POLITECNICO DI TORINO

Master's Degree in Ingegneria biomedica
Strumentazione biomedica

Development of quantitative signal and image analysis techniques in speech therapy and psychomotricity

Supervisor

Prof. Filippo MOLINARI

Co-supervisor

Prof. Massimo SALVI

Tutors

Dott. Francesco PETRIGLIA

Dott. Andera MEIRONE

Candidate

Marco IRIDE

March 2023

Abstract

The linguistic assessment of the developmental age is an extremely delicate task to perform as the problem is influenced by several factors, and a diagnosis based on quantitative data could help the standardisation process of clinical assessment. Currently, assessment operations rely heavily on the individual practitioner's experience, and it is in this context that the project carried out through a collaboration between the Turin Polytechnic and the Paideia Foundation comes to life. The aim of the study is to create automatic computer programs for the extraction of quantitative parameters to help standardise the assessment of language and motor-oral skills of children of developmental age. The problem was approached based on the theory of dynamic systems, whereby the skills analysed in children were seen as the result of the sum of several components, seeking to overcome the dichotomy between the motor and phonological approaches, which will be seen as two aspects that run together with the child's linguistic production as well as its speech and psychomotor assessment. The dataset used for the study was collected by the Paideia Foundation and comprised 147 children, 125 of whom came from three different schools in Turin, to which were added 22 clinical cases under analysis at the foundation itself. The children were video-recorded by means of a PC frontal camera while performing some language production exercises, for the talktiz project, carried out by the foundation for language assessment. For the purpose of identifying children with language disorders through an oral-motor analysis, a facial mesh was applied to all of the children being analysed using the MediaPipe framework (created by Google), from which relevant measures were then retrieved. The second phase of the project focused instead on extracting information from the audio signal to distinguish clinical cases from normative cases and thanks to openSMILE software, 6373 features were extracted, that were used in the training of a classifier based on machine learning techniques. Three different classifiers were analysed for our purposes:

- K-nearest neighborhood (**K-NN**)
- Support vector machines (**SVM**)
- Feed forward neural network (**FFNN**)

Classifications were carried out taking only audio features into account, precisely because attempting to correlate audio and video features for the type of dataset being analysed was found to degrade the performance of the classifier.

Contents

List of Figures

List of Tables

Acronyms

K-NN
K-nearest neighborhood

SVM
Support vector machines

FFNN
Feed forward neural network

SLD
Specific language disorders

DVD
Development verbal dyspraxia

DAS
Development apraxia of speech

CAS
Childhood apraxia of speech

HOG
Histogram of oriented gradients

SSD
Single shot detection

ML
Machine learning

DLT

Discrete linear transformation

DDK

Diadochokinetic

EEND

End-to-end neural network

PIT

Permutation-invariant-training

DPCL

Deep Clustering loss function

BLSTM

Bidirectional long short-term memory

LSTM

Long short-term memory

LLD

low-level descriptors

MFCC

Mel Frequency Cepstral Coefficients

NCA

Neighbourhood components analysis

PCA

Principal component analysis

LOO

Leave-one-out

RMSProp

Root mean square propagation

FDDB

Face Detection Data Set and Benchmark

SRN

Selective Refinement Network

STC

Selective Two-step Classification

STR

Selective Two-step Regression

RFE

Receptive Field Enhancement

DSFD

Dual Shot Face Detector

FEM

Feature Enhancement Module

PAL

Progressive Anchor Loss

IAM

Improved Anchor Matching

GIF

Glottal inverse filtering

QCP

Quasi-closed phase method

ROC-AUC

receiver operating characteristic-area under the curve

ENT

Ear-Nose-Throat

Chapter 1

Communicative analysis of the developmental age

Language disorders are one of the most studied problems in the field of childhood language rehabilitation. Even today, it is difficult to have a single international classification, and this is caused both by the difficulty of understanding the nature of the disorders themselves and by the lack of a standardised and quantitative assessment process. Even though numerous studies have been done in an effort to identify the root reasons of linguistic problems, these factors are still unknown. Which is why multi factorial influences are hypothesised that are difficult to identify. At the basis of language disorders we have both cognitive and motor-praxic aspects but following the theory of dynamic systems [1] they can also be influenced by inadequate interactions between the child and the environment.

1.1 Typical evolution of motor-linguistic skills

1.1.1 Motor and communication development

The acquisition of motor patterns occurs throughout the early stages of language development. Specific movements will then manifest in one of two ways:

- *Gross-motor* skills: used to perform extended movements in space
- *Fine-motor* skills: used to perform precision movements

Three separate systems have an impact on how these motor skills develop: central nervous system, biomechanics of the neuromuscular system, Environmental characteristics. Cognitive, neurological, and biological elements all affect how motor patterns and related skills evolve. These factors can influence the development of those automatic movements that emerge over time.

1.1.2 Building verbal and expressive skills

Motor and linguistic abilities are both used in communication, which is the process of interaction that permits information to be passed. Humans have a phono-articulatory apparatus made up of breathing-related structures, which is necessary for sound production. Where air that has exited from the lungs flows down the trachea is the larynx, which contains the vocal folds. The muscles of the vocal folds vibrate to produce sound waves, which subsequently travel to the supra-laryngeal cavities. After passing the pharynx, the air encounters the soft palate, a structure made of flexible muscle tissue that allows air to pass through the mouth and nasal cavities during respiration. During phonation, it can be raised or lowered to regulate the airflow from the nasal cavities. Finally, the presence of the tongue and lips modifies sound produced.

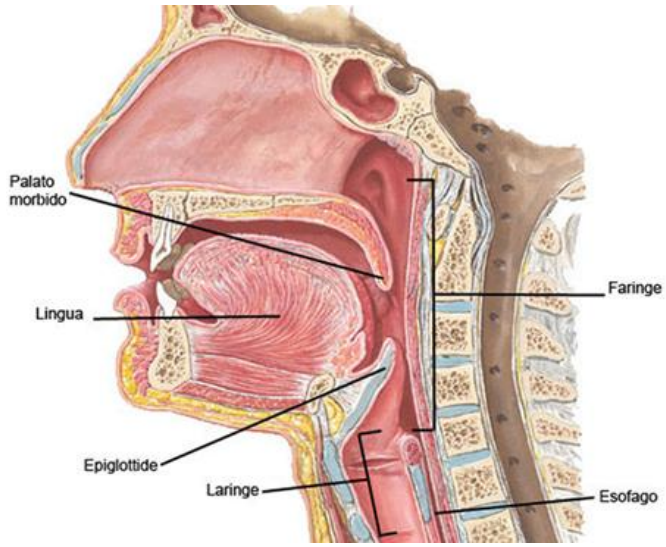


Figure 1.1: Sagittal diagram of the phonatory apparatus
[2]

As the product of the coordination of several abilities a child learns during development, language is a complicated adaptive system that is influenced by the environment, emotional factors, motor factors, and metacognitive factors. Furthermore, articulatory performance features are a crucial part of language development.

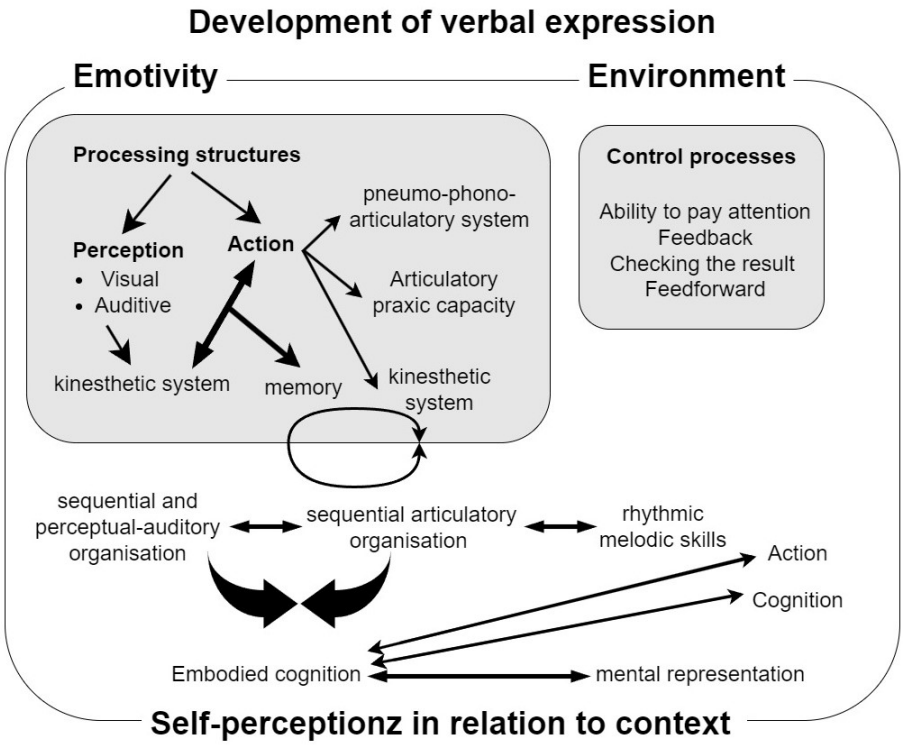


Figure 1.2: Correlation between motor development, executive functions (FE) and verbal expression skills

[1]

Without the acquisition of *fine-motor skills*, the development of linguistic abilities cannot proceed, which is why linguistic analysis and joint analysis are conducted concurrently. To produce a precise communication output, the child must also be able to organise movements according to sequences and patterns. Nevertheless, there are a number of other factors that might have an effect on the assessment of language competence including:

- Emotional state
- Context
- Process control
- Cognitive aspects
- Neurological aspects connecting eyes-hands-mouth
- Motor organisation

1.2 Atypical evolution of motor-linguistic skills

The most common childhood language disorders are:

- Dyspraxia
- Specific Language Disorder (SLD)

1.2.1 Verbal Dyspraxia

Because of its poorly defined diagnostic criteria and unclear aetiology, verbal dyspraxia is still a difficult illness to assess. In the literature, dyspraxia is frequently described using the terms:

- Development verbal dyspraxia (DVD)
- Development apraxia of speech (DAS)
- Childhood apraxia of speech (CAS)

The use of the **CAS** term is intended to emphasise severe deficits of the phonatory apparatus. The name **DAS** puts the focus on the verbal and articulatory production. The term **DVD** refers to a language-based developmental verbal expression problem. Some characteristics of dyspraxia are easily identifiable in:

- Difficulty in voluntary articulatory control to carry out verbal expression
- Difficulties in planning and managing sequential movements

A typical child with verbal dyspraxia has higher linguistic abilities for recreating ambient sounds, iconic or emotional expressions. Compared to a normally able child, the dyspraxic child has difficulty producing speech spontaneously or on demand. The dyspraxic child's linguistic output is inconsistent because he is unable to locate and maintain sites of articulation (*groping*). Similar to stuttering, language production may be slowed down as a result of the groping. *Verbal dyspraxia* is different from *oral dyspraxia* even though they may in some cases coincide. There is a noticeable deficit in the sequential processes in the case of verbal dyspraxia without oral dyspraxia, while on the other hand, there is a lack in the development of the phonatory apparatus, which affects verbal production. The word "DVD" refers to one of the most severe forms of dyspraxia, which includes voluntary automatic dissociation, hypo-fluent speech, problems with articulatory programming, phonological errors, inappropriate co-articulation, and hypo-fluent speech. Children who have Down syndrome or Williams syndrome, for example, may also have "DVD". Distinctive features of DVD are:

- Low motor coordination
- Deficits in verbal production capacity
- Difficulty in the production of syllables, sounds and words
- Disturbance in prosody, suprasegmental and metalinguistic traits
- Perceptual phonetic difficulties Disorders in reading and writing skills

1.2.2 Specific language disorders (SLD)

They are a range of disorders in one or more language development areas that have been identified in children with an IQ of at least 80. Children with SLD present problems in several areas such as phonetics, phonology, semantics, morphology and syntax. It is critical to know which phase of the SLD one is in since early intervention in this condition may also be essential for successful rehabilitation and recovery:

- Emergency phase (18-36 months) specific development does not occur or occurs in an immediately atypical manner
- Structuring phase (36 months-5 years) stabilisation of SLD in differential disturbances
- Transformation phase (4-5 years) secondary neuropsychological and psychopathological disorders occur
- Secondary disturbance structuring phase,it presents itself until adolescence with the predominance of learning and psychopathological disorders

Familiarity is one of the possible causes of SLD. According to recent research, some characteristics on chromosomes 16q and 19q may play a role in language disorders, particularly in cases with isolated phonological disorder. Between the first and second year of life, the presence of recurrent and fluctuating otitis might be considered a source of impairment in the proper discrimination of sounds, resulting in insufficient acquisition of the rules of language itself. There are different classifications of SLD, the international classification given by the World Health Association and the classification according to the American Psychiatric Association are among the most famous. According to the worldwide ICD-10 classification, SLD is a condition in which the development of typical language abilities is interrupted without any neurological delay abnormal physiological processes or external causes. We can distinguish three subgroups of SLD:

- Specific articulation disorder

- Expressive Language Disorder
- Receptive language disorder

The subdivision of the American Psychiatric Association is based on the distinction between phonation disorder and more specific types of disorders, such as language expression disorder and mixed disorder of expression and reception. Language difficulties cannot be the only factors included in the definition of SLI. Due to the heterogeneity of this illness, multiple cognitive processes are compromised. The development of motor abilities is influenced by genes linked to communication issues. Additionally, there is a strong relationship between linguistic and metacognitive skills. The difficulty in learning sequential movement is frequently thought to be associated with the language difficulties of children with SLI. Balance tests and the use of fine motor skills are typically quite difficult for children with this disorder.

1.3 Evaluation of the child: state of the art

There are various formal and specialised evaluation examinations available for the Italian-speaking population. Although this is one of the limits of the evaluation itself as the test is susceptible to being influenced by various external influences, the most accurate assessment is still conducted in the presence of spontaneous language. When children are playing or engaging in other activities, it is customary to record their conversations. The average length of the utterances, accounting for articles, prepositions, pronouns, verbs, and nouns, is typically obtained from the recordings as a point of reference. A minimum of 50 utterances are analysed in order to assess how linguistic production can change over time as it transitions from telegraphic language to ever-more-complex lexical ones. Playtime observation of the child has consequences for analysing the growth of symbolic abilities connected to the linguistic domain. The video recording also provides information on the child's gestural production. It was discovered through McNeill's studies in 1992 [1] that gesture and word are based on a common communicative process, which is why a "neurogestural" model is frequently utilised. In the "neurogestural" paradigm of communication, actions within Broca's region are arranged into gestural images. The assessment of SLI is made using standardised tests appropriate for various age groups. These tests assess linguistic production, verbal, and morphosyntactic understanding by looking for free morphemes associated with morphological deficits. Other predictive indices for evaluation are:

- Absence of lallazione from 5 to 10 months (vocal expression present in the child after 6 months that causes the repetition of chanted syllables)
- Absence of deictic (indicative) or referential gestures, i.e. referring to a specific referent

- Failure to acquire patterns of action
- The capacity for symbolic play has not developed (24-30 months)
- Deficiency in understanding non-contextual orders
- Presence of idiosyncrasies after 30 months

The diagnosis of SLI is hindered by the ability to interpret the data provided and the few instruments available, which is why the data must be correlated with the results of the tests used for assessment. SLI in children is detected after the age of three, and two groups of children are identified:

- **Late bloomers** with delayed language development catching up within a year
- **Late talkers** who differ from normatives in the area of language (comprehension and production)

The child's IQ must be at least 80 for a diagnosis of SLD, and the language assessment must be at least 1-2 standard deviations below average. The greatest technique to accurately assess patients is using a multisystemic approach to diagnosis. In the first age range being studied (between 3 and 4 years), it is critical to evaluate the child's exposure to both parental and environmental language. The greatest technique to accurately assess patients is using a multisystemic approach to diagnosis.

Chapter 2

Oral-motor analysis:methods and processes

The dataset used for the study considers a sample of 147 normative and clinical children. The children were observed while participating in language-production activities created by the Foundation's internal **talktiz** programme. The kids were asked to describe particular images created by this computer programme (see. **Figure (2.1)**):

Figure 2.1: Examples of the talktiz software's images that children are required to describe

Before the video files were used, the children were assessed by health personnel, obtaining a sample of 125 normative children from three different schools in Turin and 22 clinical children from patients treated by the foundation itself. Video and audio signals of varying lengths [6min-20min] were acquired via the front camera of a PC with a resolution of 720p. The choice of the acquisition medium was made

with consideration for the value of spontaneous production in the clinical evaluation in order to minimise potential disruptions from an audio-video recording device that was more effective but might have drawn the child's attention more during the exercise, invalidating the clinical evaluation. The dataset considered has several inherent problems, which are not limited only to the equipment used to make the acquisitions. The video files often present data from which it is difficult with the methods used to extract clinically significant parameters since it was preferred to leave the children as free as possible in their movements.

2.1 Face detection

In the implemented programs, the first operation performed was face detection within the videos. The choice of the face detection algorithm took into account two main factors:

1. Computational cost
2. Accuracy

The detection of faces and the localisation of their position within the image requires independence from

- Position
- Orientation
- Scale
- Facial expression

Face detection can be influenced by external factors such as lighting or the complexity of the background being analysed. The algorithms can be distinguished into:

- **Feature-based** techniques: exploit low-level features to give a definition to the human face
- **Image-based** techniques: the problem is considered as a generic pattern recognition problem, whereby the image of a face is recognised from some training examples

Three different face detection algorithms were examined: the Viola-Jones algorithm and two others that exploit Python's Dlib and MediaPipe toolkit, respectively. All toolkit that are used for research in this thesis are configured in the mode of use only CPU, without a GPU (NVIDIA Cuda).

2.1.1 Viola-Jones algorithm

The Viola-Jones algorithm is based on a classifier trained using multiple instances of the class to be identified (positive samples) and a series of images without the class under consideration (negative samples). During training, features are extracted from the samples and only those useful for classification are selected. This algorithm is an image-based method whereby the classifier under analysis associates a specific detected pattern with the face or non-face class. The basic idea is that faces have common properties, whereas images that do not represent faces are highly irregular. The Viola-Jones classifier is based on three different contributions:

- Extraction and evaluation of Haar-like features
- Classification by boosting
- Multiscale detection

Prior to being sent to the classifier, the image was also converted grayscale, and the adaptive equalisation of the histogram was utilised to increase local contrast. Before applying the classifier, the integral image is calculated (see. **Fig.2.2**). The value of each pixel is the sum of all pixels above and to the left including the target pixel and then calculates the sum of the pixels in the orange rectangle following the formula $D - B - C + A$. The Integral Image is computed in order to reduce computational cost.

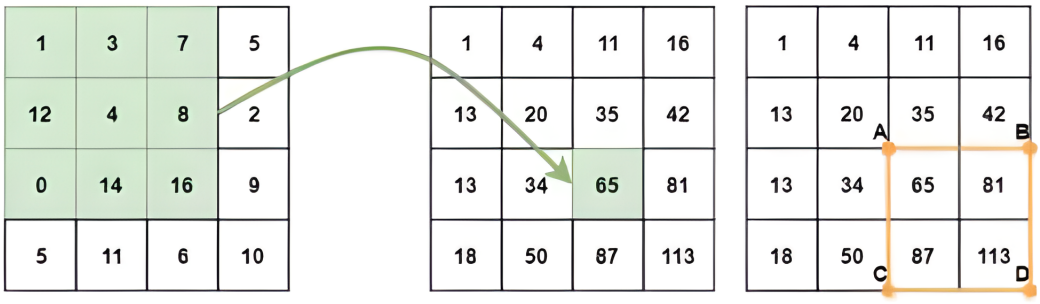


Figure 2.2: Calculation of an Integral Image
[3]

During face detection, a window of variable size is scrolled over the image and the features in the window are extracted to determine whether there is a face in the window or not. The classifier used is trained for face detection in a frontal position and also exploits the AdaBoost technique to improve performance. AdaBoost has

the task of constructing a complex non-linear classifier from a linear combination of (M) simpler weak classifiers (see. **Fig.2.3**).

$$H_M(x) = \frac{\sum_{m=1}^{M} \alpha_m h_m(x)}{\sum_{m=1}^{M} \alpha_m} \tag{2.1}$$

In the equation (2.1) x is a pattern to be classified, $h_m(x) \in \{-1, +1\}$ are the weak classifiers, $\alpha_m \geq 0$ are the weights, $\sum_{m=1}^{M} \alpha_m$ is the normalisation factor.

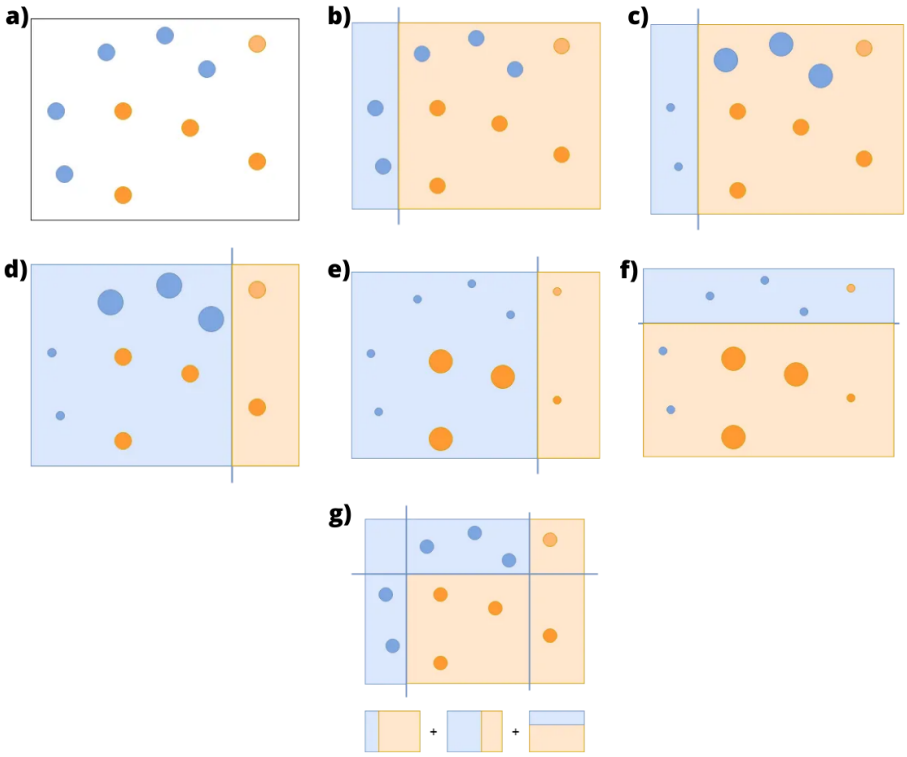


Figure 2.3: AdaBoost example of classification; **a)** Samples to be classified; **b)** Implementation of the first weak classifier; **c)** Blue samples wrongly classified are given a higher weight; **d)** Implementation of the second classifier; **e)** Orange samples not correctly classified are given a higher weight; **f)** Implementation of the third classifier; **g)** Combination of the three weak classifiers
[3]

Each weak classifier represents a stage of the stronger classifier. When an image region enters the classifiers cascade, it is evaluated starting with the first stage if it gets a positive evaluation from the first classifier then it will be sent as input to the second classifier

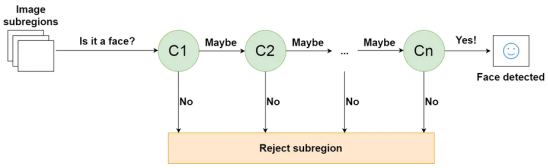


Figure 2.4: Cascade of classifiers
[3]

In the Viola-Jones algorithm, basic features called Haar-like features are used. Each feature is placed in a sub-region of a sub-window of the image with different dimensions. Haar features are used in image processing to classify the intensity of pixels, and they are typically represented as rectangular regions of the image. The classifier consists of two or three rectangular, which detect features continuously within the window.

