Skeleton Keypoint Detection using YOLO and Gait Recognition via Modified GaitGraph

Aditya Malik
amalik28@buffalo.edu

Abstract

Gait recognition is a biometric identification technique that focuses on analyzing and recognizing an individual's unique walking pattern or gait. Gait has several advantages as a biometric modality as it is Non-intrusive, Persistent, Difficult to spoof. Skeleton keypoint detection and gait recognition are crucial tasks in computer vision and biometric identification systems. This project proposes a novel approach that combines the YOLO (You Only Look Once) algorithm for skeleton keypoint detection and a modified gaitGraph2(Teepe et al., 2021) technique for gait recognition. By incorporating gait recognition into biometric systems, it provides an additional layer of identification and authentication, complementing other biometric modalities. The combination of gait recognition with other techniques, such as skeleton keypoint detection using YOLO, enhances the overall accuracy and reliability of biometric identification systems.

1 Introduction

Human pose estimation refers to the task of estimating the spatial locations of key body joints or keypoints in an image or video, enabling the understanding of human posture and movement. It plays a crucial role in various applications, including action recognition, human-computer interaction, augmented reality, and surveillance systems. With the adoption of CNN and availability of 3 dimensional data(along with regular RGB data) via Kinect like sensors the solution to pose estimation using heatmaps has made advancements by leaps and bounds(Sarandi et al., 2021). However not every camera available today is capable of capturing the information so we focused on performing human pose estimation using publically available CASIA-B(Yu et al., 2006) 1 silhouette images.

Gait recognition is a biometric identification technique that focuses on analyzing and recognizing an individual's unique walking pattern or gait.

Figure 1: Sample set of frames of a test subject from Casia-B (Yu et al., 2006) dataset (Chao et al., 2018)

The YOLO (Maji et al., 2022) algorithm is utilized to efficiently detect and localize skeleton keypoints in real-time. By employing a single neural network, YOLO processes the entire image simultaneously, enabling faster inference compared to traditional region-based methods. This approach improves the accuracy and speed of skeleton keypoint detection, contributing to more precise human pose estimation.

In parallel, gait recognition is enhanced through a modified GaitGraph(Teepe et al., 2021) approach. Gait graphs capture and represent the temporal relationships between consecutive frames, effectively modeling an individual's walking pattern. The proposed modifications and feature extraction techniques optimize the gait graph's performance by utilizing additional loss function, updating the prediction from walking conditions to predicting subjects.

In conclusion, through the integration of YOLO for skeleton keypoint detection and the modified gait graph for gait recognition, the combined approach achieves promising results.

2 Related Work

2.1 Human Pose Estimation

Before the advent of prevalent Machine Learning based human pose estimation techniques used to rely on mathematical morphology and geometric operations. One such method was proposed in (Ding et al., 2010) that worked on silhouette images which utilised image preprocessing, edge detection, skeletonization, and skeleton pruning.

With the emergence of Neural Network, the 2D Human pose estimation got divided into two approaches.

1. **Top Down Approach** In case of Top Down Approach first the humans are identified and then the keypoints for each human are identified. Example of such an approach is using Faster RCNN (Ren et al., 2016) for detecting the human bounding box and utilizing Mask-RCNN (He et al., 2018) to detect keypoints as segmentation mask.

2. **Bottom Up Approach** In case of Bottom Up Approaches first the keypoints are detected using single shot after which the attempt is made to group the detected keypoints. YOLO-Pose(Maji et al., 2022) works by first detecting people in an image using the YOLO object detection framework. Once people have been detected, YOLO-Pose then estimates the pose of each person by predicting the location of 25 keypoints on their body. The keypoints that are predicted by YOLO-Pose include the head, neck, shoulders, elbows, wrists, hips, knees, and ankles.

2.2 Gait Recognition

Researchers have proposed various gait recognition methods based on the CASIA-B(Yu et al., 2006) dataset and achieved notable performance. For instance, (Li et al., 2022) proposed a gait recognition method that combines deep learning models with spatio-temporal information. They achieved superior recognition accuracy compared to traditional methods.

