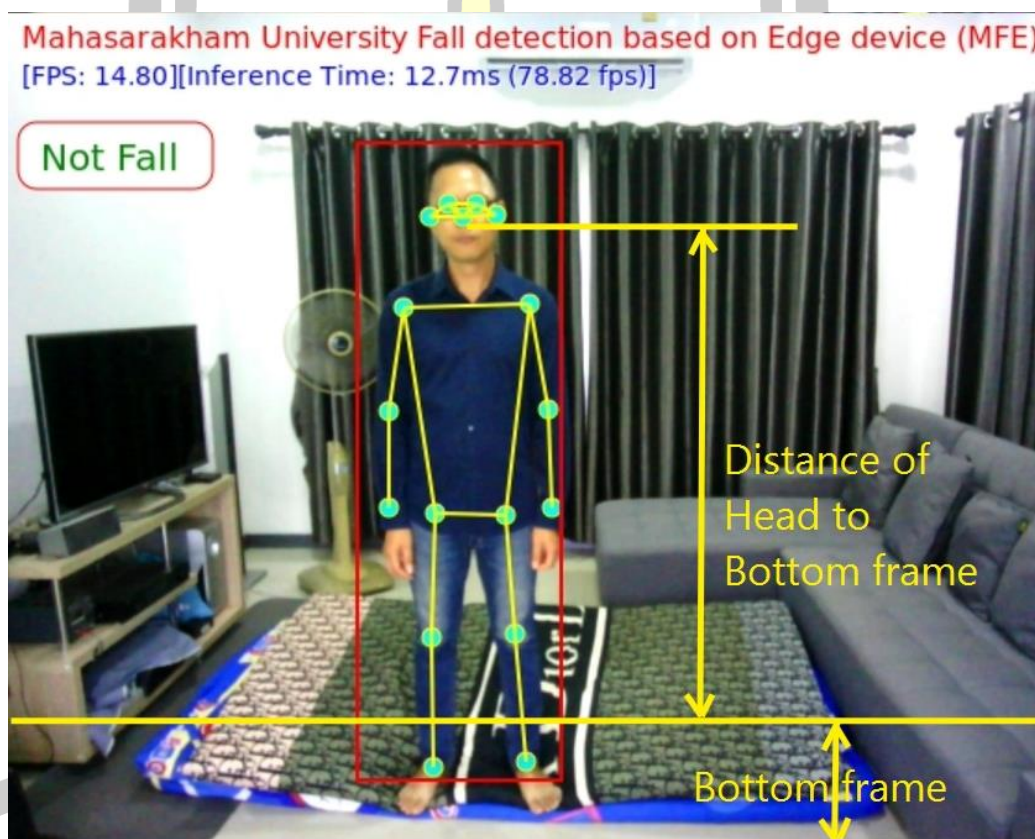


ภาพประกอบ 22 การวัดระยะทางจะหัวไปยังข้อเท้า

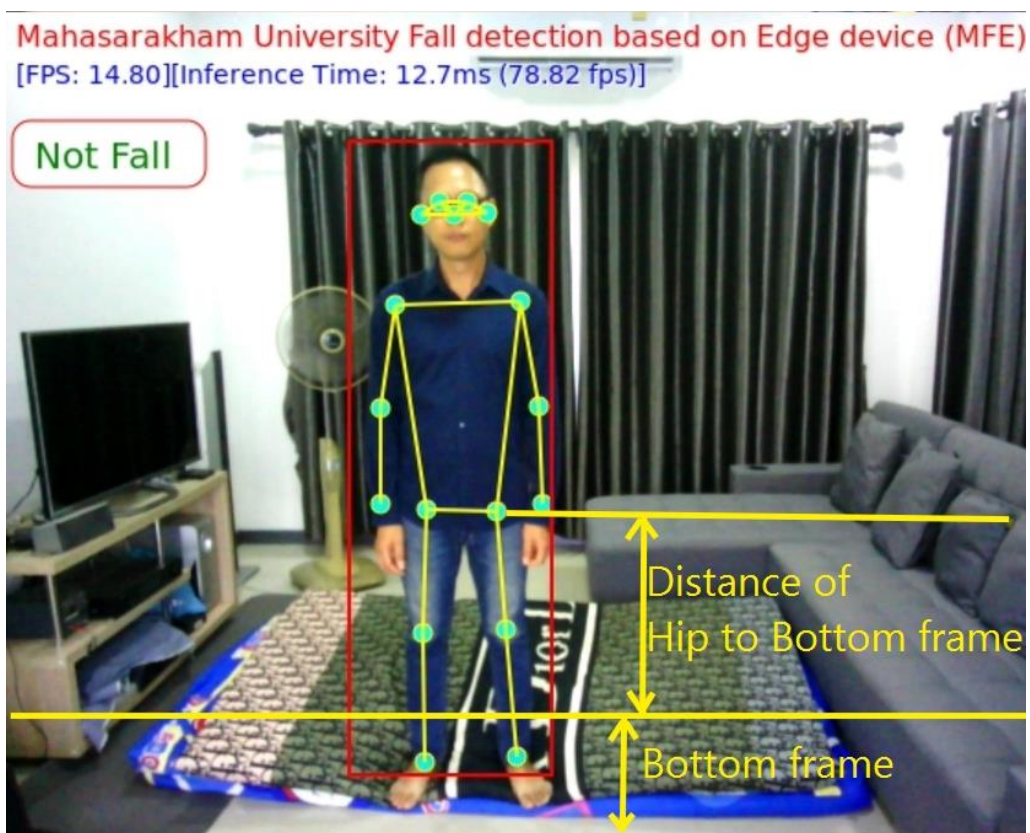
จากภาพประกอบที่ 22 เป็นการแสดงจุด keypoint ส่วนหัว และ ส่วนข้อเท้า เพื่อนำไปใช้ในการคำนวณหาระยะทาง เพื่อนำไปใช้ประกอบการตัดสินใจการล้ม โดยระยะทางจะได้ไม่น้อยกว่า 200 ซึ่งเป็นค่าที่ได้จากการทดลอง

3.3.5 การคำนวณตำแหน่ง Keypoint

ในงานวิจัยนี้ได้หาค่าตำแหน่ง Keypoint จำนวน 2 จุด คือ จากสะโพกไปยังพื้นที่ด้านล่างของเฟรม และ จากหัวไปยังพื้นที่ด้านล่างของเฟรม เพื่อนำไปใช้เป็นส่วนหนึ่งในการตัดสินใจการล้ม โดยได้กำหนดค่า Threshold ที่เป็นพื้นที่ด้านล่างของเฟรม จะกำหนดไว้ที่ 350 ซึ่งเป็นค่าที่ได้จากการทดลอง ในการตัดสินใจนั้น ถ้าส่วนหัวและส่วนของสะโพกในแนวตั้ง (แกน Y) มีค่าพิกัดเกินกว่า 350 แสดงว่าเกิดการล้ม ซึ่งค่านี้จะนำไปเป็นส่วนหนึ่งของการตัดสินใจการล้ม โดยจะใช้ สมการที่ 2 ในการคำนวณหาค่า



