บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงขั้นตอนวิธีการวิเคราะห์ภาพวีดีโอการล้ม และเครื่องมือที่ใช้ในการสร้างแบบจำลองการเรียนรู้ของเครื่อง ด้วยโปรแกรม Visual studio code และ Google Colab

3.1 เครื่องมือที่ใช้ในการทำวิจัย

ตารางที่ 3.1 เครื่องมือ วัสดุและอุปกรณ์ที่ใช้ในการวิจัย

เครื่องมือ	
1.	เครื่องคอมพิวเตอร์ LAPTOP-8FES7VFD -หน่วยประมวลผลกลาง: AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz -หน่วยความจำ: 8.00 GB -หน่วยประมวลผลกราฟฟิก: NVIDIA GeForce GTX -ระบบปฏิบัติการ: Windows 10
โปรแกรม	
2.	โปรแกรมคอมพิวเตอร์ Vision studio code version
3.	โปรแกรมภาษาไพธอน : Google Colab

3.2 ข้อมูลที่ใช้ในการศึกษา

วีดีโอจำลองการล้ม ประกอบไปด้วย วีดีโอจำลองการล้ม 4 แบบ คือ การล้มคว่ำหน้า การล้มหงายหลัง การล้มไปทางซ้าย การล้มไปทางขวา และวิดีโอการเดิน เป็นข้อมูลประเภท avi จาก Multiple cameras fall data set , Dataset for human fall recognition in an uncontrolled environment และ Fall Video http://foe.mmu.edu.my/digitalhome/FallVideo.zip มีรายละเอียดดังตารางที่ 3.2

ตารางที่ 3.2 วีดีโอจำลองการล้มและการเดินที่ใช้ในการศึกษา

ชุดข้อมูล	ปี	รูปแบบไฟล์	การควบคุมสิ่งแวดล้อม	จำนวนวีดีโอในชุดข้อมูล
Multiple cameras fall data set	2010	avi	มี	192
FallVideo	2013	avi	มี	21
Dataset for human fall recognition in an uncontrolled environment	2022	avi	ไม่มี	100

