Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Investigating The Performance Of Joint-based Human Action Recognition Models in Children

Sherin Miriam Cherian, BTech

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Data Science)

Supervisor: Dr. Inmaculada Arnedillo-Sanchez

August 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Sherin Miriam Cherian

August 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Sherin Miriam Cherian

August 19, 2022

Investigating The Performance Of Joint-based Human Action Recognition Models in Children

Sherin Miriam Cherian, Master of Science in Computer Science

University of Dublin, Trinity College, 2022

Supervisor: Dr. Inmaculada Arnedillo-Sanchez

This project seeks to perform Human Action Recognition on video datasets of children for a given set of actions, namely - Hop Left, Hop Right, Gallop and Skip. The objective is to automatically recognise if a certain defined set of fixed criteria match the actions performed by the children in the videos. For this purpose, individual joint locations were detected and used to create a heuristic based approach to label the videos frame-by-frame. The labelled joint coordinates data was used to train different models to predict the labels of frames as a supervised classification problem; and observe what heuristic performs the best for a particular action by evaluating their performance in terms of frame-by-frame classification accuracy and pre-collected human based analysis of the data. It was observed that the models trained based on this approach perform the best for the Hop Left/Right action, followed by Skip. The project also discusses the challenges of working with raw video data and what approaches can be used to solve some of them - based on existing literature in the area of Human Action Recognition and new approaches to fit the dataset.

Acknowledgments

I would like to thank my dissertation supervisor Dr. Inmaculada Arnedillo-Sanchez and Benoit Bossavit for their guidance and support throughout the project period. Their inputs helped the project to progress at multiple points where I faced difficulty and needed advice. I would also like to thank my fellow course mates Arun Jayaprakash, Tom Thomas Matthew, Azin Makarnath and Manu Prasannakumar who also worked under Dr. Sanchez on similar projects for the continual sharing of their knowledge and learning during our weekly meetings that we all mutually benefited from. These discussions helped brainstorm different ideas, think of solutions from different perspectives and effectively problem solve - all of which was highly beneficial for a project as complex as this. I thank Dr. Jason Wyse for his feedback during the demo presentation.

I also thank my other course mates in Trinity, my friends back in India and my parents for being a constant source of support and encouragement in the past one year, during what I believe was one of the most challenging phases of my life.

Sherin Miriam Cherian

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.0.1 Human Action Recognition

Human action recognition is used in a variety of applications, from rehabilitation (often athletic), preventing repeated injuries, analyzing progress or mastery in certain movements related to a sport, monitoring of human activities, energy conservation in buildings, anomaly detection, surveillance, security etc. It removes the need for labor intensive manual monitoring by automating the process entirely, hence greatly cutting these costs. A lot of the state-of-the-art human activity recognition algorithms and models are trained to perform classification based on datasets of adult human beings, for privacy reasons, legal and ethical complications. There is also a general lack of such publicly available data – both facially obfuscated (i.e. anonymised) and not. This project aims to perform human action recognition on a collected dataset of children. It also seeks to identify improper actions that are performed by the children. This can be used to analyse and screen the development of gross motor skills in children, leading to the early detection and intervention in case of identified issues.

The proposed approach in this project involves extracting the location of the bodily joints frame-by-frame in the videos. Google AI's BlazePose based pose detection framework MediaPipe was used for this. Since the dataset consists of blurred faces, the project will also investigate how well such frameworks can handle anonymised data. Much of the existing research surrounding human action recognition involves the usage of wearable sensors, smartphone sensors or depth based sensors like the MS Kinect for skeletal detection. This may not necessarily be the most accessible solution for an action recognition model that identifies issues in motor skill development, and hence this project focuses on an image/video based model for action recognition.

1.0.2 Research Objectives

The research objectives of the project involve the following:

- Finding methods to extract joint/landmark data from video data.
- Investigating methods to best deal with video data that is not labelled/ annotated.
- Creating models and heuristics to analyse and automatically predict different aspects of the data such as the number of times an action is performed.
- Evaluate model performance based on real-time results.

1.0.3 Report Structure

The rest of the report is organised by the following overall structure. The Background section will go through different types of actions, types of Human Action Recognition (HAR), the advantages and disadvantages of these, the working of MediaPipe and existing research in the fields of HAR and HAR for gross-motor skill analysis in children. The Methods section will go through the description of the dataset and the overall flow of the project discussing aspects like joint extraction, heuristics, normalisation, model training etc. The Experiments and Results section evaluates model performance on different actions and discusses the results to understand their significance. Finally, the Conclusion and Future Work section summarises the project and discusses future work and possible improvements.

Chapter 2

Background

2.1 Types of Actions

Das Antar et al. (2019) divides ambulation activities into 3 categories - static, dynamic and postural transitions. We add a further subdivision of dynamic based on whether the activity is translational in nature or not, as illustrated in Figure 2.1. The takeaway is that it is difficult to distinguish between actions that are have the same bodily movements with only slight differences in intricacies, for instance the difference between run, walk and jog is only based on pace and minor differences in stance.

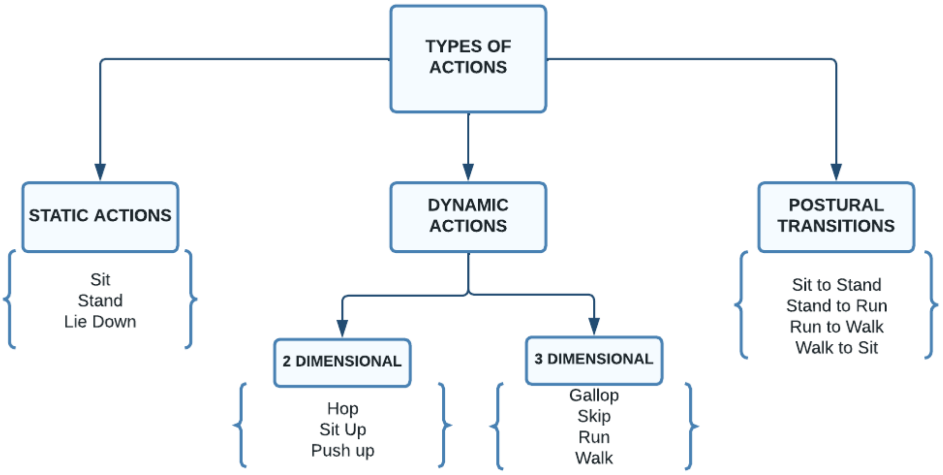


Figure 2.1: Types of Ambulation Activities

The paper also mentions that postural transitions like standing up or sitting down are easier to recognise than repetitive activities like running, walking. As an extension to that, some of the 2-Dimensional activities like Sit Up are also easier to recognise than 3-Dimensional repetitive activities like Skip/Run.

2.2 Types of HAR

Human Activity or Action Recognition (HAR) is traditionally tackled using a time series based approach that is used to either classify different actions or different sub parts of an action. A lot of the existing work in the area is based on sensor data from wearable sensors as part of Ubiquitous Sensing, which is the idea of extracting useful information from data gathered by pervasive sensors (Perez et al., 2010). HAR is used for a variety of purposes - from monitoring how well patients with diabetes or heart diseases perform the exercises prescribed to them as part of recovery and treatment (Jia, 2009), (Samir et al., 2021), monitoring sleep quality and finding its correlation to physical activity (Sathyanarayana et al., 2016) etc. Broadly, HAR is performed by sensors that either present on the bodies of participants, or present externally - which either prompt the participants to interact with them or are simply present in the environment like cameras.

Yin et al. (2008) explore an Support Vector Machine (SVM) and kernel Non Linear Regression (KNLR) method to detect abnormalities in human activity based on data from wearable sensors. They talk about how existing methods have a high false positive rate because most of the available and collected data is not abnormal, making models biased towards the normal data. Data from smartphone mobile embedded sensors like accelerometers and gyroscopes have also been used for the purpose of HAR in combination with Deep Neural Networks (Ronao and Cho, 2016), (Zeng et al., 2014), (Jiang and Yin, 2015). The MS Kinect is a depth and camera based high accuracy motion sensor developed by Microsoft. It works by means of a high resolution RGB Camera and an infrared depth sensor in combination with multiple microphones. The Kinect can identify and track 48 key points or landmarks on the body, based on Machine Learning models trained to represent different age groups, genders and build of body. Developed for the purpose of motion sensing gaming consoles, the Kinect has since been explored in multiple other use cases such as posture and movement recognition in workplaces (Dutta, 2012), movement measurement for patients with Parkinson's disease (Galna et al., 2014) etc. Because of the Kinect sensor's highly accurate landmark detection, it can be used for more than just the traditional posture and gesture recognition, like by Liu et al. (2013) where HAR is performed with by means of a 2 step process of action learning and recognition accomplished with K-Means Clustering and Hidden Markov Modeling.

2.2.1 Sensor Based Challenges

Some of the disadvantages that sensor-based HAR systems pose are human discomfort (Lara and Labrador, 2012), and although smartphones are an alternative solution to this, it requires users to constantly carry them around and contributes to a quicker loss of energy in the phone's battery. They are also highly dependent on the location of the sensor, as readings can differ at different parts of the body for the same action (Das Antar et al., 2019). Unlike traditional cameras, depth based sensors like the MS Kinect are expensive and not readily available in all environments, often requiring a separate installation and set up that not everyone can opt for.