ภาพประกอบ 23 ระยะทางจากหัวไปยังพื้นที่ด้านล่างของเฟรม

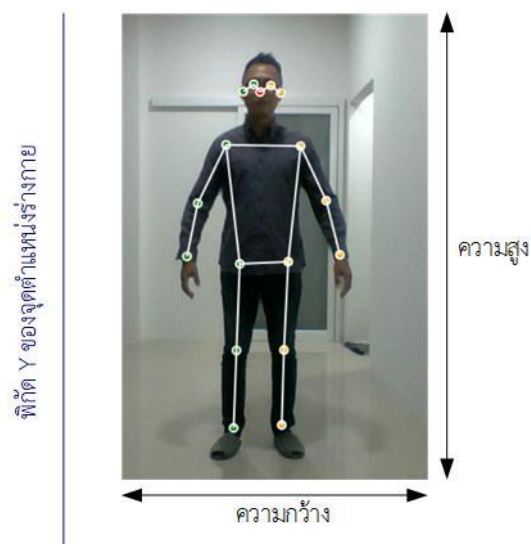


ภาพประกอบ 24 ระยะทางจากสะโพกไปยังพื้นที่ด้านล่างของเฟรม

3.3.6 อัตราส่วนความกว้างและความสูงของร่างกาย

วิธีการสกัดลักษณะเด่นของการล้มที่ตรวจพบคือการเปลี่ยนแปลงในขอบเขตของร่างกาย (กล่องขอบลำตัว) จึงสามารถตัดสินได้โดยการเปรียบเทียบความยาวแนวนอน (ความกว้างของร่างกาย) และแนวตั้ง (ความสูงของลำตัว) ของท่าก่อนและหลัง สำหรับอัตราส่วนความกว้างและความสูงของร่างกาย ทั้งความกว้างและความสูงของท่าทางจะเปลี่ยนไปตามการเปลี่ยนแปลงของท่าทางที่เคลื่อนไหวและระยะห่างระหว่างกล่องกับตัวแบบแต่ไม่รู้อัตราส่วนความห่าง การจำแนกท่าทางการเคลื่อนไหวของการล้มโดยการเปลี่ยนความกว้างและความสูงของรูปสี่เหลี่ยมผืนผ้าที่ล้อมรอบร่างกาย ได้แก่ ท่ากำลังเดินหรือยืน ท่าล้มหรือนั่ง และท่านอนหรือล้มลง แต่ละท่ามีอัตราส่วนความกว้างต่อความสูง ซึ่งสามารถเป็นพารามิเตอร์ในการตรวจจับการล้มได้

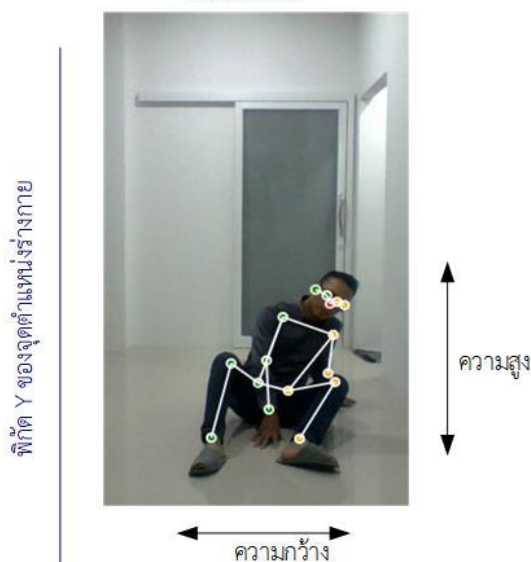
ท่ากำลังเดินหรือยืน



พิกัด X ของจุดตำแหน่งร่างกาย

ภาพประกอบ 25 ท่าเดินหรือยืน

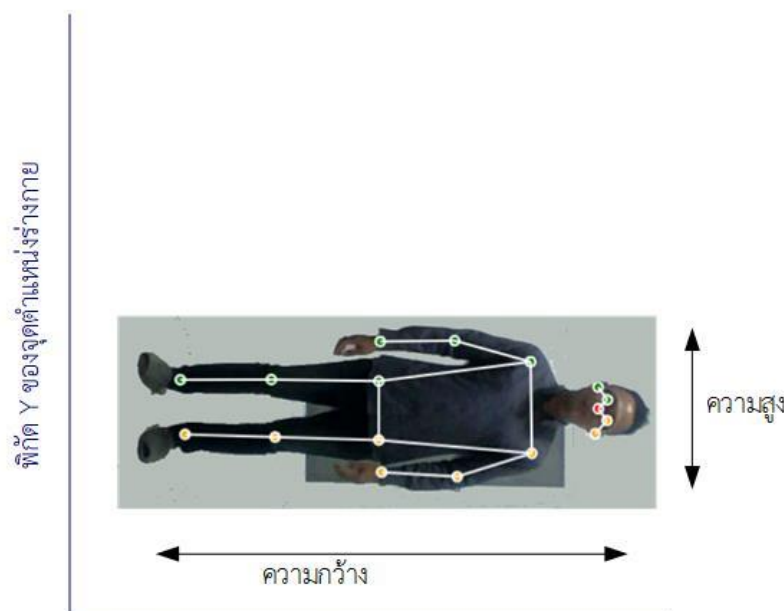
ท่าล้มหรือนั่ง



พิกัด X ของจุดตำแหน่งร่างกาย

ภาพประกอบ 26 ท่าล้มหรือนั่ง

ท่านอนหรือล้มลง



พิภพ X ของจุดตำแหน่งร่างกาย

ภาพประกอบ 27 ท่านอนหรือล้มลง

ภาพประกอบ 23 เป็นทำการเดินหรือยืน ซึ่งเป็นกิจกรรมในชีวิตประจำวัน ซึ่งความกว้างและความสูงของสี่เหลี่ยมผืนผ้าด้านนอกของร่างกายมนุษย์และอัตราส่วนของความกว้างต่อความสูงสามารถหาค่าได้ดังนี้

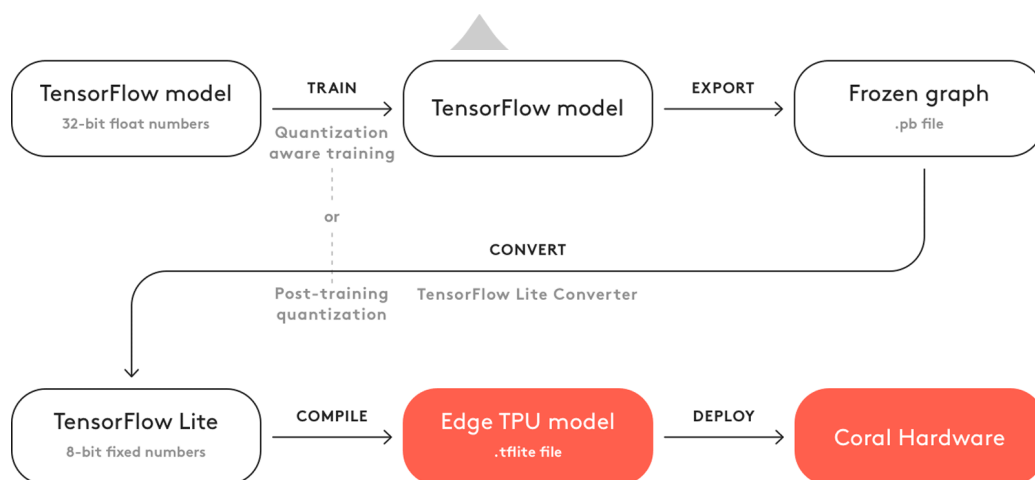
$$R = \frac{\text{Width}}{\text{Height}}$$

(4)

$$R < 1$$

เมื่อ R = อัตราส่วน, Width = ความกว้าง, Height = ความสูง ซึ่ง R ที่ได้จะต้องมีค่าน้อยกว่า 1 ในทางกลับกันอัตราส่วนความกว้างต่อความสูงที่ ของภาพประกอบ 21 ท่านอนหรือล้มลง จะมีค่ามากกว่า คือ R จะมีค่ามากกว่า 1 หรือ $R > 1$

3.4 แปลงโมเดลให้ทำงานบน Coral Dev Board



ภาพประกอบ 28 การแปลงโมเดลสำหรับ Coral Dev Board

เมื่อได้โมเดลที่สร้างจาก TensorFlow Cloud ต้องทำการแปลงให้อยู่ในรูปแบบ TensorFlow Lite เพื่อให้สามารถประมวลผลบน Coral Dev Board เมื่อแปลงแล้วจะได้ไฟล์นามสกุล .tflite ต่อจากนั้นก็สมารถนำไฟล์ที่ได้ไปประมวลผลบน Coral Dev Board