Figure 2.5: Haar features
[4]

2.1.2 Dlib toolkit algorithm

This classifier is based on the Histogram of Oriented Gradient technique used successfully for object detection. The object to be detected is represented as a feature vector which identifying specific regions of space. The calculation is performed for all regions of the image, varying the scale. The HOG-based classifier is first trained through the use of positive samples from the training data, from which the HOG descriptors are extracted. Next, negative samples containing no face are used and HOG descriptors are also extracted from these samples.

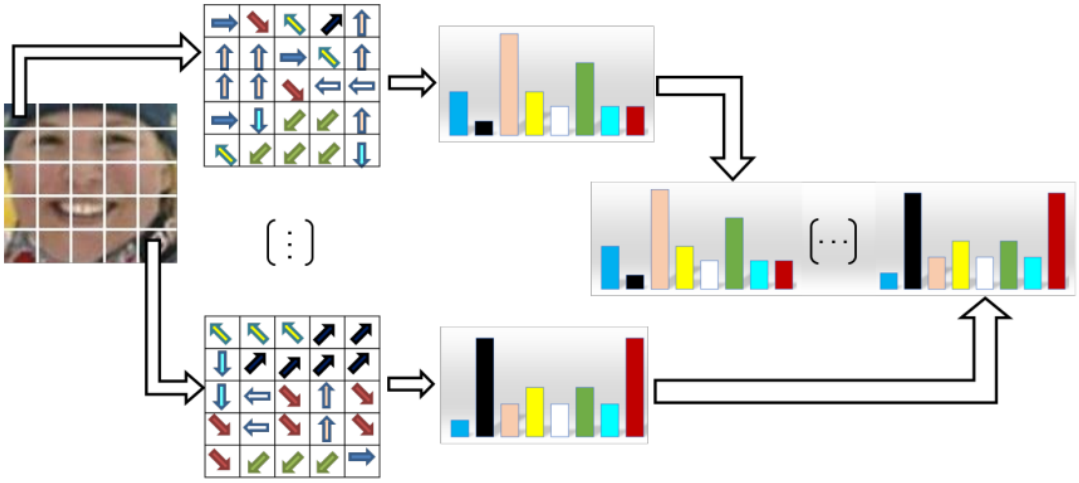


Figure 2.6: Calculation of the histogram of oriented gradients (HOG). Each cell's gradient orientation is determined, and the histograms are all combined to produce the global HOG descriptor.

[4]

A sliding window is applied to each image, in each window extracting the HOG descriptors. This technique counts the occurrences of the gradient orientation in a localised portion of the image. A histogram is generated for the regions under image analysis using the magnitude and orientation of the gradient as characteristics. HOG descriptors are based on the structure and shape of the object and extract information on the magnitude and angle of the gradient from which the histogram is derived. To apply the analysis using HOG, the image is first divided into several connected components called **Cells**. For each Cell, the gradient is calculated pixel by pixel, producing the histogram. We then obtain the image descriptor from the linear combination of all cells in the image. Prior to feature extraction the image has undergone a process of Gamma normalisation, this action aims to eliminate the impact of highlights and shadows in local regions of the image [5]. In **Equation 2.2**, $I(x, y)$ denotes the pixel intensity in grayscale, γ represents the constant for

gamma normalisation, for which a typical value is 0.5. **Equation 2.3** shows the calculation of the first-order gradient including both magnitude and angle.

$$I\left(x,y\right) = I_0\left(x,y\right)^{\gamma} \tag{2.2}$$

$$Grad\left(x,y\right) = \sqrt{\left(\left(I\left(x+1,y\right) - I\left(x-1,y\right)\right)^2 + \left(I\left(x,y-1\right) - I(x,y+1)\right)\right)^2} \tag{2.3}$$

$$Ang\left(x,y\right) = \arccos\left(\frac{I\left(x+1,y\right) - I\left(x-1,y\right)}{Grad\left(x,y\right)}\right)$$

After feature extraction, a support vector machine (SVM) with a linear kernel is applied as classifier.

2.1.3 Mediapipe toolkit algorithm

The face detection algorithm in the MediaPipe toolkit, is based on BlazeFace, which is a computationally light and high-performance face detector, so that it can also be adapted to devices with mobile GPU. BlazeFace is based on the structure of a convolutional neural network called MobileNetV2 [6] [7]. MobileNetV2 uses depthwise separable convolution as network building blocks. Version V2 introduces two new features compared to version V1:

- Linear bottlenecks between the layers
- Shortcut connections between the bottlenecks

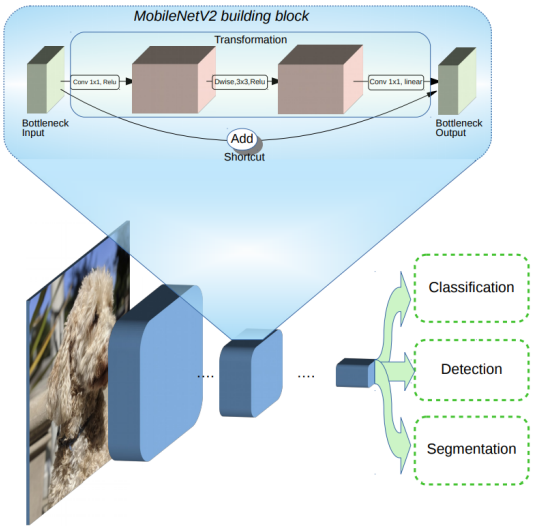


Figure 2.7: MobileNetV2 architecture [8]

The model's intermediate inputs and outputs are encoded by bottlenecks. Shortcuts then allow for faster training and better accuracy. The BlazeFace model produces six facial coordinates as key points (for the eyes, ears, mouth and nose) the use of these six key points is to estimate the rotation of the face, to reduce the need for invariance with respect to translation and rotation.

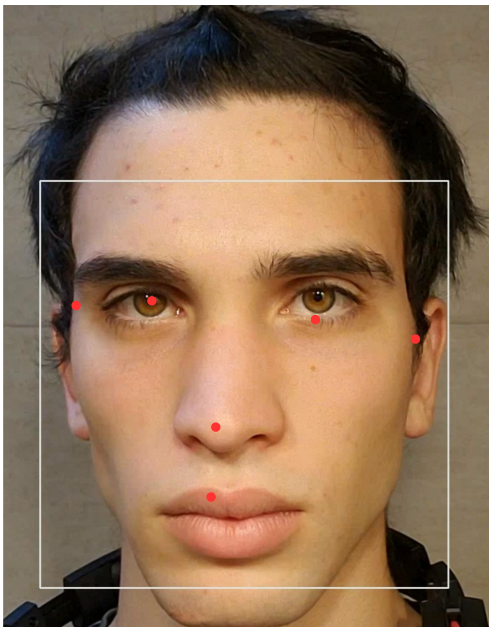
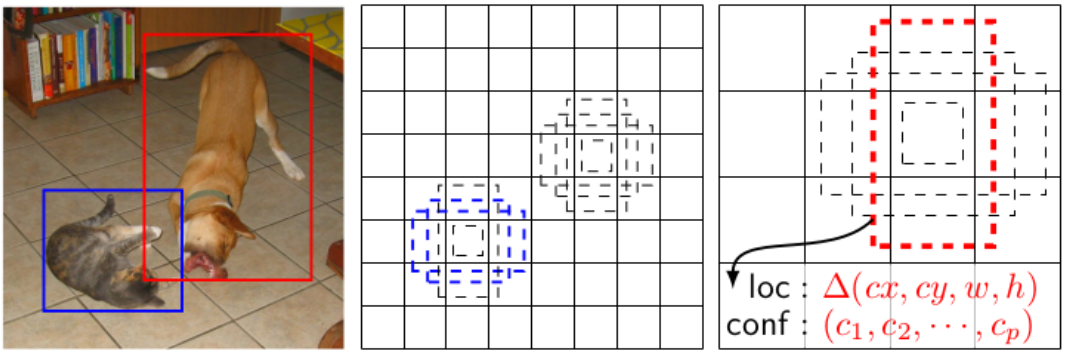


Figure 2.8: Face detection result by MediaPipe, in red are identified the six key points used to estimate the rotation of the face
[9]

The model used focuses on identifying the face in the frontal position in order to refer to a smaller scale of objects, and therefore has lower computational requirements. The BlazeFace model is a SSD (single shot detection) object identification model that is based on predefined, fixed-size bounding rectangles called anchors. For each anchor, parameters such as offset centre and dimensions are set. The anchors are defined at different resolutions to fit with the scale range of the object. The SSD-based approach discretises the box output space into a set of predefined boxes of different proportions and scales. For each of these defined boxes, the network establishes scores for the presence or absence of each object category. The network also combines the predictions of several feature maps with different resolutions, to handle objects of different sizes. This is a faster and more accurate algorithm than the previous state-of-the-art (**YOLO**). Using a set of convolutional

filters applied to feature maps, the SSD model predicts a score and box offsets for a defined set of bounding boxes [10]. The algorithm only needs an input image and ground truth boxes for each object during training. In **Figure 2.9** there is an example of convolutional evaluation of a small set (e.g. 4) of predefined boxes at different aspect ratios, at each position in different feature maps and at different scales. For each box, we predict both shape offsets and confidences for all object categories.



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 2.9: SSD framework. (a) The image and the ground truth boxes for every object. We assess a small set of default boxes with various aspect ratios at each position in a number of feature maps with various scales (for example, 8x8 and 4x4 in (b) and (c)).. We forecast the shape offsets and confidences for all item categories for each default box (c1, c2, ... , cp). At training time, we begin by matching default boxes to the boxes from the ground truth
[11]

2.2 Mesh creation and adaptation

The definition of a facial mesh for the extraction of several key parameters was the second step in the extraction of the oral-motor features. The physiological measures used were determined after an examination of the scientific literature on the issue and in response to requests from the Foundation's clinical personnel. Two potential implementation solutions for the construction of a facial mesh have been investigated for our objectives. The first is based on the well-known CANDIDE model [12]. The second facial mesh tested was created via Google's MediaPipe toolkit.

2.2.1 CANDIDE model

To achieve a rapid and computationally efficient reconstruction, the CANDIDE model parametrizes facial features using a small number of polygons, roughly 100. The CANDIDE model is based on the concept of Action Units (global and local), which are the fundamental movements of single muscles or groups of muscles. The global action units govern the rotation around the three axes, meanwhile the local action units control the facial expressions. The original CANDIDE model had 75 vertices and 100 triangles but it is no longer in use. The CANDIDE-3 model, introduced in the 1990s, is currently the most common. This model originally had 113 vertices and 168 surfaces, however in version 3.1.4, the number of surfaces has increased from 168 to 184. For this research, we employed version 3.1.6 of the CANDIDE model, removing some triangular surfaces connected to the forehead which were not useful for the acquisition of parameters. As a result, the final model has 113 vertices and 175 faces.

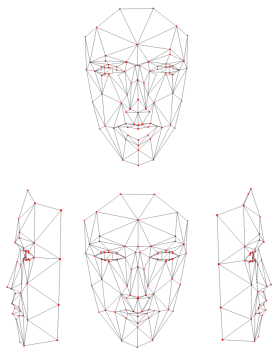


Figure 2.10: CANDIDE face mesh model
[13]

When using the CANDIDE facial mesh, we start with a neutral state of the face that corresponds to the model's vertices and surfaces. To align the constructed facial mesh to the face in the image, we must adapt the mesh's coordinates defined in a 3D space to some 2D markers found on the face. The shape predictor provided by the Python Dlib toolkit was used to obtain the 2D position of 68 important points on the face.

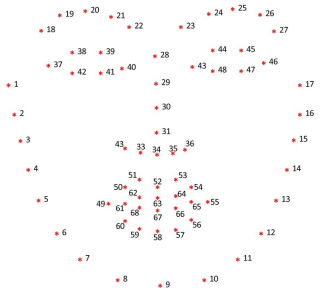


Figure 2.11: 68 markers from Dlib's shape predictor

All techniques to extract key points on the face attempt to identify specific regions such as:

- Mouth
- Eyebrows
- Eyes
- Nose
- Jaw

The Dlib toolkit's facial landmark detector [14] is based on a series of training images in which the face markers have been manually identified by specifying their (x, y) coordinates. The prior probability on the distance was then calculated for each pair of input pixels and combined with the gradient boosting approach to generate a sparse pixel set for use as input. A sequence of regression trees are trained using the dataset with the goal of identifying the positions of face landmarks in real time directly from the intensities of the pixels themselves. Some of the action units available in the CANDIDE 3.1.6 model were considered in the subsequent steps, to adjust the mesh to the typical movements of the face. The aim of these action units is to mimic the facial deformations, modifying the mesh for each frame to match the expressions on the face. In particular, 14 animation units were chosen, 9 of which match to those in the CANDIDE 3.1.6 model and the remaining 5 to asymmetric facial movements. The facial mesh, whose points are given in 3D space, is projected for each frame using the given equations [13]:

$$s = aP\left(S_0 + \sum_{i=1}^{i=n} w_i S_i\right) + t \tag{2.4}$$

Where s is the projected shape, a is the scaling factor, P are the first two rows of a rotation matrix that rotates the 3D mesh [13]. S_0 is the neutral shape w_i are

some S_i weights, S_i are the blendshapes,which include the animation unites. All six degrees of freedom, three translational and three rotational, known as Euler's angles, must be known in order to represent a rigid body in three dimensions. To project the three-dimensional points given by the facial mesh in two dimensions, add the 14 parameters that correspond to the animation units to the six that represent the rigid body. First, the three-dimensional positions of the mesh in the neutral state are computed, which correspond to the 68 markers predicted by the Dlib model. The mean is subtracted from 2D and 3D measurements,and the initial parameters are computed.

- The scale factor of the mesh
- The translation vector (t) between the 3D and 2D indices

After obtaining the initial parameters, the mesh was projected and adapted using the Gauss-Newton least squares reduction method. Model fit is obtained by reducing the difference between the projected shape and the localised reference points. In this situation, the **Equation 2.4** represents the function to be reduced using the Gauss technique. Using the Rodrigues formula,we pass from the original rotation vector to the rotation matrix.

$$R = I + (sin\,(\theta))\,K + (1 - cos\,(\theta))\,K^2 \tag{2.5}$$

Where R is the rotation matrix, θ is the rotation's angle, I is the identity matrix and K is the unit vector's matrix expression, which denotes the rotational axis. We take the first two rows of the rotation matrix, which define the rotation in 2D space. The outcomes of applying the function for the Gauss-Newton algorithm are known as residuals $r = (r_1, ..., r_m)$, which are functions of n variables $\beta = (\beta_1, ..., \beta_n)$. To minimize the sum of squares (**Equation 2.6**), the algorithm uses an iterative approach:

$$S(\beta) = \sum_{i=1}^{m} r_i^2\,(\beta) \tag{2.6}$$

This equation is a cost function, which is used to valuate the alignment between the mesh and the face. A threshold value is specified below which the cost function is considered acceptable and the CANDIDE model is declared properly adapted to the face. To use the algorithm, we must compute the Hessian matrix (**Equation 2.7**) and gradient (**Equation 2.8**) using the following equations:

$$H_{jk} = 2\sum_{i=1}^{m}\left(\frac{\partial r_i}{\partial \beta_j}\frac{\partial r_i}{\partial \beta_k} + r_i\frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k}\right) \tag{2.7}$$

$$g_j = 2\sum_{i=1}^{m} r_i\frac{\delta r_i}{\delta \beta_j} \tag{2.8}$$

Defining the Hessian matrix and the gradient as Jacobian functions:

$$H_{jk} \approx 2\sum_{i=1}^{m} J_{ij}J_{ik} \tag{2.9}$$

$$g_j = 2\sum_{i=1}^{m} r_i J_{ij} \tag{2.10}$$

The terms with second derivatives are ignored while computing the Hessian. The **descent direction** is now calculated as:

$$\Delta = -\frac{g}{H} \tag{2.11}$$

Performing the minimization while knowing the direction yields the stride length (α). Finally, for each iteration, the parameters are calculated as:

$$x = x + \alpha\Delta \tag{2.12}$$

Where x are the variables of **Equation 2.4**. Figure (2.12) shows the results of mesh adaptation.

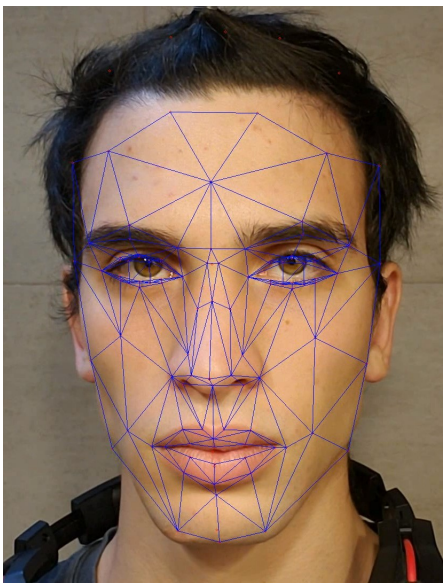


Figure 2.12: Face mesh adaptation of CANDIDE model [9]

2.2.2 MediaPipe face mesh

Using the MediaPipe toolkit solution, 468 3D markers on the face can be estimated in real-time, in addition to 10 extra markers for irises. The system uses machine learning (ML) techniques to estimate the surface of the face and has been optimised to function on mobile devices. No depth sensor is necessary for the estimation; only an input camera is required. In order to create the face mesh, MediaPipe employs a simple statistical shape analysis technique known as **Procustean analysis**. The ideal rotation and orthogonal linear transformation for the overlap of two objects can be discovered using this statistical shape analysis. Procustean superposition is obtained by suitably translating, scaling, and rotating the objects. Two deep neural network models are used to operate the *mesh definition* pipeline. The first model is the same used in during the face detection step, whereas the second operates on the face position and uses a regression model to predict the approximate 3D surface. This last model was trained utilising the transfer learning technique. Using the transfer learning method, it is possible to apply the information obtained solving one problem to another that is somewhat related. The network takes video frames as input and returns both the location of the 3D face markers and the likelihood that a face is actually present in the frame. **Bootstrapping** was used to estimate a population parameter from data through repeated sampling. In this instance, the facial markers are sampled multiple times, and the extracted samples are then re-incorporated into the population of points that represent the face. Therefore, even after numerous extractions, the likelihood of selecting that sample from the data population remains constant. Instead of producing a single point estimate, the bootstrap technique is based on the idea of providing a distribution of estimates. The mesh prediction model has a relatively simple residual neural network architecture, with the greatest sampling taking place in the network's initial layers, which are also where the most complex operations are concentrated. On the other hand, the deeper layer neurons's aim is to distinguish between the features of the mouth and eyes [15]. In addition to the model already mentioned, another model called **Attention** is used, which seeks to replicate human cognitive attention [16]. The gradient descent technique is used to select the data that the network should "pay attention" to. By more precisely predicting the points around the lips, eyes, and irises, the Attention is used to force the network to focus on those parts of the face that are considered to be semantically significant. The *flexible weights* that make up the Attention model have the advantage of being able to change as the algorithm is running. The face detection technique provides 256×256 images to the network as input.
The model is divided into many sub-models and a 64×64 feature map is extracted. The 64×64 feature map is utilised as input to one sub-model, which is used to predict all 478 face mesh landmarks.

The other sub-models predict the markers from the 24 x 24 ROI that are obtained from the Attention model.

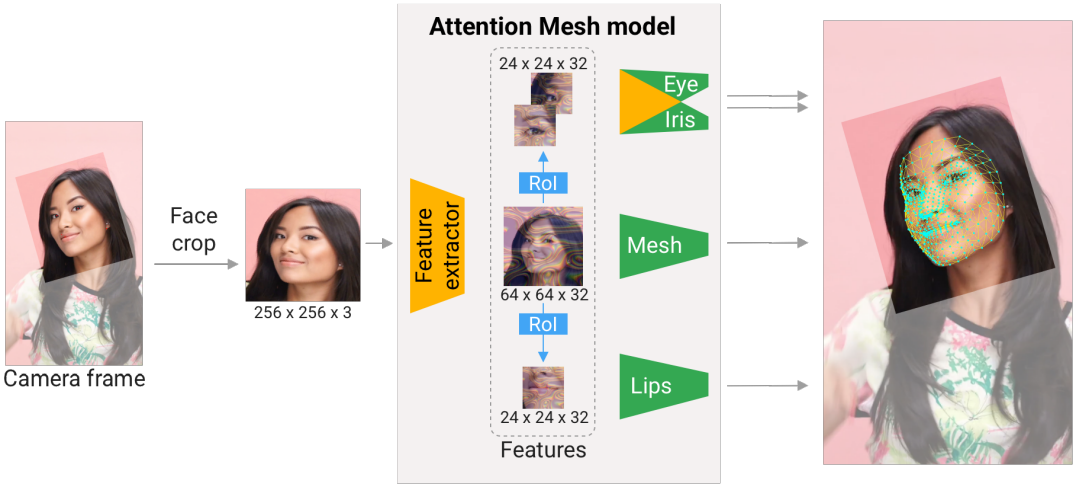


Figure 2.13: Attention Mesh model [16]

The lips and the two eyes are ROI extracted from Attention. Once a resolution of roughly 6 × 6 is achieved, the eye sub-models also predict the iris. There are two stages to training the attention network, the different models are first independently trained after an ideal crop is created taking into account the ground truth. The network is then retrained to fit the sub-models to the regions of interest using the image crops the model gives. By using techniques like affine transformations, differentiable interpolations, or 2D Gaussian kernels, the Attention mechanism samples a grid of 2D points and extracts features. The 24 × 24 feature regions are extracted from the 64 × 64 feature map using a spatial transformer module. A transform matrix regulates the spatial transformer.

$$\Theta = \begin{bmatrix} s_x & sh_x & t_x \\ sh_y & s_y & t_y \end{bmatrix} \tag{2.13}$$

The output of the sub-model defining the facial mesh can also be used to determine the parameters of this transformation matrix.

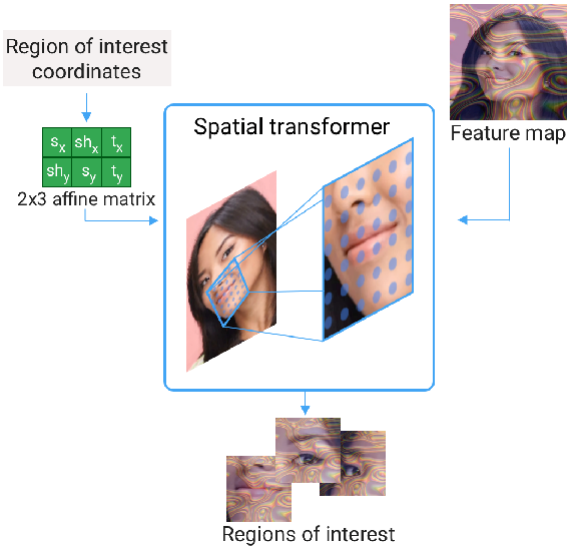


Figure 2.14: Spatial transformer of Attention technique [16]

An orthogonal 3D metric space with right-handed coordinates is defined in the MediaPipe face mesh pipeline. A virtual perspective camera is placed inside this area and is virtually placed at the origin of the area, pointing away from the Z axis. The input frames are assumed to have been observed by this camera with programmable parameters, if those parameters are used as closely to the camera's actual values as possible, higher results will result.