2.3 Vision Based Approaches

As an alternative to sensor-based approaches, sole reliance on cameras or video feed is also used for HAR, although there is lesser literature in the area as mentioned in a survey by Gupta et al. (2020), owing to it being a relatively newer field. Vision based approaches of HAR generally consists of the following steps (Banjarey et al., 2021) - (1) Pre-processing - Removal of background or noise around the person. (2) Creation of bounding boxes around the person. (3) Normalisation/ calibration of cameras. (4) Training Deep Learning/Machine Learning algorithms on extracted joints. (5) Analysing classification/ prediction results.

2.3.1 Using Body Landmarks

In specific, pose estimation can take place by means of two approaches as mentioned by Banjarey et al. (2021). The first is a bottom-up approach where each of the landmarks are detected and then put together as groups of different skeletons in the frame. The second approach is top-down, where an object detector is used first to identify different targets to draw bounding boxes around, followed by identifying the landmarks in each of the boxes.

The usage of pose estimation and body landmarks from normal camera feed as an alternative to wearable sensors for HAR has been explored in Gatt et al. (2019) for the use case of anomalous fall detection using PoseNet (Papandreou et al., 2017) and OpenPose (Cao et al., 2017), two publicly available pre-trained pose estimation models. Only skeletal data and no further feature engineering is used for their approach. Additionally, they also discard frames of the videos where the human detection confidence is low, as these frames do not provide any useful information for the models to learn. The pose estimation data was converted to 1-D time series data and fed into deep LSTM and CNN models to perform anomalous activity detection with the aid of the Tensorflow and Keras Python libraries and GPU support. Zhang et al. (2019) discuss a CNN and LSTM based method for HAR that is adaptive in nature. It has the ability to re-position or transform the skeletal joints based on a module that was trained to identify the best observational viewpoints to perform the HAR task.

Yang et al. (2022) discuss a method to perform HAR with time-based path signatures extracted from skeletal joint data. For pre-processing, they perform augmentation by the adding of Gaussian noise over the skeletal joint coordinates. They also normalise the data by converting world centric joint coordinates to their relative coordinates wrt to the center of the person's body. Biometric differences from person to person are compensated by normalising the joint coordinates between [-1,1].

2.3.2 Challenges

There are multiple issues that come with vision based approaches to HAR (Lara and Labrador, 2012). One of them is privacy - which hugely limits the amount of data that is publicly available for ongoing research. Not everyone is willing to be monitored on a constant basis and most of the publicly available datasets explored and collected in this domain are that of adults (Banjarey et al., 2021). Some examples are the UCF-101, UCF Sports Dataset, UCF-50 etc. The issue of privacy is trickier and more crucial in the case of children, further restricting the availability and accessibility of datasets. Hence, a lot of the vision-based models that are trained to perform HAR are biased towards older individuals. The second issue deals with pervasiveness as a lot of the data also depends on the picture quality of the camera and its positioning relative to the body of the subject that is to be monitored. The third issue is scalability. A lot of the approaches that handle video data are based on Convolutional Neural Networks (CNN) and other Deep Learning architectures that learn important features by pixel-wise matrix based mathematical calculations that are computationally intensive, hence increasing the reliance on appropriate infrastructure that is not necessarily always scalable. These reasons contribute to why there is lesser research in the area, and why sensor based HAR systems are still a more popular option, despite their drawbacks and in some cases, their lesser accuracy.

2.4 MediaPipe

MediaPipe (GOOGLE, 2020) is a ML based real-time pose detection library developed by Google AI. It is similar to state-of-the-art approaches like OpenPose and PoseNet in output, but unlike them is lightweight enough to be suitable for mobile phones, laptops and the web, and not just powerful desktop environments with GPU support. MediaPipe's Machine Learning pipeline works based on the top-down approach mentioned in 2.3.1 - a two step process that first locates the ROI (Region of Interest) and then identifies 33 skeletal joint landmarks in the ROI. This is a superset of the COCO topology standard of 17 landmarks (COCO, 2020). The first step of locating the ROI or pose estimation is performed with the lightweight BlazePose detector (Bazarevsky et al., 2020), (Ivan Grishchenko, 2020). This is accomplished by predicting the mid-point of the hip and the radius of the circle that encloses the person, along with the angle between the shoulder and the mid point of the hip. Multiple applications such as fitness trackers can be developed with MediaPipe with use cases such as counting the number of push ups etc. More details about MediaPipe's functionality is explored in 2.4.

2.5 Gross Motor Skills

Gross Motor Skills that are acquired during childhood represent normal physical development in children. An early detection of issues related to them can lead to proper timely rectification. There are several standards and tests to evaluate gross motor skills that assess full body movements, and these vary from country to country. TMGD-3 (Test of Gross Motor Development) (Prieto et al., 2019) is one such example for the United States and consists of 13 skills for evaluation. These require manual observation, checking and scoring by formally conducting the proposed tests on a cohort of participants. This kind of manual approach is subject to a deficit of skilled professionals and time (Suzuki et al., 2020). Is it hence of interest to develop an automated approach that can evaluate gross motor skill development. Bossavit and Arnedillo-Sánchez (2019) developed 5 activities for children between the ages 2-7 years and designed MotorSense (Bossavit and Arnedillo-Sánchez, 2020) , an automated approach that can use depth based sensors like the MS Kinect to evaluate gross locomotor skills by a series of interactive activities in a virtual environment. This involved recreating or mirroring the movement of the children in the environment and providing audio and visual inputs and feedback that guided them through the testing process.

2.5.1 Research in HAR

Suzuki et al. (2021) discuss a an approach to identify anomalous or improper body movements in children based on the TGMD-3 standard that was discussed above. This TGMD-3 manual assessment was carried out by skilled professionals. The paper uses the OpenPose library for skeletal detection as a base and built a person tracking module of top of it. The skeletal data from this module was used to build time series input for Deep Learning models. Videos of the children were shot from different angles to increase amount of the video data. These videos were also cut or segmented into shorter clips of 2-4 seconds to create a uniform input to feed into the Deep Learning models that created motional time series images, which were divided into normal and abnormal images based on the previously mentioned assessment. The normal images were used to train an Auto-Encoder and hence when an abnormal image is fed into this network, its pseudo correct image is the output of the decoder. A comparison of this pseudo-correct image and the faulty image can identify the region/limbs of the fault or anomaly. The reported accuracy of this method was 99.3%. The series of papers by Suzuki et al are between the years 2020-2021 and there has been not much other research in the area of HAR for gross-motor skill evaluation in children.

Chapter 3

Methods

The end-to-end pipeline followed in this project is described in this section.

3.1 Data Description

The dataset was collected as part of a separate study by Bossavit and Arnedillo-Sánchez (2019) that aimed to devise a set of actions and criteria that checks for possible issues of motor-skill development in young children. The data is in the form of video feed captured for different age groups of children performing the set of actions that were devised.

The set of actions that were assigned to me were Hop Left, Hop Right, Skip and Gallop. Each of the actions are defined by a set of criteria that need to be checked, and these are given in Table 3.1. Each of the 4 actions is divided into two sets of age - younger and older kids, making 8 sets of data in total. The filenames are constructed in a format that is indicative of important metadata as follows - action_number_gender_level_trial_anonymised.

Action	**Requirement**	**Criteria**
Hop Left	Count Number of Successive Hops	Jump starts on left leg and ends on left leg.
Hop Right	Count Number of Successive Hops	Jump starts on right leg and ends on right leg.
Skip	Count Number of Skips	Step forward, Hop on same foot, alternate feet.
Gallop	Count Number of Gallops	Jump starts and ends with same foot in front.

Table 3.1: Criteria for Different Actions

Where "level" and "trial" refers to the number of attempts to perform a certain action. This is so children can get accustomed to and understand the action they are performing and do it to their best ability. Each child is given 3 opportunities to perform an action - Level 0,1,2. Sometimes the child is asked to perform the action a second time - Trial 1,2. For example, in the action Skip, the Level 2 and Trial 2 videos would be a child's best attempt at performing the Skip action with the desired criteria.

The children stand on a mat with a marked black center at the start of every action. Ideally this mat should also be at the center of the frame in the video, but this is not the case because the camera is not placed at the same position in every video. The children are also dressed in their school uniforms, which are often monochrome and of the same color from head to toe. Both of these and other issues with the data that can pose problems for image-based action recognition. These are discussed in more detail in the subsequent sections.

3.2 Ethics Approval and Privacy

The dataset involves videos of children across different ages and hence requires an ethics approval, which has been obtained by my supervisor Dr. Sanchez. In addition to this, the data is anonymised by blurring the faces of the children. In cases where the children's faces are not blurred, they are wearing a face mask. There is no other data available that is related to the identity of the children, other than their gender. The methods used in this dissertation are based on only the skeletal data of the participants.

3.3 Overall Flow

3.3.1 Noise in Data

- **Inconsistency in data length:** Traditional approaches in image and video-based human action recognition methods often use LSTM and CNN models. LSTMs are a type of RNNs used to learn temporal relationships within data and could potentially be useful in learning the sequence of steps in a particular action. However, all of these models would expect input data of the same size and in this case that translates to videos of the same length/ number of frames. The data however is not of this nature, as the videos are of varying length depending on a number of factors - how much time a child takes to do an action, the point at which the child starts the action and the number of times the action is repeated. Another issue associated with the usage of time-based deep learning models is that the timestamps where different sequences or phases of the action occur different from video to video based on the physical ability and build of the child. For e.g. some children may hop higher, or alternatively make shorter and more frequent hops. There are no uniform or consistent timestamps where different phases of an action occur at the same time from video to video.

