

University of Southampton
Faculty of Physical and Applied Sciences
Electronics and Computer Science

Alleviate Children's Health Issues through Games and Machine Learning



Author: Pier Paolo Ippolito

Project Supervisor: Koushik Maharatna

Second Examiner: Martin Charlton

April 30, 2019

A progress report submitted for the award of
MEng Electronic Engineering with Artificial Intelligence

“The reasonable man adapts himself to the world: the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man.”

— George Bernard Shaw, *Man and Superman*

Acknowledgements

I would like to thank my supervisor Professor Koushik Maharatna, for giving me the opportunity to do this project. This experience enabled me broaden my knowledge on a variety of topics and skills which will play a crucial role in my upcoming career.

I would additionally like to thank, Rafael Gutierrez Nuno and Noura Alotaibi for having introduced me to the Enobio 20 EEG Cap system and to the data-set used throughout this project.

Finally, I express my gratitude to my second examiner, Professor Martin Charlton, for his support throughout the project.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

I have acknowledged all sources, and identified any content taken from elsewhere.

During the course of this project I followed different online tutorials and courses. All of them have been acknowledged in the body of the report and referenced in the bibliography (such as: [25], [26], [28], [33], [36]). Any minor additional source used, has been acknowledged in the code files available in the Project Design Archive as a comment.

I did all the work myself and I have not helped anyone else.

The material in the report is genuine, and I have included all my data/code/designs.

I have not submitted any part of this work for another assessment.

My work received ethical approval from the University of Southampton Ethics Committee. My application for Secondary Data Analysis has been sent through the ERGO II portal and is available in Appendix F (Submission ID = 48201, Category D). This application was necessary to carry out the data-sets exploration related to this project.

Abstract

Children can develop disabilities at birth. Identifying their problems at an early stage can lead to more consistent improvements later in life. This is mainly dependent on neuroplasticity (the brain's ability to reorganise itself throughout life), which is much higher during the first few years of our life.

Children may be affected by different types/forms of disorders like disability and autism. These can be caused by factors such as: genetics, environment, diet and lifestyle. Therefore, prevention can play a crucial role in decreasing the number of children's negatively affected.

Different intervention approaches such as The Early Start Denver Model (ESDM) and Applied Behavior Analysis (ABA) can be used to aid improvements. Using both software and hardware solutions can enhance progresses increasing children's level of engagement.

List of abbreviations

- ASD** = Autism Spectrum Disorder
- ESDM** = Early Start Denver Model
- ABA** = Applied Behavior Analysis
- EEG** = Electroencephalogram
- qEEG** = Quantitative EEG
- AI** = Artificial Intelligence
- NLP** = Natural Language Processing
- DL** = Deep Learning
- ANN** = Artificial Neural Network
- LSTM** = Long Short Term Memory
- CNN** = Convolutional Neural Network
- AUC** = Area Under The Curve
- ROC** = Receiver Operating Characteristics
- TPR** = True Positive Rate
- FPR** = False Positive Rate
- Wiimote** = Wii Remote controller
- IR** = Infrared
- GUI** = Graphical User Interface
- EXE** = EXEcutable file
- APK** = Android application package
- JA** = Joint Attention
- GIF** = Graphics Interchange Format
- NIC** = Neuroelectrics Instrument Controller
- CSV** = Comma separated values
- TCP** = Transmission Control Protocol
- SVM** = Support Vector Machines
- LDA** = Linear Discriminant Analysis
- GNB** = Gaussian Naive Bayes
- PCA** = Principal Component Analysis

Contents

Acknowledgements	i
Statement of Originality	ii
Abstract	iv
List of Abbreviations	v
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	1
2 Background Theory	3
2.1 Autism Spectrum Disorders (ASD)	3
2.2 Applied Behavior Analysis (ABA)	4
2.3 Early Start Denver Model (ESDM)	5
2.4 Electroencephalogram (EEG)	5
2.5 Serious Games and Gamification	6
2.6 Artificial Intelligence (AI)	8
2.6.1 Long Short-Term Memory (LSTM)	10
2.6.2 Convolutional Neural Network (CNN)	12
2.6.3 Performance Metrics	13
2.6.4 Ethics	17
3 Gaming Platform	18
3.1 Hardware implementation	18
3.2 First Game Development	19
3.3 Imitation Games	21
3.3.1 Second Game Development	21
3.3.2 Third Game Development	22
3.3.3 Fourth Game Development	23
3.4 Joint Attention Games	24
3.4.1 Fifth Game Development	25
3.4.2 Sixth Game Development	26
3.4.3 Seventh Game Development	27

3.5 EEG Cap	28
4 Artificial Intelligence	30
4.1 Faces Responses in Children with ASD	30
4.1.1 ML Classification	31
4.1.2 Deep Learning Classification	33
4.1.2.1 Long Short-Term Memory	33
4.1.2.2 Convolutional Neural Network	36
4.2 Autism Study Using Behaviour Imaging	39
4.2.1 Game Application	39
5 Project Management	41
5.1 Time Management	41
5.2 Ethics Application	42
5.3 Project Challenges	42
6 Conclusion	43
6.1 Further Advancements	43
Bibliography	44
Appendices	49
A Project Management	50
B Evaluation Board wiring and stand	55
C Project Brief	57
D Hardware Costs	58
E Listings	59
F Ethics Application	61
G ML MATLAB Analysis	65
H Individual Stimulus Data-sets Results	71
I Decision Tree Classification	74
J LSTM Training/Test Loss and Accuracy	75
K Design Archive Guide	76
L Word Count	77

Chapter 1

Introduction

1.1 Motivation

Autism Spectrum Disorder (ASD) is a type of neuro-developmental disorder. These types of disorders are mainly considered as disruption of regular brain functioning. Over the last few years there has been reported an increased number of children developing this pathology. Data drawn from a 2014 survey in the United States has shown an increase of 16 percent compared to the previous same survey conducted in 2012. According to this research, about 1 in 59 children in the United States have Autism Spectrum Disorder ("CDC's Autism and Developmental Disabilities Monitoring (ADDM) Network", [1]). Recent research have shown that intensive early start intervention approaches can lead to major improvements to this condition. Although early intervention can be used with all the children on the ASD spectrum, response to these treatments can vary substantially from child to child. There can be many different reasons to cause diversity in improvements such as: symptom severity, joint attention, adaptive skills, interest in objects, imitation and play skills [2].

1.2 Objectives

Current strategies used to alleviate autism conditions relies on three interdependent steps:

1. Early detection.
2. Approximately 25 hours per week of intensive personalised intervention.
3. An effective monitoring of a child's progress.

This project aims, firstly to outline the main behavioural intervention techniques used nowadays such as the The Early Start Denver Model (ESDM) and Applied

Behaviour Analysis (ABA) and to then offer a suite of multi-player games as personalised delivery intervention for the children.

These type of games are designed to be accessible both in a clinical and a home environment to facilitate intervention. In order to record the brain activity of children while playing the games, EEG cap readings are synchronised with the game and results are stored in a CSV file. These recordings will play a crucial role in observing if the game is stimulating the child and which areas are stimulated most in particular. Using this information and keeping track of the child's game scores, it will then be possible to examine what kind of improvements have been made during a predetermined time period. Analysing these results will also serve as feedback to understand what type of games can lead to most substantial improvements over others.

Finally, the use of Machine Learning can facilitate early detection of disabilities, making use of available data to find patterns to distinguish between children suffering the disorder and those not.

Chapter 2

Background Theory

2.1 Autism Spectrum Disorders (ASD)

Autism Spectrum Disorder is used as an umbrella term to enclose different types of disabilities. Suffering from ASD can lead to many types of difficulties, which can vary vastly between patients, such as communication, social, language and motor disorders. Different types of Autism Spectrum Disorders include:

1. Asperger Syndrome = children affected can have a high intelligence quotient (IQ) , but suffer of social interaction problems.
2. Autistic Disorder = children affected demonstrate unusual behaviours, language delays and communication difficulties.
3. Pervasive developmental disorder, not otherwise specified (PDD-NOS) = children who display some ASD symptoms but do not meet all the Asperger Syndrome or Autistic Disorders criteria.

Genes can play an important role in determining if a child is affected or not by autism, but there is still a lot that can be done to prevent it. Mothers who suffer a low iron intake, have an unhealthy lifestyle or are exposed to a high level of air pollution are in fact more likely to give birth to autistic children.

Early signs to understand if a child is affected or not by autism can be lack of: happy expressions, response to stimulus, gesturing for communicating and attempts to speak [3] (Figure 2.1).

Many behavioural therapies have been designed during the last few decades in order to aid children affected by autism such as: Floortime, Pivotal Response Treatment (PRT), Verbal Behavior Therapy, ABA and ESDM. During Floortime, parents and therapists take part in games the child likes to play, encouraging the child to take initiatives on how to develop the game and share this activity with other people. In

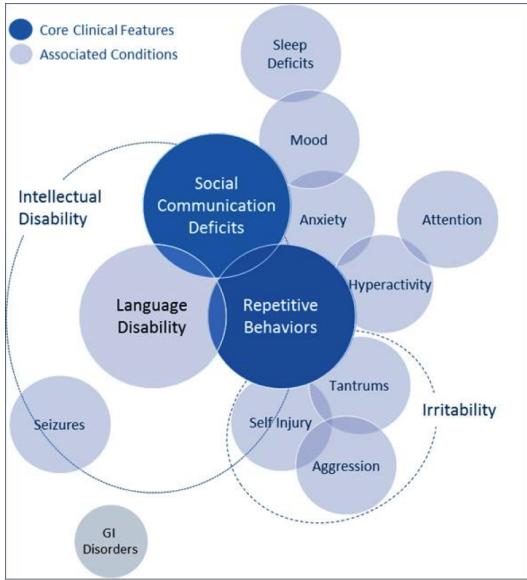


Figure 2.1: Autism Symptoms. Image reproduced from: [4]

PRT the therapist aims instead to stimulate specific vital developmental areas such as self-management and response to cues. Verbal Behavior Therapy was designed to help a child improve its language and communicative abilities. This is done by making children aware of the importance and benefits of using words as a way to express their needs and desires.

Nowadays, there does not exist a recognised medical test for autism diagnostic. Cases are examined individually by doctors for classification. Online screening tools such as Q-Chat are currently available to help parents understand if their child is affected or not by autism [5]. Uses of Machine Learning (to analyse patients EEG readings) and Computer Vision (to detect, from video recording, behavioural and communication impairments) might provide a useful solution to this problem.

2.2 Applied Behavior Analysis (ABA)

Applied Behavior Analysis is a therapy approach based on finding out how learning takes place and how behaviour can be affected by the surrounding environment. Its main objective is to encourage behaviours that are positive for the patient and discourage behaviours that can negatively affect learning or have detrimental effects on the patient. One of the main parts ABA is interested in, is to understand what happens before and after the occurrence of a behaviour. ABA therapy can lead to

an increase in social, communication skills and attention. This type of intervention approach can be used not only to help children (and potentially also adults) with autism but also other types of physical behaviour disorders such as eating disorders. ABA efficacy has been successfully proven thanks to many years of studies accomplished by acclaimed institutions and researchers [6].

2.3 Early Start Denver Model (ESDM)

The Early Start Denver Model is a teaching play based behaviour therapy used to help children with disabilities. This approach is based on ABA, and is aimed at children between 1 to 4 years old. The main objective of ESDM is to help children improve their communication, language and social skills. This is achieved through games and joint activities. ESDM is typically used in both a domestic and clinical setting (additionally also in schools). In this way, both therapist and parents get personally involved in the therapy.

Conducting part of the therapy at home (between children and parents) can also lead to substantial economical savings because it consistently reduces the number of weekly hospital hours of treatment needed (traditionally about 25 hours per week). Additionally, thanks to ESDM, children can also achieve better results because they might feel more comfortable to play at home with their parents rather than with a doctor in a hospital [7]. ESDM approach relies on one to one sessions between therapist and child and personalised intervention tailored according to the needs and improvements of the child.

Studies have shown that ESDM can lead to more substantial improvements in children's IQ and adaptive behaviour compared to typical community-intervention approaches [8].

2.4 Electroencephalogram (EEG)

EEG uses electrodes (small metal disks) placed on the scalp of the patient to track brainwave patterns and send the results to a computer for later analysis. Brain

waves are observed measuring the electrical activity in the brain (electrical pulses of neurons communicating each other). EEG is generally used to identify occurrences such as changes of behaviour and seizures. Brainwaves can be divided in different bands according to their travelling speed (Hertz) [9]:

- Infra-Low: less than 0.5 Hz = basic cortical rythm
- Delta Waves: 0.5 to 3 Hz = dreamless sleep
- Theta Waves: 3 to 8 Hz = sleep and meditation
- Alpha Waves: 8 to 12 Hz = brain resting, flow of thoughts
- Beta Waves: 12 to 38 Hz = standard consciousness waking state
- Gamma Waves: 38 to 42 Hz = love, altruism etc...

Another form of EEG analysis is quantitative EEG (qEEG). QEEG aims to analyse brainwaves using statistic and mathematics. Thanks to a process of quantitative comparison and accurate measurements, qEEG is able to recreate a map of the brain. A qEEG application example can be a research carried out by Lineu C. Fonseca about "Quantitative EEG in children with learning disabilities" [10], which has shown that children who suffer with reading and/or writing disabilities had higher delta activity in frontal-temporal regions than average.

2.5 Serious Games and Gamification

One of the aims of this project was to design a suite of games to help children with disabilities. Games created for a purpose other than entertainment are notoriously known as Serious Games. While playing a Serious Game, players will be likely asked to solve problems. Instead, traditional games mainly focus on action. In order to make learning effective, the main component required is engagement with the subject. Different mediums for learning are nowadays used such as: reading books, articles, manuals and educational videos. Each of these mediums have their advantages and disadvantages; the strength of delivering educational content through a game-based environment is the increase of student's engagement. What contributes to that

are factors such as: the presence of a story-line, challenge and competition against themselves and others to improve their game scores, taking risky decision and paying the consequences for their actions, and learning to follow and understand rules. These concepts have applications in both games and in the real world, users makes mistakes and learn from them to make improvements.

Serious Games can have a wide ambit of application as: health-care, defence, scientific and educational purpose's. Multinational companies have made use of Serious Games during the last few years. Examples include: "Microsoft Flight Simulator", "IBM city one" and "Amnesty the Game" [11]. Another application of game mechanics in non-gaming contexts is Gamification.

In ordinary use, the terms Serious Games and Gamification get confused with each-other. Gamification aims to examine what engages most its potential players and then builds a game based on the gathered data. Designing games suited for each individual according to their needs and abilities can lead to positive behavioural improvements and increase player motivation. In contrast, Serious Games take business concepts and attempts to apply them using gaming techniques. Serious Games are not generally used as an exclusive learning technique but to reinforce understanding [12].

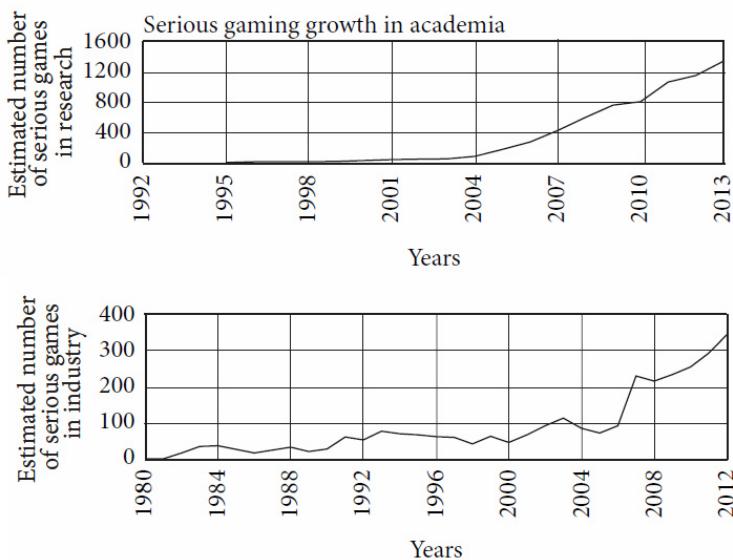


Figure 2.2: Serious gaming applications growth in academia and industry - Image reproduced from [13] - "An overview of Serious Games" - Fedwa Laamarti et al.

2.6 Artificial Intelligence (AI)

The birth of AI goes back to the end of the Second World War, starting with Alan Turing. At that point in time, German forces were using the Enigma Machine to encrypt their messages. Turing, using the Bombe machine, successfully managed to crack the Enigma code. These two machines (Enigma and Bombe) constituted the foundations of Machine Learning. In fact, according to the Turing Test, a machine could be considered as intelligent if it is able to converse with humans without letting them find out its true nature (therefore winning the "imitation game") [14].

Today, Artificial Intelligence has undergone impressive advancements. AI can be subdivided into three different levels according to the ability of machines to perform intellectual tasks logically and independently:

- Narrow AI = machines are more efficient than humans in performing very specific tasks (but not trying to perform other types of tasks).
- General AI = machines are as intelligent as human beings.
- Strong AI = machines perform better than humans in different ambit (in tasks that we might or not be able to perform at all).

Right now we are only at a Narrow AI level. Artificial Intelligence can be broken down in many different categories such as: Machine Learning, Natural Language Processing (NLP), Vision, Robotics and Autonomous Vehicles [15] (Figure 2.3).

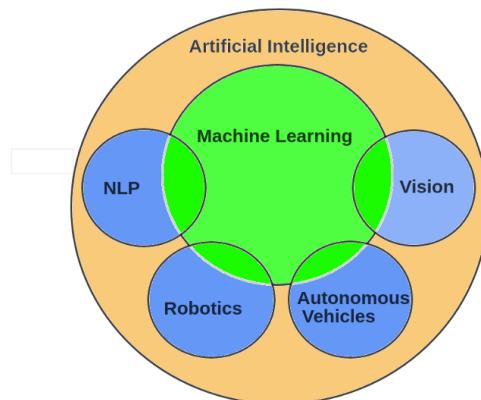


Figure 2.3: Different AI fields. Image reproduced from: [15]

One of the most frequently used AI tools is Machine Learning. There are three main types of machine learning algorithms used:

- Supervised Learning = using a labelled training set to train a model, to then make predictions on unlabelled data.
- Unsupervised Learning = giving a model an unlabelled data-set, the model has then to try to find patterns in the data to make predictions.
- Reinforcement Learning = training a model through a reward mechanism to encourage positive behaviours in case of good performance (particularly used in robotics applications).

The most important sub-field of Machine Learning is Deep Learning (DL). DL is mainly inspired by the structure and function of the human brain and therefore makes use of artificial neural networks. One of the first attempts in this field was the perceptron (which had very limited performance). Successive development of deep neural networks, adding multiple hidden units, using non linear activation functions and back-propagation algorithms lead to far greater results.

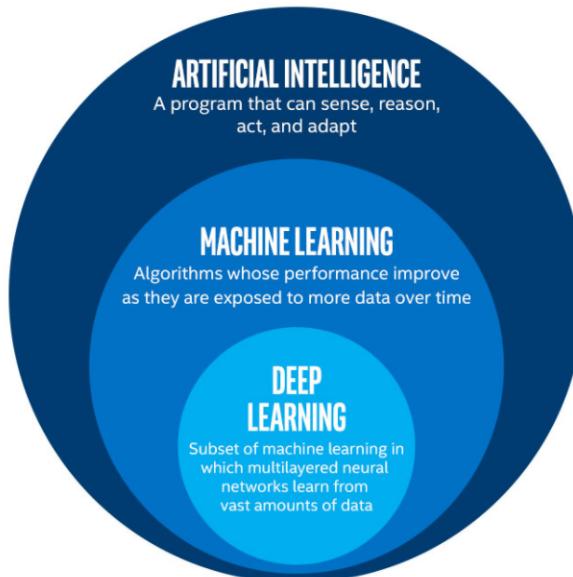


Figure 2.4: AI, ML and DL relation. Image reproduced from: [16]

2.6.1 Long Short-Term Memory (LSTM)

LSTM is a type of Recurrent Neural Network (RNN). RNN were initially developed to make up for the inability of traditional neural networks to work with time series. This is because traditional neural networks, in contrast with RNN, lack of a memory mechanism to remember its past input values (all the inputs are fed in at once and information moves only in one direction; forward). RNN are able to overcome these shortfalls keeping information in a loop, therefore enabling the network to make decisions using not only the current input but also the previous ones (input are added sequentially, one input is added after the previous one has been computed). Therefore, to some extent, a RNN can be considered as a collection of artificial neural networks passing each output from the previous one.

