Human Action Recognition: An approach to assess the loco-motor skills in children

Azin Makaranth, B.Tech

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: Prof. Inmaculada Arnedillo-Sánchez

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.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have also completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

Azin Makaranth

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.

Azin Makaranth

August 19, 2022

Human Action Recognition: An approach to assess the loco-motor skills in children

Azin Makaranth, Master of Science in Computer Science
University of Dublin, Trinity College, 2022

Supervisor: Prof. Inmaculada Arnedillo-Sánchez

Human action recognition is a prominent interdisciplinary field of research in the area of Computer Science, Machine Learning, and Artificial Intelligence due to its the potential revolutionary applications and the prolonged list of challenges. The focus of this dissertation is to apply Human action recognition to assess the loco-motor skills of children. This is a challenging problem, especially because of the unpredictable nature of the children, which exacerbated the inherent nature of humans to execute the same action in a plethora of different ways. In addition, there are also the classical challenges to the problem, including the variations in the camera view point, background cluttering and the quality issues. The dissertation starts with a brief review of the state-of-the-art literature on Human Action Recognition. Based on the literature reviewed, an overview of the tasks are presented, followed by a brief discussion of the technologies used to achieve the feature extractions and models used to perform the action recognition. Further, the paper focuses on the methodology employed to achieve the specific task of action recognition from the videos of children. The set of actions focused on the dissertation are specifically designed to assess the loco-motor skill of the children, and the final outcome of the work will deduce the number of times a specific action was perform. The feature extraction was performed using the MediaPipe library and custom heuristics were defined to label the extracted features. The action recognition was performed using Deep Neural Network, Random Forest Classifier and LSTM models. The models showcased around 80% accuracy based on the heuristics defined.

Acknowledgments

The journey of this dissertation from scratch to its final draft has been a very strenuous one. Amidst the difficulties there are a few people to whom I am extremely grateful to.

To begin with, I would like to thank the universe for creating the opportunity to reach where I am now. Thereafter I am much obliged to my most lovable parents, Geethamma and Synu, it is their immense blessings and prayer that has made me complete this work. They have been the strongest pillar that supported me through the ups and downs in my life. I would also like to thank my brother Amin who has been a great motivation during my hardships. I am grateful to Sneha for taking good care of me during the turmoil.

I also take this opportunity to thank my roommates, Akhil, JP, John, Shybu, Tom and Unni for helping me out when I was stuck. A special shout-out to my teammates which includes Arun, Manu, Sherin and Tom. They were always there filling in new ideas when it was most needed.

Lastly and most importantly I would like to thank Prof. Immaculada, our supervisor, for being the perpetual support and Dr. Benoit Bossavit for guiding and providing technical support. I also thank all my family, friends, and well-wishers for their blessings. Thank You.

Azin Makaranth

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Our interactions with modern machines and computers are rapidly evolving. Digital computers of today's era have the capability to identify and comprehend human gestures and actions. A human action recognition (HAR) system which aims to identify human behaviour in a particular scenario has become increasingly popular in the field of computer vision due to its immense potential to find applications in various fields to assist and ease human endeavours. These systems are crucial for a number of real-world applications ranging from autonomous navigation systems, for enabling the safe operating environment for autonomous vehicles [28], visual surveillance systems[26] for identification of potential human threats and suspicious human activities, human-machine interactions [33] enabling the humans to communicate with the robots and for entertainment. The HAR systems have gained increasing popularity in the last couple of decades due to the potential of their applications. With the growth of processing power, advancements in camera technologies, and improvements in battery efficiency, computer vision has witnessed a lot of progress in the last decade. The field of computer science is witnessing a race for advancements in vision-based recognition systems as a substrate of the unquenching thirst for progress in Artificial Intelligence Technology. The vision-based systems identify or infer and classify the human actions into predefined categories [23]. Even though there exists a number of systems for this purpose, human activity recognition is still a challenge for the computers, and the outcome of such systems are affected by a number of factors depending on the sensors and the algorithms. For instance, for a camera based human action recognition system, the background, the quality of the input or the frames, number of human subjects, the view-point or the perspective, number of human subjects in the frame, etc poses a number of challenges to the efficacy of HAR systems. Hence, there exists a behemoth of opportunities yet to be unfolded with several applications that ameliorate human life in multiple fields. However, the plethora of challenges and roadblocks limits the efficiency of such systems.

As discussed, the HAR systems find its application in many fields of human life. However, studies seldom focus on the use of such systems to monitor the loco-motor skill development of children. The loco-motor development is the process by which a child learns the ability to create simple and complex movement patterns by developing the associated muscles [16]. Evidence suggests that the loco-motor skills development is highly cohesive with the cognitive development of children [19]. Traditionally, the assessment of the loco-motor skill development is performed by trained professionals using standard tests, and it can be hampered by the availability and cost of these facilities. A vision based HAR system can be used to infer the action performed by a child and determine whether the child has performed the action as expected, in par with the particular developmental age group that the child belongs. Such a system is cost-effective, autonomous, interactive and highly available. It can be extended to develop children-friendly games which are much more appealing and less cumbersome for children than the traditional standard tests such as Movement-ABC2 [10]. There exists many voids in the field which can be filled with the application of HAR and this dissertation aims to fill one such gap.

1.2 Problem Definition

As discussed, the loco-motor skill development of a child plays a vital role in the physical and psychological evolution of a child. Any hindrance in affecting the loco-motor skill development can significantly reduce and leave a major impact on the later life of a child. There is an urgent need to determine any problems that impact the loco-motor skills of a child, as the earliest, an impairment is detected, the quicker the remedial measures can be provided, to improve the quality of life of that child. However, with the current technology, the early assessment of the loco-motor skills of a child is a time-consuming, and costly process which mandates the need for a trained professional. This does not guarantee that the service reaches every child as all the children are not living in the same conditions with equal opportunities, and there exist many socio-economic inequalities in the society. For instance,1) a parent may not be aware of the importance of loco-motor skill development, 2) may not have the economic capacity to afford a costly appointment with a medical practitioner, 3) Does not have enough trained professionals to do the assessments, etc might be the reasons.

This study aims to create a solution to assess the loco-motor skills of children with the aid of Human Action Recognition (HAR) using which we can identify the actions

Actions	**Criteria**
Push Up	Count the number of push-up; The child is lying on the ground facing down on the hands. The child lowers the trunk until (almost) reaching the ground and rises again the trunk
Sit Up	Count the numbers of sit-up; The child is sitting on the ground. The child lowers the trunk until (almost) reaching the ground (face up)and rises the trunk again.
Kneel	Count the number of times the child turns leftwards and rightwards while kneeling down.
Imitate	Imitate the pose displayed to the child. One of the 7 poses will be displayed to the child.

Table 1.1: Actions and Criteria

performed by children. This project is an extension of the existing work conducted by Bossavit and Arnedillo-Sanchez where they developed a novel approach to monitor loco-motor skills in children using motion detection gaming technologies [8]. Bossavit and Arnedillo-Sanchez have identified and defined a set of actions to be performed by the children for the assessment of their loco-motor skills. The children were instructed to perform the predefined actions, and their performance has been monitored using sensors and cameras. A dataset with the videos of children performing the predefined actions was created. The objective of this dissertation is to create a machine learning model which can identify a subset of actions defined by Bossavit and Arnedillo-Sanchez which evaluates the fundamental motor skills of children, and then use the generated model to quantify the actions performed by the children. This dissertation primarily focuses on the actions such as 1) push up, 2) sit up, 3) Imitate and 4) Kneel. In this project, the aforementioned actions are governed by stringent criteria. The proposed solution will identify the actions performed by the children in accordance with these predefined criteria and emits the number of times the action was performed. This data can be used to infer, whether the child has performed the actions as expected from the developmental age group that the child belongs. The table 1.1 presents the criteria that defines each of the actions.

1.3 Contributions

According to Gandotra et al. children with Autism Spectrum Disorder (ASD) exhibited a high degree of impairments in fundamental movement skills such as object control and loco-motor skills compared to normal children in the same age group [17]. The study also proves that such impairments in motor skills are evident in the early stages of development of a child. The Centre for Disease Control (CDC) reported a 1.68% prevalence rate for

ASD in the US for the surveillance year of 2014 and a rate of 2.6% in South Korea [14]. It is also worth noting that, the National Council for Special Education on Supporting Students with Autism Spectrum Disorders in Schools reports that 1 in 65 of the school going population of Ireland has been diagnosed with ASD. These reported values indicate the importance of the early detection of the impairments to motor skills of children, and there is an urgent need for a systematic approach to find a solution to the detection of such impairments. With the help of a system for assessing the loco-motor skills of the children, the authorities, parents, or caregivers of the children can determine the loco-motor skill disorders. As an initial step to the solution to this problem of identifying the debilitation in motor skills, this work aims to identify the actions such as push up, sit up, imitate and kneel using machine learning techniques. As discussed in the previous section, this work contributes to the research [8], conducted by Bossavit, Benoit and Arnedillo-Sánchez, Inmaculada (2019). The major contribution of the study is a new approach to action recognition designed specifically for children with actions specialized to assess their loco-motor skills.

1.4 Dissertation Layout

The ensuing dissertation is structured into chapters as follows:

1. **Chapter 2** : A review of the state-of-the-art research related to the problem, starting from a quick overview to a brief review of the techniques involved.

2. **Chapter 3** : Details the methodology used to solve problem detailed in the dissertation including the challenges faced.

3. **Chapter 4** : The experiments conducted and the results obtained are detailed.

4. **Chapter 5** : Concludes the dissertation with the reflections and a discussion on the possible future works for the dissertation.

Chapter 2

State of the Art

As Human Action Recognition is a promising field with a lot of potential applications, there are a lot of research happening in this field. A brief review on the state-of-the-art in the field of Human Action Recognition Technology is presented in the ensuing section.

2.1 Background

The vision is a very important sensory perception which enables the humans to comprehend the world around us. The ability of sight and the capability to process the vision is one of the characteristics that differentiate the humans from the computers. But, what if the computers has the ability to perceive and comprehend the world? What if the computers of futures are able to identify the human gestures and actions ? Computer vision is the field of computer science, which has the potential to such a future, where the computers can mimic the visual comprehension ability of humans. There are a lot of research happening in the field of computer vision. The vision-based Human Action Recognition (HAR) is an interdisciplinary subject in the field of computer vision and machine learning. It enables the computers to automatically identify the actions performed by a human subject in the videos. Even though there are numerous research in this field, the HAR is still a challenging field of research. Machine learning algorithms can detect the patterns in any forms of data. However, the most major issue in HAR is that there are a number of different ways to do the same action. Even the same individual will be performing the same action uniquely in every attempt. In addition, the actions performed by individuals of different age groups are vastly different. There are other challenges in this field such as the variations in the camera view point, noise, compression quality of the video, processing power, occlusions, variations in the actions and the appearance of the human subjects, etc. In the context of HAR, the terms ‘actions’ and ‘activities’ are used interchangeably. In the ensuing discussion, it is necessary to define the terms. Turaga et al. define the terms as follows: [37]

Actions : An action constitutes simple loco-motion patterns, typically performed by a single person for a short duration of time. Examples are kneeling, walking, sit-up, push-up, swimming, etc.

Activities : In contract to actions, the activities are a complex sequence of actions performed by several individuals who would be interacting with each other in a constrained manner for a longer duration of time. Examples are a group of chefs cooking a meal, tennis players playing tennis, two persons shaking hands, etc.

In this project, the focus is on the recognition of the actions rather than the activities. Generally the human action recognition is composed of two main delegates which are the representation of the human actions and the classification [30]. In the representation step, the features encapsulating the actions are extracted and are converted to input feature vectors. In the classification stage, the input feature vectors are used to identify and label the actions. The overview of the general approach for the recognition of the actions is as follows [37]:

1. Extraction frames from the videos;
2. Extraction of low-level features from the frames;
3. Mid-level action descriptions from the extracted low-level feature;
4. High-level semantic interpretation of the actions from the mid-level action descriptions.