- **Unwanted Background:** Many of the videos have more people than just the children performing the action. There is often people walking in the background, and sometimes a second adult positioned towards the side of the frame, who is showing the child how to perform the action by doing so themselves. This poses a problem to the action recognition process and requires a method to single out on the data that is useful from a single frame of the video. The sheer size of the dataset makes it cumbersome to manually crop out unwanted actors in the background. Figure 3.1 shows an example of this where there are multiple other actors in the frame of the video and someone else's landmarks are recognised instead of the child's.

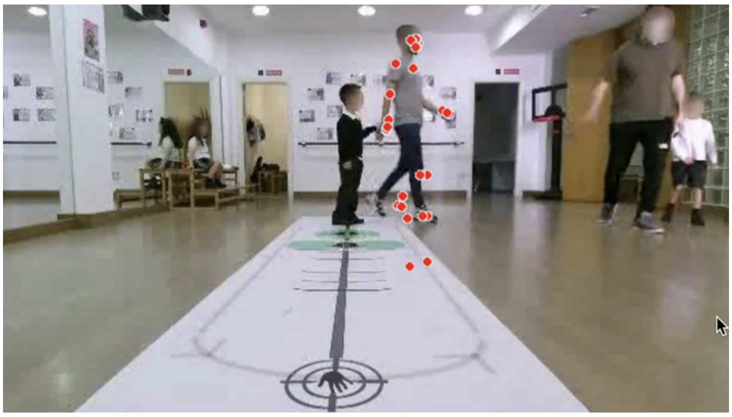


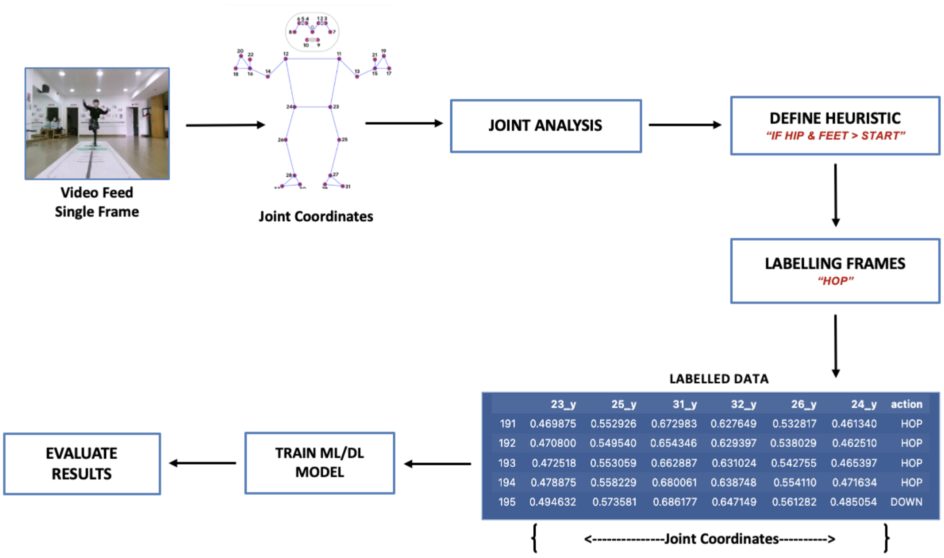
Figure 3.1: Multiple people in the background of the frames.

Manually annotated, standardized data.

- **Random Outliers:** There are also multiple videos that are not useful - either because of bad format, or the children not performing any action at all for the entire length of the video, or performing the wrong action/ being very inconsistent. A few of the videos are also unusually sped up. While the aim of the project is to identify if children are performing actions as per requirements, training a model requires data that exhibits patterns that are learnable and such outliers throw off the models and affect their accuracy and ability to effectively learn from the data that is usable.

3.3.2 Proposed System

In order to solve the issue of inconsistent number of frames for input to the model, a new method is devised to extract the joint locations from each frame of a video, and label the frames based on a pre-defined heuristic. Models are then trained on this labelled data with the goal of being able to classify each frame in the video as a specific phase of an action. This is better described with the flow chart in Figure 3.2.



	23_y	25_y	31_y	32_y	26_y	24_y	action
191	0.469875	0.552926	0.672983	0.627649	0.532817	0.461340	HOP
192	0.470800	0.549540	0.654346	0.629397	0.538029	0.462510	HOP
193	0.472518	0.553059	0.662887	0.631024	0.542755	0.465397	HOP
194	0.478875	0.558229	0.680061	0.638748	0.554110	0.471634	HOP
195	0.494632	0.573581	0.686177	0.647149	0.561282	0.485054	DOWN

Figure 3.2: Overall end-to-end process

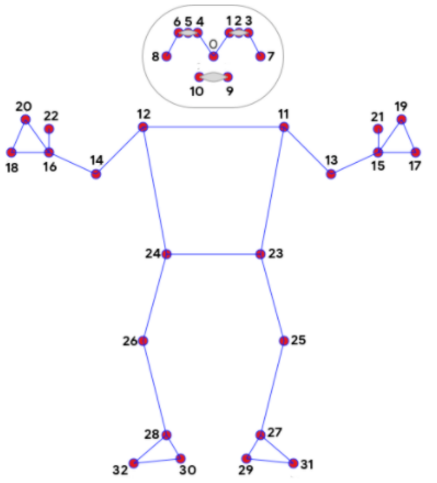
As an example, the flowchart describes the process taking the **Hop Left** data. Bodily joint locations are extracted from every single frame of the video and are analysed to define a heuristic. For instance if the location of the hip and leg are above a certain value, the label for that specific frame is defined as "Hop". This collection of labelled data is then used to train Machine Learning and Deep Learning models to classify individuals frames. The models are evaluated against test data and hold out data based on both how well they can classify individual frames, and by how accurately they can overall recognise different actions and criteria. For example how well a model can classify a frame as a "Hop" and how well it can count the number of valid Hops in a number of videos. A similar process is followed for the other actions Gallop and Skip as well.

3.4 Data Preparation

3.4.1 Joint Extraction - Media Pipe

The second step in Figure 3.2 is performed with Google AI's pose detection library Media Pipe which is based on the BlazePose Model as described in Section 2.4. The library provides multiple solution API that help solve some of the issues related to noise in data, mentioned in Section 3.3.1. Media Pipe identifies 33 joint landmarks as shown in Figure 3.3 and also a segmentation mask of the Region of Interest (ROI). The output coordinates **POSE_LANDMARKS** contain the following:

- **x:** the x coordinate of the landmark.



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

Figure 3.3: 33 Landmarks of Media Pipe

- **y:** the y coordinate of the landmark.
- **z:** the depth or the landmark as taken from the mid-point of the hips. This value gets smaller as the person/ landmarks move closer to the camera.
- **visibility:** indicates how visible the landmark is in the image, whether it is occluded or jut absent overall.

The blurring of the faces or existence of face masks does not seem to affect MediaPipe's ability to recognise a person in the frame and identify the desired joint landmarks. There are also the following solution APIs that were tuned as per the requirements of this project:

- **STATIC_IMAGE_MODE**: This is used to set the input stream to a video over a single image. If set to a video stream, MediaPipe attempts to detect/track the most prominent person in the first frame of the video and then subsequently tracks those landmarks for the rest of the video. This helps reduce the latency and computational power of the system.
- **MODEL_COMPLEXITY:** [0-2] The accuracy of ROI and landmark detection goes up as this is increased.
- **MIN_DETECTION_CONFIDENCE:** [0-1] The threshold of probability for a person or Region of Interest to be detected. This can be varied and changed to a higher value to put more emphasis on the main actor in the video i.e. the child performing the action and not the people in the background. It is currently set to 0.5 and can be modified.

- **MIN_TRACKING_CONFIDENCE:** [0-1] The threshold of probability for the landmarks to be tracked. A higher value of this would mean that the model would more often try to re-invoke the detection of a ROI in subsequent frames. This would improve robustness but affect the latency and performance of the system.

3.4.2 Joint Extraction - Limitations

There are some limitations to MediaPipe's accuracy and robustness of landmark detection wrt this dataset. These are based on the 2 stages of MediaPipe's ML pipeline (1) Identifying the ROI in the frame (2) Identifying the 33 landmarks in the ROI.

Validity of data:

There are videos where MediaPipe fails to recognise any landmarks for a majority/ all of the frames. These frames are not usable for training models as no joint data can be extracted from them. On observing different videos with similar issues, it can be concluded that there are two major reasons for this. Figure 3.4 shows some examples of such invalid frames. The first reason is uniform or clothing that the child is wearing, which is often monochrome, and seems to make it difficult for MediaPipe to recognise landmarks. In some cases as in (A) of Figure 3.4, the background matches the color of clothing and hence MediaPipe cannot discern a ROI in the frame. In (B) we see that the child is standing oriented sideways to the camera i.e. the front of the body is not fully visible, which also makes it difficult for landmarks to be identified.

