Human Action Recognition - Intelligent Models for Classifying Actions By Children

Tom Mathew Thomas, Bachelor of Technology

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: Inmaculada Arnedillo-Sanchez

August 2022

Declaration

I hereby declare that this is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

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`.

Tom Mathew Thomas

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.

Tom Mathew Thomas

August 19, 2022

Human Action Recognition - Intelligent Models for Classifying Actions By Children

Tom Mathew Thomas, Master of Science in Computer Science

University of Dublin, Trinity College, 2022

Supervisor: Inmaculada Arnedillo-Sanchez

The main purpose of this paper is to develop individual models that can predict actions such as Stand on one leg, Stand on Tiptoe, imitating pre-defined custom poses and Run-Pick. Which are performed by children of different age groups, along with this the time taken for performing the action is also predicted. The input data consist of unlabelled videos in which children are performing the mentioned action. Based on analysing various similar models, a system design is formulated which mainly consist of feature selection and model selection. Various methods such as Convolutional neural nets, Media Pipe library for feature selection are evaluated and based on their compatibility with the existing set of inputs MediaPipe is selected. The paper also discusses methods through which annotation of raw video inputs can be converted into information that is suitable for model training. Custom heuristics and video normalization techniques have been discussed that can be applied for each action and corresponding label set is generated. Based on this information various ML models such as Deep Neural Network, Random Forest are trained, and their performance is evaluated. For time dependent action, as Deep Neural Networks fails a Recurrent Neural Network using LSTM cells are implemented. Generated models are further analysed and optimizations that can be incorporated with in the future iteration are mentioned.

Keywords: Gesture Detection, Deep Neural Networks, Random Forest, Recurrent Neural Network, Data Annotation, Normalization techniques, LSTM, Convolutional Neural Networks, MediaPipe.

Acknowledgments

Going through this phase of dissertation, which was arduous, there were a lot of people with whose help I would not have been able to complete this work. First and foremost, I extend my heartfelt gratitude to God to bless me with this opportunity. I am extremely grateful to my parents,Thomas and Elsy. Without their blessings and prayer nothing would have been possible. I would also like to thank my sisters and their families for their immense support.

I would also like to appreciate Akhil, Azin, JP, John, Shybu,Snamaria and Unni, my roommates, for their help when I got stuck. A special thanks to Arun, Azin, Manu, and Sherin, my friends. They were always eager to contribute new ideas when needed.

Finally, and most importantly, I am grateful to Prof. Immaculada, my supervisor, for her unending support. I also thank everyone of my family, friends, and well-wishers for their support.

Thank you

Tom Mathew Thomas

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Gestures or Actions can be considered as a simple means of communication. Over the span of a few years various research have been conducted over Human Action Recognition Models, which have evolved from detecting blinking of an eye towards more complex actions used in sign languages (1). Complexity of these recognition systems lies over various factors such as object detection/recognition, object tracking and custom model training. All these phases must be compatible with one another inorder to build better action recognition models. As time passes and due to technological advancement, the domain for action recognition systems has been expanded to health care, surveillance and other fitness applications.

The same action will be performed in a different manner (temporal difference) by different people which further enhances its complexity. As a result the models must be able to generalise actions performed by various people, that is to identify proper features. Numerous studies are conducted on the feature selections that can be applied over the Gesture Detection Model (2). In a nutshell Action Recognition models are quite complex models, which are undergoing major transformations and their domain is huge.

As a masters student, for me conducting research on such a complex system is interesting as well as challenging. In my experience, I have generated models over data that have been pre-processed. Whereas processing raw data into useful information was a phenomenal experience which I have never encountered before.

1.2 Problem Statement

A project that has been initiated by Prof. Immaculada and Mr. Benoit, is the analysis of the motor skills exhibited by childrens across different age groups. Based on this study, they are able to identify children who lacks motor skills corresponding to their age group. For this project, in the initial phases they have determined various actions/ gestures that can be used for the analysis along with various criterias. Using Kinet development kit, videos performed by the childrens are evaluated based on the custom defined heuristics for each action. As the Kinet development kit can be a hardware dependency and the heuristic defined may hinter the future scalability of the project. As a result instead of these two, gesture detection models that can classify or evaluate the actions performed from a camera feed can be considered as a suitable replacement.

The problem statement, that I have been assigned was to develop individual machine learning models that can identify gestures mentioned below:

- Stand on one leg - The child will be standing on one leg either left or right, in certain cases both. The model has to identify the gesture and will also be able to specify the time that the child has performed those actions.
- Stand on Tiptoe - In this case the child is asked to stand on tiptoe and the objective is to measure the time, the child has performed the tiptoe action.
- Run and Pick - In this scenario, the child is asked to run forward to a particular point, and pretend to pick something from the floor and has to run back to his starting point. The model has to classify whether the set of actions have been performed and the time taken for achieving this.
- Imitate - Certain predefined poses are determined as shown in fig 1.1, either on a screen or a person will be showing a particular pose from the set mentioned in fig 1.1 and the child is asked to imitate the posture.

Figure 1.1: Imitate Action

One of the common factors in all this action is the calculation of time (Imitate is an exception), based on the action performed and time taken are being considered inorder to analyse his/her motor skills against their age group.

1.3 Dissertation Layout

The entire dissertation is subdivided into 4 chapters, where Chapter 2 focuses on the existing methodologies that have been carried out in various fields of human action recognition. Based on these studies we identify and analyse various sections within the detection models such as feature selection and its importance. It also discusses the various methods through which these sections can be effectively applied.

In chapter 3, focus is mainly on the overall design that has been applied over the project. The challenges that have been faced during the development phases and various methodologies that have been considered in order to overcome the same.

Chapter 4 mainly shows the various model algorithms that have been used and its performance. A brief explanation of the performance of the model against the test data is also included within this chapter.

Final Chapter 5, summarises the work so far done and points out the limitations of the generated models. It also comprises the further works that can be carried out over the existing models in order to improve its performance.

Chapter 2

State of the Art

It's been a decade since various approaches were implemented in order to build human action recognition systems. The versatility of different beings performing the same set of actions leads to its challenging state, due to this the same set of actions may have different timings and temporal variations. For Human Action Recognition(HAR), it contains very complex cases that include body motion and interaction with surroundings, therefore a single approach cannot be applied for the entire set of actions. This chapter mainly focuses on the generic approaches that have been introduced within the HAR detection system and models that have been developed for classifying the same.

2.1 Related Works

Numerous research and works have been carried out in human action recognition system, the works that are very much similar to our objective is 'Deep neural networks for video analysis of human pose estimation' (3),'Hierarchuical Randon Forest For Senior Action Recognition In Videos' (4). After analysing the methodologies followed with in the above mentioned research and few other relatable once like (5)(6)(7) . We could find that even if the algorithms or model structure are different but the over all process remains the same. Therefore the generic approach identified from these works have been detailed in section 2.2 and various methods through which these can be implemented are detailed in section 2.3 and section 2.4 .

2.2 Standard Approach

Numerous research has been conducted for developing Human Action Recognition models. After detailed analysis, the basis workflow pattern observed is as shown in fig 2.1.In which

the major states are feature extraction and model creation.

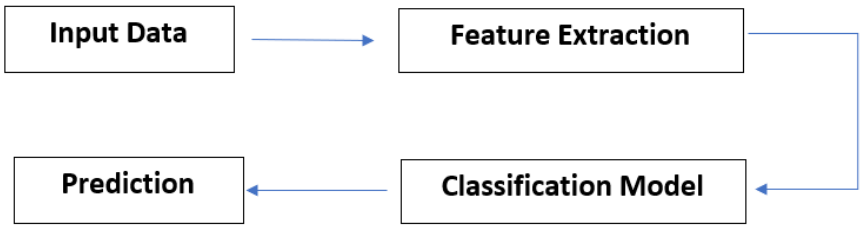

Figure 2.1: Standard Workflow for Action Detection

2.3 Feature Extraction

Feature extraction is a key process in which concrete features are extracted or identified from the images which are essential for gesture identifications, which are later on used as the input for the model. In case of Action recognition the joint locations such as elbow, knee, foot etc.. are considered as the features for the model. Feature extraction can be carried out either by building convolution neural networks (CNN), modifying the existing CNN and its weights (Transfer Learning) or by using pre-trained libraries for extracting the same.

2.3.1 Feature Detection using CNN

Convolutional neural networks is a deep learning algorithm that mainly processes images and tries to identify the spatial and temporal features within an image. Using filters CNN reduces pixel idensed image into a simpler form that can be easily processed. Convolutionary kernels, pooling and striding will reduce the dimensionality of the image. One of the most prominent CNN architecture for image processing is AlexNet (8) shown in Fig2.2

As shown in the Fig2.2, it consists of 4 sections each consisting of a convolution, pooling and normalisation layer. Each sector within the Alexnet focuses on estimating a feature such as background elimination, identifying the region of interest and so on. A very deep network, with a large number of filters are needed inorder to extract the necessary features from the image. For finding the optimum values for the kernel used in the convolution layers, a large amount of training is required. Therefore a major limitation in this case is that a large amount of training set is required for building a CNN model for feature extraction from scratch. Whereas in Transfer learning we are using existing

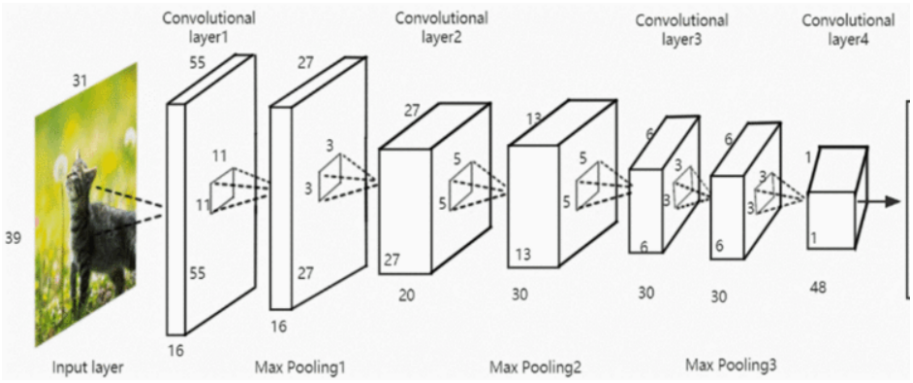

Figure 2.2: Standard Object Detection CNN - Alexnet

trained CNN model and modifying the end layers of this model in order to satisfy the required objective.

