Affective Computing for Understanding of Neurological Child Development

A thesis submitted in fulfillment of the requirements

for the degree of Interdisciplinary Dual Degree Programme

by

Akshat Gautam

190110004

under the guidance of

Prof. Sharat Chandran

to the

Centre for Machine Intelligence and Data Science

Indian Institute of Technology Bombay

October, 2023

Statement of Thesis Preparation

I, **Akshat Gautam**, declare that this thesis titled, **"Affective Computing for Understanding of Neurological Child Development"**, hereby submitted in partial fulfillment of the requirements for the degree of **Interdisciplinary Dual Degree Programme** and the work contained herein are my own. I further confirm that:

1. The "Thesis Guide" was referred to for preparing the thesis.
2. Specifications regarding thesis format have been closely followed.
3. The contents of the thesis have been organized based on the guidelines.
4. The thesis has been prepared without resorting to plagiarism.
5. All sources used have been cited appropriately.
6. The thesis has not been submitted elsewhere for a degree.

Name: Akshat Gautam

Roll No.: 190110004

Centre for Machine Intelligence and Data Science

October, 2023

ABSTRACT

Name of student: **Akshat Gautam** Roll no: **190110004**

Degree for which submitted: **Interdisciplinary Dual Degree Programme**

Department: **Centre for Machine Intelligence and Data Science**

Thesis title: **Affective Computing for Understanding of Neurological Child Development**

Name of Thesis Supervisor: **Prof. Sharat Chandran**

Month and year of thesis submission: **October, 2023**

Neurodevelopmental disorders (NDDs) encompass complex conditions characterized by cognitive, communication, behavioral, and motor skill impairments that arise from irregular brain development. This thesis directs its attention toward the critical need for solutions in detecting NDDs, particularly within resource-constrained low and middle-income countries (LMICs). The STREAM project emerges as a response to this pressing need, leveraging mobile technology to consolidate existing screening tools.

However, we need to extract meaningful metrics from the assessment data in the STREAM tool to perform further analysis. The report delves into the methods for generating valuable metrics from STREAM data, focusing on the challenging task of calculating distance from cameras in metric units. Additionally, the report covers the pre-processing steps for deriving metrics from other tasks within the STREAM tool, which involves processing Excel files and images.

Acknowldgements

I would like to express my sincere gratitude to Professor Sharat Chandran for his unwavering support and invaluable guidance throughout this research. His expertise and mentorship have been instrumental in the successful completion of this work.

I extend my special thanks to Shubham Chitnis, whose significant contributions ranged from providing essential code for distance calculations to actively participating in data collection for our experiments.

I would also like to acknowledge Rahul Bishain, whose code contributions provided guidance for both modifying existing scripts and crafting new ones for the pre-processing phase of this research

Akshat Gautam

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Neurodevelopmental disorders (NDDs) encompass complex conditions defined by challenges in cognitive, communication, behavioral, and motor skills due to atypical brain development. This category includes conditions such as intellectual disabilities, communication disorders, autism spectrum disorder (ASD), and attention deficit/hyperactivity disorder (ADHD) [2].The nationwide findings revealed significant proportions of children in India between the ages of 2 and 9 years affected by one or more neurodevelopmental disorders (NDD): 10% in hilly areas, 13% in urban areas, and 18% in rural areas [20].With such a large number of children suffering from NDD, it becomes a considerable challenge for low- and middle-income countries like India, with limited resources and a scarcity of medical professionals, to provide timely diagnoses and initiate the necessary interventions. This results in a significant detection gap.

The START [6] project was envisioned with the goal of bridging this gap. It serves as a scalable and accessible tool for the early identification of autism or nuerodevelopemental markers. Notably, this screening can be conducted by individuals who are not necessarily specialists and can be administered via a tablet within the child's home. This approach significantly enhances the app's reach, making it accessible even in rural and remote areas. The primary target demographic for this tool spans from 0 to 6 years of age. In the next phase, STREAM project was established with its vision extending beyond autism detection. While there are several standardized charts available for physical development, such as those for weight, height, and other physical parameters, there is a notable absence of easily accessible neurological developmental standards. For instance, the WHO's child growth standards provide ideal height-weight charts based on age, and even adults rely on indices like BMI to monitor their physical health. Nevertheless, there is a significant lack of neurological developmental standards that can be readily accessed. STREAM project's aim is to develop a tool that not only facilitates autism screening but also provides a mental development score that can be monitored as the child grows and develops.

1.2 Problem Statement

The data produced through the STREAM assessment requires thorough preprocessing to make it suitable for machine learning or statistical analysis. This process entails the extraction of relevant metrics from each of the tasks. However, not all metrics are equally straightforward to obtain. For example, calculating the distance of a child from the camera based on video recordings during the wheel task, where the child interacts with a black and white wheel displayed on a tablet screen, presents a particular challenge. The main problem statement of this work is to implement techniques to extract these metrics.

This work contributes a robust pipeline for accurately extracting metric distances of the face from the camera, substantiating the methodology through an extensive series of experiments. Furthermore, it demonstrates the practical application of this metric in the context of autism classification. In addition to addressing the distance measurement task, this report encompasses comprehensive pre-processing methodologies for other pertinent tasks, ensuring the extraction of essential metrics from a diverse range of assessments within the STREAM tool.

1.3 Report Structure

The structure of this report unfolds as follows: In Chapter 2, we delve into the existing literature within this domain. Chapter 3 offers valuable context on pivotal topics for readers. Chapter 4 outlines the process of feature extraction from the data. Chapters 5,6 are dedicated to the intricacies of distance calculation. Finally, we conclude by presenting potential avenues for future research and the extension of this work in Chapter 7.

Chapter 2

Literature Review

In this chapter, we'll delve into the existing literature related to our thesis topics. This chapter is divided into sections, each covering the methods used to address a specific topic. We'll also discuss what questions remain unanswered and the limitations of the methods used by other authors. This chapter forms the core of our project report, providing a solid foundation for our research.

2.1 Machine Learning for Autism screening

There is also a lack of medical professionals and affordable diagnostic tools in these countries. People have started to use machine learning techniques to build scalable and accessible screening tools. [16] being the most notable, which used an app named "SenseToKnow" to detect children with neuro-developmental disorders. They used computer vision algorithms to identify and recognize the child's face and estimate the frame-wise facial landmarks, head pose, and gaze. The XGBOOST machine learning algorithm used these features for classification. SHAP value statistics were also used for better explainability of the model. [10] provides a comprehensive review of different machine learning techniques used in Autism Spectrum Disorder(ASD) screening. [5, 1] use eye tracking and machine learning to detect autism. However, both approaches rely on proprietary hardware for gaze tracking, rendering them inaccessible and cost-prohibitive in low income countries. [18] used machine learning on specifically designed question-answer. They applied their techniques to 3 publicly available datasets explicitly designed for children, adolescents, and adults [12] showed that postural sway characteristics and displacements could be used for autism screening. They concluded children with ASD could not stay still in a spot by measuring displacement from the initial position using force plates.