บทที่ 4

ผลการทดลองและการอภิปรายผล

4.1 ผลการทดลอง

ในการวิจัยการประยุกต์ใช้ท่าทางเพื่อตรวจจับการล้มของมนุษย์โดยประมวลผลบนอุปกรณ์ปลายทาง ซึ่งได้เลือกใช้ Google Coral Dev Board เป็นตัวประมวลผลอุปกรณ์ปลายทาง (Edge devices) สามารถสรุปผลการทดลองได้ดังนี้

4.1.1 ผลการใช้โมเดล PoseNet

ในการสกัดเอาข้อมูลโครงกระดูก (Skeleton) จากกล้องวิดีโอจับภาพ ณ เวลาจริง (Real time) เข้ามายัง Coral Dev Board โดยใช้โมเดล PoseNet MobileNet V1 โดยข้อมูลวิดีโอที่นำเข้า (Input size) มีขนาด $3 \times 641 \times 481$ ข้อมูลเอาต์พุตที่ผ่านการ stride (Output Stride) มีขนาด และใช้ 16TF V1 (Tensor flow Version 1) ในการประมวลผลการทำงานทางด้านซอฟต์แวร์ ใช้ภาษาไพธอน (Python 3) ในการเขียนโปรแกรม ใช้ Debian 10 เป็นระบบปฏิบัติการ ทำให้ได้ชุดข้อมูลทั้งหมด จุด คือ 17จุด ,ตาซ้าย ,ตาขวา ,หูซ้าย ,หูขวา ,ไหล่ซ้าย ,ไหล่ขวา ,ข้อศอกซ้าย ,ข้อศอกขวา ,ข้อมือซ้าย ,ข้อมือขวา ,สะโพกซ้าย ,สะโพกขวา ,เข่าซ้าย ,เข่าขวา ,ข้อเท้าซ้าย และ ข้อเท้าขวา นำข้อมูลดิบที่ได้มาคำนวณหาตำแหน่งพิกัดที่ถูกต้องในเฟรมขนาด 480×640 โดยคำนวณจากสมการที่ 8, 9, 10, 11

$$\text{Scaled X-coordinate} = (\text{original X-coordinate} / 640) * \text{scale factor} \quad (8)$$

$$\text{Scaled Y-coordinate} = (\text{original Y-coordinate} / 480) * \text{scale factor} \quad (9)$$

$$\text{Translated X-coordinate} = \text{original X-coordinate} + \text{offset in X-direction} \quad (10)$$

$$\text{Translated Y-coordinate} = \text{original Y-coordinate} + \text{offset in Y-direction} \quad (11)$$

ตัวอย่าง หากต้องการหา offset X-direction ในตำแหน่ง $dx = 100$ พิกเซล และ Y-direction $dy = 50$ พิกเซล จะได้ $\text{scale factor} = 2.0$ และกำหนดให้พิกัด $(x, y) = (320, 240)$ สามารถคำนวณหาค่าได้ดังต่อไปนี้:

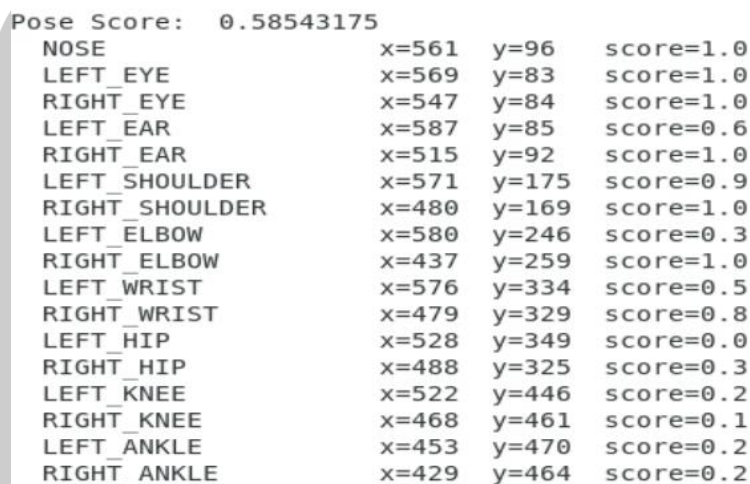
$$\text{Scaled X-coordinate} = (320 / 640) * 2.0 = 1.0$$

$$\text{Scaled Y-coordinate} = (240 / 480) * 2.0 = 1.0$$

Translated X-coordinate = $320 + 100 = 420$

Translated Y-coordinate = $240 + 50 = 290$

ดังนั้น พิกัดใหม่ของจุดหลังการแปลงออฟเซตและมาตราส่วนจะเป็น (420,290)



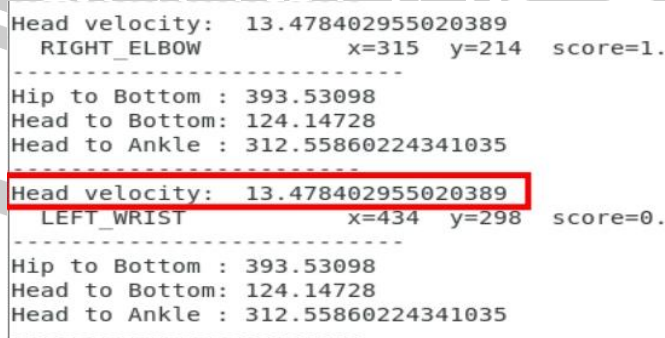
Pose Score: 0.58543175

NOSE	x=561	y=96	score=1.0
LEFT_EYE	x=569	y=83	score=1.0
RIGHT_EYE	x=547	y=84	score=1.0
LEFT_EAR	x=587	y=85	score=0.6
RIGHT_EAR	x=515	y=92	score=1.0
LEFT_SHOULDER	x=571	y=175	score=0.9
RIGHT_SHOULDER	x=480	y=169	score=1.0
LEFT_ELBOW	x=580	y=246	score=0.3
RIGHT_ELBOW	x=437	y=259	score=1.0
LEFT_WRIST	x=576	y=334	score=0.5
RIGHT_WRIST	x=479	y=329	score=0.8
LEFT_HIP	x=528	y=349	score=0.0
RIGHT_HIP	x=488	y=325	score=0.3
LEFT_KNEE	x=522	y=446	score=0.2
RIGHT_KNEE	x=468	y=461	score=0.1
LEFT_ANKLE	x=453	y=470	score=0.2
RIGHT_ANKLE	x=429	y=464	score=0.2

ภาพประกอบ 29 ตัวอย่างผลที่ได้จากการหาตำแหน่งพิกัดในแต่ละเฟรมจากการใช้ PoseNet

4.1.2 ผลการคำนวณหาความเร็วส่วนหัว

ในการหาความเร็วของส่วนหัวนั้นจะใช้เพื่อเป็นตัวตัดสินใจว่า ความเร็วขนาดไหนที่ถือว่าเป็นการล้ม ซึ่งในงานวิจัยนี้ได้หนดค่าเอาไว้ที่ ค่าตั้งแต่ 20 ขึ้นไปถือว่าเป็นต้องระวังเกิดการล้ม และจะต้องนำไปตัดสินใจร่วมกับตำแหน่งของร่างกายในข้อที่ 4.1.3 โดยจะใช้สมการที่ 1 ในการคำนวณหาความเร็ว จะได้ผลลัพธ์ ดังรูปที่ 30



Head velocity: 13.478402955020389
 RIGHT_ELBOW x=315 y=214 score=1.

 Hip to Bottom : 393.53098
 Head to Bottom: 124.14728
 Head to Ankle : 312.55860224341035

 Head velocity: 13.478402955020389
 LEFT_WRIST x=434 y=298 score=0.