2.3.2 Feature Detection using Pre-trained libraries

The primary objective of this project is to build models for gesture recognition, and therefore the necessary feature points can be considered as the the joint locations/coordinates within the video. Two of the most prominent open-source libraries that are able to detect these landmarks are OpenPoseAI and MedaiPipe (9).

MediaPipe

In 2019 google open-sourced their framework MediaPipe, which can be used to create ML pipeline for processing videos and other time series data. This is a lightweight framework that can be used in embedded devices such as raspberry pi. The framework is available in multiple languages such as python,c++, android, javascript.

MediaPipe solutions consist of pre-trained models using tensorflow and tensorflow lite. Some of the most suitable solutions for action recognition are

- MediaPipe face detection - Capable of detecting multiple faces within a given frame and provides 6 feature points such as eyes(2) , mouth(1), nose(1),ears(2) for each face identified within the frame.
- MediaPipe Face Mesh - An advanced version of face detection, which is capable of identifying multiple faces and identifies 438 3D landmarks within the detected faces. This solution is mainly used for sentimental analysis.

- MediaPipe Hands - A complex model for tracking the hand movements. With the ML model it detects the hand positions and further classifies 21 feature points on it.

- MediaPipe Pose - It is a ML solution for pose tracking within a frame. The solution estimates the 3D coordinates of 33 joints/ features within the body. Landmark points representing various joints are illustrated in fig 2.3 Based on the provided joint locations and it's transition over time further models can be developed for applications such as full-body pose analysis, sign language detection and so on.

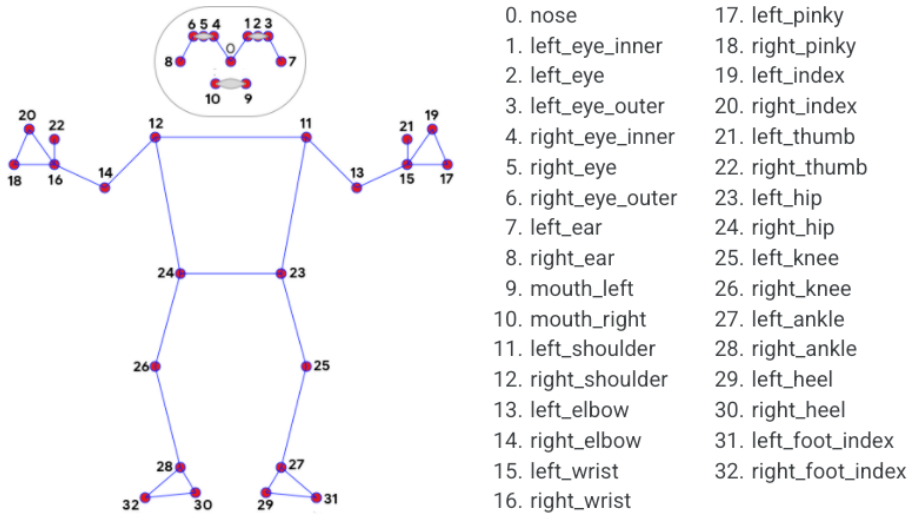

Figure 2.3: Landmark Points of MediaPipe Pose (10)

OpenPose AI

It is a real time library capable of multi pose detections. Similar to media-pipe pose solution, open pose detects a total of 13 joints. OpenPose library is capable of both 2D and 3D coordinate positioning of features. The basic building block of this library is CNN networks training over large set of data consisting of peoples of varying age groups. Various applications for pose estimation, real life application in sports and fitness have been developed using OpenPose.

2.4 Models for Classification

An action can be described as the collection of time series of 3D positional coordinates of joints and the variations within these joints describes the actions. The state of art models for action recognition revolves around the complexity of the action performed. When it

comes to generating models mainly there are two kinds of approaches either supervised learning models or unsupervised learning models (11).

2.4.1 Unsupervised Learning Models

The main objective behind these models is to self identify the features and learn all by itself without any human intervention. As we are not teaching the model, it doesn't need any sort of labelled training data. Clustering can be considered a prominent method for unsupervised learning, where the user defines the number of clusters and the model tries to generate clusters based on the existing data. One of the most prominent clustering algorithms is K means clustering .

K means Clustering

In K means clustering, we are analysing the unknown data and trying to generate a specific number of clusters, that follows a similar pattern. K means clustering can be considered as a simple and novel approach for action detection and much research has been carried out within this domain (12) (5). For example, if we are plotting the hand's positional coordinates, when performing hands up and down action. Then if we generate 2 clusters on this data. The clusters generated will be representing these 2 actions. For this first we are assuming two cluster centroids as random, and on further iterations we are optimising the centroid positions by comparing its distance against input points and grouping each input into one of these clusters. And for evaluating the test/new data, it's positional distance is compared with the obtained centroid position and it's assigned to the centroid which tends to be the closest.

This approach will be most suitable if we are having a large set of unlabelled data, which has clear separation as labelling of data can be very complex.

2.4.2 Supervised Learning Models

In this kind of models we are labelling the inputs and the models are generated inorder to predict these labels. Various kinds of algorithms can be used within these methods, the basic algorithms are logistic regression, SVM classifications, Random Forest and deep neural networks. For action recognition, various models are generated using random forest and deep neural networks. Whereas for evaluating complex actions which contain variation in time series of positional coordinates Recurrent Neural Networks are implemented.

Random Forest

Random Forest can be considered as an ensemble of decision trees. As shown in fig 2.4

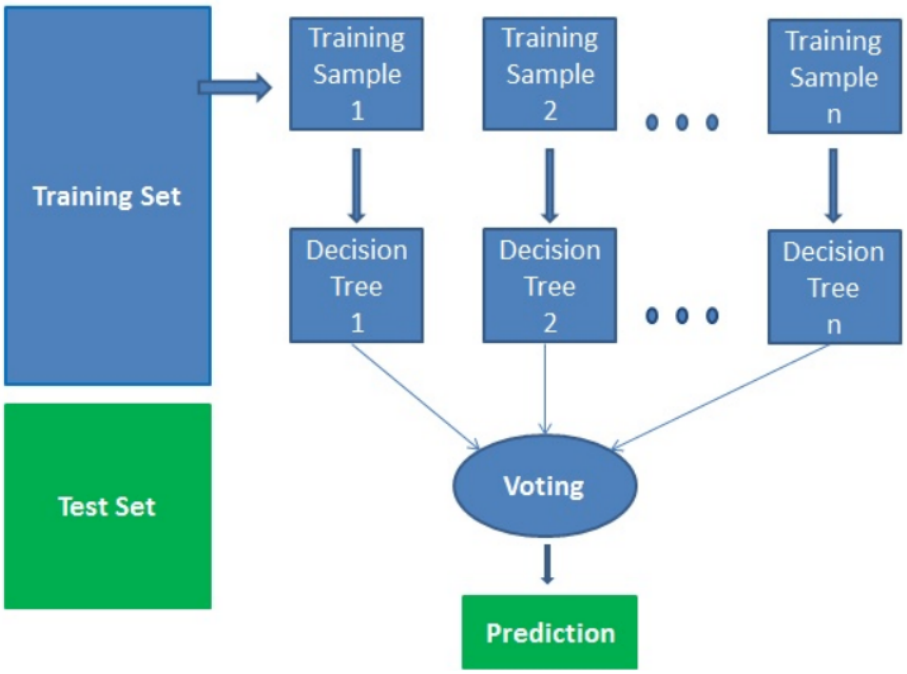

Figure 2.4: Basic Overview - Random Forest (13)

Based on the training set, numerous decision trees are constructed over different sets of features and training size. In decision trees, we are making intermediate decisions within each layer based on certain features and by identifying its corresponding threshold values. The selection of the best feature within each layer is derived based on their gini-index value. This method is iterated over many times and the final node predicts the output of the classification.

As we are using the input as such for predictions, decision trees tend to overfit over the training data. Therefore in order to limit the over-fitting we are generating a number of decision trees built on various sets of features and the overall output is derived from evaluating (fitting function) these sets of outputs.