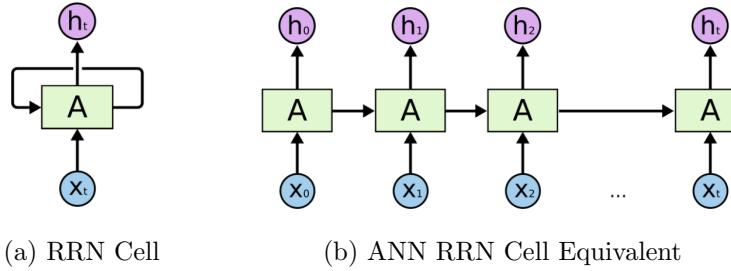


Figure 2.5: Artificial Neural Network equivalent of RNN Cell. Image reproduced from [17]

RNN are although still far from being perfect. They are in fact just able to retain short term information and might suffer from vanishing/exploding gradients when performing the backwards pass. In order to solve these kinds of problem, the LSTM architecture was created. LSTMs are used to tackle time series forecasting, speech recognition and music generation tasks. The difference between an RNN and an LSTM lies in the repeating module (A in Figure 2.5). LSTM makes use of a far more structured architecture (Figure 2.6). Each module has three inputs: the current time step, an input from the previous module and the memory from the previous module. This way, each module can alter or preserve the network memory.

Each LSTM module can be considered to be formed by three gates: input (i_t) ,

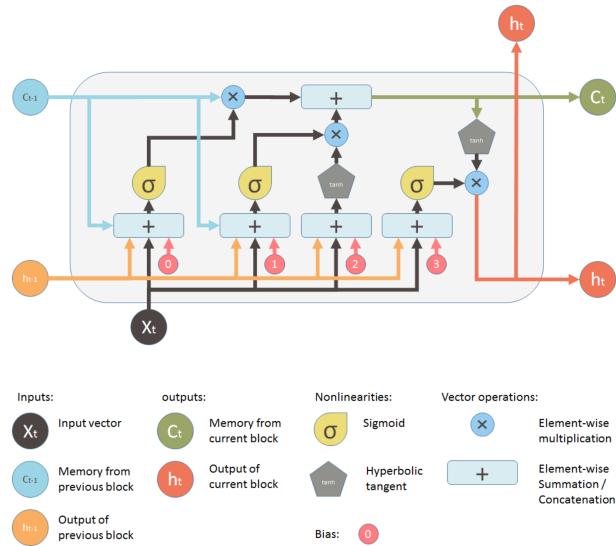


Figure 2.6: LSTM Module. Image reproduced from: [18]

forget (f_t) and output gate (o_t). The input gate decides if a piece of information is important enough or not to be remembered (Equation 2.1). The forget gate (Equation 2.2) decides if a piece of information stored is still relevant or not (and therefore has to be deleted). The output gate determines if a particular information has to have or not a weight in the current time step (Equation 2.3).

$$i_t = \sigma(w_i[h_{t-1}, X_t] + b_i) \quad (2.1)$$

$$f_t = \sigma(w_f[h_{t-1}, X_t] + b_f) \quad (2.2)$$

$$o_t = \sigma(w_o[h_{t-1}, X_t] + b_o) \quad (2.3)$$

where: w_x neuron gate weight

h_{t-1} LSTM previous step output

X_t LSTM current input

b_x gates biases

The Sigmoid function (σ , Equation 2.4) is used to squish the output between any value from zero (making the gate block everything) to one (making the gate pass through everything). A neuron gate weight (w_x) represents the strength of the connection, while the bias (b_x) is used shift the activation function to fit best the data.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

2.6.2 Convolutional Neural Network (CNN)

CNN are a class of neural networks typically used for image recognition/classification, but can be also used for time series analysis. This can be done by converting the time series in a grayscale image like format. The network takes as input a three-dimensional array ($Height \times Width \times Depth$) representing an image. A CNN is composed in two main sections: a feature extraction part and a classification part. During the feature extraction part, convolutions and padding layers are applied to extract the main characteristics embedded in the input vector. In the classification part, fully connected layers (converts our three-dimensional input into a one-dimensional array) are then used applying a Softmax function to classify the detected features.

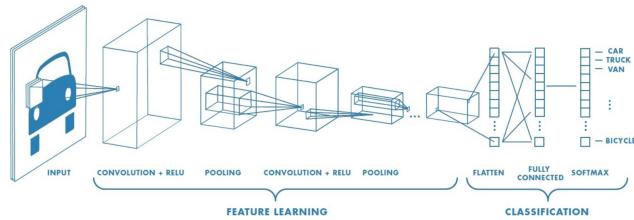


Figure 2.7: CNN structure. Image reproduced from: [19]

In mathematical terms, convolution (\circledast) is an operation between two functions to create a third one, which represents to what extent a function can be modified by another (Equation 2.5).

$$(f \circledast g)' = \int_{-\infty}^{\infty} f(\tau)g'(t - \tau)d\tau \quad (2.5)$$

In the convolution stage, a filter matrix (of smaller dimensions than the input image) is applied to the input image in order to produce a feature map. The filter is made move above the image to cover all the pixels (Figure 2.8). Computing a matrix

multiplication between the input and the filter, and then adding together the results, the single pixels of the feature map can be produced. A stride value is chosen to determine how much the filter moves during each step.

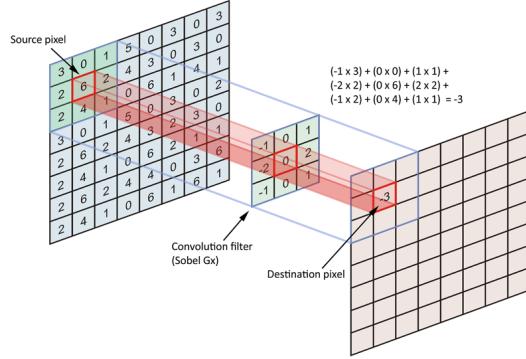


Figure 2.8: Image Convolution. Image reproduced from: [19]

Padding is then performed on the resulting feature map (adding pixels of values equal to zero) in order to avoid having a feature map of smaller size than the input image. Finally, after a convolutional layer a max pooling layer is usually applied in order to reduce the size of the input image, so to avoid overfitting (learning just the main features from the input, not also the added noise) and to speed up the model training.

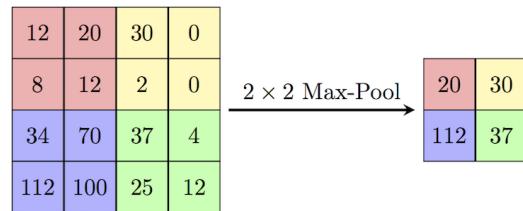


Figure 2.9: Max Pooling. Image reproduced from: [20]

2.6.3 Performance Metrics

There are many types of metrics used nowadays in Artificial Intelligence in order to evaluate a model's overall performance. The three main metrics used in classification problems are:

- Classification Accuracy
- Confusion Matrix

- AUC - ROC Curve

Classification Accuracy is the most used metric. It can be calculated by dividing the number of correct predictions by the total number of predictions made. This metric gives a good understanding of the model performance particularly if the data-set is perfectly balanced (all the classes have the same number of samples).

A Confusion Matrix is a representation of all the predictions made by the model, compared to their real values. In binary classification (for example considering zero as patient being health and one as being ill), the Confusion Matrix is composed of just four cases:

- True Negatives (TN) = both the model and the real output are equal to zero.
- False Positives (FP) = the model predicted one, while the real output was equal to zero.
- False Negatives (FN) = the model predicted zero, while the real output was equal to one.
- True Positives (TP) = both the model and the real output are equal to one.

		Model Predictions		Total
True Outputs	Negative (0)	TN	FP	$TN + FP$
	Positive (1)	FN	TP	$FN + TP$
	Total	$TN + FN$	$FP + TP$	N

Table 2.1: Confusion Matrix

Using the Confusion Matrix, the model Accuracy can be also calculated as:

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (2.6)$$

From the Confusion Matrix it is then possible to calculate a model Sensitivity and Specificity. In a medical context, sensitivity quantifies the model's ability to determine patients effected by a certain medical condition. Specificity instead measures the model ability to correctly identify patients not effected by this condition.

$$Sensitivity(TPR) = \frac{TP}{TP + FN} \times 100\% \quad (2.7)$$

$$Specificity(TNR) = \frac{TN}{TN + FP} \times 100\% \quad (2.8)$$

The AUC (Area Under The Curve) - ROC (Receiver Operating Characteristics) curve evaluates a model's ability to correctly discriminate between the different classes using different thresholds.

Plotting the False Positive Rate (Specificity) against the True Positive Rate (1 - Sensitivity), the ROC Curve can be calculated (Figure 2.10).

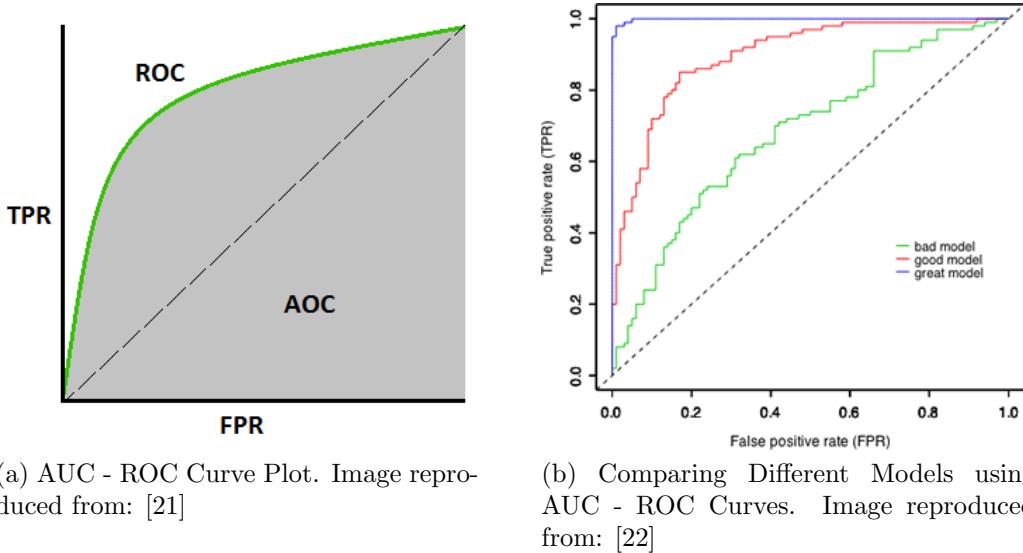


Figure 2.10: AUC - ROC Curve Examples

An AUC of 0.5 would represent the model is randomly guessing between the different classes (as shown in Figure 2.10 (a) by the black dotted line) and an AUC of one would instead show the model is making a perfect classification. If the AUC score would be less than 0.5, that would mean the model is doing exactly the opposite of what it has been designed to do.

The ROC curve can be additionally calculated (in a binary classification case) from the data classes probability distributions, calculating the Sensitivity and Specificity (Figure 2.11). Choosing an arbitrary threshold value, the True Negatives will be equal to the Class 1 area on the left side of the threshold. The True Positives will

instead be equal to the Class 2 area on the right side of the threshold. The addition of True Negatives and False Positives is represented by the total area covered by Class 1 while the addition of True Positives and False Negatives is represented by the area covered by Class 2.

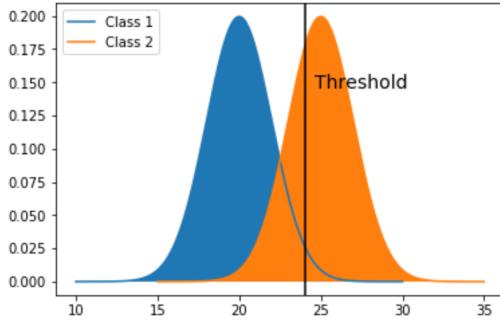
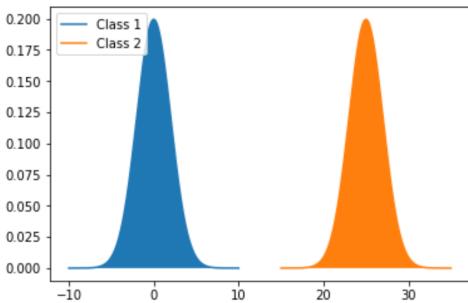
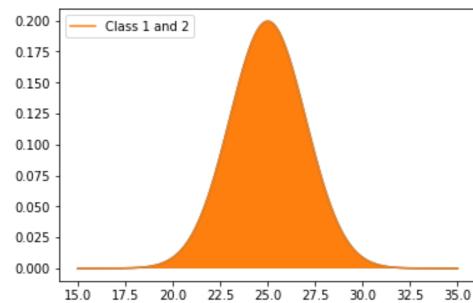


Figure 2.11: Classes Probability Distributions

The AUC would yield better results (closer to 1) if the intersection between the two probability distributions is as low as possible. In fact the model False Positives can be represented as the Class 1 area on the right of the threshold and the False Negatives as the Class 2 area on the left of the threshold. If there are no points in common between the distributions (the two output classes are perfectly distinguishable one from the other) then the AUC would be equal to 1 (Figure 2.12 (a)). As shown in Figure 2.12 (b) if the two distributions are instead perfectly overlapping (having all their points in common), the AUC would be equal to 0.5 (the model is randomly guessing between the two classes).



(a) Perfectly Separated Distributions



(b) Perfectly Overlapping Distributions

Figure 2.12: Probability distributions and AUC relationship

Other types of classification metrics commonly used are: Logarithmic Loss and Classification Report.

2.6.4 Ethics

AI is now used in many fields. Applications in sectors such as medicine and self-driving vehicles, are raising concerns for private individuals as well as public authorities.

This is because, this technology can be used not just as a tool to help humans perform tasks, but also take the place of humans themselves in carrying them out. This can be applied not just for repetitive tasks (such as industry automation), but to more decision making intensive tasks (such as medical prescriptions).

There is now therefore a sense of urgency for Explainable AI. Creating models able to explain their conclusions will reassure the public about their trustworthiness and enable developers to understand if they are affected by bias.

Companies such as Google, have decided to create a list of ethical principles to be followed when creating AI models [23]. AI should:

- ”1. Be socially beneficial.
- 2. Avoid creating or reinforcing unfair bias.
- 3. Be built and tested for safety.
- 4. Be accountable to people.
- 5. Incorporate privacy design principles.
- 6. Uphold high standards of scientific excellence.
- 7. Be made available for uses that accord with these principles.”

Chapter 3

Gaming Platform

3.1 Hardware implementation

In regard to a hardware solution, to help children's with disabilities, I decided to make a projector using a DLPDLCR2000EVM Evaluation Board and a Raspberry-Pi 3B (running the Raspbian operating system). Succesively, I managed to then make the area projected touchscreen using a Wiimote (Wii Remote controller [24]) and an Infrared (IR) pen I built myself. The Wiimote is connected via Bluetooth (to the Raspberry-Pi) and is used as an infrared receiver to detect the activities of the infrared pen on the projected surface. Finally, I programmed a range of games that can be loaded on the Raspberry-Pi for children to play.

In order to wire and configure the connections between the Evaluation Board and the Raspberry-Pi, I followed an online guide by Frederick Vandenbosch [25] (Appendix B). I then set up the Raspberry Pi Bluetooth (to enable the Wiimote connection) by connecting the Bluetooth adapter and then installing the Bluez and Wiimote integration packages through Linux terminal. Lastly, I installed a free python-whiteboard software by Martin Rudel [26] to be used in order to calibrate the surface of the projected area.

It is also possible to implement this solution on Windows using a standard projector and the free whiteboard calibration software Smoothboard Air with Duo [27].

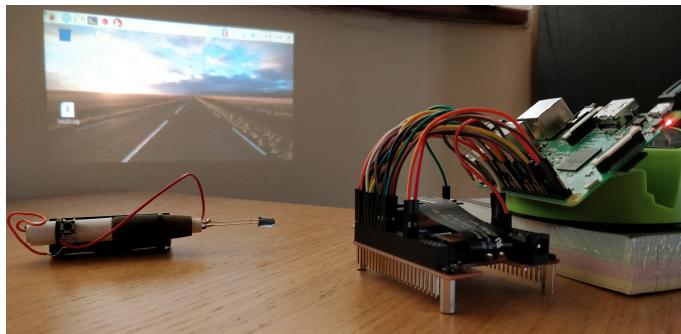


Figure 3.1: Infrared Pen, DLPDLCR2000EVM Evaluation Board and Raspberry Pi

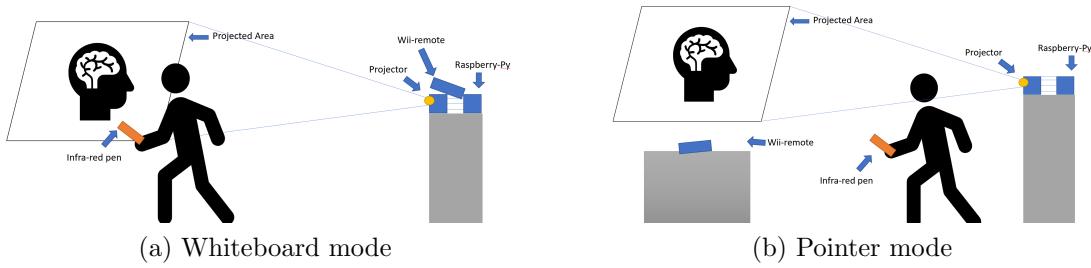


Figure 3.2: Touchscreen projector playing modes

Figure 3.2 shows the two different playing modes which can be used. In the first one, the projected area can be used as a whiteboard, while in the second one the user can choose to play far from the surface using the infrared pen as a pointer. Choosing the playing mode to use is important to pick the right calibration position of the Wiimote.

This hardware interface was mainly designed to help children affected by Developmental Coordination Disorders. This type of motor disorder causes coordination problems (eg. wrong movement timing/balance) due to incorrect communication between the brain and the body.

3.2 First Game Development

I decided firstly to make a game designed for children with cognitive disabilities to test both their rate of reception and their ability to work with others. In order to create this game I made use of references such as "Python game development" [28] and "Beginning Game Development with Python and pygame" [29]. I decided to program this game using Python as programming language and pygame as my main working library.

Figure 3.3(a) shows the menu interface I designed for the game. I decided to name the game "Avalanche Escape", since it consists of moving characters around the graphical user interface (GUI) to avoid falling snowflakes. The GUI can be considered as a Cartesian coordinate system. During the game, the snowflakes fall from the top of the window, always changing their Y axis starting position. The snowflakes velocity slowly increases every-time the character successfully manages to avoid them (Figure

3.3(b)). This way the game will seem relatively easy to play at the beginning but as the user practises and gets better at playing it, the game difficulty will also increase. It is possible to read the game instructions using the game instruction button present on the menu. In the menu the user can choose to play in either single or multi player

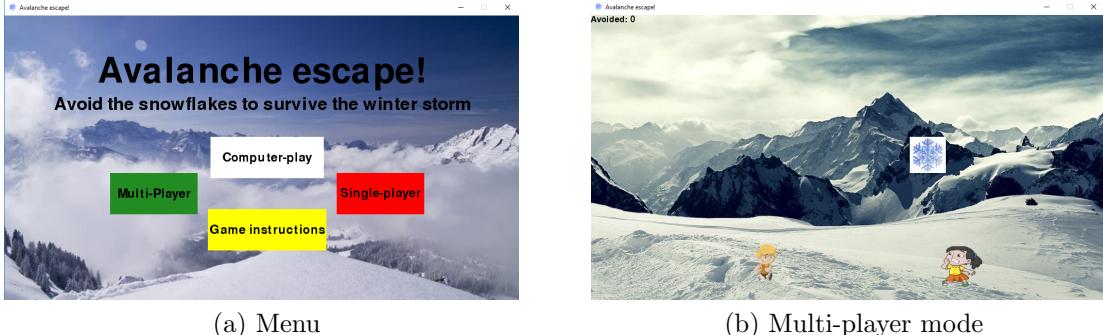


Figure 3.3: Avalanche escape

mode. In single player mode, the male character is moved by either using the mouse or keyboard arrows. In multi player mode, the female character is instead moved by using the mouse and the male one by using the keyboard. If one of the characters is hit by a snowflake, a menu will prompt warning of the collision and ask if the user wants to play again or go back to the main menu. If the user manages to avoid the predetermined number of snowflakes asked by the task, a menu will appear on the screen congratulating the user for winning. Each of these actions are accompanied by sounds which are correspondent to whether the user wins or loses the game in order to increase user engagement. Throughout the whole playing time, there is music playing in the background.

