Managing Affective-learning THrough Intelligent atoms and Smart InteractionS

D4.2 MaTHiSiS sensorial component (M18)

Workpackage	WP4 - Affective and Natural Interaction Instruments
Editor(s):	Dimitrios ANTONARAS, CERTH Ioannis TSINGALIS, CERTH Martha POLITOU, CERTH Christina CHRYSOULI, CERTH Dorothea TSATSOU, CERTH Nicholas VRETOS, CERTH Enrique HORTAL, UM Esam GHALEB, UM Evaggelos SPYROU, NCSR Dimitrios SGOUROPOULOS, NCSR
Responsible Partner:	*CERTH*
Quality Reviewers	Stylianos Asteriadis, UM Thomas Techene, DXT
Status-Version:	Final – v1.0
Date:	Project Start Date: 01/01/2016; Duration: 36 months Deliverable Due Date: 30/06/2017

This document is issued within the frame and for the purpose of the *MATHISIS* project. This project has received funding from the European Union's Horizon 2020 Programme (H2020-ICT-2015) under Grant Agreement No. 687772

	Submission Date: 30/06/2017
EC Distribution:	Report, Public
Abstract:	This deliverable presents the progress of the different modalities employed within MaTHiSiS in order to extract learners' affective cues from different sensors, i.e. from depth and RGB cameras, microphones and inertial sensors embedded in mobile devices. The modalities that take advantage of this sensorial input in order to understand the affective state of the user include facial expression analysis, gaze estimation, speech recognition and speech-based affect recognition, skeleton motion analysis and inertia sensor-based affect recognition in mobile devices.
Keywords:	Sensorial Component, affect recognition, facial expression analysis, gaze estimation, speech recognition, speech-based affect recognition, skeleton motion analysis, mobile inertia sensors-based affect recognition
Related Deliverable(s)	D2.4 *Full System Architecture (M15)* D2.6 *Framework for impact assessment of MaTHiSiS against LEPOSA requirements* D2.8 *Report on monitoring of LEPOSA requirements (M18)* D4.1 *MaTHiSiS sensorial component (M12)* D6.1 *Adaptation and Personalization principles based on MaTHiSiS findings* D7.2 *MaTHiSiS Platform, 1st release* D8.1 *Report on Autism Spectrum Case pilots (M16)* D8.4 *Report on Profound and Multiple Learning Disabilities Case pilots (M16)* D8.7 *Report on Mainstream Education Case pilots (M16)* D9.1 *Report on Industrial Training pilots (M16)* D9.4 *Report on Career Guidance Distance Learning pilots (M16)*

Document History

Version	Date	Change editors	Changes
0.1	18/05/2017	Dorothea TSATSOU, CERTH Nicholas Vretos, CERTH	Table of contents.
0.2	13/06/2017	Esam Ghaleb, UM Enrique Hortal, UM	Input on chapters 2.2.3, 4.2, 5.2.
0.3	16/06/2017	Dimitrios Antonaras, CERTH Martha Polition, CERTH, Ioannis Tsingalis, CERTH Christina Chrysouli, CERTH	Aggregation of UM input in main document. Input on chapters 2, 3, 4, 5.
0.4	20/06/2017	Evaggelos Spyrou, NSCR Dimitrios Sgouropoulos, NCSR	Input on chapters 2.2.5, 3.5.
0.5	21/06/2017	Dimitrios Antonaras, CERTH	Aggregation of NSCR input to main document. Input on chapters 2, 4.
0.6	22/06/2017	Enrique Hortal, UM	Input on chapters 3.1.2, 3.3
0.7	23/06/2017	Dorothea Tsatsou, CERTH	Input on chapters 1, 6. Additions and modifications, all chapters (1-6).
0.8	23/06/2017	Dorothea Tsatsou, CERTH Nicholas Vretos, CERTH	Proofreading, homogenization, formatting. Version sent for internal review.
0.9	28/06/2017	Dimitrios Antonaras, CERTH Dorothea Tsatsou, CERTH Nicholas Vretos, CERTH	Revisions based on internal review. Version sent to UM for second review.
0.95	29/06/2017	Ana Piñuela, ATOS	Final quality check
1.0	29/06/2017	Dimitrios Antonaras,	Final version, to be sent to the EC.

Version	Date	Change editors	Changes
		CERTH Dorothea Tsatsou, CERTH	

Table of Contents

List of Tables

List of Figures

List of Acronyms

Abbreviation / acronym	Description
AFEW	Acted Faces Emotion In The Wild
ASC	Autism Spectrum Case
ASD	Accumulated Squared Derivatives
CGDLC	Career Guidance Distance Learning Case
CLS	Cloud-based Learner's Space
CNN	Convolutional Neural Network
DAT	Data Acquisition Tool
DSS	Decision Support System
D-SFA	Discriminant Slow Feature Analysis
D-SIFT	Dense Scale Invariant Feature Transform
EM	Expectation Maximization
ES	Experiencing Service
GMM	Gaussian Mixture Models
HCI	Human-Computer Interaction
HMA	Human Motion analysis
HoF	Histogram of optic Flows
HoG	Histogram of Gradients
ITC	Industrial Training Case
LSTM	Long Short-Term Memory
MEC	Mainstream Education Case
MMF	Multimodal Fusion
PA	Platform Agent
PMLDC	Profound and Multiple Learning Disabilities Case
RNN	Recurrent Neural Network

Abbreviation / acronym	Description
SC	Sensorial Component
SDM	Supervised Descent Method
SD-SFA	Spatially-discriminative Slow Feature Analysis
SFA	Slow Feature Analysis
SIFT	Scale Invariant Feature Transform
SLA	Smart Learning Atoms
SVM	Support Vector Machine
S-SFA	Supervised Slow Feature Analysis
U-SFA	Unsupervised Slow Feature Analysis

Table 1: Definitions, Acronyms and Abbreviations

Project Description

The MaTHiSiS learning vision is to provide a novel advanced digital ecosystem for vocational training, and special needs and mainstream education for individuals with an intellectual disability (ID), autism and neuro-typical learners in school-based and adult education learning contexts. This ecosystem consists of an integrated platform, along with a set of re-usable learning components with capabilities for: i) adaptive learning, ii) automatic feedback, iii) automatic assessment of learners' progress and behavioural state, iv) affective learning, and v) game-based learning.

In addition to a learning ecosystem capable of responding to a learner's affective state, the MaTHiSiS project will introduce a novel approach to structuring the learning goals for each learner. Learning graphs act as a novel educational structural tool. The building materials of these graphs are drawn from a set of Smart Learning Atoms (SLAs) and a set of specific learning goals which will constitute the vertices of these graphs, while relations between SLAs and learning goals constitute the edges of the graphs. SLAs are atomic and complete pieces of knowledge which can be learned and assessed in a single, short-term iteration, targeting certain problems. More than one SLA, working together on the same graph, will enable individuals to reach their learning and training goals. Learning goals and SLAs will be scoped in collaboration with learners themselves, teachers and trainers in formal and non-formal education contexts (general education, vocational training, lifelong training and specific skills learning).

MaTHiSiS is a 36 month long project co-funded by the European Commission Horizon 2020 Programme (H2020-ICT-2015), under Grant Agreement No. 687772.

Executive Summary

This document reports on the progress and final solution in terms of methodologies followed, technologies developed and results of the MaTHiSiS Sensorial Component (SC), i.e. Task 4.1 MaTHiSiS Sensorial Compnent, as well as the ground truth data collection, the implementation strategy and the in situ use of the SC within MaTHiSiS. This is broken down for each of the five Sensorial Component modalities, namely facial expression analysis, gaze detection, skeleton motion analysis, audio analysis and mobile-based inertia sensors analysis.

SC development follows initial technical and user requirements, driven by the system architecture (Deliverable 2.3 *Full System Architecture M6,* Deliverable 2.4 *Full System Architecture M15*) and the first scenarios developed for MaTHiSiS (Deliverable D2.2 *Full Scenarios of All Use Cases*) respectively. This document (Deliverable 4.2 *MaTHiSiS Sensorial Component M18*) is a result of an iterative process and while the main implementation core of the SC has concluded along with the respective Task 4.1, the work will keep being refined based on the MaTHiSiS datasets and towards the requirements of the multimodal fusion (Task 4.3) for multimodal affect recognition.

To this end, this document provides a detailed description of the established SC algorithms for all SC modalities, along with comparative scientific results (Section 3). Furthermore, the document describes the data collection conducted within MaTHiSiS, in order to build datasets for all MaTHiSiS use cases to appropriately train the affect recognition algorithms of WP4 (Section 2), along with first observations. It also details the communication strategy of the Sensorial Component with the MaTHiSiS platform (Section 4) and finally it provides an overview of the implementation of the SC in the integrated MaTHiSiS ecosystem, towards multimodal affect recognition, including its presence in the driver pilots (Section 5).

1. Introduction

As already mentioned in the first iteration of this document (Deliverable D4.1 *The MaTHiSiS Sensorial component M12*), effective, hyper-personalised learning within MaTHiSiS relies on the adaptation of parameters in the learning experience, based upon each learner's affective state.