The first step in the process of Human action recognition is the extraction of frames from the input videos. OpenCV is an open-source python library for computer vision applications. OpenCV is widely used to process the videos and especially to extract the videos into frames.

In the next step, the low-level features are extracted from the videos. The videos contain a lot of raw information in detail, in addition to the human subject who is performing the action. However, most of these information are irrelevant for the task of identifying the actions. According to a classic experiment conducted by Johansson, humans have the ability to identify the patterns of actions such as walking, using the point light sources placed in a few of limb joints [22]. Hence, for the identification of actions from the videos, it is not necessary to extract the information regarding the colour of the cloths, lighting conditions, skin complexity, etc. The following are the two general approaches for the representation of the features :

Global feature representation methods: In this approach, the entire human body is detected and are represented using the methods such as background clipping, human contour silhouette, optical flow, etc and extract the features from the detected area of interest [41, 35].

Local feature representation methods: The local feature representation methods detect the parts of the human body in motion to perform an action. In contrast to the global representation methods, they extract the regions of interest in the human body. These methods are commonly used in the image retrieval, target identification, lip sync recognition, feature matching and video analysis [27].

After the extraction of the features, intermediate action descriptors or tags are appended to the extracted features to facilitate the learning process. In the final stage, machine learning techniques are employed to interpret the high level actions from the intermediate descriptors.

2.2 Review of Feature Extraction Approaches

As the feature extraction process is a very important part of any human action recognition process, there are a number of different techniques and libraries to ease the process. Feature detection is the first step where the features to be extracted are identified. There are a lot of advancements in the feature detection specific to the Human action recognition, there is a special field of research renowned as Human Pose Estimation, for estimating the human pose from the videos. In the ensuing section, different techniques for human pose estimation are explored :

1. **Pictorial structure framework(PSF)**: In PSF a two layered non-linear random forests are employed as the joint regressors [13]. The first layer act as the discriminator, which models the likelihood of the existence of certain parts of the human body at a particular location. The second layer is the prior which imbibes the output from the discriminator for modelling the probability distribution over the entire pose. In essence, the PSF models the human body as a set of co-ordinates for each part of the human body.

2. **Histogram Oriented Gaussian (HOG/HOF)** : The PSF fails to perform well, when the limbs are hidden or not visible from certain angles. HOG/HOF was introduced to address these problems of the PSF and has the ability to be immune to the background clutter, appearance, and occlusions [32]. HOG approach relies on computing the histograms of the spatial gradients and optical flow. Harris and

Forstner interest point operators are used to detect the spatio-temporal events in the videos [24].

3. **Deep Learning based approaches**: Generalization ability of the Deep learning based approaches, finds their applications in computer vision applications. Deep convolutional neural networks (CNN) are widely used for Human Pose Estimations, as they are notoriously famous for their capability to detect patterns and extract features from input images. Unlike the approaches like PSF and HOG, CNN can ingest complex feature when provided with large amount of data. Toshev et al. in 2014 proposed a Deep Neural Network based regression approach called 'DeepPose' which resulted in high precision human pose estimation [36]. Inspired from this research, Wang et al. in the year 2015 introduced a new representation for the features called trajectory-pooled deep-convolutional descriptor (TDD) which was an amalgamation of handcrafted features as well as deep learned features [39]. Similarly, in the same year 2015, Donahue et al. later used Long-term Recurrent Convolutional Networks (LRCN) for human pose estimation [15]. The technology has gone further down the line and in the year 2020 a generic Temporal Pyramid Network (TPN) which can integrate both 2D and 3D data at the feature level was introduced by Yang et al. [40].

Compared to PSF and HOG/HOF approaches which are the classical-handcrafted solutions with limitations in terms of accuracy and applications, the DNN based approaches are in great demand, owing to their ability to generalize. However, the DNN struggles to perform well when the input video contains multiple human subjects. These issues are addressed in the state-of-the-art, ready-to-use human pose estimation models such as the Mediapipe, OpenPose, etc. The succeeding section presents a brief review of these models.

2.3 Review on Deep Learning Models

MediaPipe

MediaPipe [29] is an open-source framework presented by Google to assist the development of cross-platform machine learning solutions, providing ready-to-consume pipelines to perform analysis over arbitrary sensory data such as audio and video streams. It enables the rapid prototyping of applications featuring, object-detection and localization, landmark detection, etc. The developers can prototype a pipeline as a directed graph of independent and maintainable components called Calculators which are connected by data streams. These pipelines can be redefined by adding additional components. MediaPipe supports the executions on GPU reducing the rendering and execution time for complex pipelines. MediaPipe offers solutions for Face Detection and Segmentation, Hair Segmentation, Object Detection, Box Tracking, Hand Detection, Instant Motion Detection, Objectron, Human Pose Estimation, etc.

The Human Pose Estimation is performed using Blaze Pose Detector [6], which has a light weight convolutional neural network capable of detecting 33 landmark points on the human body. The 33 key-point topology is shown in the figure 2.1 encapsulating the BlazeFace[5], BlazePalm[4] and Coco[25] topology renowned for Human feature representations. The BlazePose uses a combination of heatmap, offset and regression approaches to predict the 33 landmark points by stacking an encoder-decoder heatmap-based network on top of a regression encoder network. The encoder-decoder predict the heatmaps for all the joints and the regression encoder regresses directly to the co-ordinates of the joints. During the inference, stage the encoder-decoder can be occluded reducing the inference time significantly. This detector is capable of addressing the occlusion problem by training with stimulated occlusion on a per-point visibility classifier, which indicates when a part is occluded.

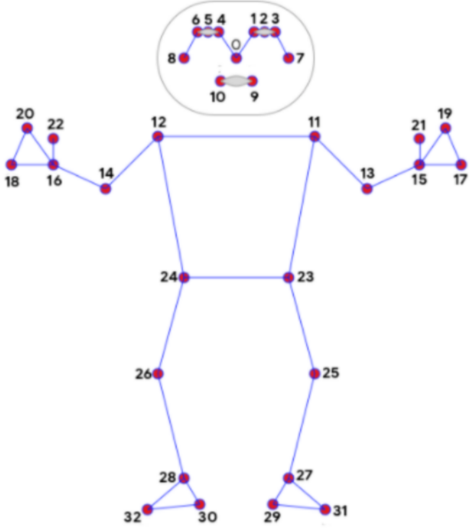


Figure 2.1: 33 Keypoints of BlazePose Topology [6]

OpenPose

The OpenPose was proposed by Zhe Cao et al [11]. in the year 2019 as a real-time approach to determine 2D poses of multiple people in an image. It follows a bottom-up approach where it detects the human body parts and then estimates the pose from the image. This approach uses a set of 2D vector fields which encodes the location and orientation of the limbs as the representation of the features. The system consists of a multistage CNN, where the initial stage predicts the Confidence map of the body part locations and the latter stage predicts the Part Affinity Fields which encodes the degree of association between the parts.

2.4 Review on Models for Action Classification

This section presents various state-of-the-art models for the classification of the actions. Deep Neural Networks, Random Forest Classifier and LSTM are taken for the high level interpretation of the actions from the intermediate labels.

Deep Neural Networks

McCulloch and Pitts in 1943 published ‘A Logical Calculus of the Ideas Immanent in Nervous Activity’ [31] which sought to understand the inner working of the human brain and their work was extended by Rosenblatt in the year 1958 creating Perceptron [34], a predecessor to the Artificial neurons. Artificial Neural Network are inspired by the biological neurons in the human brain, stimulating the interactions between them. The terms Neural networks and Deep Neural networks are frequently used interchangeably, however they both constitute two different networks. Deep Neural Networks (DNN) employs multiple layers of Artificial Neural Network (ANN) with two or more hidden layers, with each layer composing of multiple node. DNN typically consists of three or more layers-inclusive of the input and output layers. Each node in the network, or an artificial neuron, is connected to another neuron and are associated with a weight and threshold. A particular neuron is activated if the input signal to that neuron is above the associated threshold value, and transmit the signal to the neuron in the next layer. Similar, to the biological neurons, the input to the network creates a peculiar pathway to the output layer. During the training process, the network learns this pathway from the input layer to the output layer. By increasing the number of layers of the DNN, it can accomplish various intricate tasks such as Natural Language Processing, Fraud Detection, etc.

Random Forest Classifier

Random Forest Classifier is an ensemble learning algorithm trademarked by Leo Breiman and Adele Cutler, which ensembles the output from multiple decision trees which has the capability to deliver both classification and regression solutions. A decision tree is a hierarchical tree with multiple layers of decision node which eventually leads to a final decision, denoted by the leaf node. A decision tree attempts to split the data into multiple subsets based on the value of a particular attribute. During the training process, the algorithm builds this tree that best splits the data using metrics such as Gini impurity, Information Gain and Mean Square Error. The ensemble learning methods are the amalgamation of multiple classifiers producing a result by combining the results from them. One of the popular ensemble learning method is Bagging which was introduced by Breiman in the year 1996, where a random sample from the training dataset is selected with replacement [9]. Random forest is an extension to the bagging method which reduces the number of feature splits, by selecting a only a subset of the features. Random forest relays on randomness and reduces the overfitting by attempting to fit all the samples in the training set. However, they have more time and space complexity.

Long-Short Term Memory Networks

Feed-forward neural networks are not equipped to perform well with time-series data due to the inexistence of memory. Recurrent nueral networks (RNN) overcome this issue by introducing feedback loops, which stimulates the ability to retain the information for a very short time. With the addition of feedback loops, a RNN can be considered as multiple copies of the same network. However, when it requires learning from 'long-term dependencies' the RNN struggles. Long-Short Term Memory networks (LSTM) are special cases of RNN with ability to learn long-term dependencies, and was initially introduced by Hochreiter and Schmidhuber (1997) [20]. LSTM contains a memory cell known as 'cell state' which maintains the states over the time. The information can be added or regulated into the cell state by the use of gates which are composed of pointwise multiplication operation and a sigmoid neural network layer. LSTM networks are widely used in the language modelling, speech recognition, etc.

2.5 Related Work

This section presents the existing research related to the work addressed in this dissertation. There are substantial amount of research in the field of Human Action Recognition on Adults. However, there exits only a few research on Human Action Recognition which focuses on children. Turarova et al. have presented an on going research, where they employ a two camera system to capture the RGB and RGB-D data, then extracted the features using OpenPose and OpenNI tools [38]. However, this research is not focusing on how to recognize the actions. Huang et al. proposed a human action recognition system for elder and childcare using 3D convolutional network as the feature extractor along with

Support Vector Machine(SVM), Neural Network(NN), Recurrent Neural Network(RNN) as the classifiers, achieving an accuracy of 93.46% on HMDB-51 dataset, UCF101 dataset [21].

2.6 Summary

The literature reviewed so far, gives an overview of the state-of-the-art developments in the field of Human Action Recognition. It is evident that, the representational features have to be extracted from the raw videos. Numerous approaches to address this problem has been reviewed, and the problem remains to select one approach that best find the solution to the problem as stated in the section 1.2. Possible candidates for feature selection and extraction are 1) MediaPipe, 2) OpenPose as they are ready to use and less cumbersome. The table 2.1 compares the human pose estimation results of MediaPipe to that of OpenPose.