The user can pause the game at any time by pressing the keyboard escape button and switch between full-screen display mode (pressing F) and standard screen mode (pressing N). A computer play mode is also present which can be used to demonstrate to the user how to play. In order to implement the computer play mode I used a simple algorithm. Using Machine Learning instead of traditional coding could potentially lead to better gaming performance and score.

Finally, I made an executable version of this game which can be easily installed on any type of Windows platform using the cx_Freeze python library.

3.3 Imitation Games

Imitation is the ability to mimic other people behaviours. Children usually start imitating others from around the age of two and to mimic different types of behaviour at different age stages. The ability to imitate is not something inherently transmitted but is instead the result of our human social life organisation.

Studies demonstrate that children with autism usually suffer delays in their Imitation and Joint Attention abilities. Imitation skills can play a fundamental role in a child's behavioural development. During this project, I decided to design different games to try to stimulate and enhance these kind of abilities.

3.3.1 Second Game Development

This game is divided into two different modes (Figure 3.4). When selecting the first mode, the child is asked to recognise the animals in the pictures (Figure 3.5 (a)). Clicking on each of the different buttons an audio will play the sound of the corresponding animal and a message box will appear. The child will then be asked to type in a text-box the appropriate animal name. If the child types the right animal name (it doesn't matter if the name is written in all lowercase, all uppercase or a mixture of them) a congratulations message box would appear. If instead the child types the wrong answer, it would be encouraged to try again.

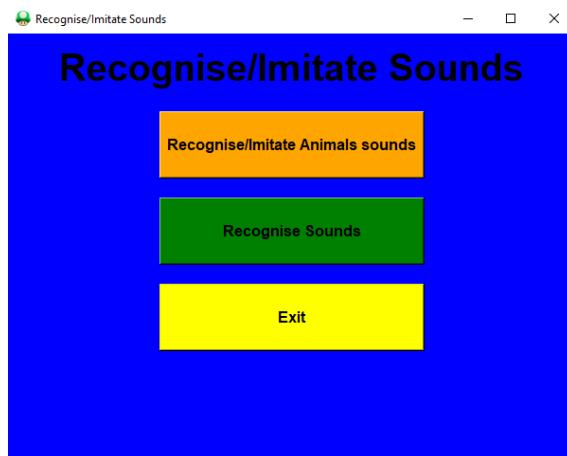


Figure 3.4: Game 2 Menu

When selecting the second mode (Figure 3.5 (b)), the child is instead asked to recog-

nise and then write the animals names, this time by just hearing without corresponding images (therefore increasing the game difficulty). When playing either of the game modes the child will be encouraged by the parent/therapist to try to talk about the identified animals and to imitate their sound.

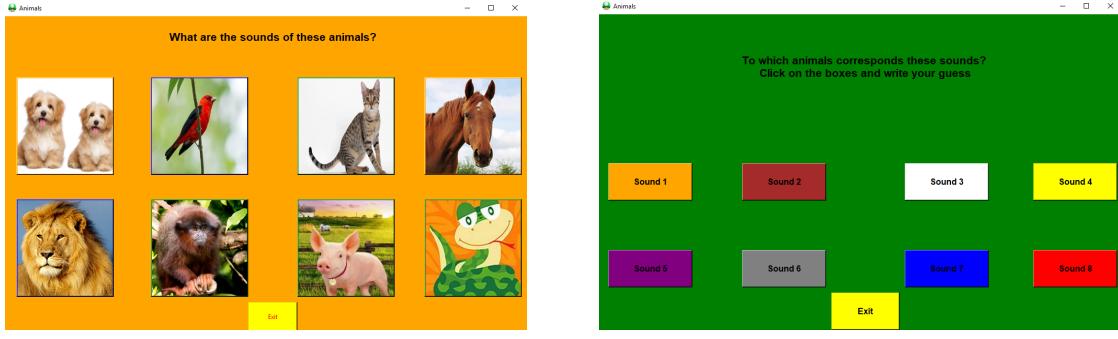


Figure 3.5: Game Mode Options

This game therefore aims to improve a child's ability to recognise and imitate animal sounds in the first mode and in the second one to guess (encouraging them to first take a risk and then learn from his/her mistakes) and imitate animals sounds. Different levels of difficulty can eventually be designed as further development of this game. The modes designed so far are aimed to not be too difficult, in order to try to not discourage a child from playing (if the game would have been too difficult from the start, a child might lose interest).

This game was designed using Python and libraries such as Tkinter, pygame and PIL.

3.3.2 Third Game Development

The aim of this game is to encourage children to imitate the therapist/parent's actions and to build objects/shapes (Figure 3.6). The game GUI consists of two canvases which can interchangeably be used by either the child or therapist/parent. Selecting the correspondent buttons, the users would be able to make appear on the screen up to six blocks which they can drop and drag around the screen to create objects or shapes. If an user tries to place a block either outside the game window or in the canvas of the other player, he/she will be stopped by a warning prompt

asking him to try again. While playing this game, the parent/therapist should build one or more object(s) using the given blocks on one canvas and the child should then try to imitate the same object(s) on his canvas either while the parent/therapist is making it or after.

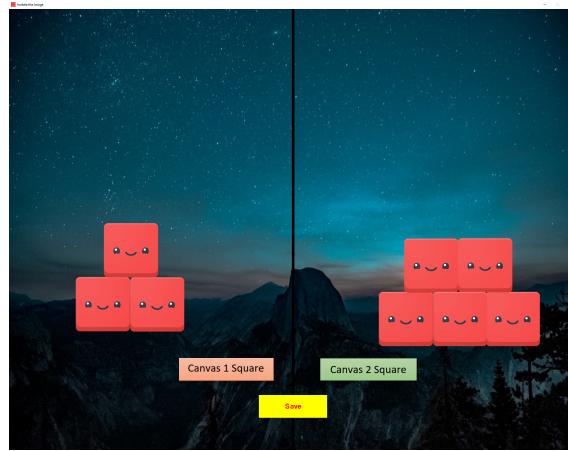


Figure 3.6: Game 3

Once the child has come up with his solution to this problem, it is then possible to store the result by clicking the "Save" button on the GUI. A screen-shot will then be generated and stored as a record of the child's performance. Keeping a record of all the different attempts the child makes while trying to build different types of objects, is then possible to see whether the child has improved over time and understand if the child has a problem imitating certain specific shapes or patterns.

I also decided to implement an EXE version of this game to make implementation easier. This game was designed using Python and libraries such as pyoutogui, pygame and time.

3.3.3 Fourth Game Development

This game was designed to help children develop their recognition and matching skills solving puzzles (Figure 3.7).

I decided to design this game using Unity for Graphic Development and C# as programming language for coding the graphics behaviour. I decided to opt for this solution because it enabled me to make available this game not just for desktop platforms (like in all the other games made so far), but also on Android devices (eg. mobile phones and tablets). This was made possible by creating both an EXE



Figure 3.7: Game 4 Menu

and APK format of the game. Both versions have been successfully tested on both platforms.

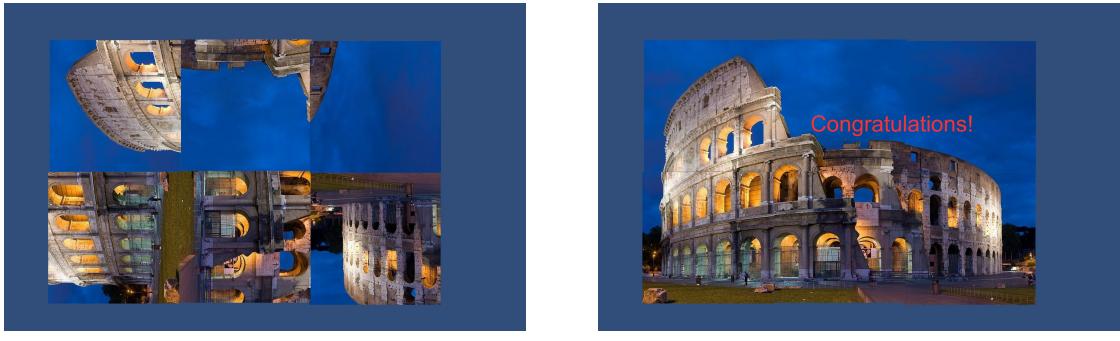


Figure 3.8: Game Mode

Clicking (or tapping using a touchscreen Android device) on each of the images that composes the puzzle makes them rotate by 90°clockwise (Figure 3.8 (a)). Once the child has selected the right orientation for each of the puzzle images, a congratulations text message appears on the screen and the user is redirected to the main menu where they can decide if to play again or end the game (Figure 3.8 (b)).

3.4 Joint Attention Games

Joint attention (JA) is the ability of two individuals to share their interest on the same event, person or object. This implies an underlying understanding between the two people in question. This can be achieved through any form of verbal or non-verbal indication such as pointing or eye-gazing. In JA, a child follows the lead of the parent/therapist and tries to copy or mimic them. JA allows children to share

enjoyment and learn from others.

Children usually start showing the first forms of JA around the age of 9 months old. Children affected by autism symptoms might instead show JA abilities later on or, in some cases, might never fully develop them. To external viewers, this might look like a lack of engagement in collaborating and interacting with other people.

There are two main forms of JA:

- Child initiating JA
- Child responding to call of JA from another person

3.4.1 Fifth Game Development

I decided to structure this game with four different levels of difficulty (Figure 3.9). This game consists of identifying the right objects between the given options (there can be either one or more right answers) and recording the child's justification for their decision. It's main application is to improve child's receptive communication.

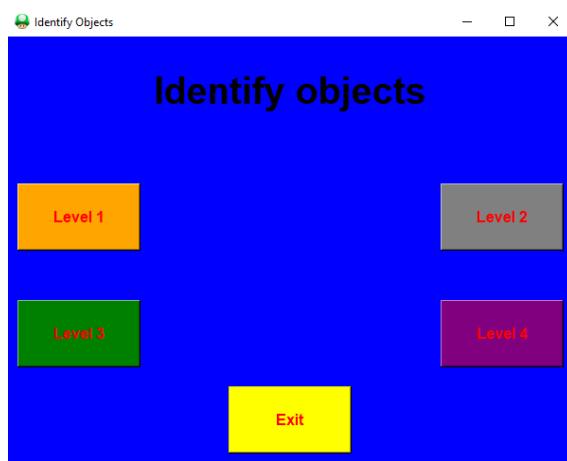


Figure 3.9: Game 5 Menu

If the child selects the right answer, a congratulations prompt will appear. The prompt will inform the child he/she made the right choice and will then inform them that closing it will start the recording. The recording will continue until the child stops speaking. For each time the child selects the right answer, the recording will be stored and could then be used for later analysis by the therapist. The recording will play a vital role in evaluating the understanding of the child and track any signs

of improvement. If the child selects instead a wrong answer, a prompt will encourage him/her to try again (Figure 3.10 (a)).

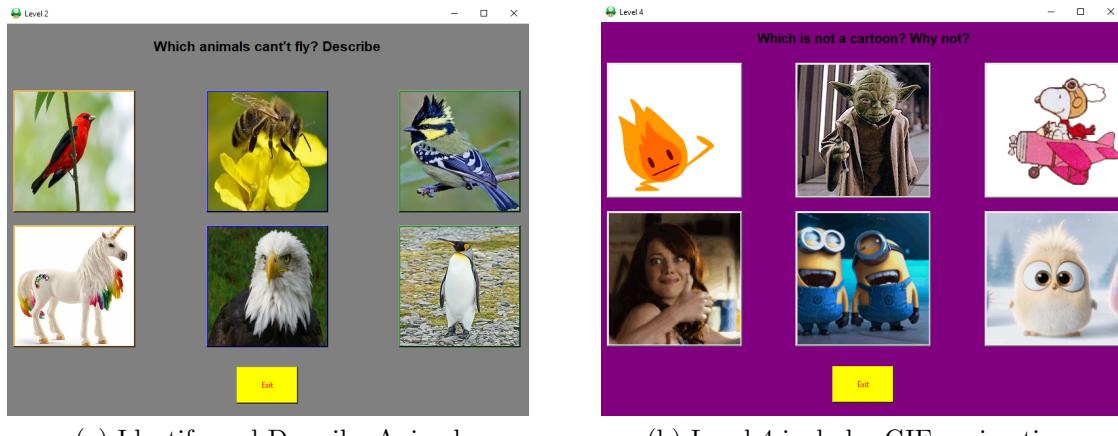


Figure 3.10: Game Mode Options

For the fourth difficulty level of this game, I decided to make the buttons interactive using instead of plain images, animated GIFs. Doing so I aimed to make the game more stimulating and interactive for the child (Figure 3.10 (b)).

This game was designed using Python and libraries such as Tkinter, PIL, itertools, pyaudio and wave.

3.4.2 Sixth Game Development

This game aims to help children improve their receptive communication, in particular their understanding of quantities and numbers. I decided to design this game using Microsoft Visual Studio and C#, which facilitated the creation of the final EXE file.

The menu interface (Figure 3.11 (a)) offers apart from the main buttons also a top drop down menu which can give additional help for the child to learn about the game. At the bottom of the GUI is also present a status bar, offering additional information to the user about the meaning of the different menu options (once clicked on them) and to keep the user updated about the status of the game.

As shown in Figure 3.11 (b), during each game level are tested the child's abilities to identify the button showing the asked right number of objects displayed in the pictures.

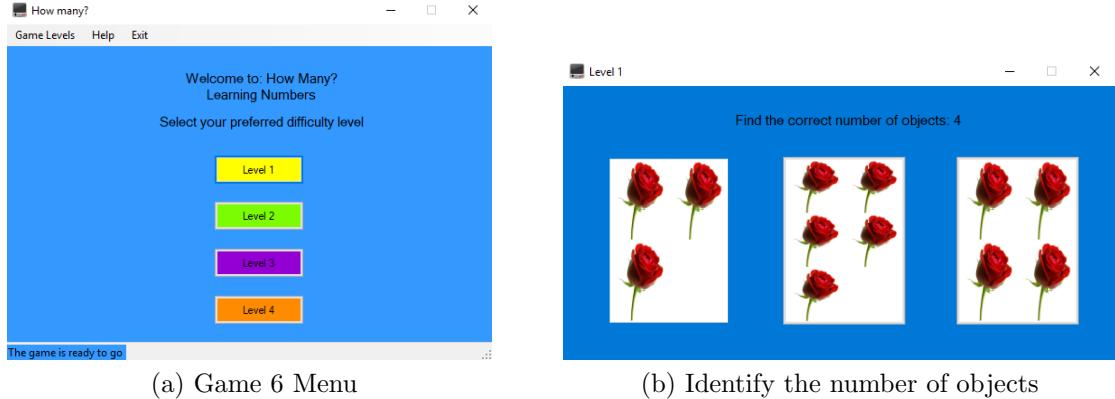


Figure 3.11: Game Six Options

3.4.3 Seventh Game Development

The seventh game I decided to design was a drawing canvas for two users (Figure 3.12). I decided to program this game using Python and the Tkinter graphics library. The two canvases consist of two independent drawing surfaces. This way, it is either possible for the user to make free drawings when playing on their own or to let the therapist/parent draw something on one of the two canvas and then let the child imitate their drawing on the other canvas. It is additionally possible, when playing with the therapist/parent, to let the fist write or pronounce a word and then let the child draw the corresponding image. This type of game aims to help children learn through imitation and respond to external stimulus.

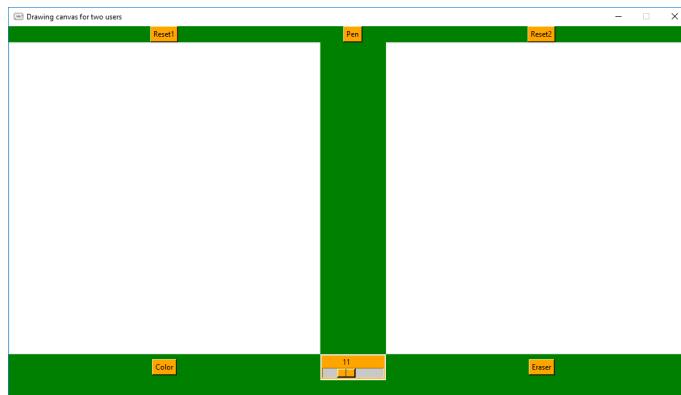


Figure 3.12: Drawing canvas for two users

As can be seen from Figure 3.12, the GUI gives the users the choice to: select their preferred font size (bottom centre slider), favourite colour (left-bottom button), switch between pen and eraser mode (top centre button, right bottom button) and

to clear the canvases (top left button for left canvas and top right button for right canvas).

3.5 EEG Cap

Having designed a suite of games and tested they functioned correctly using them with the hardware implementation, I started working to connect, synchronise, and store the data from the EEG Cap. In my case, I have been working with the Enobio 20 (produced by Neuroelectronics [30]), a wearable, wireless sensor system used for recording electroencephalogram (EEG) signals.

In order to connect and control the Enobio 20 using my laptop, I first had to install the Neuroelectrics Instrument Controller (NIC) free software offered by Neuroelectronics and then connect my laptop to the Enobio 20 via bluetooth using the NIC. Once I made sure everything was set up between my laptop and the Enobio 20, I started working on trying to control the Enobio 20 calling a MATLAB script instead of having to use the NIC. To do so, I made use of the MatNIC MATLAB toolbox produced by Neuroelectronics [31].

Making a MATLAB script calling some of the many available MatNIC functions, I managed to automate the process of: opening the NIC software, connecting the Enobio 20 via bluetooth, starting the EEG readings for a chosen predetermined amount of time and output the final reading results in a comma-separated values (CSV) file format.

When calling the MATLAB script, it is still necessary to press "Enter" to allow the NIC software to be controlled remotely. This has been included by Neuroelectrics for security reasons, in order to avoid giving unauthorised access to the EEG Readings.

Using the MatNICEEGRecord function (keeping TCP connection enabled), the CSV file is composed by as many columns as the channels of the device (in this case 20) and two more columns, one for the markers (if no markers have been sent, this column will be composed just by zeroes) and one for the timestamp. If instead the MatNICEEGRecordLSL function is used (through Lab Streaming Layer communi-

cation), the CSV file is composed by as many columns as the channels of the device (in this case 20). The units of the measurements for each channel are in nano-Volts (nV).

Finally, I managed to synchronise the seventh game I developed with the EEG readings calling the MATLAB script I made, using Python. I therefore modified the code for the drawing canvas for two users by adding the `matlab.engine` library to call the MATLAB script file and using the threading library to make the canvas game and the EEG reading run simultaneously (without dividing the code into threads they would have run sequentially, compromising synchronisation).

The main functions described above are available in Appendix E, Listings 1 and 2.

Chapter 4

Artificial Intelligence

As part of this chapter, the analysis of an EEG data-set to identify if a child is affected or not by autism will be outlined. Both Machine Learning (ML) and Deep Learning (DL) approaches used to tackle this task will be examined. Additionally, a Computer Vision based approach to ASD screening will be proposed.

4.1 Faces Responses in Children with ASD

Children affected by ASD usually show difficulties in comprehending other people's emotions. A study carried out in 2012 by Fabio Apicella, Federico Sicca et al. ("Fusiform Gyrus responses to neutral and emotional faces in children with Autism Spectrum Disorders: a High Density ERP study" [32]) aimed to register the EEG response of a group of children (some of which were affected by ASD and some who weren't) to examine their reaction to different forms of stimuli (eg. Happy/Sad/Neutral faces, and images of Cartoons and Trees).

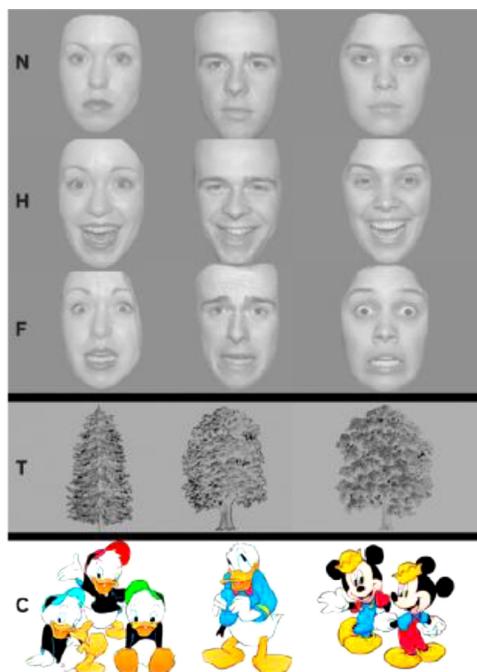


Figure 4.1: Stimuli used. Image reproduced from: [32]

Informed consent was given by all the children parent's before undertaking any mea-

surement. Upon Ethics Approval (Appendix F), I made use of the EEG data (in the time-domain) collected during this study to determine whether it was possible to accurately classify which children were affected or not by ASD using Machine Learning. The data-set consisted of the EEG data (for all the five stimuli repeated multiple times) collected from twelve children affected by ASD and twelve who weren't.