2.2 Distance from camera

The problem of estimating distance of face from the camera using video and images has been extensively studied. Numerous works address this challenge by calculating anatomical features using facial landmarks[19, 17]. Image

pyramids and template matching are implemented by [17] to perform face and eye detection. An empirical formula is calculated relating distance between the eyes and distance from the camera. Similarly, [19] relate faces sizes extracted by the Viola Jones detector with the camera distance. [3] harness Mediapipe face landmarks to get the iris radius and find a logarithmic correlation between the radius and camera distance. All three methods are based on the assumption that the same camera is used for experimentation. This doesn't align with what we see in the real world, where different cameras will project these anatomical features differently on the image plane causing any and all real world distances to get altered. Additionally, their findings rely on the premise of similar face shape and size across the populace. However we know that this is not the case, especially when working with individuals across various age groups, including infants and adults. These empirical relations will also fail when dealing with non-frontal faces, which are commonly encountered. Specific methodologies also leverage deep learning techniques for distance calculation. [4] use a VGG-16 network pre-trained on ImageNet for transfer learning on images with known distances. Their method suffers with the inability to use the same model across different camera focal lengths. Furthermore, their dataset exhibits limitations, encompassing a relatively narrow age group representation, specific lighting conditions, and ethnicities. These limitations could introduce bias into the model's performance. PnP methods have been used previously in the literature for camera distance calculation. [8] perform distance estimation on unseen faces using EPnP. For a test image, they use its 2D facial (fiducial) landmarks and match those with 3D landmarks from exemplar faces to get an approximate distance measure. Similarly, [14] estimate head poses for owls, chameleons, and humans using an iterative variant of PnP.

Chapter 3

Background

This chapter explores fundamental background information related to our thesis. It covers essential topics and concepts that provide the foundation for our research, helping us better understand and contextualize the issues addressed in this thesis.

3.1 STREAM project

The **S**calable **TR**ansdiagnostic **E**arly **A**ssessment of **M**ental health (STREAM) project was established to develop an open-source, scalable, and accessible neurodevelopmental screening tool using mobile technology for low and middle-income countries. The STREAM project team has developed a tablet app that combines screening tools like START, DEEP, and MDAT. These tools are explained separately below. The main goal of this project is to help with the early screening of developmental disorders, especially in children aged 0-6.

3.1.1 START

Screening **T**ools for **A**utism **R**isk using **T**echnology (START) [6] is an open-source autism screening tool that was extensively tested in the Delhi-NCR region. Children diagnosed with Autism spectrum disorder (n=48), Intellectual disability (n=43), and typically developing (n=40) children were recruited for evaluation. This app contains a set of tasks that try to evaluate social, motor, and cognitive skills. These are preferential-looking task, button choice task, bubble-popping task, wheel task, coloring task, and motor following task. Parent-child interaction video and parent questionnaire are also part of the assessment. Refer to 3.1 to get an idea of the various tasks.

3.1.2 DEEP

DEvelopmental Assessment on an **E-P**latform (DEEP)[13] is a cognitive assessment tool in the form of tablet games. It is a set of 14 games (Fig.3.2), each testing different cognitive subdomains. Each game usually targets more than one subdomain. This was designed so that non-specialists in remote areas could easily administer

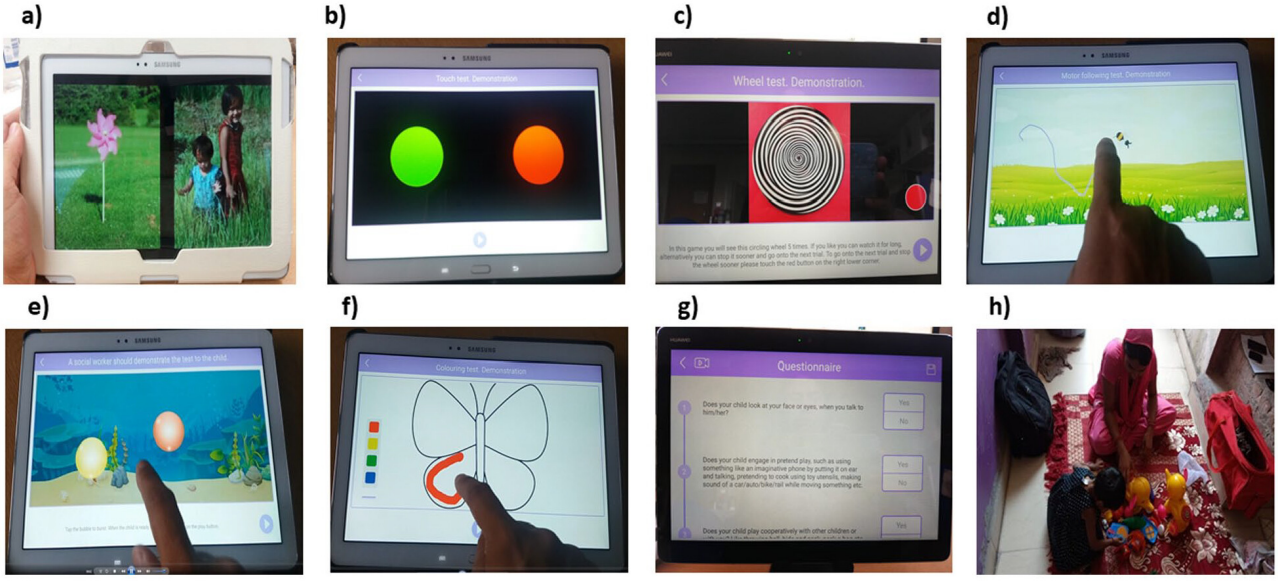


Figure 3.1: images from the (a) preferential looking task, (b) button task, (c) wheel task, (d) motor following task, (e) bubble popping task, (f) colouring task, (g) START questionnaire and (h) caregiver-child interaction observation.

it. We get over 500 features for each child after they play the game, and machine learning algorithms then use these features for supervised learning. The models generated DEEP scores after training, which showed the ability to screen autistic children.

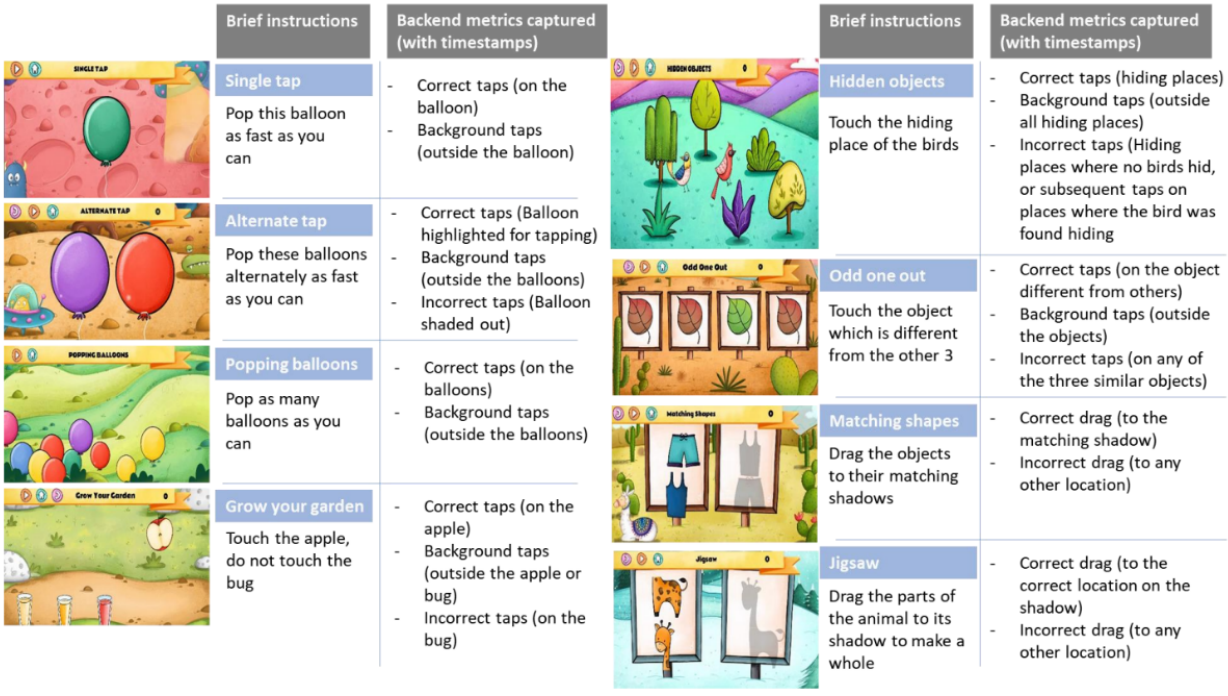


Figure 3.2: Screenshot of deep games and instruction on how to play them taken from [13]

3.1.3 MDAT

The **M**alawi **D**evelopmental **A**ssessment **T**ool (MDAT) [9] was developed to assess child development in rural African areas. This tool comprised a questionnaire containing 136 items with 34 questions belonging to each of the gross motor, fine motor, language, and social domains. These items were used to score children across the four domains.