3.3 ขั้นตอนการดำเนินงานวิจัย

ขั้นตอนวิธีการดำเนินงานแบ่งออกเป็น 5 หัวข้อหลัก ๆ คือ

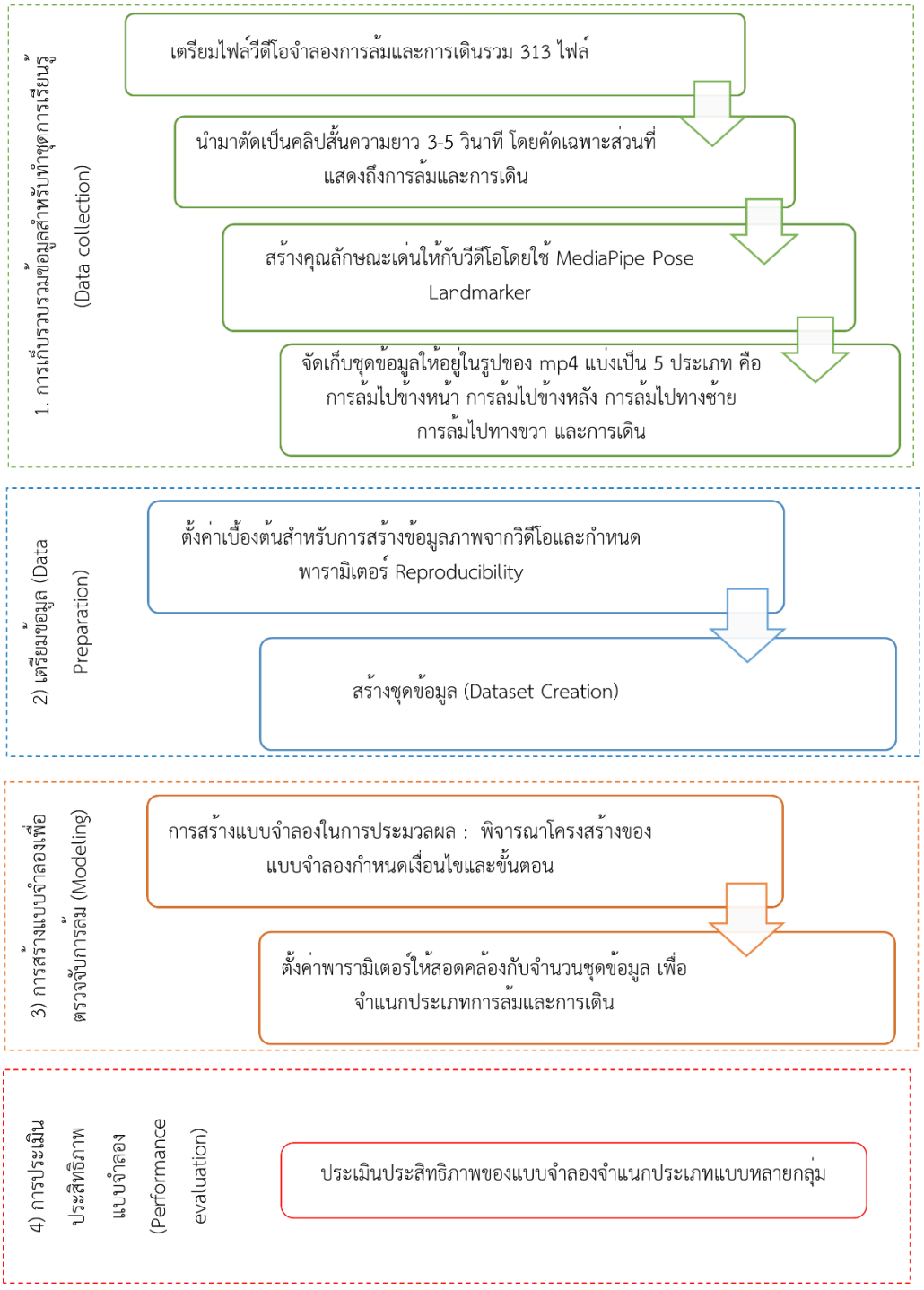
3.3.1 การเก็บรวบรวมข้อมูลสำหรับทำชุดการเรียนรู้ (Data collection)

3.3.2 การเตรียมข้อมูลสำหรับทำชุดการเรียนรู้ (Data preparation)

3.3.3 การสร้างแบบจำลองเพื่อตรวจจับการล้ม (Modeling)

3.3.4 การประเมินประสิทธิภาพแบบจำลอง

3.3.5 การทดสอบแบบจำลอง



รูปที่ 3.1 แสดงแผนผังขั้นตอนในการวิจัยสร้างแบบจำลองจำแนกแบบหลายกลุ่มโดยใช้การเรียนรู้ของเครื่อง

3.4 การรวบรวมข้อมูล (Data collection)

การเตรียมข้อมูลวีดีโอ นำไฟล์วีดีจำลองการล้มและการเดิน 313 ไฟล์ มาทำการตัดเป็นคลิปสั้นๆ ความยาวระหว่าง 3-7 วินาที แบ่งเป็นวีดีโอจำลองการล้ม 4 แบบ และวีดีโอแสดงการเดิน จากนั้นจัดกลุ่มไฟล์โดยการตั้งชื่อให้เป็นประเภท และโฟลเดอร์เดียวกัน จากนั้นนำข้อมูลมามาแบ่งเป็น 2 ชุด คือชุดข้อมูลวีดีโอปกติและชุดข้อมูลที่เพิ่มการประมาณการท่าทางด้วย MediaPipe ชุดข้อมูล 2 ชุดประกอบด้วย วีดีโอแสดงการล้ม 180 ไฟล์ แบ่งเป็นการล้มไปข้างหน้า 62 ไฟล์ การล้มข้างหลัง 32 ไฟล์ การล้มไปทางซ้าย 35 การล้มไปทางขวา 51 ไฟล์ และวีดีโอแสดงการเดิน 60 ไฟล์ รวมข้อมูลให้อยู่ในไฟล์ .rar นำเข้า google drive เพื่อเตรียมสร้างแบบจำลองในขั้นตอนต่อไป