The data originally provided was in a MATLAB format. After a first exploratory analysis, I then decided to convert each file in a CSV format and then merged them so to perform my analysis using Python. I ended up making six CSV files, one for each of the five stimuli and one containing all the five stimuli together.

For each child, the MATLAB data was stored using a three dimensional array. The first dimension represented the number of EEG channels (128), the second dimension represented the number of time-steps (250) and the third dimension the number of repetitions (between 20 and 80, depending on the child). Once the data had been converted in the CSV format, I structured it in a table. This table has 129 columns (one for each channel and one to label the data as either typical or ASD), every 250 rows was stored one stimulus repetition with their correspondent label.

The data-set containing all the five stimulus, had been demonstrated to be balanced, containing 49.03% of the data of children without ASD and 50.97% of the data of ASD children.

All the plots obtained during a first exploration of the data using MATLAB are present in Appendix G. The ML and Deep Leaning results obtained using the individual data-sets for each stimulus are instead present in Appendix H.

4.1.1 ML Classification

Once I constructed the full data-set containing all the five stimuli, I then decided to try to perform a classification task using Machine Learning. In order to do so, I made use of many Python libraries such as: pandas, numpy, matplotlib.pyplot, sklearn and itertools.

The pre-processing part consisted of, firstly loading the data, standardising it and

then dividing it in training (70%) and test sets (30%) for both the inputs (X) and outputs values (Y). The total number of rows in the data-set was equal to 1906500 (934750 rows about typical children EEG data and 971750 about ASD data). Because of the 70% against 30% train/test split ratio, 5338 predictions were made during the training set (1334550/250 time-steps) and 2288 predictions during the test set (571950/250 time-steps).

I finally decided to train and then test the binary classification results using different algorithms such as: Logistic Regression, Support Vector Machines (SVM), Decision Trees, Linear Discriminant analysis (LDA) and Gaussian Naive Bayes Classifier (GNB). The classification results are show in Table 4.1. At the output, a zero represent a typical child and a one represents a child affected by ASD. The models used attempted to identify if a child is affected or not by ASD using just a single stimulus repetition ($128\text{channels} \times 250\text{timesteps} \times 1\text{repetition}$).

Classifier	Accuracy (%)
Logistic Regression	53
Support Vector Machines	53
Decision Tree	80
Linear Discriminant Analysis	53
Gaussian Naive Bayes	60

Table 4.1: ML Classification Accuracy

The Decision Tree achieved the most accurate result. Decision Tree is a type of supervised learning algorithm which can be used for either classification or regression tasks. What distinguishes Decision Tree from other ML algorithms, is their ability to display their decision making process using an upside-down tree like representation. In the tree, each node represents a feature in the data-set, each branch a decision rule and each leaf a decision outcome. In this implementation, I decided to use the CART (Classification and Regression Trees) algorithm implementation which makes use of the Gini Index as metric (code available in Appendix E, Listing 3). Appendix I shows the beginning of the decision making process made by the Decision Tree to decide its classification criteria for this data-set.

Performing PCA (to reduce data dimensionality to two dimensions) on the data lead

to a sharp decrease in accuracy (51% accuracy at maximum using a Decision Tree).

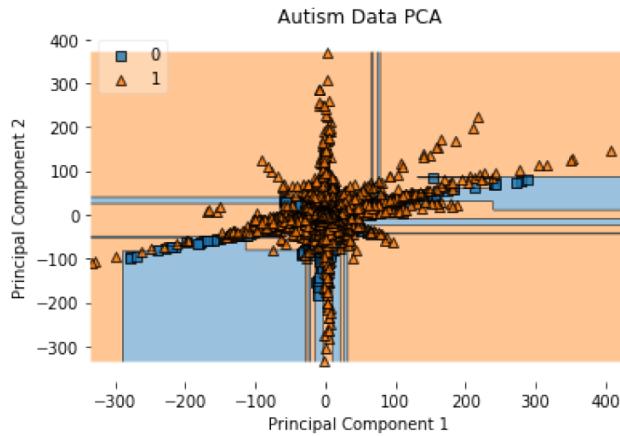


Figure 4.2: PCA Decision Tree Classification Results (0=Typical, 1=ASD)

The trained Decision Tree model has then been successfully saved and stored (using the pickle library) to be ready for later use.

4.1.2 Deep Learning Classification

To increase accuracy of results, I then decided to implement Neural Networks (LSTM and CNN). I preferred to use these typologies of ANN instead of a Feed-Forward one because of their ability to work with sequential data. Most of the libraries used for ML Classification have been reused for the Deep Learning implementation, also the train/test ratio split has remained unaltered.

4.1.2.1 Long Short-Term Memory

In order to code this Neutral Network I decided to use the Google Open Source Deep Learning library Tensor-Flow. After careful literature background research, I decided to design the ANN architecture following the "Human Activity Recognition using LSTMs on Android" Medium guide [33]. For this implementation, I decided to change the pre-processing stage and some of the architecture parameters in order to best suit the data-set shape and characteristics (pre-processing code available in Appendix E, Listing 4) .

Letting training this model for 50 epochs, using the parameters shown in Table 4.2, lead to an impressive overall validation accuracy of 96.935 % (The full training

Parameter	Number
Time Steps	250
Features	128
Steps	25
Classes	2
Hidden Units	32
L2 Regularization	0.0015
Learning Rate	0.0025
Epochs	50
Batch Size	256

Table 4.2: LSTM Parameters

accuracy and loss evolution is present in Appendix J). From Figure 4.3, it can be witnessed that both the training/test loss and the training/test accuracy curves closely match. This confirms the correct L2 regularisation (Ridge Regression) parameter was chosen.

L2 regularisation aims to eliminate over-fitting by minimising the model complexity and loss. By using L2 regularisation, the model complexity is evaluated by summing and squaring all the model feature weights. One of the easier ways to identify overfitting in a model is by comparing the training and test sets loss functions. If the two are really similar, our model is successfully able to generalise to new data without losing accuracy. If instead the two functions are very dissimilar, our model might be able to perform well in the training phase but not during the test phase. Indicating that during the training our model overfitted the data learning also irrelevant information (such as noise) and was not able to correctly classify new data (Bias-Variance Trade-off).

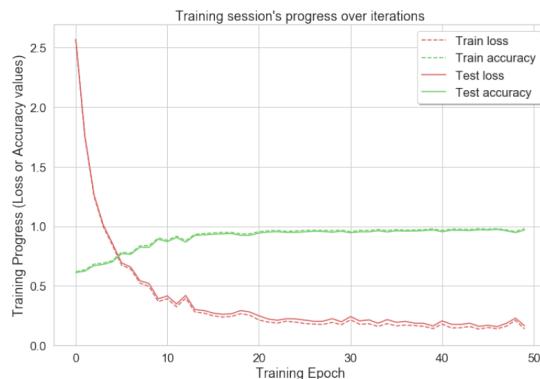


Figure 4.3: LSTM training/test loss and accuracy against epochs

On the other side, the LSTM required a considerable amount of memory and time to train which could potentially become a problem if working with a greater amount of data. To avoid having to retrain the model, it was successfully saved and stored using the pickle library. Different metrics have been performed to test the overall efficiency of the model such as calculating the Confusion Matrix and AUC-ROC curve.

	Negative (0)	Positive (1)	Total
Negative (0)	1093	29	1122
Positive (1)	22	1144	1166
Total	1115	1173	2288

Table 4.3: LSTM Confusion Matrix

From the Confusion Matrix it is then possible to calculate the model Sensitivity and Specificity.

$$Sensitivity(TPR) = \frac{TP}{TP + FN} \times 100\% = \frac{1144}{1166} \times 100\% = 98.1\% \quad (4.1)$$

$$Specificity(TNR) = \frac{TN}{TN + FP} \times 100\% = \frac{1093}{1122} \times 100\% = 97.4\% \quad (4.2)$$

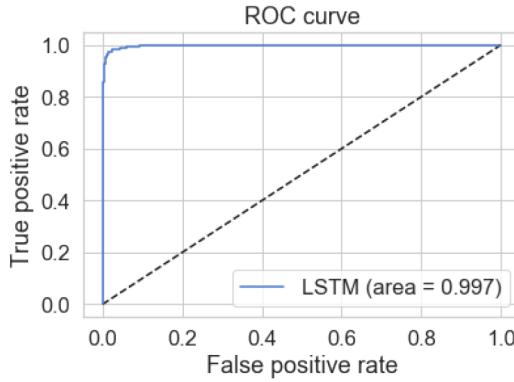


Figure 4.4: LSTM ROC plot

All these metrics confirmed that the model did not overfit the data as seen in Figure 4.3. In fact:

- The ROC curve score was close to 1 (0.997).

- The number of False Positives and False Negatives in the Confusion Matrix were close to each-other (therefore the model gave the same weight to both classes).
- Both the Sensitivity and Specificity results were higher than 97%.

4.1.2.2 Convolutional Neural Network

I finally decided to design a CNN in the hope to construct a faster model than the LSTM that would still able to achieve high accuracy. In order to construct the network architecture, I decided to use the Keras Python library.

The model consisted of:

1. One 2D Convolutional Layer of 64 filters, a kernel size of 5×5 , a ReLU (Rectified Linear Unit, Equation 4.3) function and same padding.

$$f(x) = \max(0, x) \quad (4.3)$$

2. Another 2D Convolutional Layer having 32 filters, a kernel size of 5×5 , a ReLU (rectified linear unit) function, same padding and an L2 regularisation coefficient of 0.01 (to prevent overfitting).
3. A 2D MaxPooling layer of 2×2 size.
4. A Dropout layer of 0.2 intensity (in order to avoid over-fitting the data).
5. A layer to first flatten the data from three dimensions to just one, and then another one to condense the input to give to the classifier 128 features (always using the ReLU function and L2 regularisation).
6. A second Dropout layer of 0.5 intensity.
7. Finally, a Dense layer (of two neurons) to produce the classification result, using a Softmax activation function. The Softmax function (Equation 4.4) will take in this case the two inputs from the neurons and convert them in two probabilities that sums up to one. The greater probability was rounded

up/down to either one or zero to represent the CNN output choice (Typical(0), ASD(1)).

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^k e^{x_k}} \quad \text{for } j=1, \dots, k \quad (4.4)$$

In order to optimise the training, the Adam (Adaptive Moment Estimation) gradient descent algorithm was used, and the cross-entropy function was used to compute the model loss. The cross-entropy function ($H_{y'}(y)$), in a binary classification case can be calculated by using Equation 4.5.

$$H_{y'}(y) = \sum_i (y'_i \log(y_i) + (1 - y'_i) \log(1 - y_i)) \quad (4.5)$$

where: y'_i expected model output

y_i real model output

This model achieved an overall validation accuracy of 94.58% in just thirty epochs of training.

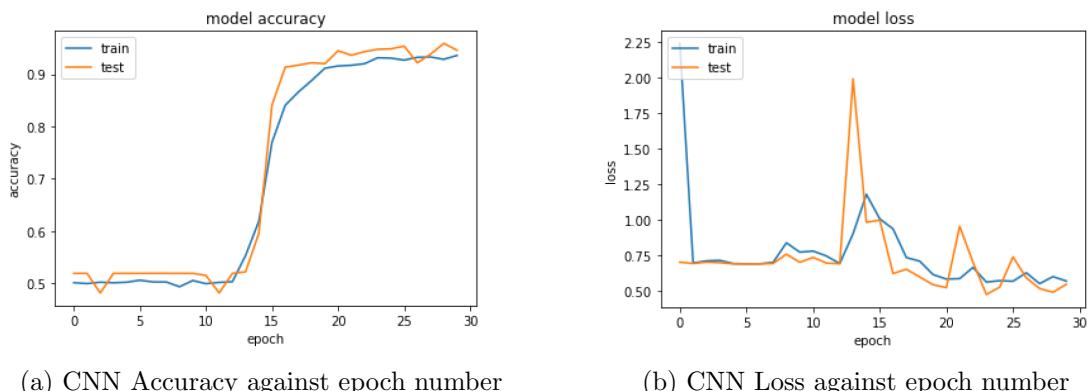


Figure 4.5: CNN Model Training

Both the LSTM and the CNN are able to classify if a child is affected or not by ASD just by observing one stimulus repetition (128 channels, 250 time-steps, 1 repetition, 1 stimulus). The overall accuracy of the CNN was 2% lower compared to the LSTM one, but the training time and amount of power needed to run this model was much lower. Also this model was successfully saved and stored (in a json and h5 file type)

for later use (Code available in Appendix E, Listing 5).

Also in this case different metrics have been performed to test the overall efficiency of the model such as calculating the Confusion Matrix and AUC-ROC Curve.

	Negative (0)	Positive (1)	Total
Negative (0)	1001	100	1101
Positive (1)	24	1163	1187
Total	1025	1263	2288

Table 4.4: CNN Confusion Matrix

From the Confusion Matrix it is then possible to calculate the model Sensitivity and Specificity.

$$Sensitivity(TPR) = \frac{TP}{TP + FN} \times 100\% = \frac{1163}{1187} \times 100\% = 97.9\% \quad (4.6)$$

$$Specificity(TNR) = \frac{TN}{TN + FP} \times 100\% = \frac{1001}{1101} \times 100\% = 90.9\% \quad (4.7)$$

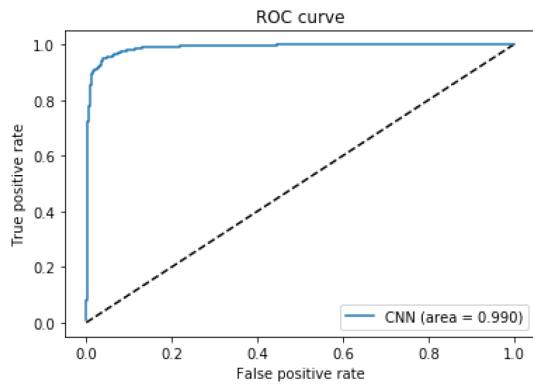


Figure 4.6: CNN ROC Curve

For the same reasons examined in the LSTM analysis, the CNN model does not seem to be affected by over-fitting. However, the CNN performed overall worse than the LSTM. That's because of the bigger difference between the registered False Positives and False Negatives in the Confusion Matrix, which caused the Specificity score to be negatively affected.

4.2 Autism Study Using Behaviour Imaging

Computer Vision is an AI area which has seen exponential improvements during the last few years, especially because of the introduction of Convolutional Neural Networks. According to some research papers such as "Computer vision and behavioral phenotyping: an autism case study" [34] and "Behavior Imaging: Using Computer Vision to Study Autism" [35], this subject can potentially have a huge impact in ASD detection in children. This can be done by:

1. Making Computer Vision games to test child's ability to recognise and imitate facial expressions or feelings.
2. Recording the child-therapist sessions to then examine the recording to find out if the child has shown: some reaction delay to the applied stimulus, any repetitive behaviour or any abnormal walking/speaking behaviour.

4.2.1 Game Application

As part of this project I decided to design a game to test a child's ability to make a series of facial expressions. In order to realise this game, I partially followed the "Real Time Facial Expression Recognition on Streaming Data" post by Sefik Ilkin Serengil [36] to make use of his pre-trained model for facial expression recognition. I successively coded the game using Python libraries such as: cv2, keras, tkinter and PIL.

The game interface is formed by two windows. In the first one, the user is told if their face is correctly detected and what facial expression they should perform (Figure 4.7 (a)). In the second one, the camera recording is displayed and if a face is detected an empty rectangle will cover the face area displaying on the top left corner the detected expression (Figure 4.7 (b)). Once the user has made the asked expression, a pop-up window will appear to congratulate the user and the first window will update to ask to make a different face expression. The list of available face expressions consist of: angry, fear, happy, sad, surprise and neutral faces.

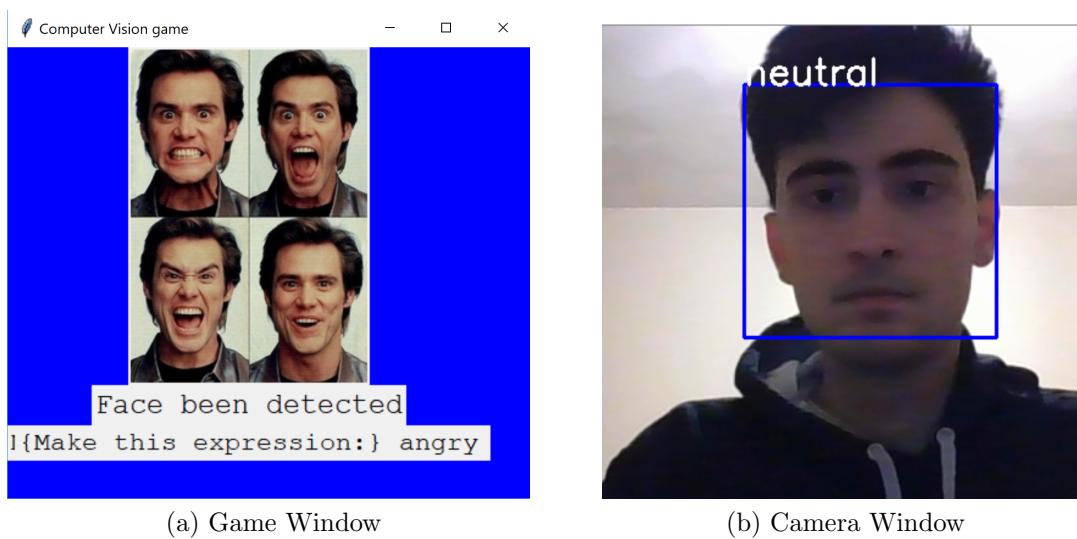


Figure 4.7: Computer Vision Game

Chapter 5

Project Management

5.1 Time Management

Throughout this project I made use of backup repositories such as:

- Dropbox = to store files related to this project, and to be able to access them from any type of station at any moment.
- Google Drive = to store and share (to selected users) project hardware and software advancements (images and videos).
- University of Southampton GitLab = to retain a version-control of all the written code.

Appendix A provides a wealth of project management tools which have been used throughout this project:

1. A table representation of the executed tasks during the project, including the planned start date, months of execution and weeks to complete each task.
2. A Gantt Chart of the planned time-schedule.
3. A Gantt Chart of the actual time-schedule.
4. A Risk Assessment Matrix to display all the possible risks associated with this project.
5. A Work Breakdown Structure to show in a tree-like representation the identified main tasks of this project.

A Hardware Costs report of the project can be found in Appendix D.

Appendix L provides the total report word count, while Appendix K the Design Archive Guide.

5.2 Ethics Application

Appendix F provides the Ethics Application necessary to carry out the data-sets exploration. This approval has been obtained by the University of Southampton Ethics Committee through application using the ERGO II portal.

5.3 Project Challenges

Throughout this project, many challenges have been faced and overcome. Some of these can be noticed from the differences between the Planned and Actual Gantt Charts provided in Appendix A.

One of the main challenges I faced, was the first exploratory data analysis of the used data-set. The original data-set was not in fact structured in a suitable way to be used for Deep Learning applications. After a first unsuccessful attempt to use the MATLAB Deep Learning Toolbox with the original data-set, I therefore decided to restructure the whole data-set and to convert it in a CSV format. This process, although being time consuming, resulted to be a successful approach. Having the inputs features in the right shapes allowed me to make the design of the LSTM and CNN architectures much easier. Additionally, this new restructured data-set can now be used not just in MATLAB but also with other programming languages such as Python and R (making this data-set more accessible to anyone in case of future further developments).

Chapter 6

Conclusion

Overall this project had a positive outcome and all the objectives in the Project Brief have been established.