In order to understand how much of the data needs to be potentially discarded because of the existence of these ”invalid” frames, a script was written to go through the data and collect information about the percentage of invalid frames in every video, for different batches of data. A threshold of 50% was chosen to differentiate the videos i.e. if more than 50% of the frames in a video are ”invalid”, the video is classified as not useful and discarded. Figure 3.5 shows the number of discarded videos vs total number of videos for different sets of data. It can be seen that the invalid videos are negligibly small and not a sizable amount to be concerned about if removed. Hence these videos were removed. This is similar to the approach by Gatt et al. (2019) mentioned in 2.3.1

Improper Landmarks:

There are also videos where MediaPipe does recognise landmarks, but not accurately. There are a variety of reasons why this can happen - most often just as discussed earlier, because of orientation of the body towards the camera or because of monochrome clothing. These kind of outliers can mislead models that are trained on the data and affect the

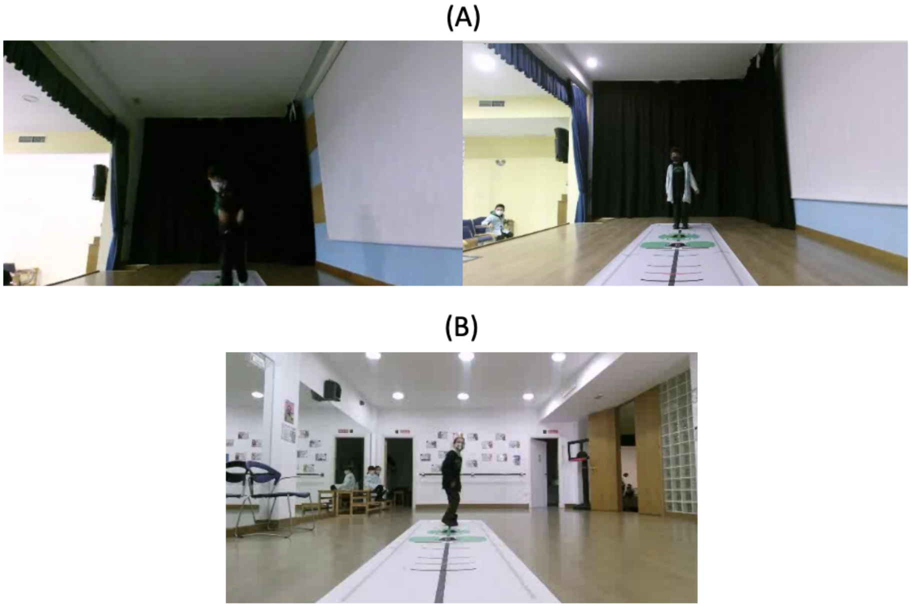
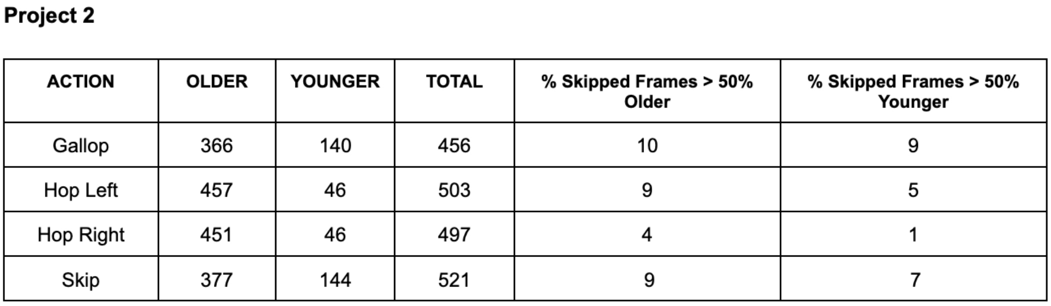


Figure 3.4: Frames where MediaPipe cannot recognise a ROI or landmarks.



Project 2

ACTION	OLDER	YOUNGER	TOTAL	% Skipped Frames > 50% Older	% Skipped Frames > 50% Younger
Gallop	366	140	456	10	9
Hop Left	457	46	503	9	5
Hop Right	451	46	497	4	1
Skip	377	144	521	9	7

Figure 3.5: % of Invalid Data vs Total Data

performance of action recognition. Figure 3.6 shows an example of MediaPipe giving an output of completely wrong landmarks for an example of the Gallop action.

It can be seen that the child is standing to the side, with the front body not facing the camera. Unlike before, it is harder to determine how much of the data is affected in this way by means of a script or an automated method. It is also cumbersome to manually go through the data and remove such videos.

Normalization:

The normalisation of joint landmarks is necessary to ensure that the model properly learns from the data and is not mislead by numerical differences that do not represent any useful information about the nature of the actions. This is especially relevant because of the

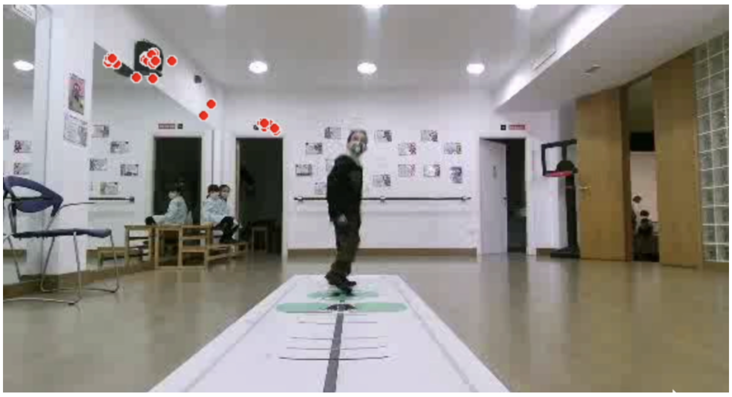


Figure 3.6: Improper Landmarks - Gallop Example

nature of the video data, where we are dealing with joint locations of videos that are not all in the same frame of reference.

MediaPipe provides an existing normalisation of the x and y coordinates of the identified joint locations such that they lie between [0,1]. This was further improved by a hip-based normalisation technique, where the original coordinates of the landmarks were transformed into a plane/frame with a fixed size, such that the new coordinates are the relative offsets from the location of the left hip which is fixed. These offsets are calculated from the distance between the landmarks and the starting position of the hip (i.e. the position of the hip in the first frame where MediaPipe can correctly identify the ROI and landmarks) and then mapped to the newly constructed plane.

Note: The above mentioned normalisation technique was developed by me and other peers working under Prof. Sanchez as a collaborative effort.

3.5 Heuristics

3.5.1 Heuristic - Hop Left

The **Hop Left** action is defined by the criteria of hopping on the left leg throughout the video, while the right leg stays above the ground for the entire duration. The requirement is to count the number of such valid hops. The key objectives here are (1) Checking if the left leg is rising above the ground level during a hop (2) Checking if the right leg stays above the ground. So the aim is to develop a heuristic that can accommodate both of these objectives to the best of its ability. In order to do this the videos were examined to understand if there are any learnable patterns in the movement of the joints from frame to frame. Out of the 33 landmarks that MediaPipe can recognise, the hip and legs were examined as the Hop action involves mainly their movement. Figure 3.7 shows the y coordinates of the hip + leg landmarks over a video. Note that MediaPipe's axis is oriented such that a decrease in the y coordinate means upward movement.

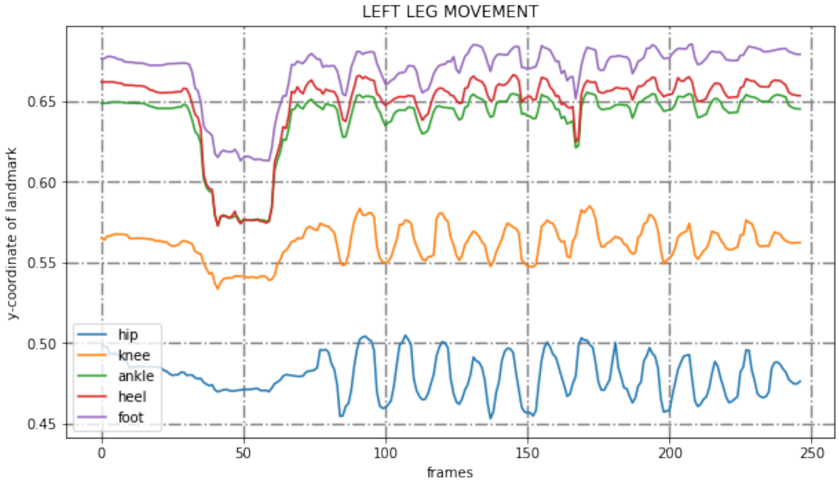


Figure 3.7: Analysing Left Leg + Hip Movement - Hop Left

We can see that the hip shows the most prominent oscillations about a a central mean as the child hops up and down on the ground. As we go further down the body to the knee, ankles and feet, the oscillations get lesser and lesser distinct. This is because the children do not hop by very large heights, and the leg often gets displaced i.e. the leg do not go up and down in a constant straight line. The hip exhibits a more stable kind of movement that can be learnt from and can be used to develop the heuristic. We also want to check if the right leg ever touches the ground once the child begins to hop. MediaPipe's **POSE_LANDMARKS** are very sensitive to the smallest of change, so if we were to track the starting position of the right foot on the ground and check if the subsequent movement was above that, it would not work as expected because of the perspective and distance from which the camera is situated. Instead we pick a range as shown in the Figure 3.8 and define the heuristic such that the right leg is considered to be above the ground when it is within the range of between the max height of the foot and the 75% quartile height of the foot. All of this is collectively used to define the heuristic with the labels in 1.