3.5 การสร้างแบบจำลอง (Modeling)

การสร้างแบบจำลองในการตรวจจับการล้ม โดยใช้โปรแกรม Google Colab ขั้นตอนและวิธีการสร้างแบบจำลองมีดังนี้

3.5.1 เตรียมสิ่งแวดล้อม (Environment Setup)

3.5.1.1 นำเข้าไลบรารีที่จำเป็น เช่น TensorFlow, keras, NumPy, cv2, matplotlib, sklearn เป็นต้น

3.5.1.2 Mount Google Drive เพื่อเข้าถึงไฟล์ Dataset

3.5.1.3 แยกไฟล์ .rar เพื่อเตรียมข้อมูลมาใช้ในการเทรนแบบจำลอง

3.5.2 เตรียมข้อมูล (Data Preparation)

3.5.2.1 ตั้งค่าเบื้องต้น: กำหนด seed เพื่อให้การทำซ้ำได้ผลเหมือนกัน

3.5.2.2 ตรวจสอบข้อมูล โดยการสุ่มเลือกวิดีโอจากแต่ละคลาส และแสดง frame ตัวอย่าง

3.5.3.3 กำหนดพารามิเตอร์

1) กำหนดขนาดภาพเป็น 64x64

2) กำหนดจำนวนสูงสุดของรูปต่อคลาสเป็น 8000

3) กำหนดรายชื่อคลาส ดังนี้ Falling Backward, Falling Forward, Falling left Falling right และ Walking

3.5.3 สร้างชุดข้อมูล (Dataset Creation)

ทำการอ่านข้อมูลวีดีโอทีละเฟรมและปรับขนาดเป็น 64x64 และทำการ Normalize pixel ให้อยู่ระหว่าง 0-1 จากนั้นสร้างป้ายกำกับ (Label) และสุ่มเฟรมมาเก็บสูงสุด 8000 ภาพต่อคลาส สุดท้ายแปลง labels เป็น one-hot encoding

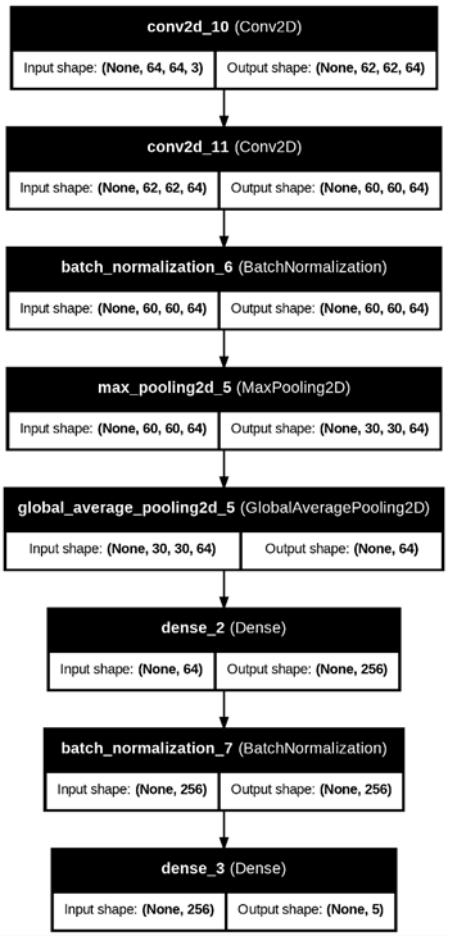
3.5.4 แบ่งชุดข้อมูล (Splitting Dataset)

ทำการแบ่งข้อมูลออกเป็น ชุดฝึก (Train) และ ชุดทดสอบ (Test) โดยกำหนดให้ 20 เปอร์เซ็นต์ของข้อมูลทั้งหมดเป็นชุดทดสอบ