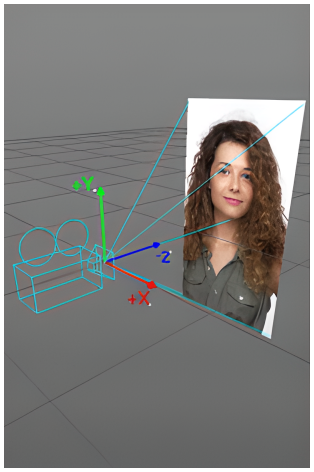


Figure 2.15: The orthogonal 3D metric space with right-handed coordinates [17]

The model is originally defined in a neutral position termed **Canonical Face Model**, same like with the CANDIDE face mesh. The scale of the Canonical face model, which is typically 1cm, determines the metric units of 3D space. The Θ transformation matrix of the face posture is a linear map that enables the change from the canonical model of the face to the set of reference points computed on each frame. A pipeline containing the fundamental processes to obtain the mesh points in 3D space is displayed below.

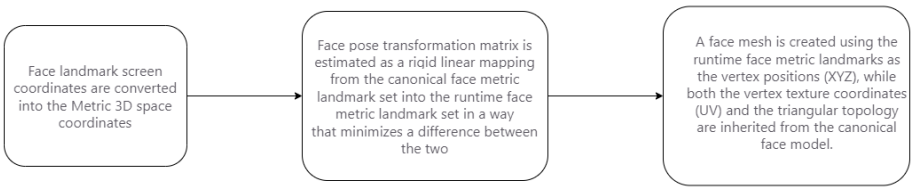


Figure 2.16: Pipeline of the process to get face landmarks positions

The MediaPipe mesh application's results

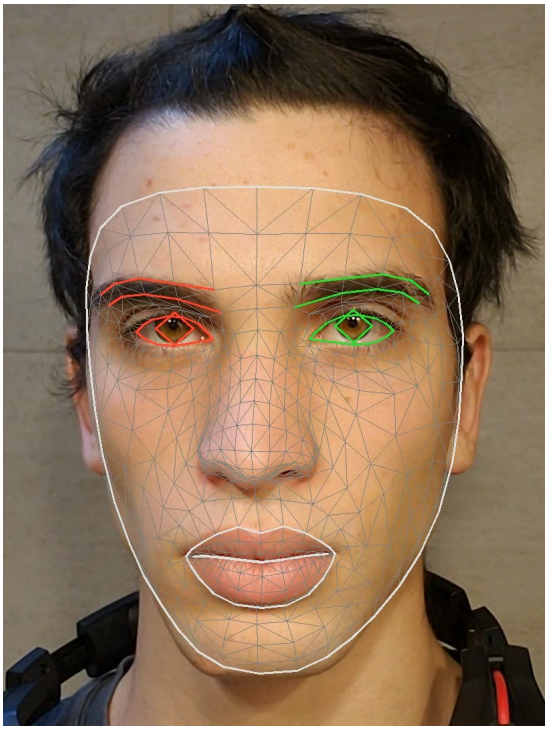


Figure 2.17: MediaPipe mesh adaptation on a face
[9]

2.3 Clinical features extraction

Several indicators that were obtained through the application of face meshing were thought by clinicians to be crucial in separating children into clinical and normative groups. After reviewing the scientific literature on the evaluation of facial movements in activities including speech and non-speech, other factors were considered vital. The child's face has to be in a frontal position in order to collect the parameters. In fact, it was found that the head's rotation frequently manifested as an artefact on the signals, to the point that they were clinically useless. The location of the head in space was determined using six facial markers (see. **Fig.2.18**).

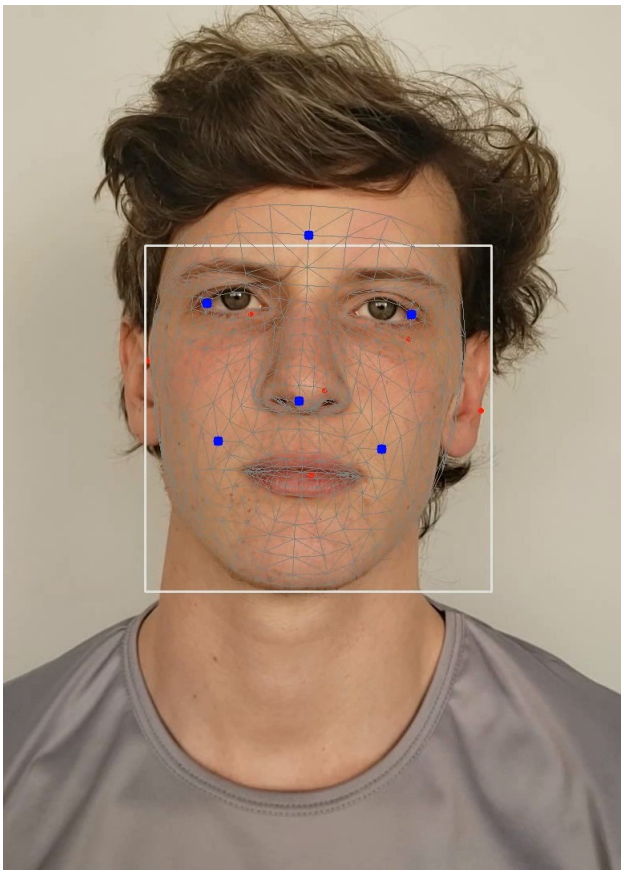


Figure 2.18: Markers used to make the head position are indicated in blue [9]

The focal length of the acquisition camera must be known in order to perform the estimation. Since it was hard to determine the precise focal length values for each acquisition PC, an educated guess was chosen instead.

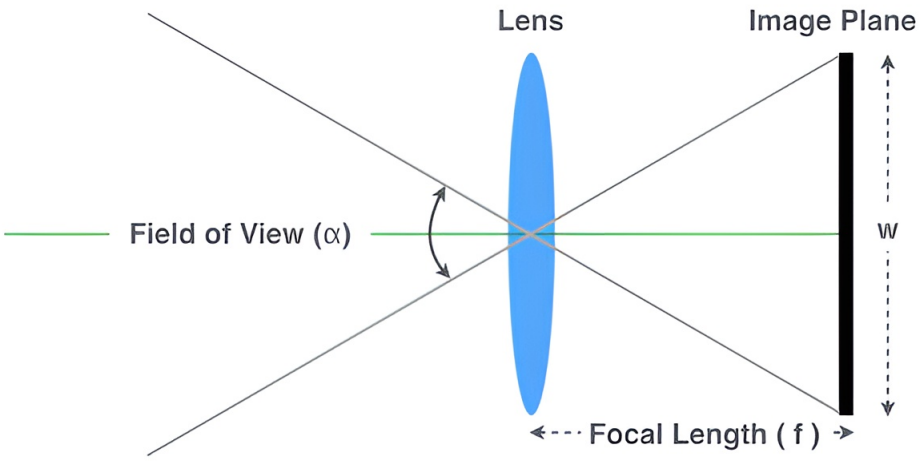


Figure 2.19: Field of view, focal length, and sensor size relationships [18]

Trigonometric relationships are used to determine how the field of view relates to the size of the image and the focal length

$$f = \frac{w}{2} \cot\left(\frac{\alpha}{2}\right) \tag{2.14}$$

Because you always require a specific face size for a distance that is appropriate for this type of device, webcams for PC and mobile phones typically have fairly comparable fields of view. These webcams typically have a field of view between 50° and 70°. for this reason, the focal length is between:

$$0.7w \leq f \leq w \tag{2.15}$$

In our case, using empirical evidence, it was decided to choose a focal length equal to:

$$f = w \tag{2.16}$$

After determining the focal length, we defined the camera matrix under the assumption that the camera's optical centre was precisely in the centre of our image.

$$CM = \begin{bmatrix} f & 0 & \frac{(h-1)}{2} \\ 0 & f & \frac{(w-1)}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.17}$$

Where CM is the camera matrix, f is the focal length, h is the height of the image and w is the width of the image. The camera distortion settings weren't established because no calibration was done, thus they were set to 0. The calculated parameters were used to solve the pose computation problem. The goal of this task is to determine the rotation and translation of the points in image space (2D) and the corresponding 3D that minimises projection error. We define the reference system of the camera (see. **Fig 2.20**)

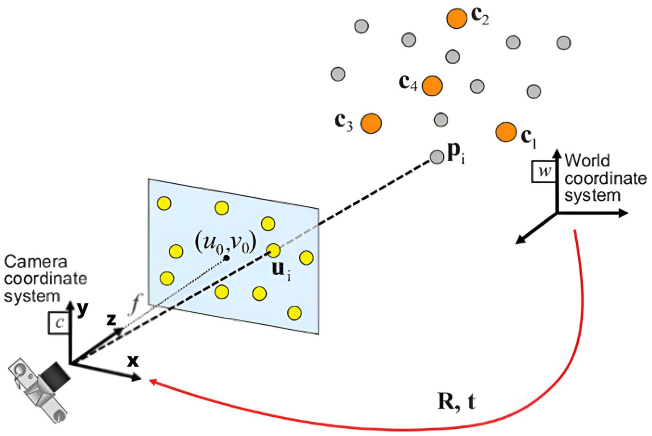


Figure 2.20: Reference system for pose estimation [19]

Using the perspective projection model (Π) and the intrinsic camera parameters A, the coordinates of the points in the word frame X_w are projected into the image plane $[u, v]$.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A\Pi^c T_w \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2.18}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2.19}$$

Consequently, the points in the camera's coordinate system are

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{22} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2.20}$$

T_w is the 3D matrix of translation and rotation of the camera, (f_x, f_y) is the focal length and (c_x, c_y) is the optical center. The projection approach is based on the iterative Levenberg-Marquardt [20] minimization method (it solves non-linear least squares problems similarly to the Gauss-Newton method). The discrete linear transformation (DLT) approach is used to find the initial solution before the rotation and translation vectors are calculated. The rotation vector was then converted into a rotation matrix using Rodriguez's formula. Consequently, using a QR decomposition, the angles of rotation around the three axes were determined. A limiting value was set using empirical experiments to determine the position of the head and identify frames where the child is not looking directly at the camera (see. **Fig.2.21**).

Figure 2.21: Head pose estimation

The extracted features were separated into the following categories: shape/geometry features, symmetry features, range of motion features, and speed of motion features. Distance measurements would change if there were movements in our reference system's z-direction, which stands in for the depth. To solve this issue, all distances determined between different face markers were normalised in relation to a distance that was assumed to be constant during the acquisition, namely the inner canthal distance.

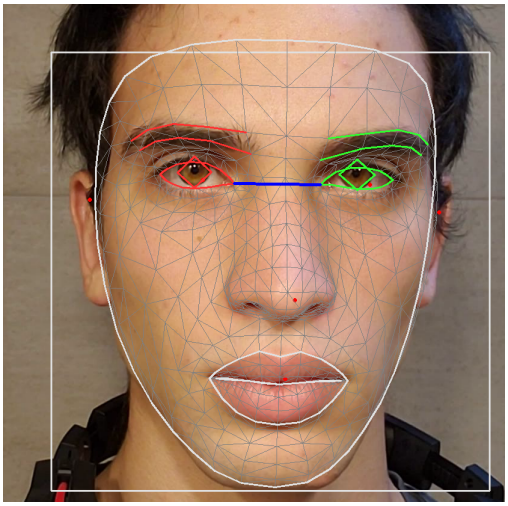


Figure 2.22: Measurement normalisation distance on a MediaPipe face mesh [9]

Some parameters were derived for all collected features and are summarised in the following table.

Table 2.1: Features extracted from signals

Features	maximum
	minimum
	maximum index
	minimum index
	mean
	median
	range(maximum-minimum)
	standard deviation
	variance
	kurtosis
	skewness
	25 percentile
	75 percentile
	range 75p-25p
	range 75p-50p
	range 50p-25p

2.3.1 Features defining the range of motion

Distance between lower and upper lip: The calculation of the mouth's opening and closing during linguistic and non-linguistic production exercises is one of the most essential aspects of the child's language evaluation. Two landmarks were acquired on the lower and upper lips in order to do these measurements, and their normalised relative distance was measured during the entirety of the task.

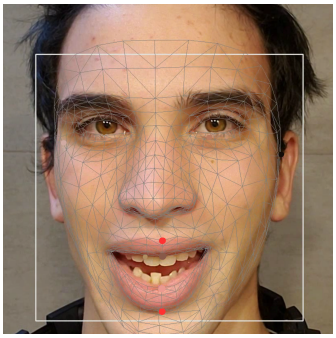


Figure 2.23: Identification of lip markers. In red are the markers used for opening and closing mouth measurements

[9]

Jaw movements: Jaw movements were recorded in relation to a point of reference on the tip of the nose. The average position of the three markers on each side was computed, three markers from the left jaw and three from the right jaw [21].

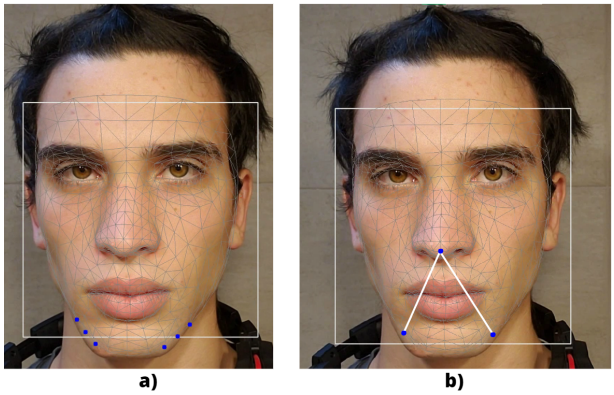


Figure 2.24: Tracking of jaw movements. **a)** markers on each side of jaw. **b)** distances between the tip of the nose and the mean of the three markers to measure jaw movements

[9]

Mouth width: The distance between the two labial commissures was used to assess the width of the mouth.

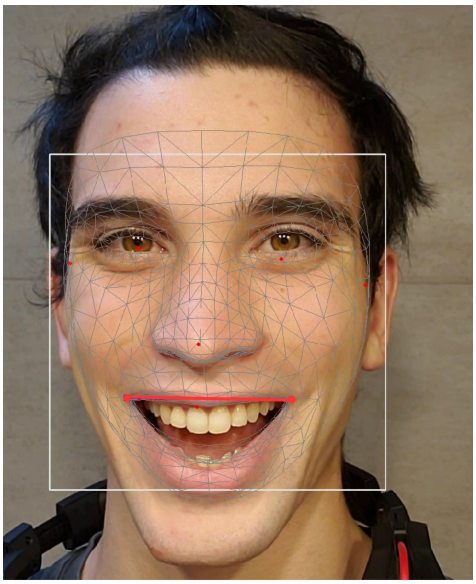


Figure 2.25: Measure of the mouth width
[9]

2.3.2 Features to evaluate speed of motion

These collected parameters are excellent for determining how quickly the patient can repeat syllables. In fact, they play a crucial role in oral diadochokinetic (DDK) tasks. The first derivative to the mouth opening and width were accounted for measuring speed. This method allowed for the assessment of the velocity of lip movements along the vertical and horizontal axes. The literature has suggested that the maximum and minimum values of these characteristics serve as differentiating factors for individuals with language difficulties caused by improper syllable articulation.

2.3.3 Features of symmetry

Measurements were obtained symmetrically on the face in addition to the semi-areas of the mouth to look for any asymmetries in the face, (see. **Fig 2.27**). Calculations were made to determine the separation between the lateral corner of the eye and the labial commissure (left/right). Additionally measured was the separation of the upper and lower eyelids (left/right).

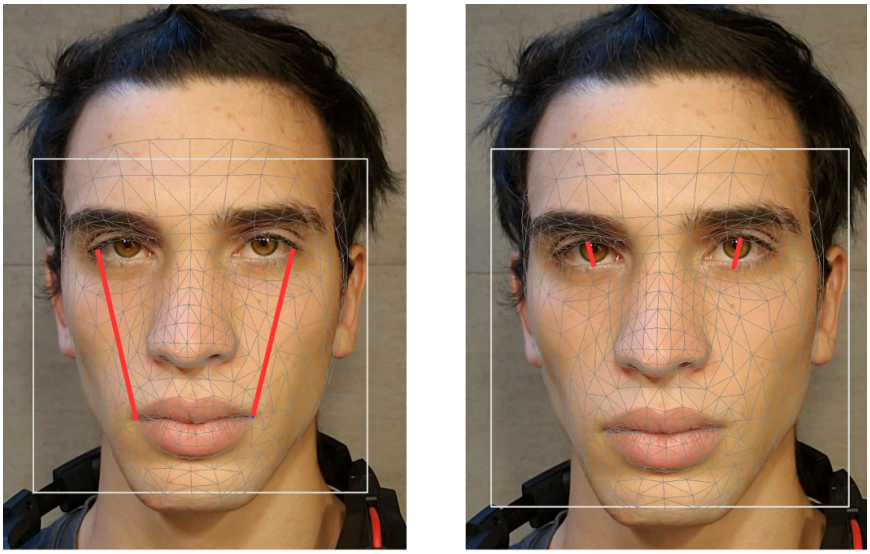


Figure 2.26: Face symmetries. **a)** Distances between Lateral angle of the eye and lateral commissure of the mouth (left/right); **b)**Distances between the eyelids (left/right)

[9]

Mouth area: The left and right halves of the mouth were taken into consideration while calculating the mouth area.The total area was calculated as the sum of the areas of the two triangles, representing the left and right areas of the mouth.

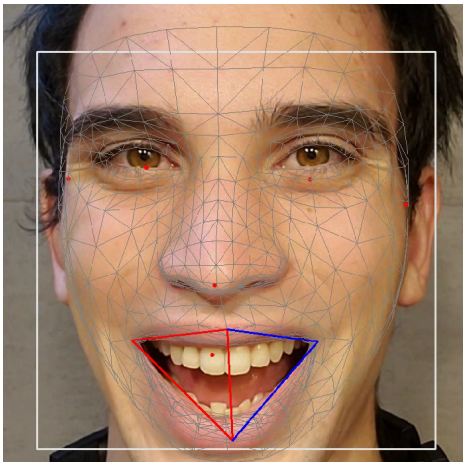


Figure 2.27: Area of the mouth. in red is shown the area of the right mouth and in blue the area of the left mouth

[9]

2.3.4 Features of shape and geometry

When performing the task, the roundness of the lips was used as a shape indicator. The computation was done by estimating the eccentricity of an ellipse with the mouth opening and its width as axis.

$$e_i = \begin{cases} \sqrt{1 - \frac{W_i^2}{O_i^2}}, & W_i < O_i \\ \sqrt{1 - \frac{O_i^2}{W_i^2}}, & W_i > O_i \end{cases} \tag{2.21}$$

Significant features included the mean value and the difference between the minimum and maximum value of eccentricity [22].

2.4 Signal filtering

All collected measures were filtered using an 8-pole low-pass Butterworth filter [23] with a cutoff frequency of 10 Hz to remove high-frequency noise caused by mesh fitting on the face.

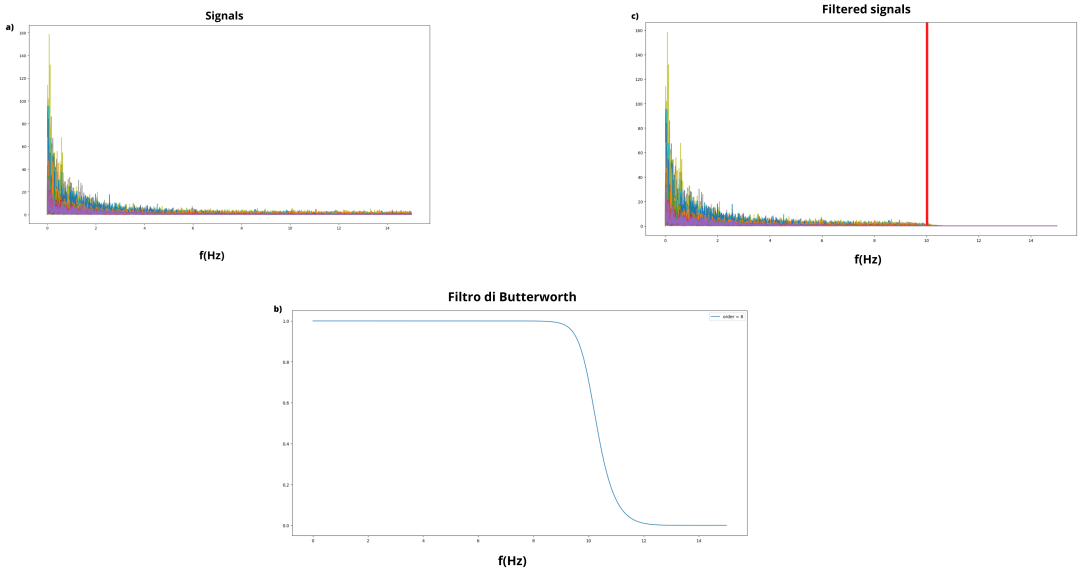


Figure 2.28: Signal filtering; **a)** Spectrum of original signals, **b)** Butterworth filter, **c)** Filtered signal spectrum

Chapter 3

Speech analysis

It was chosen to extract the audio signals from the recordings in order to evaluate their information content for a more precise and detailed analysis. In earlier studies, machine learning techniques were also employed to identify children with SLI or dyspraxia from voice recordings made during particular tasks [24][25]. The goal of the thesis work, at this stage, is to develop a computer programme that can distinguish between clinical children with speech impairments and normative children, using data collected in a non-laboratory setting. Three steps were carried out in order to conduct the classification under the working conditions mandated by the experimental setup:

- Preprocessing
- Feature extraction
- Feature selection

The features retrieved from the signals are both time and frequency dependent, with formants frequently used instead of frequencies to analyse speech. The sound's resonant or distinctive frequency, known as the formant, is the frequency value at which the amplitude reaches a peak. The human voice is made up of many formants because of the resonances of the Ear-Nose-Throat (ENT) cavities.

3.1 Preprocesing

3.1.1 Speaker diarisation

The recordings were captured by allowing the children to engage in the tasks as naturally as possible, both in terms of language use and movement. Many times, in order to get the children's attention during these experimental situations, the operator had to step in, corrupting the acquired audio signal. Speaker diarization seeks to locate and then remove any segments of the signal where the operator's voice is detectable, this was achieved by using the Python *pyannote* toolkit [26] [27] [28]. The PyTorch machine learning framework, which offers a collection of trainable end-to-end neural building blocks, is a prerequisite for the diarization toolkit. There are three distinct phases in the speaker-diarization pipeline:

- Speaker segmentation through a sliding window
- Speaker embedding
- Agglomerative clustering

Speaker segmentation: With a 5s window and a 500ms step, segmentation is carried out through an end-to-end neural network (EEND) (see. **Fig 3.1**). A sequence of audio signal measurements are sent into the model as input, $X = \Big(x_t \in \mathbb{R}^F | t = 1, .., T\Big)$. Each observation has a label associated to it $Y = (y_t | t ..., T)$, with $y_t = [y_{t,c} \in \{0,1\} | c = 1, ..., C]$ where c denotes the particular speaker. When the label is set to 1 in both cases for two speakers, we are in an overlap condition. The most likely label sequence is then estimated by the model as:

$$\hat{\mathrm{Y}} = \arg\max_{Y \in \mathbf{Y}} P(Y|X) \tag{3.1}$$

Where $\mathbf{Y}$ are the sequence of all possible speaker labels. The model is actually unable to distinguish between two sequences with the same labels but different speaker orders (*labels ambiguity*) (see. **Fig.3.1**). The ambiguity was resolved by the introduction of two permutation-free loss functions. All combinations of the ground-truth speaker labels are taken into account using the loss function permutation-invariant-training (**PIT**). The Deep Clustering loss function (**DPCL**) is used to boost the activation of networks that can distinguish between speakers. A bidirectional long short-term memory (BLSTM) building block is present in the EEND model. The BLSTM is a special variety of recurrent neural network where the input data sequence (in this application, our audio stream) flows both backward and forward. Because each element in the input sequence provides information about both the past and the future, this network is frequently employed in natural

language analysis since both are occasionally essential for comprehending the relationship between words inside a sentence (see. **Fig 3.2**).