GaitSet (Chao et al., 2018) is a person identification based on gait patterns which consists of two main components: a set encoder and a set classifier. The set encoder takes a set of frames as input and outputs a set feature representation. The set classifier then takes the set feature representation as input and outputs a probability distribution over all identities. The set encoder is a convolutional neural network (CNN) and triplet loss to learn discriminative features from gait sequences.

3 Proposed Algorithm

3.1 Algorithm

The Proposed Solution is two fold. We divide the task of Gait Recognition into two tasks Human Pose Estimation and Gait Recognition as follows:

3.1.1 Task 1 - Human Pose Estimation

The initial step involves analyzing an image and plotting its keypoints, known as joints. Subsequently, we will utilize these joint details to generate image embeddings.

The proposed solution involves training a YOLO-based(Maji et al., 2022) pose estimator, specifically utilizing the YOLOv8 version instead of the YOLOv5 version mentioned in the paper. We employ the UltraAnalytics provided API to train the pose estimator, configuring our data format according to COCO formatting rules. Additionally, we modify the joint information by flipping it (e.g., changing the order for left and right joints such as eyes, hands, legs, etc.) since the algorithm flips the image for improved learning. The pose estimator is trained for 50 epochs, with each epoch taking approximately 12 minutes. The network architecture is depicted in the Figure 2

3.1.2 Task 2 - Gait Recognition

In task two, we utilize the joint information to generate gait embeddings and train a classifier to recognize the subject. This task involves working with two sets of data: the original ground truth data and the data generated from Task 1. The purpose of this is to conduct a comparative study between the two datasets.

For this task, the proposed solution involves employing a modified version of GaitGraph2 (Teepe et al., 2021) that incorporates Residual Graph Convolutional Network (RESGCN) (Song et al., 2020) RESGCN utilizes graph convolutional layers to capture both local and global dependencies among the joints of human skeletons. Additionally, the solution incorporates a novel attention mechanism called Part Wise

	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	7360	ultralytics.nn.modules.block.C2f	[32, 32, 1, True]
3	-1	1	18560	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
4	-1	2	49664	ultralytics.nn.modules.block.C2f	[64, 64, 2, True]
5	-1	1	73984	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
6	-1	2	197632	ultralytics.nn.modules.block.C2f	[128, 128, 2, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	460288	ultralytics.nn.modules.block.C2f	[256, 256, 1, True]
9	-1	1	164608	ultralytics.nn.modules.block.SPPF	[256, 256, 5]
10	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
12	-1	1	148224	ultralytics.nn.modules.block.C2f	[384, 128, 1]
13	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
15	-1	1	37248	ultralytics.nn.modules.block.C2f	[192, 64, 1]
16	-1	1	36992	ultralytics.nn.modules.conv.Conv	[64, 64, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.conv.Concat	[1]
18	-1	1	123648	ultralytics.nn.modules.block.C2f	[192, 128, 1]
19	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.conv.Concat	[1]
21	-1	1	493056	ultralytics.nn.modules.block.C2f	[384, 256, 1]
22	[15, 18, 21]	1	1035934	ultralytics.nn.modules.head.Pose	[1, [17, 3], [64, 128, 256]]

YOLOv8n-pose summary: 250 layers, 3295470 parameters, 3295454 gradients

Figure 2: Task 1 Architecture

Attention. This attention mechanism is specifically designed to identify the most important body parts throughout an entire action sequence, resulting in more interpretable representations for various skeleton action sequences.

In our approach, we make modifications to GaitGraph2 (Teepe et al., 2021) by altering the data splitting method and incorporating additional steps in each epoch's end. We adjust the data splitting to follow a different approach than the original intention. Furthermore, we enhance each epoch's end by calculating the confusion matrix to measure accuracy. In addition to utilizing the Supervised Contrastive Loss (Khosla et al., 2021), we introduce Cross Entropy Loss during training to encourage the model to grasp the relationship between labels and predictions. For validation purposes, we employ the Contrastive Loss.