3.2 Face Mesh

An important term that the reader will frequently encounter in this work is facial landmarks. Landmarks are points of interest like eye margins, center of the nose, elbow and shoulder joints, etc. The skeleton of such landmarks can prove to be a suitable proxy for certain applications where privacy is essential. We deal with face-mesh, which is a dense representation of a typical human face fitted on the subject in consideration. There are multiple off-the-shelf face-mesh detection tools available for getting this representation. The mediapipe face-mesh detection tool is particularly suited for our application. Performance wise, the detection system is able to capture faces well in varying lighting conditions, for videos captured in both lab and in-the-wild settings. In total, 468 facial landmarks are captured as (x, y, z) triplets. Here x and y values are normalized to the pixel space with 0,0 being the bottom-left point and 1,1 the top-right one. The normalized z coordinate is a little tricky as it is not the actual depth. Instead, the depth values are transformed to match the x coordinate scale. The choice of scale for the z axis makes the conversion to metric distances non-trivial.

(a) (b)

Figure 3.3: (a)Medipipe Facemesh (b)Pyramid of vision

Given the normalized landmarks, we employ a view frustum transformation made available by the mediapipe team to calculate metric landmarks. At the end of this, we will have coordinates that are distanced from each other as they would in the real world. The question is what origin are these coordinates in reference to. We believe the x and y coordinate origin is at the center of the imaging window. The z coordinate however, is aligned with the middle of the subject's skull. The convention followed by mediapipe is such that the closer a landmark is to the image plane, its z values will be lesser (not the absolute value, but the actual value). Consecutive frames from the video are passed through this system to generate an array of metric coordinates with respect to some origin and corresponding points in the pixel space.

3.3 Perspective-n-Point

It is essential that we move from an arbitrary origin that changes across frames when using facemesh to something more consistent. As our final goal is to calculate distance from the camera, it makes sense to have the camera as the origin. We have multiple face landmark coordinates (3D) with respect to a coordinate system and their pixel coordinates (2D) on the captured image. Our task is to find a matrix that transforms a point in the original world coordinate system to the camera coordinate system. A classic algorithm that deals with this is the Perspective-n-Point (PnP) algorithm. The original work by [7] coined the term Perspective-n-Point, an algorithm to obtain camera pose with n known 3D-2D correspondence. More recent literature target iterative and non-iterative solutions for the general PnP problem or propose solutions for a subset (P3P, P4P). [15] provide details about the PnP landscape in their work. There are tradeoffs involved in choosing between iterative and non-iterative techniques. Iterative methods tend to be more accurate because of the refinement steps involved in their pipeline; however, their convergence is highly dependent on the choice of initial guess. Closed-form solutions offer speed and simplicity but often involve assumptions that might not hold in specific scenarios. We use the Efficient PnP (EPnP) algorithm, in specific opencv's solvepnp(fig 3.4) based on the work by [11]. In addition to the complexity being O(n), EPnP offers an accurate non-iterative solution for $n \geq 4$ correspondence pairs. However, a pinhole camera model is enforced, and the camera's intrinsic parameters are required. Additionally, the original implementation does not take distortion into account and expects points to be sufficiently far from the camera.

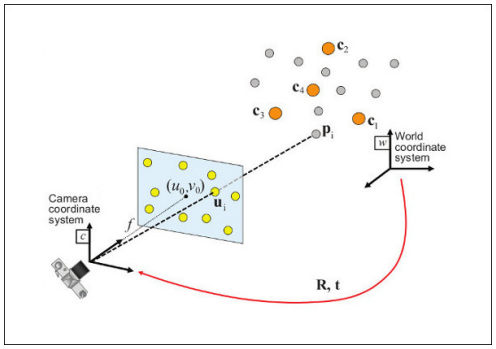
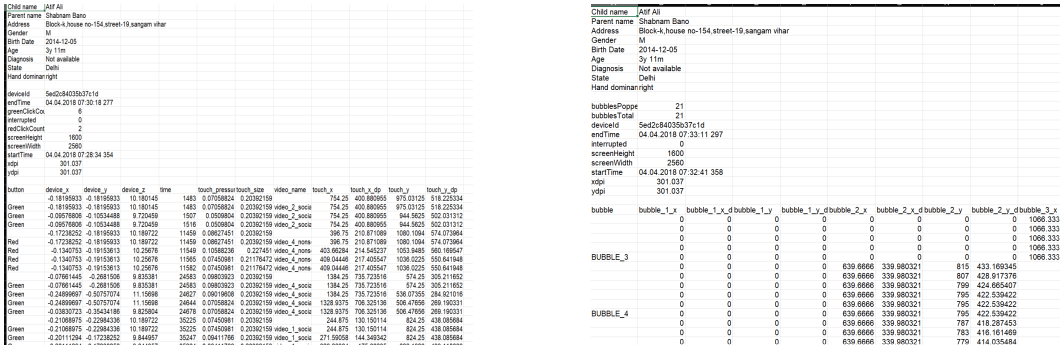


Figure 3.4: Rotation and Translation matrix by PnP algorithm

Chapter 4

Pre-processing

This chapter is dedicated to the preprocessing of START data, aimed at generating metrics suitable for machine learning and statistical analysis. After a child completes a game on the tablet, JSON files are generated on the backend. A Python script is employed to convert these JSON files into XLSX format, which is subsequently processed using MATLAB scripts. The upcoming sections will detail the processing steps for each of these tasks.



Child name	Atif Ali
Parent name	Shabnam Bano
Address	Block-k,house no-154,street-19,sangam vihar
Gender	M
Birth Date	2014-12-05
Age	3y 11m
Diagnosis	Not available
State	Delhi
Hand dominan	right
deviceId	5ed2c84035b37c1d
endTime	04.04.2018 07:30:18 277
greenClickCou	6
interrupted	0
redClickCount	2
screenHeight	1600
screenWidth	2560
startTime	04.04.2018 07:28:34 354
xdpi	301.037
ydpi	301.037

button	device_x	device_y	device_z	time	touch_pressu	touch_size	video_name	touch_x	touch_x_dp	touch_y	touch_y_dp
	-0.18195933	-0.18195933	10.180145	1483	0.07058824	0.20392159		754.25	400.880955	975.03125	516.225334
Green	-0.18195933	-0.18195933	10.180145	1483	0.07058824	0.20392159	video_2_socia	754.25	400.880955	975.03125	516.225334
Green	-0.09576806	-0.10534488	9.720459	1507	0.0509804	0.20392159	video_2_socia	754.25	400.880955	944.5625	502.031312
Green	-0.09576806	-0.10534488	9.720459	1516	0.0509804	0.20392159	video_2_socia	754.25	400.880955	944.5625	502.031312
	-0.17238252	-0.18195933	10.189722	11459	0.08627451	0.20392159		396.75	210.871089	1080.1094	574.073964
Red	-0.17238252	-0.18195933	10.189722	11459	0.08627451	0.20392159	video_4_nons	396.75	210.871089	1080.1094	574.073964
Red	-0.1340753	-0.19153613	10.25676	11549	0.10588236	0.227451	video_4_nons	403.66284	214.545237	1053.9485	560.169547
Red	-0.1340753	-0.19153613	10.25676	11565	0.07450981	0.21176472	video_4_nons	409.04446	217.405547	1036.0225	550.641948
Red	-0.1340753	-0.19153613	10.25676	11582	0.07450981	0.21176472	video_4_nons	409.04446	217.405547	1036.0225	550.641948
	-0.07661445	-0.2681506	9.835381	24583	0.09803923	0.20392159		1384.25	735.723516	574.25	305.211652
Green	-0.07661445	-0.2681506	9.835381	24583	0.09803923	0.20392159	video_4_socia	1384.25	735.723516	574.25	305.211652
Green	-0.24899697	-0.50757074	11.15698	24627	0.09019608	0.20392159	video_4_socia	1384.25	735.723516	536.07355	284.921016
Green	-0.24899697	-0.50757074	11.15698	24644	0.07058824	0.20392159	video_4_socia	1328.9375	706.325136	506.47656	269.190331
Green	-0.03830723	-0.35434186	9.825804	24678	0.07058824	0.20392159	video_4_socia	1328.9375	706.325136	506.47656	269.190331
	-0.21068975	-0.22984336	10.189722	35225	0.07450981	0.20392159		244.875	130.150114	824.25	438.085684
Green	-0.21068975	-0.22984336	10.189722	35225	0.07450981	0.20392159	video_1_socia	244.875	130.150114	824.25	438.085684
Green	-0.20111294	-0.17236252	9.844957	35247	0.09411766	0.20392159	video_1_socia	271.59058	144.349342	824.25	438.085684