Model	**FPS**	**AR Dataset**	**Yoga Dataset**
OpenPose	0.4	87.8	83.4
BlazePose (MediaPipe)	10	84.1	84.5

Table 2.1: OpenPose vs MediaPipe [6]

The table 2.1 shows that OpenPose performs better on AR dataset whereas on Yoga Dataset, MediaPipe exhibits superiority. The Yoga Dataset is more aligned towards the problem 1.2, and MediaPipe is performing better. Also, mediaPipe is modular and platform-indepentant. The computational resources to accomplish the objective addressed in the project is a very important factor to consider. The MediaPipe is lightweight and have superiority in performance when compared to OpenPose. Due to these factors, MediaPipe is the better candidate to perform the feature extraction. After the feature extraction, the intermediate labelling and the action recognition is to be performed using the models reviewed.

Chapter 3

Methodology

3.1 Overview of the Approach

Human action recognition play a quintessential role in the areas of Augment reality, Sign Language Recognition, Full-body gesture control, Human activity surveillance and the quantification of physical activities, etc [3]. As discussed in the previous sections, in this project we apply the human action recognition to identify and assess the loco-motor skills of the children using the videos of children performing the action. This section presents the overview of the steps followed in this project to determine the action performed by the children. Figure 3.1 consolidates the approach taken to accomplish the lingering task.

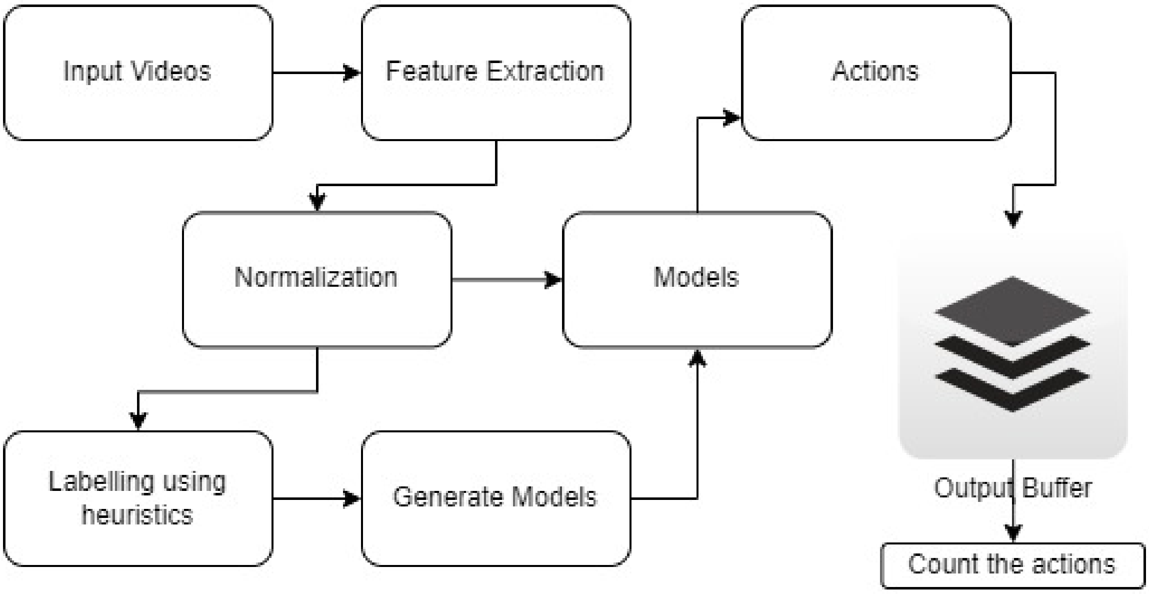


Figure 3.1: Overview of the methodology

The process requires the videos of the children performing the actions under analysis. The system is implemented in Python using various libraries such as Pandas, Sklearn, TensorFlow, OpenCV, and Mediapipe.

As shown in the figure 3.1, the first step in this process is the feature extraction. In this step, the input features which are to be ingested by the machine learning models are identified and extracted. Since, the manual normalization of the videos are not feasible, we extracted the features and based on the extracted features, the normalization is performed in the later stages. The feature extraction is performed using MediaPipe library, which has built-in pipelines for extracting the key landmark points on the human body. The landmark points are extracted with X, Y and Z co-ordinates with the top left corner of the frame as the reference point. By the end of this stage, the skeletal coordinates of the human subject in the videos are extracted frame by frame. In the normalization step, the extracted co-ordinates are remapped to a common dimension to facilitate the later stages of the process. This normalization transforms co-ordinates of the landmark points in the videos into such a way, that the extracted points are in the centre of the frames. After the normalization, the next stage is the labelling of video frames, in which we use the extracted features to label the video frames to quantify them in accordance with their respective action label. This step is performed using predefined custom heuristic functions which identify the actions from the co-ordinates of the landmark points. Upon the completion of the labelling stage, the labelled frames are fed to the appropriate models to predict the actions frame by frames. The possibility of unsupervised learning has been omitted, as the performance of a particular action varies significantly in each attempt. Hence, the possibility of employing clustering to identify the actions requires another research. An output buffer is used to store the actions in each frame, to count the number of times the action was performed. The algorithm 1 summarizes the steps involved in the process described above.

Algorithm 1 Human action recognition for children's video

```
Input video of the child doing an action
Extract the frames using OpenCV
Determine the landmark points using Mediapipe
Normalizing the landmark points
Label the extracted datapoints using heuristics defined
Train the models with the labelled features
Classify the actions using the generated model
Output the classified action along with the count
```

3.2 Data

In any machine learning project, the success of the project is determined by how effectively the associated data can be utilized to obtain the primary objective of the project. The quality and the quantity of the data is paramount in the effectiveness of the machine learning model. The primary objective of this project is to build a human action recognition system which can identify the action performed by children. This section focuses on the data necessary for achieving the purported target, specifically, the data collection and the structure of the collected data.

The data for this project was collected as a part of the previous work of Bossavit and Arnedillo-Sanchez [8]. As a part of their study, they have developed a set of activities matched to the developmental stages of the children and monitored these activities using cameras. They used the videos obtained from this monitoring activity to create a dataset of the videos of children performing the actions such as push up, sit up, kneel, jump, hop, etc. The children belong to different age groups such as, toddlers up to 36 months, the pre-operational stage which includes the children of the age between 2 and 7 years, and the concrete-operational stage including children of the age between 7 and 12 years. The actions performed by the children are categorized into 3 levels in accordance with their ability to perform the action, ranging from level 0 to level 2. The levels are in the chronological order of the complexity and the experience level of the child performing the actions. The collected videos are of varying lengths depending on the amount of time taken by the children to perform a particular action, and in each of the videos, the children attempts to perform the action multiple times. In addition, the videos are anonymized to preserve the privacy of the children in the videos. Since, the collected data involves the videos of the children under the age of 16, the article 8 of GDPR mandates the procurement of parental obligation [18], and the videos were collected through proper legal channels. There are 485 videos depicting imitation action, of which 162 fall under Level 0 category, 162 under Level 1, and 161 under Level 2. There are 462 videos in the Push-ups category, of which 157 are in Level 0 category, 155 under Level 1, and 150 under Level 2. Similarly, there are 475 videos of Sit-ups being performed, of which 160 fall under the Level 0 category, 159 under Level 1, and 156 under Level 2. There are 491 videos of Kneel action, of which 165 videos in the Level 0 category, 164 belongs to level 1, and 162 in the level 2. The table 3.1 summarizes the details of the data in each category.

3.2.1 Challenges

The previous section focused on how the data was collected and was structured. It is widely accepted that the quality of the predictions by any model would be affected by the

Action	**Number of videos**			
	Level 0	**Level 1**	**Level 2**	**Total**
Imitate	*162*	*162*	*161*	*485*
Push Up	*157*	*155*	*150*	*462*
Sit Up	*160*	*159*	*156*	*475*
Kneel	*165*	*164*	*162*	*491*
			Total	***1913***

Table 3.1: Details of the dataset

quality of the data. The source of the data which is to be ingested by the model are often disregarded and misinterpreted. In this project, the data source is the camera capturing the actions performed by the children. Since, the children are in the process of undergoing their cognitive development, with impairments in the ability to read and comprehend the instructions, injects a high degree of variability and unaccountability into the data. They are frequently, unpredictable and unreliable depending on the developmental stage of the child. This volatile nature of children can be attributed as the noise instigated into the dataset. The said fact and the dependency of the models on the data would induce challenges and stonewalls in the process of finding an appropriate solution to the problem to be solved. This section is focused on addressing these challenges in solving this human action recognition problem, and some of them are the following,

Amount of Data: The quantity of the data can significantly impact any machine learning model. The table 3.1 depicts that there are a total of 1913 videos, with approximately 400-490 videos belonging to 4 actions. Depending on the intricacy of the action and the level of expertise required, each action is further divided into three levels. However, after careful evaluations, it was found that, most of the videos in the level 0 and level 1 are unusable. In the level 0 videos, the child was performing the actions for the very first time and was unable to perform the action as instructed. Consequently, in the majority of the level 0 videos, it is observed that the child was doing the actions on freewill in their own terms or completely ignored the instructions. A similar pattern is also apparent in the few of the level 1 videos, where the children learn to perform the actions. On visual inspection, it is evident that in a few instances, the child will be performing some other actions than the ones they are instructed to perform, resulting in videos with a very small portion with the performance of a valid action. Therefore, even though there are 1913 videos with children performing the actions, the amount of the frames in which the children performing valid usable actions are significantly low.

Quality of the videos: Studies proves that the compression quality of the videos can also impact the model performances. Aqqa et al. when evaluated the performance of the deep neural network on videos with 4 different levels of video compression, it was found that the detectors were susceptible to the quality distortions instigated by the compression of video artifacts [2]. The quality distortions are evident in the videos and in addition, in a significant portion of the videos the uniform of the children matches the background of the scenes. This will impact the performance of the tools used for extracting the features from the video frames, as the detectors will face difficulty to distinguish between the child and the background. The amount of compression of the video will remove the patterns, textures, and details from the frames which are paramount for the detection of shapes and edges on which the detectors rely for the classifications. In deep neural networks, the filters in the initial layers might not work as expected, impacting the later stages of the network, and thereby affecting the quality of the model performance.

Quality of the actions performed: The actions defined when performed by an adult will be significantly different from those performed by the children. The children in different stages of development will be having different abilities of loco-motor skills. Individual children, when asked to perform the same action, will perform the action uniquely. For instance, a child when instructed to perform a push-up would be performing a push-up by standing on the knee and lowering the torso to the ground, another child will stand on the toe and lower the torso, similarly a sit-up would be performed with the legs extended rather than laying with bent knees. This suggests that the majority of the motions in the videos corresponding to different actions are invalid in terms of the definition of the action, and this also adds complexity to the task of labelling the actions in each frame.

Normalization of the videos: Normalization is a very essential part of the data pre-processing tasks to prevent or reduce the biasing of models. The normalization is significant in this project because the subject or the child in the videos may not be in the centre of the frames. A fixed position of the child cannot be guaranteed in all the videos. The distance between the child and the position of the camera may not be the same in each and every frame. In order to facilitate the models to learn the actions from the frames, it is desirable to ensure that the child is always in the same position. As the models are learning from the changes in the position of the key points on the child, it is essential to ensure that the positional changes are from a common reference point. It is also important to remove the unnecessary actions or the invalid actions from the videos, and thereby, make sure that the videos of a particular action contains that action alone.

Feature extraction and labelling the frames: As the objective of the project is to develop a supervised learning model, it is inevitable to label the frames used in the training process. This raises the questions such as

- How to generate the labels for the plethora of frames in a single video?
- How to define the criteria to label the frames?
- What features to be extracted from the videos to facilitate the labelling process? How to extract those features?

Privacy: The privacy is always a matter of grave concerns, and the problem is further complicated as the data used in the project are of children under the age of 16. In order to address the privacy concerns, the data has been anonymized by blurring the faces of the children appearing in the videos.

