TriDA : Trial-Channel Double Attention Feature Integration for Fall Detection Using Ensemble Machine Learning Models

by Kalah Menang Tetap Nganggur

Our Team

Institut Teknologi Telkom Purwokerto

Fauzan Ihza Fajar

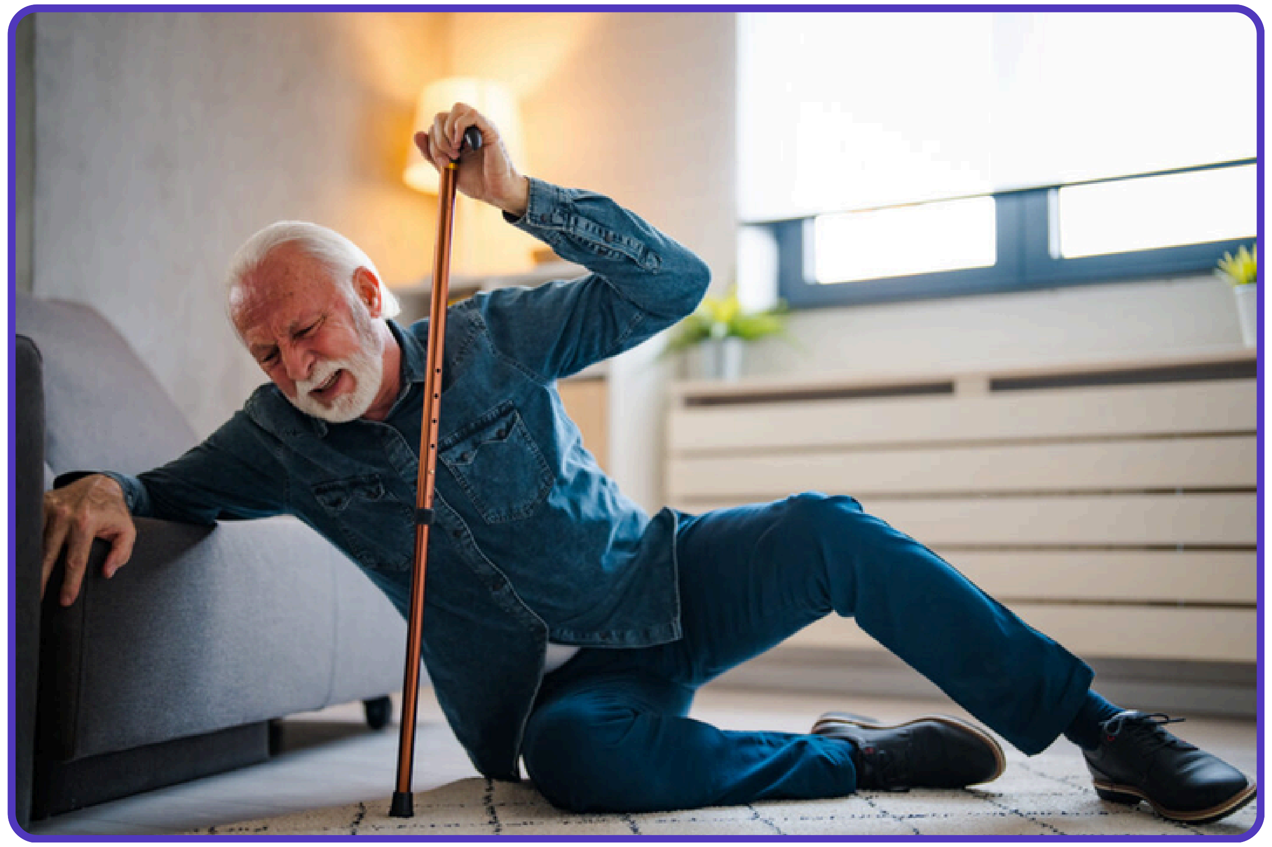
Krisna Bayu Dharma Putra

Akmal Muzakki Bakir

Outline

Introduction

1 Backgrounds



High rates of **injury** and **mortality** caused by **undetected falls**, especially among the **elderly**.

Difficulty **distinguishing falls** from **normal activities**, leading to **false positives** or **negatives** in detection systems.

Objectives & Benefits

Objectives

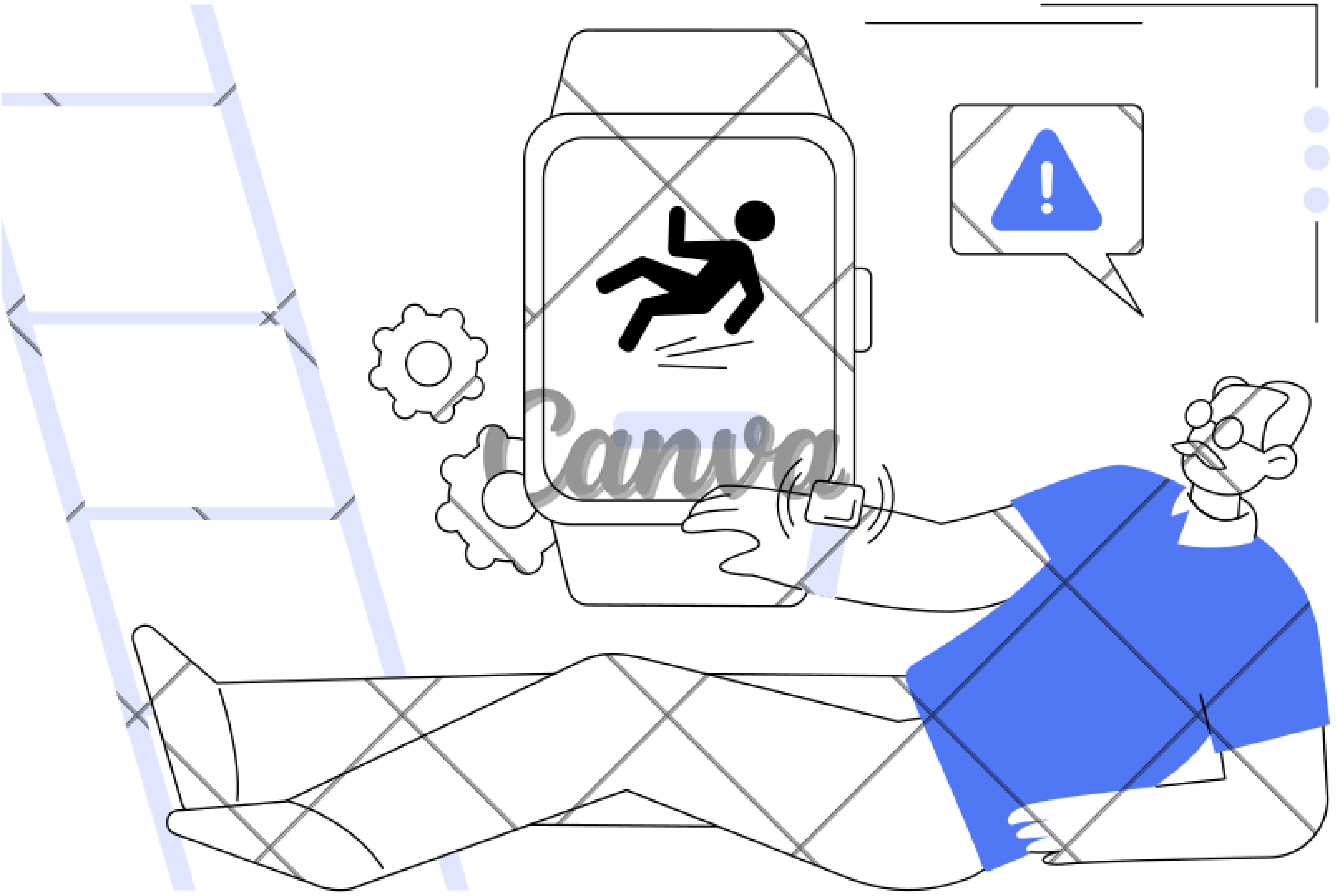
To **develop** an **accurate** and **reliable system** for **detecting falls in real time** to reduce the risks associated with delayed response.

Benefits

Enhanced Safety

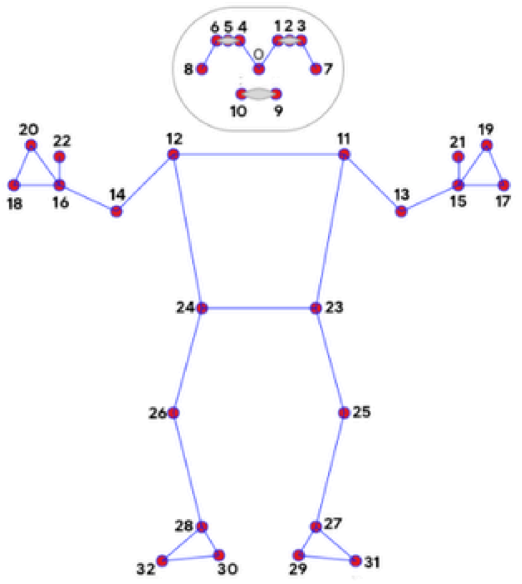
Resource Efficiency

Improved Quality of Life



Our Solution

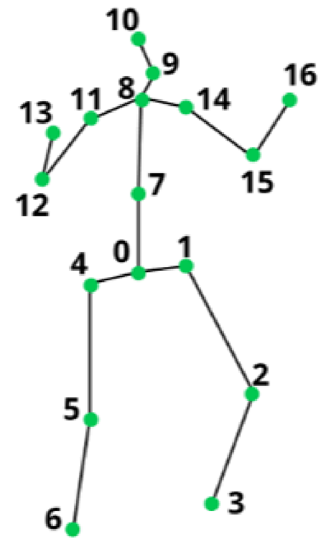
Mediapipe Pose



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

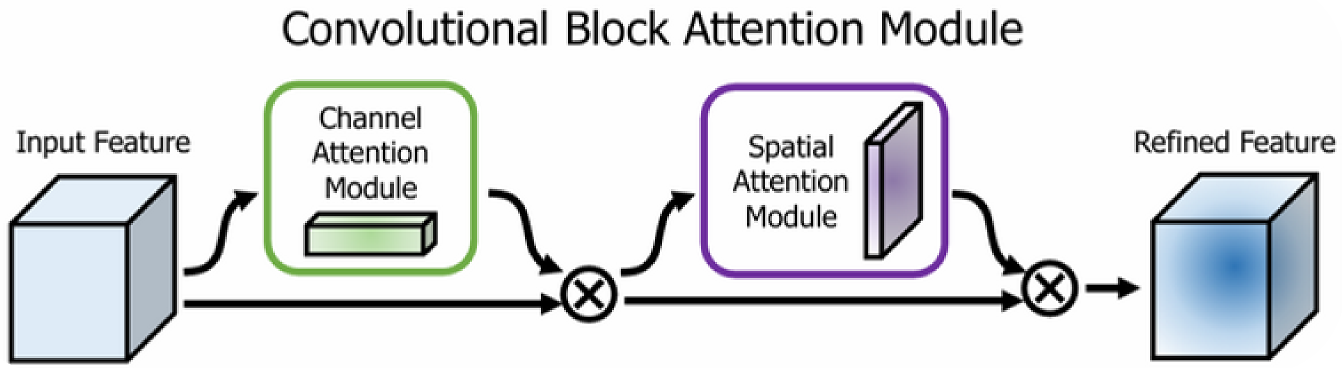
YOLOv8 Pose

0 — Bottom torso
1 — Left hip
2 — Left knee
3 — Left foot
4 — Right hip
5 — Right knee
6 — Right foot
7 — Center torso
8 — Upper torso