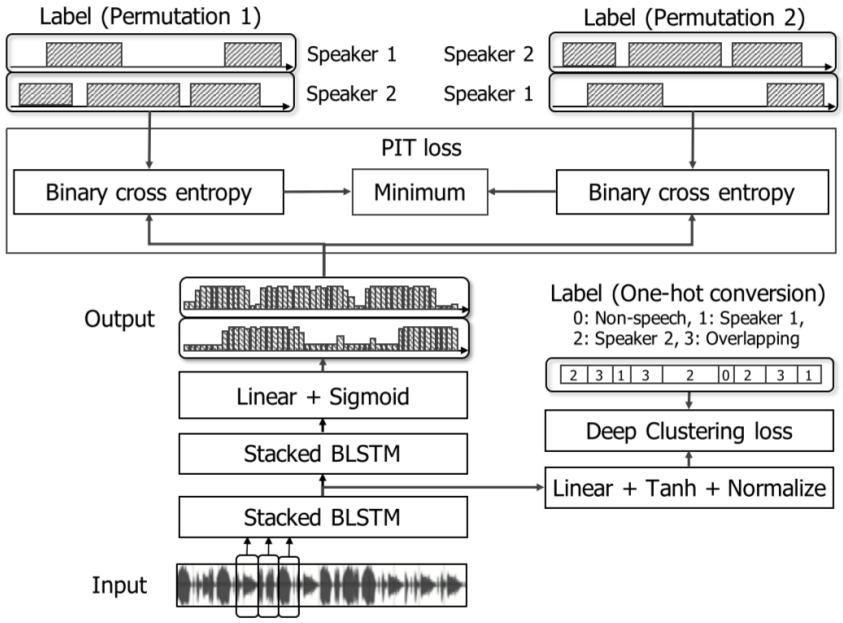


Figure 3.1: Neural end-to-end speaker diarisation model [29]

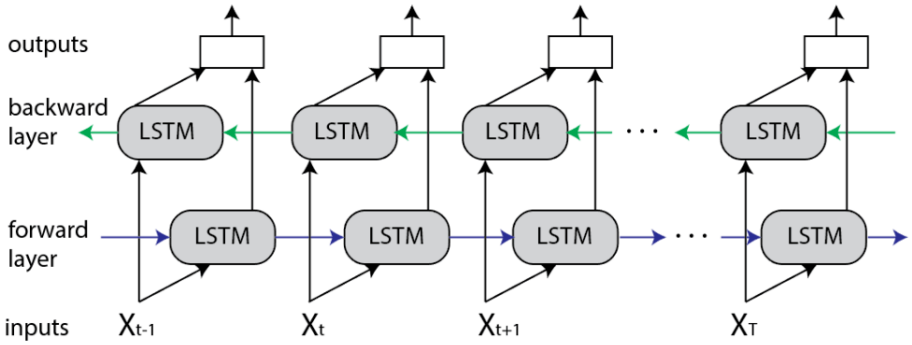


Figure 3.2: BLSTM layer of the EEND neural network; It is a combination of two long short-term memory (LSTM) layers. [30] [31]

The segmentation model was trained permutatively with a maximum of three speakers, which is why the same speaker may be given a different index in different

signal segments. The output of the segmentation model corresponds to a three-dimensional vector calculated every 16ms for each 5s window into which the signal is split and represents the likelihood that each of the three speakers is active at that moment (see. **Fig 3.3**).

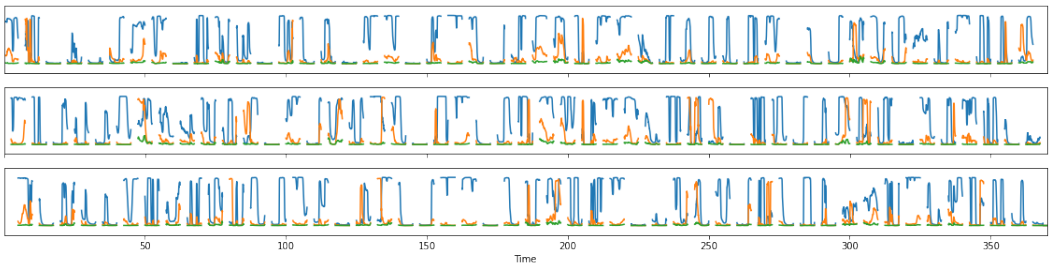


Figure 3.3: Speaker segmentation for each one of the 3 possible speakers. For ease of visualising, a 2.5s step has been chosen in the image, however the actual step is 500ms

[32]

The next binarization phase used $\theta \in [0,1]$ as a threshold, which is the first hyper-parameter of the model.

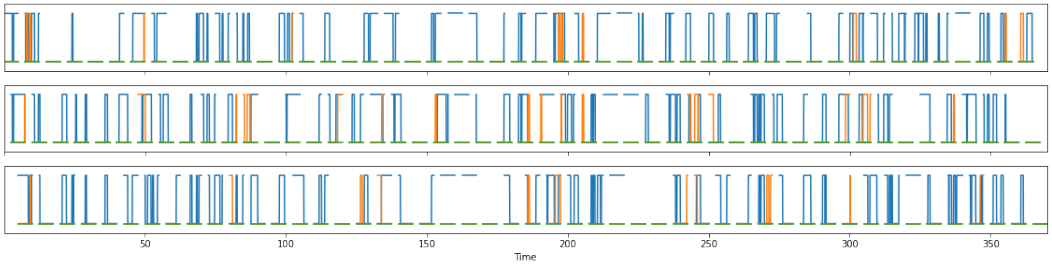


Figure 3.4: Binary speaker segmentation. The speaker whose probability surpassed the threshold (θ) was chosen for each window.

[32]

Speaker embedding: Every speaker embedding for each window is extracted, precisely one speaker embedding for each of the three potential speakers.

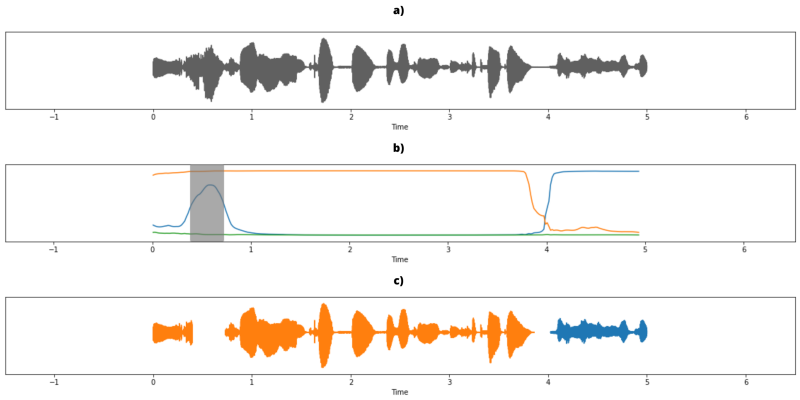


Figure 3.5: Speaker embedding; **a)** signal inside the 5s window. **b)** The speaker segmentation model detects two active speakers (orange and blue). **c)** Speaker embedding of the two speakers in orange and blue
[28]

It is possible for speakers to occasionally partially intersect in the same window, as seen in the grey area in **Figure 3.5**. In these circumstances, the concatenation of the samples with no overlap is used to determine the embedding of speakers. This technique has the drawback that it is dependent on the segmentation model, which is a performance bottleneck.

Global agglomerative clustering: The final stage is to perform a clustering procedure, in which each speaker detected in the multiple windows is given a global index. Clustering is carried out using the *UPGMC* method, a centroid-based clustering algorithm. At this stage, a new hyper-parameter called δ is added. It works as the clustering process's stop condition. The final stage will combine the clusters to produce the diarization's result (see. **Fig 3.6**).

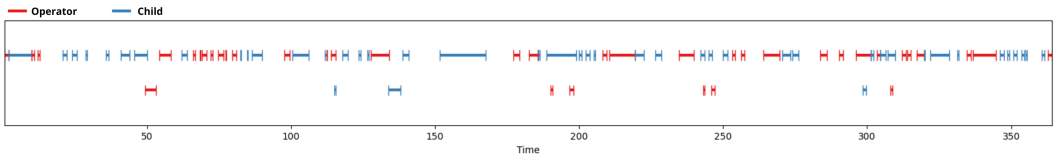


Figure 3.6: Diarization result
[32]

3.1.2 Segmentation and windowing

Using a Hamming window, the signals were then separated into 1s epochs with 50% overlap. This final step was carried out to allow us to assume stationarity for these time intervals while preserving the nature of the signal and therefore expanding, artificially, the number of observations in our dataset. To eliminate any potential DC signal impacts on the recordings, the average was subtracted for each signal.

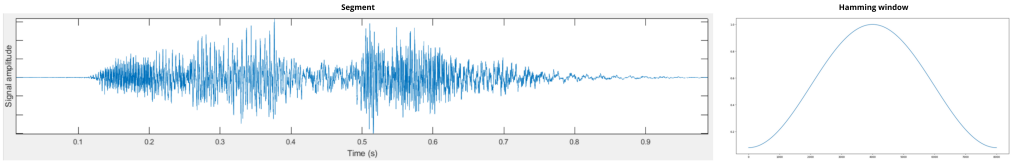


Figure 3.7: Signal segmented into 1s epochs via Hamming window

3.2 Feature extraction and feature selection

3.2.1 Feature extraction

The feature extraction procedure was conducted in accordance with earlier work of Mittapalle Kiran Reddy et al [25]. The feature extraction was carried out using the openSMILE toolkit developed for the INTERSPEECH 2016 Computational Paralinguistic Challenge [33]. The Challenge's goal was to assess the sincerity and native tongue of some speakers. This thesis project utilised the same feature set as the challenge, which consists of 6373 features that were calculated using some statistical functionals on low-level descriptors (LLD) and on ΔLLD. These features are based on 64 LLD that relate to parameters:

- Energetic
- Spectral
- Related to voice

Additional categories for features include **prosodic**, **spectral/cepstral**, or **sound quality-related**.

Energy related LLD	Sum of auditory spectrum (loudness)	Prosodic
	Sum of RASTA-style filtered auditory spectrum	Prosodic
	RMS energy, zero-crossing rate	Prosodic
Spectral LLD	RASTA-style auditory spectrum, bands 1–26 (0–8 kHz)	Spectral
	MFCC 1–14	Cepstral
	Spectral energy 250–650 Hz, 1 kHz–4 kHz	Spectral
	Spectral roll off point 0.25, 0.50, 0.75, 0.90	Spectral
	Spectral flux, centroid, entropy, slope	Spectral
	Psychoacoustic sharpness, harmonicity	Spectral
	Spectral variance, skewness, kurtosis	Spectral
Voicing related LLD	$F0$ (SHS and viterbi smoothing)	Prosodic
	Prob. of voice	Sound quality
	Log. HNR, Jitter (local, delta), Shimmer (local)	Sound quality

Table 3.1: The feature set ComParE 2016 uses 64 LLD for the description of acoustic signals
[34] [35] [36]

Functionals Applied to LLD and ΔLLD	Quartiles 1–3, 3 inter-quartile ranges	Percentiles
	1% Percentile ($\approx min$), 99% percentile ($\approx max$)	Percentiles
	Percentile range [1%,99%]	Percentiles
	Position of (min/max), range ($max - min$)	Temporal
	Arithmetic mean, root quadratic mean	Moments
	Contour centroid, flatness	Temporal
	Standard deviation, skewness, kurtosis	Moments
	Rel. duration LLD is above 25/50/75/90% range	Temporal
	Rel. duration LLD is rising	Temporal
	Rel. duration LLD has positive curvature	Moments
	Gain of linear prediction (LP), LP coefficients 1–5	Modulation
	Mean, max, min, SD of segment length	Temporal
Functionals applied to LLD only	Mean value of peaks	Peaks
	Mean value of peaks – arithmetic mean	Peaks
	Mean/SD of inter peak distances	Peaks
	Amplitude mean of peaks, of minima	Peaks
	Amplitude range of peaks	Peaks
	Mean/SD of rising/falling slopes	Peaks
	Linear regression slope, offset, quadratic error	Regression
	Quadratic regression a, b, offset, quadratic error	Regression
	Percentage of non-zero frames	Temporal

Table 3.2: Functionals applied to LLD. The third column shows the statistical functional's type
[34] [35]

The descriptors were selected from those that are most frequently used in speech and sound analysis as well as those that are used to extract information from music. For instance, speaker identification often involves the use of Mel Frequency Cepstral Coefficients (**MFCC**). These coefficients use a cepstral representation of

the audio signal. The cepstrum, is a representation of the calculation of the inverse Fourier transform (IFT) of the logarithm of the signal spectrum. When computing the cepstrum, the effects of tone excitation and voice formants are additive and thus separable.
Other computed features include:

- Loudness
- Energy
- Foundamental frequency
- Psychoacoustic sharpness
- Jitter
- Shimmer

3.2.2 Feature selection

The distribution of the features retrieved in the preceding stage was examined before feature selection to ensure that they had a normal distribution. The **Jarque-Bera** [37] test was utilised to assess the distribution of features. Its performance is comparable to that of the Shapiro-Wilk test, but it is more efficient when a large number of samples need to be studied [38] [24]. This test ensures that a data distribution's skewness and kurtosis match those of a normal distribution. The formula for the test is as follows:

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right) \tag{3.2}$$

where n is the number of observations, S represents the sample skewness, and K represents the sample kurtosis.

$$S = \frac{\hat{\mu_3}}{\hat{\sigma}^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{\frac{3}{2}}} \tag{3.3}$$

$$K = \frac{\hat{\mu_4}}{\hat{\sigma}^4} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{2}} \tag{3.4}$$

- $\hat{\sigma}$ is estimation of the variance
- $\hat{\mu_3}$ and $\hat{\mu_4}$ are the estimates of third and fourth central moments
- $\bar{x}$ is the mean

The Jarque-Bera test's results provided a list of 56 features that appeared to have a normal distribution ($p \leq 0.05$). Two tests were then used to carry out the selection of statistically significant features to differentiate between normative and clinical patients. For the normally distributed features, an independent samples t-test [39] was employed; for the other features, the Mann-Whitney U [40] non parametric statistical test was applied. Following the Jarque-Bera test, 56 features had a normal distribution and 6317 had a non-normal distribution. The 56 features with normal distribution underwent a Levene's test [41] to valute homogeneity of variance between the normative and clinical populations. Welch's t-test [42] used for 14 of the 56 features with non-homogeneous variance, whereas independent-samples t-tests were used for 42 of the 56 features for which the variance of the two populations was determined to be homogeneous.

t-test: the two samples independent t-test is performed for equal variance population :

$$t = \frac{\bar{X_1} - \bar{X_2}}{s_p \sqrt{\frac{2}{n}}} \tag{3.5}$$

Welch's t-test is performed for populations with non-homogeneous variance:

$$t = \frac{\bar{X_1} - \bar{X_2}}{\sqrt{s^2_{\bar{X_1}} + s^2_{\bar{X_2}}}} \tag{3.6}$$

with:

$$s_{\bar{X_i}} = \frac{s_i}{\sqrt{N_I}} \tag{3.7}$$

where s_i is the corrected sample standard deviation

Mann-Whitney U test: This test uses independent samples and is a non-parametric test based on U-statistic. $X_1, .., X_n$ are independent identically distributed samples from $\mathbf{X}$, $Y_1, .., Y_n$ are independent identically distributed samples from $\mathbf{Y}$. The definition of the Mann-Whitney U statistic is:

$$U = \sum_{i=1}^{n} \sum_{j=1}^{m} S\left(X_i, Y_j\right) \tag{3.8}$$

$$S = \begin{cases} 1, & X > Y \\ \frac{1}{2}, & X = Y \\ 0, & X < Y \end{cases} \tag{3.9}$$

During the selection process, 857 features out of the initial 6373 were dropped, leaving a total of 5516 features. A dimensionality reduction was done to strengthen the selection to shorten classifier training times, lower computational costs, and

prevent overfitting issues [24]. Particularly, two alternative dimensionality reduction techniques were tested:

- Neighbourhood components analysis (**NCA**)
- Principal component analysis (**PCA**)

Neighbourhood components analysis [43]: With the use of linear transformations, this method derives a distance value in order to improve the performance of a leave-one-out (LOO) classifier. By establishing a differentiable objective function for the matrix itself and using the gradient descent approach, it is possible to identify the matrix encoding the linear transform. With this method, after the linear transformation inside the LOO classifier, we will take into account the complete dataset rather than just the k-neighbors at each location. The entire dataset is therefore redefined using a softmax function of the squared Euclidean distance.

$$p_{ij} = \begin{cases} \frac{e^{-||Ax_i - Ax_j||^2}}{\sum_{k \neq i} e^{-||Ax_i - Ax_j||^2}}, & j \neq i \\ 0, & j = i \end{cases} \tag{3.10}$$

where p_{ij} is the probability of classifying neighbour j of point i. The probability of classifying data i properly is equal to the probability of identifying each of its neighbours' points with the same class C_i.
The objective function is:

$$f(A) = \sum_i \sum_{j \in C_i} p_{ij} = \sum_i p_i \tag{3.11}$$

Where A is the linear transform matrix, which are obtained by using an iterative solver such conjugate gradient descent.
Principal component analysis [44]: This is used to compress huge datasets while maintaining the information's integrity. The component that maximises the variance of the data is the first principal component. The direction that maximises the variance of the projected data and is orthogonal to the first $(i-1)$ principal components is the i_{th} principal component. We consider a matrix X with dimension $n \times p$, where n are the repetitions and p are the columns representing the various features. A set of p-dimensional vectors of weights of size l define the transformation; $w_{(k)} = (w_1, ..., w_p)_{(k)}$. Each row vector is transformed into a new vector that contains the principal component scores $t_{(i)} = (t_1, ..., t_l)_{(i)}$, wihch is: $t_{k(i)} = x_{(i)} \cdot w_{(k)}$. Hence, in order to reduce dimensionality, l is typically chosen to be strictly less than p. Due to dimensionality reduction, 2921 features were ultimately chosen, and they were enough to explain for 99% of the variance.

3.3 Classification

Following feature selection, three distinct machine learning-based classifiers were evaluated and compared:

- K-nearest neighbors (K-NN)
- Support vector machine (SVM)
- Feedforward Neural Network (FFNN)

3.3.1 K-NN [45]

Both classifications and regressions can be done using this approach. In the latter situation, classification is done based on its neighbours. The sole hyperparameter that needs to be determined for this approach is the number of neighbours to compare our object in order to determine its class. The most prevalent class among its neighbours is given to the sample that has to be classified.

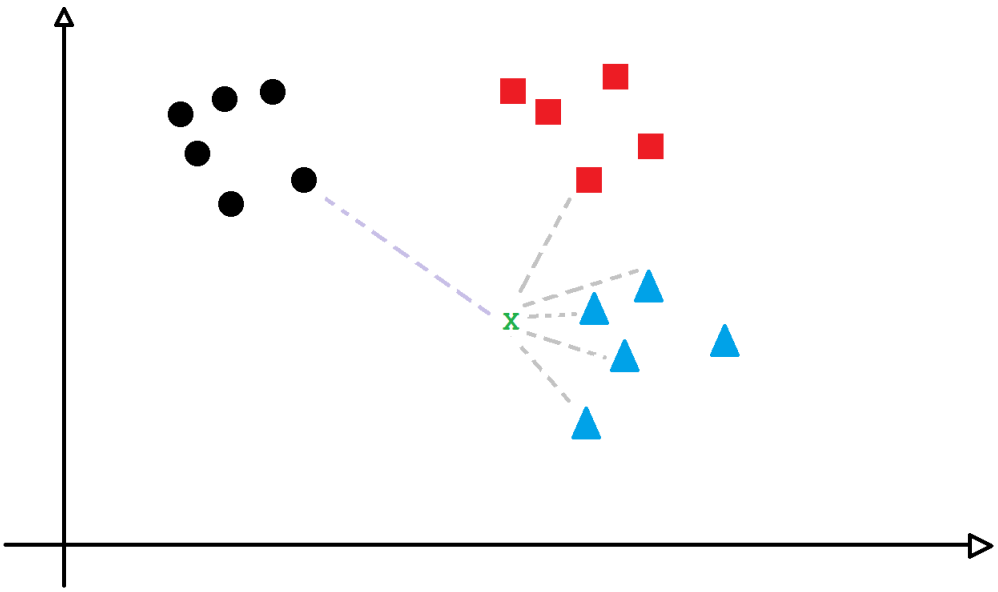


Figure 3.8: Application example of the KNN algorithm [45]

3.3.2 Support vector machine (SVM) [46]

The SVM creates a line or hyperplane whose task is to divide the data into classes. In the first approximation, the SVM tries to create a dividing line between the two classes. The SVM classification algorithm is based on the idea of finding the best line or hyperplane that allows the separation between classes, so as to have as general a classifier as possible. To find the best line, the distance of the nearest data points to the line or hyperplane for both classes is calculated, these points are called **support vectors** while the distance is called the **margin**. The hyperplane or line for which the margin is maximum will be the one chosen for classification.

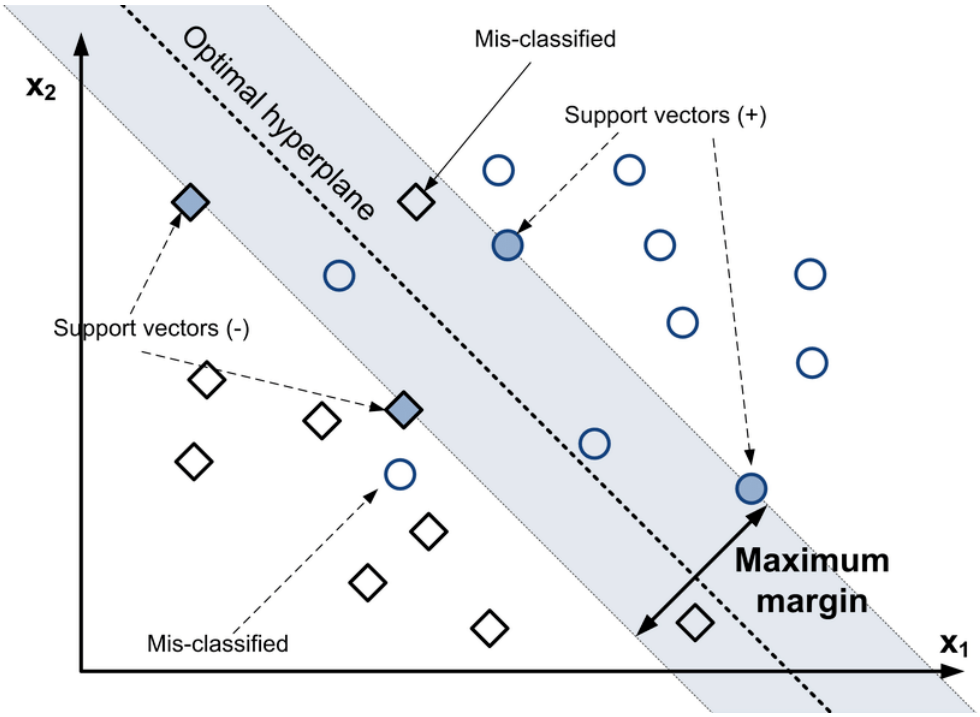


Figure 3.9: SVM linear classifier
[46]

3.3.3 Feedforward-neural-network (FFNN) [47]

These networks are distinct from recurrent neural networks in that there are no circular connections between the nodes. In this network, data moves from the input neurons to the output neurons via the hidden layer. Here, an input layer, an output layer, and three hidden layers made up the FFNN. There are 16 neurons in the input layer, while there are 8, 4, and 2 neurons in each of the 3 hidden layers (see. **Fig 3.10**) .