To this end, MaTHiSiS' Sensorial Component (SC) gathers (physical) behavioural cues of the learner and interprets them into affective states, with the use of machine learning, in order understand each learner's uptake of knowledge during the learning process. Different modalities extract and represent affect-related features stemming from the learners' face, gaze, body postures, speech and inertia-sensor captured interaction with mobile PAs of behavioural expression. These in term are fused into a common affective state in Task 4.3, which in provides a holistic overview of the learner's affective state from all sensory resources available.

An inherent part of the SC (and all of WP4) is the ground truth data that are going to be used to train the respective machine learning algorithms. Although for most of the SC modalities, public datasets are available, and developed algorithms have been trained and tested on such information, where available (Section 3), the particularities of the MaTHiSiS use cases (wide range of age groups and learners with special needs) cannot be met through all of them. It is also worth noting that there are no datasets available for data captured through the newest sensors, such as mobile inertial sensors.

Therefore, WP4 has organised the collection of real-life data, for all types of MaTHiSiS learners and for all use cases, in the premise of the project and by users that take part in the MaTHiSiS piloting phases, as described in Section 2. Section 4 presents the WP4 components flow within the MaTHiSiS ecosystem and the SC's positioning in that and describes the communication flow and protocols of the SC in relation to the MaTHiSiS Platform Agents and cloud. Finally, Section 5 presents information of connection of the SC with the multimodal fusion and details about the implementation of the SC in the MaTHiSiS driver pilots.

2. MaTHiSiS dataset

This section provides information about the datasets gathered from MaTHiSiS users, representative of the varied MaTHiSiS use cases, during the data acquisition pre-pilots, with the use of the Data Acquisition Tool (DAT) described in Section 4.4 of Deliverable D4.1. An overview of the datasets per modality and per user type & learning environment type, where applicable, is presented, along with conclusions/observations made.

2.1 General information

The lack of publically available datasets, representing the variety of learner types that are involved in the MaTHiSiS use cases, that could be used to train the algorithms of the MaTHiSiS Sensorial Component modalities, as well as the rest of the WP4 algorithms (Interaction with PAs (T4.2) and Multimodal Fusion for uniform affect recognition (T4.3)) led to the design and collection of the MaTHiSiS affect dataset. This dataset is a collection of video, sound, skeleton and mobile data and ground truth annotations (automatically or manually derived) over them. It involves data gathered using the Data Acquisition Tool (DAT), while it will continue growing, adding data from the following pilots phases.

Data gathering took place in the premises of the MaTHiSiS-collaborating educational institutions, where a subset of the MaTHiSiS learners, that take part in the pilots, were asked to undertake a series of activities. These gamified activities were set up in a way that allowed for a) the automatic collection of first level ground truth per each SC modality (e.g. specific emotive facial expressions and body motions, controlled eye gaze direction identification, etc.), as well as for b) the manual expert-based annotation of all recorded activities, with salient affect state labels based on the Theory of Flow (namely boredom, engagement, frustration) [1].

Data were collected according to legal and ethical policies established in MaTHiSiS project described by the "MaTHiSiS Ethics protocol" and "MATHISIS Data Sharing and Data Security Protocol" documents. Learners are identified by pseudonyms while all data were stored locally on the local PCs' hard drives. All the data were encrypted and transferred to CERTH's Network-Attached Storage (NAS)[1] server over a secured network or via an external drive sent via courier to the CERTH premises, that CERTH transferred on the NAS server, again encrypted according to the Data Sharing and Data Security Protocol. T4.1 partners apart from CERTH that needed to use the datasets were provided with secure accounts to access the encrypted data and download them locally to their local premises. More details about the procedures of data transfer and storage can be found in deliverable D2.8 *Report on monitoring of LEPOSA requirements (M18).*

The sensors used to gather data are the following:

- Kinect v2 camera: using this sensor high-resolution (1920x1080) image sequences, skeleton data and high quality (16000 Kbit) audio are gathered.
- Web camera: provided low resolution (640X480) image sequences .
- Mobile devices' gyroscope, accelerometer and touch panel on the screen of smartphones and tablets collected data concerning the device's orientation, positioning and interactions on it.

Table 2 presents information regarding the MaTHiSiS dataset collected so far, per each user partner, per use case and in total, about the amount of the data, the kind of sessions and the provided second level annotations over the data. More details about the learners and the number of them selected

[1] https://en.wikipedia.org/wiki/Network-attached_storage

per MaTHiSiS case are presented in (8.1, 8.4, 8.7, 9.1, 9.4). This chapter also describes the dataset in depth per modality, while it show the general observations about the data gathered.

Table 2: Data Acquisition Tool Participation

Partner	Learners (male/female)	Learners per UC	Sessions	Sessions per UC	Annotated Sessions (2nd level manual annotation)	Annotated Sessions (2nd level manual annotation) per UC
FMD	10 (6/4)	MEC: 3 ASC:7	29	MEC: 9 ASC: 20	27	MEC:8 ASC: 19
PE	9 (6/3)	MEC: 8 PMLD:1	28	MEC: 25 PMLD: 3	26	MEC: 23 PMLD: 3
RIX	6 (3/3)	MEC: 2 PMLD: 4	12	MEC: 7 PMLD: 5	2	MEC: 1 PMLD:1
EOPPEP	5 (3/2)	CGDLC: 5	14	CGDLC: 14	14	CGDLC: 14
UoN	11 (8/3)	MEC: 9 PMLD: 2	31	MEC:28 PMLD:3	29	MEC: 28 PMLD:1
JCYL	11 (10/1)	MEC: 5 ASC: 6	28	MEC: 14 ASC: 14	20	MEC: 12 ASC: 8
AV	7 (3/4)	ITC: 7	15	ITC: 15	15	ITC: 15
SUM	**59 (39/20)**	**MEC: 27** **ASC: 13** **PMLD: 7** **ITC: 7** **CGDLC: 5**	**157**	**MEC: 86** **ASC: 34** **PMLD: 11** **CGDLC: 14** **ITC: 15**	**133**	**MEC: 72** **ASC: 27** **PMLD: 5** **CGDLC: 14**

2.2 Datasets per modality

This section presents details and comments about the datasets collected per each Sensorial Component modality. The reliability, the volume and the adequacy of the data are also discussed. Finally, overall observations and information about the technical quality of the data are presented.

2.2.1 Facial expressions dataset

Regarding the facial expressions modality, the DAT images (sequences in videos) of the users were captured using web and Kinect cameras. Web cameras gathered images of size 640×480, while Kinect images are 1920×1080. An example with the blurred, anonymised face of a participant from the dataset, with its facial landmarks, as detected by the facial expression recognition algorithms, can be seen in Figure 1.

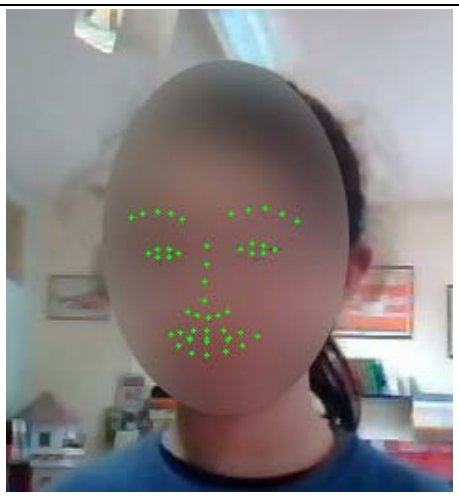

Figure 1: A participant of the MaTHiSiS dataset (face blurred for anonymization), with their facial landmarks mask superimposed

One of the main platform requirements for the facial expression modality was the ability to predict accurately the affect state of the users regardless of the quality of the resolution of the cameras used as input, to support institutions where HD cameras or even Kinects were not available.

Considering this, the use of web cameras, that would also simulate the robot cameras (where applicable, like in the case of NAO robots), were opted as the basic input sensor that should be used for building the corresponding dataset. This approach provided data that can be used to train facial analysis algorithms which can be used for desktop, mobile and robotic PAs, for both resource-abundant and resource-constrained learning environments.

In general, the quality of images collected was very good, except from a few sessions in which low color analysis depth cameras were used, resulting into extremely poor quality images. Another important observation is that, in cases, the web camera was too close to the learner (or vice versa), leading to the fact that the learner's head exceeded the camera's field of view, resulting to unreliable images for facial expression analysis – since the entire face of the subject was not captured. Given this, a new requirement concerning the desired span of the cameras' field of view was set for subsequent recordings in the piloting phases.

A positive and desired specification was that most images show clearly the faces of the learners from different points of view, an important trait for training robust 'in the wild' facial expression algorithms. However, there are many frames, which include the tutor or the teacher or other people apart from the learner, when the requisite is for face-per-image captures. This might lead to the necessity to pre-process the captured data to retain only the useful part of the images (learner interacting with the tool) before ingestion to the algorithm training set. Another limitation is that learners are moving in and out of frame, leading to cropped faces in frames of an image sequence, or with occlusions and sometimes they are not at all in front of the camera's field of view. Finally, there are videos with very low fps (frame per rate), which affects the smooth representation of facial expressions. As far as the annotation is concerned, this dataset involves an adequate set of reliable first and second level annotation.

Enriching the already created dataset with similar data from subsequent data collection in the MaTHiSiS pilots, while communicating the aforementioned limitations to piloting partners and institutions in order to re-enforce the specific set-up requirements (e.g. adequate camera quality, proper positioning of learners in camera's field of view), is expected to improve the performance of the facial expression analysis modules, not only algorithmically, but also practically.