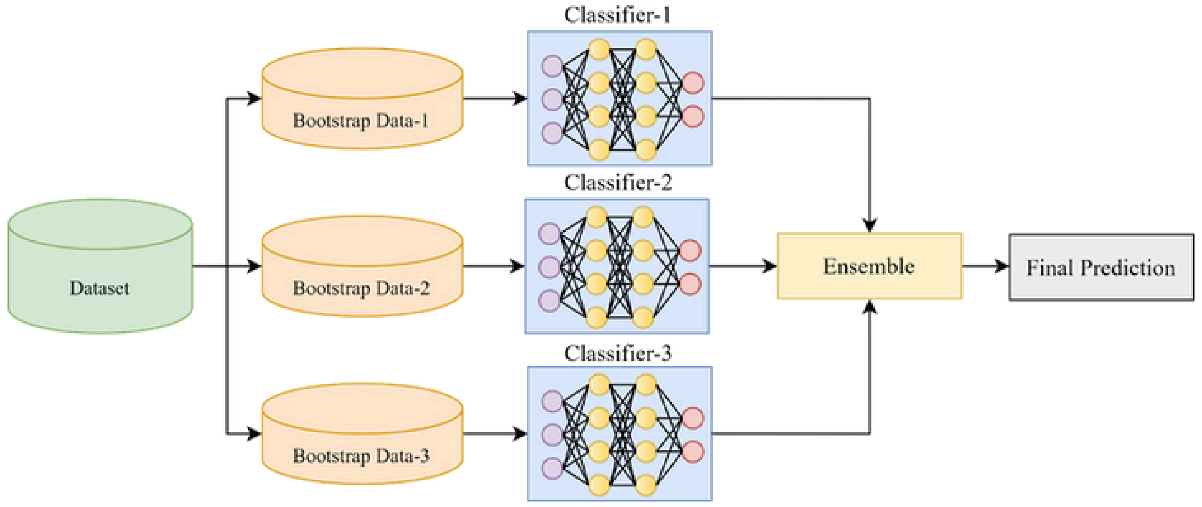


9 — Neck base
10 — Center head
11 — Right shoulder
12 — Right elbow
13 — Right hand
14 — Left shoulder
15 — Left elbow
16 — Left hand

CBAM

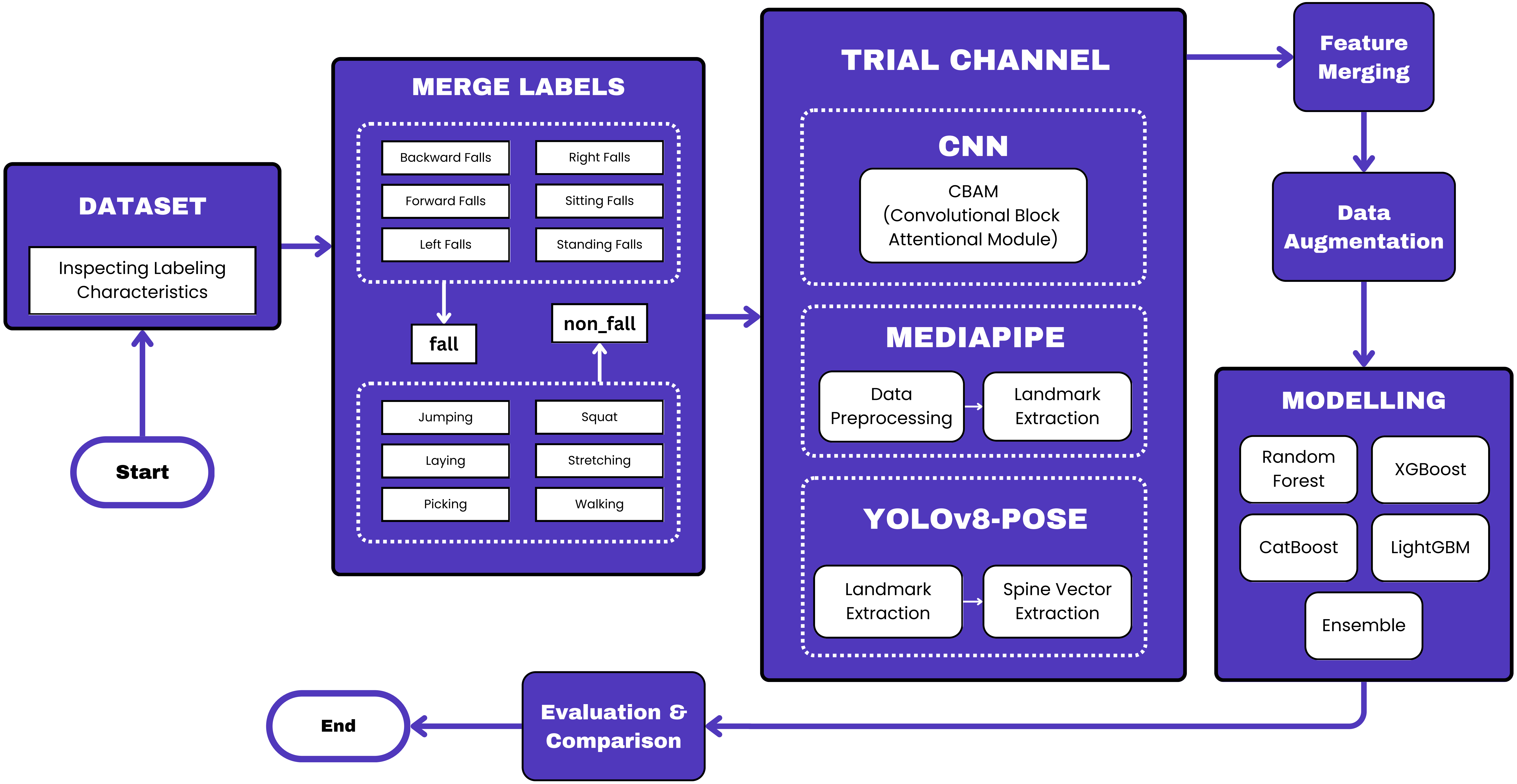


Ensemble Model



Solution Workflow

Workflow


DATASET
Inspecting Labeling Characteristics
Start
MERGE LABELS
Backward Falls
Right Falls
Forward Falls
Sitting Falls
Left Falls
Standing Falls
fall
non_fall
Jumping
Squat
Laying
Stretching
Picking
Walking
TRIAL CHANNEL
CNN
CBAM (Convolutional Block Attentional Module)
MEDIAPIPE
Data Preprocessing
Landmark Extraction
YOLOv8-POSE
Landmark Extraction
Spine Vector Extraction
Feature Merging
Data Augmentation
MODELLING
Random Forest
XGBoost
CatBoost
LightGBM
Ensemble
Evaluation & Comparison
End

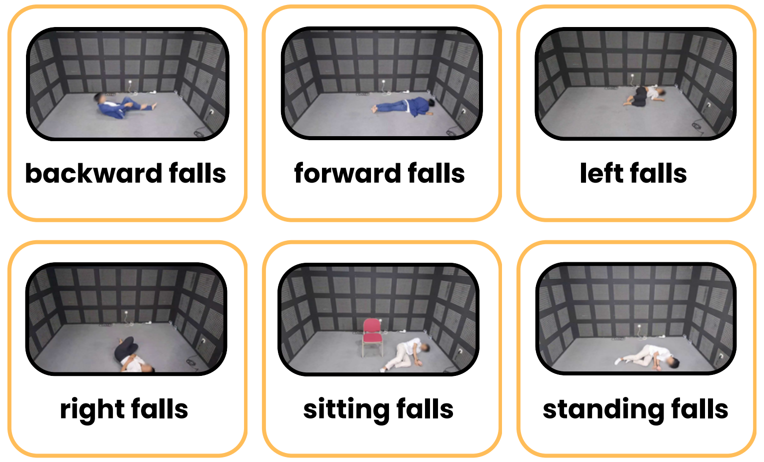
Dataset

Dataset Folder Organization

```
Train
├── subject-1
│   ├── fall
│   │   ├── backward_falls
│   │   ├── forward_falls
│   │   ├── left_falls
│   │   ├── right_falls
│   │   ├── sitting_falls
│   │   └── standing_falls
│   └── non-fall
│       ├── jumping
│       ├── laying
│       ├── picking
│       ├── squat
│       ├── stretching
│       └── walking
├── subject-2
│   ├── fall
│   │   ├── backward_falls
│   │   ├── forward_falls
│   │   ├── left_falls
│   │   ├── right_falls
│   │   ├── sitting_falls
│   │   └── standing_falls
│   └── non-fall
│       ├── jumping
│       ├── laying
│       ├── picking
│       ├── squat
│       ├── stretching
│       └── walking
├── subject-3
│   ├── fall
│   │   ├── backward_falls
│   │   ├── forward_falls
│   │   ├── left_falls
│   │   ├── right_falls
│   │   ├── sitting_falls
│   │   └── standing_falls
│   └── non-fall
│       ├── jumping
│       ├── laying
│       ├── picking
│       ├── squat
│       ├── stretching
│       └── walking
└── subject-4
    ├── fall
    │   ├── backward_falls
    │   ├── forward_falls
    │   ├── left_falls
    │   ├── right_falls
    │   ├── sitting_falls
    │   └── standing_falls
    └── non-fall
        ├── jumping
        ├── laying
        ├── picking
        ├── squat
        ├── stretching
        └── walking
```