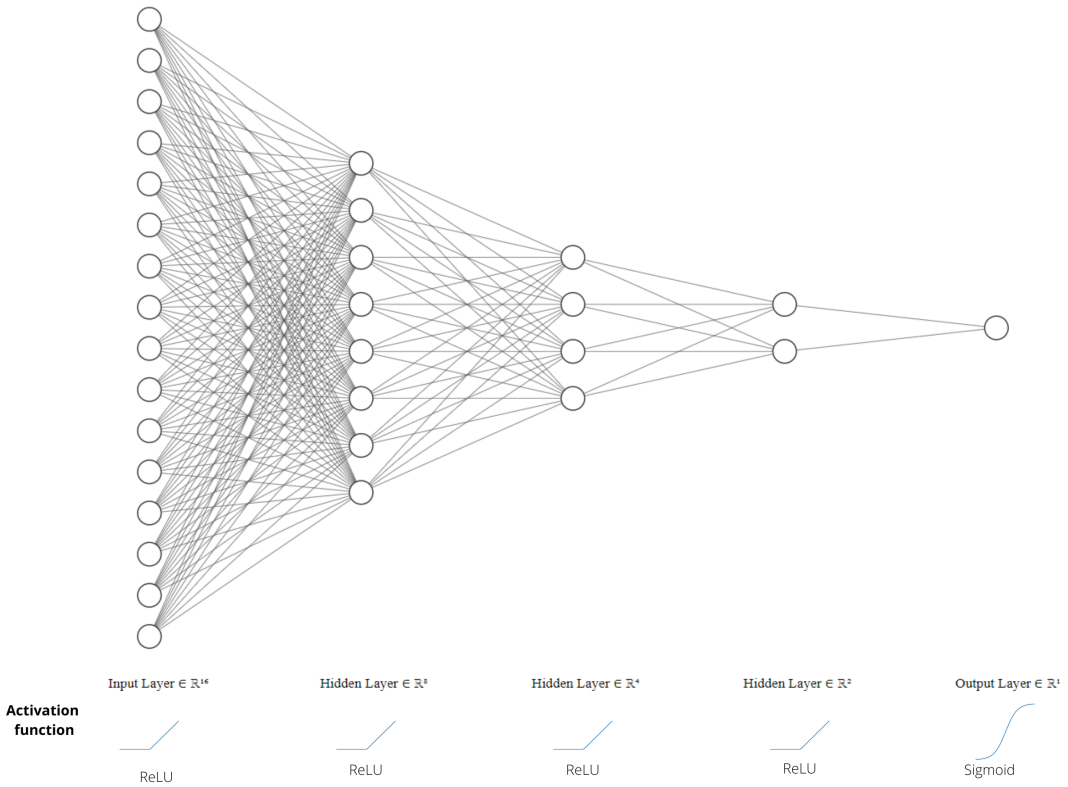


Figure 3.10: FFNN layers with the relatives activation functions

To prevent overfitting, dropout levels were implemented. This method involves randomly removing some neurons during training such that they do not affect the firing of downstream neurons and their weights are not updated later. This prevents neurons from becoming specialised on training data by forcing the network to generalise. The neurons that are left over must make up for the loss of the ones that are turned off by periodically adjusting their weights in order to make the necessary prediction, leading to less precise weights [48]. It was chosen at random, with a 40% probability, which nodes would be removed. A **ReLU** activation function was applied for the input layer and the three hidden layers, which is a

non-linear function with a very low computing cost [49]. Instead, the sigmoid, a better activation function for binary classification tasks, was selected for the output layer (see. **Fig 3.10**) [50]. All activation functions share these two characteristics:

- Output $\in$ [0,1]
- Output ≈ 1 with enough stimulation (threshold effect)

The optimisation function is another significant parameter in the neural network. The optimization function is the one that enables me to assess each node's error at each iteration, and the relative weights are modified in accordance with the error. ADAM's function, an extension of root mean square propagation (RMSProp) that considers the first and second moments of the gradient, was chosen as the optimization function. The SGD gradient descent and RMSprop itself are frequently rejected in favour of the ADAM function because it is less sensitive to noisy gradients, consumes less memory, and works well with large datasets [51].

Chapter 4

State of the art

4.1 Face detection

In order to determine the state of the art in face identification techniques, several algorithms developed throughout time were evaluated and applied to numerous widely used public datasets, including:

- Face Detection Data Set and Benchmark (FDDB) [52]: It has 5171 faces in 2845 images
- Annotated Faces in the Wild [53]: 205 images with 473 faces
- WIDER Face (Easy) [54]
- PASCAL Face [55] 1335 faces in 851 images

The Cheng Chi et al. algorithm is the one that currently outperforms the others on average in these datasets [56]. The technique developed by Cheng Chi is a single-shot face detector (SSD) called the Selective Refinement Network (SRN). When compared to other face identification algorithms, this algorithm, which is of the anchor-based type, introduces two extra classification and regression steps. Two modules, the Selective Two-step Classification (STC) module and the Selective Two-step Regression (STR) module, respectively, carry out these steps [56]. The first module's goal is to remove as many negative anchors from the lower description layers, and the second module's aim is to change the anchors' sizes and positions. To identify the most unusual facial poses, an additional model known as Receptive Field Enhancement (RFE) was introduced to the end of the network.

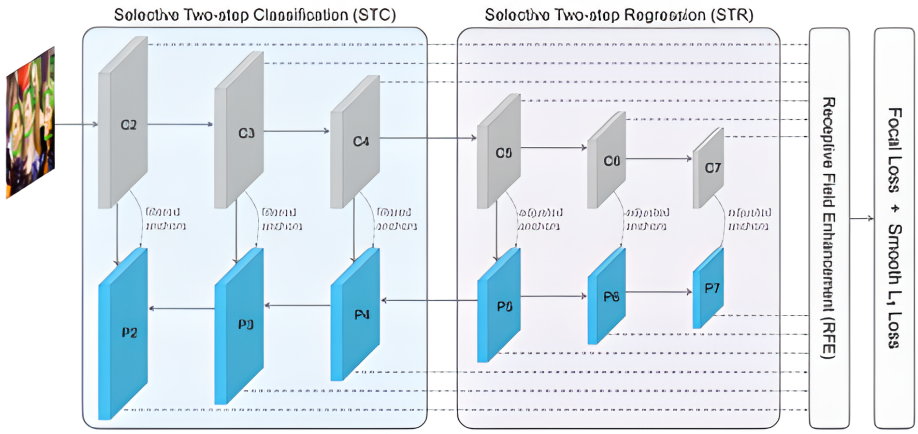


Figure 4.1: Network structure of SRN. It is made up of RFB, STR, and STC. In order to narrow the search space for the second-step classifier, STC employs the first-step classifier to filter the negative anchors from low level detection layers. For better initialization of the second-step regressor, STR uses the first-step regressor to alter the positions and sizes of anchors from high level detection layers. To better record faces in unusual poses, RFE offers more varied receptive fields. [56]

The ResNet-50 [57] with a 6-level pyramid structure makes up the SRN network. In **Figure 4.1** the blocks C_n with $n \in (2,3,4,5)$, are the features maps extracted, after $C5$, $C6$ and $C7$ are obtained by two down-sample 3 X 3 convolution layers. P_n with $n \in (2,3,4,5)$, are extracted from lateral connection, after $P5$,$P6$ and $P7$ are down-sampled by two 3 X 3 convolution layers. On the FDDB dataset, on the other hand, another model based on a neural network and belonging to the anchors-based performed slightly better. The model proposed by Jian Li et al [58], called Dual Shot Face Detector (DSFD), introduces the Feature Enhancement Module (FEM) to improve the characteristics' robustness and discriminability. The following neural network employs a Progressive Anchor Loss (PAL) for each level and each shot, where the anchor in the first shot gets bigger than in the second shot. In order to better align anchors and target faces, an Improved Anchor Matching (IAM) is carried out, which combines an anchor partition technique with anchor-based data argumentation.

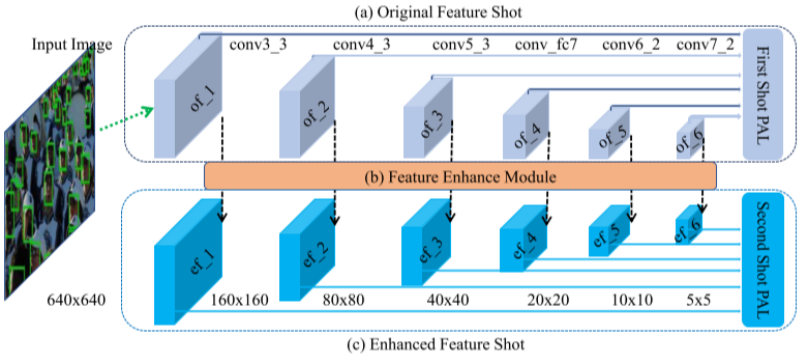


Figure 4.2: DSFD architecture; **b)** is the Feature Enhance Module, it is applied on the VGG/ResNet architecture and produce enhanced features **c)** from the starting features **a)**

[58]

These two face detection models, which are both based on ResNet technology, are the state-of-the-art. For instance, they obtained accuracy scores of 0.991 for DSFD and 0.988 for SRN on the FDDB dataset. The best methods for lowering false positive detection rates and raising precision are those mentioned above. The main drawback of these strategies is the amount of time and hardware resources needed to execute the models, which are unusable without GPU accelerators and require significantly more resources for the DSFD model.

4.2 Marker-less oral-motor feature extraction

As in the experiments by Gonzalo D. Sad et al., recent investigations have used markerless technology to derive suggestive features, particularly for diseases like ALS (Amyotrophic Lateral Sclerosis) or other primarily motor disorders [9][59]. In the field of rehabilitation, using increasingly adaptable technologies that can extract data with settings and conditions far removed from the laboratory has become a prerequisite. A CANDIDE-type model was employed in the specific instance of the investigations by Gonzalo D. Sad et al. on the ALS to identify all the facial asymmetries that are typical of the condition.

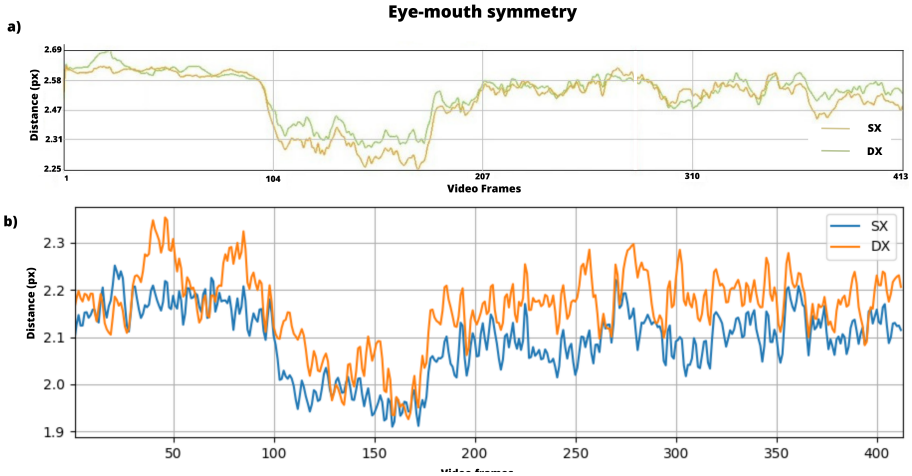


Figure 4.3: A comparison between the outcomes on the same video produced by the developed computer programme and the asymmetry detecting software created by Gonzalo D. Sad et al. for ALS patients. **a)** Eye-mouth distance measured with Gonzalo D. Sad et al. software. **b)** Eye-mouth distance measured with our programme

[59]

It can be seen that the produced programme has equivalent performance when compared to the software developed in the works of Gonzalo D. Sad et al. for the detection of facial asymmetries. Due to mesh adaptation, the feature extraction approach developed in this work is in this instance more sensitive to noise. However, because it does not require the mesh on the face to be adjusted and modified before the start of the acquisition, the MediaPipe method is simpler for medical staff to utilise, especially in the absence of a dedicated interface. It was preferable to switch to a fully automated approach because this condition for the type of patients on whom the research was conducted was limited because, particularly for clinical patients, it was challenging to obtain even a few frame instants without movement to allow manual adaptation of the mesh on the face. In recent years, a number of studies have employed markerless technology, particularly for tracking jaw and lip movements in experimental lab settings, frequently employing the Dlib algorithm to identify facial markers, as in the experiments by Andrea Bandini et al [21] [22].

4.3 Natural language processing

To date, technologies based on deep learning have not been widely used in the analysis of natural language to perform assessments of children in the developmental age. This is due to the fact that specifying the features to be assessed for classification is a necessary step to achieve a good classification. Machine learning techniques have already been used to identify children with SLI, as in the studies of Mittapalle Kiran Reddy and Yogesh Sharma, two of the pioneering studies in the field [25] [24]. In order to create the models for both works, the features that were deemed to be the most important were extracted. In the first case, in particular, the glottal-type features, the MFCC features, and those originating from the openSMILE software were compared in order to evaluate their classification effectiveness. Contrarily, the classification in the second study is purely based on pitch-related characteristics. These investigations introduce the glottal features that are also connected to a child's motor skills, the lack of which results in an irregular vocal vibration that may be identified by the glottal waveform. Recovering the glottal excitement is the aim of glottal inverse filtering (GIF). The GIF filter recognised the output signal, or speech pressure waveform, from the vocal input [60]. The MATLAB APARAT toolbox is one of the most popular methods for glottal parameter extraction [61].

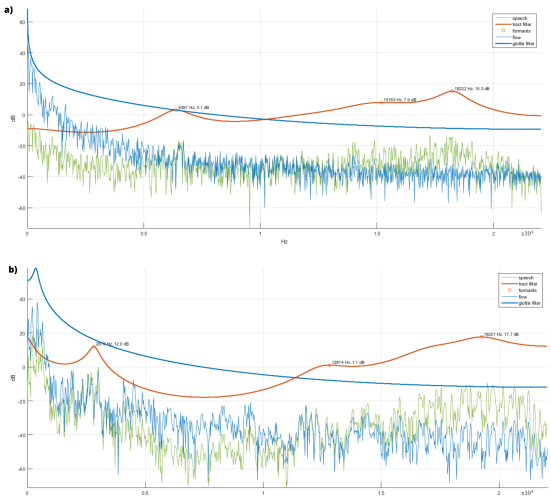


Figure 4.4: Aparat software output spectrum for audio signal analysis; **a)** Clinical child, **b)** Normative child

Figure 4.4 shows the output of the Aparat software for a clinical and a normative child. The software displays the vocal tract filter's formants, the GIF filter', the speech signal spectrum, and the glottal flow spectrum. The spectra of the glottal signal was acquired using the quasi-closed phase method (QCP) [62] as the GIF algorithm, after the application of a low-pass-filter with 60Hz cut-off frequency. Because it requires the guidance of an experienced operator and the Aparat extraction tool is only accessible with a MATLAB licence, glottal feature extraction was not used in this thesis study. Studies by Mittapalle et al. further demonstrate that, in comparison to openSmile characteristics, the use of glottal features does not now provide any discernible benefits to classification performance. There have already been more recent studies that examined the use of deep learning techniques for categorising specific developmental language disorders, such as the studies by Kanimozhiselvi et al. [63] that employ a convolutional neural network (CNN) to perform a multi-class classification among 4 different language disorders by creating software that is also suitable for mobile devices. The benefit of deep learning in this case is that it eliminates the need for a specialist operator to perform the feature extraction, and once trained, the model can also be used on devices such mobile phones. However, because the issue of language problems, as already indicated in chapter 1, is a multi-factorial issue, it is frequently helpful for medical staff to be able to check the features used for the classification.

Chapter 5

Results

The outcomes of audio and video analysis will be discussed in this chapter. The various approaches will be compared, which will determine the final decisions made in the programme given to the Foundation.

5.1 Analysis of oral-motor features

5.1.1 Face-identification

We examined the various face detection algorithms (described in Chapter 2) to determine which one would work best with the oral-motor feature extraction software. The computation time and measurement precision were taken into consideration when making the comparison. A random selection of 5 clinical and 5 normative patients was made. The area of the rectangle defining the face and its centre was determined for each frame. The Dlib toolkit creates boxes, on which the area of the face is calculated, and these boxes have fixed sizes. Because of this, the pertinent Bland-Altamn diagrams do not have a uniform distribution of points (see. **Fig 5.1**). For the same reason, the amount of outliers produced by the Dlib toolkit are frequently overlapped in boxplot diagrams (see. **Fig 5.2**).

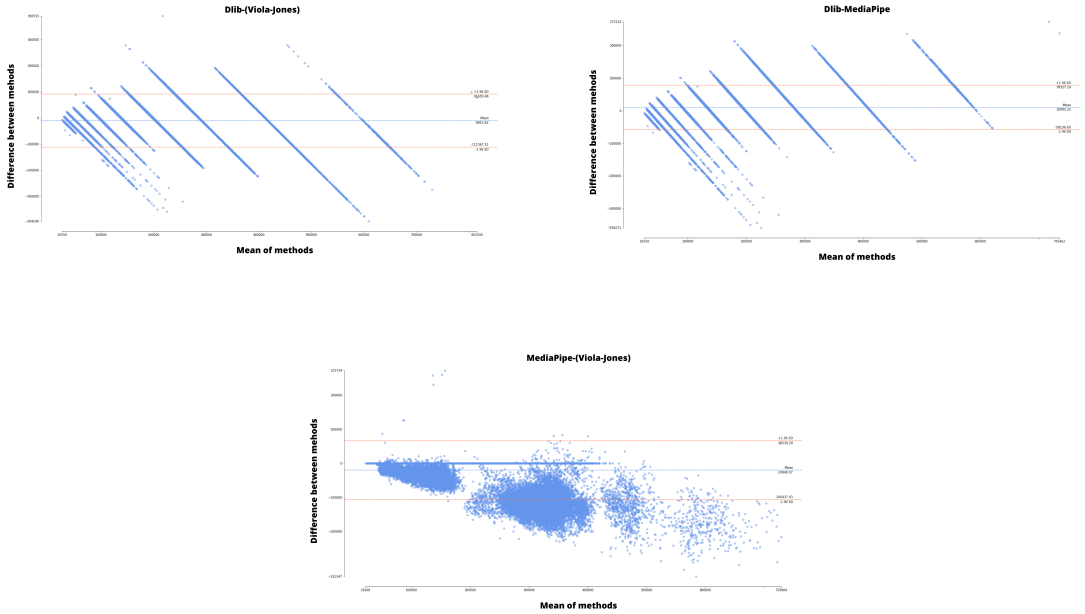


Figure 5.1: The degree of agreement between the three distinct measures of face area was assessed using Bland-Altman graphs.

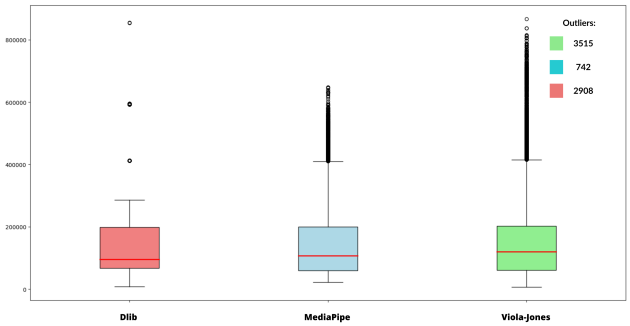


Figure 5.2: The boxplots compare the areas created using the three algorithms while also numerically expressing the number of outliers for each one. Visually, the Dlib algorithm has fewer outliers than the other two since the area can only take on a certain range of values, and many of the outliers are overlapping.

	Dlib	**MediaPipe**	**Viola-Jones**
Dlib	88.47%	52.71%	57.60%
MediaPipe	52.71%	97%	48%
Viola-Jones	57.60%	48%	61.68%

Table 5.1: The overlap of the face rectangles is used in the table to compare the face detection methods in pairs. The main diagonal displays the number of frames in which each algorithm successfully recognises a face. The total of frames considered for the study is 171202

A comparison of the three face detection methods is presented in **Table 5.1**. The number of frames in which the rectangles of the two methods can be regarded as superimposable can be determined by crossing the table's rows and columns. The main diagonal shows how many frames the algorithms were able to recognise a face. $c = (x, y)$ and $c1 = (x1, y1)$ are defined as the centre of the box produced by algorithm Q and algorithm $Q1$, respectively. If the area of the rectangle formed by $c1 \pm 80$ contained c, the two boxes were regarded as being overlapped.

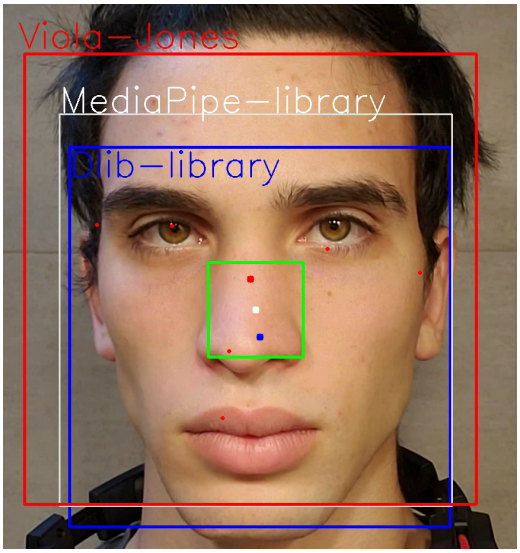


Figure 5.3: Results of face detection algorithms on a single frame. For the boxes to be deemed overlapping, the rectangles' centres must be inside the ROI in green.

Additionally, the processing speed of a 16s video was compared to determine which of the three algorithms was the quickest. The MediaPipe algorithm was

Time	40.36s	Dlib
	8.07s	MediaPipe
	23.43s	Viola-Jones

Table 5.2: Time to process a 16s video

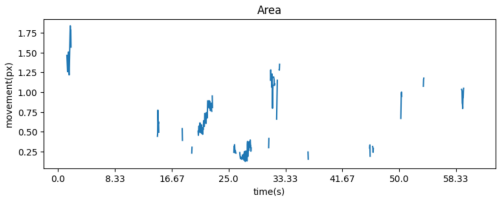
selected to carry out face detection based on the analysis that was conducted. This is due to the fact that it seems to be the optimal trade-off between processing efficiency and precision in identifying the face for each frame.

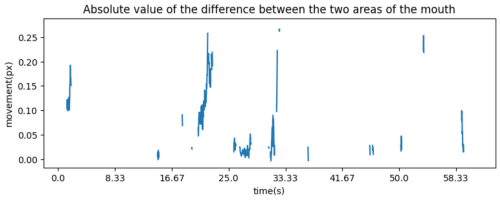
5.1.2 Face mesh assessment

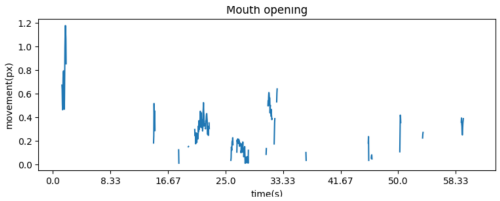
Healthcare practitioners can already make an early assessment of the child's linguistic abilities and executive function development from the mesh fitting on the face, which enabled the extraction of various clinical features (described in Chapter 2). This section compares the results produced by the programme in terms of processing speed and acquisition accuracy, using both the CANDIDE and MediaPipe models for a clinical and normative child. To lose the least amount of data and have a more accurate algorithm, it will also assess how effectively the two models can monitor head position. The model's sensitivity to noise, particularly the noise produced by the mesh's fit to the face itself, is another crucial factor to consider.