3.5.5 สร้างแบบจำลอง (Model Creation)

3.5.5.1 แบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ 2 เลเยอร์ แผนผังโครงสร้างเทคนิควิธีโครงข่ายประสาทเทียมแบบ 2 เลเยอร์ แสดงในภาพที่ 3.2 แบบจำลองนี้ได้การใช้งาน Batch Normalization และ Global Average Pooling มีรายละเอียดโครงสร้างดังนี้

1) ชั้น Convolution



รูปที่ 3.2 แสดงโครงสร้างแบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ 2 เลเยอร์

ชั้นที่ 1 (conv2d_10)

1) รับภาพขนาด 64x64x3

2) ใช้ตัวกรองขนาด (3x3) จำนวน 64 filters

3) ได้ผลลัพธ์ภาพขนาด 62x62x64

ชั้นที่ 2 (conv2d_11)

1) รับภาพจากชั้นแรก ขนาด 62x62x64

2) ใช้ตัวกรองขนาด (3x3) จำนวน 64 ตัวกรอง

3) ได้ผลลัพธ์ภาพขนาด 60x60x64

2) Batch Normalization

มีการใช้ Batch Normalization เพื่อช่วยให้การฝึกโมเดลเร็วขึ้น ลดปัญหา Vanishing Gradient ได้ผลลัพธ์ภาพขนาดยังคงเดิม 60x60x64

3) ชั้น MaxPooling

ชั้นนี้จะทำการลดขนาด feature maps ลงครึ่งหนึ่ง จาก 60x60 เป็น 30x30 ได้ผลลัพธ์ของภาพ ขนาด 30x30x64

4) Global Average Pooling

สำหรับฟังก์ชั่นนี้จะลดขนาดจาก Feature maps เหลือแค่เวกเตอร์ โดยการหาค่าเฉลี่ยแต่ละ feature map ซึ่งช่วยลดความซับซ้อน และลดการ Overfitting ของแบบจำลอง

5) ชั้น Fully Connected Layers

ชั้นนี้ประกอบไปด้วย 3 ส่วน คือ Dense ชั้นที่ 1 (dense_2) เพื่อเพิ่มความสามารถในการจำแนกโดยใช้ nodes จำนวน 256 nodes ส่วนที่ 2 คือการทำ Batch Normalization อีกครั้ง เพื่อช่วยให้ชั้น Dense ฝึกได้เสถียรขึ้น และส่วนสุดท้าย คือOutput Layer (dense_3) มี 5 nodes ตามจำนวนคลาส และใช้ SoftMax activation

3.5.5.2 แบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ VGG แบบจำลองนรูปที่ 3.3 คือแบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ VGG16 ดัดแปลง ซึ่งปรับปรุงเพื่อลดความซับซ้อนและเหมาะกับข้อมูลที่เล็กกว่าเดิม เพิ่มประสิทธิภาพด้วย Global Average Pooling และ Batch Normalization มีรายละเอียดโครงสร้าง ดังนี้

1) Input Layer

รับภาพขนาด (64, 64, 3) ซึ่งหมายถึง รูปขนาด 64x64 พิกเซล และมี 3 channels (RGB)

2) Convolutional Blocks

โครงสร้างประกอบด้วย 5 บล็อกใหญ่ แต่ละบล็อกมี Convolution layer หลายชั้น ตามด้วย MaxPooling layer รายละเอียดแต่ละบล็อกแสดงในตารางที่ 3.3

ตารางที่ 3.2 แสดงจำนวนชั้น Convolutional Blocks,ขนาดตัวกรองและขนาดภาพผลลัพธ์

Block	Convolutional Layers	Filter Size	Output Size
1	2 layers	64 filters (3x3)	(32x32x64)
2	2 layers	128 filters (3x3)	(16x16x128)
3	3 layers	256 filters (3x3)	(8x8x256)
4	3 layers	512 filters (3x3)	(4x4x512)
5	3 layers	512 filters (3x3)	(2x2x512)