Initial Folder Organization

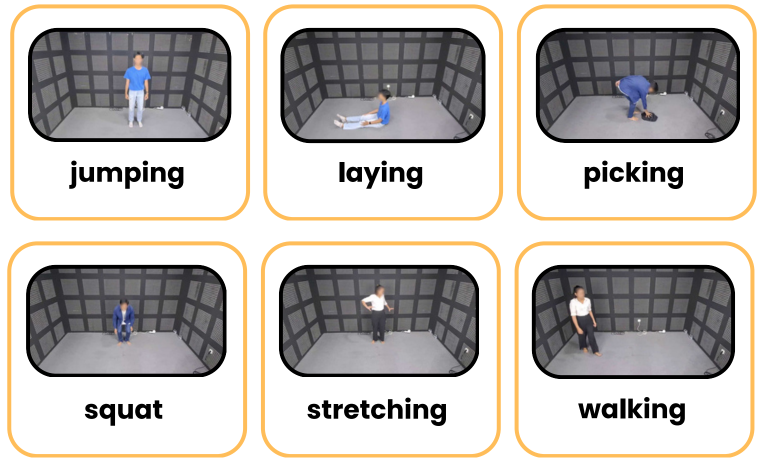
FALL LABELS



Organized into

FALL

NON FALL LABELS



Organized into

NON FALL

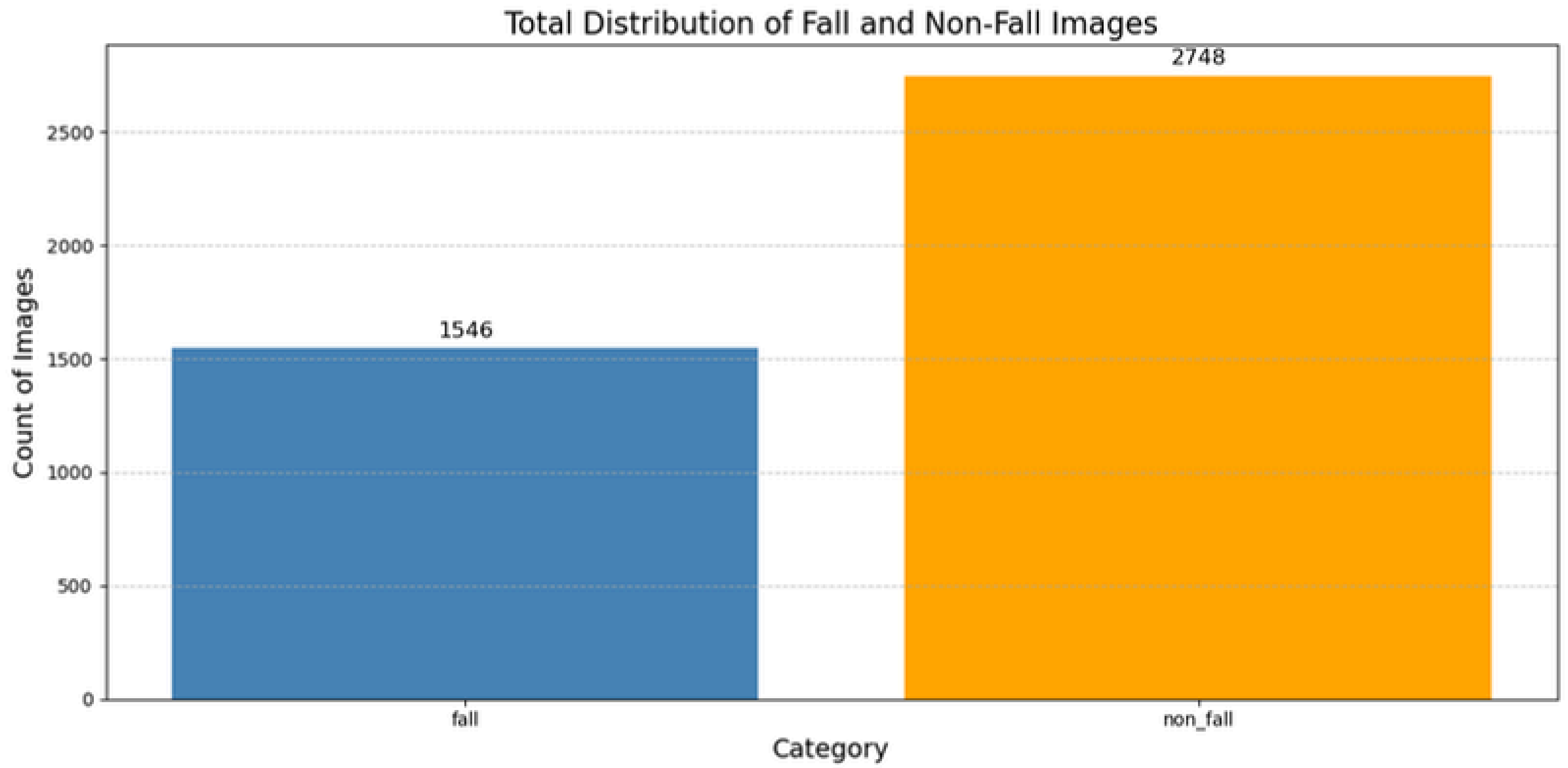
Train

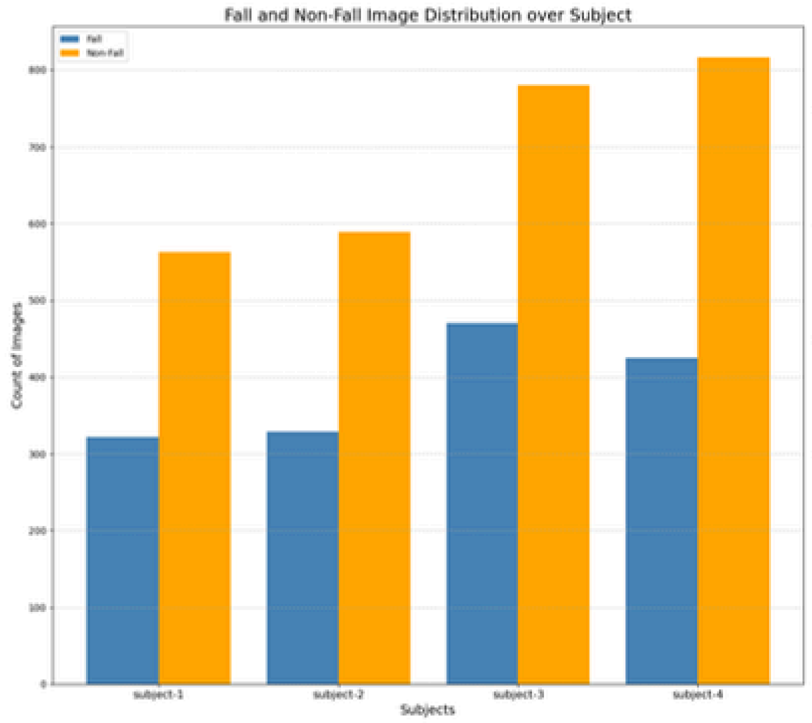
Folder Organization Processing

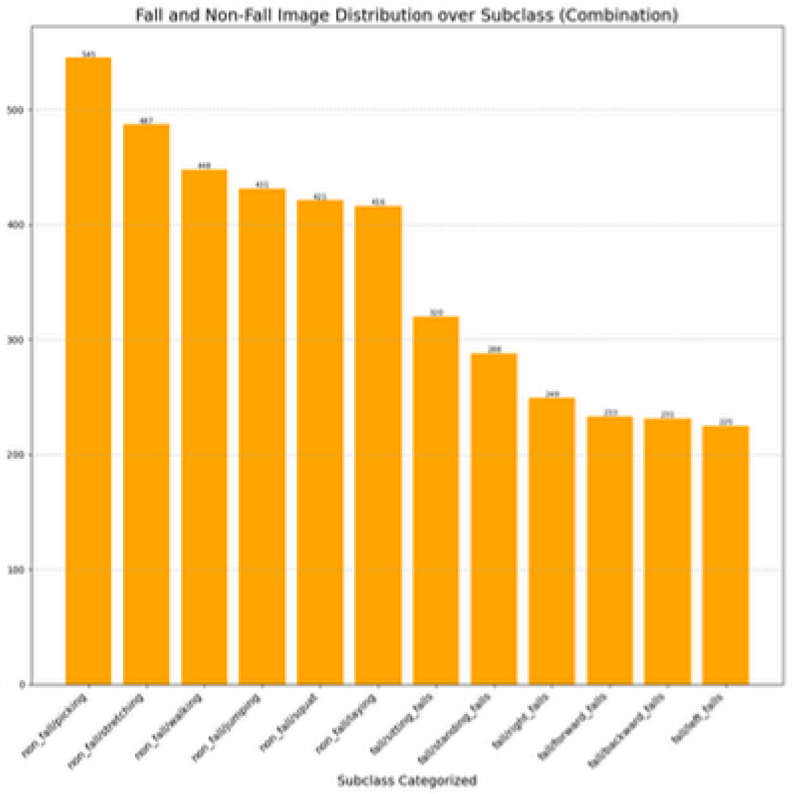
Fall	Non Fall
Backward Falls	Jumping
Forward Falls	Laying
Left Falls	Picking
Right Falls	Squat
Sitting Falls	Stretching
Standing Falls	Walking