Algorithm 1 - HOP LEFT HEURISTIC

```
if Right Foot between Max Height and 75% quartile height then
    if Left Hip > Mean and Left Foot above ground then
        action ← JUMP
    else
        action ← DOWN
    end if
end if
```

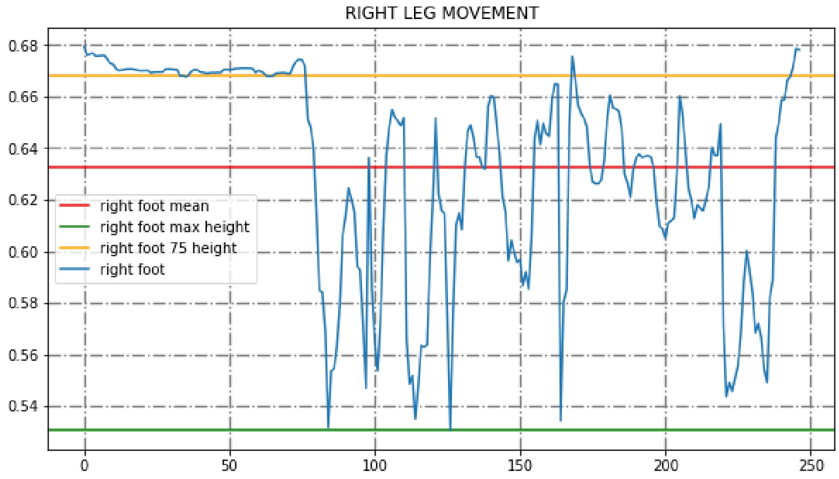


Figure 3.8: Analysing Right Leg Movement - Hop Left

3.5.2 Heuristic - Hop Right

The **Hop Right** action is defined by the criteria of hopping on the right leg throughout the video, while the left leg stays above the ground for the entire duration. The requirement once again is to count the number of such valid hops. The approach to be followed is similar to the Hop Left action in Section 3.5.1.

Algorithm 2 - HOP RIGHT HEURISTIC

```
if Left Foot between Max Height and 75% quartile height then
    if Right Hip > Mean and Right Foot above ground then
        action ← JUMP
    else
        action ← DOWN
    end if
end if
```

3.5.3 Heuristic - Skip

The Skip action is defined by the criteria of stepping forward and hopping on the same foot, and alternating this movement between feet. The objective is to count the number of skips performed by the child.

Unlike Hop discussed in 3.5.1, the Skip action is not 2-Dimensional i.e. it also involves a depth based movement forward towards the camera. MediaPipe's z coordinate for the joints were investigated for this, but did not show any definite decrease as illustrated in Figure 3.9. In comparison, the distance between the hip and the foot increases as the child moves closer to the camera, because of perspective. The videos were analysed to

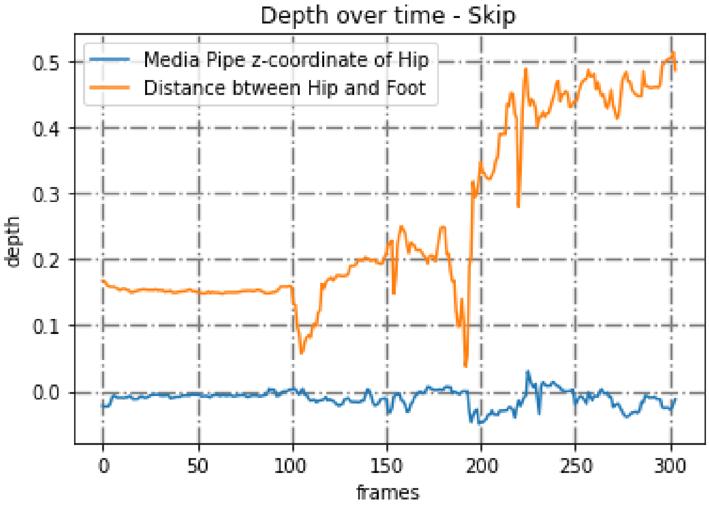


Figure 3.9: Comparing MediaPipe's depth z-coordinate with perspective

identify any learnable information from the joint data and the hip and leg landmarks were analysed for this purpose, as the movement of these contribute the most to the activity. Figure 3.10 is a plot of the movement of the legs and hip during the skip action. The Skip action can be thought of as broken up hops, but on alternating feet with forward movement. The troughs in Figure 3.10 are not as well defined as the oscillations that we see for the Hop action in Figure 3.7 and this can be attributed to the perspective shift and changing depth of the child from the camera over time. Notice that the troughs are deeper at around the 250 frame mark in this example, when the child is closer to the camera.

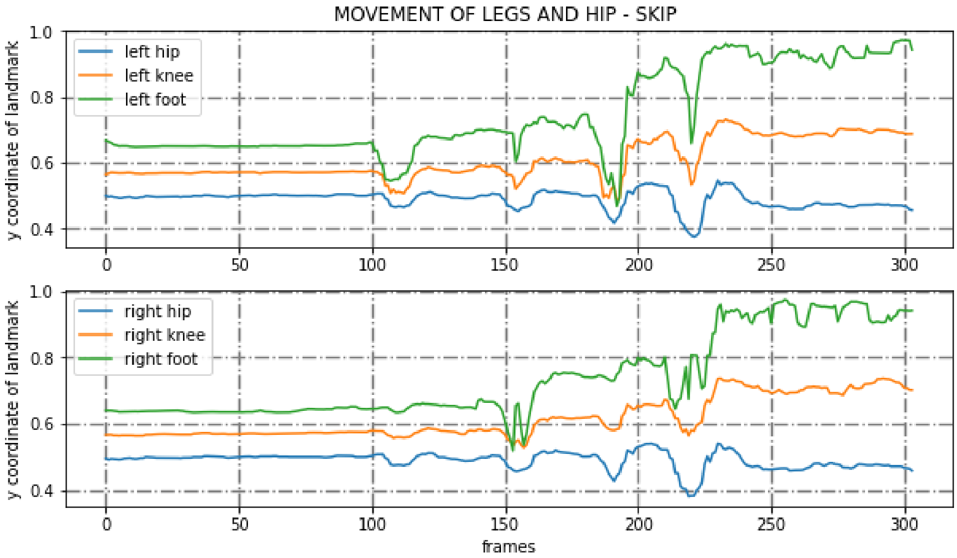


Figure 3.10: Analysing Leg + Hip Movement - Skip

The heuristic was chosen with theses aspects in mind, so as to reflect both the movement of the right and left foot. For the first skip, the starting frame of the video is used to find the initial positions of the left and right foot. Subsequently, this position is updated after the completion of every skip. As mentioned in Section 3.5.1, it is useful to take both the movement of the hip and feet into account as the hips show the most pronounced movement. Hence the starting position of the hips are tracked and updated as well, for each skip. If the current position of the feet are above the initial position by a threshold of 0.1, and if the current position of the hip is above the starting position, the frame is labelled as a Skip. This threshold was chosen after going through different videos and making an estimate of a good choice based on perspective and the rough starting distance of the children from the camera. Note that the change in perspective as the child moves towards the camera results in skips that are seemingly higher, and so this threshold of 0.1 really represents the first skip that is furthest from the camera. 3 summarizes this.

Algorithm 3 - SKIP HEURISTIC

```
if Left Foot - Left Foot Start $> 0.1$ then
    if Left Hip $>$ Left Hip Start then
        $action \leftarrow$ SKIP
    else
        $action \leftarrow$ NONE
      Update Left Hip and Left Foot Start
    end if
end if
if Right Foot - Right Foot Start $> 0.1$ then
    if Right Hip $>$ Right Hip Start then
        $action \leftarrow$ SKIP
    else
        $action \leftarrow$ NONE
      Update Right Hip and Right Hip Start
    end if
end if
```

3.5.4 Heuristic - Gallop

During the process explained in 3.4.2, it was observed that the Gallop action had the highest % of invalid frames where MediaPipe did not recognise any landmarks i.e. he videos that were classified as not useful had 70%-80% of invalid frames, which was more than the other actions. On random checking of multiple videos, the Gallop action also showed a lot of cases of improper landmark detection. The reason for why this is more prevalent in the Gallop action as compared to Hop and Skip is majorly because of the orientation of the child wrt the camera, where most of them are standing with their bodies tilted to the side