3) GlobalAveragePooling2D

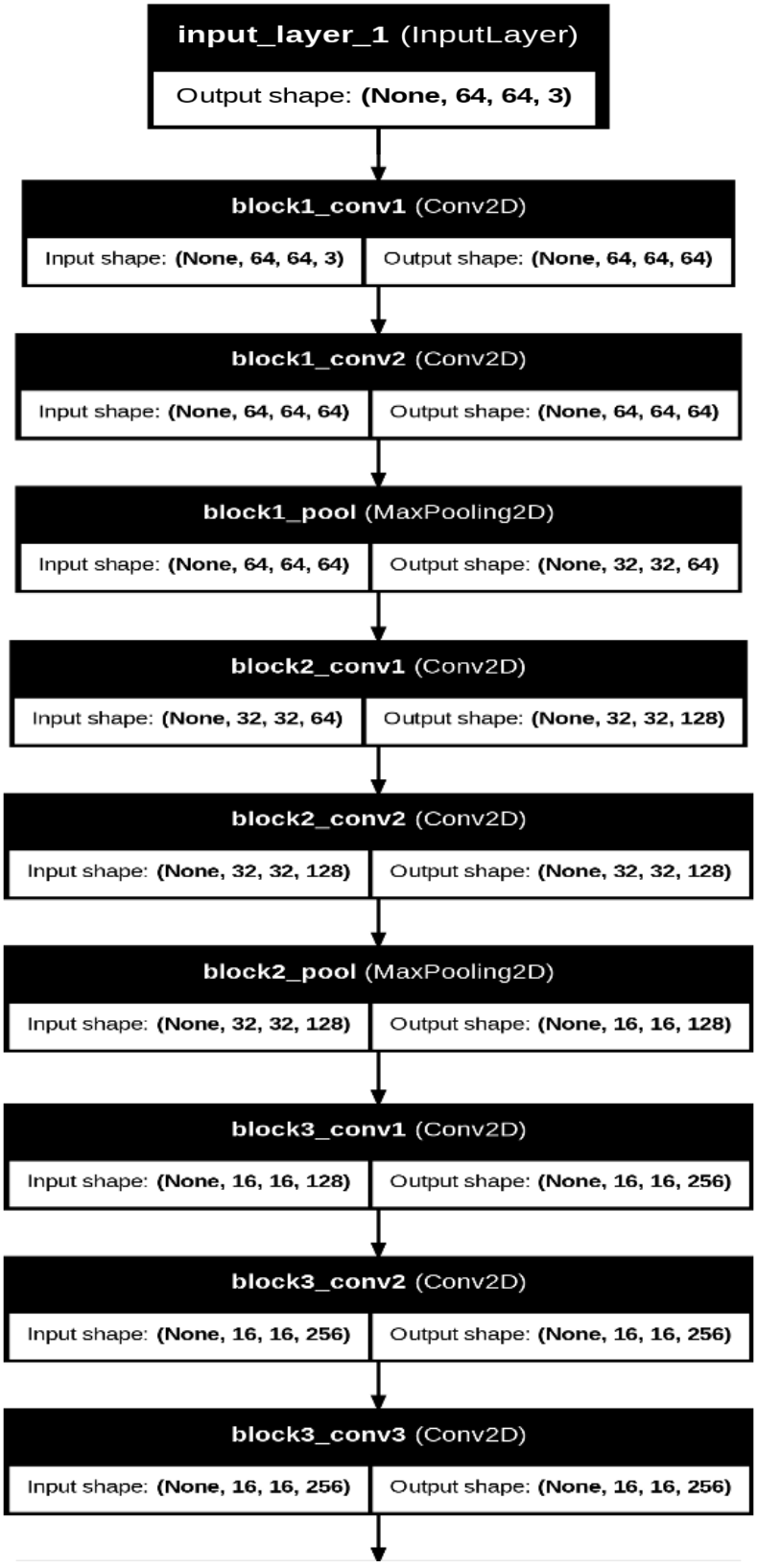
ใช้ Global Average Pooling ลดขนาด feature map จาก (2, 2, 512) เป็น vector ขนาด 512