(a)

Child name	Atif Ali
Parent name	Shabnam Bano
Address	Block-k,house no-154,street-19,sangam vihar
Gender	M
Birth Date	2014-12-05
Age	3y 11m
Diagnosis	Not available
State	Delhi
Hand dominan	right
bubblesPoppe	21
bubblesTotal	21
deviceId	5ed2c84035b37c1d
endTime	04.04.2018 07:33:11 297
interrupted	0
screenHeight	1600
screenWidth	2560
startTime	04.04.2018 07:32:41 358
xdpi	301.037
ydpi	301.037

bubble	bubble_1_x	bubble_1_x_d	bubble_1_y	bubble_1_y_d	bubble_2_x	bubble_2_x_d	bubble_2_y	bubble_2_y_d	bubble_3_x
	0	0	0	0	0	0	0	0	1066.333
	0	0	0	0	0	0	0	0	1066.333
	0	0	0	0	0	0	0	0	1066.333
	0	0	0	0	0	0	0	0	1066.333
BUBBLE_3	0	0	0	0	0	0	0	0	1066.333
	0	0	0	0	639.6666	339.980321	815	433.169345	
	0	0	0	0	639.6666	339.980321	807	428.917376	
	0	0	0	0	639.6666	339.980321	799	424.665407	
	0	0	0	0	639.6666	339.980321	795	422.539422	
	0	0	0	0	639.6666	339.980321	795	422.539422	
BUBBLE_4	0	0	0	0	639.6666	339.980321	795	422.539422	
	0	0	0	0	639.6666	339.980321	787	418.287453	
	0	0	0	0	639.6666	339.980321	783	416.161469	
	0	0	0	0	639.6666	339.980321	779	414.035484	

(b)

Figure 4.1: (a)Sample input for button task (b)Sample input for bubble task

4.1 Button-Task

The Button task involves displaying two buttons on the tablet screen. One button, when pressed, plays a social video, while the other plays a non-social video. Figure 4.1a provides a sample input XLSX file used in the MATLAB script.

ChildID	Social	Non-social	Interrupt	Trials	Soc_prop
151	4	4	0	8	0.5
152	5	3	0	8	0.625
153	6	2	0	8	0.75

Table 4.1: Sample pre-processed output of button task

The script begins by determining which button (red or green) is linked to the social video. This information is then utilized to calculate the number of social and non-social clicks and subsequently determine the proportion of social clicks. The sample output for the button task after processing is shown in the table 4.1.

4.2 Bubble-Task

In this task, bubbles appear on the screen over time, and the child is supposed to pop the bubbles by touching them. The tablet device records the force applied on the screen. Refer to Figure 4.1b for sample input.

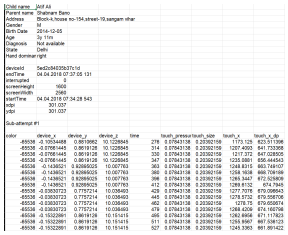
ChildID	Mean Force	Mean DisX	Mean DisY	Interrupt	Bubble Popped
151	0.2066	31.7619	55.2857	0	21
152	0.2085	30.5238	35.8571	0	21
153	0.1719	23.1429	49.3333	0	21

Table 4.2: Sample pre-processed output of bubble task

MATLAB script takes the mean of the force applied and calculates the mean distance between the child's touch point and the bubble's center. The output after pre-processing should look like table 4.2

4.3 Colouring-Task

(a) Subfigure A



(b) Subfigure B

Figure 4.2: (a)Coloured flower by a child (b)Sample input for colour task

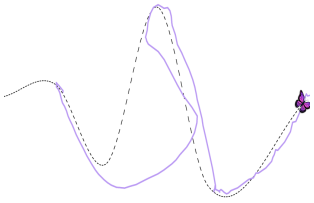
During the Coloring Task, children are presented with simple outline figures and are instructed to carefully fill them with colors. This task utilizes both an XLSX file and a colored JPG image as input, as depicted in Figure 2.

Child ID	Interrupted	Points Inside	Points Outside	Crossover Counts	Proportion
477	0	1047.5	26	11	0.536
478	0	1055.5	107	38	0.647

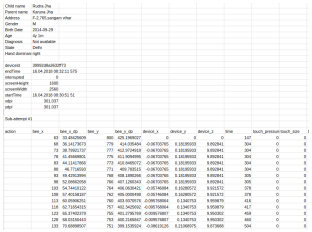
Table 4.3: Sample pre-processed output of colouring task

In the data processing phase, MATLAB employs the inpolygon function to determine the number of times the child crossed over the outline in the figure. Furthermore, the colored image is used to calculate the proportion of the image that has been colored. This is achieved by converting the image into a binary format and subsequently computing the proportion of the colored area. Table 4.3 shows the sample output for this task.

4.4 Motor Following Task



(a) Subfigure A



(b) Subfigure B

Figure 4.3: (a)Child tracing butterfly path (b)Sample input for Motor task

This task involves instructing the child to trace the path of a target butterfly. The target butterfly follows random trajectories with varying velocities in both the x and y axes.

ChildID	rmse_mn	weighted_x_freq_gain_mn	weighted_y_freq_gain_mn	jerk
480	595.98	2.814	37.241	0.020537
484	368.06	1.7563	7.68	0.020204
485	879.18	1.7761	8.3876	0.032486

Table 4.4: Sample pre-processed output of motor following task

To analyze this task, we employ a MATLAB script to compute the root mean square error between the child's touch trajectories and those of the butterfly. Additionally, we calculate the jerk of the child's traced path. A Discrete Fourier Transform (DFT) is applied to determine the frequency gain in this context. For sample output, refer to Table 4.4.

Chapter 5

Obtaining metric distances

This chapter outlines our proposed methodology for calculating the distance from the camera to the subject's face in metric units. Leveraging the capabilities of the Mediapipe FaceMesh toolkit and the EPnP algorithm, we have devised a robust pipeline to accomplish this task. To substantiate our method's effectiveness, we conducted a series of experiments, which are elaborated upon in this chapter.