2.2.2 Gaze estimation dataset

Regarding gaze estimation, the data that were collected comprise of the same videos as the facial expressions dataset. The cameras were set quite close to the user's face, which resulted in reliable

images extracted for this task. The dataset has a good variety of users, as it involves data for all cases that the MaTHiSiS project exploits, including recordings with children with autism and PMLD.

The most important part of the images captured, in gaze estimation, is the eyes, along with the head pose, while again the resolution of the images is highly important to this task as well, therefore sharing some of the limitations already mentioned in the previous subsection. In total, the data collected from all three different cameras (web camera, Kinect and NAO) bear adequate resolution in their majority for this modality. Concerning the quality of the videos collected, in most frames we can clearly see the eyes of the user from different angle, an important factor in order to train robust eye gaze estimation algorithms. Generally, the quality of videos was good, except from some observations below.

As aforementioned, in some frames, a second person appears in the frame (besides the user) which, however, does not usually occlude the most important part of the image for gaze estimation: the user's eyes. Another important limitation observed concerned users wearing glasses. This seems to be problematic in some occasions, but only when the glasses reflect the screen or are worn/recorded in such an angle in a way that the glasses hide the eyes. Such data cannot be useful for gaze estimation training, therefore will be excluded from the training set. However, in general, the amount of such cases was deemed not significant to impact the overall usefulness of the collected dataset.

Lastly, some MaTHiSiS use cases include users with under-developed coordination skills (mostly in the PMLDC), which in some occasions results to uncontrolled movements. This, in turn, results in blurry video frames, especially when the frame rate is low. This problem has been observed in a small amount of videos with a low frame rate, making it difficult to distinguish the eyes. Examples of eyes extracted in the dataset, of different gaze orientations, demonstrating aforementioned observations can be seen in Figure 2.

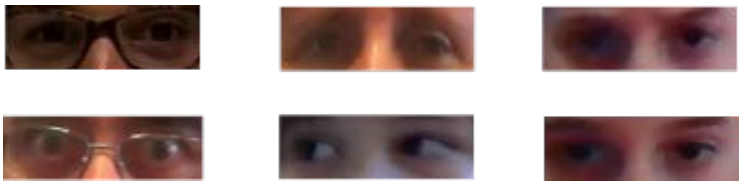

Figure 2: Eyes of users from all MaTHiSiS use cases in the MaTHiSiS dataset

2.2.3 Mobile device-based emotion recognition dataset

After initial analysis of the data collected, referring to inertia sensors in mobile devices, it has been concluded that further data collection will be necessary, in order to train efficient techniques. This is an ongoing procedure and, as stated above, richer training instances will derive from MaTHiSiS pilots. During the DAT sessions, unexpected constraints made the acquisition of inertia data difficult regarding some use cases. In the case of learners with special needs (ASD and PMLD use cases), for instance, it is not always possible for the child to hold the mobile devices. Under these circumstances, some learners used some support system or required the help of a teacher, making the data gathered through inertial sensors useless for the extraction of affect-related features. Furthermore, it is important to highlight that different learning applications were developed for use in mobile devices. In some, the use of the inertial sensors was not ambient but required in order to interact with the learning content (for example, the learner controls the movements of an object in the screen tilting the device accordantly). In such cases, the dataset will be analysed separately per each case in order to define a method that will allow the extraction of affect-related features for both kinds of applications (were inertial sensory expressions are ambient vs when they are obligatory).Moreover, due to technical constraints, some data acquired on the DAT sessions are not appropriate for further analysis due to several difficulties that did not permit synchronization of the mobile data and the annotations provided by experts in most of the sessions.

In the general case, the data was provided following the format provided below:

```
{
   "platform_type": "mobile",
   "sensors_type": (type of sensor, namely gyroscope, accelerometer or 2D touch),
   "data": [
         "timestamp": (unixtime format)
         "features": (features vector, e.g.: [x, y, z]) ]
}
```

Consequently, an important part of subsequent data collection in the pilots will focus on the population of a larger and more adequate training set for the mobile based emotion recognition modality.

For a thorough description of the techniques used for emotion recognition using inertia sensors, please refer to deliverable 4.1.

2.2.4 Skeleton Motion Analysis dataset

Skeleton has been collected though the Kinect v2 cameras. Kinect v2 has the capability to detect human skeletons (silhouettes) and track them, as exemplified in Figure 3. The data were stored in a specific format, described in deliverable D4.1 section 4.4.5, so as to be readable in a post processing step. During captures, we have noticed that the reliability of skeleton tracking depends mostly on the environmental lighting and the actors distance from the camera.

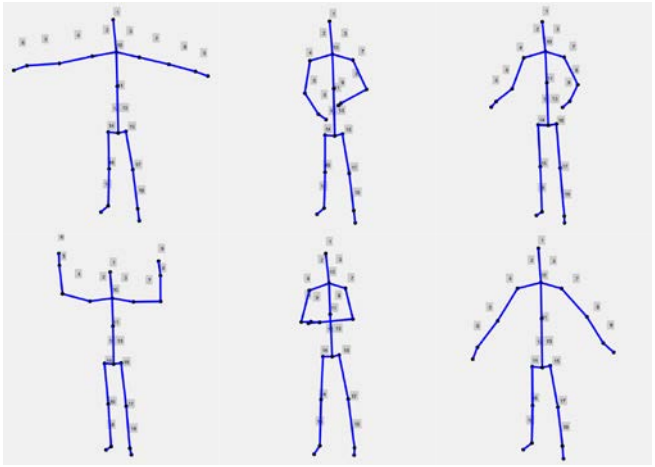

Figure 3: Example of MaTHiSiS dataset skeleton silhouettes

Therefore, reliable skeletons were recorded for captures where the actors are in the distance defined in the DAT set up guidelines (D4.1, Section 4.3.2). Moreover, when it comes to captures conducted on actors sitting in a chair, where not the whole body is visible, the tracked skeleton contains significant noise yielding them not sufficiently reliable.

2.2.5 Speech recognition and speech-based affect recognition dataset

In the case of speech recognition, the clear capture of sound produced by the learners through Kinect's microphone was required, with minimal noise that would not affect the quality of the recorded audio.

In the CGDL case, a lot of good quality recordings were retrieved, but in some cases loud ambient noise (mostly street noise) was recorded, on top of people talking inaudibly. In the MEC, ASC and PMLDC cases in general the audio quality is good in all premises, although in there are cases where

more than one people were talking, with their audio overlapping. In some sites, a few empty (noise only) records were observed. The only questionable audio was retrieved in the IT use case recordings, where many recordings are empty or contain only noise. Further pre-processing will aim in alleviating such data of low fidelity.

Overall, the produced data and corresponding annotations are a good starting point for training with the MaTHiSiS different use cases, while the continuous augmentation of the dataset with subsequent stringer reinforcement of set up requirements (minimize ambient noise, avoid overlapping speech) will improve the speech-based affect recognition algorithms' performance for the MaTHiSiS settings.

2.3 Observations

Overall, the DAT worked as expected and the anticipated set of data was acquired during the data acquisition sessions. During the process however, particularly interesting observations arose from the difficulties met and the reaction of the learners towards the content presented to them and the interaction with technological agents, yielding noteworthy technical, scientific and pedagogical insights.

Initially, to sum up the errata reported in the previous chapter:

- In most ASC and PMLD sessions, learners had to be assisted by the tutor to complete necessary tasks, resulting to the tutors being frequently included in the images gathered from cameras, their voice being captured form Kinect microphones on top of learners' voices or the tutors using the mobile device before they gave it to the learners.
- Some of ASC learners were moving continuously during the session, which resulted to them moving out of the camera field of view. Some also placed their hands on their face or turned to look to their tutors/teachers. Under these circumstances, data for facial and gaze algorithms are not suitable for training.
- In some cases, the tutors/teachers had to explain to the learner the task/game, before each session or in some stages that the learner does not understand exactly what they are asked to do. Consequently, during this period, the affect and behavior of the learner was not natural and correlated to the task at hand.
- Kinect cameras were occasionally set too close to the learner or the learner was sitting and thus, no skeletons could be detected by Kinect.
- The position of the mobile device was another factor that had impact on the data gathered. Some learners placed the device on a table disabling, in essence, the mobile sensors (gyroscope and accelerometer).

All these issues produced some anticipated amount of noise in the collected data that may require appropriate pre-processing of collected datasets before being used for algorithm training. Potential mitigation strategy includes correlation with manually annotated portions of the data, even for algorithms that do not directly use the expert annotations in their training, but just to verify which is the useful part of the collected data.

Other difficulties included:

- Due to different capturing mechanisms of the learner recording and the screen recording (the latter was used as reference in the DAT's manual annotation tool, as described in D4.1), videos presented to annotators were "muted", which made the annotation process a little difficult. Therefore, the tutors' personal experience with their students played a pivotal role in the subsequent second level annotations derived in the dataset. This brought the tutors' innate knowledge of the learners to the foreground.
- Few students, especially in the PMLDC, did not find all the games appealing enough (not enough rewards/prompts or too easy activity) and would quit the recording before the completion of the sessions. This provided useful information about the development of Learning Material for the actual platform, to be used in the MaTHiSiS pilots.