3.3 Feature Identification and Extraction

This section presents the methods used to perform the identification and extraction of the features from the videos to facilitate the learning process by the machine learning models in the subsequent stages of the project. Due to the limited quantity of the amount of data, there are limitations in the application of Convolution Neural Networks which can detect the patterns in the features of the image frames. In this scenario with limited number of videos per action and the low resolution of the videos, the feasibility of a CNN is dubious. Hence, the approach is to extract the features to represent the actions and later use appropriate models to predict the actions. Now the questions are :

- What data points are to be extracted?
- How to extract the points?

MediaPipe library, in conjugation with OpenCV, has been used to answer the aforementioned questions. OpenCV is a library for computer vision based applications, and it is used to extract the frames from the videos. The MediaPipe machine learning library detects the region of interest, which is the area of the frame in which the human subject appears, and identifies the landmark points on the skeletal structure of the human body in the frame. This library uses the BlazePose detector, a lightweight convolutional neural network for pose estimation, which identifies 33 key points on the human body. [7]. The key points topology is shown in the figure 2.1. This project uses these key points as the features to perform the action estimation. The landmark points or the key points estimated by the MediaPipe library are exported as a data frame. As it is a time-consuming and computationally intensive process, the key points are extracted from all the level 2 videos frame by frame, of a particular action on a single go. A normalization process was applied on to these extracted points which will be discussed in the next section. The processed points were exported to create a .csv file to be imbibed in the later stages of the process. The names of the landmark points were used as the column names, along with the file name of the video from which the frames were extracted, facilitating the backward referencing to the videos.

3.4 Normalization of the Frames

Normalization is performed in any machine learning activity to ensure that the model is not skewed towards any unwanted patterns in the data. In most of the machine learning processes, the normalization of the data is performed in the early stages of the data preparation. In this project, the normalization is paramount owing to the unpredictable nature of the children. For this reason, it is not guaranteed that the position of the child is always in the centre of the frame. A model trained on these data might produce undesirable results. Hence, we need to ensure that the child or the region of interest is in the prime focus, in each of the frames extracted using the OpenCV library. In this project the normalization cannot be performed manually as it is not a viable approach. Hence, we attempt to automate this step by creating a function which takes the input from the Feature identification and Extraction step.

In the normalization stage, the input frames are normalized with respect to the hip of the child. The frame extracted with OpenCV is fed to the MediaPipe pose detection algorithm. The MediaPipe emits 33 landmark points on the skeletal structure of the child in the frame. Then the x and y co-ordinate position of the hip is used to ensure that the child is in the centre of the frame. This is done by taking each of the landmark points and remap them to a 640*360 frame while ensuring that the hip is in the centre of the frame. The algorithm 2 represents the normalization function. With this algorithm, we remap all the x and y coordinates to a normalized frame. As discussed earlier, these new coordinates are extracted into a .csv file for the later stages of the process.

3.5 Labelling the Extracted Frames

This section focuses on the process of labelling the frames to be ingested by the model. As this project involves the use of a supervised learning process, it is essential to label the frames before they are fed to the models. This poses a new challenge of how to perform

Algorithm 2 Normalization Algorithm

```
Initialize,
$x$ ←x co-ordinate of the point to be remapped,
$y$ ← y co-ordinate of the point to be remapped,
$ht$ ← height of the frame,
$wd$ ← width of the frame,
$n\_ht$ ← new height of the normalized frame,
$n\_wd$ ← new height of the normalized frame,
$x\_hip$ ←x co-ordinate of the hip,
$y\_hip$ ← y co-ordinate of the hip,
for all $x$ and $y$
repeat
    $n\_x \leftarrow (x * wd + (n\_wd/2 - x\_hip * wd))/n\_wd$
    $n\_y \leftarrow (x * ht + (n\_ht/2 - x\_hip * ht))/n\_ht$
until no more $x$ and $y$
```

the labelling. One solution to this problem is to manually label all the frames. However, it is an unfeasible approach which requires unquantifiable amount of time. Hence, it is undeniable that a new approach is to be taken which require no manual intervention. Therefore, for each of the action, custom heuristics are to be defined to perform the labelling of the frames.

The frames are extracted in the form of a .csv file with names of the landmark points representing the columns and the file name which points to the original video from which the frames were extracted. This data is the input, on to which the heuristics are to be applied to perform the labelling. It is not viable to define a single heuristic to label all the actions and hence, there is a necessity to define different heuristics for each of the actions in question. The following section discusses the rationale behind each heuristic and explains how it was defined to overcome the challenge of labelling the frames.

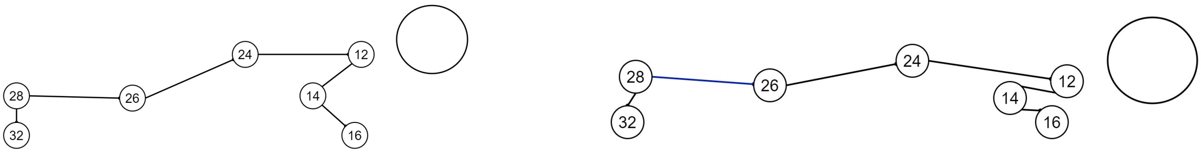
3.6 Heuristics

Upon reaching this stage, the data points from the frames are extracted and are normalized. That being said, the next step is to define custom heuristics functions to enable the labelling of the extracted frames. This is a very important step as it directly influences the training and thereby, impact the performance of the generated models. Hence, the data points have to be analysed carefully and the performance of the heuristics function have to be optimal. On the analysis of the data points it was found that, it is not pragmatic and therefore, it is necessary to consider each of the actions separately in order to define the heuristics specific to a particular action. For this reason, in the following section, the heuristics are examined in detail for each of the actions under consideration.

1. **Push Up**

   As discussed in the section 1.2, a push is defined as follows:

   “The child is lying on the ground facing down on the hands. The child lowers the trunk until (almost) reaching the ground and rises again the trunk”

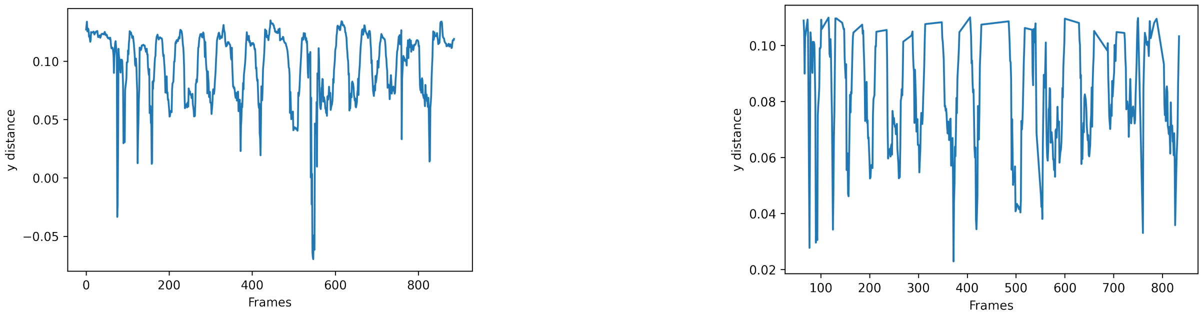


(a) Push-up Pose 1

(b) Push-up Pose 2

Figure 3.2: Anatomy of a Push-up

   The objective here is to define a heuristic function to classify the frames into push up and non push up. The figure 3.2a represents the skeletal topology of, with the key points representing the initial position of a push-up, whereas the figure 3.2b depicts that of the final position of the push-up. From the figures, it is evident that the points 12, 14, and 16 are exhibiting a significant amount of motion. Considering the points 12 and 16, the points on the shoulder and wrist respectively, there is a clear change in the y co-ordinates. In the initial pose of the push-up, where the child lays with trunk parallel to the ground, the shoulder and the wrist are away from each other, meanwhile, when the child lowers the trunk to perform the final pose of the push-up, the shoulder and the wrist are closer to each other. These changes in the y co-ordinates can be used to create a heuristic function to label the frames. The heuristic function classifies the frames into push-up and non push-up based on a threshold value in the difference of y co-ordinates of the shoulder and the wrist. The figure 3.3a represents the plot of the difference in the y-coordinate of the shoulder and wrist over each of the level 2 videos of children performing the push-up action. On detailed analysis of the extracted data and the videos it was clear that, the outliers in the present in the beginning of the plot 3.3a indicates that the child is in preparation for performing a push-up, and those at the end of the plot, corresponds to the child returning from the push-up. From the manual verification of 15 videos, it is evident that when the y difference in the co-ordinates of the shoulder and the wrist are greater than 0.12, the child is not performing a push-up. Using OpenCV, we have trimmed the videos by extracting the part of the videos in which the child is performing a push-up. The plot from the trimmed videos are shown in the figure 3.3b. From the resulting plot, a threshold value of 0.08 was set and the extracted frames where annotated as push-up, if the y difference is less than this threshold. The defined heuristics was manually verified over 15 videos and was working as expected.

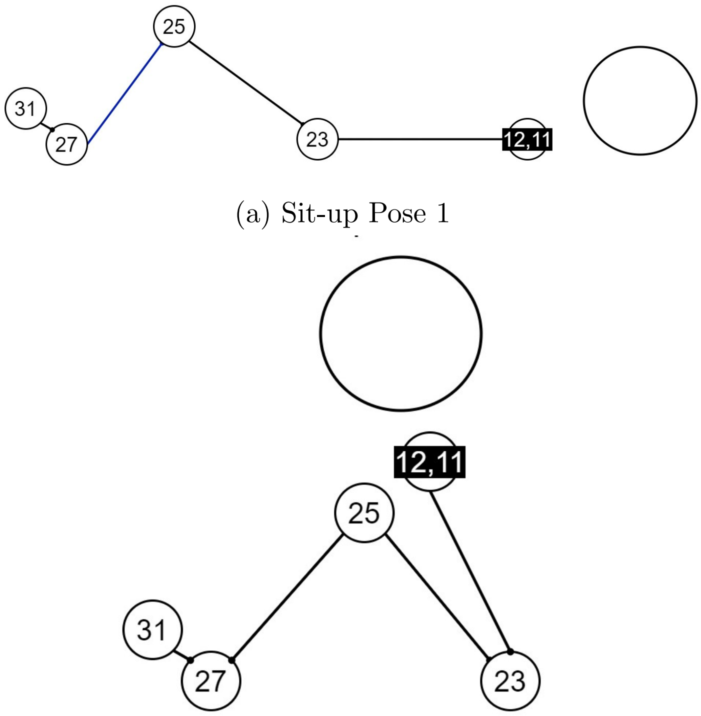


(a) Before Trimming

(b) After Trimming

Figure 3.3: Difference in y co-ordinates between shoulder and wrist

2. **Sit Up**



(a) Sit-up Pose 1

(b) Sit-up Pose 2

Figure 3.4: Anatomy of a Sit-up

The criteria for the sit-up as defined in the section 1.2, is as follows :