 Hip to Bottom : 393.53098
 Head to Bottom: 124.14728
 Head to Ankle : 312.55860224341035

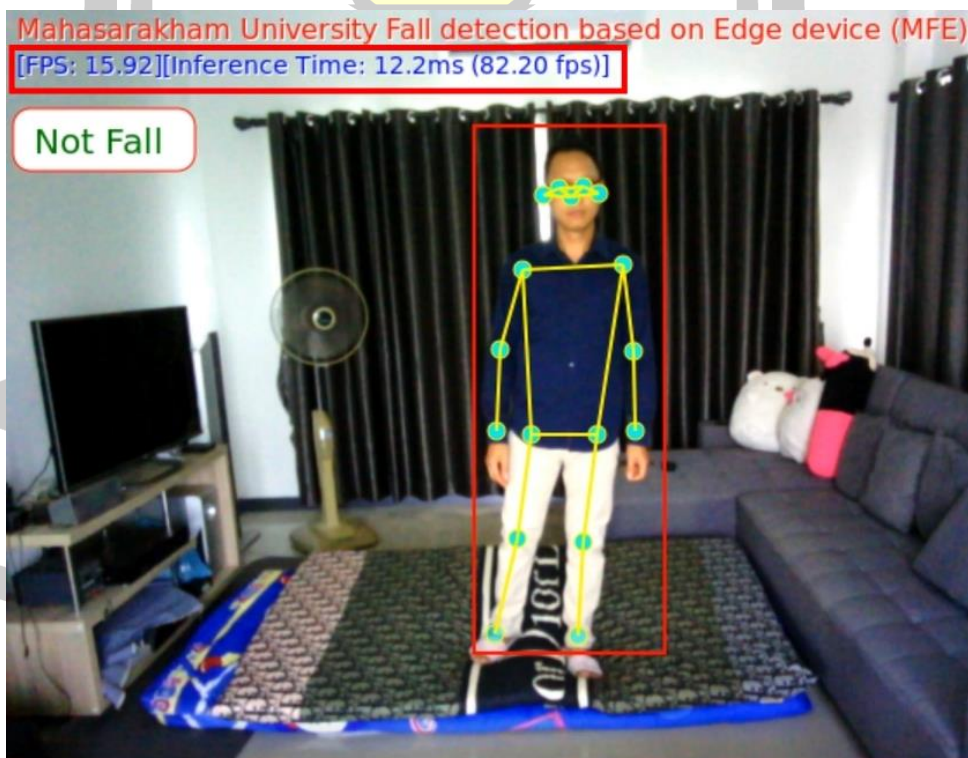
ภาพประกอบ 30 ผลลัพธ์ค่าความเร็วส่วนหัว

4.1.3 ผลการคำนวณหาค่าระยะทางของ Keypoint

ในการหาระยะทางของ keypoint จากจุดหนึ่งไปยังอีกจุดหนึ่ง ในงานวิจัยนี้ ใช้เพื่อเป็นตัวตัดสินใจการล้มในส่วนของตำแหน่งร่างกายเมื่อล้มแล้วจะมีส่วนใดบ้างที่เปลี่ยนแปลงไปในทิศทางของการล้ม จะใช้ทั้งหมด 3 จุด คือ จุดแรกจากส่วนหัวมายังข้อเท้า จะกำหนดค่า Threshold ไว้ที่ 200 จุดที่สองจากตำแหน่งหัวในแนวตั้ง (แกน Y) มายังบริเวณพื้นที่ด้านล่างของเฟรม จะกำหนดค่า Threshold ไว้ที่ 350 และจุดที่สามจากตำแหน่งเอวในแนวตั้ง (แกน Y) มายังบริเวณที่ด้านล่างของเฟรม จะกำหนดค่า Threshold ไว้ที่ 350 โดยจะใช้สมการที่ 2 ในการคำนวณหาค่าผลลัพธ์ที่ได้จะแสดงดังภาพประกอบที่ 31

Pose Score: 0.5177221
 Hip to Bottom : 585.20447
 Head to Bottom: 87.35857
 Head to Ankle : 394.911686964872

ภาพประกอบ 31 ผลที่ได้จากการหาค่าระยะทางในแต่ละเฟรม



ภาพประกอบ 32 ผลที่ได้จากการประมวลผล FPS และ Inference time

ผลการทดสอบประสิทธิภาพ 4.2การตรวจจับการล้มด้วยวิธีการประมวลผลบนอุปกรณ์ที่พ็ย

ผลของการพัฒนาการประยุกต์ใช้ท่าทางเพื่อตรวจจับการล้มของมนุษย์โดยประมวลผลบนอุปกรณ์ปลายทาง โดยการเลือกใช้บอร์ดประมวลผลปัญญาประดิษฐ์ Coral Dev Bord ซึ่งจะใช้ TPU (Tensor flow Processing Unit) ในการประมวลผล จะพบว่า สามารถแสดงอัตราเฟรมเรตไปจนถึงประมาณ 15 เฟรม/วินาที (fps) เวลาในการอนุมาน (Inference time) ประมาณ 12.2 mS หรือประมาณ 82.20 fps ซึ่งถือว่าเพียงพอสำหรับการตรวจจับการล้ม แสดงดังภาพประกอบที่ 30 สามารถทำนายการล้มในวิธีที่ 1 (M1) มีความแม่นยำอยู่ที่ 93% และวิธีที่ 2 (M2) มีความแม่นยำอยู่ที่ 92 % ซึ่งทั้งสองวิธีมีค่าความแม่นยำที่ใกล้เคียงกันมาก

ตาราง 7 ผลการแจ้งเตือนระบบตรวจการหกล้มวิธีที่ 1 (M1)

การทดลองครั้งที่	ท่าล้ม	แจ้งเตือน	ไม่แจ้งเตือน
1	ล้มไปทางซ้าย 20 ครั้ง	19	1
2	ล้มไปทางขวา 20 ครั้ง	18	2
3	ล้มไปทางหลัง 20 ครั้ง	19	1
4	ล้มแนวทะแยง 20 ครั้ง	19	1
5	ล้มมาทางด้านหน้า 20 ครั้ง	18	2
รวม	100 ครั้ง	93	7

จากตารางที่ 7 เป็นผลการทดลองจากวิธีที่ 1 (M1) โดยทำการทดลองออกเป็นท่าทางต่าง ๆ จำนวน 5 ท่า แต่ละท่า ทดลองล้มทั้งหมด 20 ครั้ง ซึ่งสามารถทำนายได้ถูกต้อง 93 ครั้ง มีความแม่นยำอยู่ที่ 93 %

$$\text{Accuracy} = \frac{93}{100} \times 100$$

$$\text{Accuracy} = 93 \%$$

ตาราง 8 ผลการแจ้งเตือนระบบตรวจการหล้มวิธีที่ 2 (M2)

การทดลองครั้งที่	ท่าหล้ม	แจ้งเตือน	ไม่แจ้งเตือน
1	หล้มไปทางซ้าย 20 ครั้ง	19	1
2	หล้มไปทางขวา 20 ครั้ง	18	2
3	หล้มไปทางหลัง 20 ครั้ง	18	2
4	หล้มแนวทะแยง 20 ครั้ง	19	1
5	หล้มมาทางด้านหน้า 20 ครั้ง	18	2
รวม	100 ครั้ง	92	8

จากตารางที่ 8 เป็นผลการทดลองจากวิธีที่ 2 (M2) โดยทำการทดลองออกเป็นท่าทางต่าง ๆ จำนวน 5 ท่า แต่ละท่า ทดลองหล้มทั้งหมด 20 ครั้ง ซึ่งสามารถทำนายได้ถูกต้อง 92 ครั้ง มีความแม่นยำอยู่ที่ 92 %

$$\text{Accuracy} = \frac{92}{100} \times 100$$

$$\text{Accuracy} = 92 \%$$

จากการทดลองทั้งสองวิธี พบว่า สาเหตุที่เครื่องไม่สามารถแจ้งเตือนนั้นเกิดจากการที่ไม่สามารถตรวจจับท่าทางได้เพราะอยู่นอกเหนือตำแหน่งที่กำหนดไว้ในเฟรม และในท่านอนจะตรวจจับได้ค่อนข้างช้าและขาดความแม่นยำ เนื่องจากปัญหาจากโมเดล PoseNet ไม่ใช่ปัญหาจากตัว coral dev board