Deep Neural Network (DNN)

Deep neural network is a type of artificial neural network (3), where the number of hidden layers is more than one. As shown in the fig 2.5, a DNN consists of mainly 3 layers, input layers, hidden layers and output layers. In the Input layer the multidimensional vectors are flattened and fed into the model. The main objective of hidden layers is to derive

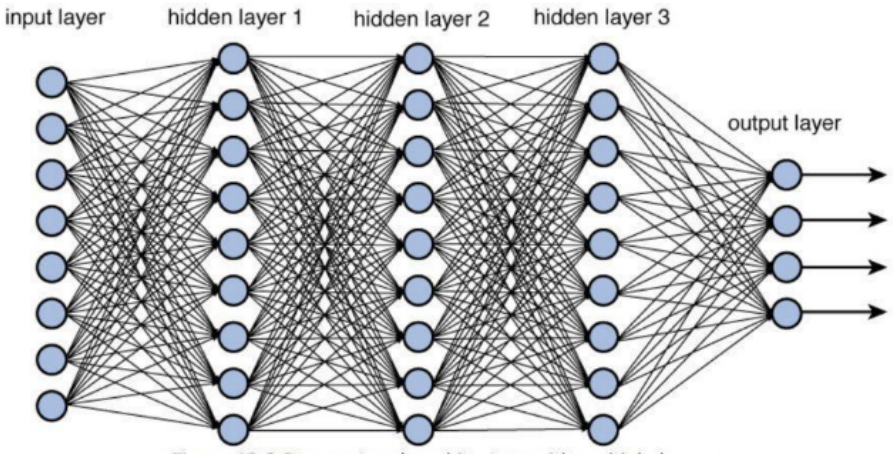

Figure 2.5: Deep Neural Network With Multiple Hidden Layer

various features from the model. And as the total number of hidden layers increase, it implies we are trying to derive more features based on the output from the preceding layer. The output layer focuses on the model output. If the objective of the model is to predict 3 labels/classes then the output layer will consist of 3 nodes each representing the probability of that particular action being performed, based on the given feature vector. During the training phase the model will try to increase the prediction accuracy by optimising weights and bias values for each node within the hidden layers and output layers.

Recurrent Neural Networks

In a deep neural network, we predict the output based on the set of features. But there exist, instances where there are relationships that are temporal in nature. Therefore in order to define this time-series problems Recurrent Neural Networks are used. RNN can be categorised as an advanced version of DNN, where we are preserving the previous outputs or features within the model and then by evaluating the current features along with previous outcomes the model output is predicted. The basic illustration of RNN is shown in fig 2.6

In RNN, we are keeping track of the previous output and features based on a predefined window. One of the main problems encountered in RNN is that as we are trying to persist a large number of features or intermediate outcomes from previous frames (large window size), the model has to keep track of all the features. Which may lead to vanishing or exploding gradients and will negatively affect the effect of previous outcomes over the current prediction. There are certain variants in RNN that have been developed inorder to deal with this vanishing and exploding gradients and one such variant is LSTM (15).

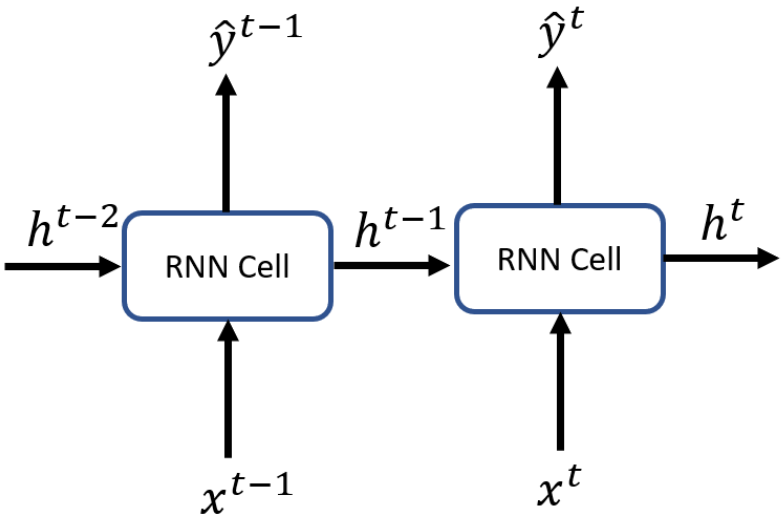

Figure 2.6: RNN Cell - Basic Structure (14)

Long Short Term Memory (LSTM)

Rather than keeping track of all intermediate outputs and features within a window size, LSTM cells try to identify the key features that influence the future predictions and will key these highly correlated features within the memory models. As a result the exploding and vanishing gradient effect can be neutralised. An image of an LSTM cell is shown in fig 2.7. LSTM uses various gates in order to update and to keep track of various features.

- Forget Gate - The main purpose of this gate is to forget unnecessary features and keep track of the essential one within the model.
- Input Gate - The two main functions of this gate is to update and keep track of data within memory of the model and also determines whether the current output or feature identified will be useful for future prediction, that is whether to keep these values within memory of the model.
- Output Gate - Determine the final output of the current input/state. By evaluating the forget gate, input gate and current feature set the output gate determines the prediction.

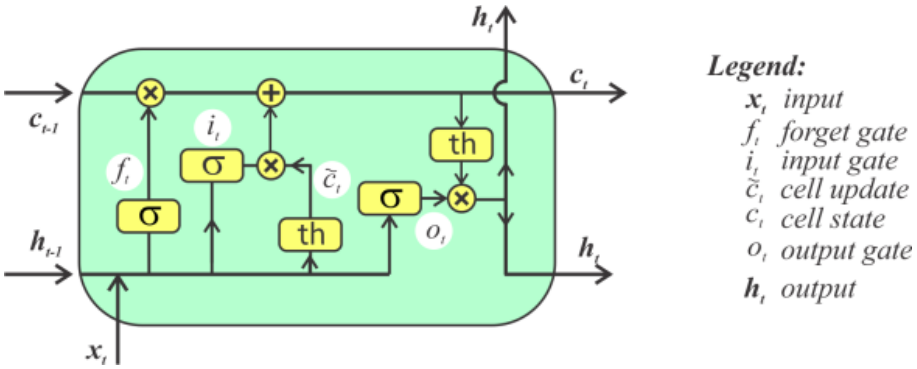

Figure 2.7: LSTM Cell (15)

2.5 Summary

A study of various research and methodologies undertaken for human action recognition has been carried out. Based on this, a generic approach followed in most of the referred projects has been formulated, which mainly consist of feature extraction and various models for classification. The most common methods for feature extractions are CNN and predefined libraries like MediaPipe. Whereas in case of models, various types such as supervised and unsupervised have been clearly explained. With this we are able to have a clear picture of various methodologies that have been implemented for Human Action Recognition.

Chapter 3

Methodology

Human action detection models are very complex, whereas actions performed by children's of various age groups are even complex. Because as the performers are children's, the way in which they perform the specified action differs in a large manner. As the main objective for my work revolves around building models for evaluating the actions performed by children. Detecting or classification of action is the initial phase and further analysis like time taken for performing is considered. In this chapter we will be discussing the various approaches that have been taken into picture for developing such a robust model that is able to predict a particular action by children's of varying age groups. It also discussed the data that have been collected and the various transformations, normalisation and labelling techniques that have been applied over this dataset in order to convert the raw data into useful information for the models.

3.1 Dataset

As mentioned in section 1.2. As part of the existing project, numerous videos of childrens from various age group and school, performing the mentioned actions have been already collected. A detailed information regarding the total number of videos across various actions is tabulated in table 3.1.

Action	Level 0	Level 1	Level 2	Total
Stand on One Leg	164	163	163	490
Imitate	162	162	161	485
Stand On Tiptoe	162	160	159	481
Run and Pick	167	167	164	488

Table 3.1: Total Number of Videos Available Per Action

Different levels within the action specifies the complexity in performing the action. For example, for the action stand on one leg, in level 0 the child is asked to perform the action and hold the posture that is stand on one leg for ‘n‘ seconds and as the level increases the time required to perform increases to ‘m‘ where m greater than n. As the main purpose is to determine the motor skills of the child according to his/her age.
As we are handling data regarding to childrens, security and privacy is a major concern. Therefore in order to protect the personal information from leaking as an initial step the videos has been anonymised, that is faces have been blurred within the video.

3.2 General Workflow

Based on the existing system for gesture detection as mentioned in chapter 2, an additional layer termed as output buffer has been added. Which will be able to carry out project specific objectives such as time for performing the action along with gesture detection. The generic workflow of the project is shown with in fig 3.1.

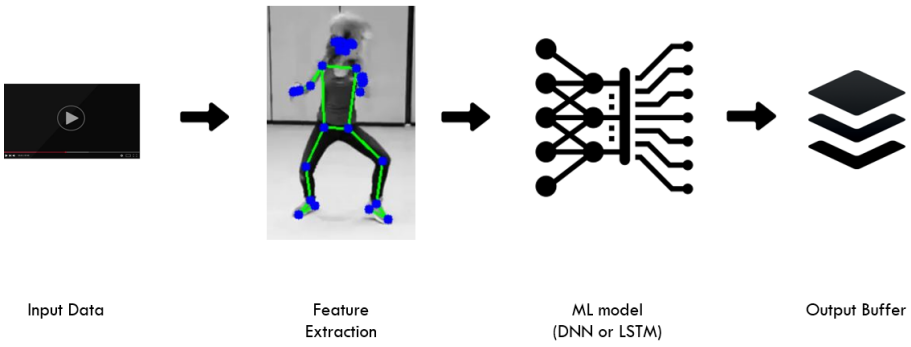

Figure 3.1: High Level Design

It mainly consists of 4 phases, where the function of feature extraction and model creation is detailed in section 2.3 and section 2.4. The other two phases are.