The model is trained using the AdamOptimizer for a maximum of 200 epochs, with a batch size of 768. The final layer of the model provides logits with a size corresponding to the number of classes (in this case, 8). The network architecture is illustrated in Figure.3

3.2 Dataset

For this task, we utilize the publicly available CASIA-B silhouette dataset (Yu et al., 2006). The dataset consists of 124 test subjects observed under three distinct walking conditions: Normal (NM)

	Name	Type	Params
0	backbone	ResGCN	319 K
1	distance	LpDistance	0
2	train_loss	SupConLoss	0
3	train_loss_crossent	CrossEntropyLoss	0
4	val_loss	ContrastiveLoss	0

Figure 3: Task 2 Architecture

with six sequences per subject, walking with a bag (BG) with two sequences per subject, and walking with a jacket/coat (CL) with two sequences per subject. Each view is captured from 11 different angles (ranging from 0 to 180 degrees). However, due to the extensive amount of data and limited computer resources, we narrow our focus to only eight subjects for both of the aforementioned tasks.

To label the keypoints, we utilized the pre-processed data provided by GaitGraph2 (Teepe et al., 2021) due to the large volume of data. This preprocessed data was used as the ground truth for both tasks, with Task 2 being trained on data obtained after Task 1. The data was divided into training and testing sets, but the specific method and ratio of the split differed for each task.

For Task 1, the joint data underwent preprocessing and was converted to the COCO format for

the freely available images. The data was then divided into training and testing sets. Specifically, subjects 001-006 were used as the training data, while subjects 007-008 were reserved for testing (following a 75-25 train-test split). Once Task 1 was completed, a CSV data file was generated. This file contained predicted keypoints and their corresponding confidence values for all the images. The CSV data file is subsequently utilized in Task 2 to facilitate a comparative study.

Likewise, for Task 2, an 80-20 split was performed. All frames from the first scene of "Walking with a Bag" and "Walking with a Coat" were allocated as the test split, while the remaining frames were assigned as the train split for all participants. Since this task involves recognition, it was necessary to generate embeddings for all subjects.

4 Results

The results and metrics from Task 1 are provided in Table 1 and Figure 4, 5, 8.

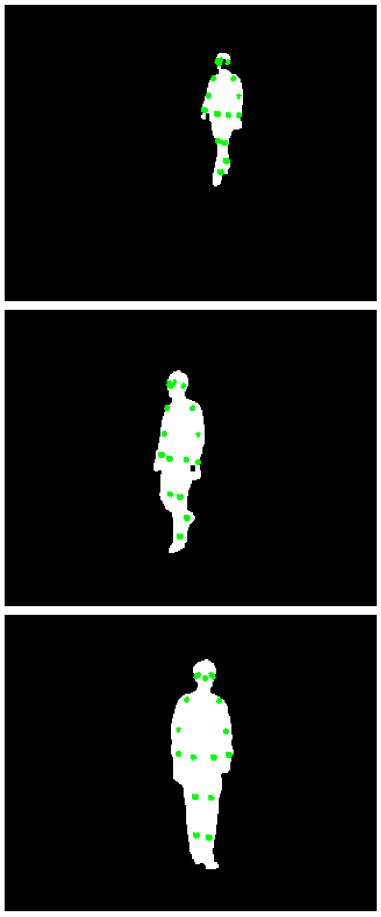

Figure 4: Task 1 Results

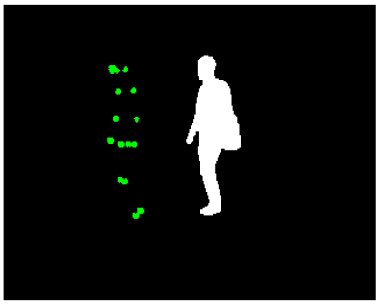

Figure 5: Task 1 Error

For Task 2 the results are provided in Table 2, 3, 4 and Figure 6 7

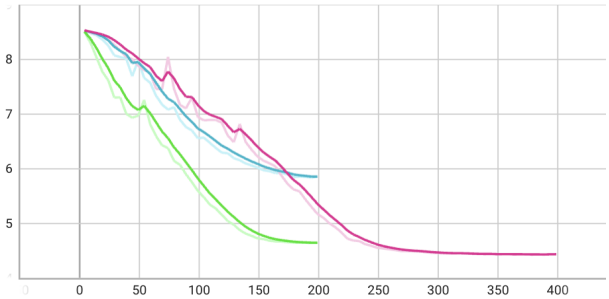

Figure 6: Training loss of Task 2 with loss(y-axis) vs epoch(x-axis) with green(Ground truth) and pink and blue(Data based on Task 1)