4.3 การอภิปรายผล

การออกแบบพัฒนา การประยุกต์ใช้ท่าทางเพื่อตรวจจับการหล้มของมนุษย์โดยประมวลผลบนอุปกรณ์ปลายทาง ได้ดำเนินการศึกษาคนควาตามการออกแบบด้วยวิธีการ เอาตำแหน่งจุดต่อของโครงกระดูกมนุษย์ โดยใช้รูปแบบ PoseNet และนำตำแหน่งที่ได้มาทำการประมวลผลเพื่อทำนายว่าหล้มหรือไม่ โดยใช้ Coral dev board ในการประมวลผลเพื่อหารูปแบบตรวจจับการหล้ม สามารถอภิปรายผลได้ดังนี้

4.3.1 การออกแบบการตรวจจับการล้มโดยตัดสินใจจากการประเมินท่าทางของมนุษย์

การประเมินท่าทาง (Pose Estimation) จะทำการดึงข้อมูลจากจุดต่อโครงกระดูก (Skeleton) ของมนุษย์ โดยใช้รูปแบบของ PoseNet ทำให้ได้ข้อมูลที่มี ความจำเป็นในการประเมินท่าทาง ซึ่งข้อดีในการออกแบบในลักษณะนี้ คือ สามารถตอบสนองการล้มแบบ Real time ได้ อย่างรวดเร็ว เนื่องจากใช้ข้อมูลน้อยในการประมวลผล สามารถเลือกใช้ข้อมูลจุดต่อโครงกระดูกตำแหน่งใดก็ได้ในการวิเคราะห์ท่าทาง ประมวลผลได้ทั้งเครื่องคอมพิวเตอร์และอุปกรณ์ปลายทาง ส่วนข้อเสีย นั้น ระบบอาจจะไม่สามารถตรวจสอบการล้มได้ ถ้า PoseNet ไม่สามารถตรวจจับท่าทางของมนุษย์ ซึ่งอาจจะต้องมีการตั้งตำแหน่งของกล้องให้เหมาะสมกับแต่ละสภาพแวดล้อม

4.3.2 การใช้อุปกรณ์ปลายทางในการประมวลผล

อุปกรณ์ปลายทาง (Edge devices) เป็นอุปกรณ์ประมวลผลขนาดเล็กที่อยู่ใกล้กับฝั่งของข้อมูลอินพุตมากที่สุด จะทำหน้าที่ประมวลผลด้วยตัวเอง ในงานวิจัยนี้ได้เลือกใช้ Google Coral dev board ซึ่งมีข้อดีคือ สามารถทำงานได้อย่างรวดเร็วเป็นที่น่าพอใจ ประมวลผลได้ในเวลาจริง สามารถตรวจจับคนได้อย่างแม่นยำ มีขนาดเล็กสามารถนำไปติดตั้งได้ง่าย ส่วนข้อเสีย นั้น เนื่องจากเป็นอุปกรณ์ที่ประมวลผลปลายทางจึงมีทรัพยากรที่น้อยต้องใช้อย่างจำกัด



บทที่ 5

สรุปผล และข้อเสนอแนะ

5.1 สรุปผล

การวิจัยครั้งนี้เป็นการศึกษาพัฒนาการประยุกต์ใช้ท่าทางเพื่อตรวจจับการล้มของมนุษย์โดยประมวลผลบนอุปกรณ์ปลายทาง โดยผู้วิจัยได้สรุปผล และข้อเสนอแนะตามลำดับดังนี้

5.1.1 ผลการทดสอบประสิทธิภาพ วิธีที่ 1 (M1)

โดยแบ่งการทดลองออกเป็น 5 ท่า ท่าละ 20 ครั้ง พบว่า ท่าที่ 1 ล้มไปทางซ้าย 20 ครั้ง มีการแจ้งเตือน 19 ไม่มีการแจ้งเตือน 1 ครั้ง ท่าที่ 2 ล้มไปทางขวา 20 ครั้ง มีการแจ้งเตือน 18 ไม่มีการแจ้งเตือน 2 ครั้ง ท่าที่ 3 ล้มไปทางหลัง 20 ครั้ง มีการแจ้งเตือน 19 ครั้ง ไม่มีการแจ้งเตือน 1 ครั้ง ท่าที่ 4 ล้มแนวทะแยง 20 ครั้ง มีการแจ้งเตือน 19 ครั้ง ไม่มีการแจ้งเตือน 1 ท่าที่ 5 ล้มมาทางด้านหน้า 20 ครั้ง มีการแจ้งเตือน 18 ไม่มีการแจ้งเตือน 2 ครั้ง รวมการแจ้งเตือนทั้งหมด คิดเป็นร้อยละ 93 และไม่มีการแจ้งเตือนทั้งหมด เตือน 7 ครั้ง คิดเป็นร้อยละ 7 ดังนั้นสรุปได้ว่าลักษณะการล้มของมนุษย์ มีผลต่อค่าค่าความเร็วและท่าทางในการล้ม เนื่องจาก ลักษณะการลมนั้นขึ้นอยู่กับเวลาการเคลื่อนที่ของร่างกาย ถ้าเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งในระยะเวลาสั้นหรือแต่มีระยะทางการเคลื่อนไหวยาวและตำแหน่งของร่างกายในลักษณะหัวกับเท้าอยู่ในระนาบเดียวกัน แสดงว่าเกิดการล้ม

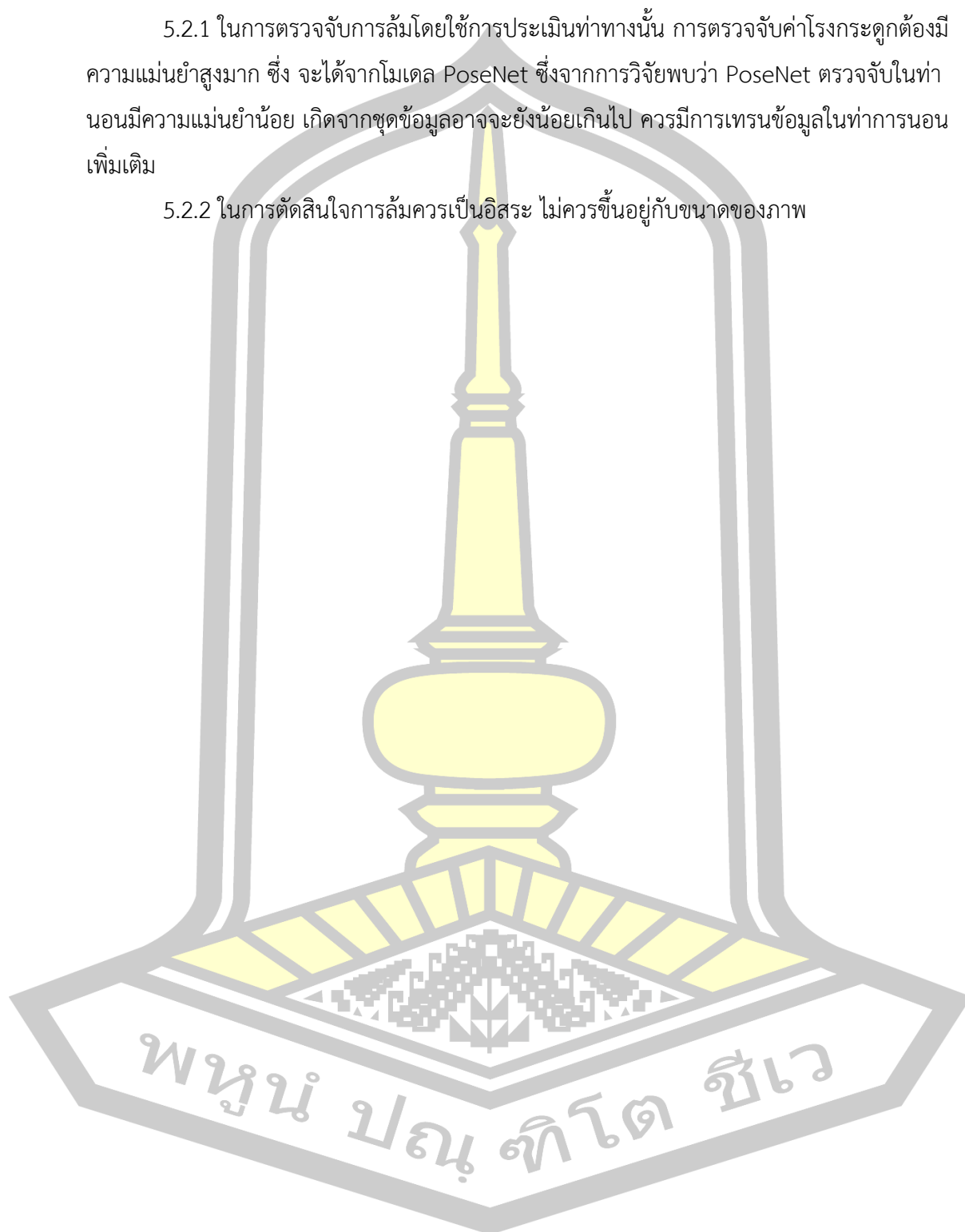
5.1.2 ผลการทดสอบประสิทธิภาพ วิธีที่ 2 (M2)