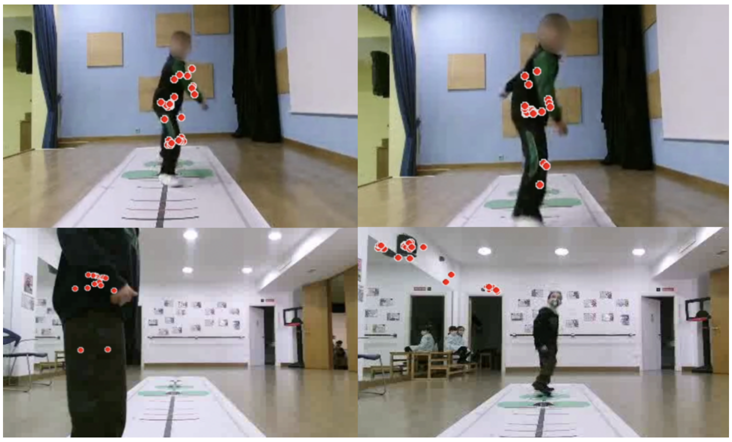


Figure 3.11: Improper Landmark Detection - Gallop

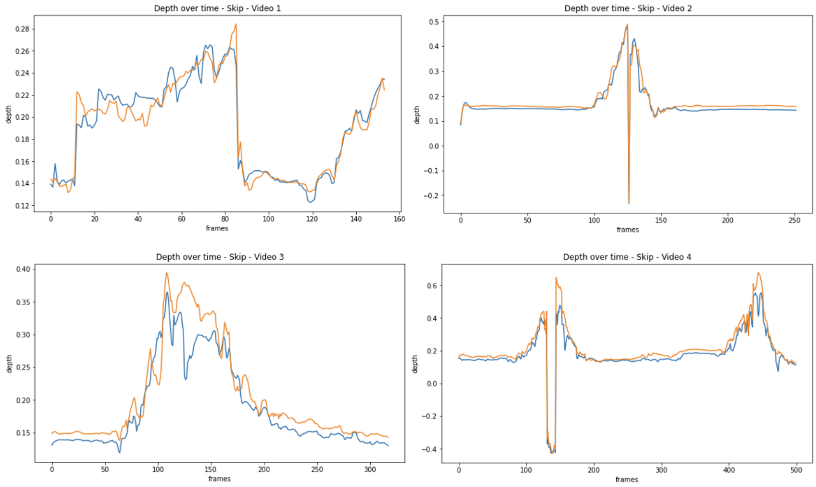


Figure 3.12: Comparing Depths for Different Videos - Gallop

A few examples are shown in Figure 3.11. The problems vary from displaced landmarks, entirely wrong ones and also half the landmarks disappearing when the child is too close to the camera. There are also other factors that add to the problem as discussed in 3.4.2 like background matching the color of the uniform/ monochrome uniforms etc. There were also an unusual number of videos where the children were either standing completely still, or performing a different action entirely - like running forward, or hopping forward. This might have been because Gallop is the most physically complex action to understand and execute, as compared to Hop and Skip.This can be further illustrated by plotting the joint coordinates over the frames of a video.

A few examples of this are shown in the figures above. Figure 3.12 shows 4 different

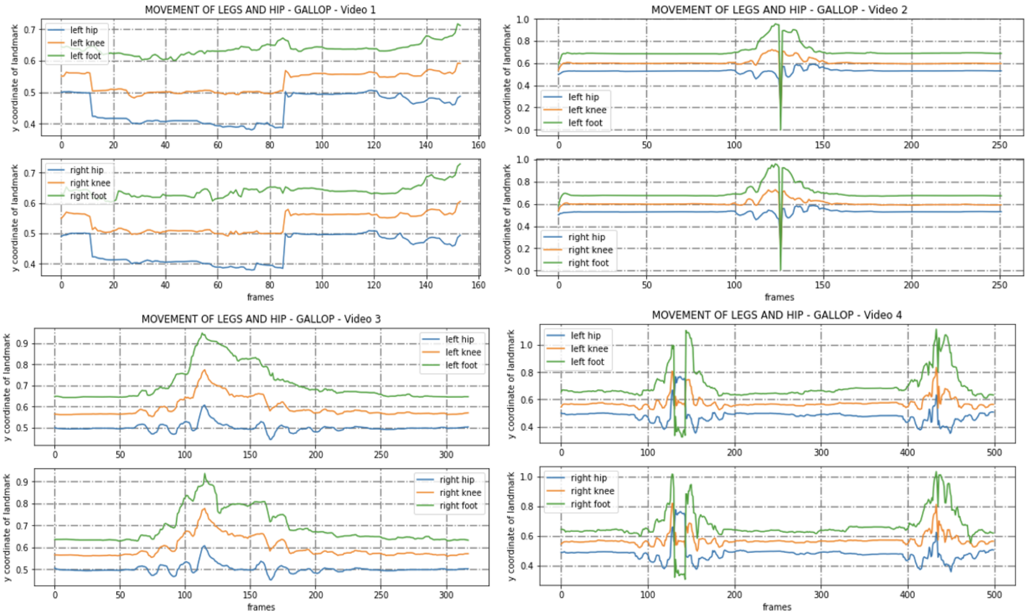


Figure 3.13: Comparing Leg + Hip Movement for Different Videos - Gallop

videos of the Gallop action and compares their depth, which is indicated by the distance between the hip and feet, which should increase as the child moves closer to the camera because of perspective. Comparing it to same kind of plot we can see that although there was a clear gradual increase for the Skip action in Figure 3.9, the Gallop action in Figure 3.12 shows varying behaviour for all 4 examples showing that there is no consistency in what the children are doing in different videos. Figure 3.13 shows the movement of the Hip and the Legs during the Gallop action for the same 4 videos, and once again we can see that there is no consistent behaviour that can be observed and used to train a model. Comparing this to the Hip and Feet Analysis performed in the previous sections for Hop in Figure 3.7 and Skip in Figure 3.10, we can see how there are learnable patterns in those plots unlike the Gallop action.

3.6 Training Models

The labelled frame data generated from the heuristics above were used to train different models for different actions with the Random Forest algorithm and a Neural Network. The data and the problem are treated as a supervised classification task, where the aim is to classify a frame of the video as one of the labels that were assigned to it by the heuristic. The specifics of the models and the hyperparameters chosen are stated below:

Random Forest:

The Random Forest algorithm is an ensemble method that uses an uncorrelated set of decision trees that can be used to perform a supervised classification task. A majority vote is used to decide the predicted class from the individual decision trees. The algorithm has a reduced chance of overfitting because it averages out or polls the result from different uncorrelated decision trees. Two of the hyperparameters that were fixed are the **max_depth** and **the number of estimators**, which were set as 5 and 100 respectively. The metrics that were taken into account are the train accuracy, test accuracy, confusion matrix and F1 score.

Neural Network:

An Artificial Neural Network (ANN) is a system of neurons that aim to mimic the functioning of the brain by means of a network with weights associated with each link. They are especially suitable for learning relationships in data that cannot be articulated by means of rules or correlations. The networks learns by adjusting the weights on the links as it learns, in an attempt to accurately predict different classes. They are also suitable for noisy data.

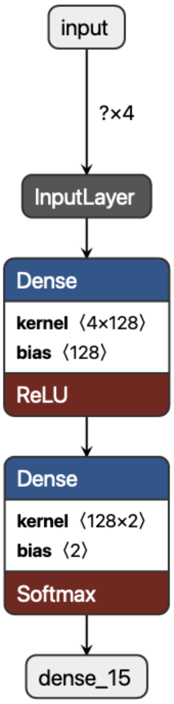


Figure 3.14: Neural Network Architecture for Classification

For the purpose of the classification task, a basic network was set up with the architecture mentioned in Figure 3.14. The **optimizer** used was Adam (a combination of

AdaGrad and RMSProp, which are improvements to stochastic gradient descent) with a Categorical Cross-Entropy **loss function** The model was trained for 100 **epochs** (or lesser based on the observed convergence of the validation accuracy) and a **batch size** of 10. The metrics used to evaluate the performance of the model are the training accuracy and validation accuracy respectively.

Chapter 4

Experiments and Results

4.1 Model Performance

This section presents the results of training the models on the labelled joint data collected from the frames of the videos as described in Section 3.5.1 and Section 3.5.3. It also describes the joint features that were chosen and any data preparation steps that were done before the training process.

4.1.1 Hop Left/Right

Some of the videos were discarded by the approach mentioned in Section 3.4.2. Since the heuristic that was developed involved an analysis of both legs and the hip, only these joints/features were chosen to train the models, and different combinations of them were tested to compare results. A train-test spilt of 90%-10% was taken for the Random Forest model and a validation split of 0.1 for the Neural Network. As described in 3.1, there are different Levels of the data, and the children gain a better understanding of the data to perform it more accurately in Level 2, as compared to Level 0 and Level 1. Hence these subsets of the data were also used to train the data separately to investigate if there was a change in the accuracy. 4.1 shows the results for Hop Left for different combinations of joints, and Levels using the Random Forest Algorithm.

Joints	**Levels**	**Train Accuracy**	**Test Accuracy**	**F1 Score**
Legs + Knees + Hips	0,1,2	0.8079	0.8038	0.8038
Legs + Hips	0,1,2	0.8081	0.8048	0.8048
Legs + Knees + Hips	2	0.8268	0.8256	0.8256
Legs + Hips	2	0.8239	0.8213	0.8213

Table 4.1: Hop Left Results - Random Forest

The classes/labels in the data were categorically encoded as [0,1] which represents the labels ['DOWN', 'HOP'] using a Label Encoder and then fed into the Neural Network. The input dimensions of the network were changed appropriately based on how many joints/ input features were selected. A validation split of 0.1 was chosen and the results of the Neural Network are shown in 4.2.