4) Fully Connected Layers (Dense Layers)

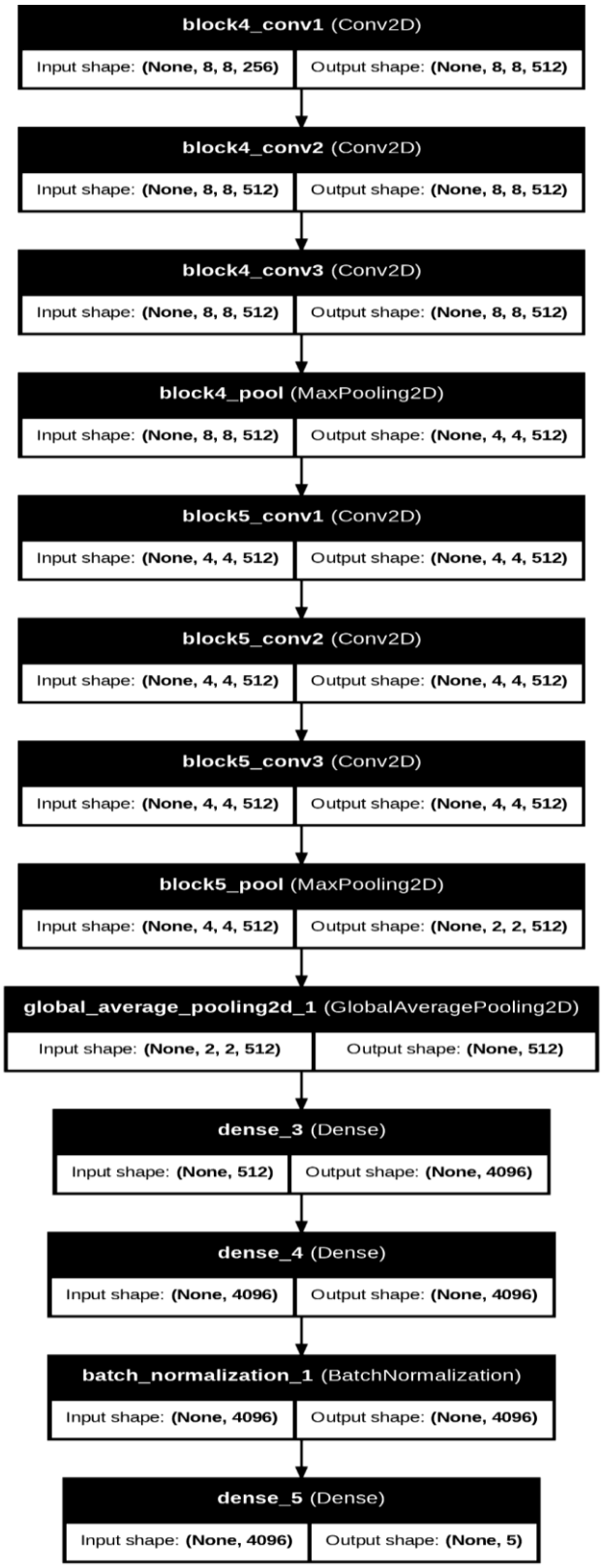
สำหรับ Dense layer แรก มี 4096 units ส่วน Dense layer ที่สอง มี 4096 units และ ใช้ Batch Normalization เพิ่มความเสถียรของโมเดล