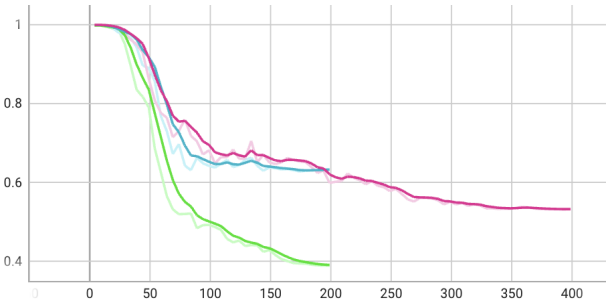

Figure 7: Validation loss of Task 2 with loss(y-axis) vs epoch(x-axis) with green(Ground truth) and pink and blue(Data based on Task 1)

4.1 Quantitative analyses

4.1.1 Task 1 - Human Pose Recognition

The inference per image requires approximately 15ms, and as depicted in Figure 8, the loss function consistently decreases over the training epochs. However, the mean Average Precision (mAP) score reaches a plateau around epoch 20, coinciding with a decline in Recall.

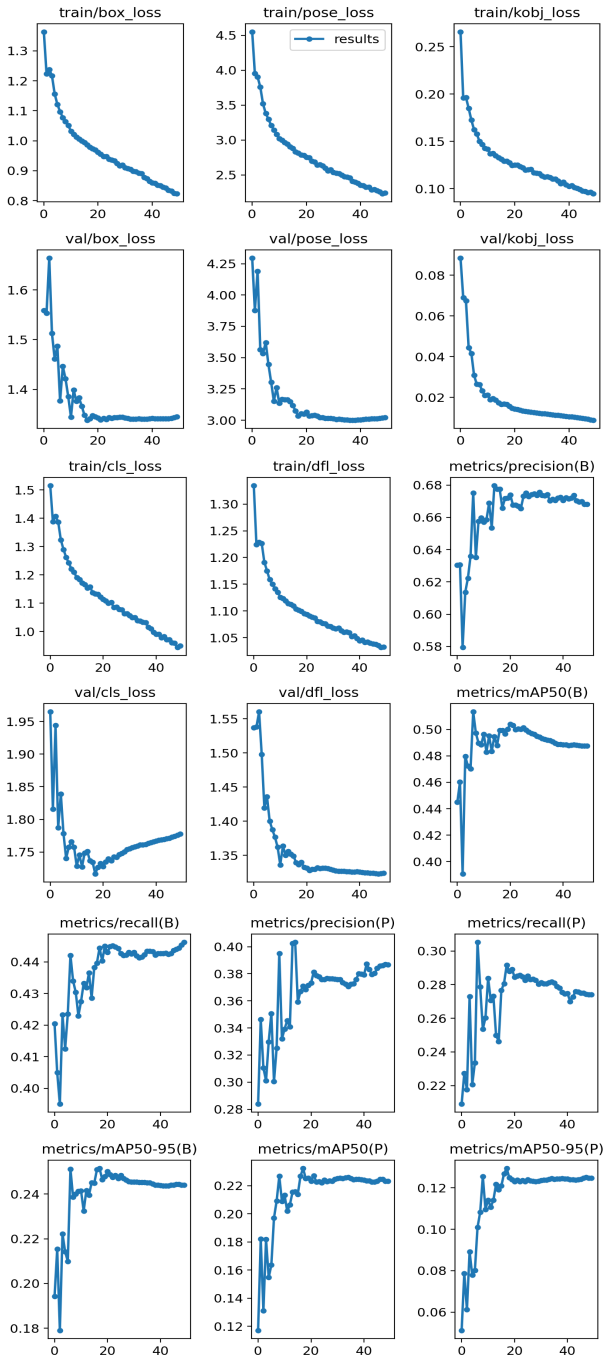

Figure 8: Task one results across various losses and metrics

Loss/Metric	Description
box_loss	Box loss for the human class
pose_loss	Pose Loss
kobj_loss	Keypoint(joints in this case) obj loss
cls_loss	a loss that measures the correctness of the classification of each predicted bounding box and background
dfl_loss	Dual Focal Loss
Metric Precision(B)	Precision for boxes
Metric mAP50(B)	mean Average Precision at an IoU threshold of 0.5 for boxes
Metric Recall(B)	Recall for boxes
Metric Precision(P)	Precision for poses
mAP50(P)	mean Average Precision at an IoU threshold of 0.5 for poses
Recall(P)	Recall for poses

Table 1: Metrics used in Figure 8

In Figure 4, we observe that the poses are accurately estimated even when the user is carrying additional baggage. However, Figure 5 shows that the detected points do not align perfectly with the image. Despite this, the relative distance and position of the keypoints (joints) remain correct. The mAP score is presented in the Table 5

4.1.2 Task 2- Gait Recognition

For Task 2 we should refer Figure 6, 7. The continuous decrease in loss indicates that the model is consistently learning and improving over time.

For Qualitative measurements the results mentioned in Table 2, 3, 4 should be referred. Using the confusion matrix the accuracy for the model can be viewed in Table 6

4.2 Inference

4.2.1 Task 1 - Human Pose Recognition