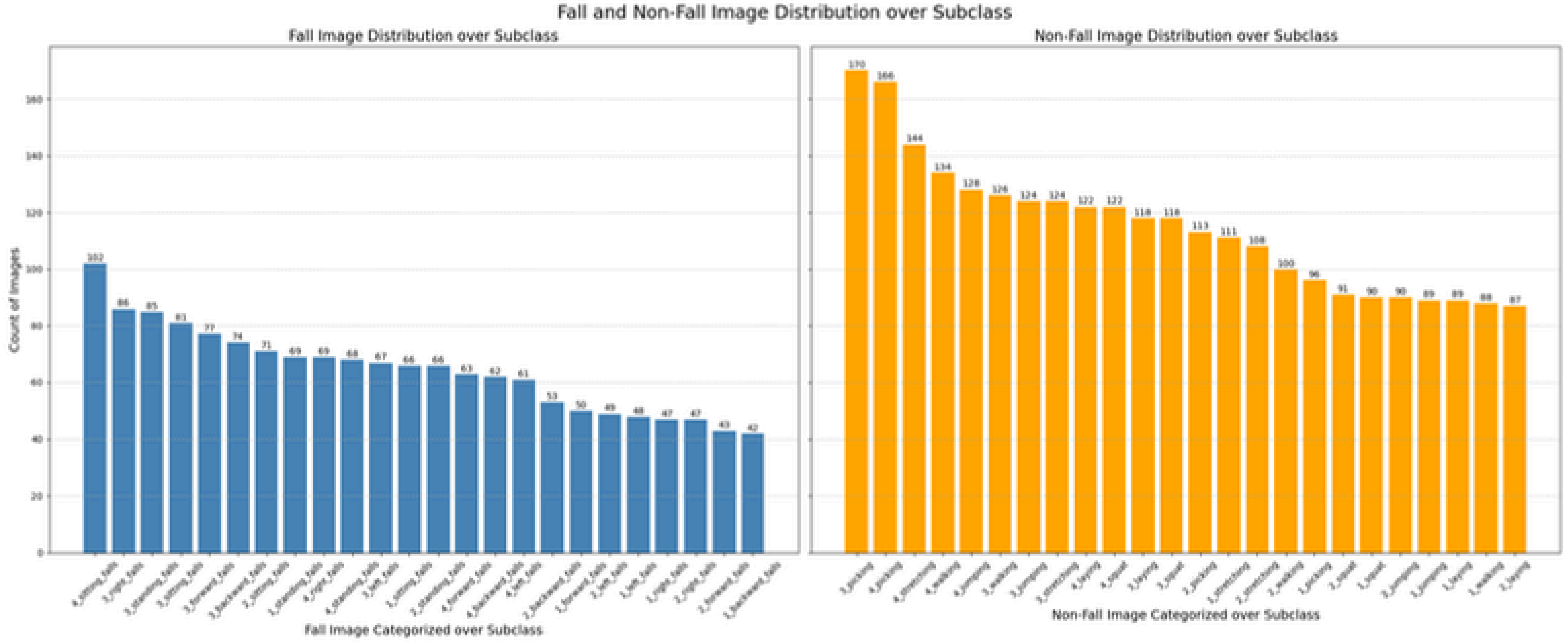
Final Folder Organization

Dataset Label Distribution



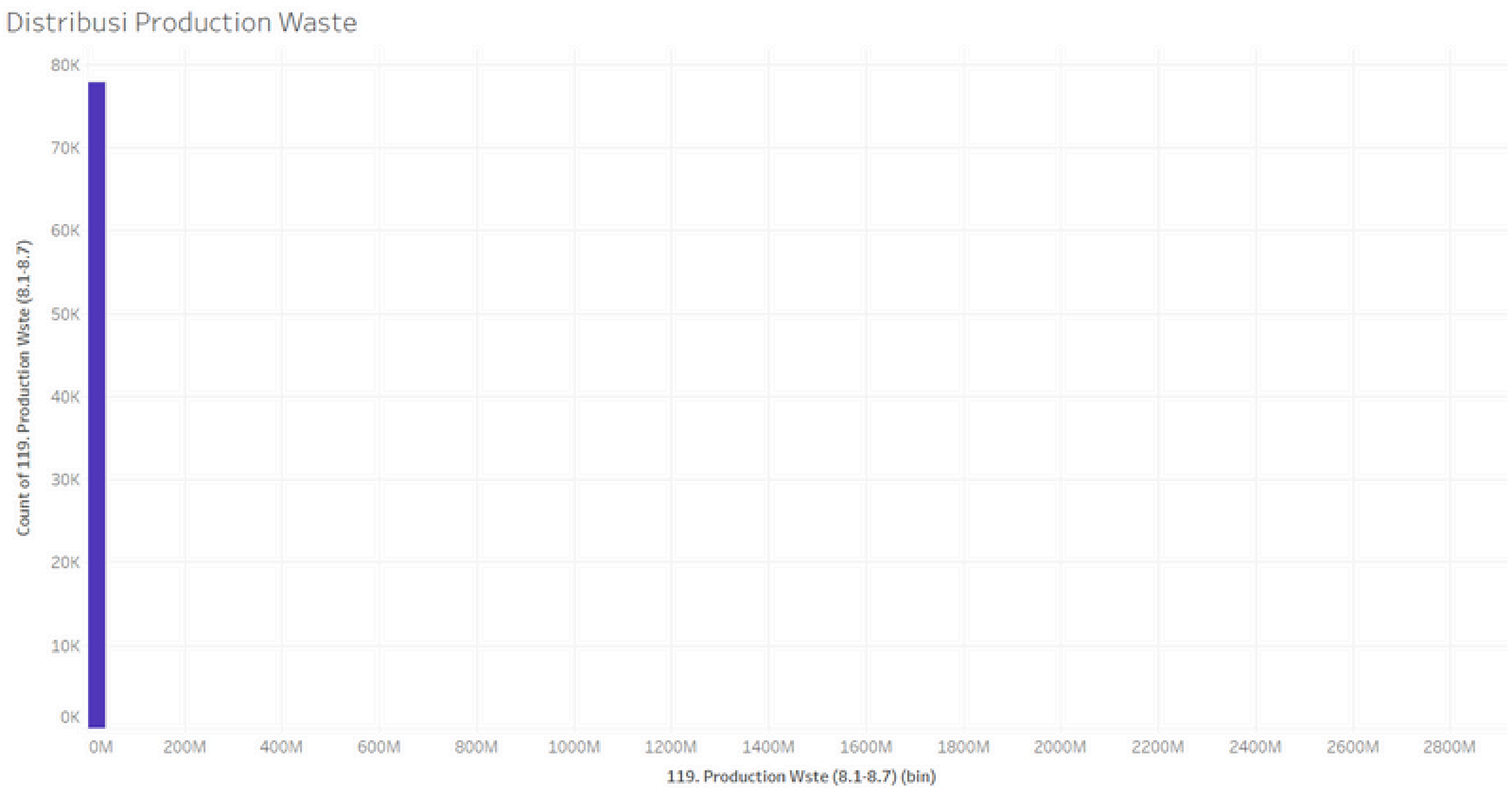






Based on the analysis of subclasses, subjects, or categories, it can be concluded that the data owned has an unbalanced distribution. The non-fall class has a much larger amount of data compared to the fall class.

Dataset Label Distribution

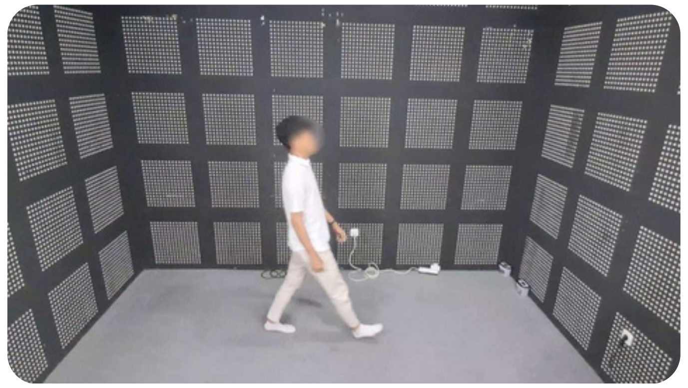


Distribusi kolom

Special Characteristics

FALL

head on left
(backward falls)

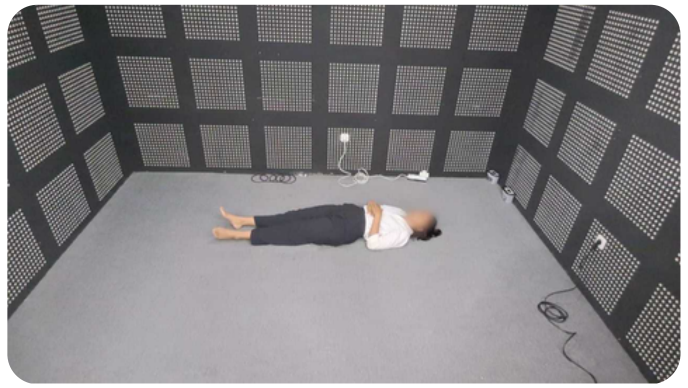


longer strides
(backward falls)

have a chair
(sitting falls)

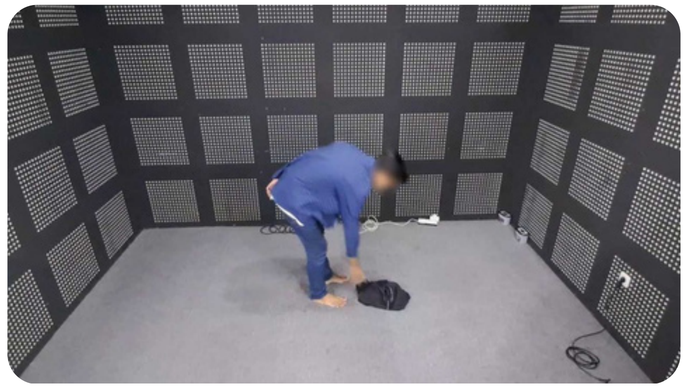
body slightly tilted
(standing falls)

NON FALL



head on right
(laying)

smaller strides
(walking)