5.1 Method

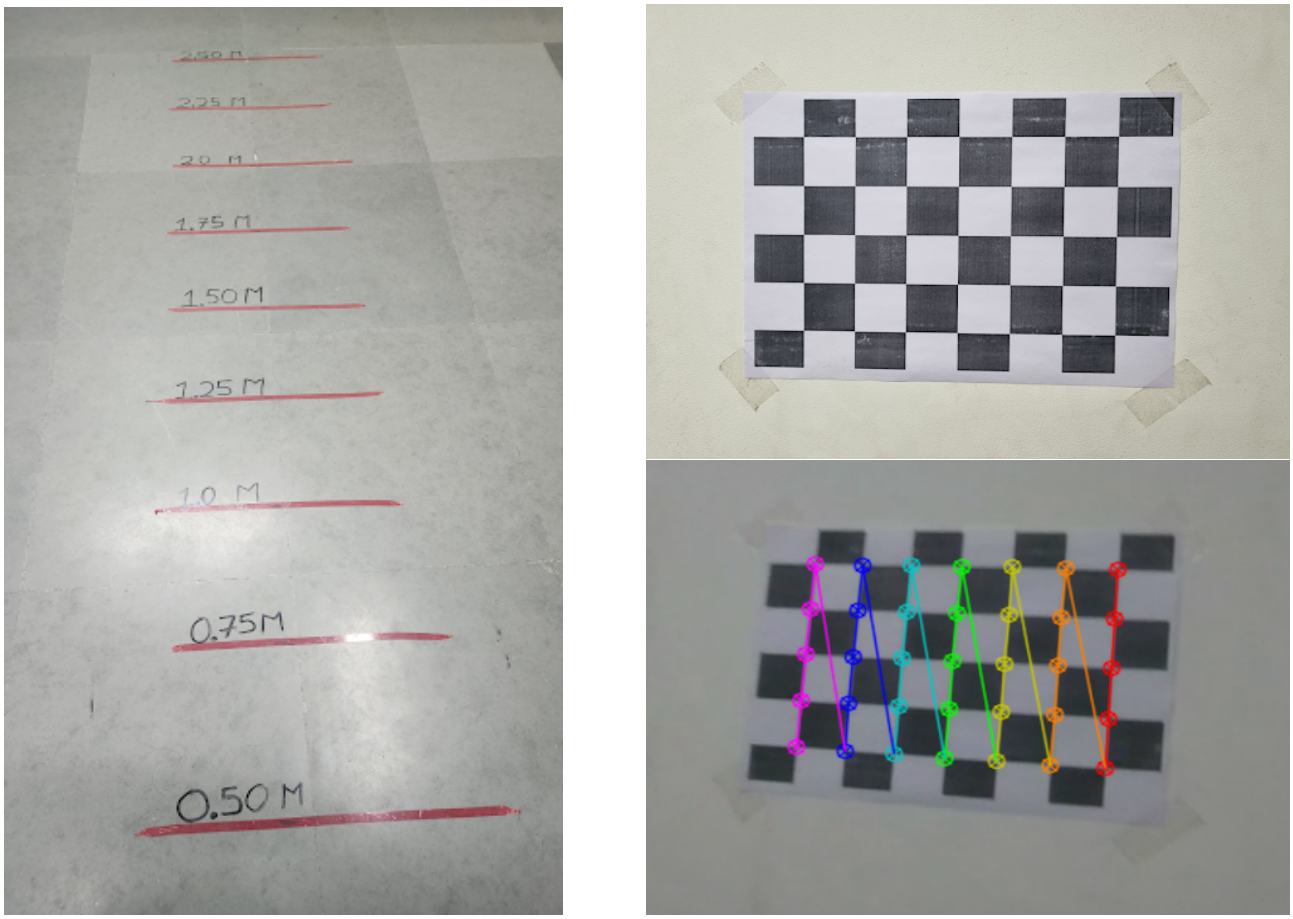
Our methodology commences with a vital step: camera calibration, which allows us to derive the camera's internal matrix. We captured multiple views of a standard chessboard with the capturing device during this process, as illustrated in Figure 5.1b. To obtain the internal matrix and distortion coefficients, we employed OpenCV's findChessboardCorners and calibrateCamera functions.

We utilized mediapipe's faceMesh to extract 468 normalized landmarks (Xn, Yn, Zn). These coordinates were used in generating metric landmarks (Xm, Ym, Zm) by perspective camera frustum (PCF) transformation. Notably, this transformative step also relies on utilizing the internal camera matrix. The EPnP algorithm was employed for each video frame to minimize the projection error between the normalized 2D coordinates (Xn, Yn) and the metric 3D coordinates (Xm, Ym, Zm). Furthermore, it determines the rotation and translation between the camera's coordinate system and the world coordinate system. We specifically extracted the z-component (Tz) of the translation vector as our distance metric.

5.2 Experimental Setup

The experimental setup for all trials took place within our lab. We marked the floor using a standard 3-meter measuring tape to establish reference points for distance measurements. Markings were done at nine equally spaced points starting from 2.5 m to 0.5 m away from the camera. Each test subject held a cue card displaying their actual distance from the camera. We intentionally included subjects with varying heights to ensure a

comprehensive assessment of our algorithm's performance. We fixed the recording device securely to a tripod for consistent and stable readings. The device that we used is Lenovo TAB4 10 PLUS (this being the tablet used for recording START videos). All videos for both calibration and experimentation were taken with the front camera.



(a) (b)

Figure 5.1: Left: Ground truth distance markings, Right: Chessboard frames for camera calibration

In the first variation of our experiment, the test subject was instructed to stand at each of the four designated reference points while holding the corresponding cue card for a duration of 3 seconds. The subject was first positioned at the furthest distance, 1.25 meters away from the camera, and asked to progress towards the closest reference point, 0.5 meters from the camera. The estimated distances at each reference marking for both the

Figure 5.2: Subjects 1 and 2 holding measurement cues

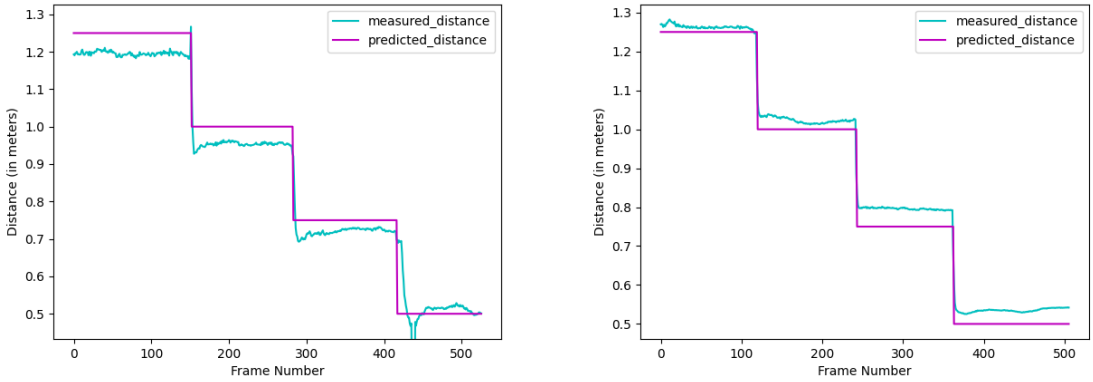


Figure 5.3: First variation: Measured and Predicted distances for Subjects 1 (left) and 2 (right)

subjects can be seen in Fig 5.3. We also tabulate distance-wise median predictions and corresponding errors in Tables 5.1, 5.2.

Real Distance (in meters)	Median Predicted Distance (in meters)	Median Error (in meters)
1.25	1.1957	0.0543
1.00	0.9535	0.0465
0.75	0.7214	0.0286
0.50	0.5146	0.0146

Table 5.1: Results for Subject 1

Real Distance (in meters)	Median Predicted Distance (in meters)	Median Error (in meters)
1.25	1.2623	0.0123
1.00	1.0215	0.0215
0.75	0.7966	0.0466
0.50	0.5347	0.0347

Table 5.2: Results for Subject 2

For the second variation, the subject stood at a constant distance of 0.75m, which is the closest reference mark. The subject then performed in-planar head rotations, that is, facing front and moving the head sideways, while staying at the same camera distance. The results of this experiment for both the subjects can also be seen in Fig 5.4.