Joints	Levels	Train Accuracy	Validation Accuracy
Legs + Knees + Hips	0,1,2	0.8023	0.7839
Legs + Hips	0,1,2	0.8036	0.7930
Legs + Knees + Hips	2	0.8087	0.7949
Legs + Hips	2	0.8065	0.8104

Table 4.2: Hop Left Results - Neural Network

The best performing model was then tested on different videos to check how well they performed in real-time. MediaPipe was used once again to extract the joints from the frames in the video, and the selected features/joints were used to predict the output label for the frame with the trained models. Figure 4.1 shows these results for the Hop Left action.

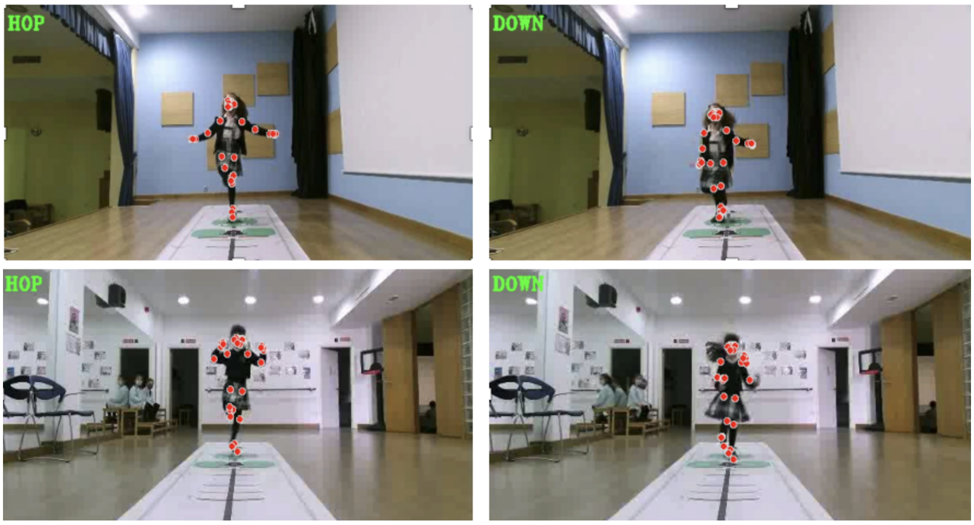


Figure 4.1: Hop Left - Real Time Classification Results

Joints	Levels	Train Accuracy	Test Accuracy	F1 Score
Legs + Knees + Hips	0,1,2	0.8088	0.8053	0.8053
Legs + Hips	0,1,2	0.8062	0.8007	0.8007
Legs + Knees + Hips	2	0.8450	0.8347	0.8347
Legs + Hips	2	0.8428	0.8366	0.8366

Table 4.3: Hop Right Results - Random Forest

Joints	Levels	Train Accuracy	Validation Accuracy
Legs + Knees + Hips	0,1,2	0.8014	0.7663
Legs + Hips	0,1,2	0.7999	0.7731
Legs + Knees + Hips	2	0.8323	0.7651
Legs + Hips	2	0.8382	0.7679

Table 4.4: Hop Right Results - Neural Network

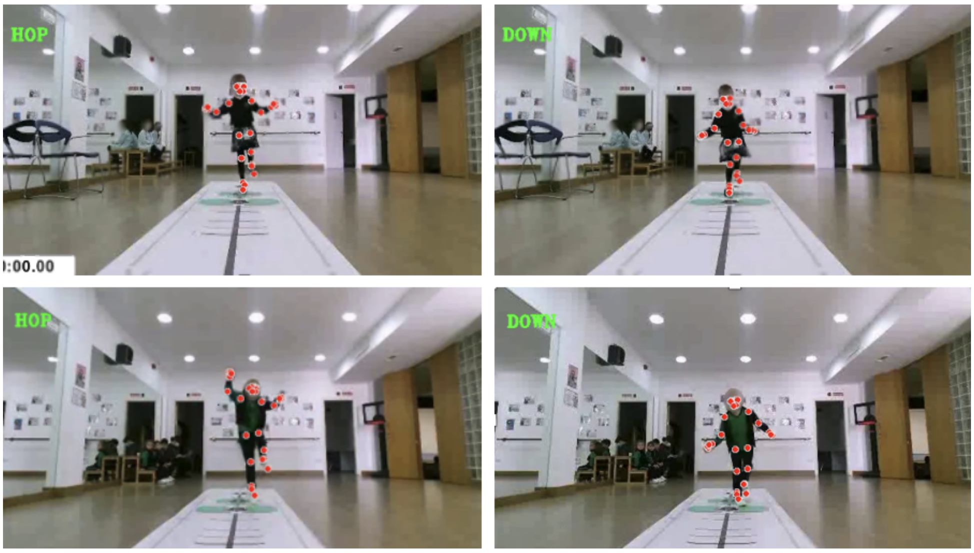


Figure 4.2: Hop Right - Real Time Classification Results

Similar experiments were carried out for the Hop Right action and the results are given in Table 4.3, Table 4.4 and Figure 4.2, which are the training results for Random Forest, Neural Network and real-time results respectively.

4.1.2 Skip

Once again, some videos were discarded based on the Section 3.4.2. Since the heuristic for Skip was developed by analysing the Legs and the Hips, only these features/joints were included for the training process. Different combinations of joints and levels were used for the training process to compare the results. A train-test split of 90%-10% was taken and 4.5 shows the results of the Random Forest algorithm.

Joints	Levels	Train Accuracy	Test Accuracy	F1 Score
Legs + Knees + Hips	0,1,2	0.8826	0.8817	0.8817
Legs + Hips	0,1,2	0.8818	0.8812	0.8812
Legs + Knees + Hips	2	0.8838	0.8856	0.8856
Legs + Hips	2	0.8833	0.8843	0.8843

Table 4.5: Skip Results - Random Forest

The classes/labels in the data were encoded categorically as [0,1] representing the labels ['NONE','SKIP']. A validation split of 0.1 was taken and the results of the Neural Network are shown in Table 4.6.

Joints	Levels	Train Accuracy	Validation Accuracy
Legs + Knees + Hips	0,1,2	0.8664	0.9283
Legs + Hips	0,1,2	0.8649	0.9269
Legs + Knees + Hips	2	0.8579	0.9012
Legs + Hips	2	0.8609	0.9091

Table 4.6: Skip Results - Neural Network

The best performing model was tested on the videos to check classification of frames in real-time. Some snippets of the results are shown in Figure 4.3.

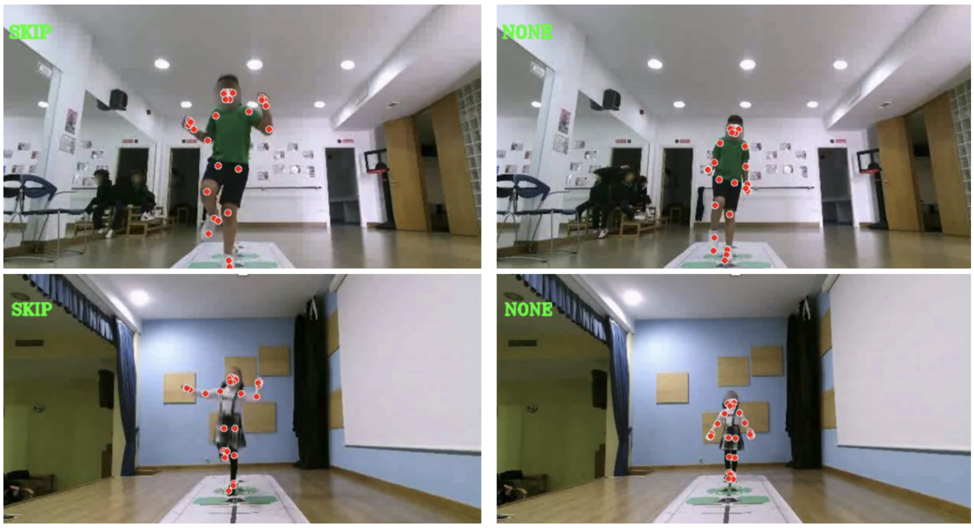


Figure 4.3: Skip - Real Time Classification Results

4.2 Discussion

4.2.1 Hop Left/Right

We can observe from Tables 4.1, 4.2, 4.3, 4.4 that the models that were trained on just the Level 2 data show a higher accuracy than the models that were trained on Level 0,1,2 data. This could indicate that the models learn better from the videos in Level 2, because the children were more accustomed to the actions at that stage and hence performed them better. The Random Forest and Neural Network models both showed comparable performance. The model accuracies reported above are based on frame-by-frame classification of the labelled data. To test how well the model performs on the full video i.e. how well it can count the number of hops, a total of 10 videos were picked randomly from the Hop Left and Hop Right data, with a mixture of improper and proper hops. The result derived from the model predictions for these data were then checked against manually derived results based on human observation. The model was able to accurately count the hops in 7/10 and 8/10 of the videos in the Hop Left and Hop Right action respectively. Even for the videos where the model did not count the exact number of Hops correctly, it was off by only a few numbers i.e. was able to identify some hops. It was also observed that the model seems to do better at checking the validity based on whether the leg that is not performing the hop stays above the ground. On the other hand, the model does well for checking if the leg that is hopping leaves the ground only if the child jumps by a detectable height. This is because the children are standing at quite a distance from the camera and because of perspective, even a small movement upward can seem like a hop, and alternatively, the height of the hop as perceived by the camera does not fully reflect the actual height that the child hops by. It was also observed that in general, the model performs better at counting hops for the older children as compared to the younger. The reason for this could be partly because there is lesser data of the younger children, and within this data there are a lot of outliers where the children are not really performing the action well.