6.1 Further Advancements

Further developments can be envisaged especially concerning the Artificial Intelligence aspect of this project. The use of Wavelet analysis [37], Fast Fourier Transform [38], Entropy Analysis [39] or of an Auto-Encoder Network [40] during the pre-processing stage could potentially reduce the noise embedded in the registered signals therefore making classification easier. Additionally, it would also be interesting to convert the registered EEG signals from the time domain to the frequency domain to see if that could improve the neural networks performances.

Another issue to be taken into account could be the reliability of the data used (eg. if the data was collected correctly, if all the noise registered during the measurements had been correctly filtered out...) and how reproducible these accuracy results can be using other forms of data which have been collected using a different types of EEG Cap or under different laboratory conditions.

When trying to use Artificial Intelligence in medical applications, high accuracy and reliability must be obtained. Explainable AI (creating models able to explain himself their classification decision making process) could help doctors to understand how/if to use AI when making decisions.

Bibliography

- [1] *Division of Birth Defects, National Center on Birth Defects and Developmental Disabilities, Centers for Disease Control and Prevention.* Accessed: <https://www.cdc.gov/ncbddd/autism/data.html> December 2018.
- [2] Giacomo Vivanti, Cheryl Dissanayake, Cynthia Zierhut, Sally J. Rogers, Victorian ASELCC Team *Brief Report: Predictors of Outcomes in the Early Start Denver Model Delivered in a Group Setting.* Accessed: <https://link.springer.com/article/10.1007/s10803-012-1705-7> December 2018.
- [3] Susan E Bryson, Lonnie Zwaigenbaum, Wendy Roberts *The early detection of autism in clinical practice.* Accessed: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2720498/pdf/pch09219.pdf> March 2019.
- [4] Alabama Phycological Services Center, LLC *AUTISM SPECTRUM DISORDER.* Accessed: <http://alapsych.com/autism-spectrum-disorder/> March 2019.
- [5] Q-CHAT *Quantitative Checklist for Autism in Toddlers.* Accessed: <https://psychology-tools.com/test/qchat-quantitative-checklist-for-autism-in-toddlers> March 2019.
- [6] Tristram Smith, Svein Eikeseth. *O. Ivar Lovaas: Pioneer of Applied Behavior Analysis and Intervention for Children with Autism.* Accessed: <https://link.springer.com/article/10.1007/s10803-010-1162-0> December 2018.
- [7] Corinna F. Grindle, Hanna Kovshoff, Richard P. Hastings, Bob Remington. *Parents Experiences of Home-Based Applied Behavior Analysis Programs for Young Children with Autism.* Accessed: <https://link.springer.com/article/10.1007/s10803-008-0597-z> December 2018.
- [8] Geraldine Dawson, Sally Rogers, Jeffrey Munson, Milani Smith, Jamie Winter, Jessica Greenson, Amy Donaldson, Jennifer Varley. *Randomized, Controlled Trial of an Intervention for Toddlers With Autism: The Early Start Denver Model.* Accessed: <http://pediatrics.aappublications.org/content/125/1/e17.short> December 2018.

- [9] Brainworks, train your mind. *What are brainwaves?*. Accessed: <https://brainworksneurotherapy.com/what-are-brainwaves> December 2018.
- [10] FONSECA, Lineu C. et al. *Quantitative EEG in children with learning disabilities: analysis of band power*. Accessed: <https://www.ncbi.nlm.nih.gov/pubmed/16917604> December 2018.
- [11] What are serious games? - Growth Engineering. Accessed: <https://www.growthengineering.co.uk/what-are-serious-games/> February 2019.
- [12] Gamification Versus Serious Games - Andrew Hughes - Training Industry. Accessed: <https://trainingindustry.com/articles/learning-technologies/gamification-versus-serious-games/> February 2019.
- [13] Fedwa Laamarti, Mohamad Eid, and Abdulmotaleb El Saddik, "An Overview of Serious Games", International Journal of Computer Games Technology, vol. 2014, Article ID 358152, 15 pages, 2014. Accessed: <https://www.hindawi.com/journals/ijcgt/2014/358152/> February 2019.
- [14] History of AI - Shaan Ray - Towards Data Science. Accessed: <https://towardsdatascience.com/history-of-ai-484a86fc16ef> January 2019.
- [15] Artificial Intelligence: Definition, Types, Examples, Technologies - Chethan Kumar GN - Medium. Accessed: <https://medium.com/@chethankumargn/artificial-intelligence-definition-types-examples-technologies-962ea75c7b9b> January 2019.
- [16] Cousins of Artificial Intelligence - Seema Singh - Towards Data Science. Accessed: <https://towardsdatascience.com/cousins-of-artificial-intelligence-dda4edc27b55> January 2019.
- [17] Understanding LSTM Networks, colah's blog Accessed: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/> March 2019
- [18] Understanding LSTM and its diagrams, Shi Yan Accessed: <https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714> March 2019.
- [19] An intuitive guide to Convolutional Neural Networks, Daphne Cornelisse Accessed: <https://medium.freecodecamp.org/>

- an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050* March 2019.
- [20] Max-pooling / Pooling, computersciencewiki.org Accessed: https://computersciencewiki.org/index.php/Max-pooling_-/_Pooling March 2019.
- [21] Understanding AUC - ROC Curve, Sarang Narkhede, Medium Accessed: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> March 2019.
- [22] Choosing the Right Metric for Evaluating Machine Learning ModelsPart 2, Alvira Swalin, University of San Francisco, KDnuggets Accessed: <https://www.kdnuggets.com/2018/06/right-metric-evaluating-machine-learning-models-2.html> March 2019.
- [23] Artificial Intelligence at Google - Our Principles - Google AI. Accessed: <https://ai.google/principles/> January 2019.
- [24] Wikipedia, the free encyclopedia. *Wii Remote*. Accessed: https://en.wikipedia.org/wiki/Wii_Remote November 2018.
- [25] Frederick Vandenbosch *DLP LightCrafter Display 2000 EVM on Raspberry Pi*. Accessed: <http://frederickvandenbosch.be/?p=2948> November 2018.
- [26] Martin Rudel *python-whiteboard*. Accessed: <https://github.com/pnegre/python-whiteboard/wiki> November 2018.
- [27] Deck.Toys *Smoothboard Air with Duo*. Accessed: <https://en.freedomdownloadmanager.org/Windows-PC/SmoothboardAirwithDuo.html> November 2018.
- [28] Pygame Game Development. Accesed: <https://pythonprogramming.net> November 2018.
- [29] McGugan, Will *Beginning Game Development with Python and Pygame: From Novice to Professional (Beginning from Novice to Professional)* 2007, ISBN = 1590598725, Apress.
- [30] Neuroelectronics *Enobio 20*. Accessed: <https://www.neuroelectrics.com/products/enobio/enobio-20> November 2018.

- [31] Neuroelectronics *MatNIC Toolkit*. Accessed: <https://www.neuroelectrics.com/products/software/matnic-remote-stimulation-client> November 2018.
- [32] Fabio Apicella, Federico Sicca et al. *Fusiform Gyrus responses to neutral and emotional faces in children with Autism Spectrum Disorders: a High Density ERP study*. Accessed: <https://www.ncbi.nlm.nih.gov/pubmed/23124137> March 2019.
- [33] Venelin Valkov - Medium *Human Activity Recognition using LSTMs on Android TensorFlow for Hackers (Part VI)*. Accessed: <https://medium.com/@curiously/human-activity-recognition-using-lstms-on-android-tensorflow-for-hackers-part-vi-492da5adef64> March 2019.
- [34] James M. Rehg, Center for Behavior Imaging, Georgia Institute of Technology *Behavior Imaging: Using Computer Vision to Study Autism*. Accessed: <https://pdfs.semanticscholar.org/88e2/be0d767e0a4d077a96a006e2554653d709ed.pdf> March 2019.
- [35] Guillermo Sapiro, Jordan Hashemi and Geraldine Dawson. ScienceDirect *Computer vision and behavioral phenotyping: an autism case study*. Accessed: <https://www.sciencedirect.com/science/article/pii/S246845111830059X> March 2019.
- [36] Sefik Ilkin Serengil, Developer's Log *Real Time Facial Expression Recognition on Streaming Data*. Accessed: <https://sefiks.com/2018/01/10/real-time-facial-expression-recognition-on-streaming-data/> March 2019.
- [37] Neep Hazarika, Jean Zhu Chen, Ah Chung Tsoi, Alex Sergejew - ScienceDirect *Classification of EEG signals using the wavelet transform*. Accessed: <https://www.sciencedirect.com/science/article/pii/S0165168497000388> March 2019.
- [38] Maan M. Shaker - International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering Vol:1, No:3, 2007 *EEG Waves Classifier using Wavelet Transform and Fourier Transform*. Accessed: <https://pdfs.semanticscholar.org/6b88/8f863dca3e43be157189d18a7441cf802b98.pdf> March 2019.

- [39] Peng Li, Chandan Karmakar, et al. - Plos—One *Detection of epileptic seizure based on entropy analysis of short-term EEG*. Accessed: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0193691> March 2019.
- [40] Jeremy Jordan *Introduction to autoencoders*. Accessed: <https://www.jeremyjordan.me/autoencoders/> March 2019.
- [41] Doc Word Counter Accessed: <https://docwordcounter.com/en/> April 2019.

Appendices

A Project Management

Task	Start Date	Months	Weeks to complete
1) BACKGROUND RESEARCHES			
a) Disabilities: causes and detection	0 September		28
b) Intervention: Early Start Denver Model (ESDM)	0 September		3
c) Intervention: Applied Behavior Analysis (ABA)	3 September		2
d) Additional Researches	8 October		3
2) HARDWARE SOLUTION	22 February		6
a) Identify necessary components	2 September		4
b) Ordering and receiving components	2 September		1
c) Testing	3 September		1
3) SOFTWARE SOLUTION	4 October		2
a) Identify types of games to make	3 September		5
b) Programming the first game	3 September		1
c) Programming the second game	4 October		3
4) INTERFACING HARDWARE AND SOFTWARE	6 October		2
a) DLPDLCR2000EVM - Raspberry Py communication	4 October		5
b) Programmed games - Touchscreen projector	4 October		1
c) EEG Cap - Games synchronization	5 October		2
5) EEG CAP - ENOBIO 20	7 October		2
a) Application Rasearches	6 October		3
b) MATLAB configuration	6 October		1
c) Python integration	7 October		2
6) ADDITIONAL SOFTWARE GAMES	7 October		2
a) Identify types of games to make	9 November		8
b) Programming	9 November		2
7) PROGRESS REPORT	11 November		6
a) Organising/Structuring report	9 November		2
b) Report writing	9 November		1
8) MACHINE/DEEP LEARNING CLASSIFICATION	10 November		1
a) Identify data problems	17 January		11
b) Reconstruct Dataset	17 January		3
c) Training/Testing Designed Models	20 February		2
9) FINAL REPORT	22 February		6
a) Organising/Structuring report	27 March		4
b) Report writing	27 March		1
10) Project Viva	28 April		3
	31 April		2

Figure A.1: First and Second Semester Gantt Chart Data

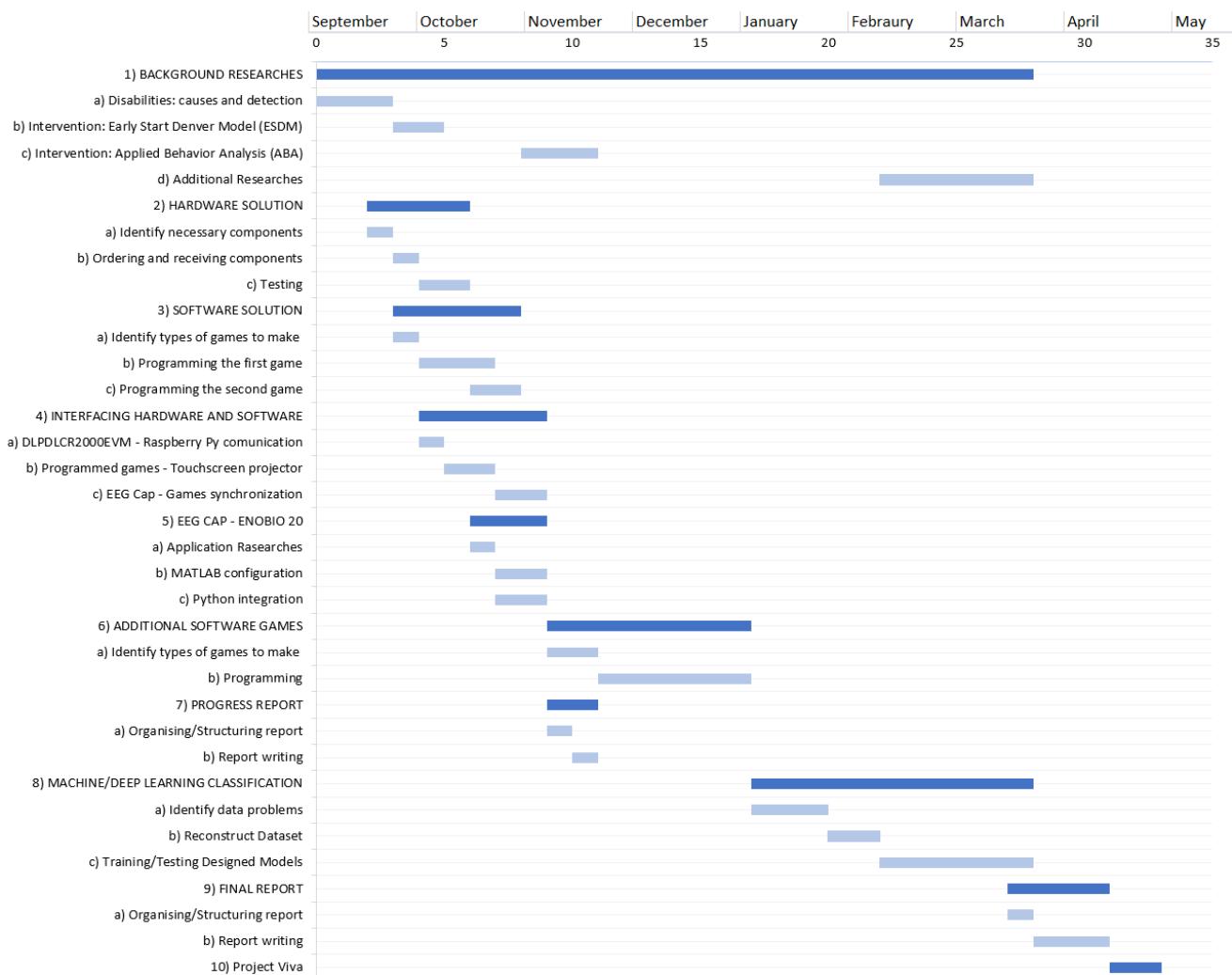


Figure A.2: Planned Gantt Chart

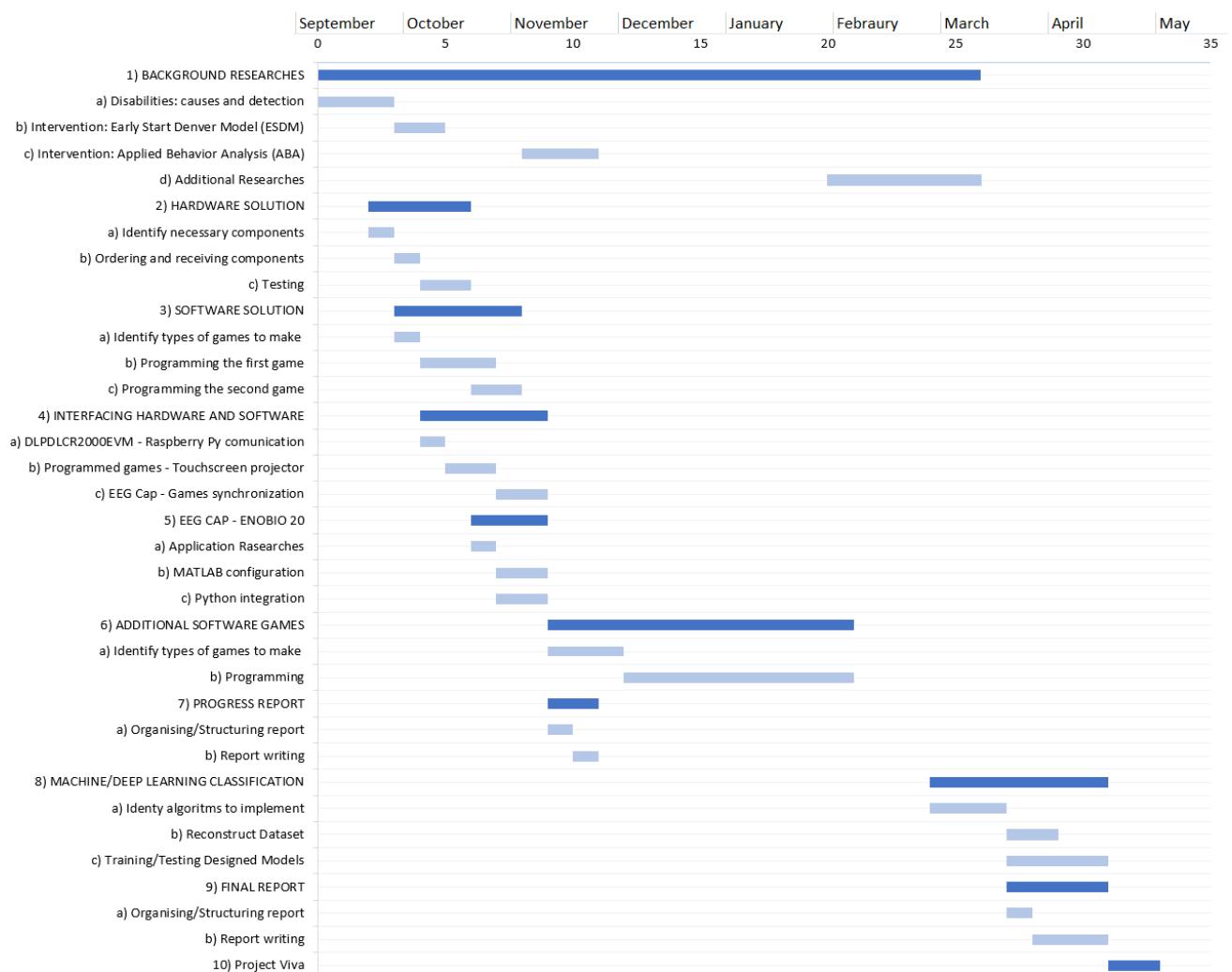


Figure A.3: Actual Gantt Chart

Risk Assessment Matrix

		Severity			
		NEGLIGIBLE small/unimportant; not likely to have a major effect on the operation of the project	MARGINAL minimal importance; has an effect on the operation of project but will not affect the event outcome	CRITICAL serious/important; will affect the operation of the project in a negative way	CATASTROPHIC maximum importance; WILL affect the operation of the project in a negative way
Probability	LOW <small>This risk has rarely been a problem and never occurred</small>	Background researches delays	Use of outdated documentation	Final Report writing delays	Machine Learning Misclassification
	MEDIUM <small>This risk will MOST LIKELY occur in this project</small>	Game programming delays	Non-exhaustive Machine Learning Computational Power	Machine Learning delays	Hardware implementation failure
	HIGH <small>This risk WILL occur at this project, possibly multiple times, and has occurred in the past</small>	Non-exhaustive game design	EEG Hardware Failure	Unresolvable software bugs	Hidden software bugs