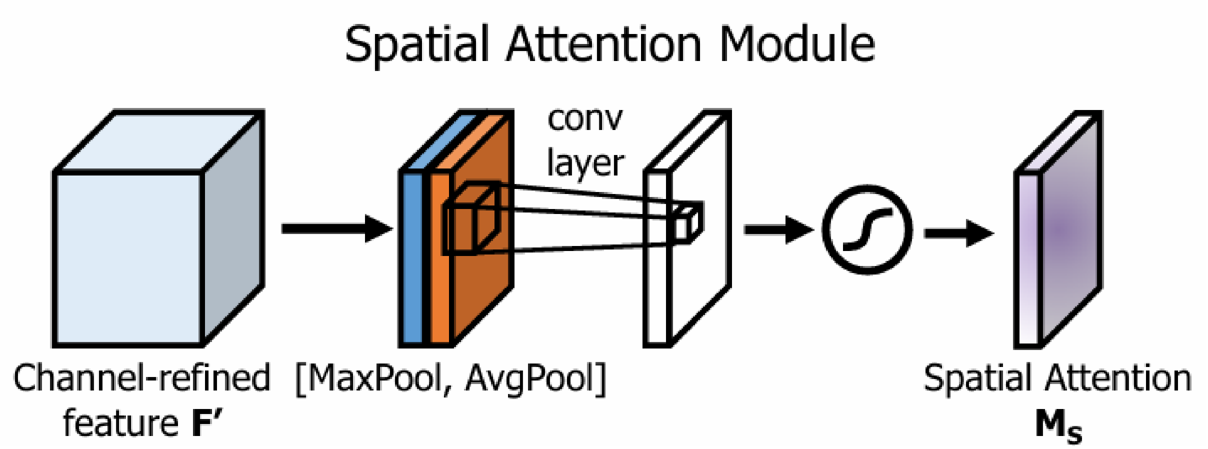
have an object
(picking)

straight body upright
(jumping)

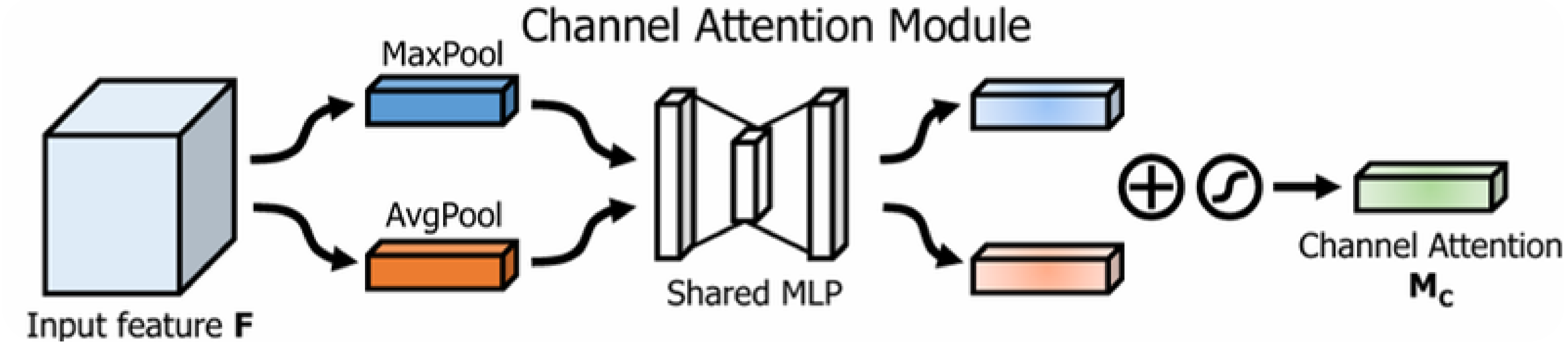
Convolutional Block Attention Module (CBAM)

CBAM Architecture

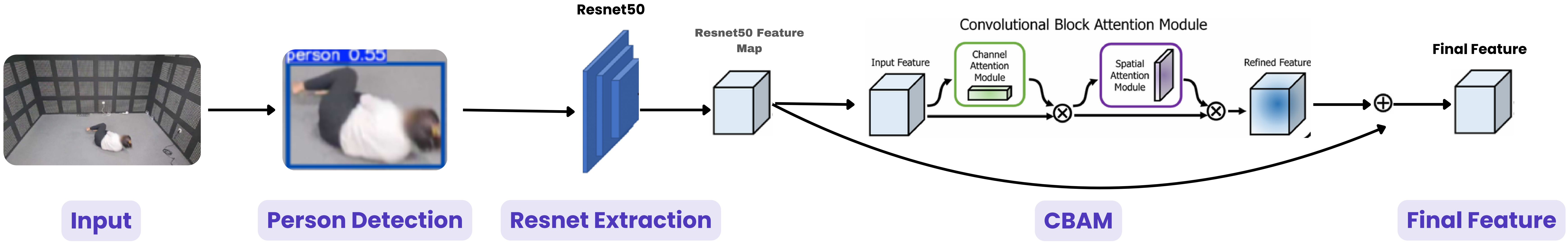
Spatial Attention Module



Channel Attention Module



Proposed CBAM Integration and Extraction



Mediapipe Pose

Pose Landmark Model

METHOD

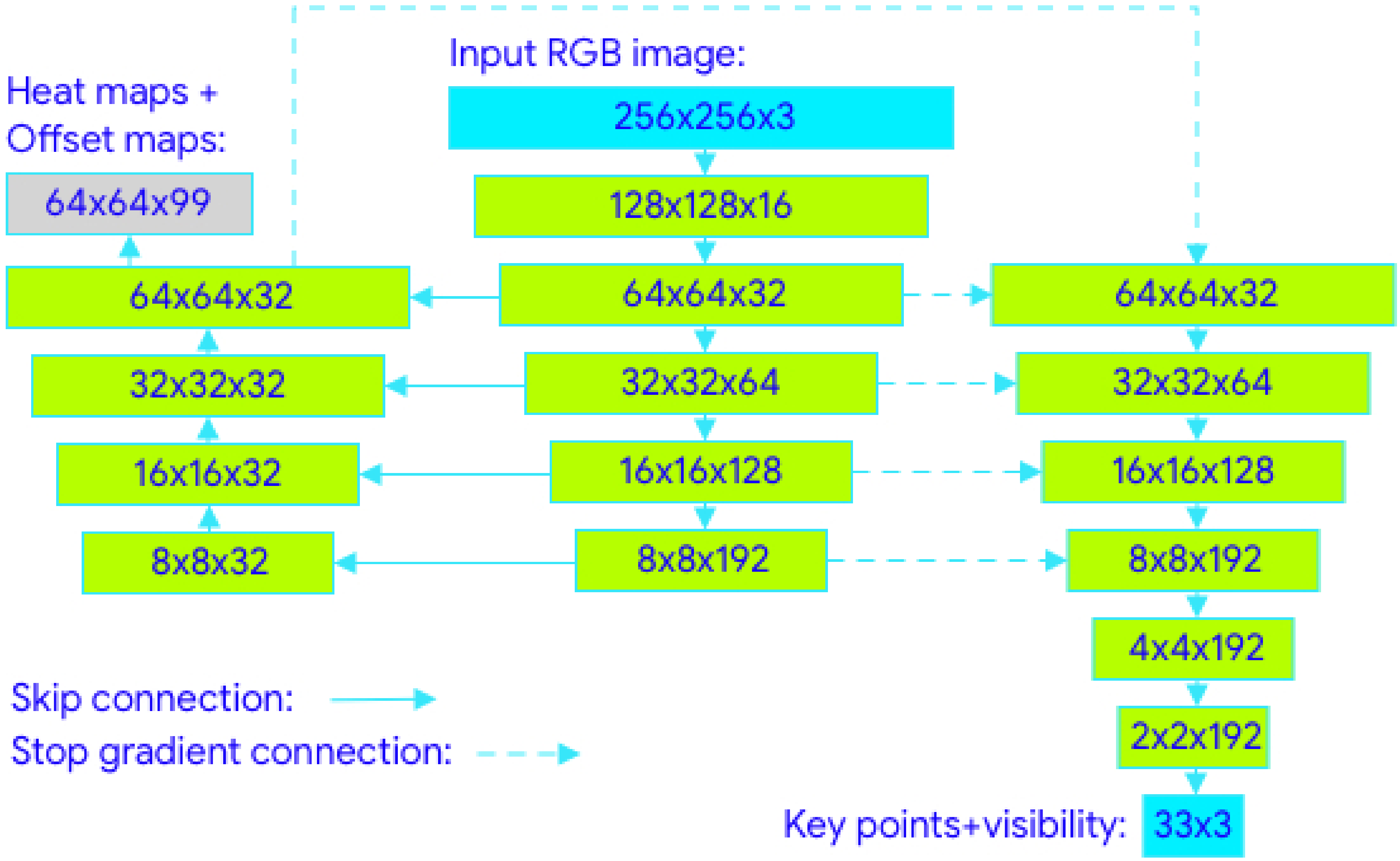
BlazePose GHUM Heavy

~~**BlazePose GHUM Full**~~

~~**BlazePose GHUM Lite**~~

We use a fairly complex method because some images in the dataset do not have their keypoints detected if we use other variations of the method.