- Input Layer/ Video Feed : In this case the videos of childrens performing the actions are considered as inputs. Therefore within the Input Layer, videos are processed and frame by frame extraction is carried out and fed to the feature extraction unit.
- Output Buffer : This layer is an add on to the generic approach of gesture detection inorder to evaluate the time taken for action. As the name suggests it consists of a buffer which will be temporarily storing the output, and in real time the buffer is evaluated to find the time taken for performing that specific action. For example consider a video of ‘n’ frames has been evaluated, and the output from the model (S - Standing, L - Left Up, R - Right Up) has been stored in the buffer as shown in

the fig. 3.2

S	S	L	L	L	S	S	R	R	R
1	2	3	4	5	6	7	8	9	10

Figure 3.2: Sample Output Buffer

By analysing the buffer we can see that, the number of frames the left leg went up was from 3 to 5 and right leg was from 8 to 10. Based on this frame count we can estimate the time the child has performed that specific part of the gesture. A more detailed explanation regarding the measure of time for each action is explained in chapter 4, section 4.1.

3.3 Feature Detection - Media pipe Solutions

Within feature extraction we are trying to identify the features that can be used inorder to classify actions from each frame. As in our case the feature points can be considered as the joint locations of the children identified within the frame. Various methodologies in order to identify the features are already mentioned in section 2.3

3.3.1 Convolutional Neural Networks Vs Pre-Trained Libraries

CNN's can be considered as the classical approach for the project, which will also reduce the dependencies with other libraries or models. But in this case the amount of data available that can be used to train the CNN's are fairly low. Proper identification and extraction of features is not possible. Whereas in the case of libraries they are already trained over millions of images or videos and are able to detect people and their corresponding joints.

3.3.2 MediaPipe Vs Other libraries

In terms of performance, the overall detections and tracking of person and their corresponding joints tends to be similar. But as mentioned earlier media-pipe is a library that can be operated over less computational resources such as mobile,raspberry pie. Therefore, considering the scalability of the project and future targeted systems, mediapipe tends to be a better option. Moreover mediapipe libraries are readily available as python

packages and can be easily integrated into the development environment. But other libraries such as OpenAI tend to exist as an API and frequent API calls have to be initiated for the feature extraction, which further increases the dependencies with in the project.

3.3.3 MediaPipe

In MediaPipe the faster identification and localization of features are happening as they are using an entirely new approach to human body pose perception called BlazePose (16). As a result the generated model is lightweight and can easily have 33 landmark/features (fig 2.3) from the single frame as shown in fig 3.3

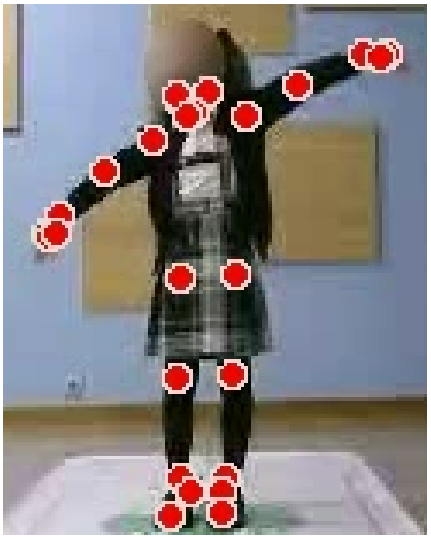

Figure 3.3: Sample - Feature Extraction, MediaPipe

As this mediapipe uses opencv libraries for image processing, the coordinated system followed by the package is as shown in fig 3.4 , that is considering a 7x7 pixel it's (0,0) located to top-left and (7,7) lies in the bottom-right corner.

Based on all these factors mediapipe pose solution tends to be the best fit for feature extraction for this scenario. But when we have implemented this over the videos of the children, one of the key observations was that in certain cases, mostly when the movement is undergoing, there are a few miss-classifications in tracking the features points. As a result, inferences drawn based on a single frame will not be accurate. Therefore for real time analysis we will be considering a group of 5 frames rather than a single frame. Only drawback in this approach is that as we are measuring time (measured based on number of frames) for most of the actions, there will be an error margin of 2 frames.

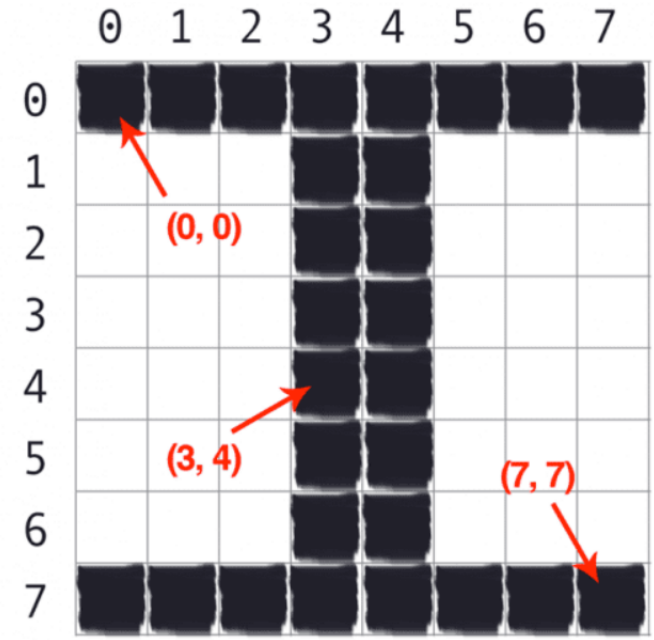

Figure 3.4: MediaPipe - Coordinate System

3.4 Challenges

Following are the major challenges faced during the development phase of the project.

- As the clarity of the videos are not high and in case of actions performed by smaller kids, the media pipe fails to extract or detect joint locations with 100% accuracy. As a result few frames within the videos have to be skipped as feature extraction fails. The total percentage of skipped frames are displayed in the table3.2

Action	% of frames skipped
Stand on one leg	$\tilde{5}$%
Imitate	$\tilde{4}$%
Stand on tip toe	$\tilde{5}$%
Run and Pick	30%

Table 3.2: Frame Skipped vs Action

- As we have already mentioned, the dataset consists of videos of childrens from various schools. In few of the schools, the uniform or the dress code such as long skirts tends to cover their joints and as a result there movement cannot be properly tracked.

- As we are dealing with the data corresponding to childrens, proper security and privacy measures have to be maintained.

- Inputs have to be normalised inorder to build effective models. Video normalizations are a challenging process and as we are dealing with a large set of data, manual normalisation was not possible. Therefore an effective way to automate this process has to be implemented.

- One of the important phases of data preparation for models is generating labels for each input. In this case each frame is acting as an input for the model, therefore an automated process for labelling each must be generated.

- Occlusion is a major problem faced with object tracking, in this case when considering run and pick due to occlusion labelling of left and right legs gets interchanged, as a result the z-coordinate representing depth does not show any trend and cannot be used as an insightful feature.

- In Run and Pick, the child moves very close to the camera and moves out of the camera's field of view. As shown in fig 3.5

Figure 3.5: Run and Pick - Feature Extraction Failure

3.5 Normalization

The obtained 3D coordinates of joints from mediapipe is of normalised form that is in the range of 0 to 1. But one key observation that was found is that even though the values are normalised, the starting points of each child tends to be different. As the positional coordinates are the key input for the model and classification is carried out for each frame, a positional normalisation has been introduced in order to offset inputs of each video and a custom normalisation function has been implemented for the same.

3.5.1 Custom Normalization Function

The starting point or position of childrens may differ in each video therefore there will be variation in positional coordinates which will reduce the performance of the model. For example consider two children A and B performing a particular action in 1D, the initial position of A was at 0.4 and and B was at 0.6, coordinates of A varies from 0.4 - 0.6 and B from 0.6 - 0.8, therefore in this scenario, by referring the initial positions an offset value is calculated for each person such that the initial position starts at 0.5, for A it is 0.1 and for B it is -0.1. This derived offset is computed across all frames within that particular video therefore the variation of A and B after normalisation is 0.5 - 0.7.
In this case the left hip point will be taken as the reference and based on the initial frame, an 2D offset is derived such that the left hip resides in the centre of the frame (0.5,0.5) and this value is added to all the points that have been derived from the entire frames within the video.

3.6 Data Labelling

Models can be of either supervised or unsupervised learning models, in this case the supervised learning models are used for classifications the main reason behind this decision is described in section 3.8.
For supervised learning models need labelled inputs for training the model, as each frame can be considered as in input for the model, manual labelling for the entire set of frames is not possible. Therefore a set of heuristics is formulated for each action based on the positions of joints within each frame and labelling is carried out. Numerous heuristics can be formulated for a set of action, based on the manually labelled set of data the formulated output of heuristics are compared and the best heuristic is determined. The formulated heuristic with best result for each action are mentioned below. The joint points mentioned in the below heuristics are shown with in fig 2.3.