โดยแบ่งการทดลองออกเป็น 5 ท่า ท่าละ 20 ครั้ง พบว่า ท่าที่ 1 ล้มไปทางซ้าย 20 ครั้ง มีการแจ้งเตือน 19 ไม่มีการแจ้งเตือน 1 ครั้ง ท่าที่ 2 ล้มไปทางขวา 20 ครั้ง มีการแจ้งเตือน 18 ไม่มีการแจ้งเตือน 2 ครั้ง ท่าที่ 3 ล้มไปทางหลัง 20 ครั้ง มีการแจ้งเตือน 18 ครั้ง ไม่มีการแจ้งเตือน 2 ครั้ง ท่าที่ 4 ล้มแนวทะแยง 20 ครั้ง มีการแจ้งเตือน 19 ครั้ง ไม่มีการแจ้งเตือน 1 ท่าที่ 5 ล้มมาทางด้านหน้า 20 ครั้ง มีการแจ้งเตือน 18 ไม่มีการแจ้งเตือน 2 ครั้ง รวมการแจ้งเตือนทั้งหมด คิดเป็นร้อยละ 92 และไม่มีการแจ้งเตือนทั้งหมด เตือน 8 ครั้ง คิดเป็นร้อยละ 8 ดังนั้นสรุปได้ว่าลักษณะการล้มของมนุษย์ มีผลต่อค่าค่าความเร็วและท่าทางในการล้ม และตำแหน่งของร่างกายในลักษณะหัวกับเท้าอยู่ในระนาบเดียวกันหรือหัวสูงกว่าเท้าเล็กน้อย แสดงว่าเกิดการล้ม

5.2 ข้อเสนอแนะ

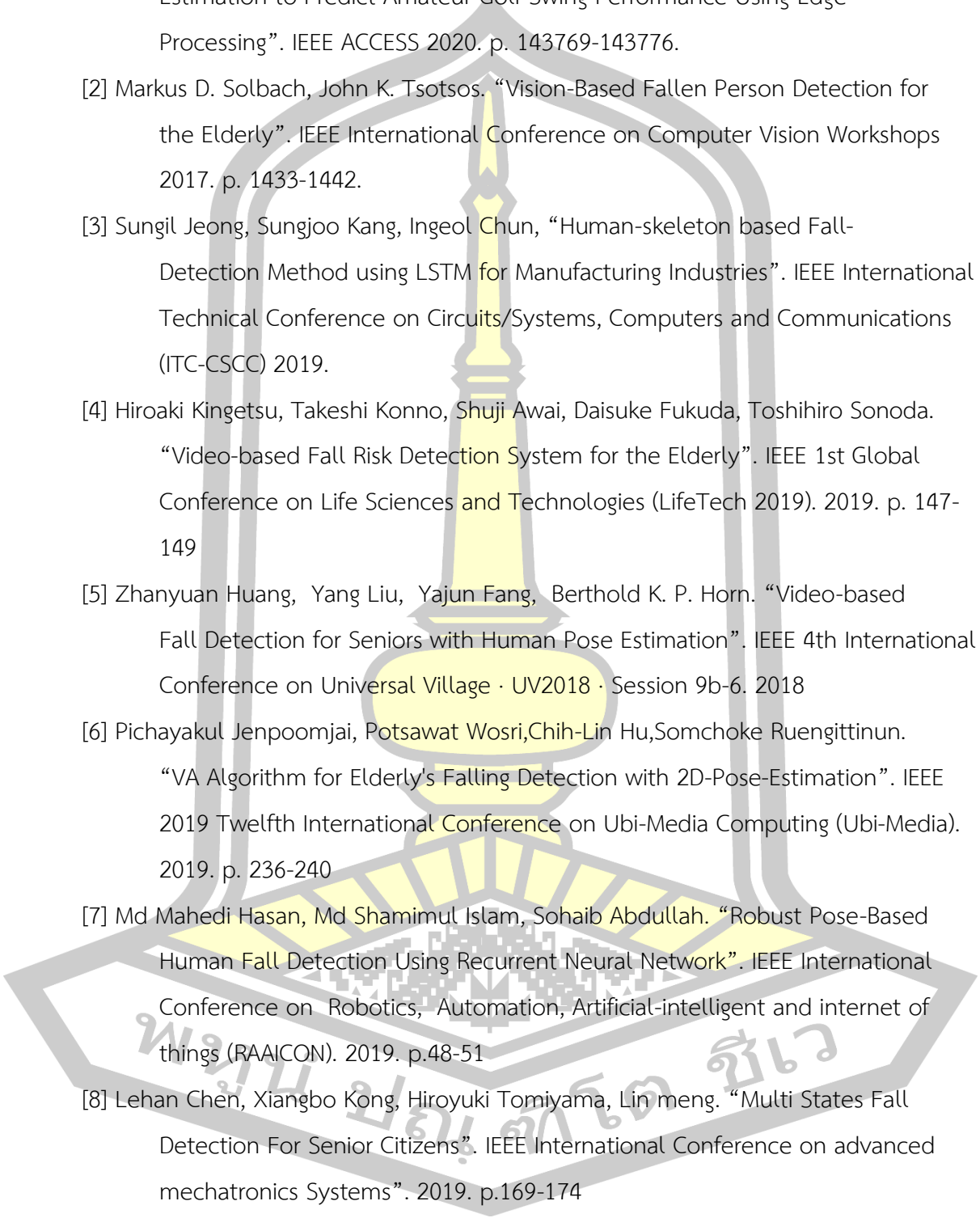
5.2.1 ในการตรวจจับการล้มโดยใช้การประเมินท่าทางนั้น การตรวจจับค่าโรงกระดูกต้องมีความแม่นยำสูงมาก ซึ่งจะได้จากโมเดล PoseNet ซึ่งจากการวิจัยพบว่า PoseNet ตรวจจับในท่านอนมีความแม่นยำน้อย เกิดจากชุดข้อมูลอาจยังน้อยเกินไป ควรมีการเทรนข้อมูลในท่าการนอนเพิ่มเติม