TRACKING NETWORK ARCHITECTURE



[Hyperlink for Blazepose GHUM Heavy Architecture]

Pose Estimate Extraction

image_name	label	kp1_x	kp1_y	kp2_x	kp2_y	....	kp33_x`	kp33_y
1_backward_fa	fall	467.14	462.173	472.073	484.113	....	479.442	503.441
1_backward_fa	fall	510.116	437.541	512.181	456.564	....	527.927	478.506
1_backward_fa	fall	536.8	426.957	557.475	444.992	....	538.256	470.033
1_backward_fa	fall	579.856	412.192	549.684	439.825	....	576.623	452.052
1_backward_fa	fall	608.895	403.939	575.393	430.866	....	602.785	449.497
1_backward_fa	fall	639.106	386.166	597.429	415.168	....	626.292	430.456
1_backward_fa	fall	666.737	358.8	615.673	380.476	....	643.726	404.589
1_backward_fa	fall	680.21	333.288	629.971	355.528	....	657.303	368.46
1_backward_fa	fall	684.644	303.929	650.114	323.814	....	674.143	344.232

Person Detection

Pose Extraction

Save to Dataframe

Input

Normalizing Image

Blazepose Mediapipe Initiation

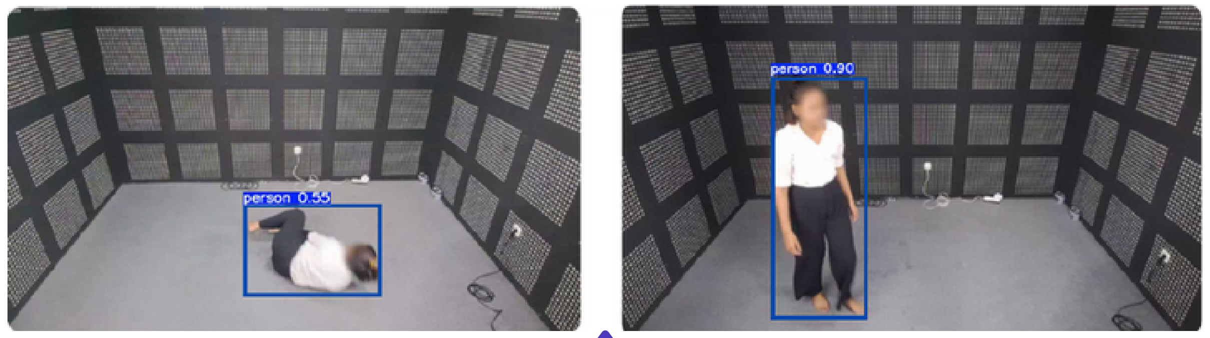
End

Start

Institut Teknologi Telkom Purwokerto

YOLOv8 Pose

Pose Estimate Extraction



image_name	label	kp1_x	kp1_y	kp2_x	kp2_y	....	kp17_x	kp17_y
1_backward_fa	fall	647.191	181.318	626.513	222.616	....	632.303	241.49
1_backward_fa	fall	661.026	186.782	634.583	229.765	....	645.337	245.243
1_backward_fa	fall	673.881	201.735	648.4	239.928	....	669.027	256.627
1_backward_fa	fall	680.539	214.327	662.655	256.787	....	668.829	269.093
1_backward_fa	fall	691.049	231.418	658.456	268.004	....	675.555	288.083
1_backward_fa	fall	700.278	247.246	656.487	283.564	....	681.699	301.573
1_backward_fa	fall	697.315	277.355	651.445	298.888	....	679.074	321.437
1_backward_fa	fall	684.644	303.929	650.114	323.814	....	674.143	344.232
1_backward_fa	fall	680.21	333.288	629.971	355.528	....	657.303	368.46

Start

Input

Resizing Image

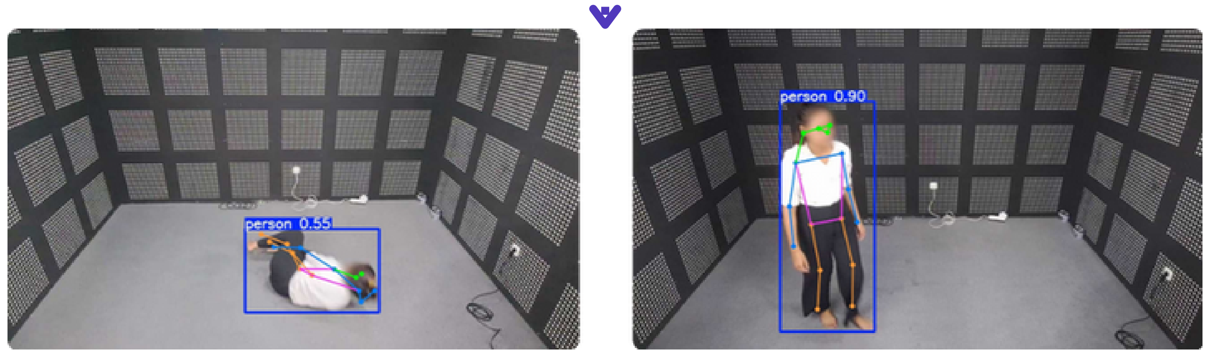
YOLOv8 Initiation

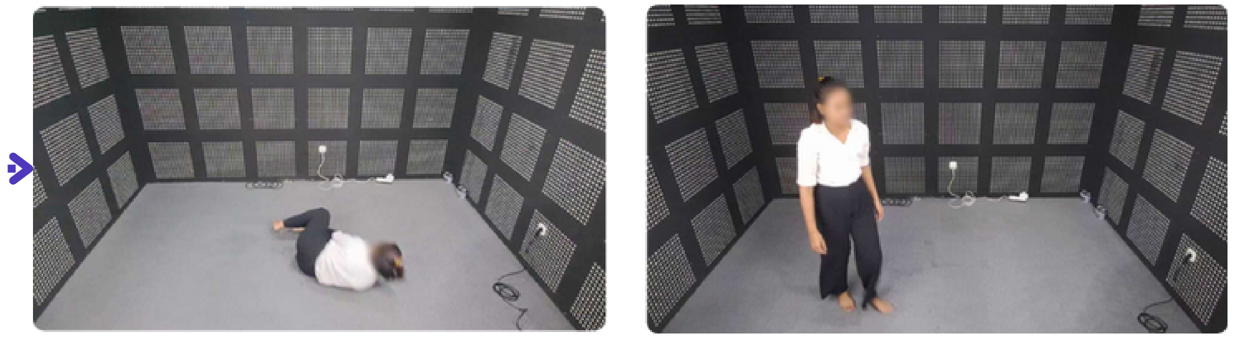
Person Detection

Pose Extraction

Save to Dataframe

End



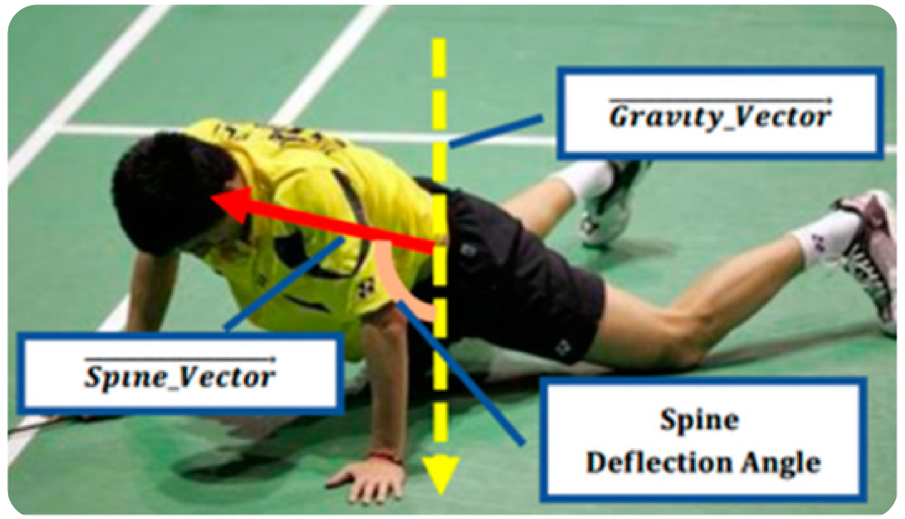


Spine Vector Extraction

Spine Vector Angle

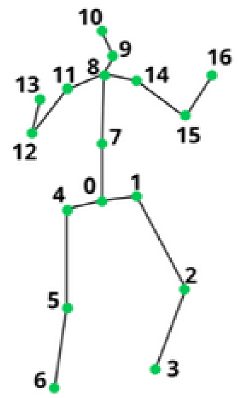
$$\cos(\theta) = \frac{\vec{v}_{spine} \cdot \vec{g}}{||\vec{v}_{spine}|| \, ||\vec{g}||}$$

$$\overrightarrow{Spine\ vector} = \begin{bmatrix} X_1 - X_8 \\ Y_1 - Y_8 \end{bmatrix}$$



3D KEYPOINTS AND THEIR SPECIFICATION

0 — Bottom torso
1 — Left hip
2 — Left knee
3 — Left foot
4 — Right hip
5 — Right knee
6 — Right foot
7 — Center torso
8 — Upper torso

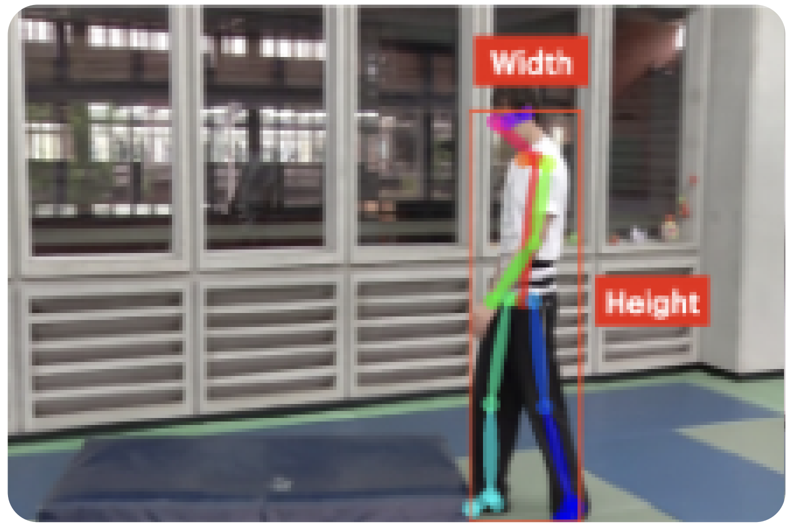


9 — Neck base
10 — Center head
11 — Right shoulder
12 — Right elbow
13 — Right hand
14 — Left shoulder
15 — Left elbow
16 — Left hand

Spine Ratio

$$HW\ Ratio = \frac{Height}{Width}$$

$$SpineRatio = \frac{Spine\ Length}{Waist\ Length}$$



Spine Vector Detection is used calculates the orientation of the spine using keypoints from pose estimation

Feature Integration

Feature Integration

Mediapipe Keypoint Data

image_name	label	kp1_x	kp1_y	kp2_x	kp2_y	....	kp33_x`	kp33_y
1_backward_fa	fall	467.14	462.173	472.073	484.113	....	479.442	503.441
1_backward_fa	fall	510.116	437.541	512.181	456.564	....	527.927	478.506
1_backward_fa	fall	536.8	426.957	557.475	444.992	....	538.256	470.033
1_backward_fa	fall	579.856	412.192	549.684	439.825	....	576.623	452.052
1_backward_fa	fall	608.895	403.939	575.393	430.866	....	602.785	449.497
1_backward_fa	fall	639.106	386.166	597.429	415.168	....	626.292	430.456
1_backward_fa	fall	666.737	358.8	615.673	380.476	....	643.726	404.589
1_backward_fa	fall	680.21	333.288	629.971	355.528	....	657.303	368.46
1_backward_fa	fall	684.644	303.929	650.114	323.814	....	674.143	344.232