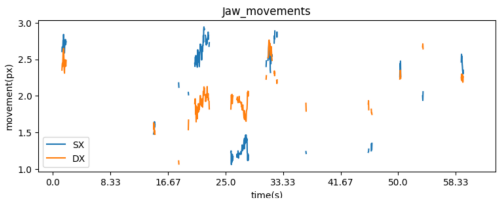
Clinical child

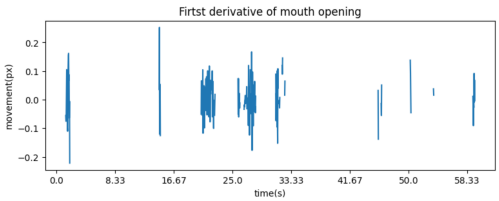
CANDIDE



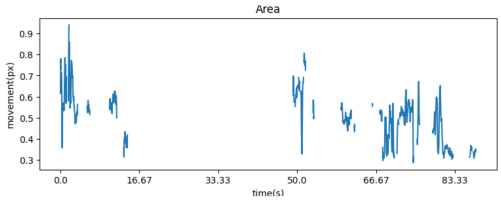


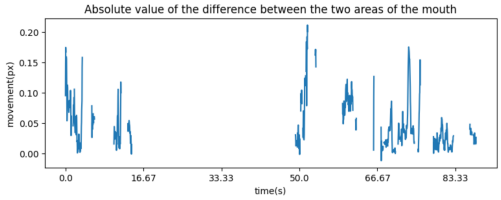


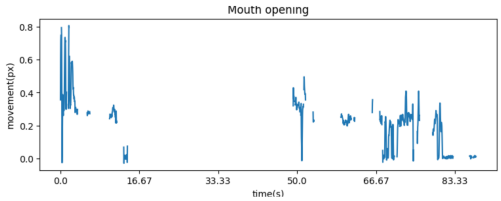


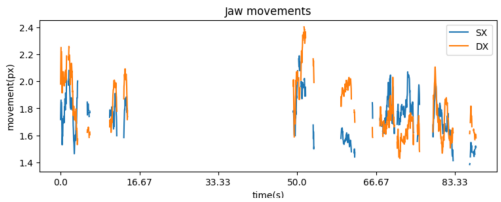


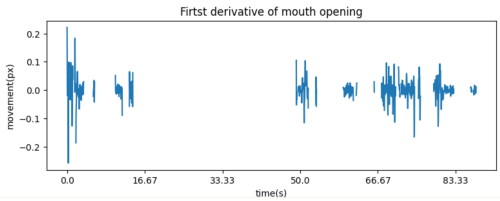
MediaPipe











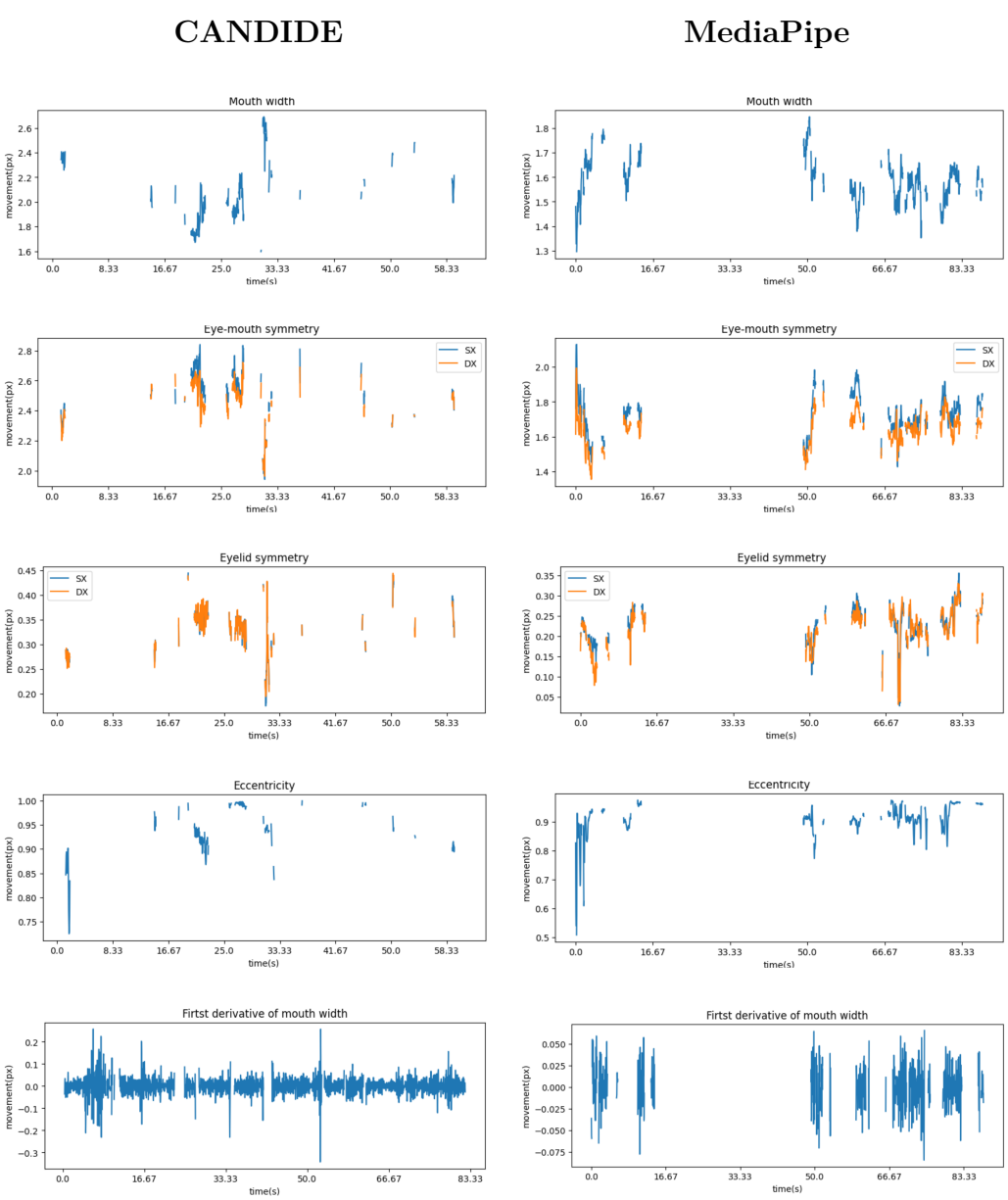


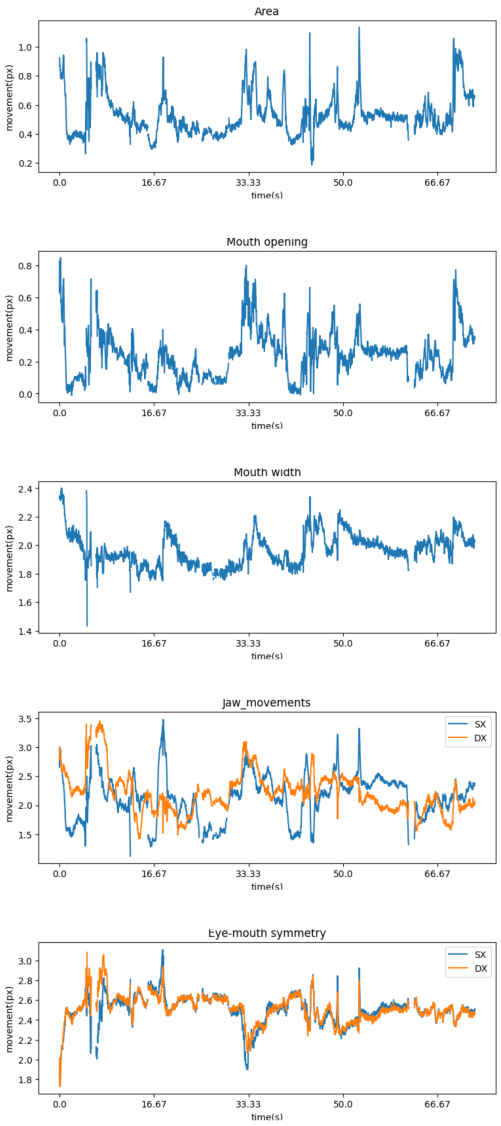
Figure 5.4: Oral-motor features extracted from CANDIDE model and MediaPipe model from the same video of clinical child

CANDIDE	350.67s
MediaPipe	93.05s

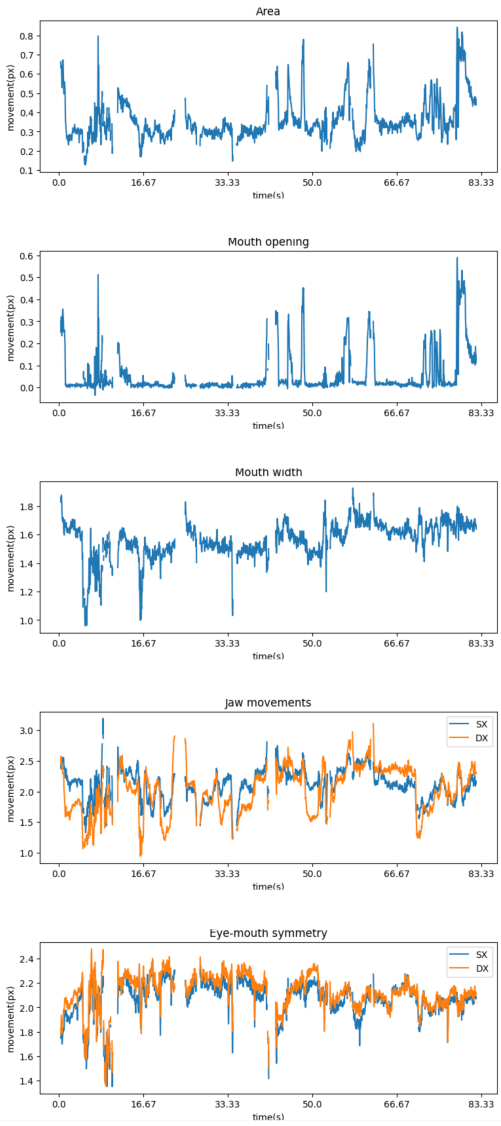
Table 5.3: Time to process the same videos for a clinical child

Normative child

CANDIDE



MediaPipe



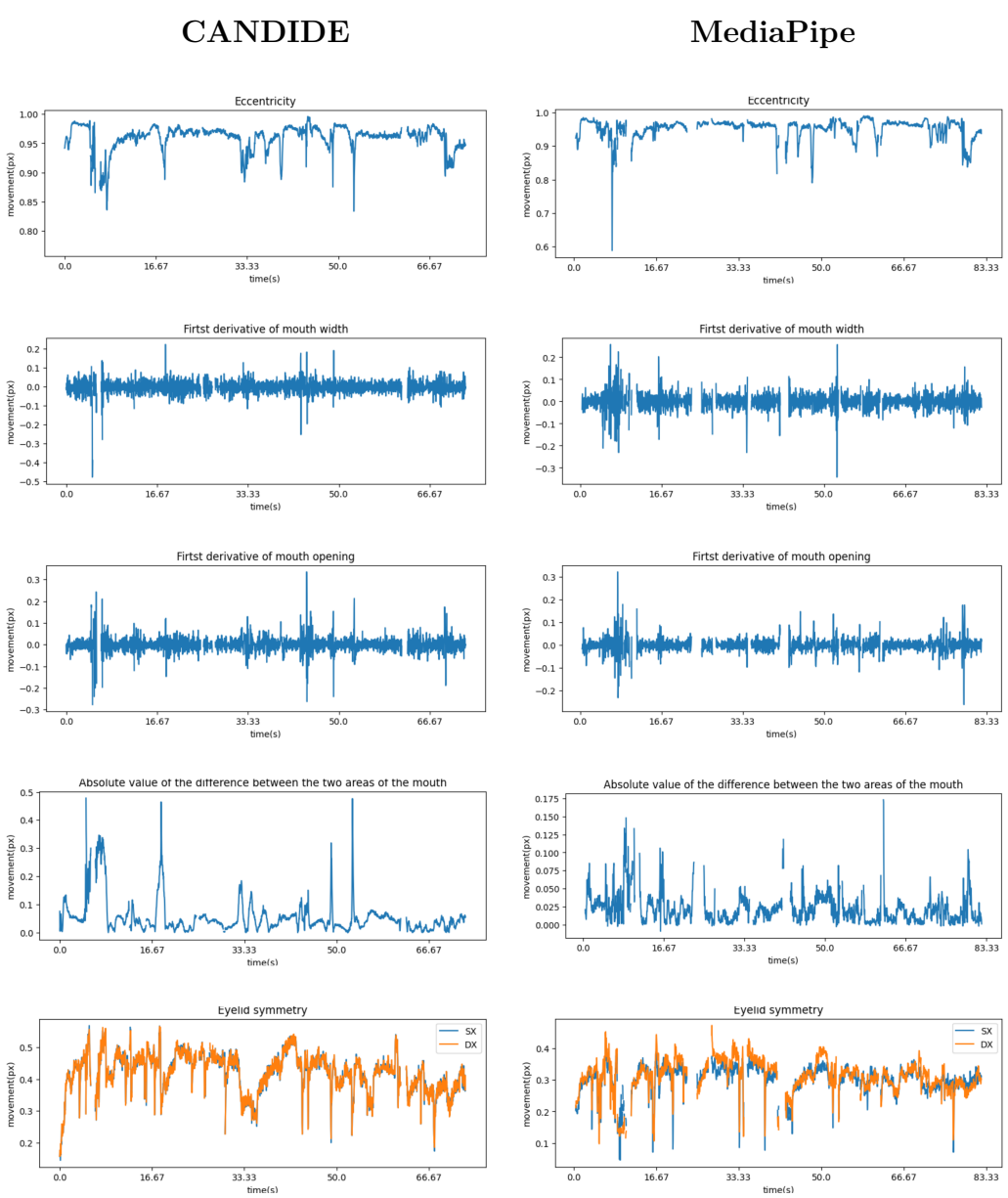


Figure 5.5: Oral-motor features extracted from CANDIDE model and MediaPipe model from the same video of normative child

CANDIDE	659.73s
MediaPipe	144.70s

Table 5.4: Time to process the same videos for a normative child

When the algorithm identifies that a child is not looking directly at the camera, it will not acquire signals for brief periods of time. As you can see, the clinical children show this condition more frequently. This is due to the fact that children with speech disorders typically struggle to pay attention for the entire test. It can be seen from a comparison of the two algorithms that the MediaPipe method is more accurate at detecting asymmetrical face changes and in correctly identifying the position of the head in space while losing much less data than the CANDIDE model. However, MediaPipe model is more sensitive to noise due to the process of adapting the mesh to the face. Due to MediaPipe's incredibly lightweight code, which is created with mobile devices in mind, the processing time for videos is also reduced (see. **Table 5.3** and **Table 5.4**). For these reasons, MediaPipe was selected as the algorithm to do the child's oral motor analysis.

5.2 Classification of children by audio signal

With 234 videos of normative children and 27 videos of clinical children (Considering also the presence of several acquisitions for the same child), the dataset contains a significant samples imbalance. In order to solve this problem, the normative group was under-sampled. At the end of the division into epochs, there were 15748 epochs for clinical children and 105479 epochs for normative children. Then a random draw was made for the number of epochs of the normative children, amounting to; $15748 + 15748 \cdot 30\%$. The original dataset was separated into training (80%) test set (10%) and validation set (10%). The data were standardised before usage. The practise of standardisation makes it possible to compare quantitative variables more successfully, especially when doing so with values in various units of measurement. Each feature in this case was scaled by the standard deviation after being cleaned of the mean.

$$x = \frac{(x - u)}{s} \tag{5.1}$$

Where u is the mean and s is the standard deviation of the feature x Cross-validation was carried out to find the best hyper-parameters for classifiers K-NN and SVM. The cross-validation process was implemented using the **Time Series Split**, a K-fold technique variant. Due to the fact that our data are derived from time series and are thus not independent of one another, using conventional cross-validation approaches would result in inaccurate correlations. The samples are typically divided into k groups of equal size using the K-fold procedure, of which $(k - 1)$ folds are utilised for training and the remaining group for testing. This method splits the dataset into k folds for training, with the $(k + 1)th$ fold serving as the test set. The training set will include all k folds plus the $(k + 1)th$ fold in the subsequent iterations, while the test set will consist of the $(k + 2)th$ fold [64]. Then,

a number of metrics were determined to carry out the evaluation of the various hyper-parameters, including:

- Jaccard similarity coefficient score.
- Computation of the receiver operating characteristic area (ROC-AUC)
- Recall score
- Precision
- Compute the F1 score

Jaccard similarity: The dimensions of the intersection and the union of the set of labels defining the two classes are used to calculate this index.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.2}$$

ROC-AUC: This metric is applied to various classification methods, including binary classification. Plotting true positives vs false positives yields a ROC curve. The likelihood that a randomly chosen instance will be correctly labelled by the classifier can be thought of as the area under the ROC curve (AUC).
Recall score: The ratio of true positives to the total of true positives plus false negatives is represented by this measure. It is regarded as the capacity to detect positive samples.

$$r = \frac{tp}{(tp + fn)} \tag{5.3}$$

Where tp are the true positive and fn are the false negative
Precision: Precision is the ability of the classifier to avoid classifying negative samples as positive and is measured as the ratio of true positives to the sum of true positives and false positives.

$$p = \frac{tp}{(tp + fp)} \tag{5.4}$$

Where tp are the true positive and fp are the false positive
F1-score: The harmonic mean of the precision and racall scores is used to get the F1 score.

$$F_1 = \frac{2}{\frac{1}{r} + \frac{1}{p}} = 2 \cdot \frac{p \cdot r}{p + r} \tag{5.5}$$

5.2.1 K-nearest neighbors (K-NN)

The classification outcomes using the K-NN classifier are displayed under four different conditions with dimensionality reduction (PCA, NCA) and without dimensionality reduction. For each condition, the one that were best for the most indicators were selected as the hyper-parameters.

Without dimensionality reduction

	Jaccard	**ROC-AUC**	**Recall score**	**Precision**	**F1-score**
n=1	0.512296	0.71749	0.666339	0.688645	0.67726
n=3	0.543526	0.746184	**0.666928**	0.745886	0.70407
n=5	**0.555396**	**0.757148**	0.662292	0.774466	**0.713797**

Table 5.5: Cross-validation results for different value of number of neighborhood. For each indicator, the top values are denoted in bold.

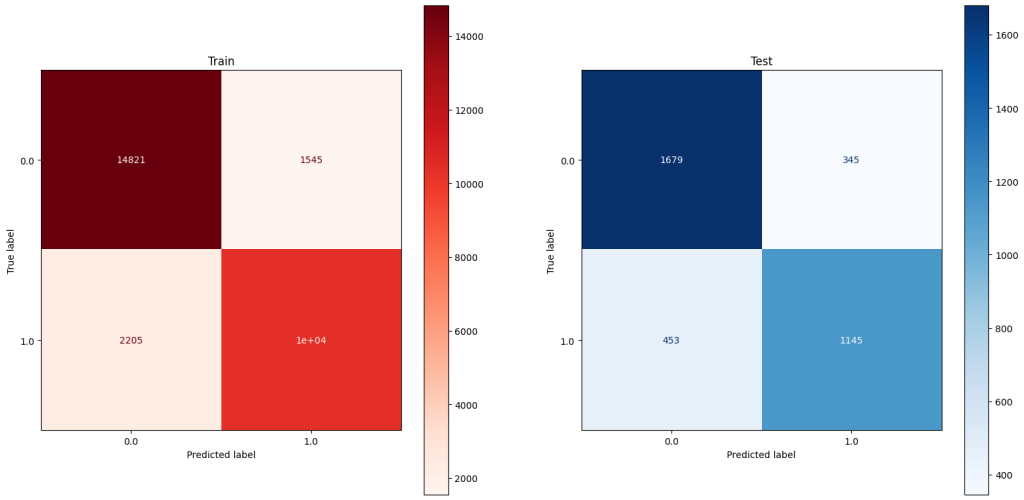


Figure 5.6: Confusion matrix after the application of a K-NN classifier without dimensionality reduction and number of neighbours equal to 5

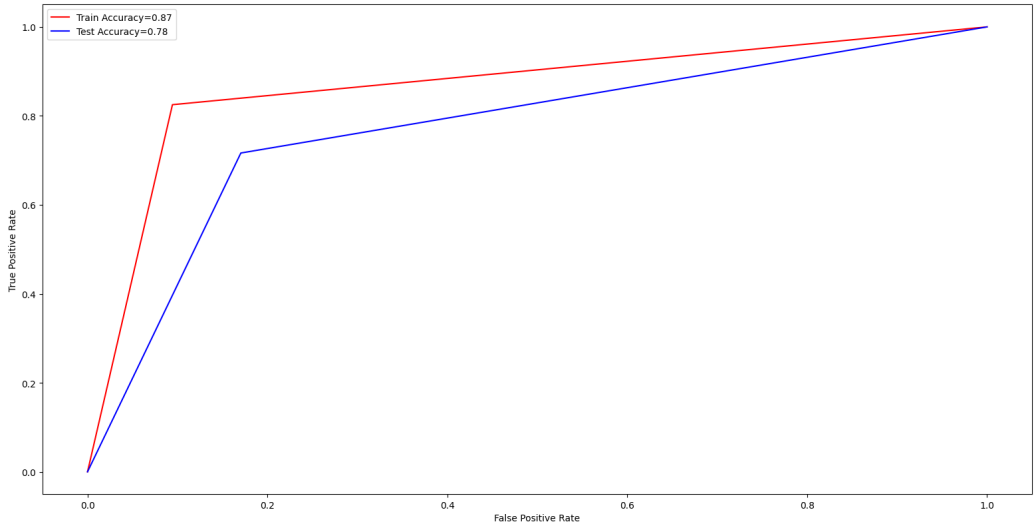


Figure 5.7: ROC curve for a K-NN classifier without dimensionality reduction and number of neighbours equal to 5

Accuracy=78%	**Precision**	**Recall score**	**F1-score**
Class(0)	79%	83%	81%
Class(1)	77%	72%	74%

Table 5.6: The classification results on the test, for a K-NN classifier without dimensionality reduction, set are summarised in the table. **0** indicates the normative class, and **1** represents the clinical class.

PCA

	Jaccard	**ROC-AUC**	**Recall score**	**Precision**	**F1-score**
n=1	0.51273	0.717925	0.666225	0.689531	0.677645
n=3	0.543934	0.746685	**0.666289**	0.747437	0.704379
n=5	**0.555919**	**0.757408**	0.663308	0.774221	**0.714265**

Table 5.7: Cross-validation results for different value of number of neighborhood. For each statistic, the top values are denoted in bold.