For the last variation, the subject moved 0.25m on both sides horizontally, keeping the perpendicular distance from the camera at 0.75m (as showcased in Figs 5.5, 5.6).

Markings beyond 1.25m are reserved for a later section. The FaceMesh model that we use fails to detect faces confidently beyond this point. Although this distance upper bound is suitable for our use case, we mention a way in which similar computations can be carried out for subjects further away.

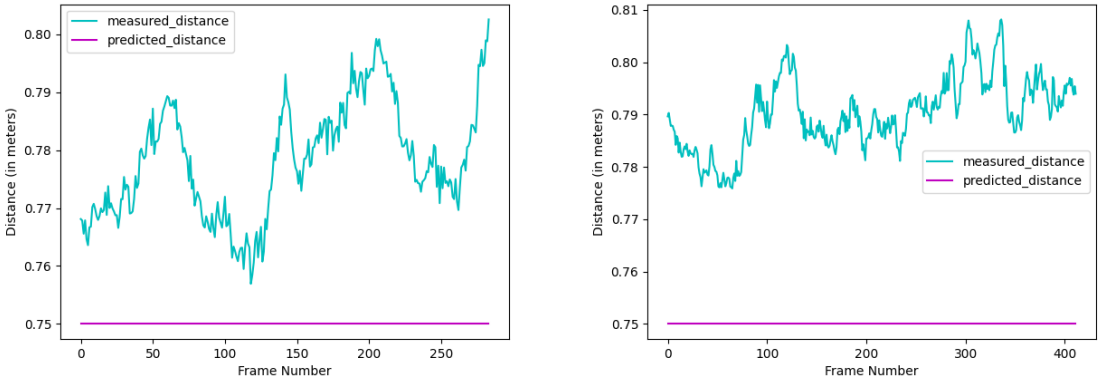


Figure 5.4: Second variation: Measured and Predicted distances for Subjects 1 (left) and 2 (right)

Figure 5.5: Subjects holding the cue and moving left to right at constant 0.75 meters distance

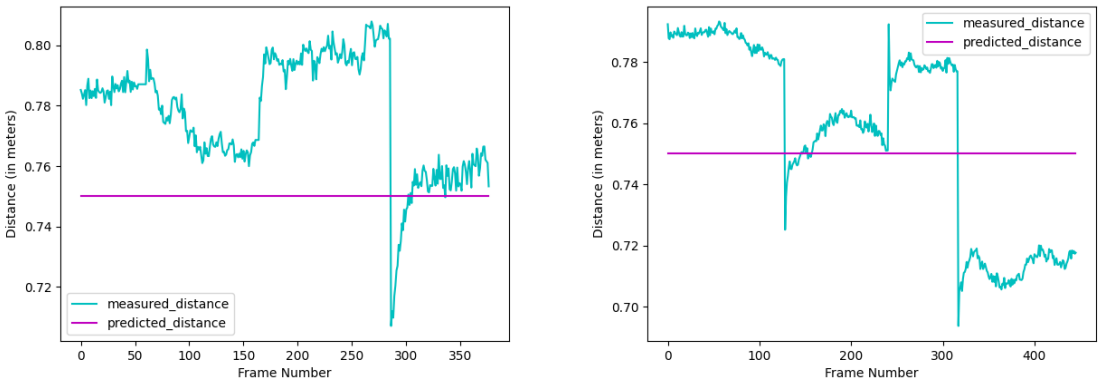


Figure 5.6: Last variation: Measured and Predicted distances for Subjects 1 (left) and 2 (right)

5.3 Discussion

5.3.1 First Variation

In the first case, the test subject just comes closer to camera in a straight line. As illustrated by the figure 5.3 and table5.1,5.2, it is evident that our method effectively estimates the correct distance with an acceptable margin of error, which is less than 6 cm. This slight error could have arisen from various factors encountered during our experiments. For instance, achieving precise alignment with the marking for every distance is challenging. Throughout our experiments, we made diligent efforts to ensure that the midpoint of the foot was aligned with the marking line while standing.

5.3.2 Second and Third Variation

In the second scenario, analyzing the results becomes challenging due to the continuous alteration of the distance caused by the ongoing head motion within the plane. Ideally, one would expect a constant distance since the movement is occurring within the same plane. However, there appears to be a slight decrease in distance (around 2-3 cm) when the subject turns their head to either side (Figure 5.4).

This effect becomes more pronounced in the final experiment (Figure 5.6) when the subject moves 0.25 m to the side. A substantial drop in the measured distance (approximately 4-5 cm) is observed. This phenomenon is consistent for both subjects, although the extent of change varies, and it occurs on both sides.

5.3.3 Method Limitations and Future Improvements

One of the limitations mentioned earlier arises from the face mesh's incapacity to detect faces at distances exceeding 1.5 meters. An alternative approach would involve using MediaPipe's own MediaPose model, which is designed to detect body landmarks rather than facial landmarks and works for longer distances. However, our experiments have demonstrated that, while we were able to predict distances using MediaPose, the results were noisier compared to the more accurate output produced by facemesh.

The second issue is addressed in the third variation of our experiments. We have observed that when a person within the frame moves to the sides or to the extremes of the image frame, the measured distances decrease. This phenomenon can be attributed to several factors. First, it may be due to camera distortion, where the individual, even when maintaining the same perpendicular distance from the camera, appears closer to the human eye when moving to the side. This issue could potentially be resolved by using a higher-quality camera that does not exhibit this distortion effect or by employing more sophisticated calibration methods to determine better distortion coefficients. For example, exploring 3D calibration techniques as an alternative to using a chessboard could be beneficial. The second reason for this ambiguity may stem from inherent assumptions associated with the pinhole camera model in the EPNP algorithm, assumptions that are not entirely accurate when dealing with modern cameras. To mitigate this, we could explore alternative variants of the EPNP algorithm that have been developed to address this particular issue.

Chapter 6

Using distance on START videos

Our work's utility is demonstrated through its application on a specific set of videos, namely, the "wheel task" videos from the START project. In the "wheel task," a black and white wheel is displayed on the tablet screen, and children are instructed to watch the video while their faces are recorded using its camera.

6.1 Method

Upon successfully verifying the functionality of our method through experiments, we applied it to the "Wheel Task" video dataset, as previously mentioned. This dataset comprises 111 videos featuring different children, each categorized into one of three groups: Autism Spectrum Disorder (ASD), Intellectual Disability (ID), or Typically Developing (TD). The ASD and ID groups were recruited from a tertiary clinic and diagnosed by a specialist clinician following the criteria outlined in the Diagnostic and Statistical Manual of Mental Disorders (5th ed.; DSM-V). In contrast, the TD group was recruited from the community. For our task, we merged the ASD and ID groups to differentiate them from the TD group. We obtained a metric distance measurement per frame for each video, following its passage through our pipeline. This yielded a distance vector of dimension n_frames x 1, where n_frames represents the variable number of frames within each video, which can vary among children. Upon conducting an in-depth analysis, we identified the median and standard deviation as effective discriminating features, as shown in Fig 6.1