3.6.1 Labelling - Stand on one Leg

For this particular action the heuristics is formulated based on the positions of left hip point [p-24], left knee [p - 26], right hip point [p-23], right knee [p-25]. Instead of comparing the positional coordinates as such, in order to minimise the effect of various heights of childrens, we are evaluating the difference of knee and hip pixel positions and variation of this value across the frame are analysed as shown in fig 3.6. Based on this a threshold value is formulated and labelling is carried out as follows.

$$abs([p-23]-[p-25]) < threshold :: right\_leg\_up$$

$$abs([p-24]-[p-26]) < threshold :: left\_leg\_up$$

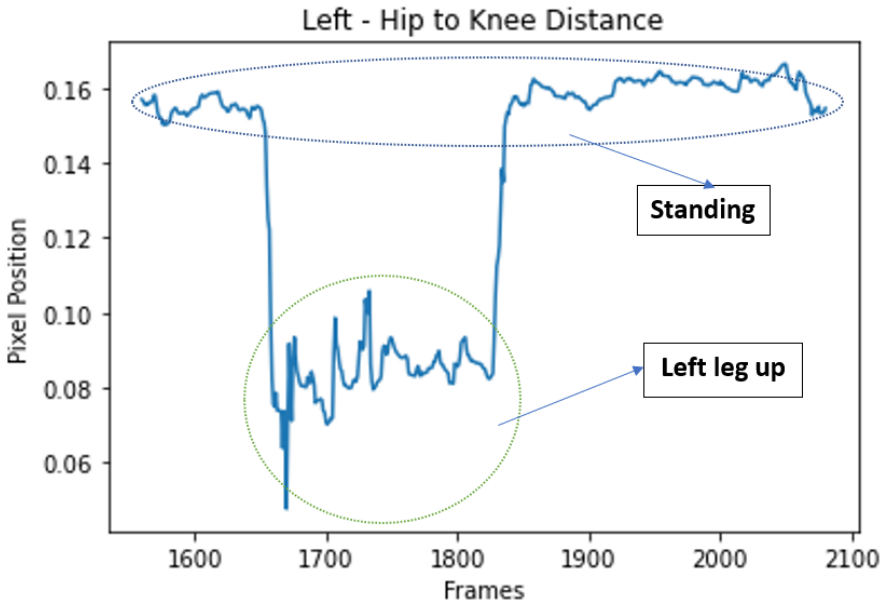

Figure 3.6: Stand On Leg - Observed Pattern

Identifying Optimum threshold Value

For each video the minimum and maximum difference between the hip and knee are stored in respective arrays min_array and max_array. The minimum value within max_array corresponds to small children in terms of height. Within the min_array values are close to zeros, the closer to zero denotes that the child had raised that corresponding leg higher or in other terms higher the perfection of action performed. Therefore the initial threshold

is formulated as the average of the above mentioned values,

$$Threshold = (min(max\_array) - min(min\_array))/2$$

Based on this initial_threshold the labelling is carried out, and the result has been cross checked across the known data set. As the number of known dataset is very few, further optimisation of the threshold value is neglected. The final observed threshold value is 0.127.

Elimination of Outliers

It was observed that in certain videos, the children are not performing the desired action, as a result a filtration has to be performed. Otherwise it will result in outliers for deriving the heuristics. Therefore for each video the minimum and maximum variation of the points are compared and if the difference is less than 0.075, those sets of videos are eliminated from the training samples. As the child is not performing any particular action in those sets. The 0.075 is obtained using trial and error, in theory it must be zero if the action is not performed.

3.6.2 Labelling - Imitate

The heuristics of imitate can be considered as a fairly simple one. The angle form between elbow - shoulder - hip is calculated for both left and right arms. Then based on the derived angle of both the arms, corresponding pose is labelled. As there is large variation in the action performed by the children we have defined a range of angles rather than a fixed one. The mapping of pose corresponding to angles formulated between the hands are shown in table3.3.As we are only considering the data points that satisfy the mentioned scenario within the table. Automatically the outliers are getting eliminated from the training data set.

Label	Left Angle	Right Angle
Pose 0	80-100	80-100
Pose 1	35-55	35-55
Pose 2	125-145	35-55
Pose 3	35-55	125-145
Pose 4	155 - 180	155 180
Pose 5	80-100	35-55
Pose 6	35-55	80-100

Table 3.3: Criteria For Imitate

3.6.3 Labelling - Standing on Tip-Toe

For actions like stand on tip toe two sets of labelling is carried out, labelling for single frame and labelling form RNN models.

Frame Wise Labelling

For this heuristics, we are supposed to track the movement of angle of both legs and based on its movement labelling has to be done, as shown in fig 3.7 . But in reality, due to the positioning and dress (socks) of the children, the media pipe is not able to detect its movement in an accurate manner for kids in smaller age groups. Therefore as a workaround that satisfies all set of data we are tracking, the knee movement and positioning of the feet throughout the video and labelling is carried out. The ground point of the feet and its corresponding knee position is calculated by analysing the entire frames with in the video. Then by considering below mentioned constraints labelling is carried out.

- Constraint 1: Checking whether the foot is placed on the ground itself. For this, the difference in value of toe for that specific frame and ground point of toe is less than 0.075, then it implies that the kid is not raising his/her foot of the ground.

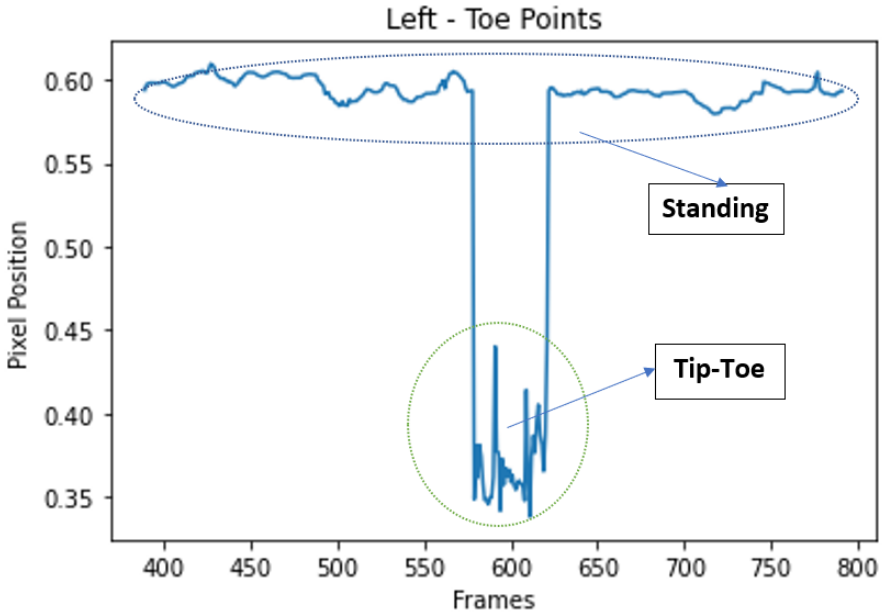

Figure 3.7: Stand On Tiptoe- Observed Pattern

- Constraint 2: Analysing the movement of knee point, if the difference between the ground point for knee and position of knee for that particular frame is greater than

0.13 . Which implies that there have been a variation in elevation.

If both this constraint are satisfied for a particular frame, then we will be labeling that particular frame as 'tip-toe' otherwise the frame is labelled as 'standing'.

Labelling for Recurrent Neural Network

Inorder to train the RNN model, multiple frames have to be processed at the same time, which is already discussed in section 2.4.2. For this action the chunk of frames has been determined as 20. As we are tracking 8 joints (x and y coordinates), therefore each sample is a 20x8x2 vector and labelling has been done for this training data.

Optimum Frame Length For Recurrent Network

The objective of this RNN model is to predict the transitional phase within the action, that is standing to tip-toes and tip-toes to standing. By analysing the existing data the average frames for transition lies between 6-10. As mentioned in section 3.3.3, 5 consecutive frames must be evaluated together to finalise a predicted output. Therefore for each transitional state the head and tail parts of the sequence must be the corresponding action that is standing and tip-toe or tip-toe and standing each consisting of 5 frames. Thus 20 is selected as the frame length. Where the frames 1-5 and 15-20 correspond to actions and 6-14 corresponds to the transitional phase for the change in action.

Labelling and Frame Selection

The second layer of labelling is carried out over the frame wise labelled data. The frames are selected in a sliding window manner, if the total number of frames within the video are 100 frames then the grouping is made as such 1-20 frames,2-21. . . 80-100. By this approach we are limiting the amount of missing data.
For labelling as mentioned above , if the labels for 1-5 frame and 15-20 frames are different. For example if 1-5 frames are labelled as standing and 15-20 frames are labelled as tip-toe. Then the corresponding label for this set of frames will be 'standing-tip-toe' similarly labelling for 'tip-toe-standing' is also generated. If there is no change in the head and tail labels 'no-change' is assigned to that particular set.

3.6.4 Labelling - Run and Pick

Inorder to classify running scenario, the best approach is to use the z coordinate from the 3D coordinates of the detected feature point. But due to occlusion, it was observed that there is miss-classification between left and right legs during running. As a result

we cannot find any particular trend by analysing the z coordinate of the features. As the full body of the performer is being in motion during the running phase and defining a specific heuristic was very complex. One thing that was observed was the variation in x coordinates within the positional values. That is as the kid moves towards the camera the x value increases. Therefore as a simple workaround, consider the hip as the reference point. If there is variation of magnitude of 0.075 between x coordinates of hip in Frame n and Frame n-5. Then it is labelled as running. But as shown in fig 3.5 , as the kid moves out of camera's field of view while performing the action, we are not able to detect the features points. As a result a proper heuristics for this action was not derived.

3.7 Security and Privacy

Privacy can be a major concern while dealing with data regarding childrens. As already mentioned the videos have been already anonymized but the data still resides useful information such as the school uniform and surrounding environments. This information may lead to privacy violations. There are numerous researches conducted on how to derive information such as weights and bias of the the neural network (17) and extract information against input data. Therefore as a simple workaround for this problem, rather than using the video for the model input, we are extracting the normalised features from the video and storing them in a single file. This operation is carried out in a secured environment and these files are further used in order to train and update the models.