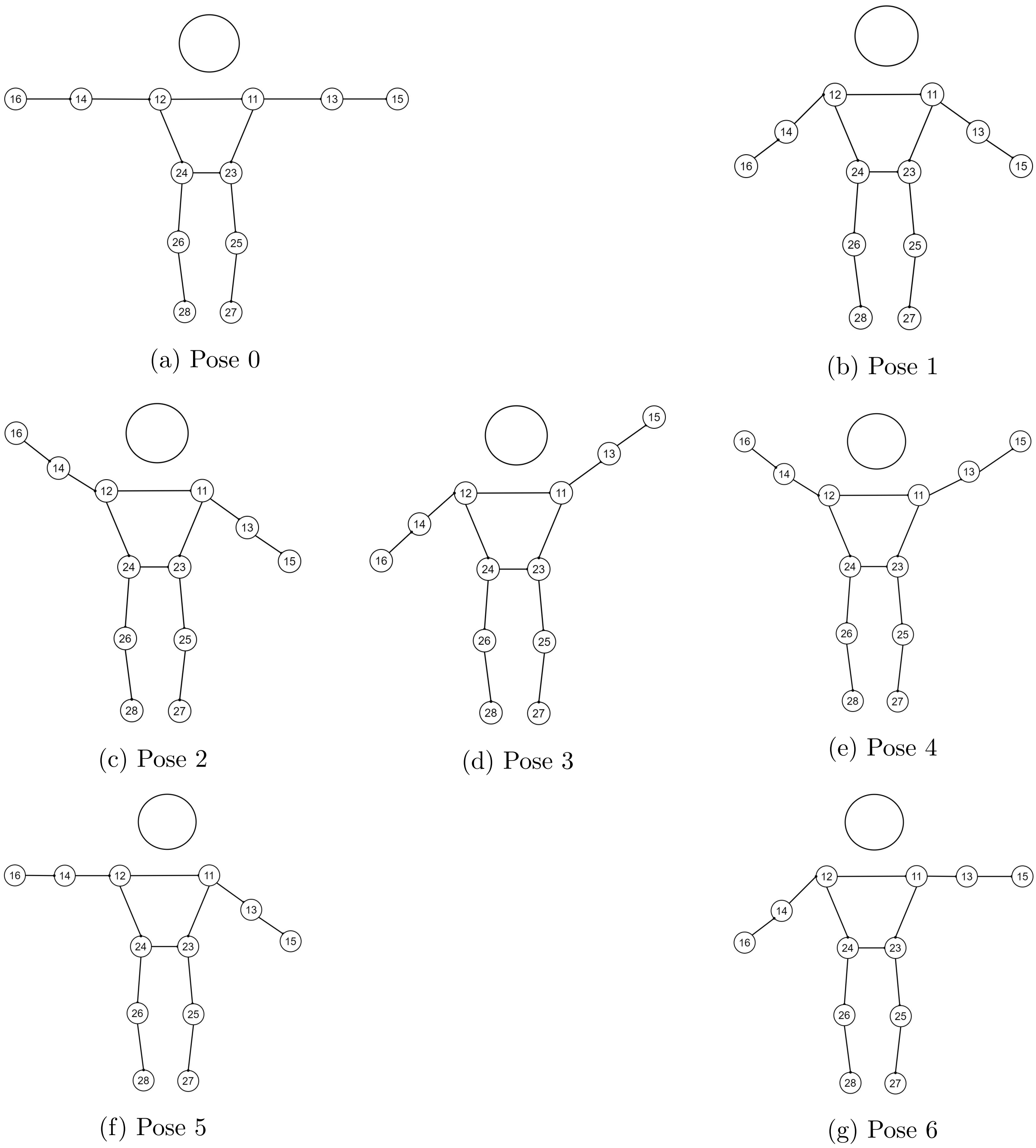
"The child is sitting on the ground. The child lowers the trunk until (almost) reaching the ground (face up)and rises the trunk again"

According to the aforementioned criteria, a sit up is composed of 2 poses. The first pose is as shown in the figure 3.4a, where the child sits on the ground with slightly bent knees and facing up, with the trunk until or almost reaching the ground. The second pose in the sit-up is depicted in the figure 3.4b, in which the child attempts to bring the trunk closer to the knees. Both the figures 3.4a and 3.4b, represents the skeletal topology of the child performing the sit-up action and are composed of the key points 31, 27, 25, 23, 11, and 12, which are the toe, ankle, knee, hip, and shoulder of the left-hand side of the child as well as the right shoulder, respectively. The objective is to label the frames into sit-up and non-sit-up using a heuristic function. While considering the figures 3.4a and 3.4b, it is evident that the points in the shoulder are coming closer to that of the knees. Hence, we can define a heuristic function based on this inference. Here, the motion of the shoulder points are evident in the x co-ordinates. The x distance between the knees and the shoulders are reduced significantly during the performance of a sit-up. Therefore, the heuristic function is defined to classify the frames sit-up and non-situp, according to the difference in the x distance of the position of the knee and the shoulder. However, when the videos under consideration were analysed, it was found that, the child is standing in the start of the videos. In order to extract the sit-ups from the videos, we need to determine the sit-up from the videos using the extracted feature. It was found that, in the standing position, there is a significant difference in the position of the x coordinates of the shoulder points 12 and 11. However, in the sitting position for a sit-up, the x distance between the right and the left shoulders are almost equal to 0.

Using this rationale, the sit-ups were extracted from the input videos and the shoulder-knee, x difference were plotted. When this distance is less than a threshold value of 0.04, the frames can be classified into sit-up, else into a non-sit-up. This threshold value was selected after analysing the changes in the x distance between these 2 key points of all the videos in which a sit-up is performed. Using this logic, the frames were classified. The final heuristic function function was able to classify the frames successfully into 2 labels, sit-up and non-situp. The results were manually analysed by comparing 15 sit-up videos.

3. **Imitate**

   In the imitate action, the child attempts to replicate one of the 7 actions shown to him/her in the best possible ways. The actions/poses are shown in the fig-



(a) Pose 0

(b) Pose 1

(c) Pose 2

(d) Pose 3

(e) Pose 4

(f) Pose 5

(g) Pose 6

Figure 3.5: Poses in Imitate

ure 3.5. The poses are numbered from 0 to 6 for the purposed of identification. The objective at this stage is to define a heuristic function which will label the extracted frames into one of the 7 poses. As discussed in the previous actions, the initial approach taken was to develop a heuristic function based on the position of the keypoints on the skeletal topology of the child as shown in the figures 3.5a, 3.5b, 3.5c, 3.5d, 3.5e, 3.5f, and 3.5g. On the analysis of the data corresponding to the points in action, it was found that it was not a hurdle free approach. However, when we consider the angles between the limps, there is a clear distinction between each of the poses. Hence, for the imitate action, a different approach has been taken, where the angles between the limps are considered to proceed with the classification. The heuristic defined here determines the angle between the points, 14, 12, and 24 giving the angle between the right limb and the hip, also the angles between the points 13, 11 and 23 giving the angles between the left limb and the hip. The angles are computed using the functions from the numpy and OpenCV library. Based on the computed angles, the heuristic function classifies the frames into 7 different poses. The table 3.2 shows the criteria defined to determine the pose from the angles.

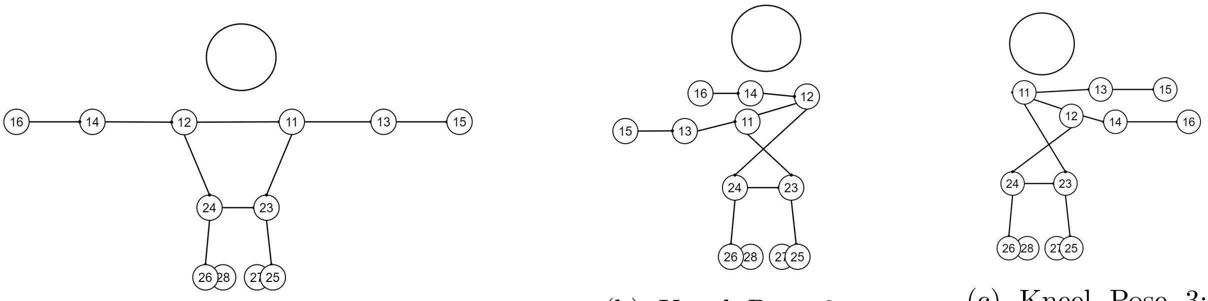
Angle between the limb and the hip		**Pose**
left	**Right**	
90-110	90-110	Pose 0
60-70	60-70	Pose 1
60-70	110-130	Pose 2
110-130	60-70	Pose 3
110-130	110-130	Pose 4
90-110	60-70	Pose 5
60-70	90-110	Pose 6

Table 3.2: Criteria for Pose inference from angles

4. **Kneel**

   The criteria for the Kneel is defined as follows:

   “Count the number of times the child turns leftwards and rightwards while kneeling down”



(a) Kneel Pose 1

(b) Kneel Pose 2: Turning to right

(c) Kneel Pose 3: Turning to left

Figure 3.6: Anatomy of Kneel

Here the objective is to label the frames into 'kneel', 'kneel-and-turn-left' and 'kneel-and-turn-right'. The figures 3.6a, 3.6b, and 3.6c shows the topology of the skeletal structure during the performance of the action. The objective here is to define a heuristic function which has the ability to perform the classification of the frames into the appropriate labels. However, this objectives was not achieved in this project. A detailed discussion of the failure would be carried out in the later sections of this dissertation, refer section 5.2.

3.7 Models

The following section focuses on answering the following research questions:

- What models were used to perform the identification of the actions?
- How the selected models were used?

The following sections, presents the methods employed to answer the above questions. The models used to perform the classification are Random Forest, Deep Neural Network and LSTM. Due to the limited resources and time constraints, the LSTM model was used only to infer the sit-up action.

Deep Neural Network

From the extracted data frames, input features which exhibited the most range of motions were selected to feed the deep neural network. Depending on the actions, the input features are varied. The network consist of an input layer with 'relu' activation function, followed by 3 dense hidden layers accombined by a 'softmax' dense layer. The number of nodes in the final layer is dependent on the number of labels to be classified by the model. The loss function used in the network is 'categorical_crossentropy' and with an 'adam' optimizer. The model was trained for 100 epochs. The tensorflow library was used to define the model. The architecture of the model is shown in the figure 3.7

Random Forest Classifier

A random forest classifier is a variation to the decision tree which aims to improve the performance by fitting numerous decision tree on various sub samples of the dataset using averaging as a means to control the inherent over-fitting of decision tree, thereby substantially increase the predictive accuracy. The random forest model used in this project was built over the entire data of a particular action, with the number of sub samples = 5, max depth = 5, and was parallelized to reduce the computational time.

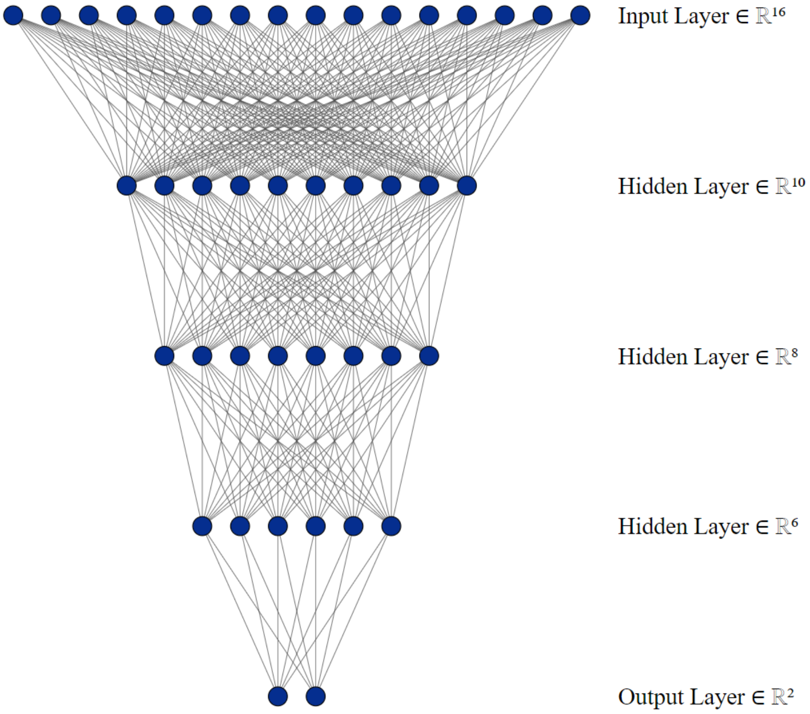


Figure 3.7: DNN Model Architecture

Long-Short Term Memory

Long-Short Term Memory is a variation of Recurrent Neural Network which has the capability to learn ‘long-term-dependencies’. The LSTM is employed in this project to infer the sit-up actions from the extracted features. There are different variations of LSTM such as the one-to-one, one-to-many, many-to-open and many-to-many, etc. This project employs the use of many-to-one LSTM, where the the LSTM takes a chunk of input data as a sequence to emit one output. The network structure of the LSTM is as shown in the figure 3.8. A cell state is an important concept in the LSTM network and is represented by the straight line on top of the diagram. The information flow through the cell state is regulated by structures called Gates. These gates accelerate the learning process by imposing control over the information to be