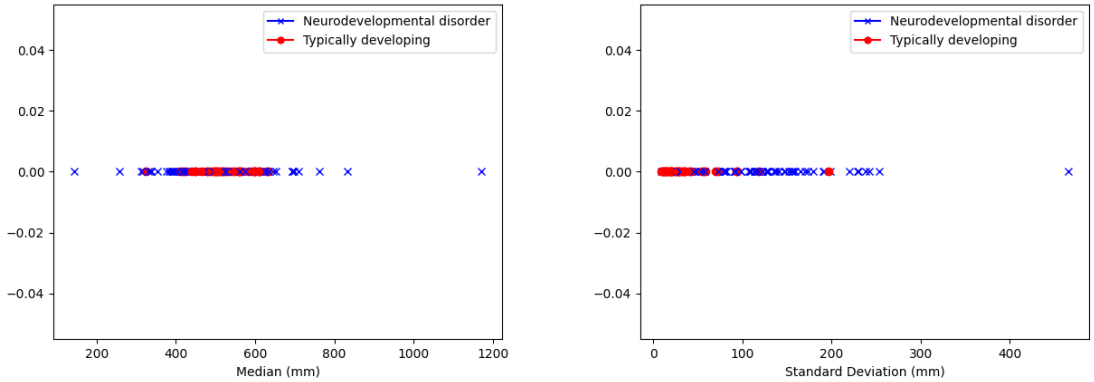
We subsequently utilized these two discriminative features as inputs for our machine learning model, which was specifically designed for a binary classification task. To assess the model's performance, we implemented a 5-fold cross-validation approach, where training was conducted on 4 of the folds, and testing was performed on the remaining fold. The final accuracy metric is calculated as the average of all five test folds. We leveraged various classification algorithms, including Logistic Regression, Support Vector Machine (SVM), and Random Forest, to classify the data. The outcomes of our classification experiments are summarized in Table 6.1. Remarkably, the highest accuracy we achieved was 81%, which was obtained using Logistic Regression, with a corresponding F1 score of 0.74. Given the data's imbalanced nature, featuring 73 children with NDD and only

38 TD children, the F1 score assumes particular significance as an evaluation metric.

Algorithm	Accuracy (%)	F1 Score
Random Forest	78.46	0.67
Logistic Regression	81.23	0.74
SVM	73.08	0.55

Table 6.1: Summary of Classification Results

6.2 Discussion



(a) (b)

Figure 6.1: (a) Plot of median distances (b) Plot of Standard deviation

Based on the observations from the plots, several significant patterns emerge. Typically developing (TD) participants tend to display lower standard deviations, and their median distances rarely deviate to extreme values. This distinctive behavior could potentially serve as a basis for classification. On the other hand, children with neurodevelopmental disorders (NDD), particularly those with conditions like autism spectrum disorder (ASD) and intellectual disabilities (ID), exhibit different characteristics. It appears that some NDD children struggle to remain still while focusing on the tablet screen. The median plots also suggest that some NDD children tend to get very close to the screen when interacting with the wheel.

The decision to combine ID and ASD into a single category is based on the recognition that there is overlap between these conditions. Some children with ASD may also exhibit features of intellectual disabilities. Currently, our dataset suffers from data imbalance, with a larger number of non-TD samples. In the future, data sampling techniques can be explored to address this imbalance. Additionally, we can consider binary classification between just ASD and TD, which would lead to a more balanced dataset. To enhance our analysis, we can move beyond simple features like mean and standard deviation and explore more complex features, such as using the distances between all frames as time series data. This could open the door to employing deep learning models like LSTM and RNN for improved classification.

Chapter 7

Future Work

Having successfully completed the pre-processing for analyzing various START tasks and estimating metric distances from wheel videos, the next step involves extending these techniques to the STREAM dataset (n$\bar{4}$000). As outlined in the introduction, the overarching objective is to generate scores or metrics that facilitate the mapping of neurological development. Our ultimate aim is to create a diverse set of scores, each specifically addressing distinct subdomains such as social, motor, and cognitive skills.

7.1 Dataset

Given that the STREAM task encompasses a combination of tools such as START, DEEP, and MDAT, we will have access to features derived from all three of these instruments. The START task data will undergo preprocessing through scripts to generate essential metrics. Similarly, we will obtain DEEP game metrics and MDAT scores. The amalgamation of data from these three sources will result in a dataset of dimensions 4000 x n_features, where 'n_features' signifies the total number of features.

Regarding the clinical ground truth, a subset of children (n=1000) will undergo a secondary evaluation involving the Griffiths Mental Development Scales (GMDS). GMDS stands as one of the foremost clinical tools for assessing a child's development, as cited (provide citation). Alongside an overall developmental score, GMDS is capable of capturing developmental scores in five distinct areas of learning: Foundations of Learning, Language and Communication, Eye and Hand Coordination, Personal-Social-Emotional, and Gross Motor. Consequently, for these 1000 samples, we will have access to five developmental scores within each of these five domains, resembling what we aim to generate through the STREAM app.

7.2 Methods

In the following subsection, we delve into the various machine learning paradigms applicable to solving our problem and provide insights from relevant literature.

7.2.1 Supervised Learning

Supervised learning, a category within machine learning, leverages labeled data to train algorithms for making predictions or classifications. In our specific context, the clinical test GMDS serves as the ground truth. As mentioned earlier, our labeled dataset will consist of 1000 samples. We propose the utilization of a deep learning network, which takes X_i as input (where i ranges from 1 to n_{features}, encompassing all the features from the three tools).

We will provide the network with input x, where the dimensions comprise a combination of all features extracted from the three tools. The primary objective is to train the model to optimize the GMDS scores, which form a vector of size 5, as they represent scores across five distinct domains. Once the model is successfully trained, it will be deployed to generate developmental scores for new children who have undergone the STREAM assessment.

Let's define the following components:

- Input data: x
- GMDS scores: $GMDS$
- Model parameters: Θ

Our objective is to minimize the loss $L(\Theta)$, which measures the discrepancy between the predicted GMDS scores and the ground truth GMDS scores:

$$L(\Theta) = \text{Loss}(GMDS_{\text{predicted}}, GMDS)$$

This loss function, based on a suitable choice (e.g., mean squared error), quantifies the difference between the predicted and actual GMDS scores. Training the model involves adjusting the parameters Θ to minimize this loss, ensuring the model's ability to generate accurate developmental scores for children based on the combined features extracted from the three tools.

7.3 Unsupervised Learning

In this methodology, we break free from the reliance on clinical GMDS scores as labels. Instead, we aim to generate scores using only the features extracted from the START tool. To achieve this, we draw inspiration from item response theory, a framework used to measure abilities based on tests and questionnaires. For example, we consider a 1-parameter logistic model (PLM), which is defined as:

$$P(U_i) = \frac{e^{\theta - b_i}}{1 + e^{\theta - b_i}}$$

Where:

- U_i represents the response to item i (1 for correct, 0 for incorrect).
- θ is the latent ability or trait of the respondent.

- b_i stands for the difficulty parameter associated with item i.

In our context, $P(U_i)$ can be substituted with a tool metric, such as the accuracy of a DEEP game. The θ parameter obtained from the model can be interpreted as the ability or score for that game. Additionally, this score may provide insights into the domains targeted by the game in terms of child development.

The literature, such as [21], has explored the fusion of deep learning and Item Response Theory (IRT). They introduced the Dynamic Key Value Memory Network (Figure 7.1) to monitor a student's learning and ability. Notably, they employed an attention-like model where each question is associated with a specific set of concepts, and the child's ability in those concepts is utilized to predict their response . While their work is centered around the temporal dimension, we can adapt their approach to our context, which doesn't involve time dependencies.

In our scenario, we can contemplate methods to separately estimate student ability and question difficulty using distinct networks. Subsequently, we can leverage IRT to optimize game metrics by combining these estimates.