3.8 Models for Classifications

As mentioned in section 2.4, for generating models for action recognition we can use supervised learning or unsupervised learning approaches. One of the most prominent unsupervised learning is through clustering where the model tries to identify clusters formed based on the extracted joint positions and classification is carried out. But in this case we are trying to evaluate the actions performed by childrens of various age groups. The spread observed with in the data points are too large, it may be of the following reason:

- As the action performed by the kids varies a lot.
- As for smaller kids they are not able to perform the required action to a level of satisfying perfection.

Therefore clustering or unsupervised learning is not suitable for this project.
In supervised learning the raw data is transferred into useful information by feature extraction and labelling is carried out. Based on this training data, models are trained.
As mentioned, standing on one leg and imitating action sets can be classified based on a single frame input. As a result models such as the random forest and deep neural networks models are constructed and their performance is evaluated in section 4.1 and section 4.2.
As for standing on tiptoe a Recurrent Neural Network, which is implemented using LSTM cells has been implemented. Reason for RNN network instead of deep neural network is clearly mentioned in section 4.3.
In the case of Run and Pick, as mentioned due to the camera positioning we are losing feature points as the child performs the pick action as shown in fig 3.5. As we have failed to determine a proper heuristic for classifying run and pick action we were not able to generate a model for its classification.

3.9 Summary

In this chapter we have discussed the project design and key challenges. The major challenges that we have identified are converting raw video data into labelled set of frames and scenarios where the feature detectors fail. Inorder to overcome the labelling challenge, we have developed heuristics for each set of actions and annotation of data is automated. Another important aspect that has been mentioned is, what type of models to be used for each action and was concluded that DNN and Random Forest for stand on leg and Imitate whereas RNN for stand on tiptoe and the challenges faced in developing a model for Run and Pick.

Chapter 4

Experiments and Results

Multiple models can be considered for predicting an action. The model selection is based on the trends that are being exhibited by each action. In this chapter for actions that can be evaluated based on a single frame, that is standing on one leg and Imitate. We are considering two types of algorithms that is random forest and Deep Neural Network and it's performance are evaluated. In case of actions like Imitate, where static positional representation alone cannot be used for evaluating. Therefore we are considering advanced recurrent neural networks like LSTM and it's performance are evaluated.

4.1 Models - Stand on One Leg

For training the models out of the 490 videos, we have taken 420 videos as training and the test as test data. Over the training set, feature extraction is carried out and frame wise labelling is done as mentioned in section 3.6.1.

For this specific action we are evaluating two machine learning algorithms random forest and Deep Neural Network and the observed results are tabulated within table 4.1. As we are labelling the frames based on the derived variable that is distance between the hip and knee, in order to increase the generalisation of the model we are not feeding this variable within training data. Because if this derived variable is present within the model, the effect of other inputs gets suppressed as the main objective is to generate a model that is able to predict action considering just the positional coordinates of various joints.

Model	Train Accuracy	Validation Accuracy
DNN	93%	90%
Random Forest	89%	88%

Table 4.1: Random Forest vs DNN for Stand-Leg

From table 4.1, it can be clearly seen that deep neural networks are generating more accurate classification when compared against random forests. One of the main reasons behind this difference is that, in random forest the division is controlled by fixed value whereas in DNN we are extracting various features from the inputs within each hidden layers and finally formulating a relationship between the derived features. As our data consist of childrens of various age groups, the positional coordinates of joints differ with a large margin, As a result the random forest fails in certain cases.

4.1.1 Deep Neural Network - Structure and Features

Feature points used for this action set are hip,knee and foot points on both left and right legs. As shown in fig 4.1, the model consists of 2 hidden layers followed by an output layer with activation function as softmax. The output layer consists of 3 classes each representing the desired labels such as standing, left leg up and right leg up . The value of each node denotes the probability of that particular action being performed.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 8)                 88
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 54
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 21
=================================================================
Total params: 163
Trainable params: 163
Non-trainable params: 0
```

Figure 4.1: DNN Structure

In model tunning we have considered other activation functions such as tanh, but there was no major difference. The optimiser that have been used in this model is the adam optimizer and loss is formulated using categorical cross entropy.

4.1.2 Observations

A sample output observed from the model is shown in fig 4.2. The output and time taken is mentioned at top corner.For calculating time, initial/first frame output is assigned as active_label, and timer based on the frame rate is updated. The timer continues to update as long as there is no change in active_label . If the last 5 predicted labels are not equal

to the active_label a the active_label is assigned as the current predicted label and the timer is reset to 0.

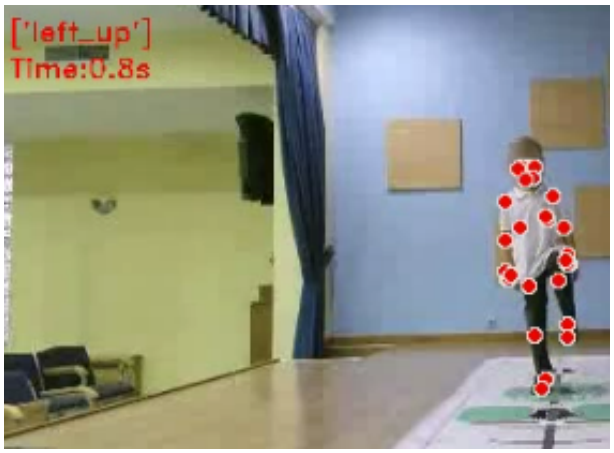

Figure 4.2: Model Output for a single frame

By analysing the confusion matrix formed over the test data,shown in fig 4.3. We can clearly see that the total number of instances for standing is higher when compaired to other two labels. As when we are considering the videos, all frames that fails to satisfy the mentioned heuristics in section 3.6.1 are being categorised as standing. In terms of miss classification, we can clearly see that the model is able to distinguish between left-up and right-up perfectly. Most of the miss-classification happens around the label standing. This is due to the fact that we are providing heuristics based on a single threshold and the model may not be able to generalize the exact relationship satisfying the entire scenario with in the training set. Therefore rather than heuristic based on a single threshold value, more sophisticated heuristic may further improve the performance of the model.

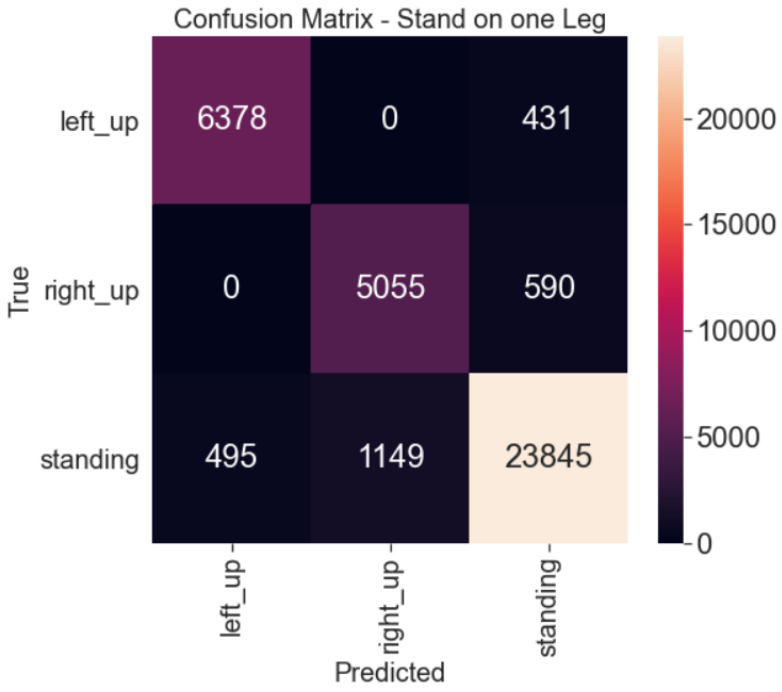

Figure 4.3: Confusion matrix over Test Data - DNN model

4.2 Models - Imitate

For training the model we have taken 85% videos and the rest 15% videos are considered as the test set. A similar approach mentioned in section 4.1 is carried out and the observation for various ML model is tabulated in table 4.2. For this particular action DNN outperforms random forest. A threshold value of 65% has been introduced for this model, therefore the confidence or output probability of a pose is accepted if it's above 65%, otherwise the predicted output is neglected for that particular frame.

Model	Train Accuracy	Validation Accuracy
DNN	87%	85%
Random Forest	83%	82%

Table 4.2: Random Forest vs DNN for Imitate

4.2.1 Reason behind threshold value

As mentioned the imitate action consists of 7 poses, and labelling of the data is mentioned as in table 3.3. In the initial approach we were considering an additional pose, such as 'unknown', which represents all the scenarios that do not lie in table 3.3. When we trained models with training data consisting of 'unknown' classes the maximum output observed was 62% . When we compared the miss-classified data, most of them were assigned

to 'unknown' because this particular class does not exhibit a particular trend. As an enhancement the 'unknown' class has been eliminated from the training set and better models have been constructed. But as the model output consists of 7 poses and for all the frames the model outputs its class to either 1-7 poses. Therefore a thresholding has been applied over this prediction, in order to eliminate predictions with lower confidence. By applying various threshold values over the known videos, through trial and error the optimum threshold value for this use case is 65%.

4.2.2 Deep Neural Network - Structure and Features

Features that have been used in order to train this model are wrist, elbow, shoulder and hip joint positions of both left and right sides. As we can be see from fig 4.4, a deeper model is constructed for this action. The reason behind this approach is that, at initial stage we have developed a shallow model, that is model with 2 hidden layers. But the observed accuracy was very low, even after further tuning and optimization there was no significant variation in accuracy. As a result in order to extract more features or relationship a deeper model with 4 hidden layers is used to build the model. With this a satisfying accuracy for both train and validation set is observed. Further increasing the number of hidden layers slightly increased the training accuracy but in terms of validation accuracy the increase was not observed, which implied that the model was over fitting, as a result further increase in hidden layer is neglected. Optimization and tuning has been applied over and the activation function providing the best classification turned to be relu and optimizer tends to be adam optimizer with default set of hyper parameter values.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 12)                156
_________________________________________________________________
dense_1 (Dense)              (None, 12)                156
_________________________________________________________________
dense_2 (Dense)              (None, 10)                130
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 88
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 63
=================================================================
Total params: 593
Trainable params: 593
Non-trainable params: 0
```

Figure 4.4: Model Structure For Imitate

4.2.3 Observations

A sample output from the model is shown in fig 4.5, where we can see the feature points that is detected the label provided by the model. When analysing the confusion matrix for this model, shown in fig 4.6. Major miss-classifications of label is taken place between adjacent set of poses like Pose 1- Pose 2. Even though the model accuracy is getting 85%, in few instances where the features like between the margin cases of poses, the model fails to predict accurately. As due to the limitation of computation power currently the model is trained over 200 epochs, increasing the number of epochs can be considered as a possible work around for this issue.