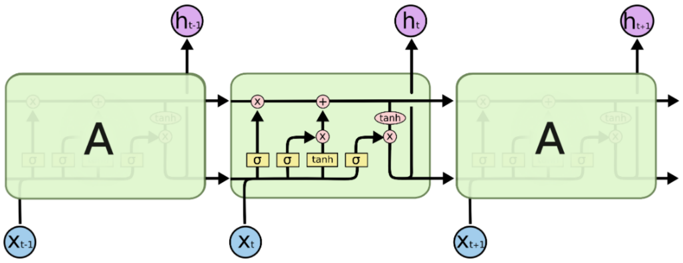


Figure 3.8: LSTM structure with 4 layers [1]

persisted and discarded. The important gates are the Forgot gate, Input gate and the Output gate. The gates are primarily an amalgamation of Sigmoid neural network layer and point-wise multiplication operation. The forget gate is a sigmoid layer useful in discarding the unnecessary information from a cell state. The input gate which is a sigmoid layer makes the decision on what information to be updated. The output gate decides what information to be emitted as the output from the current cell state. In this project, the many-to-one LSTM is employed to predict the sit-up actions. Since, the LSTM consumes a chunk of frames as the input, there is a need to group the data into time steps or sequence. The data frames extracted from the videos where labelled using the custom heuristic function. This data is inadequate for an LSTM model, and the LSTM model expects the data as a 3 dimensional matrices. Therefore, the extracted data frames has to be grouped into a sequence and to be labelled. However, there is a problem of identifying the ideal length of the sequence, as the length of the action being performed varies in different videos. There are significant variations in the duration of actions performed by the children, or the duration of actions performed by the same child. After careful consideration and analysis, it was found that the average duration of sit-up is less than 20 frames. This average number is totally depended on the defined heuristics and manually verifying all the videos is not a feasible approach. Since, the sit-up in the videos are the minority class, the 20 chunk of frames of the videos were annotated with sit-up as the label if there is a sit-up in the selected chunk of video. The data frame was created with new annotation to be ingested by the LSTM model. The results obtained are discussed in the later sections.

3.8 Output Buffer & Action Counting

After the model generation and inference of actions using the generated models, the problem definition defined in the section, 1.2 mandates to count the number of times an action is performed. In order to achieve this objective, an output buffer is used to store the labels predicted by the model. The frames from the videos of actions such as Push-Up, and Sit-Up, has been transformed as a binary classification problem. Hence, the output buffer will have two labels in the order in which they appear in the frames. Consider the example involving a push-up action. The labels are NotPushUp and PushUp. Therefore, the output buffer will hold these two labels. When a child performs a push-up, the label transforms from NotPushUp to PushUp. Using this rationale, whenever this transformation occurs, it can be interpreted as a Push-up. However, when this logic was implemented and tested, it was found that the interpreted number of push-up was significantly different from the actual number, verified manually. On analysis, it was found that, the transformations in the output buffer was much frequent than expected owing to the misclassifications from the models used. However, this effect of misclassifications can be counteracted by by optimizing the output buffer. On the analysis of a sequence of output buffer labels during the performance of an action, it was found that, a few of the labels inside this sequence, would have been misclassified. Hence, a sequence of labels are considered together and the majority class in the sequence is taken as the action of that sequence. Thus, the actions are counted by counting the number of transformations in a sequence of actions. This new approach, significantly improved the action counter and lowered the disparity with the actual numbers. However, it arises another problem of selecting the length of sequence. Currently, the length of the sequence is selected after the manual analysis of the buffer for a couple of videos. When compared to the original approach, this new approach performs significantly better.

Chapter 4

Evaluation

This section presents the experiments conducted and the discussions on the results obtained.

4.1 Experiments

The objective of the dissertation, was to produce machine learning models which are capable of identifying the actions from the in-house dataset of children performing those actions. The experiment involves the training and the evaluation of the models as explained in the section, 3.7 and was conducted on HP EliteBook laptop with i7 11$^{\text{th}}$ generation processor and 16GB RAM. From the collected videos, the features were extracted for each of the actions. Then the extracted data was split to form the test-train split. This data was fed to Deep Neural Network, Random Forest Classifier and LSTM models based on the actions and the training was conducted. Once the training was completed, the models were evaluated on the test data and the results were obtained. The metrics used to evaluate the models are Accuray and Confusion Matrix as this problem can be considered as a classification problem. Because of the quality and quantity of the data, unsupervised learning models were not used in the experiments. The results obtained after the training of the models are presents in the ensuing sections. The table 4.1 presents the accuracy obtained for the Deep Neural Network (DNN), Random Forest Classifer (RFC), and Long-Short Term Memory (LSTM) models. The table indicates, the superiority of DNN models when compared to other models. However, the LSTM models were expected to perform better by learning from a sequence of the actions. But, in terms of accuracy the LSTM is slightly behind the DNN model. The possible reason for this, will be discussed in the subsequent sections.

4.2 Results

	DNN	RFC	LSTM
Push-up	0.847	0.806	-
SitUp	0.814	0.702	0.802
Imitate	0.813	0.809	-

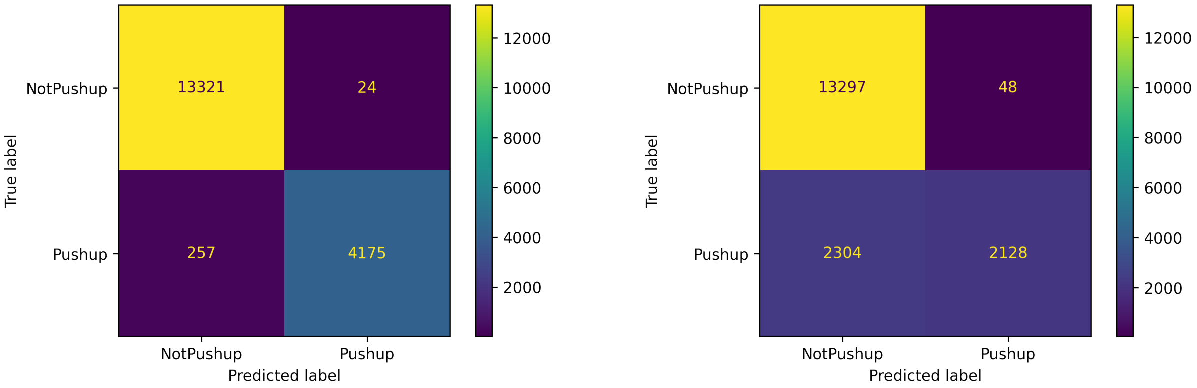
Table 4.1: Accuracy on validation data from different classifiers

In this section, each of the actions are considered separately, and presents the results obtained for each of the actions under consideration. In addition to the accuracy, the confusion matrix is used to evaluate the performance of the models. It is a contingency table which displays multivariate frequency distribution in the form of a matrix, and is extremely helpful in the calculation of Recall, Precision, Specificity, Accuracy, etc of a machine learning model. The confusion matrix displays the true labels against the predicted labels. The 'True Positives' (TP), 'True Negative' (TN), 'False Positive' and 'False Negative' can be easily determined from the matrix. The TP gives the number of outcomes from the model where the model was able to predict the actual label accurately. The TN provide the number of outcomes where the model predicted the negative labels as negatives. The FP of a class gives the number of outcomes of the classifier where it predicted the a negative label as positive and the FN is the number of outcomes when the model predicted positive labels as negative. The confusion matrix aids in the comparison between the actual and the predicted values of a model, thereby helping to evaluate the performance of a machine learning model. On the analysis of the data, it a clear class imbalance is evident in the data. Hence, the use of accuracy alone to evaluate the performance of the models might be inadequate, to understand how the classifier performs on each of the classes.

4.2.1 Push Up

The figure 4.1 presents the confusion matrix corresponding to the outcomes from the Deep Neural Network and Random Forest Classifier. From the figure, it is observed that, majority of the false predictions are made in the NotPushUp class. This might because of the imbalanced class distribution occurred from the feature extraction. Since, most of the machine learning algorithms are designed with the expectation of balanced classes in the ingested dataset, the algorithms misclassifies the minority classes with the labels of majority class. This is evident in the figure 4.1. Up-sampling and down-sampling are the techniques to balance the class distribution in multi-class imbalance data problems. In this scenario, down-sampling is not feasible due to the limited quantity of the data.

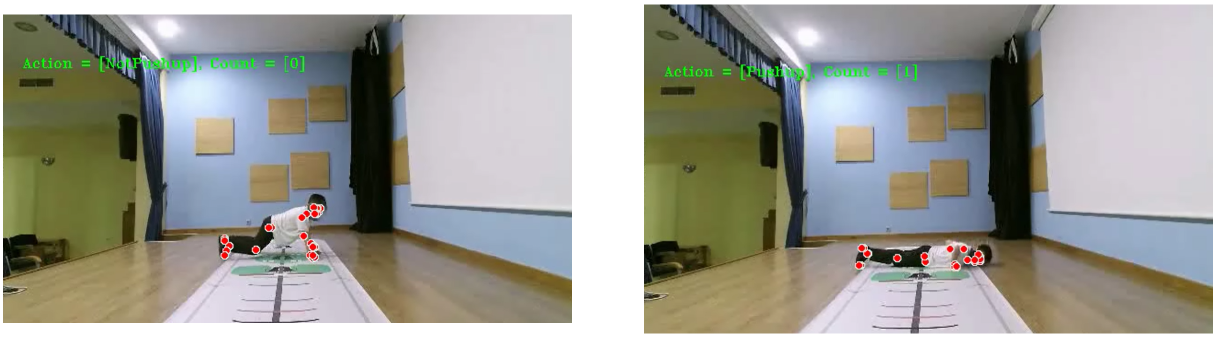
Hence, up-sampling techniques such as SMOTE[12] can be employed. However, due to the time constraints and obtained accuracy, the technique was not adopted. The table table 4.1 shows the accuracy obtained from the DNN and Random Forest Classifies. The accuracy obtained for the DNN model is 84% where as that of Random Forest Classifier is 80%. The metrics accuracy and confusion metric



(a) Deep Neural Network

(b) Random Forest

Figure 4.1: Confusion Matrix for Sit-Ups


Action = [NotPushup], Count = [0]
Action = [Pushup], Count = [1]

(a)

(b) Random Forest

Figure 4.2: Sample outputs for Push-Ups

The figure 4.2 shows sample output from the model. The table 4.2 shows the predicted count from the system and the actual count manually verified for 15 videos using the DNN model. It is clear that the difference in the count is not significant due to the output buffer optimization step performed as discussed in the section, 3.8.