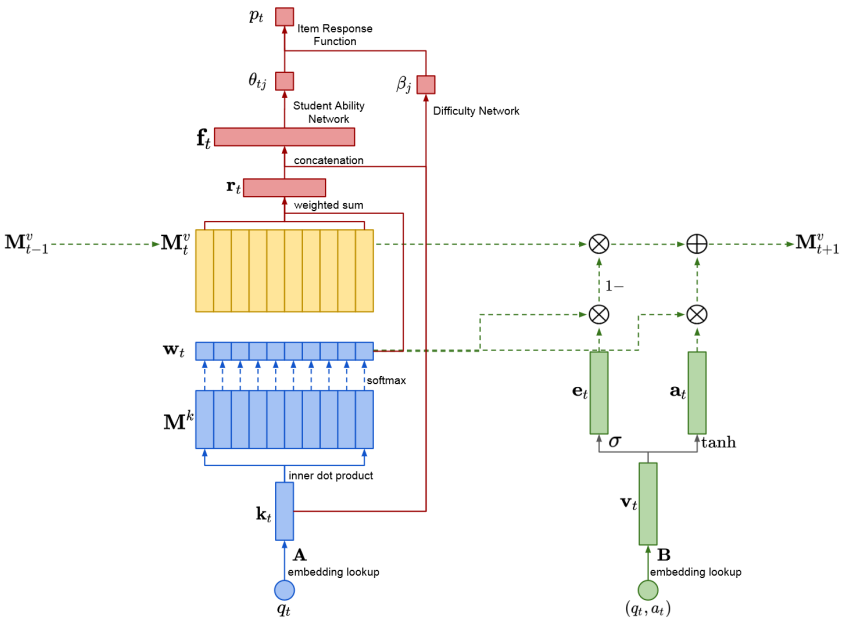


Figure 7.1: Dynamic Key Value Memory Network

Chapter 8

Conclusion

8.1 Conclusion

The report initially highlights the necessity for a tool like STREAM, particularly in resource-constrained countries. It introduces a pipeline for estimating the metric distance of an individual from a camera, utilizing Mediapipe's FaceMesh to identify facial landmarks and subsequently applying the EPNP algorithm for translation calculation. The contributions of this work encompass:

1. **Distance Estimation from Videos for Autism Screening:** This report introduces a method for estimating distances from videos, which is then applied to the crucial task of autism screening. By employing distance analysis, this work takes a significant step toward the early detection of autism.

2. **Processing STREAM App Data for Metric Generation:** Another noteworthy contribution is the development of a methodology to process data generated by the STREAM app. This process yields meaningful metrics that can be further subjected to in-depth analysis.

In conclusion, the report not only provides insights into current accomplishments but also outlines future directions toward the ultimate goal of establishing a standardized neurodevelopmental metric. These future directions aim to enhance the assessment of child development and contribute to early intervention strategies for neurodevelopmental disorders.

Bibliography

[1] Mariano Alcañiz, Irene Alice Chicchi-Giglioli, Lucía A. Carrasco-Ribelles, Javier Marín-Morales, Maria Eleonora Minissi, Gonzalo Teruel-García, Marian Sirera, and Luis Abad. Eye gaze as a biomarker in the recognition of autism spectrum disorder using virtual reality and machine learning: A proof of concept for diagnosis. *Autism Research*, 15(1):131–145.

[2] American Psychiatric Association et al. Dsm-5 development: Neurodevelopmental disorders. *http://www. dsm5. org/ProposedRevision/Pages/NeurodevelopmentalDisorders. aspx*, 2012.

[3] Syed Ausaf Hussain, Waseemullah, and Najeed Ahmed Khan. Face-to-camera distance estimation using machine learning. In *2022 3rd International Conference on Innovations in Computer Science Software Engineering (ICONICS)*, pages 1–8, 2022.

[4] Enrique Bermejo, Enrique Fernandez-Blanco, Andrea Valsecchi, Pablo Mesejo, Oscar Ibáñez, and Kazuhiko Imaizumi. Facialscdnet: A deep learning approach for the estimation of subject-to-camera distance in facial photographs. *Expert Systems with Applications*, 210:118457, 2022.

[5] Romuald Carette, Mahmoud Elbattah, Gilles Dequen, Jean-Luc Guérin, and Federica Cilia. Visualization of eye-tracking patterns in autism spectrum disorder: Method and dataset. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 248–253, 2018.

[6] Indu Dubey, Rahul Bishain, Jayashree Dasgupta, Supriya Bhavnani, Matthew K Belmonte, Teodora Gliga, Debarati Mukherjee, Georgia Lockwood Estrin, Mark H Johnson, Sharat Chandran, Vikram Patel, Sheffali Gulati, Gauri Divan, and Bhismadev Chakrabarti. Using mobile health technology to assess childhood autism in low-resource community settings in india: An innovation to address the detection gap. *Autism*, 0(0):13623613231182801, 0. PMID: 37458273.

[7] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.

[8] Arturo Flores, Eric Christiansen, David Kriegman, and Serge Belongie. Camera distance from face images. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Baoxin Li, Fatih Porikli, Victor Zordan, James Klosowski, Sabine Coquillart, Xun Luo, Min Chen, and David Gotz, editors, *Advances in Visual Computing*, pages 513–522, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[9] Melissa Gladstone, Gillian A. Lancaster, Eric Umar, Maggie Nyirenda, Edith Kayira, Nynke R. van den Broek, and Rosalind L. Smyth. The malawi developmental assessment tool (mdat): The creation, validation, and reliability of a tool to assess child development in rural african settings. *PLoS Medicine*, 7(5), 2010.

[10] Kayleigh K. Hyde, Marlena N. Novack, Nicholas LaHaye, Chelsea Parlett-Pelleriti, Raymond Anden, Dennis R. Dixon, and Erik Linstead. Applications of supervised machine learning in autism spectrum disorder research: A review. *Review Journal of Autism and Developmental Disorders*, 6(2):128–146, 2019.

[11] Vincent Lepetit, Francesc Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81:155–166, 2009.

[12] Yumeng Li, Melissa A. Mache, and Teri A. Todd. Automated identification of postural control for children with autism spectrum disorder using a machine learning approach. *Journal of Biomechanics*, 113:110073, 2020.

[13] Debarati Mukherjee, Supriya Bhavnani, Akshay Swaminathan, Deepali Verma, Dhanya Parameshwaran, Gauri Divan, Jayashree Dasgupta, Kamalkant Sharma, Tara C. Thiagarajan, and Vikram Patel. Proof of concept of a gamified developmental assessment on an e-platform (deep) tool to measure cognitive development in rural indian preschool children. *Frontiers in Psychology*, 11, 2020.

[14] Shay Ohayon and Ehud Rivlin. Robust 3d head tracking using camera pose estimation. volume 1, pages 1063–1066, 01 2006.

[15] Shiye Pan and Xinmei Wang. A survey on perspective-n-point problem. In *2021 40th Chinese Control Conference (CCC)*, pages 2396–2401, 2021.

[16] Sam Perochon, J. Matias Di Martino, Kimberly L. Carpenter, Scott Compton, Naomi Davis, Brian Eichner, Steven Espinosa, Lauren Franz, Pradeep Raj Krishnappa Babu, and Guillermo Sapiro. Early detection of autism using digital behavioral phenotyping. *Nature Medicine*, 2023.

[17] Khandaker Abir Rahman, Md. Shafaeat Hossain, Md. Al-Amin Bhuiyan, Tao Zhang, Md. Hasanuzzaman, and H. Ueno. Person to camera distance measurement based on eye-distance. In *2009 Third International Conference on Multimedia and Ubiquitous Engineering*, pages 137–141, 2009.

[18] Suman Raj and Sarfaraz Masood. Analysis and detection of autism spectrum disorder using machine learning techniques. *Procedia Computer Science*, 167:994–1004, 2020.

[19] Mohamed Tahir Ahmed Shoani, Shamsudin H. M. Amin, and Ibrahim M. H. Sanhoury. Determining subject distance based on face size. In *2015 10th Asian Control Conference (ASCC)*, pages 1–6, 2015.

[20] Donald Silberberg, Narendra Arora, Vinod Bhutani, Maureen Durkin, Shefalli Gulati, Mkc Nair, and Jennifer Pinto-Martin. Neuro-developmental disorders in india - from epidemiology to public policy (p7.324). *Neurology*, 82(10 Supplement), 2014.

[21] Chun-Kit Yeung. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. *CoRR*, abs/1904.11738, 2019.