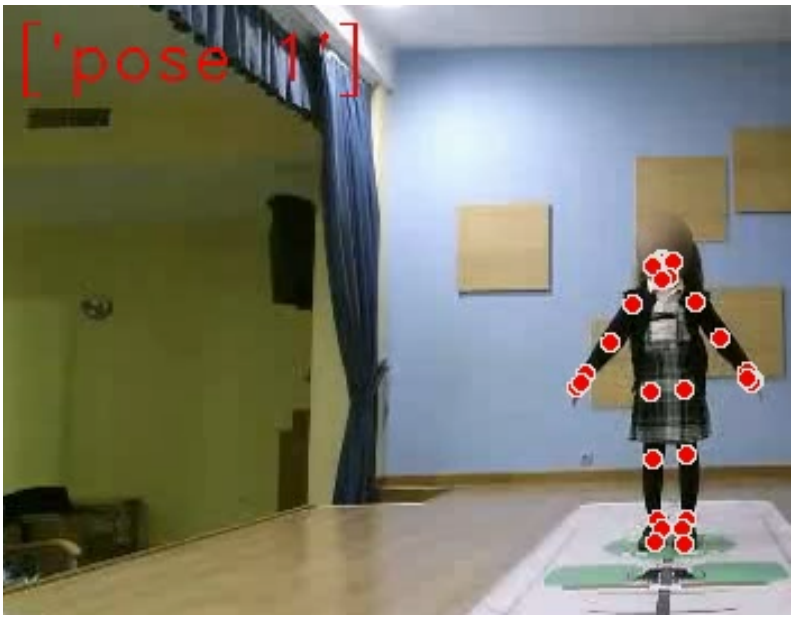

Figure 4.5: Model Output For Imitate

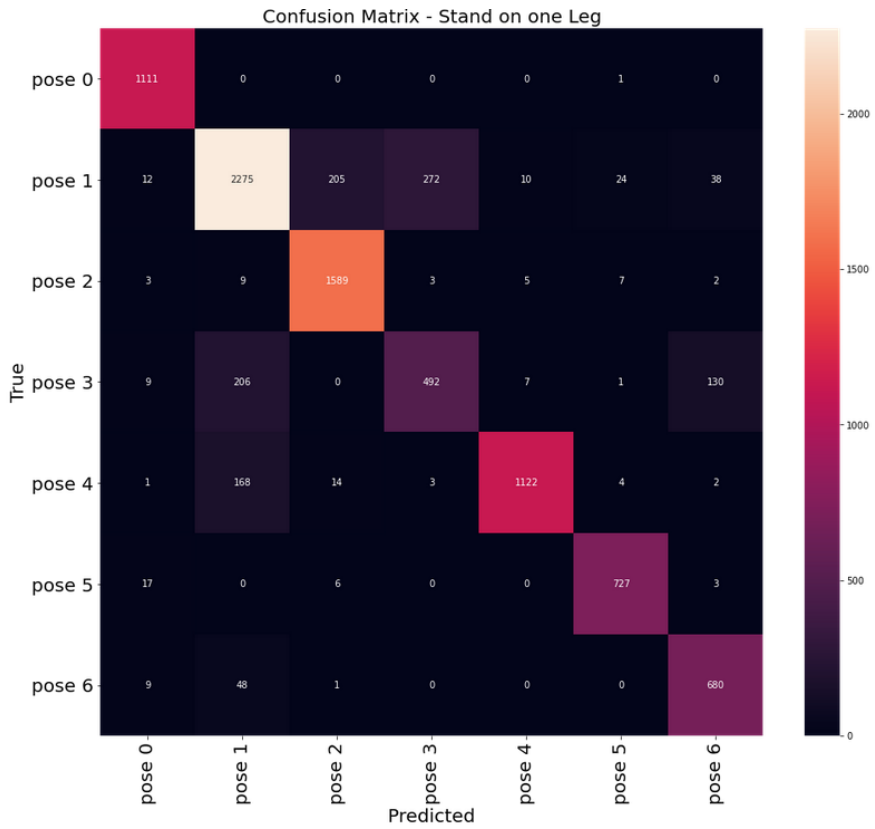

Figure 4.6: Confusion matrix over test set For Imitate

4.3 Recurrent Neural Network - Stand Tip Toe

For this model, we are considering a set of sequential frames as a single input for the model. Instead of predicting the action as such we are generating the model capable of predicting the transitional states within the actions as mentioned in section 3.6.3. The selection of frame and its corresponding labelling are mentioned in the section 3.6.2. Based on this training data, a RNN network composed of LSTM cells are trained and the overall accuracy obtained for the model is 91% (train) and 89%(test)

4.3.1 RNN over DNN for Stand on Tip Toe

The initial approach for this action was to develop a DNN model, as these actions tend to be single frame. That is whether the child is on tip-toe or not, can be determined by considering the single frame. But as mentioned earlier, due to the position of kids being very much away from the camera and due to their dress codes, even though the mediapipe was able to detect the joints for angles and toes but failed to analyse the transition precisely as only a minor variation is taking place. Therefore by analysing the entire video we are defining its heuristics, as mentioned in section 3.6.3. Therefore when

we trained a DNN model, it was giving around 80% accuracy for training but in terms of validation accuracy the observed value was around 58%. The only possible explanation for the decrease of accuracy for validation set is that the heuristic is derived by analysing the entire video rather than a single frame. Therefore when an unknown data/frame is introduced to the model it cannot make the correct prediction as it does not know its adjacent frame inputs. Therefore for this action a RNN model is implemented as the output has to be determined based on the existing as well as its adjacent frames.