- Few technical issues (system crashes) rendered some sessions unusable for annotation, since they were not fully completed. However, learners were recalled to complete the pre-defined number of sessions in most cases, which resulted in the desired first round of data. This point is mostly brought forward to identify why some sessions reported in Table 2 appear not to be annotated.

Concerning the differences in the different MaTHiSiS use cases, on ASC and PMLD learners' data, continuing movements and physical obstacles are noticed in contrast to other cases.

On the positive side, some indicators of the impact of the MaTHiSiS gamified learning activities approach produced interesting insights:

- ASD children's teachers/educators reported a positively unexpected participation and attention maintenance during the DAT, even if one of the children did not complete all the sessions. About this last case, according to a teacher: "10 minutes of attention and collaboration is an important achievement" for him.
- All of the students in the FMD sites, both ASD and ME, appreciated in particular the "faces game", used for gathering first level annotations for the facial expressions analysis modality, probably because the requests ("make this face", "express this emotion with your body") apparently come from a system's will to communicate with the child. Many of the children, in particular ASD, engaged a direct interaction with the "machine" during this game and maintained high attention and interest.
- In the PE sites, the pupils liked to play with the mobile and the face game was very engaging for them. For the interaction game, a bit too complicated questions were chosen, but after the 4 repetitions (students were asked to conduct 4 rounds of this game each each) the majority of the children were still engaged.

All these observations provided most valuable, unexpected, first insights about the value of gamified and highly tech-supported learning experiences in the retention of learners engagement to the learning process and performance of the students. It also, again, produced important information to be considered in the design of the Learning Materials to be used in the pilots.

3. Sensorial component modalities

This chapter details the progress and final state of the Sensorial Component modalities' algorithms since the first report in D4.1. In some cases, the implemented tools did not have significant algorithmic changes to report, but rather focused in the in vitro testing of the methods.

3.1 Facial expressions analysis

3.1.1 MaTHiSiS Facial expression extraction in the wild - Graph Based Method

The graph based method's algorithm exploits the ability of face to be represented as a graph, as detailed in Deliverable D4.1 Section 2.1.2. As was described there, a facial landmark algorithm locates specific areas of the faces using points of interest that are used also to create a graph. Different emotions invoke different movements of facial muscles and, thus, different positions of the points on the image yielding dissimilar graphs during an image sequence. Using this variation of the graphs, this algorithm predicts the different emotions per certain timeframes. The main steps of this algorithm are the face detection in the input image, the landmark (point of interest) detection, the feature extraction and finally, the facial expression classification. More details about this algorithm are presented in D4.1.

The initial evaluation of the algorithm is done using images from the well-known public available Cohn-Kanade (CK) database [2] leading to very satisfying results (cf. Deliverable D4.1). Although this dataset involves expressions of six basic Ekmanian emotions [3], which are Anger, Disgust, Fear, Happiness, Sadness and Surprise, a correlation of the aforementioned emotions with affective states was retrieved in Russell's Core Affect Framework [4]. A direct mapping of the spontaneous emotions to affect states conveys this correlation. Using this mapping, Sadness corresponds to Boredom, Happiness to Engagement and Surprise, Anger, Fear to Frustration. The performance of this algorithm using Cohn-Kanade dataset to predict affective stated reached a classification score that rounds up close to 100% accuracy (cf. Deliverable D4.1).

The algorithm was also evaluated using the MaTHiSiS dataset collected from the data acquisition phase (cf. Section 2). This allowed for a first overview of the effectiveness of the MaTHiSiS solution over real-world data in real, diverse use cases. The performance of the graph-based algorithm was assessed on three axes of MaTHiSiS cases, namely children with disabilities, children without disabilities and adults. Table 3 demonstrates the results.

Table 3: Performance of facial expression graph based method on the MaTHiSiS dataset

Case	Affective states recognition accuracy
Children without disabilities (MEC)	80.56%
Children with disabilities (ASC, PMLDC)	68.36%
Adults (ITC, CGDLC)	61%

The methodology followed to evaluate the algorithm with the MaTHiSiS data was five-fold cross-validation. To this end, the sample for each (super-)case was partitioned randomly to five equally-sized subsamples. The classifier was trained using four partitions, i.e. 80% of the sample, with the last partition (20%) comprising the training set. This process was repeated five times per (super-)case, to remove bias and randomness, and the average of all five evaluation rounds per (super-)case is presented in Table 3.

The result indicates good initial results for the graph-based approach when using the MaTHiSiS data and highlights the differences between use cases, which was the reason why the need for creating the MaTHiSiS dataset arose. For instance, the MEC children scored higher than any other case, since children are highly expressive in the exhibition of emotions, more so than adults. Children with special abilities on the other hand have an expected variance in the manifestation of their emotions and affect states, due to their overall physical and behavioural expression capacities, resulting to lower recognition accuracy than MEC children, as expected. In the future, with more data collected to train the algorithm, during the pilots, for all uses cases, further improvement of the algorithm's performance is expected, as is the case for all deep learning approaches, where training data availability heavily boosts the prediction outcome.

3.1.2 MaTHiSiS facial expression extraction in the wild - Appearance Based Method

In order to enrich the already used public dataset with data from different profiles, UM performed, in parallel to the DAT sessions, several experiments with volunteers in its institution (Maastricht University, the Netherlands). These experiments were performed by 33 students (16 males and 17 females). The lighting conditions during the capturing procedure were typical for an office environment and a web camera was used for capturing facial and upper torso expressivity.

This dataset contains visual data, a register of interactions and personal information of students playing the "e^{3}learning platform", a serious game based on question and answer quizzes, that was also used in the DAT. These quizzes contain questions from four different subjects (namely mathematics, geography, history and sports) with nine different levels of difficulty per subject. This serious game was also included as part of the Data Acquisition Tool and a description of it was included in [5]. The volunteers were bachelor and master students (with a mean age of 22.6 years and a standard deviation of 3.13 years), from two different knowledge profiles, namely software engineering and psychology. Each student performed four learning sessions in each of the four available subjects. After every learning session, the users were asked to assess, on a Likert scale (from zero to five), the degree of engagement, frustration and boredom that they experienced. This dataset will become soon publicly available for research purposes.

The dataset architecture was inspired from the works presented in [6] and [7]. The detailed description for the data gathering process is the following:

- A consent form was given to every participant. This form contains all the information needed, for the students to understand the reason behind the experiment and the gathering of the data. The content of this consent form was reviewed and approved by the Ethical Review Committee of Maastricht University.
- Additionally, oral instructions were given to the participants with information about the process that they had to follow.
- The duration for the whole experiment per each participant was 26±5 minutes.
- In order to facilitate the experiments' performance and avoid users feeling tired (thus, biasing acquired data), it was decided to establish a restriction per subject so that the learner only performed tests belonging to four out of nine levels of difficulty. However, to record data belonging to all the levels without exceeding the time constraint (around 30 minutes per each participant to perform the whole experimental phase), the approach implemented was dependent on the score achieved in the previous session. At the beginning, the first level

was established randomly between levels one and three (out of nine). Subsequently, the increase of the level was based on the score obtained by the user. In this way, the user was asked to answer questions related to levels four or five, six or seven and eight or nine, depending on whether or not the learner's score would reach a certain threshold. The whole procedure led to sparse matrices of learners and difficulty levels which are suitable for the performed study done and described in the experimental results.

As it is mentioned above, during the learning sessions, learners' visual information was recorded. This data was stored locally (in the same path with the "e^{3}learning platform") in ".mp4" format. The average duration per each video in the dataset is 45±12 seconds and every video clip is corresponding to each learner session. The names of those video files were constructed using the username of the learners and the corresponding session identify. In total, we obtained 528 videos belonging to different subjects, difficulty levels and leaner from different knowledge background.

In Figure 4 some captures of the participants obtained during the experiments are shown. These specific frames were extracted during the interaction with the platform and, more specifically, during the answer button event. In all of these cases, the answers of the students were wrong apart from the screenshot 4 (d) in which the learner gave the correct answer.

Figure 4: Screenshots of four learners while interacting with the "e^{3}learning platform".

(a) Screenshot 1

(b) Screenshot 2

(c) Screenshot 3

(d) Screenshot 4

Table 4 depicts the statistics about the two different populations and the total dataset (mean and standard deviation) for the learners' self-annotation during the learning sessions. It is important to notice that students tended to annotate their sessions as engagement rather than the other two affective states (boredom or frustration). This fact resulted in an unbalanced dataset. However, due to the nature of the experiments, we expect that this situation will be similar to the one experienced in an operating (MaTHiSiS) environment. This dataset will be used to enrich the dataset provided by the DAT session in connection with visual data but also interaction parameters and also multimodal fusion to be analysed as part of tasks 4.2 and 4.3.

Table 4: Mean and standard deviation for learners affect self-assessments (boredom, engagement and frustration in the scale of 0-5) during the learning sessions.