Due to the low inference time, the proposed model can be utilized effectively in real-time scenarios. However, it is worth noting that errors in the preprocessed data, which were obtained from GaitGraph2 (Teepe et al., 2021), may occur. These errors could be a result of using publicly

Users	User 0	User 1	User 2	User 3	User 4	User 5	User 6	User 7
User 0	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
User 1	9.09%	86.36%	0.00%	4.55%	0.00%	0.00%	0.00%	0.00%
User 2	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%
User 3	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%
User 4	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
User 5	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%
User 6	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%
User 7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

Table 2: Confusion matrix when trained for 200 epochs on ground truth data with 8 persons

Users	User 0	User 1	User 2	User 3	User 4	User 5	User 6	User 7
User 0	100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
User 1	81.82%	0.00%	9.09%	9.09%	0.00%	0.00%	0.00%	0.00%
User 2	0.00%	0.00%	54.55%	0.00%	0.00%	0.00%	45.45%	0.00%
User 3	9.09%	0.00%	36.36%	54.55%	0.00%	0.00%	0.00%	0.00%
User 4	0.00%	0.00%	0.00%	55.56%	0.00%	22.22%	0.00%	22.22%
User 5	0.00%	0.00%	33.33%	33.33%	0.00%	27.78%	0.00%	5.56%
User 6	0.00%	0.00%	66.67%	0.00%	0.00%	0.00%	4.76%	28.57%
User 7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

Table 3: Confusion matrix when trained for 200 epochs on data generated from Task 1 with 8 persons

Users	User 0	User 1	User 2	User 3	User 4	User 5	User 6	User 7
User 0	86.36%	13.64%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
User 1	4.55%	86.36%	0.00%	4.55%	0.00%	4.55%	0.00%	0.00%
User 2	0.00%	0.00%	81.82%	0.00%	0.00%	0.00%	18.18%	0.00%
User 3	0.00%	9.09%	0.00%	86.36%	0.00%	4.55%	0.00%	0.00%
User 4	0.00%	0.00%	0.00%	0.00%	88.89%	0.00%	0.00%	11.11%
User 5	0.00%	0.00%	0.00%	16.67%	0.00%	83.33%	0.00%	0.00%
User 6	0.00%	0.00%	14.29%	0.00%	0.00%	0.00%	85.71%	0.00%
User 7	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

Table 4: Confusion matrix when trained for 400 epochs on data generated from Task 1 with 8 persons