4.3.2 Recurrent Neural Network - Structure and Features

As shown in fig 4.7 ,the final model consists of 3 LSTM cells followed by 2 hidden layers which are densely connected and the final output layer consisting of 3 nodes for each action 'no_change', 'standing_tiptoe' and 'tiptoe_standing'. Features that are used inorder to generate the model consist of hip, knee and foot joints of both legs. The model is trained over 100 epochs and the activation function used is tanh.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 20, 64)            25856
_________________________________________________________________
lstm_1 (LSTM)                (None, 20, 128)           98816
_________________________________________________________________
lstm_2 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense (Dense)                (None, 64)                4160
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 99
=================================================================
Total params: 180,419
Trainable params: 180,419
Non-trainable params: 0
```

Figure 4.7: Model Summary - Tiptoe

4.3.3 Observations

A sample output from the model is shown in fig 4.8,the model predicts output for each frame based on the current feature points and considering features points of 20 previous frames. The logic behind the calculation of time taken is similar to one mentioned for action stand on leg in section 4.1.2. The only difference is that the timer is reset only when it encounters either 'standing_tiptoe' or 'tiptoe_standing'. Because with this model

we are predicting the transitions within the video. Let the output for the current frame be 'standing-tiptoe', which means when comparing the current frame and within the previous 20 frames the kid who was standing has done tiptoe. If the next frame gets labelled as 'no-change' which implies the kid is continuing the previous action. Therefore the timer is not reset when no_change label is encountered.

Figure 4.8: Model Output - Tiptoe

By analysing the confusion matrix shown in fig 4.9, we can see that the frequency of no_change tends to be larger when compared to other two labels. This is because in a training video there are only a few instances where the kid performs the tip-toe action and the rest of the frames gets labelled as no_change. Comparing the miss-classification labels, there are quite a few misclassifications between standing-tiptoe and tiptoe-standing. One of the reasons behind this will be due to the positional change due to media-pipes detection failure. For labelling we are considering 1-5 and 15-10 Frames of the 20 Frames input and assuming that in 5-15 Frames a transition is happening, as mentioned in section 3.6.3. Within this assumption there can be variations in position coordinates due to mediapipe feature detection failure and can affect the performance of the model.

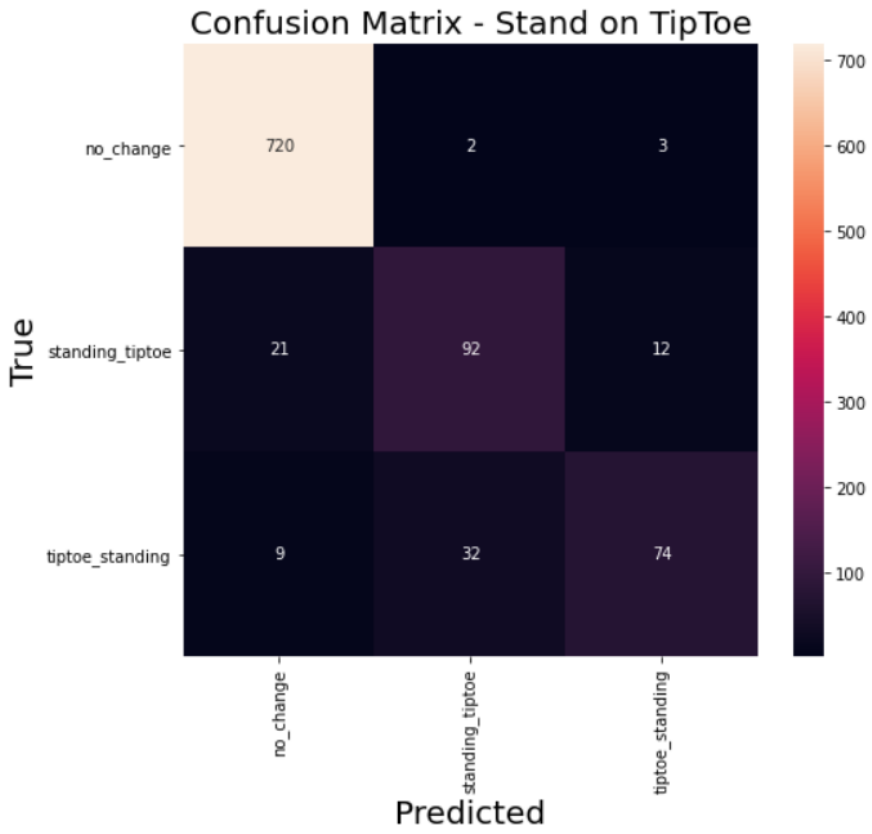

Figure 4.9: Confusion Matrix Against Test Data - Tiptoe

4.4 Models - Run and Pick

It was one of the complex actions, as the child runs towards the camera and performs the pick action then the child moves out of the camera's field of view. As a result feature extraction fails and a proper heuristic for the entire action was not able to be derived. Therefore a proper model for the same was not able to be generated.

4.4.1 Experiments

Following are the various methodologies we have tried out in order to build a proper model for predicting the run-pick action.

- Based on the standard approach, at first we have tried to extract key points from videos as the points are not getting detected while performing pick. Because the child moves too close to the camera most of the frames are getting skipped. As a result the detected frame does not contain the action pick. Therefore the standard approach was dropped.

- Another approach was to classify frames that are missing key points as 'pick' and

train the model with this set of training data. But when observing the video there are instances where the child may run past the camera, as a result the above mentioned approach cannot be applied.

- The most effective approach was to replace the missing features with constant values such as zeros, and train a RNN model for the same. As the action consists of two sub-actions 'run' and 'pick', based on the average frame for 'pick' which tends to be smaller than the action 'run', the frame set for RNN model is determined as 15. And based on this, a RNN model was developed but the accuracy obtained was very low around 52%. Various workarounds like widening the layers and optimizations are carried out but there was no major change. Replacing the missing frame with 0 may be one of the reasons for the low accuracy, but another reason may be due to occlusion and the variation of points when the child runs from one point to another.

4.5 Summary

In this chapter we have build various models like random forest, deep neural network and Recurrent Neural Networks for classification. In terms of performance for actions like stand on one leg and Imitate, DNN outperforms random forest. Whereas in case of Tiptoe, simple neural networks fails to provide satisfactory result, therefore a recurrent neural network using LSTM is generated and was able to achieve 90%+ accuracy over the train set.

Chapter 5

Conclusions & Future Work

The various methodologies and existing methods for human action recognition have been discussed so far. Custom composed models for the specific actions have been developed and detailed in the above chapters. Upgradation and enhancements must be applied to the models in a regular manner. For proper enhancements, we must find the limitations shown by the existing model or approach. Therefore in this chapter we will be focusing on the limitations of the current models and in the later parts we will be discussing the future updates that can be applied in order to rectify the mentioned limitations and further improve the performance of the models.

5.1 Conclusion

This project was an extension of an existing system which is used to track the motor skills of children. One of the main objectives was to predict the gestures performed by childrens, for multiple actions and calculate the time taken for performing the actions. For this we have initially analysed existing methodologies which have been carried out in the Human Action Recognition Problem. Through this study a deep understanding of various sections in gesture detection has been developed. Incorporating this knowledge and modifying these methodologies in such a way that our desired objectives can be achieved, we have finally mentioned a custom methodology that can be applied for detecting action performed by childrens of various age groups. The approach consists of feature extraction, labelling of raw data, frame by frame output prediction. Based on the complexity of action different types of deep learning models are created and labelling is carried out. Out of the 4 mentioned actions, we were able to develop models that can predict with 80%+ accuracy for 3 actions. For this various types of algorithms are considered and found that Deep Neural Network performs better when compared against

random forest. For action tip toe RNN models are required. Due to the limitation in the training videos for run and pick we have failed to develop a satisfying model for this action. These models (for 3 actions) can be incorporated with the existing systems in the future and the scalability of the parent project can be further increased.

5.2 Limitations

Major limitations that have been identified over the existing models are:

- Error in feature detection - Even though media pipe tends to be the optimum approach for feature detection for this case, we can observe that for smaller childrens the feature tracking of media pipe fails in few cases. During the run and pick action it can be observed that, when the kid moves or runs the tracking of feature points fails. As these features are key inputs for the models, misclassification of features tends to decrease the performance of the model.
- Heuristics - The defined heuristics cannot be categorised as a perfect heuristics for the actions. As the model is studying based on the training set labelled using heuristics, proper and accurate heuristics must be formulated.
- Know data set - As we have already mentioned, about 50 videos are manually labelled and performance of the heuristic is carried out based on this data. 50 videos are not enough inorder to justify or to validate the defined heuristics.
- Failed to detect Run and Pick - As we are missing feature points throughout the video, due to camera positions. We have failed to develop a model capable for detecting run and pick.

5.3 Future Work

The next phase that can be implemented over this project are:

- Deployment and integration of the developed model with the existing system.
- the training data quality, as the current quality of the videos are not high and can be categorised as one of the reasons for feature detection failure. Recapturing videos for run and pick action with proper camera placement such that the performer is not running out of camera's field of view.

- If more training data and computational power is available then custom training of CNN model for feature extraction can limit the existing problems faced by media pipe libraries.
- Rather than formulating a single heuristics for all the age groups, multiple heuristics based on age groups will be better. As it will be able to label frames more accurately compared to current heuristics.

Bibliography

[1] R. Bhadra and S. Kar, "Sign language detection from hand gesture images using deep multi-layered convolution neural network," in *2021 IEEE Second International Conference on Control, Measurement and Instrumentation (CMI)*, pp. 196–200, 2021.

[2] R. Lionnie, I. K. Timotius, and I. Setyawan, "An analysis of edge detection as a feature extractor in a hand gesture recognition system based on nearest neighbor," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pp. 1–4, 2011.

[3] E. Nishani and B. Çiço, "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation," in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4, 2017.

[4] R. Zhao, Y. Zhao, R. Deng, and F. Li, "Hierarchical random forest for senior action recognition in videos," in *2017 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp. 171–176, 2017.

[5] T. Liu, Y. Song, Y. Gu, and A. Li, "Human action recognition based on depth images from microsoft kinect," in *2013 Fourth Global Congress on Intelligent Systems*, pp. 200–204, 2013.

[6] S. Vantigodi and R. Venkatesh Babu, "Real-time human action recognition from motion capture data," in *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pp. 1–4, 2013.

[7] Y. Wei and L. Jiang, "Human action recognition based on convolutional neural network," in *2021 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, pp. 194–197, 2021.

[8] Y. Liu, L. Jiang, and Z. Zeng, "Embedded pedestrian detection based on the alex net model," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, pp. 91–95, 2021.

[9] V. Agarwal, K. Sharma, and A. K. Rajpoot, "Ai based yoga trainer - simplifying home yoga using mediapipe and video streaming," in *2022 3rd International Conference for Emerging Technology (INCET)*, pp. 1–5, 2022.

[10] google, "Mediapipe pose estimator." `https://google.github.io/mediapipe/solutions/pose.html`, 2021. Accessed: 2022-07-27.

[11] D. Wang, Q. Shao, and X. Li, "A new unsupervised model of action recognition," in *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 1160–1164, 2015.

[12] S. C. Lai and P. Y. Lau, "Upper body action classification for multiview images using k-means," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, pp. 1–4, 2018.

[13] "Random forest estimator." `https://www.datacamp.com/tutorial/random-forests-classifier-python`, 2021. Accessed: 2022-07-27.

[14] "Recurrent neural network." `https://medium.com/swlh/introduction-to-recurrent-neural-networks-rnns-347903dd8d81`, 2021. Accessed: 2022-07-27.

[15] C. Naguri and R. Bunescu, "Recognition of dynamic hand gestures from 3d motion data using lstm and cnn architectures," pp. 1130–1133, 12 2017.

[16] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, "Blazepose: On-device real-time body pose tracking," *CoRR*, vol. abs/2006.10204, 2020.

[17] D. Rolnick and K. P. Körding, "Identifying weights and architectures of unknown relu networks," *CoRR*, vol. abs/1910.00744, 2019.