Subject	Engineering			Psychology			Total		
	Boredom Mean(std)	Engag. Mean(std)	Frustrat. Mean(std)	Boredom Mean(std)	Engag. Mean(std)	Frustrat. Mean(std)	Boredom Mean(std)	Engag. Mean(std)	Frustrat. Mean(std)
Mathematics	0.88(0.94)	3.00(1.44)	2.03(1.45)	1.02(0.88)	3.41(1.29)	2.58(1.25)	0.93(0.92)	3.13(1.41)	2.20(1.42)
Sport	1.43(1.21)	2.28(1.37)	1.83(1.49)	1.52(1.35)	2.70(1.45)	2.10(1.31)	1.46(1.25)	2.40(1.40)	1.91(1.44)
Geography	1.11(1.15)	2.49(1.46)	1.64(1.42)	1.26(1.19)	3.23(1.42)	2.19(1.61)	1.15(1.16)	2.70(1.49)	1.80(1.50)
History	1.22(1.27)	2.53(1.48)	1.91(1.45)	1.00(0.97)	2.97(1.44)	2.00(1.46)	1.16(1.20)	2.65(1.48)	1.94(1.45)

3.2 Gaze estimation

Since Deliverable D4.1, further improvements and testing has been conducted on the proposed regression CNN-based gaze estimation algorithm, and another approach of two stream CNN that takes into account the temporal attributes of the users eye gaze sequence was investigated in order to opt for the most suitable solution for the particularities and vast diversities of the MaTHiSiS use cases. The regression-based method was used to estimate learners' engagement based on their eye gaze in the MaTHiSiS driver pilots, while the new method will be evaluated both in vitro with public datasets and, more interestingly in vivo in subsequent piloting phases.

3.2.1 Eye gaze estimation based on regression CNN

As described in *D4.1 MaTHiSiS Sensorial Component*, a regression CNN was trained using the MPIIGaze [8] dataset to estimate a person's gaze direction for each of their eyes. In that first version of the gaze estimation methodology the input pipeline utilized a state-of-the-art frontalization step [5] in order to reduce the appearance-based head pose effect on the estimation and learn pose-invariant dense visual features, while modelling the head pose as a separate feature vector. However, face frontalization was a relatively costly operation processing-wise compared to the other steps of the pipeline (face detection, landmark detection & head pose estimation). Therefore, a perspective warping step was opted for instead of the face frontilization, that was faster processing-wise and also achieved higher accuracy after training. More specifically, virtual cameras were positioned at a fixed distance in front of each eye and rotated by the estimated head pose. Then, based on the fixed camera parameters (width, height, focal length, principal point), the original image was warped on the virtual face image, producing normalized versions of each eye, which are in turn histogram-equalized.

The processing pipeline follows, with the main difference from the pipeline presented in D4.1 being the data normalisation step modification:

1. **Face and landmark detection** is already described in D4.1. For face detection Li et al.'s SURF cascade method [9] is employed while for the landmark detection, a cascade of boosted regression forests [10] regresses the positions of the facial landmarks in around 1-2 milliseconds.
2. **Head pose estimation,** by utilising the 2D detected landmarks, correspondences with pre-annotated 3D positions can be now established. These are annotated on a generic 3D mean facial shape head model and are used to estimate the user's head pose by fitting [11] the 3D model data to the 2D image correspondences via non-linear optimization. The result is the

head's pose (rotation & translation) with respect to the coordinate system defined by the camera.

3. **Data Normalisation,** where the generic 3D mean facial shape is rotated and translated according to the head pose extracted previously. Then the 3D eye position of the left and right eye is estimated, and a vector from each eye looking at the target is predicted. Moreover, similar to [8], the normalisation is done by scaling and rotating the camera-captured image so that the eye image is centered at the midpoint of the eye corners from a fixed distance d and so that the horizontal axes of the head coordinate system and the camera's coordinate system are aligned. The head pose is then parameterised as a 2D angle and the eye images are contrast-enhanced.
4. **Gaze Estimation,** the vector of each eye along with the respective image are fed into a pre-trained deep CNN which then regresses a feature vector representing the user's gaze direction.

After training the network and achieving similar accuracy to that of [1] by using a subset (15%, randomly chosen) of the dataset as a test set, it was also tested on another publicly available dataset, EYEDIAP [12]. The EYEDIAP dataset contains low resolution (VGA) video capturing 94 video sequences of 16 participants looking at three different targets (discrete and continuous markers displayed on a screen, and floating physical targets) under both static and free head motion, while it includes two different illumination conditions on some participant recordings. Moreover, along with the database, the authors include a framework of calculating performance measures, with respect to each visual target (discrete, continuous or floating).

The gaze estimation network was tested on multiple sequences from EYEDIAP and achieved a mean angular error on the VGA sequences of **10.46** degrees with high head mobility, and **9.55** degrees with no head mobility. On the contrary, on the challenging non-frontal viewpoint and the HD camera the mean angular error achieved was **18.22** degrees with high mobility and **17.43** degrees with none. In comparison, the method in [8] that outperforms other state of the art methods achieves an accuracy of 13.9 degrees on the MPII dataset and 10.5 degrees on the Eyediap dataset. These results are illustrated in more detail in Figure 5, Figure 6, Figure 7 and Figure 8.

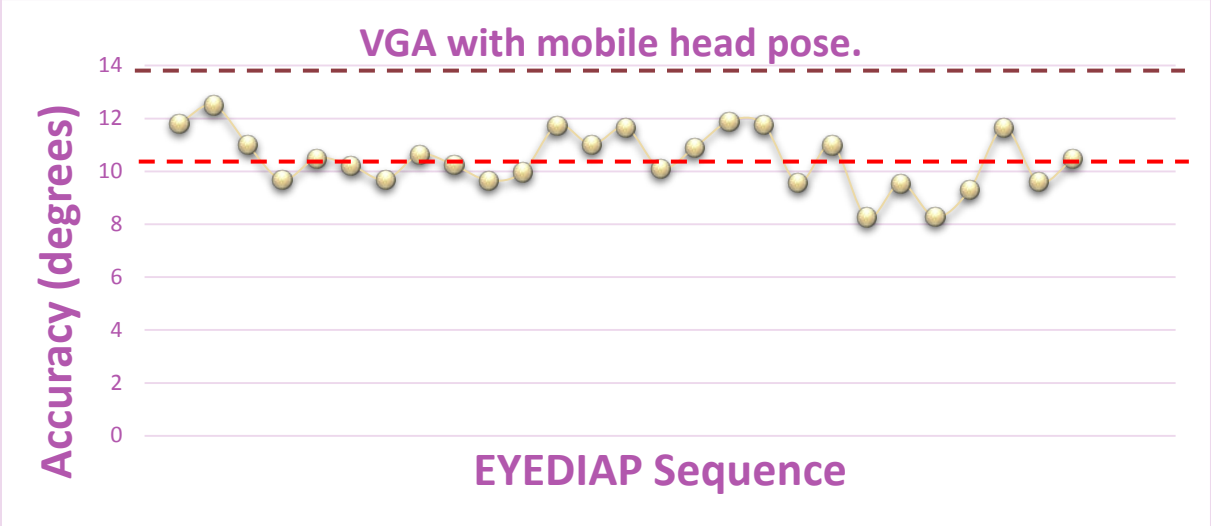

Figure 5: **EYEDIAP results for the VGA camera and high head mobility, in cross-reference with the results presented in [1] for the MPIIGaze dataset (green) and for the Eyediap dataset (red).**