5) Output Layer

Output Layer มี 5 unit ตามจำนวนคลาส และใช้ SoftMax activation

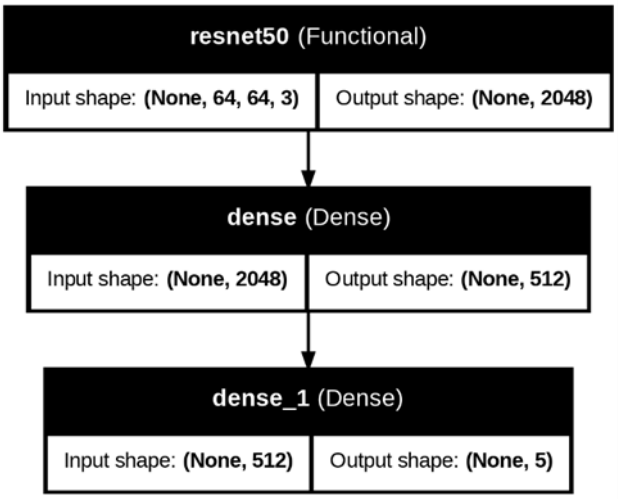


รูปที่ 3.3 แสดงโครงสร้างแบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ VGG



รูปที่ 3.4 แสดงโครงสร้างแบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ VGG (ต่อ)

3.5.5.3 แบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ ResNet5 แบบจำลองนี้มีการทำ Transfer Learning โดยใช้ ResNet50 ซึ่งถูกเทรนล่วงหน้าจากชุดข้อมูลใหญ่ (เช่น ImageNet) เพื่อดึงคุณลักษณะพื้นฐานของภาพ ก่อนที่จะนำเวกเตอร์คุณลักษณะไปประมวลผลต่อด้วย Dense Layers ที่กำหนดขึ้นใหม่ เพื่อปรับแต่งการจำแนกประเภทให้ตรงกับชุดข้อมูลที่สนใจซึ่งในกรณีนี้มีทั้งหมด 5 ประเภท



รูปที่ 3.5 แสดงโครงสร้างแบบจำลองโดยเทคนิควิธีโครงข่ายประสาทเทียมแบบ ResNet50

โครงสร้างแบบจำลองแสดงในภาพที่ 3.4 คือแบบจำลอง Deep Learning แบบ Transfer Learning โดยใช้แบบจำลองพื้นฐานคือ ResNet50 และต่อด้วย Dense Layers ซึ่งเหมาะสำหรับปัญหาการจำแนกประเภท (Classification) รายละเอียดของแต่ละส่วนดังนี้

1) ResNet50 (Functional) ประกอบด้วย Input shape โมเดลรับอินพุตเป็นภาพขนาด 64 x 64 พิกเซล มี 3 channel สี (RGB) และ None หมายถึงจำนวนตัวอย่างข้อมูลที่สามารถเปลี่ยนแปลงได้ตาม Batch Size ที่ใช้ในการเทรน และ Output shape หลังจากผ่านชั้น Convolutional และ Pooling ต่างๆของ ResNet50 แล้วแบบจำลองจะให้เวกเตอร์คุณลักษณะ (Feature Vector) ขนาด 2048 มิติ สำหรับแต่ละภาพอินพุต การตั้งค่าเป็น Global Average Pooling เพื่อสรุปคุณลักษณะสำคัญในแต่ละ channel เป็น 1 ค่า (เวกเตอร์ยาว 2048)

2) Dense Layer แรก ประกอบด้วย Input shape เป็นข้อมูลที่ส่งมาจากโมเดล ResNet50 มีความยาว 2048 มิติ และ Output shape เป็นชั้น Fully Connected (Dense) ชั้นแรกมีหน่วยประมวลผล (Neuron) จำนวน 512 ตัว ชั้นนี้จะเรียนรู้การแปลงเวกเตอร์ขนาดใหญ่ (2048) ให้มีขนาดลดลง (512) เพื่อคัดเลือกคุณลักษณะที่สำคัญในการจำแนกประเภทมากขึ้น โดยใช้ Activation Function เช่น ReLU ในการสร้างความไม่เป็นเชิงเส้น (non-linearity)