Explanation of Risk Ranking	
LOW	MEDIUM
HIGH	
EXTREME	

Figure A.4: Risk Assesment Matrix

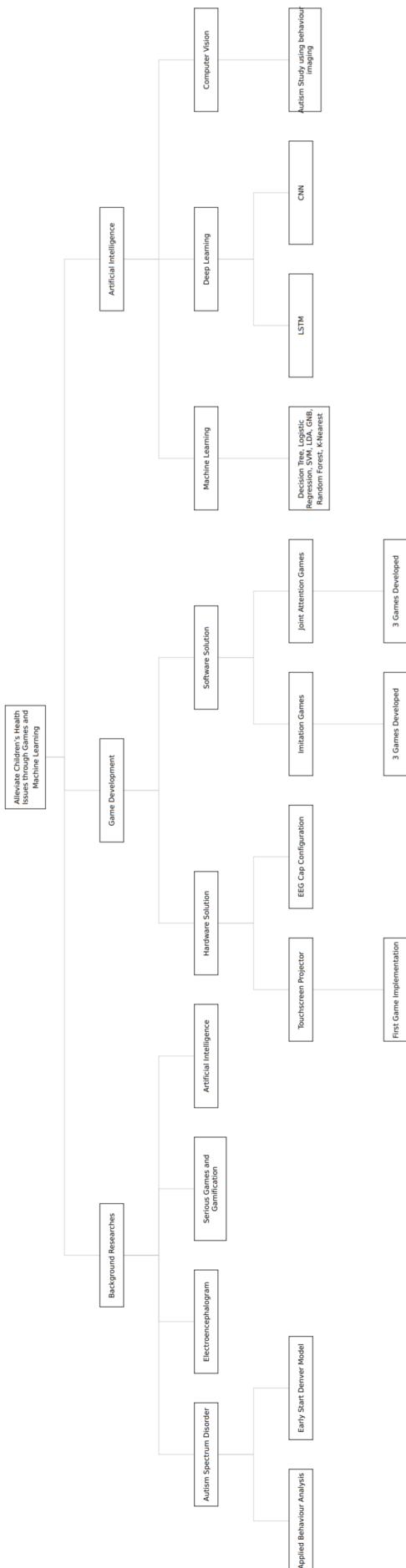


Figure A.5: Work Breakdown Structure

B Evaluation Board wiring and stand

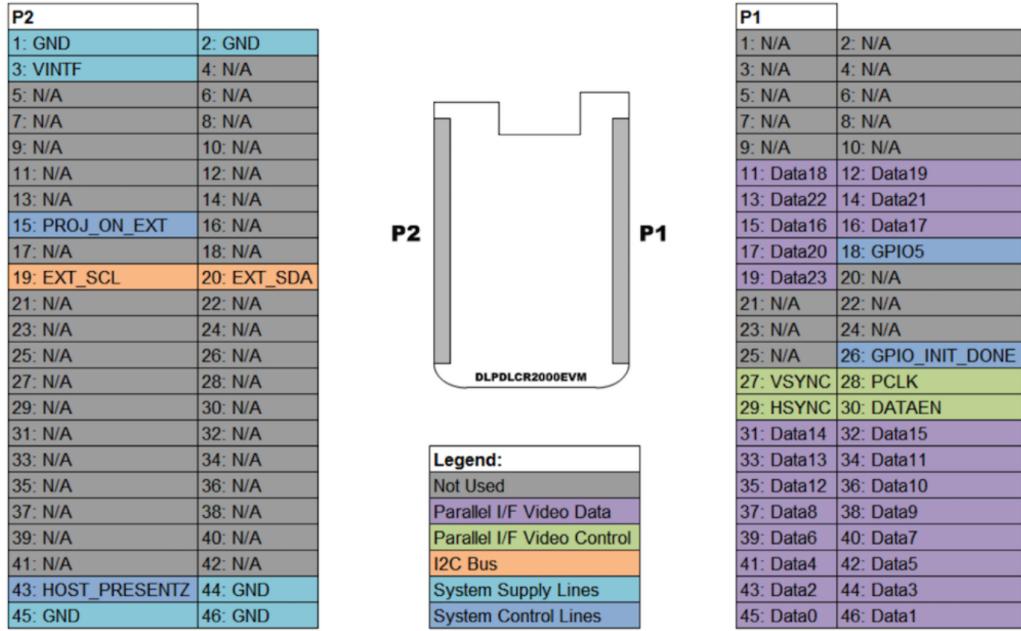


Figure B.1: Evaluation Board

1	P2_6: 5V
P1_27: VSYNC	
P1_29: HSYNC	
P1_16: Data17	P1_38: Data9
	P1_36: Data10
P1_45: Data0	P1_46: Data1
	P2_20: EXT_SDA
	P2_19: EXT_SCL
P1_42: Data5	
P1_32: Data15	
P1_39: Data6	P1_31: Data14
	P1_33: Data13
P1_28: PCLK	P1_30: DATAEN
P1_34: Data11	
P1_35: Data12	P1_40: Data7
P1_37: Data8	
P1_43: Data2	P1_15: Data16
	P1_44: Data3
P2_46: GND	P1_41: Data4

Figure B.2: Raspberry Pi's GPIO

Images reproduced from Frederick Vandenbosch DLP, LightCrafter Display 2000 EVM on Raspberry Pi [25].

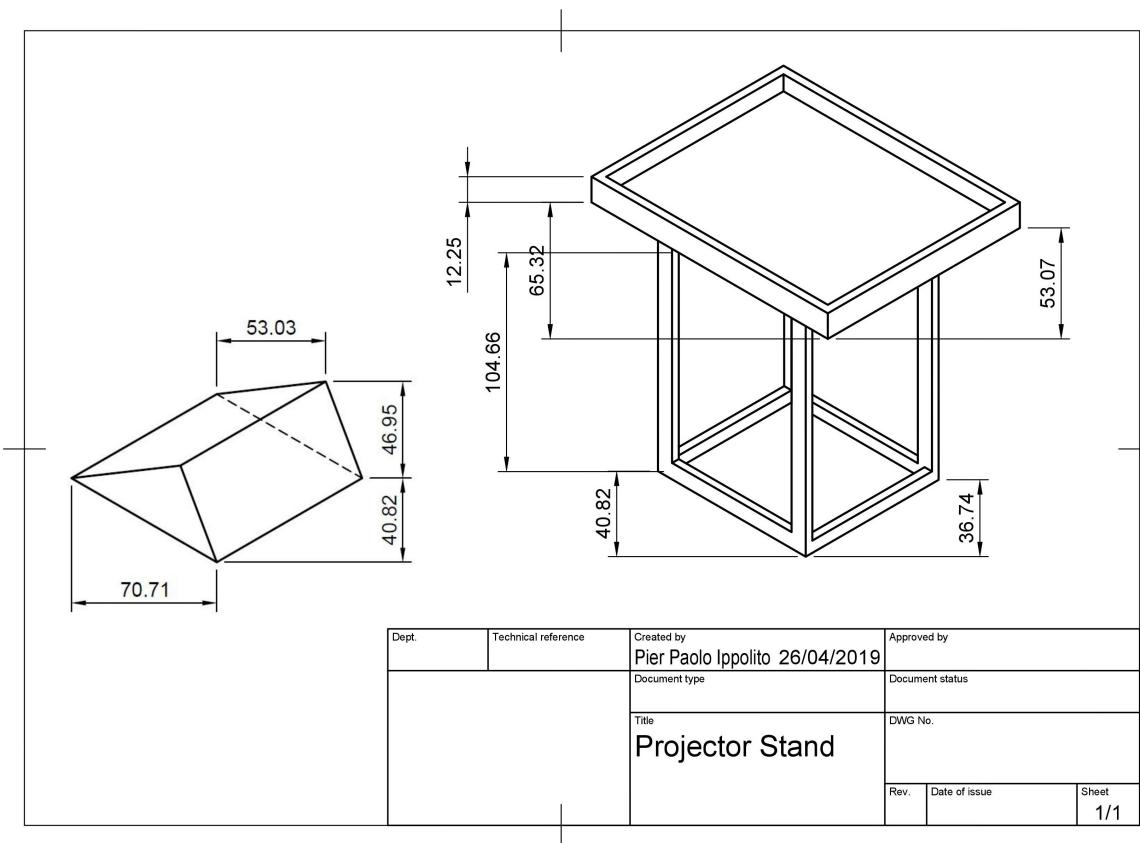


Figure B.3: Projector stand design made using Fusion 360

C Project Brief

Part III Individual Project Brief Alleviate children's health issues through games and Machine Learning

Student: Ippolito, Pier Paolo
ppi1u16@soton.ac.uk

Supervisor: Maharatna, Koushik
km3@ecs.soton.ac.uk

• Problem

Children can be subject to illness since their birth, identifying their problems at a early stage can lead to more consistent improvements for them. This is mainly dependent on neuroplasticity (brain's ability to reorganize itself throughout life), which is way higher during the first few years of our life. Children may be affected by different types/forms of disorders like disability and autism. These can be caused by factors such as: genetics, enviroment, diet and lifestyle. Therefore, prevention can play a crucial role in decreasing the number of children's effected.

• Goals

This project aims to:

1. Outline and categorize the main types of illness children's can suffer
2. Explain why these problems appear and how to prevent them
3. Research different ways to identify and mitigate their problems using games (to increase thier level of engagement) and Machine Learning (to identify patterns and suggest alternative approches).
4. Produce both a hardware and software solution to aid in the mitigation of these issues

• Scopes

The first part of the project will focus on doing background resarches and creating a hardware implementation. Different intervention approaches such as The Early Start Denver Model (ESDM) and Applied Behavior Analysis (ABA) and their outcomes will be considered. In addition to this, will be studied technologies used in hospitals to detect Children's isssues. During the second part of the project software games for the created hardware will be developed and/or existing ones will be used . Finally, using datasets from medical records it could be possible to use machine learnig to find patterns and identify models to improve the results obtained originally.

D Hardware Costs

An additional risk that had to be considered at the beginning of the project, was to make best use out of the given project budget (£150 per person). In order to make the best choice, I carried out an exhaustive research before deciding what was necessary to buy and what was not. I decided therefore, in order to limit the amount of money to spend, to make use of a Raspberry Pi 3B given to me by the University for a first year project and of a Wiimote I was already in possession before the start of the project. I also made use of other accessories necessary such as batteries, button switches and wires from the Building 16 Level 1 laboratory facilities.

Table 1 shows the order I decided to place from the approved university supplier OneCall (Non-Marketplace). The subtotal prices column (expressed in £) takes already account of applied VAT on individual items.

Quantity	Description	Subtotal
1	DLPDLCR2000EVM - Evaluation Board	106.0920£
5	LIR053 - Infrared Emitter - 16 , T-1 3/4 (5mm), 20 mA	0.9360£
1	Battery Holder, AA x 1, Wire Leads	0.8520£
1	2460 - Battery Holder, AA x 1, Through Hole	0.8520£

Table 1: Order Placed from Onecall (Non-Marketplace)

At the end of the project £41.268 out of the given budget have been saved. I planned in fact to make use of this part of the budget just in case of failure of any hardware component already acquired.

E Listings

In Listing 1 has been chosen a period of 30 seconds and 20 channels.

```

1 [ ret , intletEEG ] = MatNICEEGConnectLSL( 'NIC' )
2 [ ret , eeg_set , timestamp_set ] = MatNICEEGRecordLSL( 30 , 20 , intletEEG )
3 csvwrite( 'output-gameplay.csv' , eeg_set )

```

Listing 1: MATLAB code setupscript3.m

```

1 import matlab.engine
2 from threading import Thread
3 # EEG readings pre-game and set-up
4 eng = matlab.engine.start_matlab()
5 eng.addpath(r'C:\Users\hp\Dropbox\Individual Project\Researches\EEG\
    MatNIC_v03.08', nargout=0)
6 tf = eng.setupscript(nargout=0)
7 print(tf)
8 # Defining threading function to get and store eeg data while playing
9 def eeg_func():
10     # EEG readings during-game
11     eng = matlab.engine.start_matlab()
12     eng.addpath(r'C:\Users\hp\Dropbox\Individual Project\Researches\EEG\
        \MatNIC_v03.08', nargout=0)
13     tf3 = eng.setupscript3(nargout=0)
14     print(tf3)
15 Thread(target = eeg_func).start()

```

Listing 2: Python code added to enable EEG readings synchronisation

```

1 trainedtree = tree.DecisionTreeClassifier().fit(X_Train, Y_Train)
2 predictionstree = trainedtree.predict(X_Test)
3 print(confusion_matrix(Y_Test, predictionstree))
4 print(classification_report(Y_Test, predictionstree))

```

Listing 3: Decision Tree Code

```

1 segments = []
2 for i in range(0, len(df) - N_TIME_STEPS, step):
3     ch = []
4     for j in range(0, NFEATURES):
5         ch.append(df.iloc[:, j].values[i: i + N_TIME_STEPS])
6     segments.append(ch)
7 labels = []
8 for i in range(0, len(df) - N_TIME_STEPS, step):
9     label = stats.mode(df['Label'][i: i + N_TIME_STEPS])[0][0]
10    labels.append(label)
11 labels1 = np.asarray(pd.get_dummies(labels), dtype = np.float32)
12 reshaped_segments = np.asarray(segments, dtype= np.float32).reshape(-1,
N_TIME_STEPS, NFEATURES)
13 X_train, X_test, y_train, y_test = train_test_split(
14     reshaped_segments, labels1, test_size=0.2, random_state=
RANDOMSEED)

```

Listing 4: LSTM Preprocessing

```

1 model_json = model.to_json()
2 with open("model.json", "w") as json_file:
3     json_file.write(model_json)
4 model.save_weights("model.h5")

```

Listing 5: Storing CNN Model

F Ethics Application



Pilot Version 18th December 2015

Ethics Application Form for SECONDARY DATA ANALYSIS

Please consult the guidance at the end of this form before completing and submitting your application.

1. Name(s): Pier Paolo Ippolito
2. Current Position: 3rd Year Undergraduate Student.
3. Contact Details:

Division: Electronics and Computer Science
 Email: ppilu16@soton.ac.uk
 Phone: 07904579210
4. Is your research being conducted as part of an education qualification?
 Yes No
5. If Yes, please give the name of your supervisor:
 Koushik Maharatna
6. Title of your research project / study:
 Alleviate Children's Health Issues through Games and Machine Learning
7. Briefly describe the rationale, aims, design and research questions of your research
Please indicate clearly whether you are applying for ethics approval for a specific piece of research, or for overarching ethics approval to use certain datasets for a range of research activities. Approval for the latter will only cover the datasets specified here, for a maximum of 3 years and then subject to renewal.

This research aims to make use of the data carried out during the "Fusiform Gyrus responses to neutral and emotional faces in children with Autism Spectrum Disorders: a High Density ERP study" to use Machine Learning for classification purposes such as identifying if a child is affected or not by disabilities analysing the data-set input features (eg. EEG signals, no personal data). Fully anonymised data.
8. Describe the data you wish to analyse
Please give details of the title of the dataset, nature of data subjects (e.g. individuals or organisations), thematic focus and country/countries covered. Indicate whether the data are qualitative or quantitative, survey data, administrative data or other types of data. Identify the source from where you will be obtaining the data (including a web address where appropriate).

The dataset used will make use of the data carried out during the: "Fusiform Gyrus responses to neutral and emotional faces in children with Autism Spectrum Disorders: a High Density ERP study" (<https://www.ncbi.nlm.nih.gov/pubmed/23124137>). Children affected by Autism Spectrum Disorders (ASD) typically show early and severe impairments in understanding social and emotional information conveyed by faces. This study aimed to show to a group of children (some of which were affected by ASD and some who didn't) a



sequence of stimulus (eg. images of happy faces, sad faces, etc..) and to record their EEG response for later analysis.

- 9. What are the terms and conditions around the use of the data? Did data subjects give consent for their data to be re-used? If not, on what basis is re-use of the data justified?**

Please state what (if any) conditions the data archive imposes (e.g. registration, signing of confidentiality agreement, specific training etc.). In many cases the data controller will have given explicit permission for data re-use. Please explain how you justify the use of data if approval and consents for the original data collection and re-use are not in place. This may be the case where, for example, the original data collection predated requirements for ethics review or occurred in a jurisdiction where explicit consent and approval are not required.

Concerning the dataset, written informed consent was obtained in all cases by parents of all the children who took part in this study. The only requirement is to acknowledge and cite the original data. There are no other restrictions. As the data are fully anonymised, there is no GDPR compliance issue. The dataset will be provided by the project supervisor (Koushik Maharatna).

- 10. Do you intend to use personal data**

(https://ico.org.uk/media/1549/determining_what_is_personal_data_quick_reference_guide.pdf) or sensitive personal data (<http://www.legislation.gov.uk/ukpga/1998/29/section/2>) as defined by the Data Protection Act (even if the data are publicly available)?

Yes No

If YES, please specify what personal data will be included and why.

- 11. Do you intend to link two or more datasets?**

Data linkage refers to merging of information from two or more sources of data to consolidate facts concerning an individual or an event that are not available in any separate record. Please note that for the purposes of research ethics we are not interested in the merging of different waves of a particular survey, or the merging of data from different countries for the same survey.

Yes No

If YES, please give details of which datasets will be linked and for what purposes.

- 12. How will you store and manage the data before and during the analysis? What will happen with the data at the end of the project?**

Please consult the University of Southampton's Research Data Management Policy (<http://library.soton.ac.uk/researchdata/storage> and <http://www.calendar.soton.ac.uk/sectionIV/research-data-management.html>), and indicate how you will abide by it.

Data will be downloaded to a virtual machine provided by the university and accessed via my laptop. At the end the project the data will be given back to the project supervisor (Koushik Maharatna).

- 13. How will you minimise the risk that data subjects (individuals or organisations) could be identified in your presentation of results?**



Please consider whether disclosive ID codes have been used (e.g. date of birth) and whether it is theoretically possible to identify individuals by combining characteristics (e.g. widow in Hampshire with 14 children) or by combining datasets. How will you protect individuals' anonymity in your analysis and dissemination?

The data I will be given will be fully anonymised. There is no possibility to identify data subjects (No way to relate data to patients). I have never been in direct contact with the patients, and I haven't been involved in carrying out the measurement.

14. What other ethical risks are raised by your research, and how do you intend to manage these?

Issues may arise due to the nature of the research you intend to undertake and/or the subject matter of the data. Examples include: data or analysis that are culturally or socially sensitive; data relating to criminal activity, including terrorism, and security sensitive issues.

None.

15. Please outline any other information that you feel may be relevant to this submission.

For example, will you be using the services or facilities of ONS, ADRN, or HSCIC and/or are you obtaining ethical review from NRES (through IRAS) or other? Please confirm whether the data being used are already in the public domain.

None.

16. Please indicate if you, your supervisor or a member of the study team/research group are a data controller and/or data processor in relation to the data you intend to use as defined by the Data Protection Act, and confirm that you/they understand your/their respective responsibilities <https://ico.org.uk/for-organisations/guide-to-data-protection/key-definitions/>.

The controller and processor are defined by the information commissioner's office as follows:

- A controller determines the purposes and means of processing personal data.
- A processor is responsible for processing personal data on behalf of a controller.

Since no personal data is being used in this study, we are neither the controller or the processor.

Note: This Ethics Application Form is currently being piloted. If you have comments on any of the questions, it would be helpful if you could email them to rgoinfo@soton.ac.uk with "Secondary Data Analysis Form" in the subject line.



Guidance on applying for ethics approval for secondary data analysis

If your research PURELY involves the following, you do not need to apply for ethics approval:

- analysis of aggregated data on individuals or organisations (e.g. GDP, labour force participation rates, fertility rates);
- meta-analyses (i.e. the analysis of studies);
- literature reviews or reviews/analyses of reports, policies, documents, meeting minutes, newspaper articles, films.

Filling in the online submission form on ERGO II:

- Please give your application a title that includes 'SDA' (Secondary Data Analysis).
- Please refer to the "**ERGO II Guidance for Applicants**" document (downloadable from the ERGO II site) on how to answer the submission questionnaire correctly..

Additional Forms:

If your study PURELY involves secondary analysis of data, you only need to fill in the 'Ethics Application Form for Secondary Data Analysis'. You do not need a Risk Assessment Form.

If your study is a mixed-method study involving secondary data analysis AND some component of data collection (e.g. interviews, online survey), then you need to fill in additional forms:

- Ethics Application Form (for studies other than secondary data analysis)
- Risk Assessment Form
- Participant Information Sheet
- Consent Form
- Draft research instrument

Please note:

- You must not begin data analysis until ethical approval has been obtained.
- It is your responsibility to follow the University of Southampton's Ethics Policy and any relevant academic or professional guidelines in the conduct of your research. This includes ensuring confidentiality in the storage and use of data.
- It is your responsibility to provide full and accurate information in completing this form.

G ML MATLAB Analysis

Plot of all the channels voltages registered for all the time-steps (repetition number one) in Happy Data, Typical Child number 1.