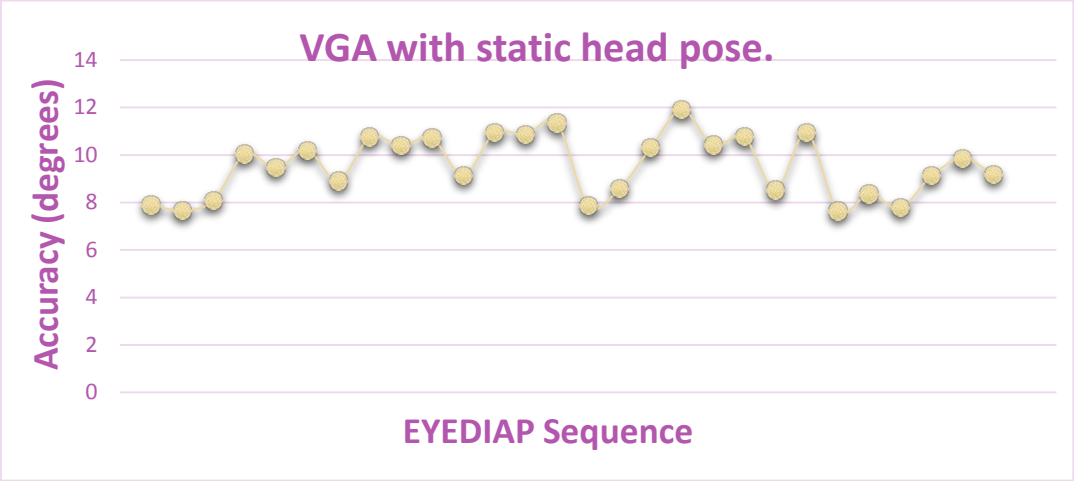

Figure 6: **EYEDIAP results for the VGA camera and static head activity.**

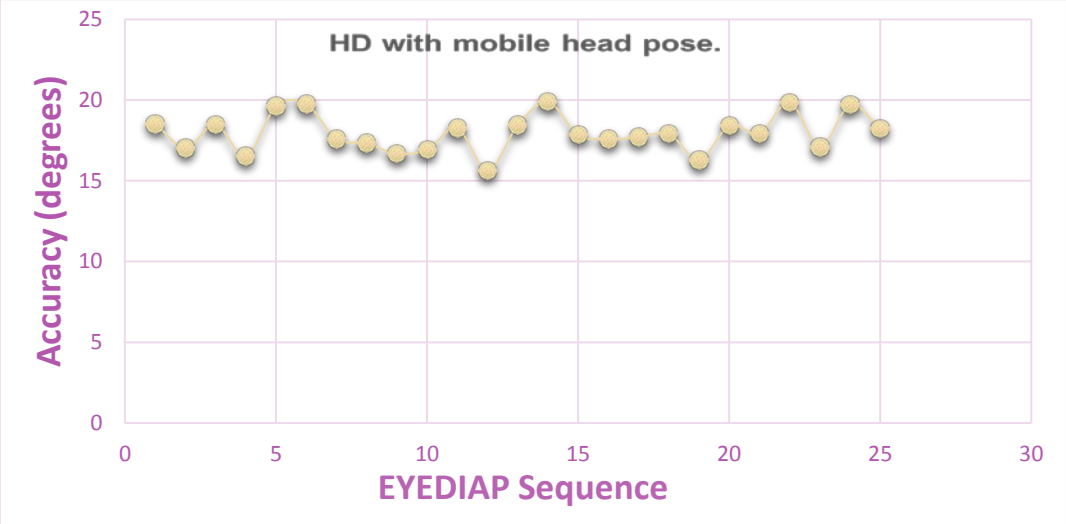

Figure 7: **EYEDIAP results for the HD camera and high head mobility.**

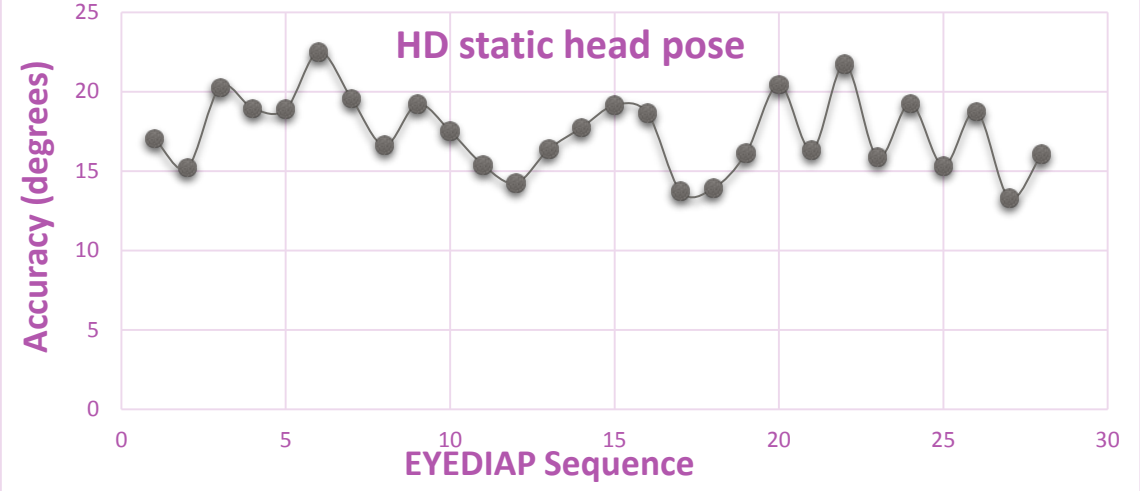

Figure 8: **EYEDIAP results for the HD camera and static head activity**

Apart from the CNN regression based approach, a Conditional Local Neural Fields (CLNF) approach was also investigated. In OpenFace [13] for example, a recent state-of-the-art approach which uses CLNFs, a complete facial behaviour framework is presented that jointly estimates facial landmark positions, head pose, facial expression and eye gaze. The recently proposed CLNFs [14] are used to locate and track the facial landmarks with a significantly good performance due to the fact that they also capture landmark shape variations (i.e. deformable landmarks). Due to the generality of the framework, they can be used in a variety of computer vision fields, including training an eye landmark detector and tracker.

After testing with the datasets previously used in the project on the OpenFace method, it produced state of the art results and was also found to be highly performant, running in real time rates. More specifically, OpenFace was evaluated on the same EYEDIAP dataset and achieved a mean angular error on VGA sequences of **11.17** degrees (detailed illustration for all sequences in Figure 9) with mobile head pose activity, therefore underperforming on the particular condition over the previously described methodology for the particular setting (VGA images). On the HD sequences the mean angular error achieved was **15** degrees (detailed illustration for all sequences in Figure 10) with mobile head pose activity, therefore outperforming the previous method in this setting (HD images). The lower performance in VGA images can be attributed to their low resolution and thus, the lower quality of the eye images that prevent the accurate localisation of the eye landmarks. OpenFace.

Therefore, it can be deducted that the initially implemented method is more appropriate for VGA images (e.g. a standard non-HD web camera or a NAO robot's camera) while the OpenFace method would be better served in HD resolutions (e.g. Kinect). Because of the requirements of MaTHiSiS for reliable gaze estimation in a variety of settings were HD cameras may not be available, the initial CNN regression-based method was opted and employed in the driver pilots.

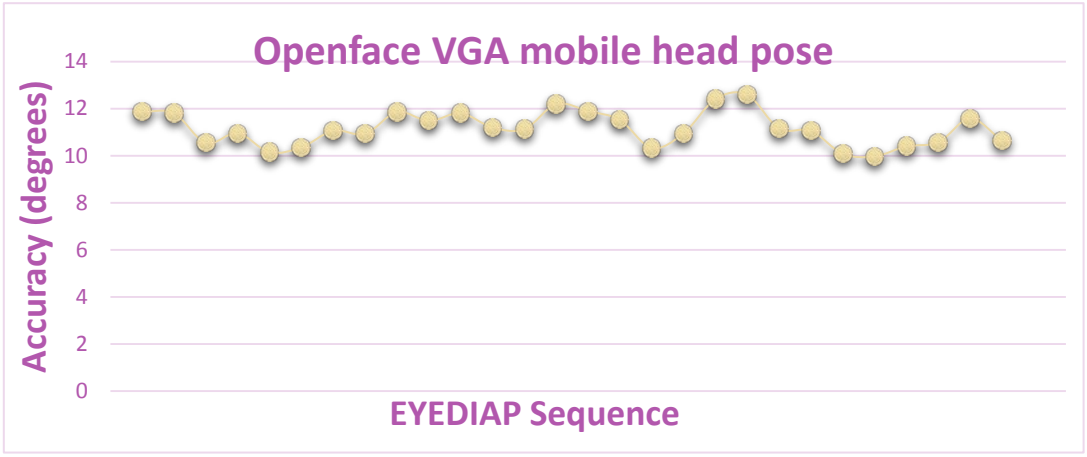

Figure 9: **Results of Openface on EYEDIAP for the VGA camera and high head mobility.**

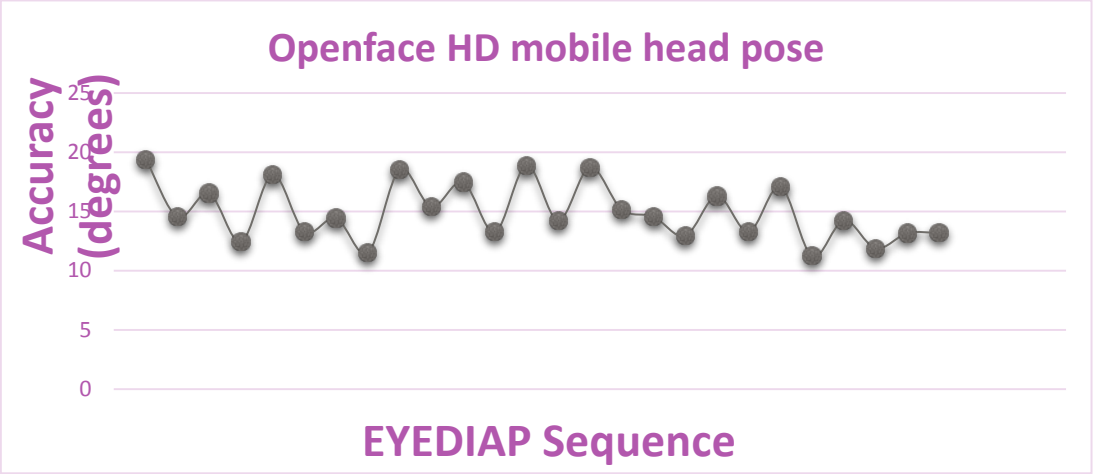

Figure 10: **Results of Openface on EYEDIAP for the HD camera and high head mobility.**

3.2.2 Eye gaze estimation based on two stream CNN

Eye gaze estimation is a very challenging task, especially when taking into account the special specifications of the MaTHiSiS project. Due to the fact that, in MaTHiSiS project, there will be a wide range of end users, including children with autism and profound and multiple learning disabilities, gaze should not only be estimated in different illumination conditions, but also without relying on the head pose from the users. The aforementioned limitations led to the selection of the EYEDIAP dataset for the initial training and testing of an alternative approach, which takes into account not only the spatial attributes that relate to users' gaze, but also the temporal conditions.