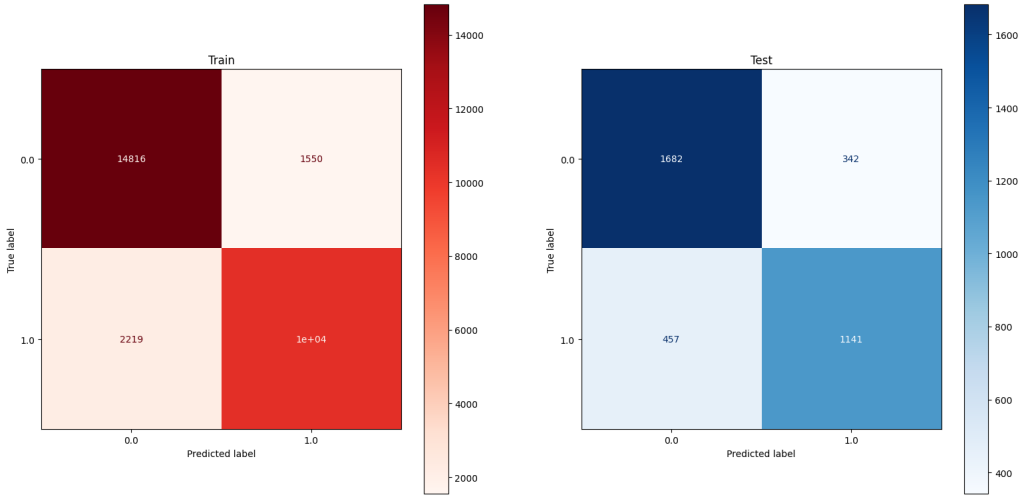


Figure 5.8: Confusion matrix for a K-NN classifier after dimensionality reduction with PCA and number of neighbours equal to 5

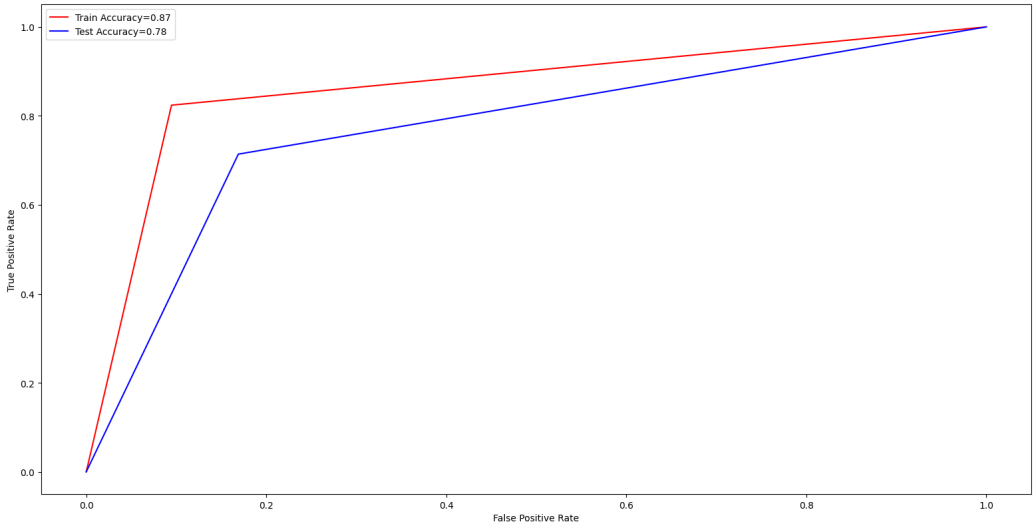


Figure 5.9: ROC curve for a K-NN after dimensionality reduction with PCA and number of neighbours equal to 5

Accuracy=78%	**Precision**	**Recall score**	**F1-score**
Class(0)	79%	83%	81%
Class(1)	77%	72%	74%

Table 5.8: The classification results on the test, for a K-NN classifier with PCA as dimensionality reduction algorithm, set are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

NCA

	Jaccard	**ROC-AUC**	**Recall score**	**Precision**	**F1-score**
n=1	0.56496	0.754951	**0.708892**	0.732049	0.720242
n=3	0.588074	0.77737	0.691168	0.794634	0.739095
n=5	**0.597787**	**0.784485**	0.695313	0.808127	**0.746784**

Table 5.9: Cross-validation results for different value of number of neighborhood. For each statistic, the top values are denoted in bold.

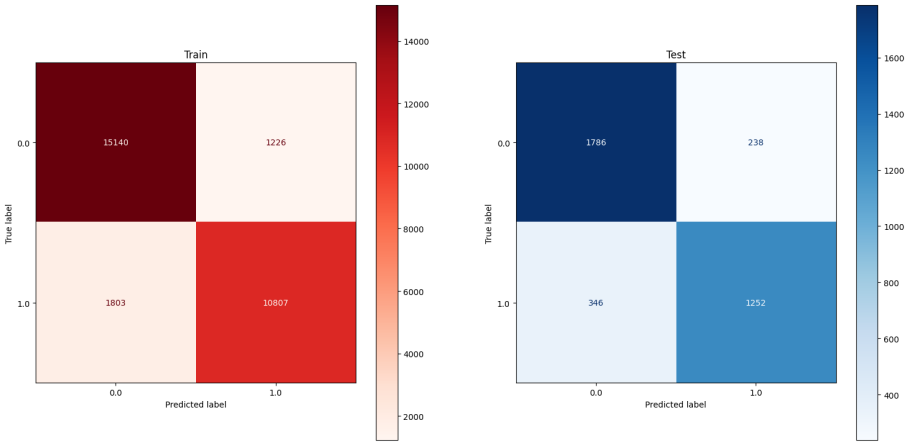


Figure 5.10: Confusion matrix for a K-NN classifier after dimensionality reduction with NCA and number of neighbours equal to 5

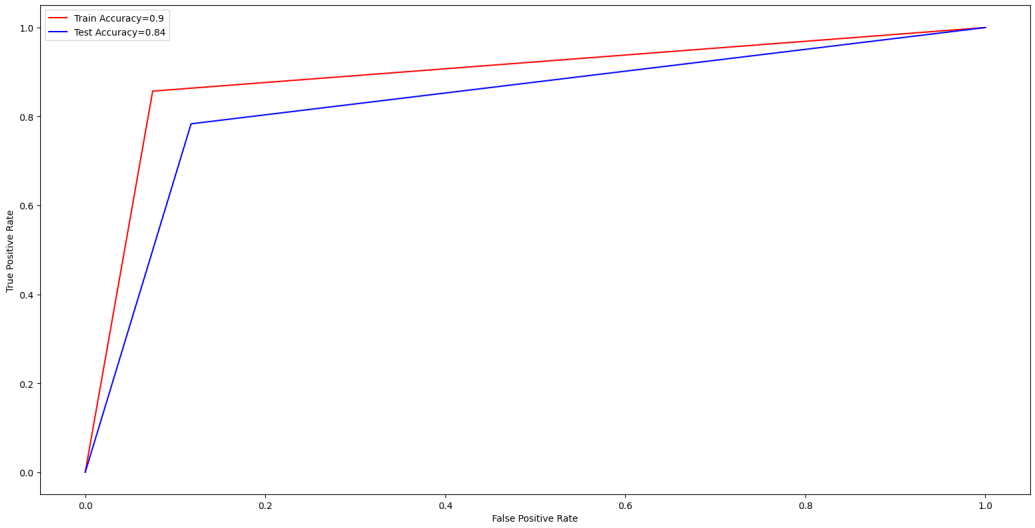


Figure 5.11: ROC curves for a K-NN classifier after dimensionality reduction with NCA and number of neighbours equal to 5

Accuracy=84%	**Precision**	**Recall score**	**F1-score**
Class(0)	84%	88%	86%
Class(1)	84%	78%	81%

Table 5.10: The classification results on the test, for a K-NN classifier with NCA as dimensionality reduction algorithm, set are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

5.2.2 Support vector machine (SVM)

It was deemed fair to apply a kernel based on radial basis function (RBF) because to the non-linearity of the supplied data. The RBF kernel function between two points serves as a gauge of their separation.

$$K(x_1, x_2) = e^{-\gamma||x_1 - x_2||^2} \tag{5.6}$$

Where σ is the variance, $||X_1 - X_2||$ is the Euclidean distance between the two points. In this instance, distance is used to measure the equality of the points. Changing the variance's value will result in a change in the number of points that are deemed equal. A cross-validation for the SVM classifier involved assessing C values ranging from 1 to 10 extremes, including. Each regularisation factor (C) was tried with two different gamma values:

$$\gamma_1 = \frac{1}{(n_{features} X.var())} \tag{5.7}$$

$$\gamma_2 = \frac{1}{(n_{features})} \tag{5.8}$$

Where $X.var()$ represent the variance of the input data. The variance of the RBF kernel and the gamma parameter are connected, so changing gamma will change the curvature of our decision boundary. Four distinct tests were conducted utilising the three different forms of dimensionality reduction and without dimensionality reduction, just like with the K-NN classifier. Below are provided the cross-validation outcomes for each of the four examples.

	No-reduction	**PCA**	**NCA**
C	3	4	3
γ	γ_1	γ_1	γ_1

Table 5.11: Cross-validation results for SVM classifier

Without dimensionality reduction

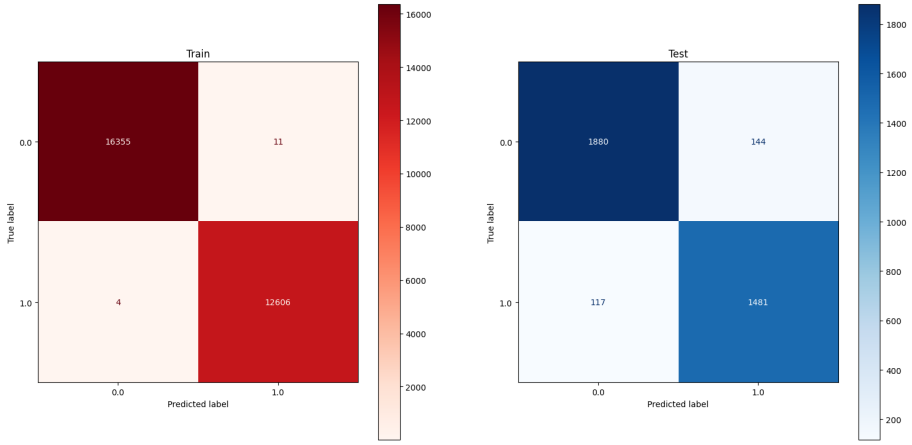


Figure 5.12: Confusion matrix for a SVM classifier without dimensionality reduction

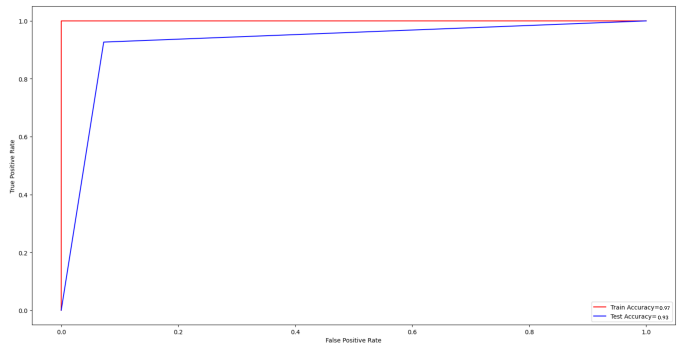


Figure 5.13: ROC curves for a SVM classifier without dimensionality reduction

Accuracy=93%	Precision	Recall score	F1-score
Class(0)	94%	93%	94%
Class(1)	91%	93%	94%

Table 5.12: The classification results on the test set for a SVM withour dimensionality reduction are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

PCA

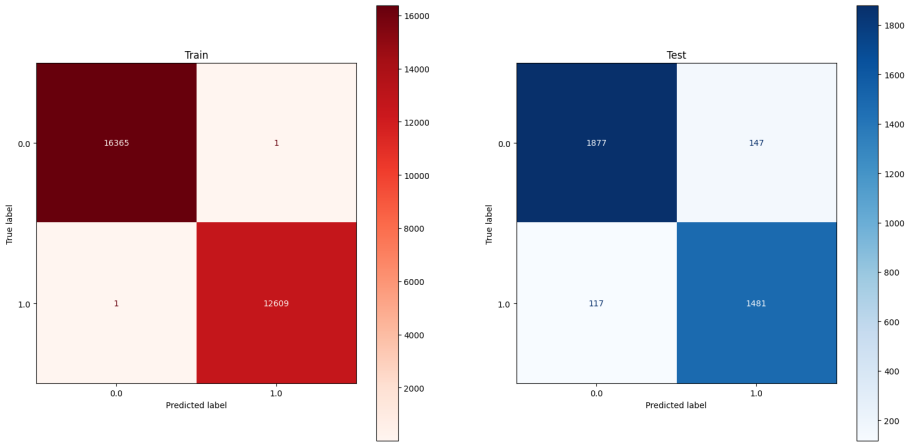


Figure 5.14: Confusion matrix for a SVM classifier with PCA

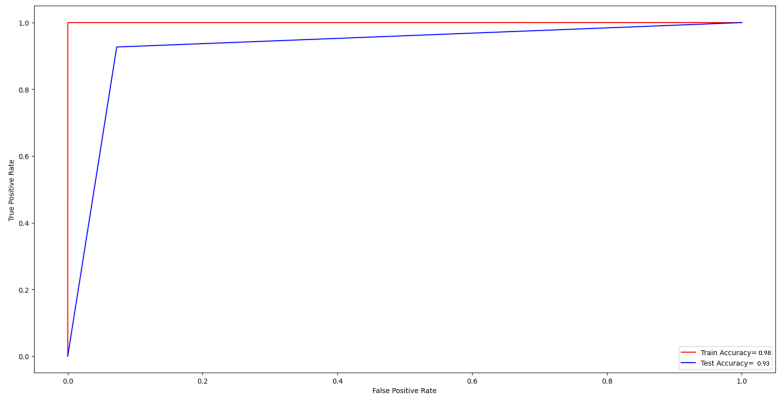


Figure 5.15: ROC curves for SVM classifier with PCA

Accuracy=93%	**Precision**	**Recall score**	**F1-score**
Class(0)	94%	93%	93%
Class(1)	91%	93%	92%

Table 5.13: The classification results on the test set, for SVM classifier with PCA as dimensionality reduction are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

NCA

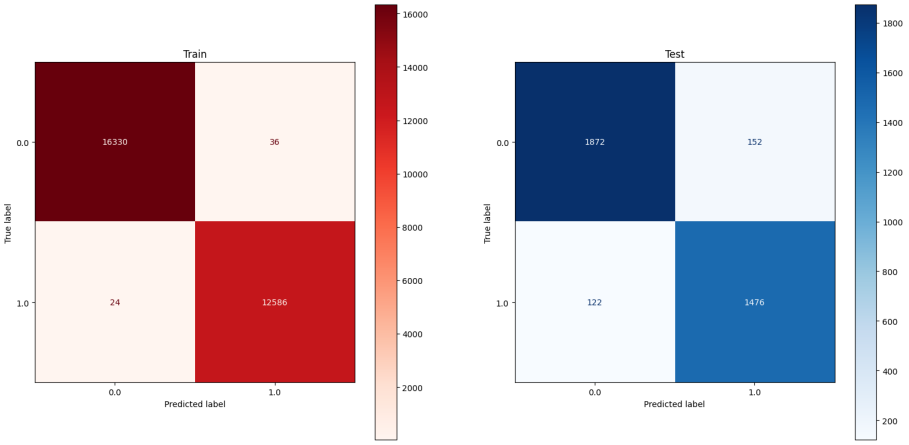


Figure 5.16: Confusion matrix for a SVM classifier with NCA

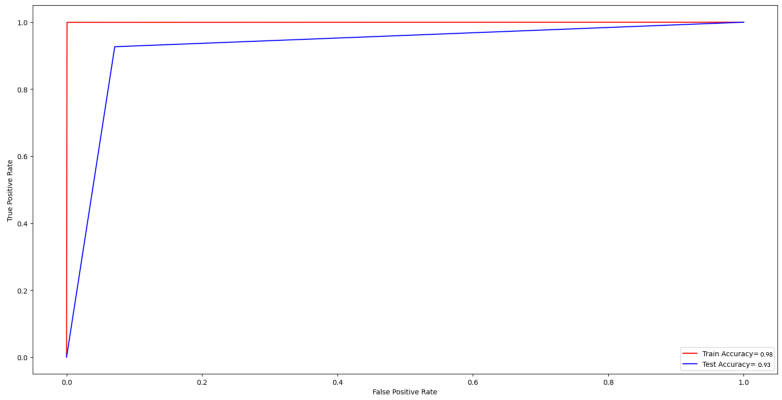


Figure 5.17: ROC curves for a SVM classifier with NCA

Accuracy=92%	**Precision**	**Recall score**	**F1-score**
Class(0)	94%	92%	93%
Class(1)	91%	92%	92%

Table 5.14: The classification results on the test set, for a SVM classifier with NCA as dimensionality reduction algorithm, are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

5.2.3 Feedforward neural network (FFNN)

Now, the FFNN neural network's results are shown.

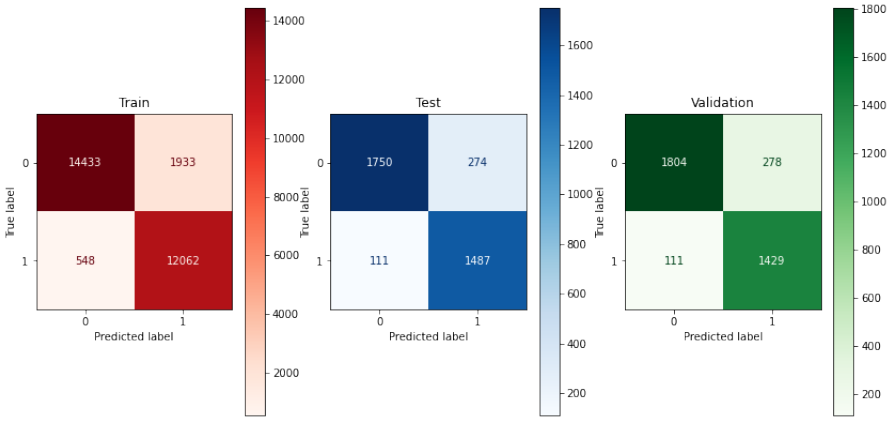


Figure 5.18: Confusion matrix for training, test and validation set after the application of a FFNN without dimensionality reduction

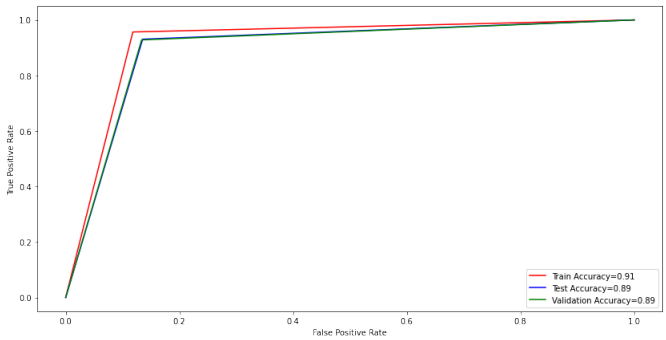


Figure 5.19: ROC curves after the application of a FFNN without dimensionality reduction

Accuracy=89%	Precision	Recall score	F1-score
Class(0)	94%	86%	90%
Class(1)	84%	93%	89%

Table 5.15: The classification results on the test, for a FFNN classifier without dimensionality reduction, set are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

PCA

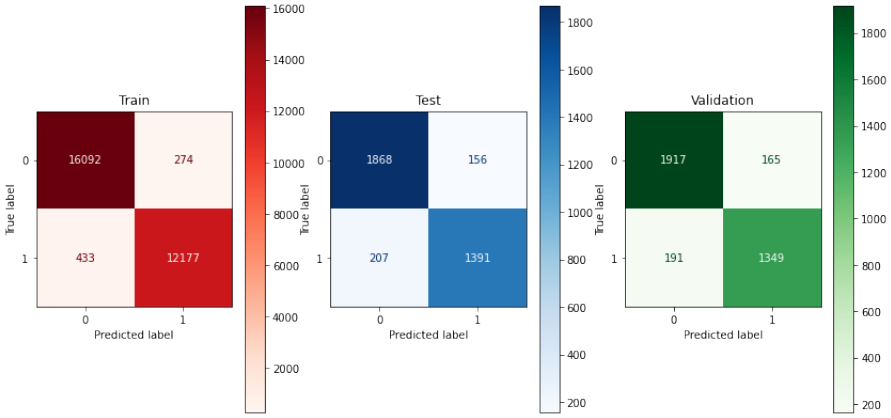


Figure 5.20: Confusion matrix after the application of FFNN with PCA

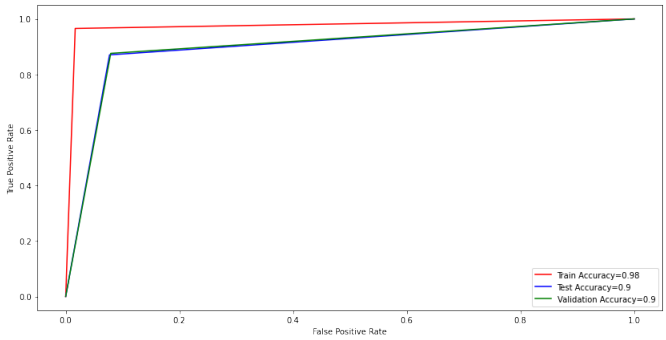


Figure 5.21: ROC curves after the application of a FFNN with PCA

Accuracy=90%	Precision	Recall score	F1-score
Class(0)	90%	92%	91%
Class(1)	90%	87%	88%

Table 5.16: The classification results on the test,for a FFNN classifier with PCA as dimensionality reduction algorithm, set are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

NCA

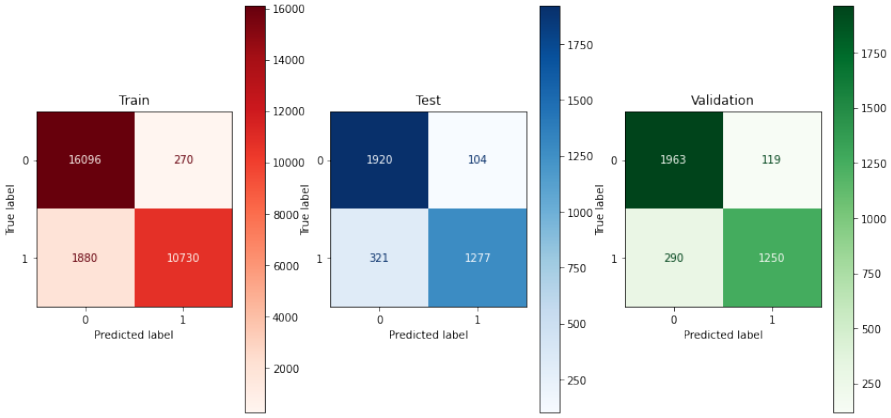


Figure 5.22: Confusion matrix for a FFNN classifier with NCA

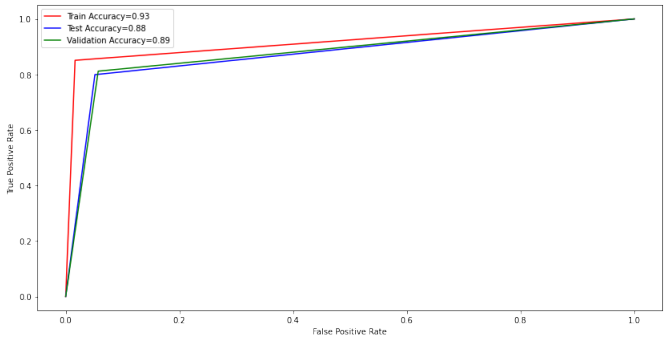


Figure 5.23: ROC curves for a FFNN classifier with NCA

Accuracy=89%	**Precision**	**Recall score**	**F1-score**
Class(0)	86%	95%	90%
Class(1)	92%	80%	86%

Table 5.17: The classification results on the test, for a FFNN classifier with NCA as dimensionality reduction algorithm, set are summarised in the table. 0 indicates the normative class, and 1 represents the clinical class.