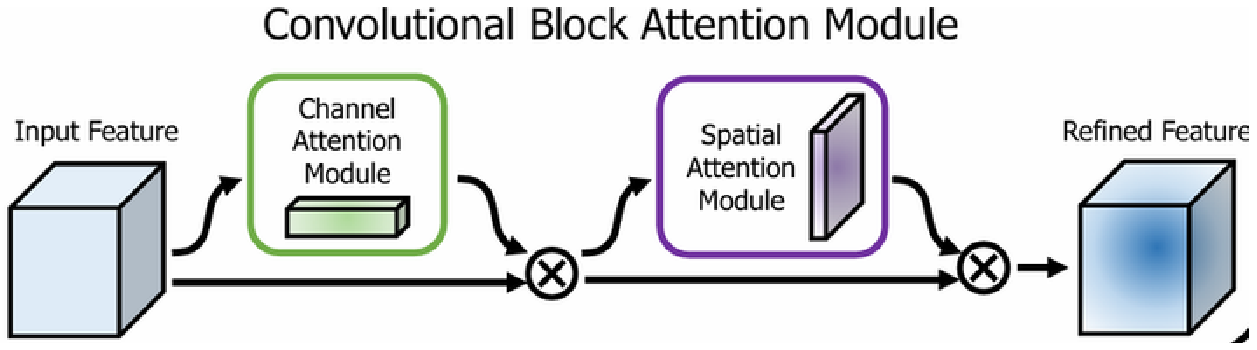
YOLOv8 Keypoint Data

image_name	label	kp1_x	kp1_y	kp2_x	kp2_y	....	kp17_x	kp17_y
1_backward_fa	fall	647.191	181.318	626.513	222.616	....	632.303	241.49
1_backward_fa	fall	661.026	186.782	634.583	229.765	....	645.337	245.243
1_backward_fa	fall	673.881	201.735	648.4	239.928	....	669.027	256.627
1_backward_fa	fall	680.539	214.327	662.655	256.787	....	668.829	269.093
1_backward_fa	fall	691.049	231.418	658.456	268.004	....	675.555	288.083
1_backward_fa	fall	700.278	247.246	656.487	283.564	....	681.699	301.573
1_backward_fa	fall	697.315	277.355	651.445	298.888	....	679.074	321.437
1_backward_fa	fall	684.644	303.929	650.114	323.814	....	674.143	344.232
1_backward_fa	fall	680.21	333.288	629.971	355.528	....	657.303	368.46

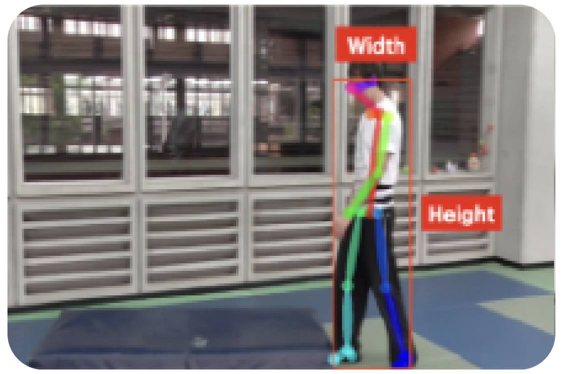
Integrate

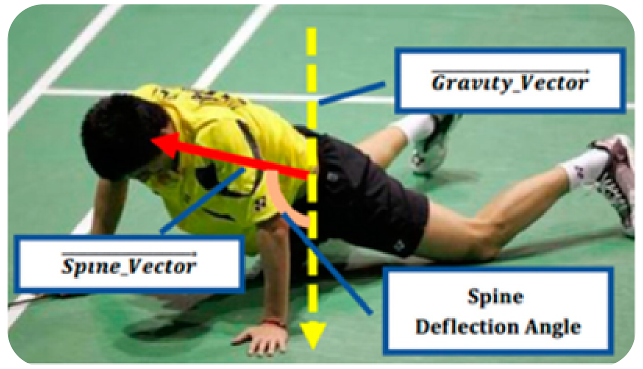
Final Dataset

CBAM Feature Extraction



Spine Vector Extraction





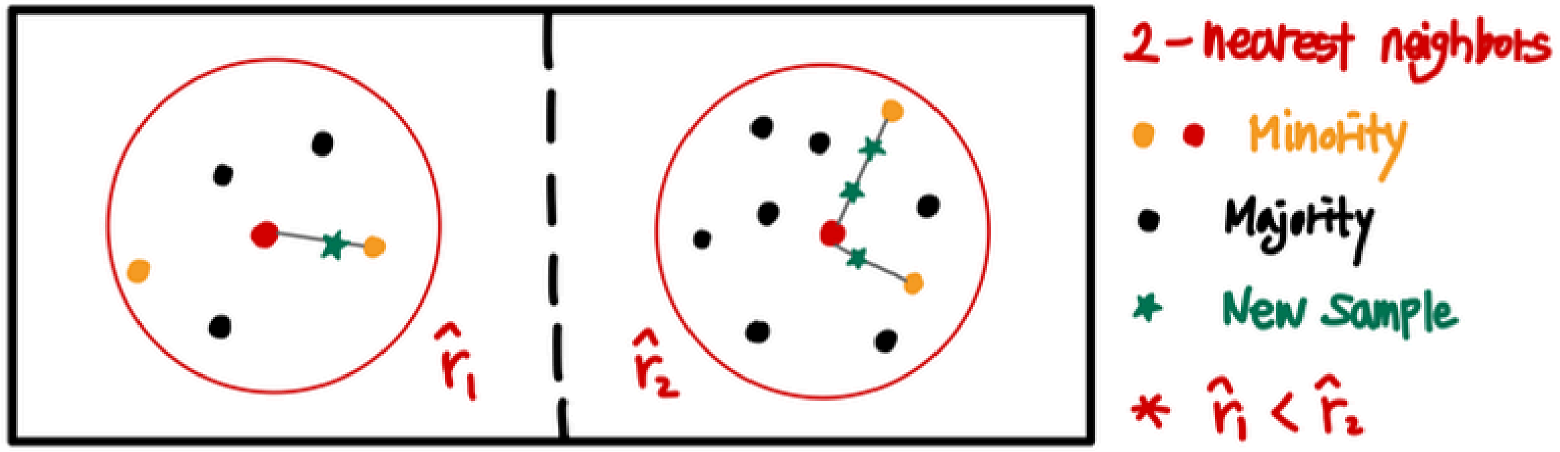
Data Augmentation

Adasyn Augmentation

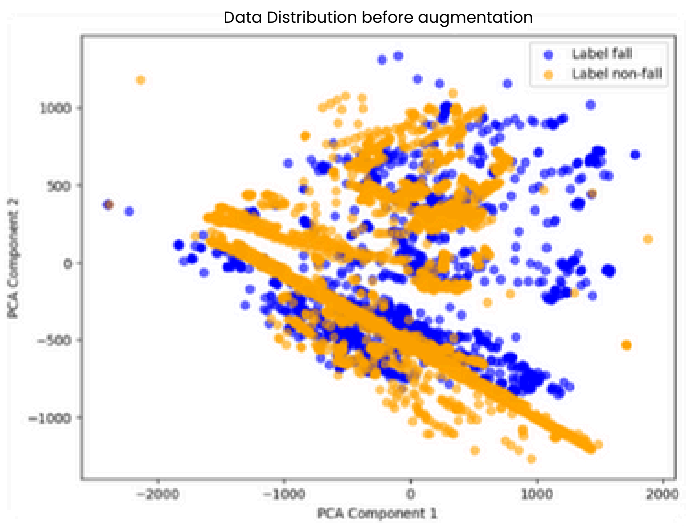
Final Dataset

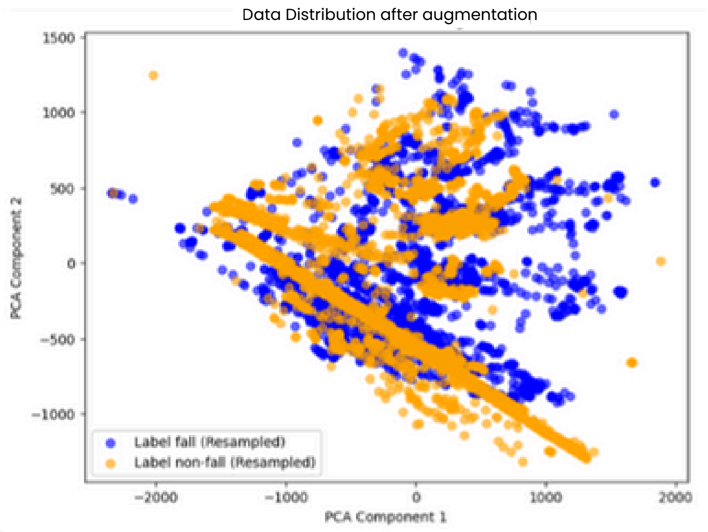
image_nai	label	kp1_x	kp1_y	kp2_x	kp2_y	....	kp17_x`	kp17_y
1_backwa	fall	647.191	181.318	626.513	222.616	....	632.303	241.49
1_backwa	fall	661.026	186.782	634.583	229.765	....	645.337	245.243
1_backwa	fall	673.881	201.735	648.4	239.928	....	669.027	256.627
1_backwa	fall	680.539	214.327	662.655	256.787	....	668.829	269.093
1_backwa	fall	691.049	231.418	658.456	268.004	....	675.555	288.083
1_backwa	fall	700.278	247.246	656.487	283.564	....	681.699	301.573
1_backwa	fall	697.315	277.355	651.445	298.888	....	679.074	321.437