In the proposed method, Convolutional Neural Networks (CNN) were adopted in order to estimate eye gaze. More specifically, the method that was developed is based on the notion of the two-stream CNN as described in [15]. In [15], the authors proposed a two-stream convolutional network architecture, which incorporates spatial and temporal networks. In their network, the spatial stream operates on individual video frames, performing action recognition from still images, while for the temporal stream dense optical flow is computed between two consecutive images. The input to the temporal network is formed by stacking optical flow displacement fields, forming a volume of data.

The aim of the authors in [15] is to perform action recognition in videos, while our goal is to estimate eye gaze. Although goals might seem different, we should note that dense optical flow has been used before for eye gaze estimation [16], [17], [18] thus we could make use of the temporal stream in order to infuse to the model information about the evolution of the eyes movement in time, which might be important to the algorithm. The overall goal is to capture the complementary information on appearance from still frames and motion between frames.

The architecture of spatial and temporal streams, proposed in [15], was altered in order to meet the project's needs. Moreover, the EYEDIAP dataset was preprocessed, so that only the useful information (i.e. eyes) of the video frames are obtained, while dense optical flow was calculated between sequences of still images.

The outline of our method steps are:

1. Using the EYEDIAP database, bounding boxes from each frame of the video were extracted, containing the subject's eyes.
2. All images that contain anything else besides the eyes of the subject (e.g. is some cases only the 3D target were visible) were discarded - not more than 10% of very specific recordings.

3. The optical flow between the pre-processed images was calculate using the algorithm proposed in [19].
4. A two-stream CNN according to the architecture of the network of [15] was constructed, with minor adaptions of the model.
5. Spatial stream: the stream was trained using the cropped and rescaled images of eyes.
6. Temporal stream: the stream was trained using the optical flow data, exported from cropped eyes images.
7. Late fusion of spatial and temporal stream was applied, resulting in a 3D eye gaze vector (regression).
8. The algorithm's performance is on process of evaluation through calculation of the mean-squared error of the recognized gaze vectors and the suitable protocols proposed in [12].

In more detail, bounding boxes of the eyes were obtained using the data provided along with EYEDIAP. For every frame of a video in EYEYDIAP database, a bounding box that contains the eyes is located and cropped. Then, the images produced from the latter procedure are rescaled to a mean bounding box and the resulting images are used as input to the spatial stream.

In Figure 11 an example of the cropped and rescaled images of the EYEDIAP database is illustrated. These images are of 20 consecutive frames of a single video session. Notice that despite the rescale, there is almost no distortion in the images.

The input for the temporal stream is composed by obtaining the optical flow between consecutive images of the pre-prossesed data, using the method described in [19].

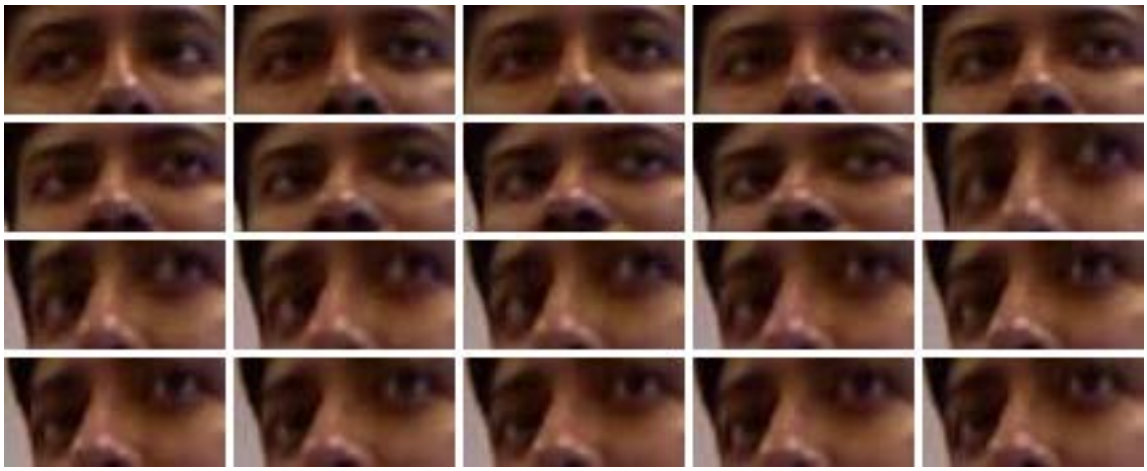

Figure 11: Example of eye images, from pre-processed EYEDIAP database, used in spatial training

The problem that the two-stream CNN needs to solve, is a regression problem, since the model needs to predict a 3D vector in space, representing where the user is looking, and from this information to decide if they are engaged or not. Keeping that in mind, some key specifications of the CNN model include:

- Spatial and Temporal stream models have the same architecture.
- The algorithm uses the rectified linear unit (ReLU) as an activation function between all layers, except when Dense layers are applied (2 last layers and output)
- Not all loss functions are suitable for the regression task, as some of them are meaningless for this problem. The loss function that is used is mean-squared-error (MSE).
- Since we need a continuous valued vector as an output, no softmax function was used in the last layer.
- Spatial and Temporal streams are combined via late fusion, allowing multi-task training.

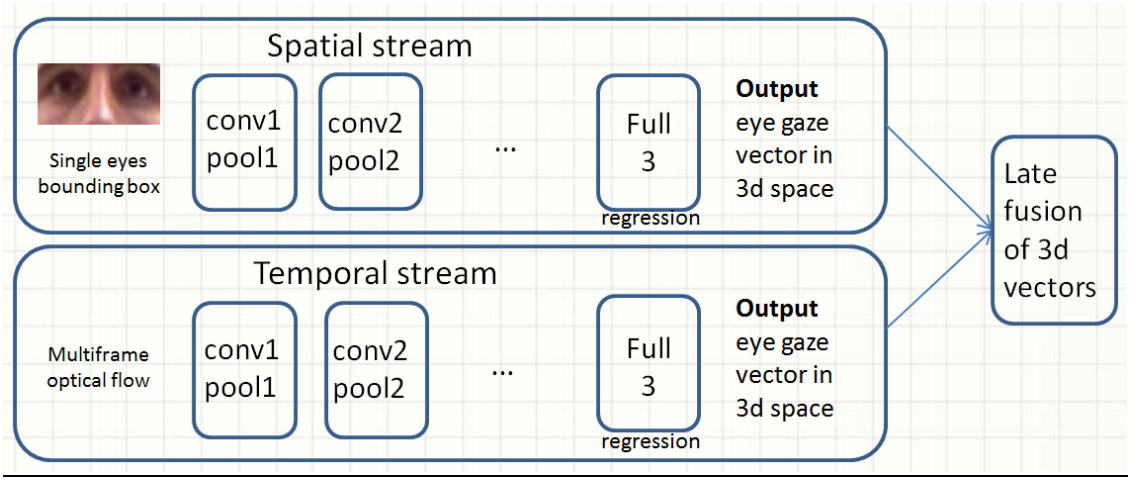

Figure 12: CNN architecture for eye gaze estimation

3.3 Mobile device-based emotion recognition

The main algorithmic implementation of the mobile-based affect recognition has already been finalised in the previous months and described in Deliverable D4.1. The major objective for this task is training and testing on a reliable dataset, also imminent due to the lack of publically available datasets. Currently, the implemented algorithms are being tested on internally developed data and a final evaluation will be provided on pilot data, given that, as mentioned in 2.2.3, DAT-acquired data need to be extended in this case.

Smartphones and tablets have become in some of the devices most used in every day living. New generations grow in a digital world where these devices are the access point to several activities such as entertainment, information seeking or, increasingly, learning activities in education centres. However, the analysis of the interactions with these devices is an area of study which is not yet as explored in that respect. Recently, the use of sensors included in this kind of devices such as accelerometers and gyroscopes have proved to be useful for recognizing users' emotional states [20], [21], [22], [23].

Inspired by the research performed by Coutrix [20], MaTHiSiS implements an affect recognition system which exploits the expression through 2D and 3D gesture using aforementioned sensors. As demonstrated in [20], these 2D but especially 3D descriptors contribute to emotion expression while interacting and using mobile devices. In this work, it was detected a high number of significant correlations in 3D motion descriptors of gestures and the arousal dimension.

In MaTHiSiS, we expand this study in order to infer affective states which are commonly experienced during the learning process, focusing our effort in the recognition of engagement, boredom and frustration. This three-dimensional and continuous space can be accurately mapped to affective states from the theory of flow. The features extracted are analysed in order to detect common patterns which can allow the system to infer the affect state of the leaners while interacting with the MaTHiSiS platform using a mobile device. Ideally, these features will help in the identification of erratic movements or unexpected behaviours such as the lack of motion or interactions with the devices. This information could denote frustration or boredom respectively and, through this analysis, we will study the usability of it in learning environments. For more details, please refer to Deliverable 4.1 [5].