The three classification techniques were then compared based on their accuracy, time of computing, and how dimensionality reduction algorithms affected them.

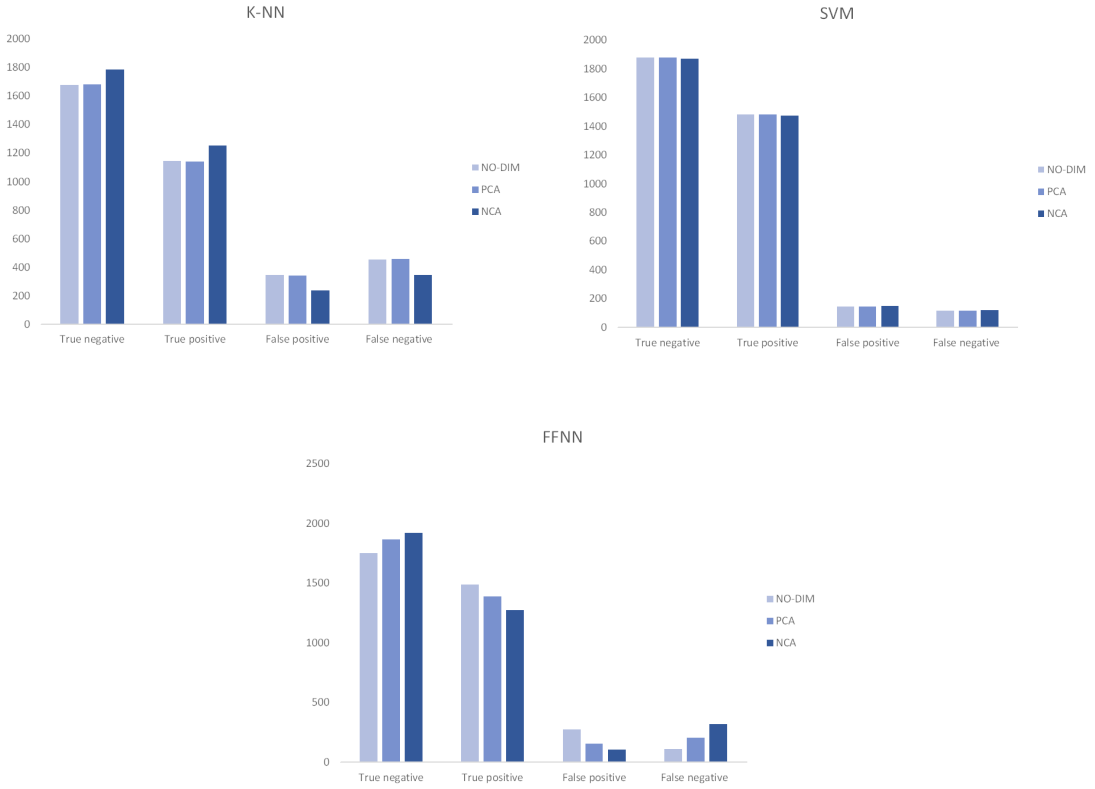


Figure 5.24: Results of the three classifiers on the test set, both without and with dimensionality reduction (PCA, NCA)

The results show that only for the K-NN classifier does the NCA algorithm improve accuracy levels. Given the accuracy levels and computational costs, PCA seems to be the most effective dimensionality reduction approach for the other two classifiers.

Time of application (s)	K-NN	SVM	FFNN
NO-DIM	65.54	1520.64	13.96
PCA	51.49	831.52	11.28
NCA	40.00	696.96	10.64

Table 5.18: Time to apply the models to training and test sets

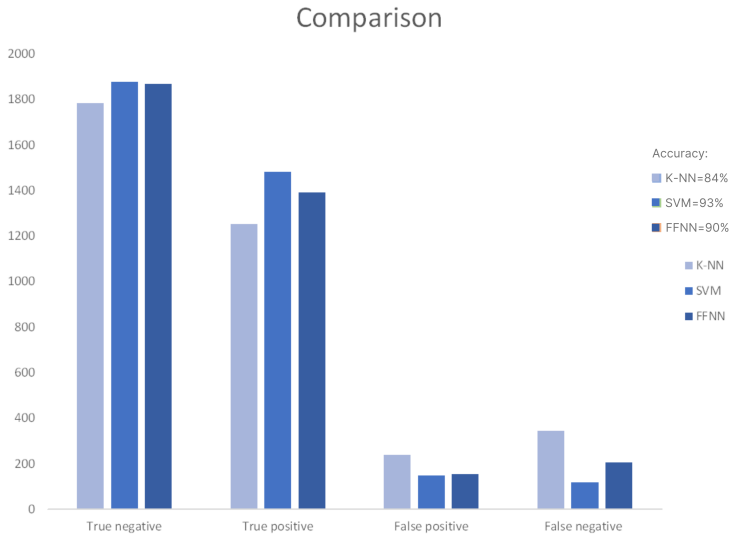


Figure 5.25: Results of the application of the three different classifiers on the test set after the application of NCA for K-NN and PCA for SVM and FFNN; The figure shows also the accuracy of all classifiers

The three classifiers have comparable performances, although the SVM classifier, when applied to this dataset, produces less false negatives than the others while maintaining the same accuracy. Figure 5.32 shows the three classifiers' training pipeline.

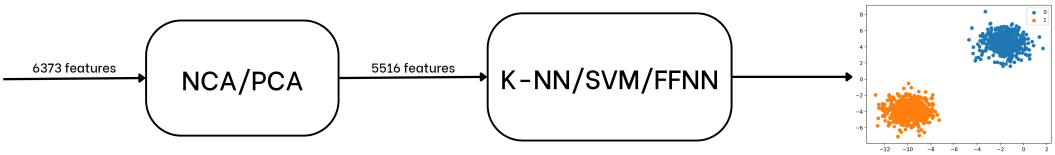


Figure 5.26: Pipeline of the classification model with the application of the dimensionality reduction algorithm and subsequently of the classifiers.
[65]

5.3 Discussion

There are many face detection algorithms that are suitable to the issue, as stated in chapter 4. It was decided to just compare these 3 algorithms in-depth because they may be utilised on a wider range of devices and yet provide acceptable performance. In the end, the MediaPipe algorithm was determined to be the most effective in terms of measurement precision and processing time. For these kinds of acquisitions, time is sometimes a deciding factor; in fact, the staff is able to repeat the work in a fair amount of time that may also be acceptable to the child. As a result, the programme may someday be changed to implement real-time acquisitions. The drawback of facial meshes is that they must go through an automatic adaption procedure that introduces high-frequency noise into the obtained data and inhibits them from tracking the quickest movements. Only significant displacements of the jaw can be observed since the markers utilised are not connected to the articular structures. Implementing dimensionality reduction has no impact on classification performance, especially when using the SVM, leaving the accuracy levels mostly unchanged (see. **Fig 5.30**). Although using dimensionality reduction before training an SVM algorithm is computationally expensive, the resulting model has significant benefits in terms of application time on the data as well as helpful effects to prevent overfitting. The NCA application performs better when using the K-NN algorithm. This is because the NCA algorithm produces a space of features, where the stochastic nearest neighbor algorithm provides the highest levels of accuracy. The fact that more hardware resources are needed for the application of the model with NCA than with the other two approaches under consideration is one of the limitations to this method. When comparing the three models, the SVM classifier with PCA had the best accuracy; however, the FFNN classifier seemed to deliver equivalent results at a lower computational expense. It will undoubtedly be necessary to use the three classifiers on various and larger datasets in order to actually evaluate how they behave.

Chapter 6

Conclusions

Healthcare professionals believed that the oral-motor features extraction programme was a good compromise for obtaining important data without interfering with the task of the children. The final results must always be thereafter be analysed and connected with additional parameters by the specialised staff because, as stated in Chapter 1, a specific evaluation cannot overlook even extremely distinctive factors like environmental and family conditions cannot be ignored in a specific examination. It was chosen not to train any automatic classifiers using the oral-motor features. This choice was reached after assessing the quantity and quality of the data at our disposal. The choice to only collect data while taking a frontal view of the child drastically limited the amount of data we had available, rendering machine learning-based classification methods useless. The length of the children's autonomous speech production was measured using the diarized audio signal, providing an additional parameter for assessment. The machine learning algorithms were trained using the audio features since they were considered to be more representative for separating the two classes of individuals. The pipelines applied to audio signals can be a significant starting point for the development of increasingly efficient models, even though the results obtained in this thesis work obviously show some biases due to the previously mentioned experimental settings. Applying and assessing the models on various datasets is required to determine how effectively robust the three techniques are. Future research could also take into account extracting more specific video features while performing particular tasks and observing how the trained classifiers behave by correlating the oral-motor and audio features. Additionally, it can be important to research the effects on the classifiers of the parameters generated by the foundation's programme, such as:

- Total number of produced words used
- Number of different words used

- Total number of words spoken
- Average length of the utterance
- Moving Average TTR: measure of lexical diversity
- Number of correct words compared to adult target production
- Number of idiosyncrasies: unrecognizable words / not belonging to the Italian linguistic system

It would be interesting to use facial meshes to get muscolar information for further projects. In fact, a preliminary analysis of the activation times of a few particular muscles may be done in the case of the CANDIDE mesh because the activation units are related to the activation of the facial muscles. Due to the fact that the altered emotional state might occasionally be a deciding factor in the diagnosis of language disorders, doing a preliminary assessment (sentiment analysis) could strengthen the classification algorithm and give the medical staff additional information.

Bibliography

[1] Letizia Sabbadini. *Disturbi specifici del linguaggio, disprassie e funzioni esecutive*. Springer, 2013 (cit. on pp. 1, 3, 6).

[2] *Apparato fonatorio: come si origina la voce*. https://www.studiarapido.it/apparato-fonatorio-come-si-origina-la-voce (cit. on p. 2).

[3] C Rahmad, R A Asmara, D R H Putra, I Dharma, H Darmono, and I Muhiqqin. «Comparison of Viola-Jones Haar Cascade Classifier and Histogram of Oriented Gradients (HOG) for face detection». In: *IOP Conference Series: Materials Science and Engineering* 732.1 (Jan. 2020), p. 012038. DOI: 10.1088/1757-899X/732/1/012038. URL: https://dx.doi.org/10.1088/1757-899X/732/1/012038 (cit. on pp. 11–13).

[4] Francisco A. Pujol, María José Pujol, Carlos Rizo-Maestre, and Mar Pujol. «Entropy-Based Face Recognition and Spoof Detection for Security Applications». In: *Sustainability* 12.1 (2020). ISSN: 2071-1050. DOI: 10.3390/su12010085. URL: https://www.mdpi.com/2071-1050/12/1/85 (cit. on pp. 13, 14).

[5] Tiemeng Li, Wenjun Hou, Fei Lyu, Yu Lei, and Chen Xiao. «Face Detection Based on Depth Information Using HOG-LBP». In: *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*. 2016, pp. 779–784. DOI: 10.1109/IMCCC.2016.92 (cit. on p. 14).

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications». In: *CoRR* abs/1704.04861 (2017). arXiv: 1704.04861. URL: http://arxiv.org/abs/1704.04861 (cit. on p. 15).

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. «MobileNetV2: Inverted Residuals and Linear Bottlenecks». In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474 (cit. on p. 15).

[8] *Video Classification using Deep Learning - Facial Detection and Feature Extraction (1)*. https://raphaellederman.github.io/articles/videoclassification/. Accessed: 2010-09-30 (cit. on p. 15).

[9] Gonzalo D. Sad, Facundo Reyes, and Julián Alvarez. «Asymmetric 3D face model for Speech Language Pathologist applications». In: *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. 2021, pp. 01–08. DOI: 10.1109/FG52635.2021.9666967 (cit. on pp. 16, 21, 25, 26, 30–33, 53).

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. «SSD: Single Shot MultiBox Detector». In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 21–37. ISBN: 978-3-319-46448-0 (cit. on p. 17).

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. «SSD: Single Shot MultiBox Detector». In: *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2. URL: https://doi.org/10.10072F978-3-319-46448-0_2 (cit. on p. 17).

[12] Jörgen Ahlberg. «CANDIDE-3 - An Updated Parameterised Face». In: 2001 (cit. on p. 17).

[13] *FaceSwap*. https://github.com/MarekKowalski/FaceSwap (cit. on pp. 18, 19).

[14] Vahid Kazemi and Josephine Sullivan. «One millisecond face alignment with an ensemble of regression trees». In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 1867–1874 (cit. on p. 19).

[15] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, and Matthias Grundmann. *Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs*. 2019. DOI: 10.48550/ARXIV.1907.06724. URL: https://arxiv.org/abs/1907.06724 (cit. on p. 22).

[16] Artsiom Ablavatski, Ivan Grishchenko, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. *Attention Mesh: High fidelity face mesh prediction in real-time*. 2020 (cit. on pp. 22–24).

[17] *Mediapipe face mesh*. https://google.github.io/mediapipe/solutions/face_mesh.html (cit. on p. 24).

[18] *Approximate Focal Length for Webcams and Cell Phone Cameras*. https://learnopencv.com/approximate-focal-length-for-webcams-and-cell-phone-cameras/ (cit. on p. 27).

[19] *OpenCV:Perspective-n-Point (PnP) pose computation*. https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html (cit. on p. 28).

[20] Kaj Madsen, Hans Nielsen, and O Tingleff. «Methods for Non-Linear Least Squares Problems (2nd ed.)» In: (Jan. 2004), p. 60 (cit. on p. 29).

[21] Andrea Bandini, Aravind Namasivayam, and Yana Yunusova. «Video-Based Tracking of Jaw Movements During Speech: Preliminary Results and Future Directions». In: Aug. 2017, pp. 689–693. DOI: 10.21437/Interspeech.2017-1371 (cit. on pp. 31, 54).

[22] Andrea Bandini, Jordan R. Green, Babak Taati, Silvia Orlandi, Lorne Zinman, and Yana Yunusova. «Automatic Detection of Amyotrophic Lateral Sclerosis (ALS) from Video-Based Analysis of Facial Movements: Speech and Non-Speech Tasks». In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. 2018, pp. 150–157. DOI: 10.1109/FG.2018.00031 (cit. on pp. 34, 54).

[23] Stephen Butterworth et al. «On the theory of filter amplifiers». In: *Wireless Engineer* 7.6 (1930), pp. 536–541 (cit. on p. 34).

[24] Yogesh Sharma and Bikesh Kumar Singh. «Classification of Children with Specific Language Impairment Using Pitch-Based Parameters». In: *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*. 2020, pp. 42–46. DOI: 10.1109/RAICS51191.2020.9332499 (cit. on pp. 36, 43, 45, 55).

[25] Mittapalle Kiran Reddy, Paavo Alku, and Krothapalli Sreenivasa Rao. «Detection of Specific Language Impairment in Children Using Glottal Source Features». In: *IEEE Access* 8 (2020), pp. 15273–15279. DOI: 10.1109/ACCESS.2020.2967224 (cit. on pp. 36, 41, 55).

[26] Hervé Bredin et al. «pyannote.audio: neural building blocks for speaker diarization». In: *ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing*. Barcelona, Spain, May 2020 (cit. on p. 37).

[27] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget. «Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks». In: *Computer Speech & Language* 71 (2022), p. 101254. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2021.101254. URL: https://www.sciencedirect.com/science/article/pii/S0885230821000619 (cit. on p. 37).

[28] Hervé Bredin and Antoine Laurent. «End-to-end speaker segmentation for overlap-aware resegmentation». In: *Proc. Interspeech 2021*. 2021 (cit. on pp. 37, 40).

[29] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. *End-to-End Neural Speaker Diarization with Permutation-Free Objectives*. 2019. DOI: 10.48550/ARXIV.1909.05952. URL: https://arxiv.org/abs/1909.05952 (cit. on p. 38).

[30] *Differences Between Bidirectional and Unidirectional LSTM*. https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm (cit. on p. 38).

[31] Chengkai Cai, Kenta Iwai, and Takanobu Nishiura. «Speech Enhancement Based on Two-Stage Processing with Deep Neural Network for Laser Doppler Vibrometer». In: *Applied Sciences* 13.3 (2023). ISSN: 2076-3417. DOI: 10.3390/app13031958. URL: https://www.mdpi.com/2076-3417/13/3/1958 (cit. on p. 38).

[32] Hervé Bredin. «PYANNOTE. AUDIO 2.1 SPEAKER DIARIZATION PIPELINE¦ PRINCIPLE, BENCHMARK, AND RECIPE». In: () (cit. on pp. 39, 40).

[33] Björn Schuller et al. «The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language». In: *17TH Annual Conference of the International Speech Communication Association (Interspeech 2016), Vols 1-5*. 2016, pp. 2001–2005 (cit. on p. 41).

[34] Felix Weninger, Florian Eyben, Björn Schuller, Marcello Mortillaro, and Klaus Scherer. «On the Acoustics of Emotion in Audio: What Speech, Music, and Sound have in Common». In: *Frontiers in Psychology* 4 (2013). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00292. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2013.00292 (cit. on p. 42).

[35] Mariana Julião. «Feature Sets for Stressed Speech Discrimination». In: 2014 (cit. on p. 42).

[36] Björn Schuller et al. «Affective and Behavioural Computing: Lessons Learnt from the First Computational Paralinguistics Challenge». In: *Computer Speech and Language* 53 (Jan. 2019), pp. 156–180. DOI: 10.1016/j.csl.2018.02.004. URL: https://hal.science/hal-01993250 (cit. on p. 42).

[37] Carlos M. Jarque and Anil K. Bera. «Efficient tests for normality, homoscedasticity and serial independence of regression residuals». In: *Economics Letters* 6.3 (1980), pp. 255–259. ISSN: 0165-1765. DOI: https://doi.org/10.1016/0165-1765(80)90024-5. URL: https://www.sciencedirect.com/science/article/pii/0165176580900245 (cit. on p. 43).

[38] Robert Tomšik. «Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov and Jarque-Bera Tests». In: 3 (May 2019), pp. 238–243 (cit. on p. 43).

[39] *Student's t-test*. https://en.wikipedia.org/wiki/Student{%}27s_t-test{#}Independent_two-sample_t-test (cit. on p. 44).

[40] *Mann–Whitney U test*. https://en.wikipedia.org/wiki/Mann{%}E2{%}80{%}93Whitney_U_test (cit. on p. 44).

[41] Howard Levene et al. «Contributions to probability and statistics». In: *Essays in honor of Harold Hotelling* 278 (1960), p. 292 (cit. on p. 44).

[42] *Welch's t-test*. https://en.wikipedia.org/wiki/Welch{%}27s_t-test (cit. on p. 44).

[43] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. «Neighbourhood Components Analysis». In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004. URL: https://proceedings.neurips.cc/paper/2004/file/42fe880812925e520249e808937738d2-Paper.pdf (cit. on p. 45).

[44] Ian T. Jolliffe and Jorge Cadima. «Principal component analysis: a review and recent developments». In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202. DOI: 10.1098/rsta.2015.0202. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2015.0202. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0202 (cit. on p. 45).

[45] *KNN Regression Model in Python*. https://towardsdatascience.com/knn-regression-model-in-python-9868f21c9fa2. Accessed: 2010-09-30 (cit. on p. 46).

[46] Huy Nguyen Duc, Innocent Kamwa, Louis-A Dessaint, and Huy Cao-Duc. «A Novel Approach for Early Detection of Impending Voltage Collapse Events Based on the Support Vector Machine». In: *International Transactions on Electrical Energy Systems* 27 (Mar. 2017). DOI: 10.1002/etep.2375 (cit. on p. 47).

[47] *Feedforward neural network*. https://en.wikipedia.org/wiki/Feedforward_neural_network (cit. on p. 48).

[48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html (cit. on p. 48).

[49] Xavier Glorot, Antoine Bordes, and Y. Bengio. «Deep Sparse Rectifier Neural Networks». In: vol. 15. Jan. 2010 (cit. on p. 49).

[50] Tomasz Szandała. «Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks». In: Jan. 2021, pp. 203–224. ISBN: 978-981-15-5494-0. DOI: 10.1007/978-981-15-5495-7_11 (cit. on p. 49).

[51] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2016. DOI: 10.48550/ARXIV.1609.04747. URL: https://arxiv.org/abs/1609.04747 (cit. on p. 49).

[52] Vidit Jain and Erik Learned-Miller. *FDDB: A Benchmark for Face Detection in Unconstrained Settings*. Tech. rep. UM-CS-2010-009. University of Massachusetts, Amherst, 2010 (cit. on p. 51).

[53] Peter M. Roth Martin Koestinger Paul Wohlhart and Horst Bischof. «Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization». In: *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*. 2011 (cit. on p. 51).

[54] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. *WIDER FACE: A Face Detection Benchmark*. 2015. DOI: 10.48550/ARXIV.1511.06523. URL: https://arxiv.org/abs/1511.06523 (cit. on p. 51).

[55] *The PASCAL Object Recognition Database Collection*. http://host.robots.ox.ac.uk/pascal/VOC/databases.html. Accessed: 2010-09-30 (cit. on p. 51).

[56] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z. Li, and Xudong Zou. *Selective Refinement Network for High Performance Face Detection*. 2018. DOI: 10.48550/ARXIV.1809.02693. URL: https://arxiv.org/abs/1809.02693 (cit. on pp. 51, 52).

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385 (cit. on p. 52).

[58] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Jilin Li, and Feiyue Huang. *DSFD: Dual Shot Face Detector*. 2018. DOI: 10.48550/ARXIV.1810.10220. URL: https://arxiv.org/abs/1810.10220 (cit. on pp. 52, 53).

[59] Gonzalo Sad, Facundo Reyes, and Julián Alvarez. «FaceTrack: Asymmetric Facial and Gesture Analysis Tool for Speech Language Pathologist Applications». In: Oct. 2021, pp. 1–10. DOI: 10.1145/3476100.3484460 (cit. on pp. 53, 54).

[60] PAAVO ALKU. «Glottal inverse filtering analysis of human voice production — A review of estimation and parameterization methods of the glottal excitation and their applications». In: *Sadhana* 36 (Oct. 2011). DOI: 10.1007/s12046-011-0041-5 (cit. on p. 55).

[61] Matti Airas, Hannu Pulakka, Tomas Bäckström, and Paavo Alku. «A toolkit for voice inverse filtering and parametrisation». In: *Interspeech*. 2005 (cit. on p. 55).

[62] Manu Airaksinen, Tuomo Raitio, Brad Story, and Paavo Alku. «Quasi Closed Phase Glottal Inverse Filtering Analysis With Weighted Linear Prediction». In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 22 (Mar. 2014), pp. 596–607. DOI: 10.1109/TASLP.2013.2294585 (cit. on p. 56).

[63] C S Kanimozhiselvi and S Santhiya. «Communication Disorder Identification from Recorded Speech using Machine Learning Assisted Mobile Application». In: *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*. 2021, pp. 789–793. DOI: 10.1109/ICICV50876.2021.9388493 (cit. on p. 56).

[64] *Cross-validation: evaluating estimator performance*. https://scikit-learn.org/stable/modules/cross_validation.html. Accessed: 2010-09-30 (cit. on p. 66).

[65] *4 Types of Classification Tasks in Machine Learning*. https://machinelearningmastery.com/types-of-classification-in-machine-learning/ (cit. on p. 81).