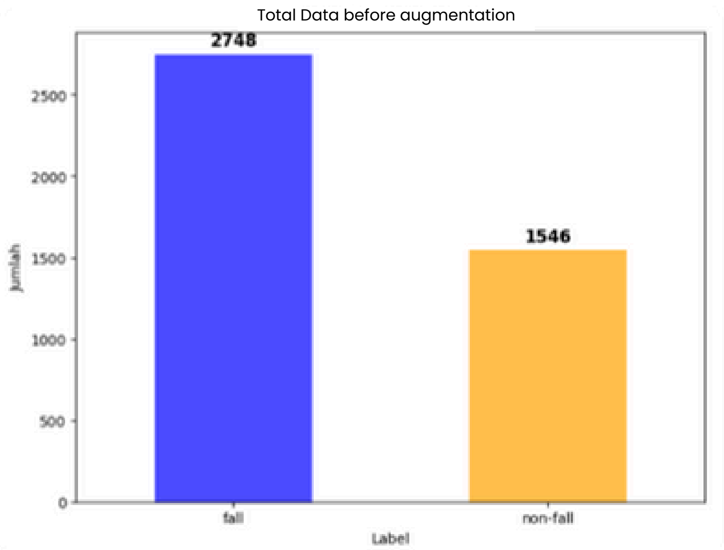
Adasyn Synthetis

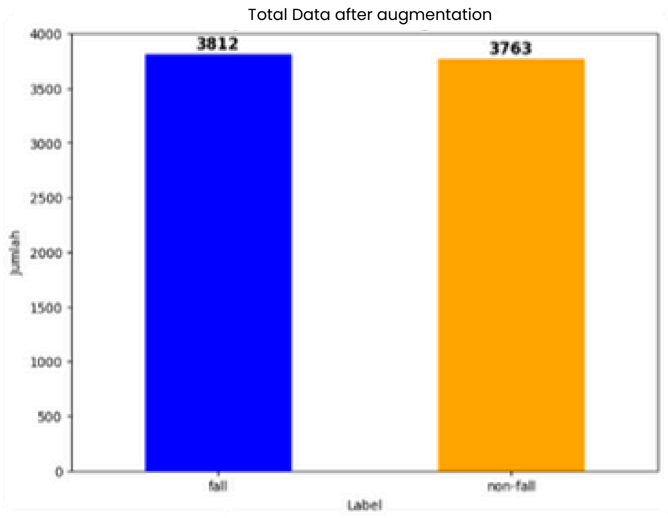


Final Comparison









Classification Modeling

Modeling Approach

image_na	label	kp1_x	kp1_y	kp2_x	kp2_y	....	kp17_x`	kp17_y
1_backwa	fall	647.191	181.318	626.513	222.616	....	632.303	241.49
1_backwa	fall	661.026	186.782	634.583	229.765	....	645.337	245.243
1_backwa	fall	673.881	201.735	648.4	239.928	....	669.027	256.627
1_backwa	fall	680.539	214.327	662.655	256.787	....	668.829	269.093
1_backwa	fall	691.049	231.418	658.456	268.004	....	675.555	288.083
1_backwa	fall	700.278	247.246	656.487	283.564	....	681.699	301.573
1_backwa	fall	697.315	277.355	651.445	298.888	....	679.074	321.437

Model Pipeline

XGBoost | LightGBM | CatBoost | Random Forest

Cross Validation

Iteration 01	Test	Train	Train	Train	Train
Iteration 02	Train	Test	Train	Train	Train
Iteration 03	Train	Train	Test	Train	Train
Iteration 04	Train	Train	Train	Test	Train
Iteration 05	Train	Train	Train	Train	Test

Ensemble Model

CatBoost → Fall

Random Forest → Non-Fall

XGBoost → Fall

LightGBM → Fall

Majority Vote (Fall)

Splitting Data

Train 75%

Val 15%

Test 15%

Single Model

Best Model Metrics Performance

Evaluation and Comparison

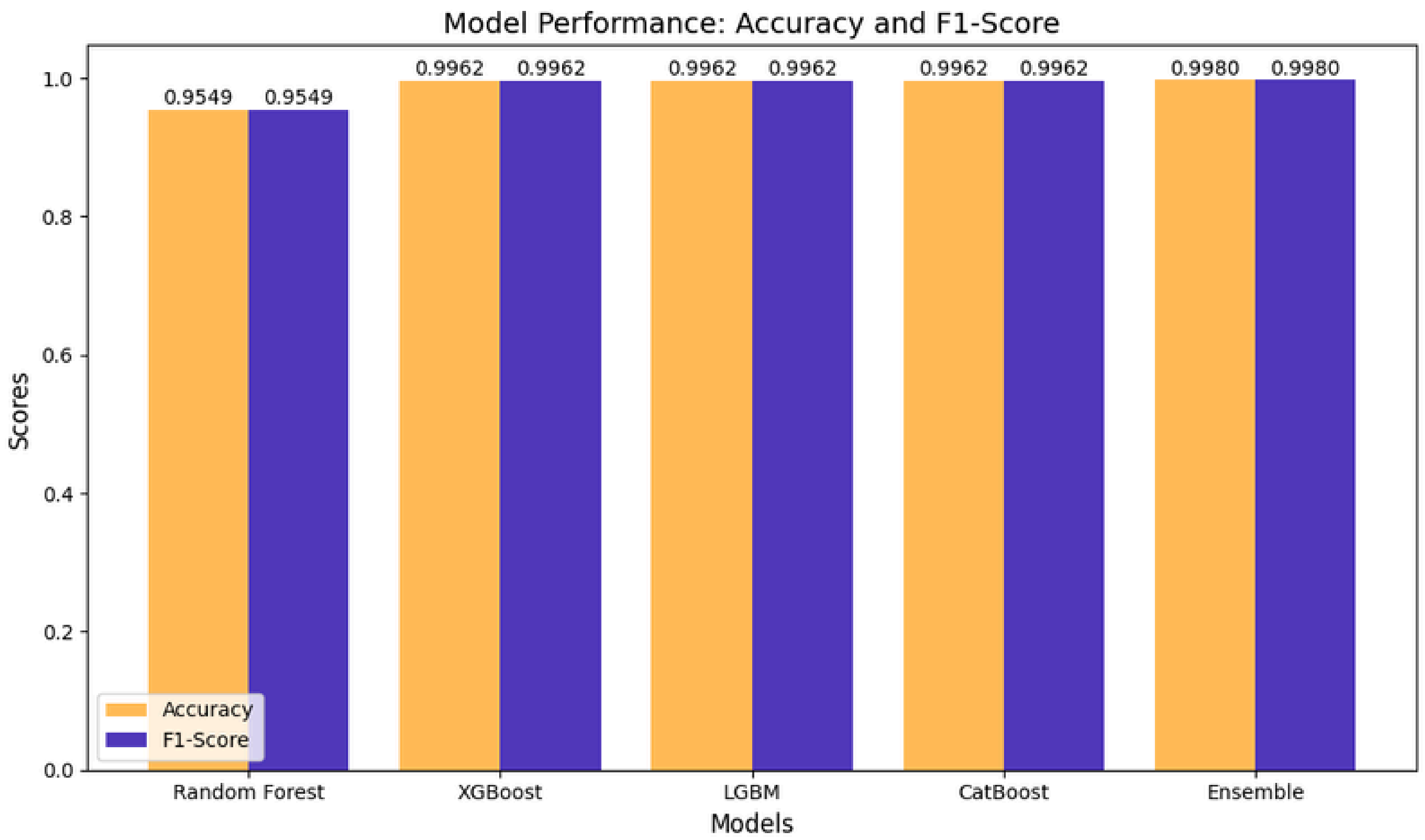
Model Performance

Model Evaluation on Validation Data

Method	F1 Score
YOLOv8 Pose Extraction	0.9232
Mediapipe Pose Extraction	0,9350
Mediapipe + YOLOv8 Pose Extraction	0,9554
Proposed Method (Trial Channel Dual Attention)	**0,9962**

The proposed method achieved the highest f1 score across all the method tested. Our proposal gain an incredible 0,9962 f1 score and outscoring other method by more than 4%.

Trial Channel Dual Attention Performance



Ensemble learning achieve both f1 and accuracy value of 0,9980, outperforming 4 other model.

Leaderboard Score

Institut Teknologi Telkom Purwokerto

	Public Leaderboard	Private Leaderboard
YOLOv8 Pose Estimation	0.93227091	0.92879256
Mediapipe Pose Estimation	0.94223107	0.94427244
Proposed Trial Channel Dual Attention	0.99690402	0.99380804

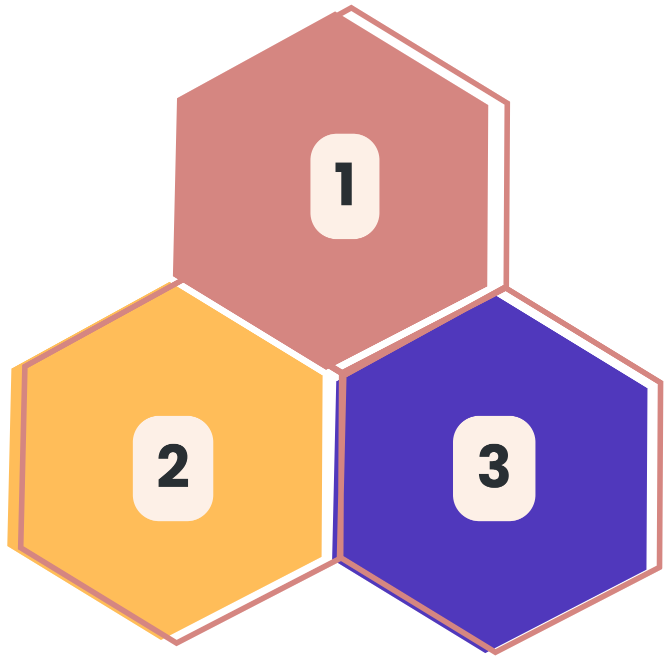
#	Team	Members	Score
1	stx		1.00000000
2	Pendobrak 4.0		0.99845201
3	**Menang Kalah Tetap Nganggur**		0.99690402

F1 Metrics

Conclusion

Conclusion

The proposed **TriDA** framework effectively integrates features using **Trial-Channel Double Attention**



The **TriDA** framework achieved an outstanding accuracy of **99.62%,** outperforming other variants significantly.

The Ensemble model achieves the highest accuracy and F1-score **(99.80%)**, demonstrating superior performance compared to other models.

THANKYOU

by Kalah Menang Tetap Nganggur