3) ส่วนที่สาม Dense Layer ที่สอง (Output Layer) ประกอบด้วย Input shape รับข้อมูลจากชั้น Dense ก่อนหน้าซึ่งมีขนาด 512 มิติ และ Output shape ชั้นนี้มี Neuron จำนวน 5 ตัว แสดงถึงจำนวนของคลาสในการจำแนกประเภท โดยปกติจะใช้ SoftMax เป็น Activation Function เพื่อให้ได้ผลลัพธ์เป็นความน่าจะเป็นของแต่ละคลาส โดยผลรวมจะเป็น 1 ซึ่งผลลัพธ์สุดท้ายจะบ่งบอกว่าข้อมูลที่ป้อนเข้าไปนั้นจัดอยู่ในคลาสใด จากคลาสที่กำหนดจำนวน 5 คลาส

3.6 การประเมินประสิทธิภาพแบบจำลอง (Performance evaluation)

ค่าการประเมินประสิทธิภาพของแบบจำลองที่ถูกสร้างขึ้นเพื่อจำแนกประเภทการล้ม แบ่งออกเป็นดังนี้

3.6.1 ค่าความถูกต้อง (Accuracy) เป็นการวัดว่าแบบจำลองสามารถจำแนกประเภทการล้มทั้งหมดได้ถูกต้องมากน้อยเพียงใด โดยคำนวณจากข้อมูลทดสอบ (Test set) และรายงานผลลัพธ์เป็นเปอร์เซ็นต์ความถูกต้องโดยรวมของแบบจำลอง

3.6.2 ค่าความแม่นยำ (Precision) แสดงถึงความแม่นยำของการทำนายในแต่ละประเภทการล้ม เช่น ทำนายว่ามีการล้มด้านข้างแล้ว ผลลัพธ์เป็นการล้มด้านข้างจริง ไม่ใช่การล้มประเภทอื่นๆ

3.6.3 ค่าเรียกคืน (Recall) วัดความสามารถของแบบจำลองในการค้นหาการล้มแต่ละประเภททั้งหมด เช่น แบบจำลองสามารถตรวจจับการล้มด้านหน้า ด้านหลัง หรือการสะดุดล้มได้ครบถ้วน เพื่อตรวจสอบว่าทุกเหตุการณ์ในการล้มถูกตรวจจับได้

3.6.4 ค่า F1-score เป็นค่าเฉลี่ยเชิงสมดุลระหว่าง Precision และ Recall ที่ช่วยให้เห็นภาพรวมของความสามารถของแบบจำลองในการจำแนกข้อมูล โดยเฉพาะในกรณีที่ข้อมูลมีความไม่สมดุลระหว่างคลาส ซึ่งค่า F1-score ที่สูงบ่งชี้ว่าแบบจำลองสามารถทำนายได้อย่างแม่นยำและครอบคลุมทั้งในด้านการระบุคลาสที่ถูกต้องและไม่พลาดคลาสสำคัญ

3.6.5 Confusion Matrix นำเสนอในรูปแบบ Heatmap เพื่อวิเคราะห์ข้อผิดพลาดระหว่างประเภทการล้มที่แบบจำลองสับสน เช่น การแยกแยะผิดระหว่าง "การล้มด้านหลัง" กับ "การล้มด้านข้าง"

3.6.6 ROC Curve และ AUC ใช้วิเคราะห์ความสามารถในการจำแนกในแต่ละคลาส วัดค่า AUC เพื่อดูว่าความสามารถแยกแยะดีแค่ไหนในแต่ละประเภท

3.6.7 Learning Curves (Accuracy & Loss) แสดงกราฟการเรียนรู้ระหว่าง Epoch เพื่อประเมินว่าแบบจำลองเกิด Overfitting หรือ Underfitting หรือไม่

การวิเคราะห์ทั้งหมดนี้ช่วยให้สามารถประเมินและพัฒนาแบบจำลองสำหรับการจำแนกประเภทการล้มได้อย่างแม่นยำ มีประสิทธิภาพ และสามารถนำไปใช้จริงในการติดตามความปลอดภัยของผู้ใช้งานได้ในอนาคต