4.2.2 Skip

The results for the Skip action were observed to be higher than Hop Left and Right, as can be seen in Tables 4.5 and 4.6. This was surprising, because based on the data and existing research mentioned in Section 2.1, actions that involve depth and movement towards the camera should be harder to perform HAR as compared to activities that move in 2-Dimensions like Hop Left/Right. To further investigate validity of these results, 10 videos of the Skip action were randomly chosen to check how well the model can count the number of skips in the videos. It was observed that the model was able to accurately count the number of Skips only in 4/10 of the videos. This is a direct indication of the fact that the frame-by-frame accuracy of the models are heavily dependent on the heuristic that was defined for the actions. The thresholds chosen for the heuristic to identify a skip were set by trial and error, on observing different videos. But with a dataset this large, it is difficult to accurately find a threshold that represents the full data. Another reason could be that the position of the camera is not the same from video-to-video. Hence even if the children are standing at the same starting location, it is difficult to determine a threshold that works across all videos. Hence, the frame-by-frame accuracy does not translate well into individual video based accuracy.

Chapter 5

Conclusion & Future Work

5.1 Summary

This projected investigated a joint-based approach to perform action recognition and identify improper actions for a dataset of children performing the actions - Hop Left, Hop Right, Skip and Gallop. The method involved the extraction of body landmarks/joints from the video data, followed by labelling individual frames by means of different heuristics. The Random Forest algorithm and a Neural Network model were trained for the supervised classification task of correctly identifying the labels of the frame data. The performance of the models were investigated by frame-by-frame accuracy with model metrics and video accuracy by testing them on randomly selected videos from the dataset. The Hop Right/Left actions showed the best performance amongst all the actions.

During the course of the project, different observations about the data and its effect on the model training and results were observed. One of the main issues with the data is that it was not labelled or annotated, much like a lot of real-world raw data that is collected from a source/ sensor/ camera. This is especially challenging because a lot of the SOTA models and research in HAR (observed in the Literature Survey) are trained on huge annotated datasets of adults and even they seek to perform action recognition, i.e. distinguishing between different actions and not really anomaly detection or ”improper” action detection. The data was also not divided into ”valid” and ”invalid” data - in terms of the criteria that define different actions. Even though the end goal is to analyse the motor-skill development of the children, the ambiguity of the data results in the question of whether there is enough data with ”proper” actions to train a model to learn from them. Even in the paper by Suzuki et al. (2021) as mentioned in Section 2.5.1, the data was divided as ”normal” and ”abnormal” before the training process. The next section will discuss how this project could be carried forward, considering these observations and challenges.

5.2 Future Work - Possible Improvements

A possible next step would be completely dividing and annotating the data into ”valid” and ”invalid” actions, so that the models can better learn the differences between the two or better learn the action from the valid data. This would need to be done manually, which is time consuming but necessary because of the nature of the data. A second step would be manually annotating the frames of the valid and invalid videos, which would remove issues that a generalised heuristic can pose. In the process, it would also be useful to manually clean out the data more, and identify more videos that are not useful for the model training process.

MediaPipe performed well for the Skip and Hop Left/Right actions, but was at a disadvantage in the Gallop action because of the orientation of the body wrt the camera. A future step could be to try out other more powerful libraries like OpenPose, PoseNet etc with GPU support. This would be computationally more intensive than MediaPipe, but possibly more accurate. There are also improvements that could be made based on the development of the heuristic for the approach mentioned in this project. For instance, one idea would be to choose different thresholds of hop/skip height based on the age/ bodily measurements of the children. Thresholds that are scaled based on distance and perspective from the camera might also work for actions like Skip and Gallop, where the child is moving towards the camera.

Since the movement of the landmarks can be treated as time-series data, Binary Segmentation could be explored to find regions of interest within them i.e. regions where there is movement or an action taking place. This might save the time and effort that goes into manually cropping the videos. Further, LSTMs and CNNs could be used if the data was all cropped to be the same input size to feed the models. Auto-encoders have been explored by Suzuki et al. (2021) to use a encoder-decoder network to learn to identify faults, and this along with data preparation and augmentation could yield better results of classification.

Bibliography

Banjarey, K., Sahu, S. P., and Dewangan, D. K. (2021). A survey on human activity recognition using sensors and deep learning methods. In *2021 5th international conference on computing methodologies and communication (ICCMC)*, pages 1610–1617. IEEE.

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., and Grundmann, M. (2020). Blazepose: On-device real-time body pose tracking.

Bossavit, B. and Arnedillo-Sánchez, I. (2019). A novel approach to monitor loco-motor skills in children: A pilot study. In *European conference on technology enhanced learning*, pages 773–776. Springer.

Bossavit, B. and Arnedillo-Sánchez, I. (2020). Designing digital activities to screen locomotor skills in developing children. In *European Conference on Technology Enhanced Learning*, pages 416–420. Springer.

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299.

COCO (2020). Coco 2020 keypoint detection task - `https://cocodataset.org/#keypoints-2020`.

Das Antar, A., Ahmed, M., and Ahad, M. A. R. (2019). Challenges in sensor-based human activity recognition and a comparative analysis of benchmark datasets: A review. In *2019 Joint 8th International Conference on Informatics, Electronics Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, pages 134–139.

Dutta, T. (2012). Evaluation of the kinect™ sensor for 3-d kinematic measurement in the workplace. *Applied ergonomics*, 43(4):645–649.

Galna, B., Barry, G., Jackson, D., Mhiripiri, D., Olivier, P., and Rochester, L. (2014). Accuracy of the microsoft kinect sensor for measuring movement in people with parkinson's disease. *Gait & posture*, 39(4):1062–1068.

Gatt, T., Seychell, D., and Dingli, A. (2019). Detecting human abnormal behaviour through a video generated model. In *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 264–270. IEEE.

GOOGLE (2020). Mediapipe pose - `https://google.github.io/mediapipe/solutions/pose.html`.

Gupta, A., Gupta, K., Gupta, K., and Gupta, K. (2020). A survey on human activity recognition and classification. In *2020 international conference on communication and signal processing (ICCSP)*, pages 0915–0919. IEEE.

Ivan Grishchenko, V. B. (2020). On-device, real-time body pose tracking with mediapipe blazepose - `https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html`.

Jia, Y. (2009). Diatetic and exercise therapy against diabetes mellitus. In *2009 Second International Conference on Intelligent Networks and Intelligent Systems*, pages 693–696. IEEE.

Jiang, W. and Yin, Z. (2015). Human activity recognition using wearable sensors by deep convolutional neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1307–1310.

Lara, O. D. and Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209.

Liu, T., Song, Y., Gu, Y., and Li, A. (2013). Human action recognition based on depth images from microsoft kinect. In *2013 Fourth Global Congress on Intelligent Systems*, pages 200–204. IEEE.

Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., and Murphy, K. (2017). Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4903–4911.

Perez, A. J., Labrador, M. A., and Barbeau, S. J. (2010). G-sense: a scalable architecture for global sensing and monitoring. *IEEE Network*, 24(4):57–64.

Prieto, L., Columna, L., and Hodge, S. R. (2019). Evaluating skill development: Tgmd-3. In *Case Studies in Adapted Physical Education*, pages 41–46. Routledge.

Ronao, C. A. and Cho, S.-B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244.

Samir, M. A., Mohamed, Z. A., Hussein, M. A. A., and Atia, A. (2021). Applying deep learning to track food consumption and human activity for non-intrusive blood glucose monitoring. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 0319–0324.

Sathyanarayana, A., Ofli, F., Fernandez-Luque, L., Srivastava, J., Elmagarmid, A., Arora, T., and Taheri, S. (2016). Robust automated human activity recognition and its application to sleep research. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 495–502.

Suzuki, S., Amemiya, Y., and Sato, M. (2020). Enhancement of child gross-motor action recognition by motional time-series images conversion. In *2020 IEEE/SICE International Symposium on System Integration (SII)*, pages 225–230. IEEE.

Suzuki, S., Amemiya, Y., and Sato, M. (2021). Skeleton-based visualization of poor body movements in a child's gross-motor assessment using convolutional auto-encoder. In *2021 IEEE International Conference on Mechatronics (ICM)*, pages 1–6. IEEE.

Yang, W., Lyons, T., Ni, H., Schmid, C., and Jin, L. (2022). Developing the path signature methodology and its application to landmark-based human action recognition. In *Stochastic Analysis, Filtering, and Stochastic Optimization*, pages 431–464. Springer.

Yin, J., Yang, Q., and Pan, J. J. (2008). Sensor-based abnormal human-activity detection. *IEEE transactions on knowledge and data engineering*, 20(8):1082–1090.

Zeng, M., Nguyen, L. T., Yu, B., Mengshoel, O. J., Zhu, J., Wu, P., and Zhang, J. (2014). Convolutional neural networks for human activity recognition using mobile sensors. In *6th international conference on mobile computing, applications and services*, pages 197–205. IEEE.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., and Zheng, N. (2019). View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1963–1978.