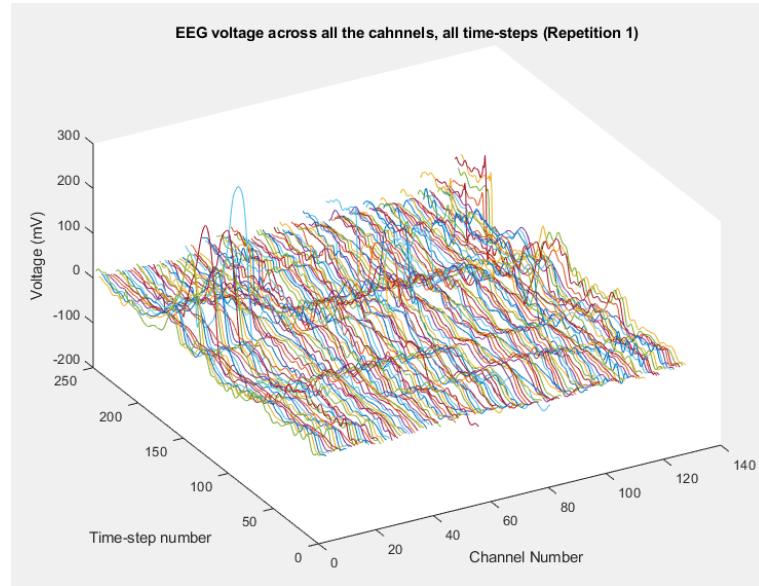


Figure G.1: EEG voltage across all the cahnnels, all time-steps (Repetition 1), Happy Data, TYP1

Plot of all the channels voltages registered for all the time-steps (repetition number one) in Happy Data, ASD Child number 1.

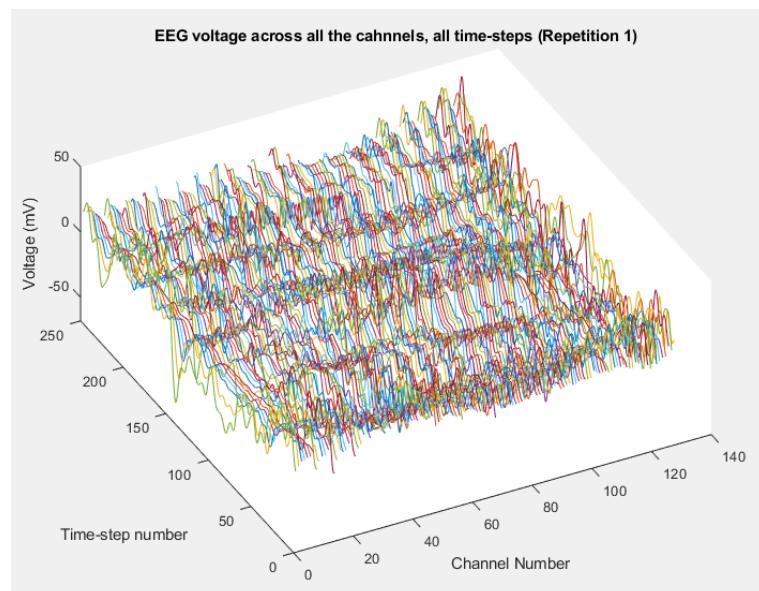


Figure G.2: EEG voltage across all the cahnnels, all time-steps (Repetition 1), Happy Data, ASD1

Plots of the channel number against the voltage amplitude of EEG brainwaves for the first typical child (on the left side) and the first ASD child (on the right side), this is done for all the 5 stimuli and for their corresponded experiment repetitions number 1 (left side of each plot) and 35 (right side of each plot). All the channels, all the time-steps and just one repetition is considered for each graph.

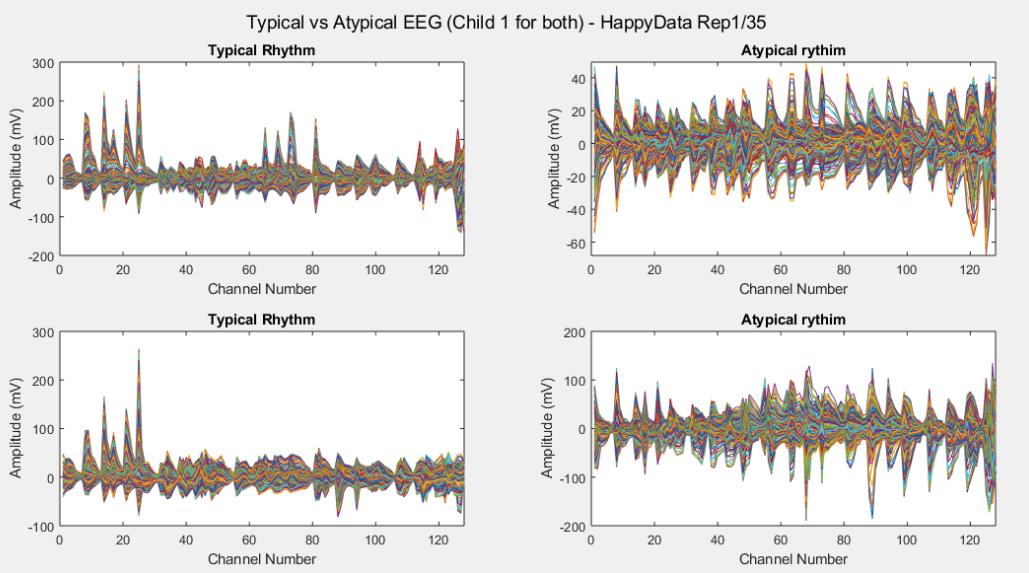


Figure G.3: Happy Data Stimulus

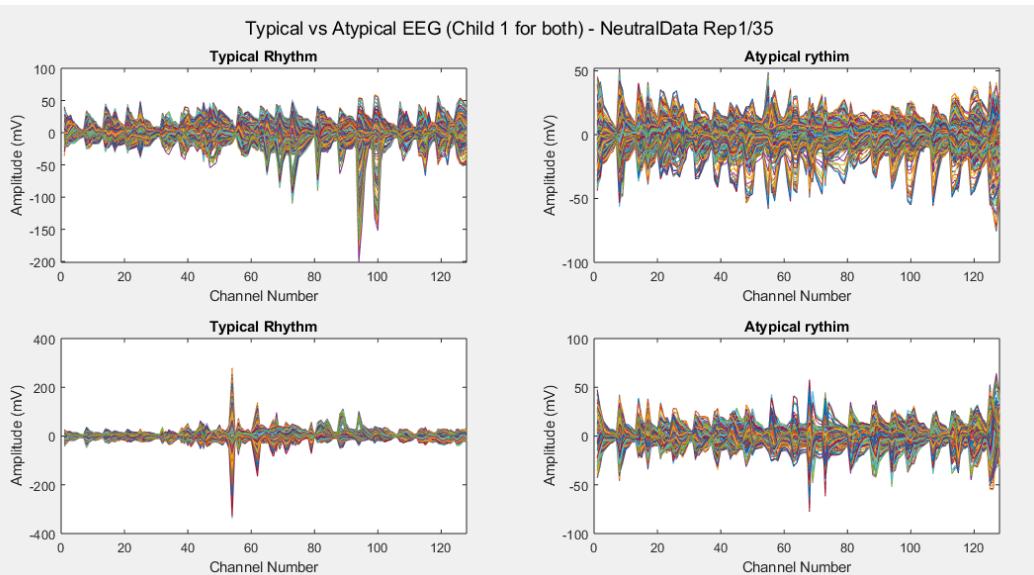


Figure G.4: Neutral Data Stimulus

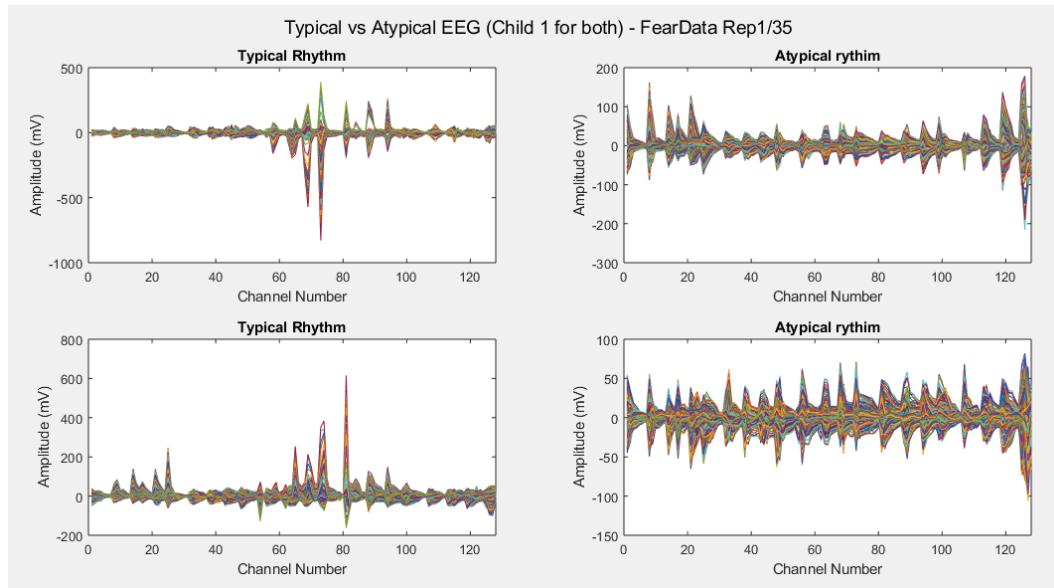


Figure G.5: Fear Data Stimulus

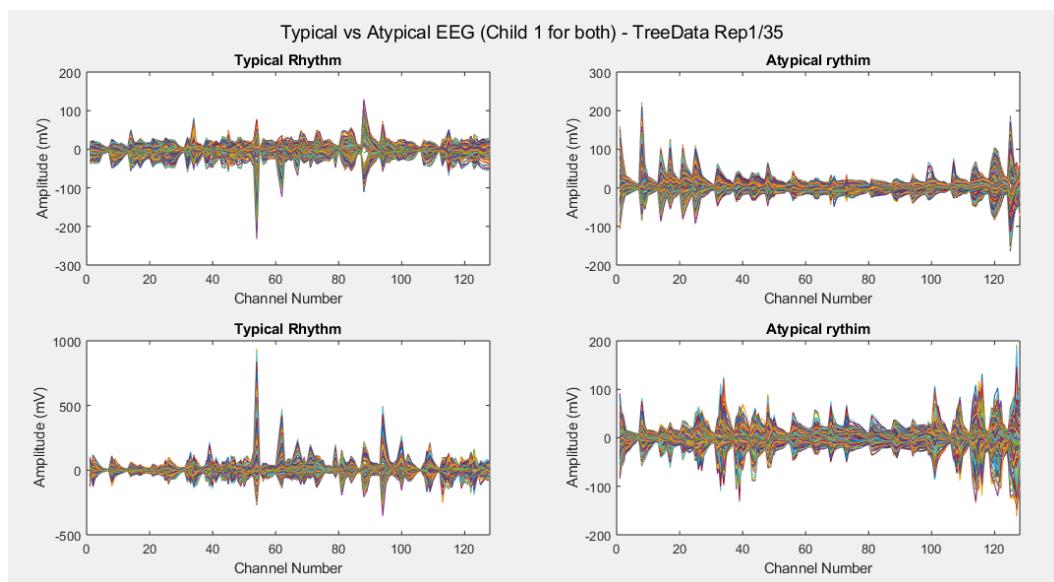


Figure G.6: Tree Data Stimulus

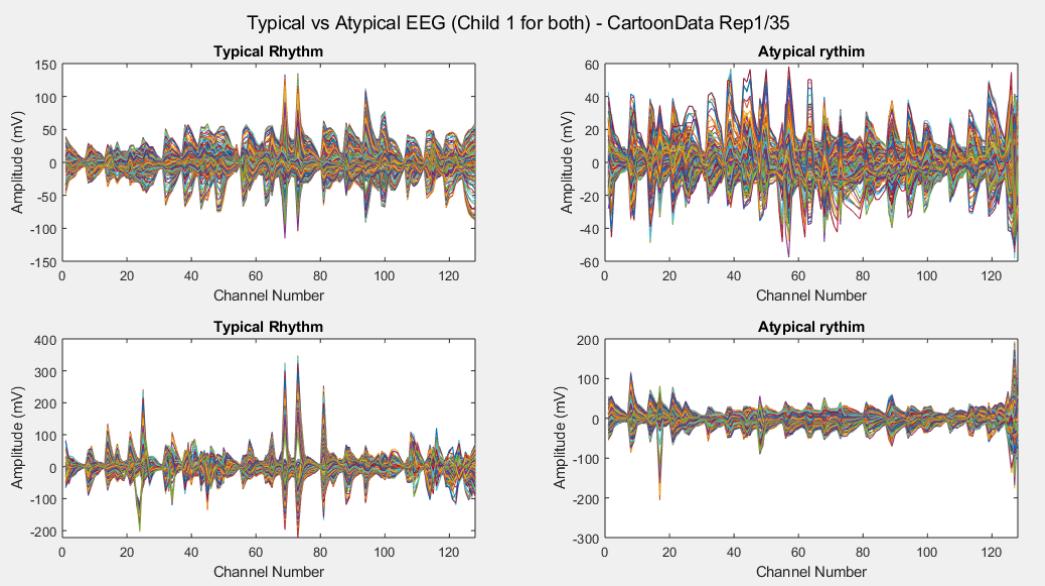


Figure G.7: Cartoon Data Stimulus

Considering just channel number 1, all the time-steps and just one repetition (Repetition number 1 on the left and repetition number 35 on the right of each plot), the following results can be obtained:

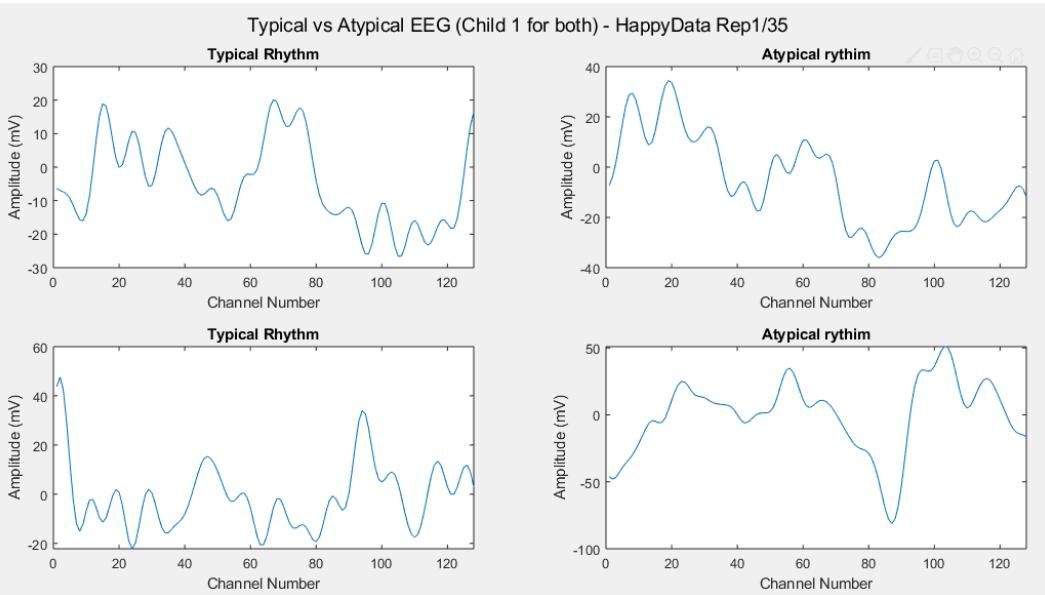


Figure G.8: Happy Data Stimulus

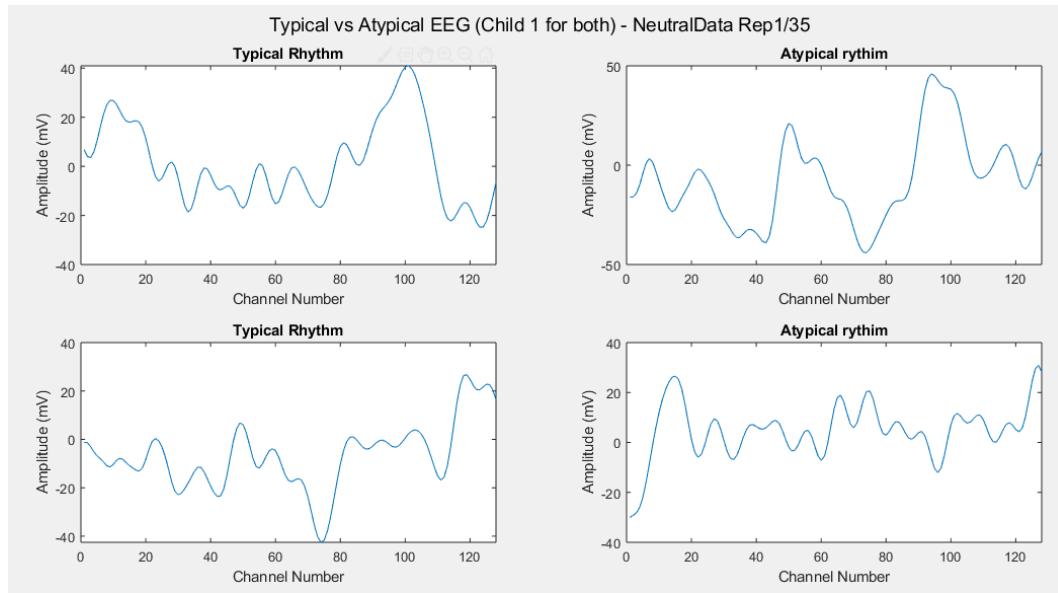


Figure G.9: Neutral Data Stimulus

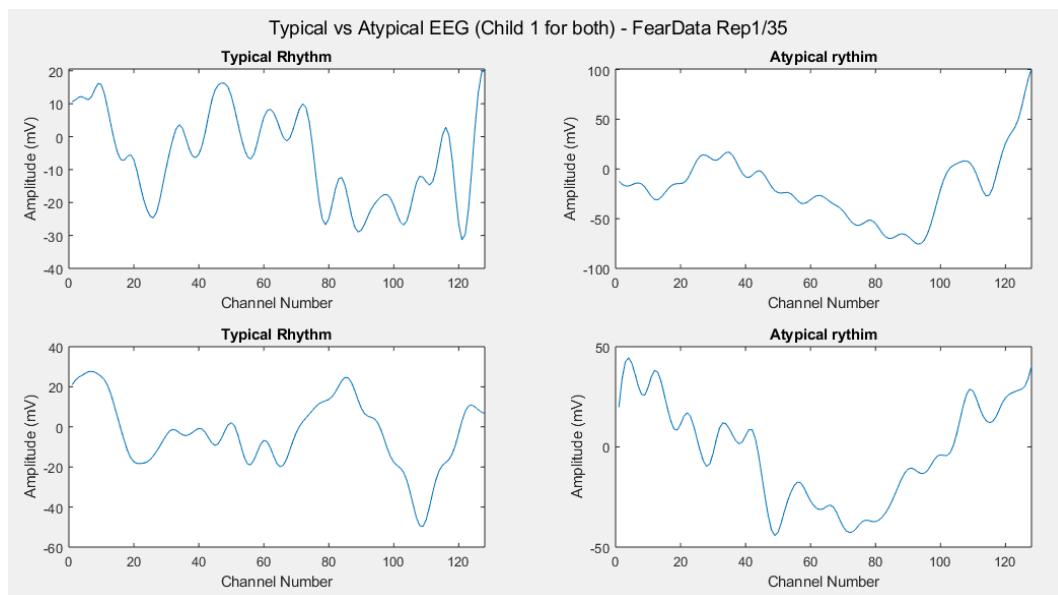


Figure G.10: Fear Data Stimulus

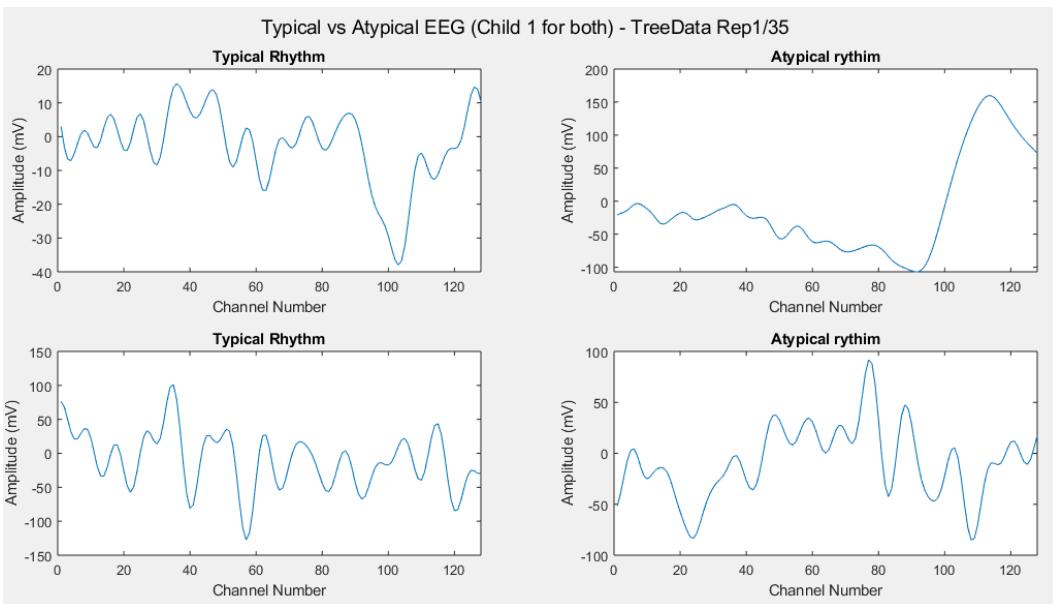


Figure G.11: Tree Data Stimulus

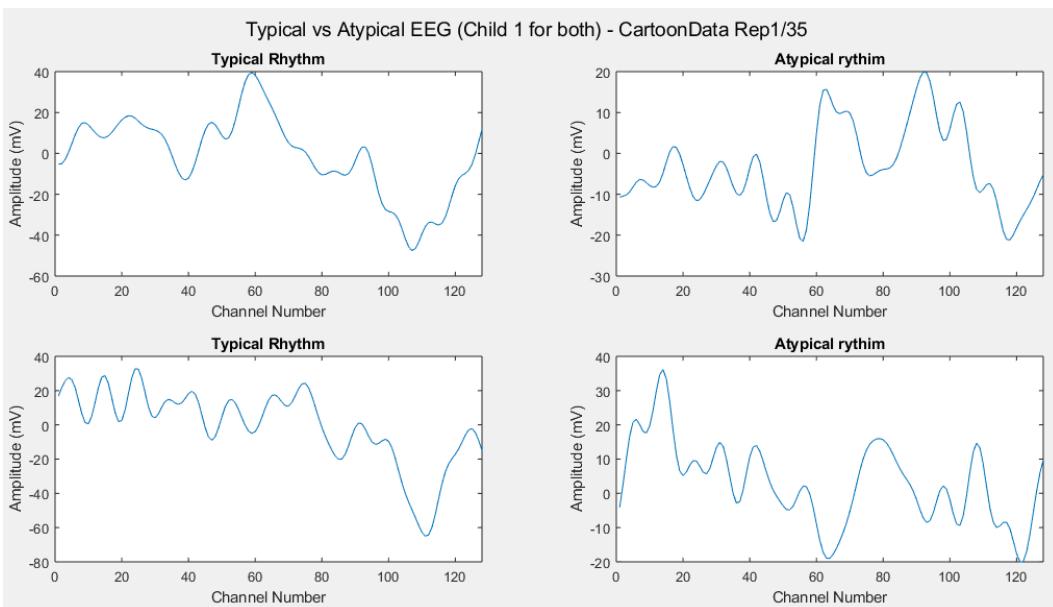


Figure G.12: Cartoon Data Stimulus

H Individual Stimulus Data-sets Results

Data-set	Decision Tree Accuracy (%)
Happy Stimulus	79
Neutral Stimulus	80
Fear Stimulus	79
Tree Stimulus	80
Cartoon Stimulus	80

Table 1: Decision Tree accuracy for Individual Stimulus Data-sets

For the LSTM because of the reduced amount of data when working with the individual stimulus, has been chosen a Train/Test split ratio of 80% against 20%.

Parameter	Number
Time Steps	250
Features	128
Steps	10
Classes	2
Hidden Units	64
L2 Regularisation	0.0015
Learning Rate	0.0025
Epochs	50
Batch Size	1024

Table 2: LSTM Parameters for Individual Stimulus Data-sets

Data-set	LSTM Accuracy (%)
Happy Stimulus	84.4
Neutral Stimulus	84.3
Fear Stimulus	79.5
Tree Stimulus	78.4
Cartoon Stimulus	83.1

Table 3: LSTM accuracy for Individual Stimulus Data-sets

For the CNN has instead been kept a Train/Test split ratio of 70% against 30% and the model architecture has been slightly modified to best fit the reduced amount of data:

1. Two 2D Convolutional Layers having 32 filters, a kernel size of 5×5 , a ReLU (rectified linear unit) function and the same padding.
2. A 2D MaxPooling layer of 2×2 size.

3. A Dropout layer of 0.2 intensity (in order to avoid over-fitting the data)
4. A layer to first flatten the data from three dimensions to just one, and then another one to condense the input to give to the classifier 128 features (always using the ReLU function).
5. A second Dropout layer of 0.5 intensity.
6. Finally, a Dense layer (of two neurons) to produce the classification result, using a Softmax activation function.

Data-set	CNN Accuracy (%)
Happy Stimulus	86
Neutral Stimulus	83.2
Fear Stimulus	84.9
Tree Stimulus	78.3
Cartoon Stimulus	86.4

Table 4: CNN accuracy for Individual Stimulus Data-sets

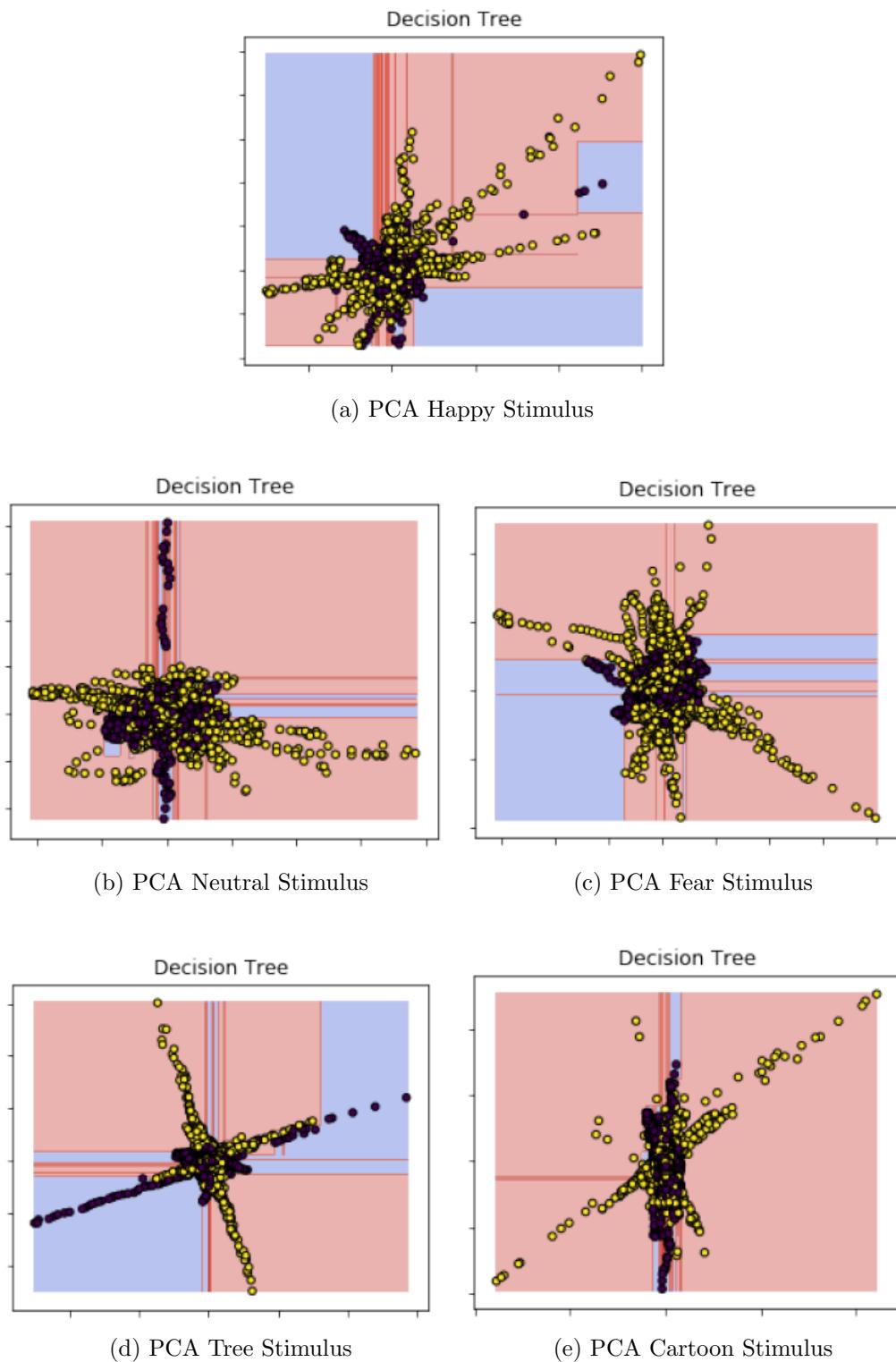


Figure H.1: PCA classification using Decision Tree

I Decision Tree Classification

The following graph was realised storing the tree as a .dot file and then using the Graphviz library. For the purposes of space, just the first 48 branches are displayed because the full decision tree is composed by more than 9000 branches.

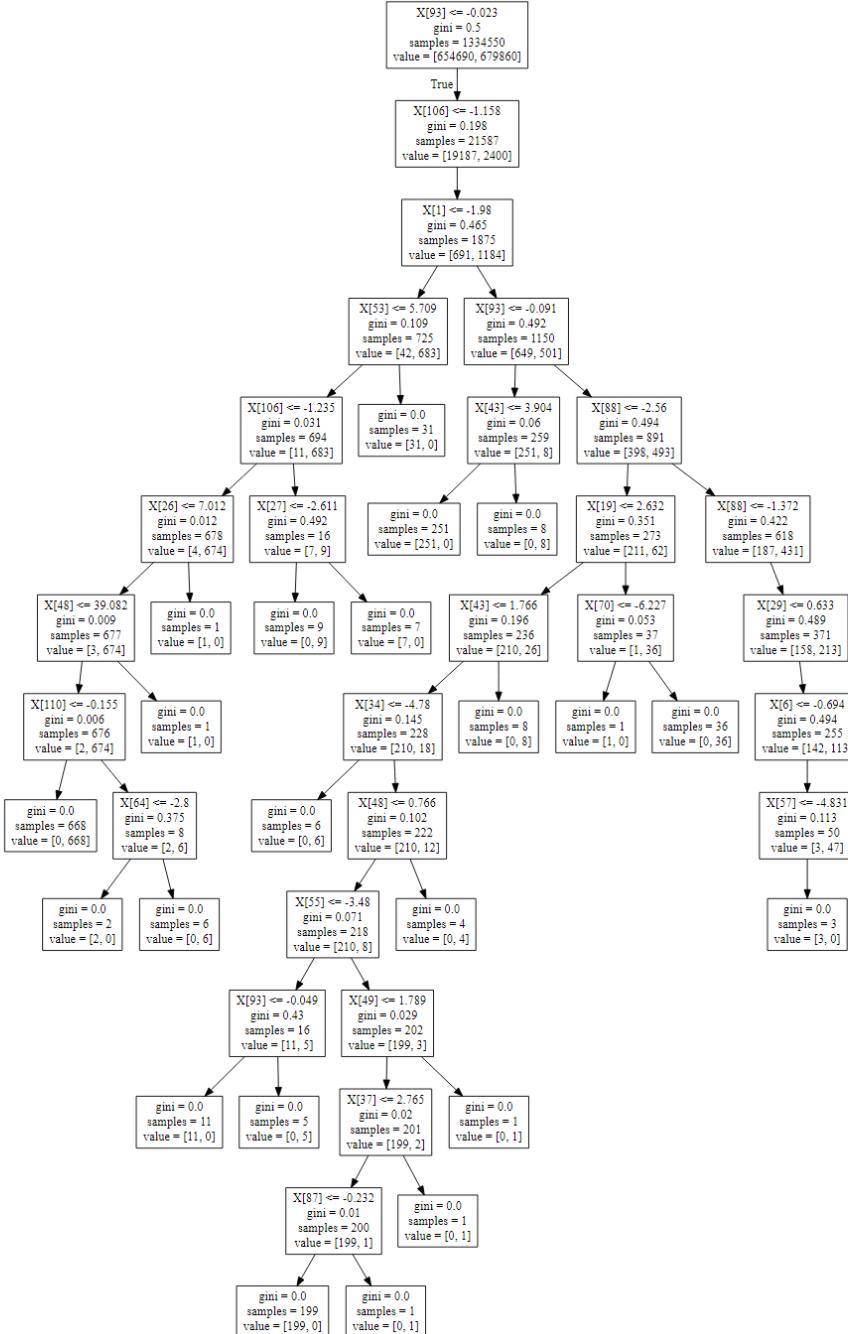


Figure I.1: Decision Tree Classification Plot

J LSTM Training/Test Loss and Accuracy

```

epoch: 1 test accuracy: 0.6118032932281494 loss: 2.5684688091278076
epoch: 2 test accuracy: 0.6243060231208801 loss: 1.7545182704925537
epoch: 3 test accuracy: 0.6711257100105286 loss: 1.2619996070861816
epoch: 4 test accuracy: 0.6800000071525574 loss: 1.011061429977417
epoch: 5 test accuracy: 0.702382504940033 loss: 0.8562345504760742
epoch: 6 test accuracy: 0.7702295184135437 loss: 0.6937601566314697
epoch: 7 test accuracy: 0.7656830549240112 loss: 0.6572901010513306
epoch: 8 test accuracy: 0.8242623209953308 loss: 0.543678879737854
epoch: 9 test accuracy: 0.8229070901870728 loss: 0.5198442339897156
epoch: 10 test accuracy: 0.8916721343994141 loss: 0.39188259840011597
epoch: 11 test accuracy: 0.8714753985404968 loss: 0.4164002537727356
epoch: 12 test accuracy: 0.9058797955513 loss: 0.34885913133621216
epoch: 13 test accuracy: 0.8671038150787354 loss: 0.42039692401885986
epoch: 14 test accuracy: 0.9237158298492432 loss: 0.301808625459671
epoch: 15 test accuracy: 0.9280437231063843 loss: 0.293853759765625
epoch: 16 test accuracy: 0.9337267875671387 loss: 0.27200374007225037
epoch: 17 test accuracy: 0.9365683197975159 loss: 0.26243898272514343
epoch: 18 test accuracy: 0.9393661022186279 loss: 0.2660514712333679
epoch: 19 test accuracy: 0.92402184009552 loss: 0.29361653327941895
epoch: 20 test accuracy: 0.9248087406158447 loss: 0.280854731798172
epoch: 21 test accuracy: 0.9429945349693298 loss: 0.2487581968307495
epoch: 22 test accuracy: 0.952262282371521 loss: 0.22244399785995483
epoch: 23 test accuracy: 0.9550601243972778 loss: 0.21271300315856934
epoch: 24 test accuracy: 0.9490272998809814 loss: 0.2249094396829605
epoch: 25 test accuracy: 0.9499016404151917 loss: 0.22356955707073212
epoch: 26 test accuracy: 0.9537048935890198 loss: 0.21349379420280457
epoch: 27 test accuracy: 0.9586448073387146 loss: 0.20318514108657837
epoch: 28 test accuracy: 0.9568524360656738 loss: 0.20390968024730682
epoch: 29 test accuracy: 0.9505136609077454 loss: 0.22578518092632294
epoch: 30 test accuracy: 0.9589071273803711 loss: 0.19915363192558289
epoch: 31 test accuracy: 0.9449617266654968 loss: 0.24285006523132324
epoch: 32 test accuracy: 0.9544481039047241 loss: 0.2065364122390747
epoch: 33 test accuracy: 0.9534863233566284 loss: 0.2154860943555832
epoch: 34 test accuracy: 0.9629727005958557 loss: 0.1869317591190338
epoch: 35 test accuracy: 0.9528743028640747 loss: 0.21775805950164795
epoch: 36 test accuracy: 0.9625355005264282 loss: 0.1950688660144806
epoch: 37 test accuracy: 0.9597377181053162 loss: 0.20305678248405457
epoch: 38 test accuracy: 0.9609180092811584 loss: 0.18695130944252014
epoch: 39 test accuracy: 0.9638907313346863 loss: 0.18611043691635132
epoch: 40 test accuracy: 0.9699671864509583 loss: 0.16401168704032898
epoch: 41 test accuracy: 0.9542295336723328 loss: 0.2071947604417801
epoch: 42 test accuracy: 0.967125654220581 loss: 0.1780594438314438
epoch: 43 test accuracy: 0.9666010737419128 loss: 0.17718955874443054
epoch: 44 test accuracy: 0.9638907313346863 loss: 0.1871069073677063
epoch: 45 test accuracy: 0.9713661074638367 loss: 0.16099700331687927
epoch: 46 test accuracy: 0.9695300459861755 loss: 0.169712096452713
epoch: 47 test accuracy: 0.9739453792572021 loss: 0.1588108390569687
epoch: 48 test accuracy: 0.9627978205680847 loss: 0.18608933687210083
epoch: 49 test accuracy: 0.9473661184310913 loss: 0.22979983687400818
epoch: 50 test accuracy: 0.9693551659584045 loss: 0.16487213969230652

final results: accuracy: 0.9693551659584045 loss: 0.16487213969230652

```

Figure J.1: LSTM Training/Test Loss and Accuracy

K Design Archive Guide

A tree representation of this Project Design Archive is represented in Figure K.1. This image has been created through the windows command prompt using the tree command in the designed directory.

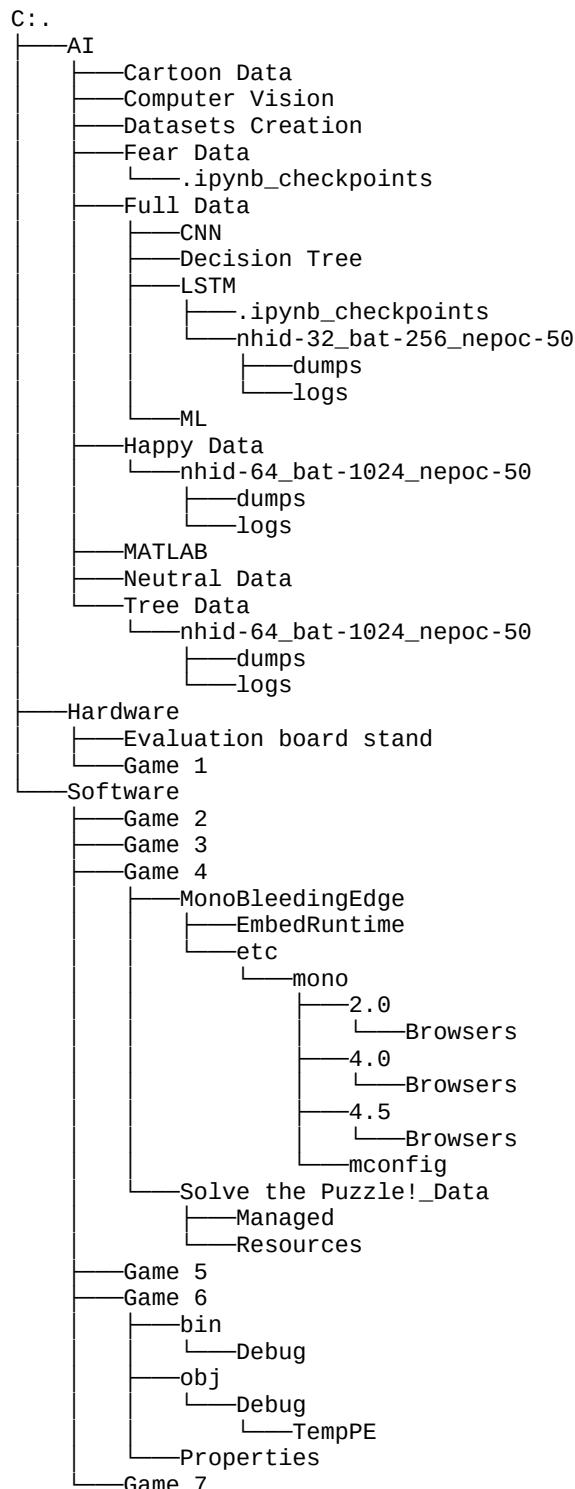


Figure K.1: Design Archive

L Word Count

The registered word count for this report, starting from Chapter 1 to Chapter 6, was equal to 9996 words. This has been measured using PDF Word Counter [41].