Model	mAP
HRNET	75.8
YoloV8(ours)	48.89

Table 5: mAP for Task 1 and Comparison with State of The Art HRNET

Data Source and epochs	Accuracy
Ground Truth 200 epochs	98.29%
Task 1 data 200 epochs	42.70%
Task 1 data 400 epochs	87.35%
GaitSet	95.00%

Table 6: Accuracy for Task 2 and Comparison with State of The Art Gaitset

available PNG versions instead of the JPEG images specified in the GaitGraph's provided CSV file. Consequently, there are instances where the model predicts incorrect outputs, as demonstrated in the Figure 5.

4.2.2 Task 2- Gait Recognition

The model demonstrates better performance when trained and evaluated with ground truth labels compared to using the data generated after Task 1's training and inference. Additionally, it takes twice as long to reduce the loss when utilizing the values achieved from the ground truth compared to the data generated post Task 1.

4.3 Comparison with State of the Art

4.3.1 Task 1 - Human Pose Recognition

Figure 5 reveals that the model encounters difficulties even when tested on the dataset. However, a comparative analysis presented in Table 5 indicates that the state-of-the-art (STOA) method outperforms our approach. It is important to note that this comparison is not ideal since the ground truth used for evaluation was generated using GaitGraph2 (Teepe et al., 2021) (as mentioned in Section 3.2) which internally has been generated using the State of the art.

4.3.2 Task 2- Gait Recognition

In Gait Recognition, the state-of-the-art approach is GaitSet (Chao et al., 2018). While GaitSet reports accuracy for various walking conditions and different views, the overall comparison is presented in Table 6. Although our approach achieves lower overall accuracy compared to the Ground Truth, it is important to consider that the GaitSet model involved calculations with equal to or more than 24 subjects, whereas our evaluation was conducted with only 8 test subjects.

5 Future work

In the future, instead of relying on silhouette images obtained from regular RGB images, an alternative approach could be explored. Converting RGB images to silhouette images involves additional processing steps that may result in the loss of valuable information that could potentially aid in identifying keypoints (joints). To address this, leveraging RGB-D based images and extracting skeletal information, as proposed in MetRabs (Sarandi et al., 2021), could be considered. This approach would allow for gait recognition to be performed on a larger number of test subjects, going beyond the current limitation of 8 subjects.

6 Conclusion

In conclusion, we have developed a gait recognition system using silhouette images to identify and differentiate among 8 users. However, considering the accuracy results presented in Table 6, it is evident that the system's performance is not highly accurate. Consequently, it is more suitable to utilize this system as a soft biometric rather than relying on it solely for precise identification purposes.

References

Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. 2018. Gaitset: Regarding gait as a set for cross-view gait recognition.

Jianhao Ding, Yigang Wang, and Lingyun Yu. 2010. Extraction of human body skeleton based on silhouette images. *Education Technology and Computer Science, International Workshop on*, 1:71–74.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2018. Mask r-cnn.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron

Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning.

Huakang Li, Yidan Qiu, Huimin Zhao, Jin Zhan, Rongjun Chen, Tuanjie Wei, and Zhihui Huang. 2022. Gaitslice: A gait recognition model based on spatio-temporal slice features. *Pattern Recognition*, 124:108453.

Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. 2022. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2016. Faster r-cnn: Towards real-time object detection with region proposal networks.

Istvan Sarandi, Timm Linder, Kai Oliver Arras, and Bastian Leibe. 2021. MeTRAbs: Metric-scale truncation-robust heatmaps for absolute 3d human pose estimation. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(1):16–30.

Yi-Fan Song, Zhang Zhang, Caifeng Shan, and Liang Wang. 2020. Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*. ACM.

Torben Teepe, Ali Khan, Johannes Gilg, Fabian Herzog, Stefan Hörmann, and Gerhard Rigoll. 2021. GaitGraph: Graph convolutional network for skeleton-based gait recognition. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2314–2318.

Shiqi Yu, Daoliang Tan, and Tieniu Tan. 2006. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. volume 4, pages 441–444.