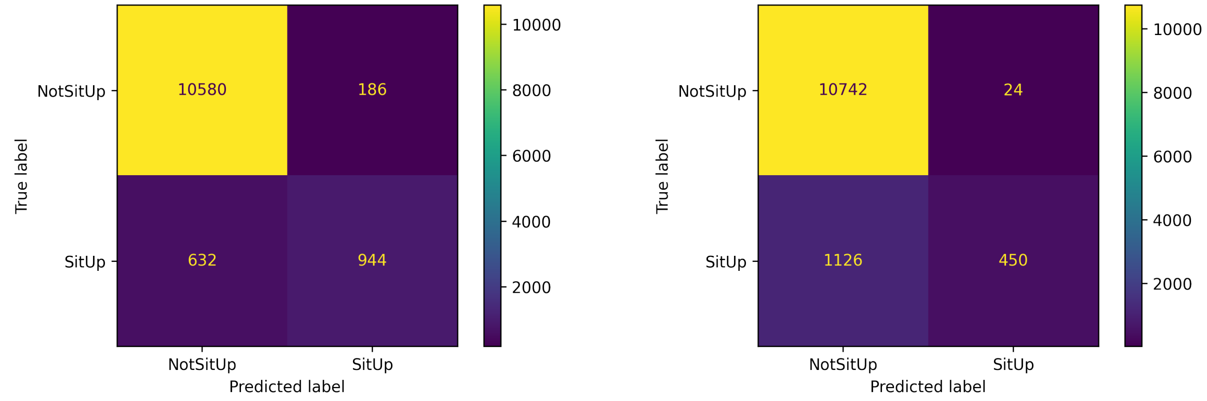
4.2.2 Sit Up

The experiment conducted for the push-up was followed for sit-up. The models used in this experiment were Deep Neural Network, Random Forest and LSTM. The confusion

	No.	**Actual Count**	**Predicted Count**	**Difference**
Push Up	1	4	4	0
	2	8	8	0
	3	6	7	-1
	4	5	5	0
	5	8	7	1
	6	9	9	0
	7	5	4	1
	8	5	6	-1
	9	7	7	0
	10	5	4	1
	11	7	6	1
	12	8	8	0
	13	9	7	2
	14	5	4	1
	15	8	8	0
SitUp	1	5	4	1
	2	9	10	-1
	3	4	3	1
	4	6	5	1
	5	3	3	0
	6	4	3	1
	7	5	4	1
	8	5	6	-1
	9	7	7	0
	10	5	4	1
	11	7	7	0
	12	8	6	2
	13	5	7	-2
	14	3	4	-1
	15	3	3	0

Table 4.2: Actual and predicted counts for Push-Ups and Sit-Ups

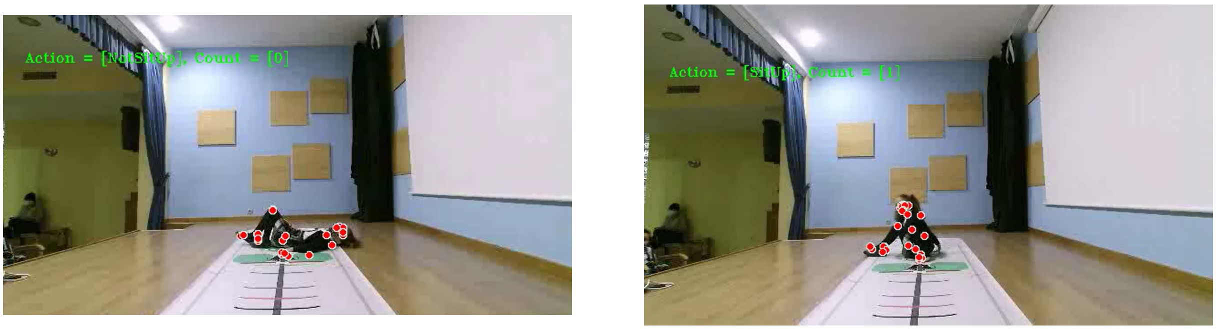
matrix obtained for DNN, and RFC are shown in the figure 4.3. The accuracy obtained for DNN was 81%, where as it was 70% for the RFC, and LSTM showed an accuracy of 80%. The LSTM was expected to perform better than the DNN. This might be because of the selected time step. There were limitations to determine an accurate time step and it was selected to 20 frames, as it was observed in the extracted data, that a duration of sit-up was less than 20 frames. This may not be correct for all the sit-ups in the videos of the training data. Due to the time constraints, further experiments were not conducted to improve the accuracy of LSTM. As a proof of concept, LSTM seems to be a good candidate for sit-up. With further research the time step can be determined accurately and the missing data can be imputed, leading to better performance for the LSTM. The observed and the actual count for manually verified videos are shown in the table 4.2.



(a) Deep Neural Network

(b) Random Forest

Figure 4.3: Confusion Matrix for Sit-Ups


Action = [NotSitUp], Count = [0]
Action = [SitUp], Count = [1]

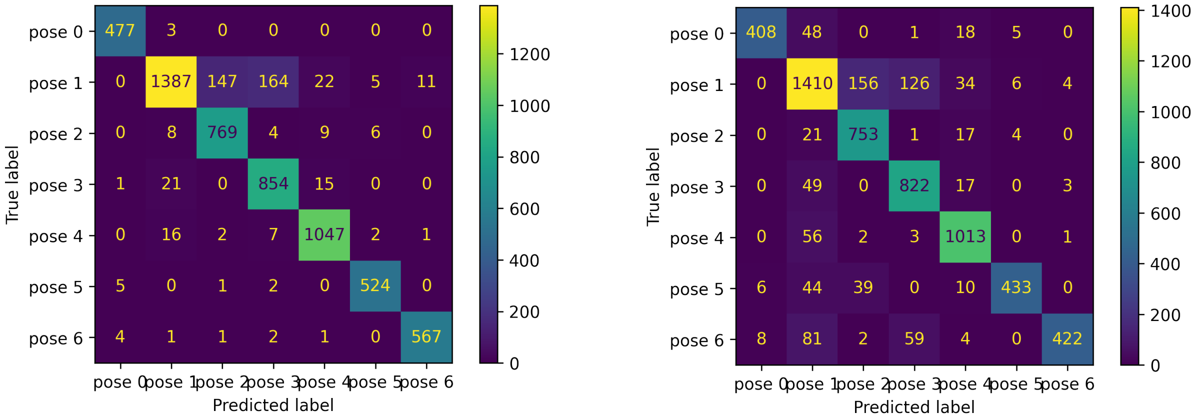
(a)

(b) Random Forest

Figure 4.4: Sample outputs for Sit-Ups

4.2.3 Imitate

The imitate actions were trained on Deep Neural Network (DNN), and Random Forest Classifier (RFC).The DNN obtained an accuracy of 81.3%, followed by RFC with an accuracy of 80.9%. The confusion matrix for both of the classifiers are shown in the figure 4.5 and the figure 4.6 shows a sample from the model performance. The pose 1 is the majority class from the confusion matrix, as it is the most common intermediary step for getting into the rest of the poses. Hence, most of the misclassifications has been labelled as Pose 1.



(a) Deep Neural Network

(b) Random Forest

Figure 4.5: Confusion Matrix for Imitate

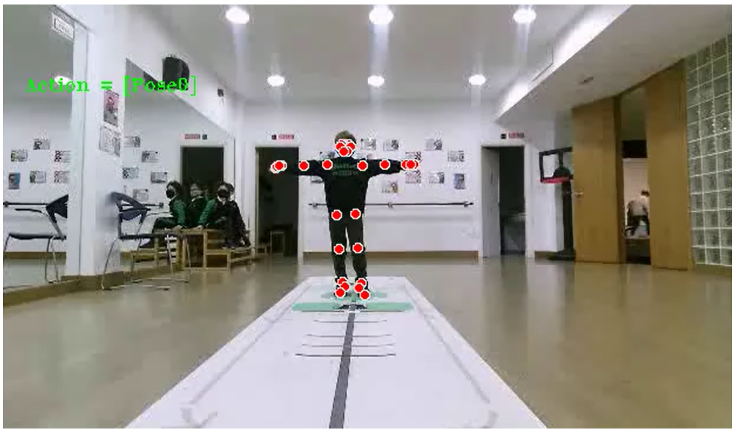


Figure 4.6: Sample output for Imitate

4.2.4 Summary

The inferences and reflections from the experiments can be summarised as follows:

- Deep Neural Network performed better for all the actions.
- Majority of the data misclassification are attributed from the models classifying the minority classes with a majority class label.
- LSTM is a better candidate for future experimentation, as it can infer the actions by considering the action as a sequence of steps.
- Class imbalance affected the accuracy of the classifications and can be addressed in the future using over-sampling techniques.
- DNN displayed better generalisation compared to Random Forest.
- The quality of the videos affected the feature extraction and there by influenced the final model performance.

Chapter 5

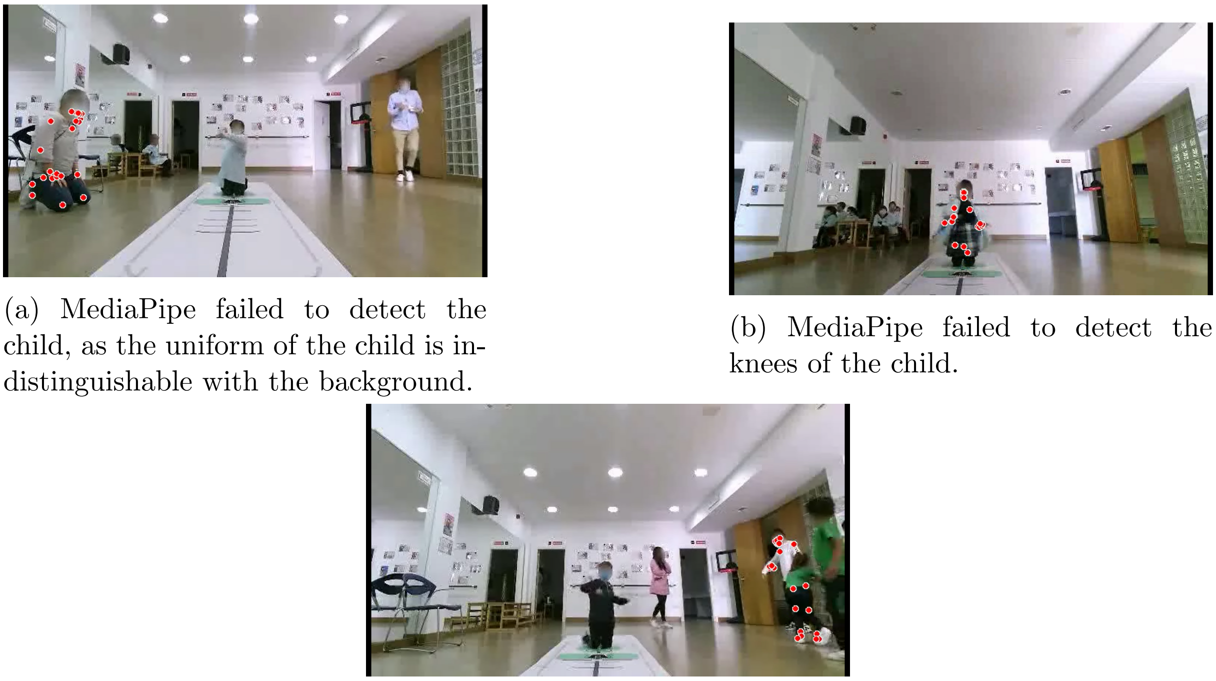
Conclusions & Future Work

5.1 Conclusion

In the modern era of digital world with mixed and augmented realities, the ways humans interact with the new sub-spaces of realities are rapidly evolving. Human Action Recognition is a very important substrate for a human-machine interactive systems. It is a growing field of challenging research with immense potential to transform the world, with applications ranging from autonomous navigation systems to humanoid robots assisting the elderly and children. The objective of this dissertation was to develop an approach to interpret actions of children to assess their loco-motor skills from the early stages of physical and cognitive development. The work aimed to identify the actions as well as count the number of times certain actions such as Push-up, Sit-up were performed. The research was started with the review of state-of-the-art literature on the subject and developed an overview of the general approaches adopted to solve the problem of human action recognition. The data for the work was obtained as a part of a previous research. The features from the videos were extracted using state-of-the-art technology and was fed to various machine learning models. Later, performance of the models were evaluated using accuracy and confusion matrices. Even though, the model performance were not as expected, the proposed system was able to infer the counts of actions with very less disparity with the actual counts.