5.2.2 ในการตัดสินใจการล้มควรเป็นอิสระ ไม่ควรขึ้นอยู่กับขนาดของภาพ



บรรณานุกรม



- 
- [1] THEODORE T. KIM , MOHAMED A. ZOHDY, MICHAEL P. BARKER. “Applying Pose Estimation to Predict Amateur Golf Swing Performance Using Edge Processing”. IEEE ACCESS 2020. p. 143769-143776.
- [2] Markus D. Solbach, John K. Tsotsos. “Vision-Based Fallen Person Detection for the Elderly”. IEEE International Conference on Computer Vision Workshops 2017. p. 1433-1442.
- [3] Sungil Jeong, Sungjoo Kang, Ingeol Chun, “Human-skeleton based Fall-Detection Method using LSTM for Manufacturing Industries”. IEEE International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC) 2019.
- [4] Hiroaki Kingetsu, Takeshi Konno, Shuji Awai, Daisuke Fukuda, Toshihiro Sonoda. “Video-based Fall Risk Detection System for the Elderly”. IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech 2019). 2019. p. 147-149
- [5] Zhanyuan Huang, Yang Liu, Yajun Fang, Berthold K. P. Horn. “Video-based Fall Detection for Seniors with Human Pose Estimation”. IEEE 4th International Conference on Universal Village · UV2018 · Session 9b-6. 2018
- [6] Pichayakul Jenpoomjai, Potsawat Wosri, Chih-Lin Hu, Somchoke Ruengittinun. “VA Algorithm for Elderly's Falling Detection with 2D-Pose-Estimation”. IEEE 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media). 2019. p. 236-240
- [7] Md Mahedi Hasan, Md Shamimul Islam, Sohaib Abdullah. “Robust Pose-Based Human Fall Detection Using Recurrent Neural Network”. IEEE International Conference on Robotics, Automation, Artificial-intelligent and internet of things (RAAICON). 2019. p.48-51
- [8] Lehan Chen, Xiangbo Kong, Hiroyuki Tomiyama, Lin meng. “Multi States Fall Detection For Senior Citizens”. IEEE International Conference on advanced mechatronics Systems”. 2019. p.169-174
- [9] Jiawei Li, Shu-Tao Xia, Qianggang Ding. “Multi-level recognition on falls from activities of daily living”. Proceedings of the 2020 on International Conference on Multimedia Retrieval, ICMR 2020, . 2020. pages 464-471

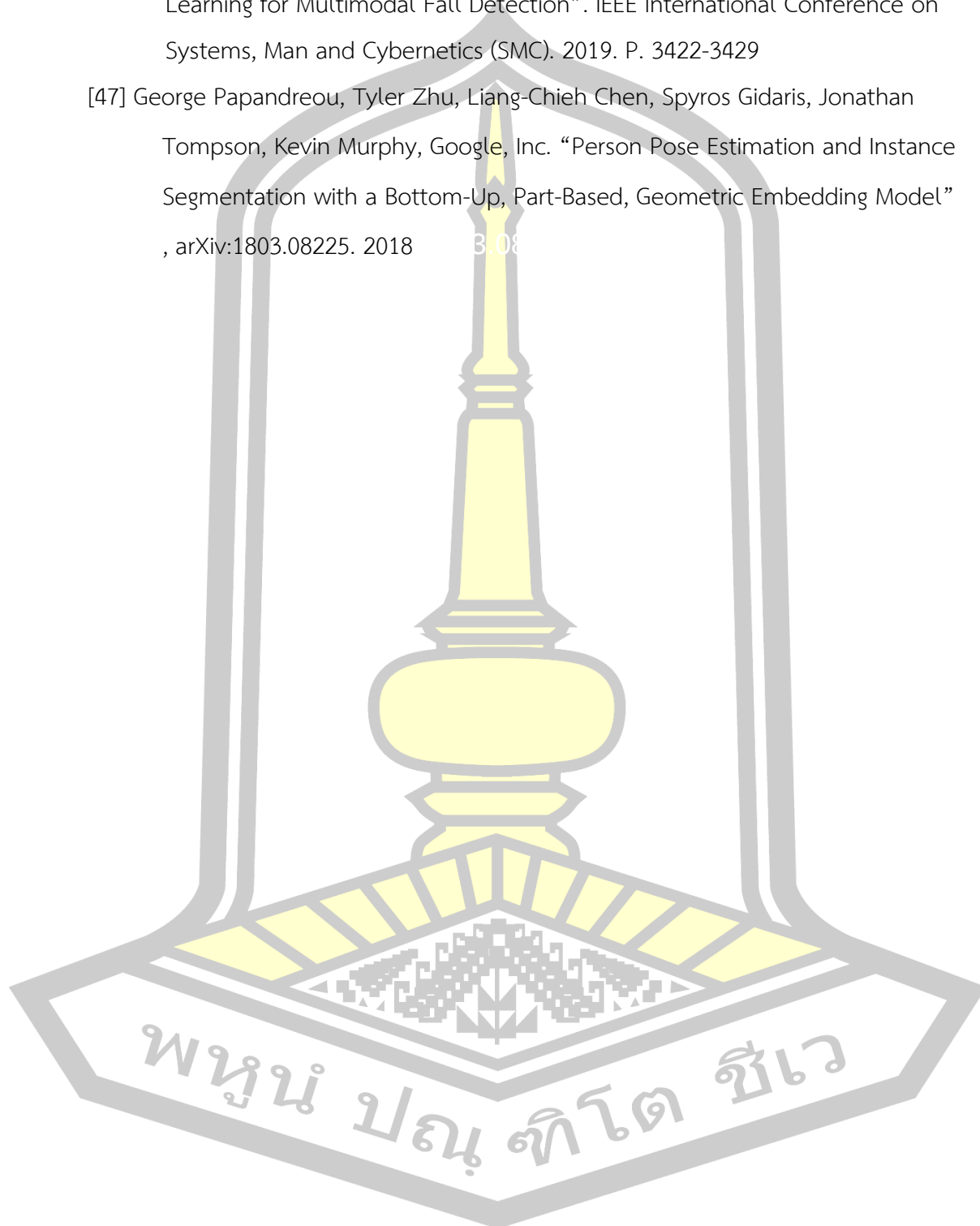
- [10] TSUNG-HAN TSAI, CHIN-WEI HSU. "Implementation of Fall Detection System Based on 3D Skeleton for Deep Learning Technique". IEEE Access Volume: 7. 2019. p.153049 - 153059
- [11] Wen-Yu Cai, Jia-Hao Guo, Mei-Yan Zhang, Zhi-Xiang Ruan, Xue-Chen Zheng, Shuai Shuai Lv. "GBDT-Based Fall Detection with Comprehensive Data from Posture Sensor and Human Skeleton Extraction". Hindawi Journal of Healthcare Engineering Volume 2020, Article ID 8887340. 2020. p. 1-15.
- [12] Xiao L, Wanmi Chen. "Fall recognition algorithm for the elderly based on home service robot". IEEE International Conference on Electronics, Circuits and Information Engineering (ECIE). 2021. p. 329-335
- [13] Divya R, Riya T B , Rona Johns P, Sreelakshmi T J, Theres Davies. "Fall Detection using OpenPose". International Journal of Innovative Science and Research Technology. 2021. p.1340-1343
- [14] Yangsen Chen, Rongxi Du, Kaitao Luo, Yuheng Xiao. "Fall detection system based on real-time pose estimation and SVM". IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE 2021). 2021. p. 990-993
- [15] Guilherme Vieira Leite, Gabriel Pellegrino da Silva, Helio Pedrini. "Fall Detection in Video Sequences Based on a Three-Stream Convolutional Neural Network". IEEE International Conference on Machine Learning and Applications (ICMLA). 2019. p. 191-195
- [16] Vincenzo Dentamaro, Donato Impedovo, Giuseppe Pirlo. "Fall Detection by Human Pose Estimation and Kinematic Theory". IEEE International Conference on Pattern Recognition (ICPR). 2020. p. 2328-2335
- [17] Yoon-Kyu Kang, Hee-Yong Kang, Dal-Soo Weon. "Fall Detection Based on Human Skeleton Keypoints Using GRU". International Journal of Internet, Broadcasting and Communication. 2020. p. 83-92
- [18] Bo-Hua Wang, Jie Yu, Kuo Wang, Xuan-Yu Bao, Ke-Ming Mao. "Fall Detection Based on Dual-Channel Feature Integration". IEEE Access. 2020. p. 103443-103453

- [19] Yin Zheng, Dongping Zhang, Li Yanga, Zhihong Zhou. "Fall detection and recognition based on GCN and 2D Pose". IEEE International Conference on Systems and Informatics (ICSAI 2019). 2019. p. 558-562
- [20] Guangmin Sun, Zhongqi Wang, "Fall detection algorithm for the elderly based on human posture estimation". IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). 2020. p. 172-176
- [21] Xiangbo Konga, Takeshi Kumakia, Lin Menga, Hiroyuki Tomiyamaa. "A Skeleton Analysis Based Fall Detection Method Using ToF Camera". ScienceDirect International Conference on Identification, Information and Knowledge in the internet of Things. 2020. p. 252-257
- [22] Cuiwei Liu, Jianxiong Lv, Xiaoxue Zhao, Zhaokui Li, Zhuo Yan, Xiangbin Shi. "A Novel Key Point Trajectory Model for Fall Detection from RGB-D Videos". IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2021, p. 1021-1026
- [23] Yves M. Galvao, Janderson Ferreira, Vinicius A. Albuquerque, Pablo Barros, Bruno J.T. Fernandes. "A multimodal approach using deep learning for fall detection". ScienceDirect Expert Systems With Applications. 2021. p. 1-9
- [24] Zhen-Peng Bian, Junhui Hou, Lap-Pui Chau, Senior, Nadia Magnenat-Thalmann. "Fall Detection Based on Body Part Tracking Using a Depth Camera". IEEE Journal of Biomedical and Health Informatics. 2014. p. 1-10
- [25] Keondo Park, Wonyoung Jang, Woochul Lee, Kisung Nam, Kihong Seong, Kyuwook Chai, Wen-Syan Li. "Real-time Mask Detection on Google Edge TPU". arXiv:2010.04427. 2020. p. 1-7
- [26] Ahmad Ammar Asyraaf Jainuddin, Yew Cheong Hou, Mohd Zafri Baharuddin, Salman Yussof. "Performance Analysis of Deep Neural Networks for Object Classification with Edge TPU". "Performance Analysis of Deep Neural Networks for Object Classification with Edge TPU" IEEE 2020 8th International Conference on Information Technology and Multimedia (ICIMU), 2020. p. 323-328

- [27] Jonah Sengupta, Rajkumar Kubendran, Emre Neftci, Andreas Andreou. "High-Speed, Real-Time, Spike-Based Object Tracking and Path Prediction on Google Edge TPU". IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS). 2020. p.134-135
- [28] Y. Hsieh, Y. Jeng. "Development of Home Intelligent Fall Detection IoT System Based on Feedback Optical Flow Convolutional Neural Network," in IEEE Access, vol. 6. 2018. p. 6048-6057
- [29] A. Dragomir, G. Simion, I. Ermalai. "Smart Mirror for Skin Type Identification". 2020 International Symposium on Electronics and Telecommunications (ISETC). 2020, p. 1-4.
- [30] S. Hosseininoorbin, S. Layeghy, B. Kusy, R. Jurdak, M. Portmann, "Scaling Spectrogram Data Representation for Deep Learning on Edge TPU" . IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). 2021. p. 572-578
- [31] Seyedehfazez Hosseininoorbin, Siamak Layeghya, Mohanad Sarhana, Raja Jurdak, Marius Portmann. " Exploring Edge TPU for Network Intrusion Detection in IoT". 2021.
- [32] Kraft, Dimitri, Karthik Srinivasan, Gerald Bieber. "Deep Learning Based Fall Detection Algorithms for Embedded Systems, Smartwatches, and IoT Devices Using Accelerometers" .2020. Technologies 8, no. 4: 72.
- [33] Dharmitha Ajerla, Sazia Mahfuz, Farhana Zulkernine. "A Real-Time Patient Monitoring Framework for Fall Detection". Hindawi Wireless Communications and Mobile Computing Volume 2019, Article ID 9507938.2019. p.1-14.
- [34] Jun Xu, Zunwen He, Yan Zhang. "CNN-LSTM Combined Network for IoT Enabled Fall Detection Applications". Journal of Physics: Conference Series, Volume 1267, 2019 3rd International Conference on Artificial Intelligence, Automation and Control Technologies (AIAC 2019). 2019.
- [35] A. H. Fakhruddin, X. Fei and H. Li. "Convolutional neural networks (CNN) based human fall detection on Body Sensor Networks (BSN) sensor data". IEEE 2017 4th International Conference on Systems and Informatics (ICSAI). 2017. p. 1461-1465

- [36] Na Lu, Yidan Wu, Li Feng, Jinbo Song. "Deep Learning for Fall Detection: 3D-CNN Combined with LSTM on Video Kinematic Data". IEEE Journal of Biomedical and Health Informatics. 2018. p. 1-10
- [37] Maitre J, Bouchard K, Gaboury S. "Fall Detection With UWB Radars and CNN-LSTM Architecture". IEEE J Biomed Health Inform. 2021. p. 1273-1283.
- [38] Jiang Wu, Jiale Wang, Ao Zhan, Chengyu Wu. "Fall Detection with CNN-Casual LSTM Network". Information. 2021.
- [39] Yuxi Chen, Xiangbo Kong, Lin Meng, Hiroyuki Tomiyama. "An Edge Computing Based Fall Detection System for Elderly Persons". ScienceDirect International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI2019) Procedia Computer Science, Volume 174. 2020. p. 9-14
- [40] YANG Yi-fei, HU Qing-chun, DAI Ming-ming, YAN Hong-ri, LING Jin-yang. "Design and Implementation of Fall Detection System Based on Video". IEEE International Conference on Computer Engineering and Application (ICCEA). 2020. p. 893-899
- [41] Mohammed Farsi, "Application of ensemble RNN deep neural network to the fall detection through IoT environment". Elsevier, Alexandria Engineering Journal, Volume 60, Issue 1. 2021. p.199-211
- [42] Sowmyayani S, Murugan V, Kavitha J. "Fall Detection in Elderly Care System Based on Group of Pictures". OPEN ACCESS Vietnam Journal of Computer Science. 2021. P.199-214
- [43] Dariusz Mrozek, Anna Koczur, Bożena Małysiak-Mrozek. "Fall detection in older adults with mobile IoT devices and machine learning in the cloud and on the edge". Information Sciences, Volume 537. 2020. p. 132-147
- [44] Adrián Núñez-Marcos, Gorka Azkune, Ignacio Arganda-Carreras. "Vision-Based Fall Detection with Convolutional Neural Networks". Hindawi Wireless Communications and Mobile Computing. Volume 2017, Article ID 9474806
- [45] A. M. Kist, M. Döllinger. "Efficient Biomedical Image Segmentation on EdgeTPUs at Point of Care". in IEEE Access, vol. 8.2020.p. 139356-139366

- [46] Lourdes Martinez-Villasenor, Hiram Ponce, Karina Perez-Daniel. “Deep Learning for Multimodal Fall Detection”. IEEE International Conference on Systems, Man and Cybernetics (SMC). 2019. P. 3422-3429
- [47] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, Kevin Murphy, Google, Inc. “Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model” , arXiv:1803.08225. 2018



ประวัติผู้เขียน

ชื่อ	นายพรชัย ทองอินทร์
วันเกิด	17 มกราคม 2517
สถานที่เกิด	จ. ศรีสะเกษ
สถานที่อยู่ปัจจุบัน	502 หมู่ 12 ต. เกิง อ. เมือง จ. มหาสารคาม
ตำแหน่งหน้าที่การงาน	ครู
สถานที่ทำงานปัจจุบัน	วิทยาลัยเทคนิคมหาสารคาม
ประวัติการศึกษา	พ.ศ. 2552 ปริญญาวิทยาศาสตรมหาบัณฑิต (วท.ม.) สาขาวิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยขอนแก่น พ.ศ. 2567 ปริญญาปรัชญาดุษฎีบัณฑิต (ปร.ด.) สาขาวิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มหาวิทยาลัยมหาสารคาม

พูน ปณ ภัโต ชีเว