5.2 Limitations

The quality of any machine learning model can be attributed to the quality and quantity of the data it imbibes. This section presents the limitations of this work. As camera was always in a fixed position, for instance in the videos of children doing the push-



(a) MediaPipe failed to detect the child, as the uniform of the child is indistinguishable with the background.

(b) MediaPipe failed to detect the knees of the child.

(c) Lot of people in the background and the MediaPipe failed to detect the child.

Figure 5.1: Response of MediaPipe on Kneel Action

ups, the camera is always capturing the side view, whereas for kneels the camera always captures the front view. Hence, the model for push-up is expected to perform better on the videos where the child is doing the action by giving the side-view to the camera. Therefore, the models mandates the action videos from a specific position. This can be considered as a limitation of this project due to the limited quantity and distribution of the training videos. The major concern in this project was the feature extraction and the intermediate labelling of the extracted features for training the models. Manually labelling the extracted features was not a feasible approach due to the time constraints and resource constraints for achieving this work, and this dissertation relayed on custom heuristics to achieve this task. Hence, the performance of the final model is dependant on the quality and performance of the defined heuristics. This work was also limited by the quality of the data as discussed in the section, 3.2.1. This work is heavily dependent on the MediaPipe library as it is used for the feature extraction.

This research was not able to develop a machine learning model for Kneel action. The objective was to count the number of times the child in the video turned to the right and to the left while in the kneeling position. This was not achieved as the MediaPipe failed to detect the knees of the child in a vast majority of the videos. A number of reasons can be stated for this anomaly such as 1) the uniforms of children were comparable to the backgrounds of the videos, 2) in all the of videos the children were in the kneeling position with their overcoats covering the knees, 3) the ground was indistinguishable as the child was wearing the dress with same colour of the ground, etc. Hence, the project failed to meet the objective of detecting the kneel action according to the predefined criteria. The figure 5.1 represent a few of the instance where the MediaPipe failed to meet the required objectives. The working models were not tested on videos were the parts of body are occluded as the data was unavailable. The model expects only one child as the subject and this can be considered as another limitation of the current system.

5.3 Future Work

This section presents the scope of future research for this systems. The scope of future research can be directed in two directions by 1) focusing on improving the efficiency of the system with the limited quantity and quality of the data, 2) diversify the data by collecting more data. Currently, the project is delegated into different groups, where each person develops models for a subset of actions. Therefore, integrating all the sub-projects into a single system is the next major milestone in the project, which will include more recognizable actions into the system. More research can be conducted to explore the feasibility of the use of other machine learning techniques such as RNN and LSTM. The feature identification and extraction was a major challenge of this project. Hence, additional man power and resources can be deployed to manually extract individual actions from the videos and annotate them, or manually annotate the extracted frames rather than relaying on the heuristics to do it. By extracting individual actions, it will be straightforward to represent these actions as a sequence and can be consumed by models which are inherent good on sequence data such as RNN, LSTM, etc. Using the current data, it is an intricate task to determine the length of a sequence to be fed to a sequence model as the time taken for each of individual action varies drastically. In the current implementation LSTM is used only for sit-up action. In this approach, the chunk of frames are labelled sit-up if there is a sit-up in the chunk. Another approach can be implemented in future, by modeling the transition happening in the chunk. Another issue, that was evident in this work, was the class imbalance problem. The future works can consider employing the techniques such as SMOTE[12] for oversampling.

Bibliography

[1] Understanding LSTM Networks – colah's blog. URL `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[2] M. Aqqa., P. Mantini., and S. Shah. Understanding how video quality affects object detection algorithms. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 96–104. INSTICC, SciTePress, 2019. ISBN 978-989-758-354-4. doi: 10.5220/0007401600960104.

[3] V. Bazarevsky and I. Grishchenko. On-device, real-time body pose tracking with mediapipe blazepose, Aug 2020. URL `https://ai.googleblog.com/2020/08/on-device-real-time-body-pose-tracking.html`.

[4] V. Bazarevsky and F. Zhang. On-Device, Real-Time Hand Tracking with MediaPipe, Aug. 2019. URL `https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html`.

[5] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus, 2019. URL `https://arxiv.org/abs/1907.05047`.

[6] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann. Blazepose: On-device real-time body pose tracking. *CoRR*, abs/2006.10204, 2020. URL `https://arxiv.org/abs/2006.10204`.

[7] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann. Blazepose: On-device real-time body pose tracking, 2020. URL `https://arxiv.org/abs/2006.10204`.

[8] B. Bossavit and I. Arnedillo-Sánchez. A Novel Approach to Monitor Loco-Motor Skills in Children: A Pilot Study. In M. Scheffel, J. Broisin, V. Pammer-Schindler,

A. Ioannou, and J. Schneider, editors, *Transforming Learning with Meaningful Technologies*, volume 11722, pages 773–776. Springer International Publishing, Cham, 2019. ISBN 9783030297350 9783030297367. doi: 10.1007/978-3-030-29736-7_86. URL `http://link.springer.com/10.1007/978-3-030-29736-7_86`.

[9] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug. 1996. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00058655. URL `http://link.springer.com/10.1007/BF00058655`.

[10] T. Brown and A. Lalor. The Movement Assessment Battery for Children—Second Edition (MABC-2): A Review and Critique. *Physical & Occupational Therapy In Pediatrics*, 29(1):86–103, Jan. 2009. ISSN 0194-2638, 1541-3144. doi: 10.1080/01942630802574908. URL `http://www.tandfonline.com/doi/full/10.1080/01942630802574908`.

[11] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2018. URL `https://arxiv.org/abs/1812.08008`.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, jun 2002. doi: 10.1613/jair.953. URL `https://doi.org/10.1613%2Fjair.953`.

[13] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013. doi: 10.1109/CVPR.2013.391.

[14] H. A. Division and the Chief Medical Officer's Division of the Department of Health. Estimating prevalence of autism spectrum disorders (asd) in the irish population: A review of data sources and epidemiological studies, Nov 2018. URL `https://assets.gov.ie/10707/ce1ca48714424c0ba4bb4c0ae2e510b2.pdf`.

[15] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description, 2014. URL `https://arxiv.org/abs/1411.4389`.

[16] D. L. Gallahue. *Understanding Motor Development*. McGraw-Hill, 2011.

[17] A. Gandotra, E. Kotyuk, A. Szekely, K. Kasos, L. Csirmaz, and R. Cserjesi. Fundamental movement skills in children with autism spectrum disorder: A systematic review. *Research in Autism Spectrum Disorders*, 78:101632, 2020. ISSN

1750-9467. doi: https://doi.org/10.1016/j.rasd.2020.101632. URL https://www.sciencedirect.com/science/article/pii/S1750946720301227.

[18] gdpr.eu. Art. 8 GDPR – Conditions applicable to child's consent in relation to information society services, Aug 2020. URL https://gdpr.eu/article-8-childs-consent/.

[19] K. R. Heineman, P. Schendelaar, E. R. Van den Heuvel, and M. Hadders-Algra. Motor development in infancy is related to cognitive function at 4 years of age. *Developmental Medicine & Child Neurology*, 60(11):1149–1155, Nov. 2018. ISSN 00121622. doi: 10.1111/dmcn.13761. URL https://onlinelibrary.wiley.com/doi/10.1111/dmcn.13761.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[21] C.-D. Huang, C.-Y. Wang, and J.-C. Wang. Human action recognition system for elderly and children care using three stream convnet. In *2015 International Conference on Orange Technologies (ICOT)*, pages 5–9, 2015. doi: 10.1109/ICOT.2015.7498476.

[22] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, June 1973. ISSN 0031-5117, 1532-5962. doi: 10.3758/BF03212378. URL http://link.springer.com/10.3758/BF03212378.

[23] Y. Kong and Y. Fu. Human action recognition and prediction: A survey. *CoRR*, abs/1806.11230, 2018. URL http://arxiv.org/abs/1806.11230.

[24] I. Laptev and T. Lindeberg. On space-time interest points. 02 2005.

[25] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

[26] W. Lin, M.-T. Sun, R. Poovendran, and Z. Zhang. Human activity recognition for video surveillance. *2008 IEEE International Symposium on Circuits and Systems*, pages 2737–2740, 2008.

[27] X. Liu, Y.-m. Cheung, M. Li, and H. Liu. A lip contour extraction method using localized active contour model with automatic parameter selection. In *Proceedings*

of the 2010 20th International Conference on Pattern Recognition, ICPR '10, page 4332–4335, USA, 2010. IEEE Computer Society. ISBN 9780769541099. doi: 10.1109/ICPR.2010.1053. URL `https://doi.org/10.1109/ICPR.2010.1053`.

[28] M. Lu, Y. Hu, and X. Lu. Driver action recognition using deformable and dilated faster r-cnn with optimized region proposals - applied intelligence, Dec 2019. URL `https://link.springer.com/article/10.1007/s10489-019-01603-4`.

[29] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann. Mediapipe: A framework for building perception pipelines, 2019. URL `https://arxiv.org/abs/1906.08172`.

[30] N. Ma, Z. Wu, Y.-m. Cheung, Y. Guo, Y. Gao, J. Li, and B. Jiang. A survey of human action recognition and posture prediction. *Tsinghua Science and Technology*, 27(6):973–1001, 2022. doi: 10.26599/TST.2021.9010068.

[31] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, Dec. 1943. ISSN 0007-4985, 1522-9602. doi: 10.1007/BF02478259. URL `http://link.springer.com/10.1007/BF02478259`.

[32] R. Poppe. A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976–990, jun 2010. ISSN 0262-8856. doi: 10.1016/j.imavis.2009.11.014. URL `https://doi.org/10.1016/j.imavis.2009.11.014`.

[33] I. Rodomagoulakis, N. Kardaris, V. Pitsikalis, E. Mavroudi, A. Katsamanis, A. Tsiami, and P. Maragos. Multimodal human action recognition in assistive human-robot interaction. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2702–2706, 2016. doi: 10.1109/ICASSP.2016.7472168.

[34] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[35] M. Singh, A. Basu, and M. K. Mandal. Human activity recognition based on silhouette directionality. *IEEE Trans. Cir. and Sys. for Video Technol.*, 18(9): 1280–1292, sep 2008. ISSN 1051-8215. doi: 10.1109/TCSVT.2008.928888. URL `https://doi.org/10.1109/TCSVT.2008.928888`.

[36] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659, 2013. URL `http://arxiv.org/abs/1312.4659`.

[37] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008. doi: 10.1109/TCSVT.2008.2005594.

[38] A. Turarova, A. Zhanatkyzy, Z. Telisheva, A. Sabyrov, and A. Sandygulova. Child action recognition in rgb and rgb-d data. HRI '20, page 491–492, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370578. doi: 10.1145/3371382.3378391. URL `https://doi-org.elib.tcd.ie/10.1145/3371382.3378391`.

[39] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. doi: 10.1109/cvpr.2015.7299059. URL `https://doi.org/10.1109%2Fcvpr.2015.7299059`.

[40] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou. Temporal pyramid network for action recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–597, 2020. doi: 10.1109/CVPR42600.2020.00067.

[41] Z. Zhang, X. Ma, R. Song, X. Rong, X. Tian, G. Tian, and Y. Li. Deep learning based human action recognition: A survey. In *2017 Chinese Automation Congress (CAC)*, pages 3780–3785, 2017. doi: 10.1109/CAC.2017.8243438.