3.4 Skeleton Motion Analysis

Human action recognition became a necessity for applications in surveillance, human robot interaction, robot perception, etc. A variety of methods has been introduced, each one applying a different feature extraction and classification methodology. In the latter decades, handmade features have dominated in the field of action recognition. Those features were carefully selected in a way such that to interpret the performing action. However, more recently, based on the deep learning approaches, automatically generated features were introduced [24]. Through a deep learning framework and a large corpus of data samples, deep learning methods have provided us with very promising results, overcoming the handmade methodologies. We also compare our method with the state of the art algorithm that uses neural networks in the field of action recognition [24]. Even though the neural network based action recognition method shows very promising results, our method outperform the reported results. Moreover, we compare our results to other state of the art techniques and with very promising results.

3.4.1 Approach

Working with skeleton data each action is a sequence of moving skeleton frames where each skeleton consists of a number of N joints-nodes connected with edges. Based on this point of view we give the following definitions.

Basic definitions on spatial domain: We define a skeleton action as a time varying graph, $\mathrm{G}^{(\mathrm{t})} = (\mathrm{V}^{(\mathrm{t})}, \mathrm{E})$, $\mathrm{t} \in [0, \mathrm{T}]$, where $\mathrm{V}^{(\mathrm{t})}$ represent the skeleton nodes at a specific time point and E the skeleton edges. More specifically, we define a skeleton frame with a matrix $\boldsymbol{\Phi}^{(\mathrm{t})} = \left[\boldsymbol{\phi}_1^{(\mathrm{t})}, \boldsymbol{\phi}_2^{(\mathrm{t})}, \dots, \boldsymbol{\phi}_N^{(\mathrm{t})}\right] \in \mathrm{R}^{I \times N}$ with each column to represent a skeleton node $\boldsymbol{\phi}_n^{(t)} \in \mathrm{V}^{(\mathrm{t})}$ where $\boldsymbol{\phi}_n^{(t)} \equiv \boldsymbol{\phi}(\boldsymbol{x}_n^{(t)})$ and $\boldsymbol{\phi}(\cdot)$ a vector value function that maps $x^{(t)}$ into the feature space. When $\boldsymbol{\phi}(\cdot)$ is the identity function then $\boldsymbol{\phi}_n^{(t)} = \boldsymbol{\phi}\left(\boldsymbol{x}_n^{(t)}\right) = \boldsymbol{x}_n^{(t)}$ where $\boldsymbol{x}_n^{(t)} = \left[x_n^{(t)}, y_n^{(t)}, z_n^{(t)}\right]^T$ is a three dimensional vector ($I = 3$) that represent its position in three-dimensional space. Given, the definition of the time varying graph $G^{(t)}$, in a specific time point t, we also define the corresponding time varying degree, weight and Laplacian matrix as $\boldsymbol{D}^{(\boldsymbol{t})}$, $\boldsymbol{\Gamma}^{(\boldsymbol{t})}$ and $\mathbf{L}^{(\boldsymbol{t})} = \mathbf{D}^{(\boldsymbol{t})} - \boldsymbol{\Gamma}^{(\boldsymbol{t})}$ respectively with $\mathbf{D}^{(\boldsymbol{t})} = \sum_i \Gamma_{ii}$. Given the skeleton nodes $\left\{\boldsymbol{\phi}_{\boldsymbol{n}}^{(\boldsymbol{t})}\right\}_{n=1}^{N}$ in a specific time point t the elements $\boldsymbol{\Gamma}_{\boldsymbol{ij}}^{(\boldsymbol{t})}$ of the weight matrix are given by $\boldsymbol{\Gamma}_{\boldsymbol{ij}}^{(\boldsymbol{t})} = \exp\left(\frac{\|\boldsymbol{\phi_i} - \boldsymbol{\phi_j}\|}{\sigma_i \sigma_j}\right)$). This means that the final weight matrix contains spatial information about the human skeleton structure. Moreover, we denote the complete skeleton action with extra bold matrix $\boldsymbol{\Phi} = \left[\boldsymbol{\Phi}^{(1)}, \boldsymbol{\Phi}^{(1)}, \dots, \boldsymbol{\Phi}^{(N)}\right] \in R^{I \times NT}$. Similarly, we define $\boldsymbol{L} = \left[\mathbf{L}^{(1)}, \mathbf{L}^{(1)}, \dots, \mathbf{L}^{(N)}\right] \in R^{N \times NT}$ and the block diagonal matrix:

$$diag(\boldsymbol{L}) = \begin{bmatrix} \boldsymbol{L^{(1)}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{L^{(T)}} \end{bmatrix} \in R^{NT \times NT}$$

where $\boldsymbol{L}^{(\boldsymbol{t})}$ is the Laplacian matrix of the graph skeleton in time t.

Basic definitions on speed domain: The speed vector of each skeleton node $\boldsymbol{\phi}_n^{(t)}$ is given by $\dot{\boldsymbol{\phi}}_n^{(t)} = \boldsymbol{\phi}_n^{(t)} - \boldsymbol{\phi}_n^{(t+1)}$. Using the speed vectors, we can apply the same computations like in spatial domain to obtain now in speed domain the corresponding Laplacian $\dot{\boldsymbol{L}}^{(t)}$, weight $\dot{\boldsymbol{\Gamma}}^{(t)}$ and degree matrix $\dot{\boldsymbol{D}}^{(t)}$ the time varying graph $G^{(t)}$.

Main-idea: Given a set of skeleton nodes, we want to find a mapping such that the new node representations will preserve their speed relation. In other words, if two nodes that are "close" in speed in the input-space, they should also be "close" in the new feature-space.

3.4.2 Slow Feature Analysis (SFA)

Given an I dimensional input signal $\boldsymbol{\phi}^{(t)} = \left[\phi_1^{(t)}, \dots, \phi_N^{(t)}\right] \in R^I$, with $t \in [0,1]$ indicate time, Slow Feature Analysis (SFA), computes an input output vector-valued function $\boldsymbol{g}(\boldsymbol{\phi}^{(t)}) = \left[g_1(\boldsymbol{\phi}^{(t)}), \dots, g_N(\boldsymbol{\phi}^{(t)})\right]^T$ such that to obtain a final output signal $\boldsymbol{y}^{(t)} = \left[y_1^{(t)}, \dots, y_J^{(t)}\right] \in R^J$ where $y_j^{(t)} = g_j(\boldsymbol{x}^{(t)})$. The g_j function in linear case is defined as $g_j(\boldsymbol{\phi}^{(t)}) = \boldsymbol{w}_j^T \boldsymbol{\phi}^{(t)}$ and for all g_j, $\boldsymbol{y}^{(t)} = \boldsymbol{g}(\boldsymbol{\phi}^{(t)}) = \boldsymbol{W}^T \boldsymbol{x}^{(t)}$ where $\boldsymbol{W} = [\boldsymbol{w}_1, \dots \boldsymbol{w}_J]$ and $\boldsymbol{W} \in R^{I \times J}$ [25]. This definition leads to a closed form solution. Thus, SFA tries finds a transformation matrix $\boldsymbol{W}$ to achieve the following $\boldsymbol{\phi}^{(t)} \xrightarrow{\boldsymbol{W}} \boldsymbol{y}^{(t)}$. To obtain the output signals SFA solves the following optimization problem:

$$\min_{g_i} \mathbb{E}_t[\dot{g}_j^2(\boldsymbol{\phi}^{(t)})]$$
$$s.t.\ \mathbb{E}_t[g_j(\boldsymbol{\phi}^{(t)})] = 0, \mathbb{E}_t[g_j^2(\boldsymbol{\phi}^{(t)})] = 1$$
$$\mathbb{E}_t[g_j(\boldsymbol{\phi}^{(t)}) g_i(\boldsymbol{\phi}^{(t)})] = 0$$
$$\forall j \neq i, i = 1, \dots., J$$

where $\dot{g}_j$ denotes the first order time derivative of the output function g_j and $\mathbb{E}_t[\cdot]$ indicates time averaging. The objective function above tries to minimize the variance of the input-output function g_j and this is done by minimizing the derivative of the function over time. It can be proved that the above optimization problem is equivalent with:

$$\min_{\boldsymbol{W}} trace(\boldsymbol{W}^T \boldsymbol{C}_{\dot{\boldsymbol{\Phi}}} \boldsymbol{W})$$
$$s.t. \boldsymbol{W}^T \boldsymbol{C}_{\boldsymbol{\Phi}} \boldsymbol{W} = \boldsymbol{I}$$

The solution of this definition leads to a generalized eigenvalue problem [26]:

$$\boldsymbol{C}_{\dot{\boldsymbol{\Phi}}} \boldsymbol{W} = \boldsymbol{C}_{\boldsymbol{\Phi}} \boldsymbol{W} \boldsymbol{D}$$

$\boldsymbol{C}_{\dot{\boldsymbol{\Phi}}}$ and $\boldsymbol{C}_{\boldsymbol{\Phi}}$ are the covariance matrix of the time differentiated and original input signal $\phi^{(t)}$whose dimension is $I \times I$.

3.4.3 Speed Relation Preserving Slow Feature Analysis (srpSFA)

We start our discussion by defining the loss function we want to minimize and we continue with the appropriate constrains needed to be imposed. Similar to the standard SFA, in our approach the optimal parameters matrix $\boldsymbol{W} \in R^{I \times J}$ needs to be computed through which the new representations are $\boldsymbol{y}_{\boldsymbol{n}}^{(\boldsymbol{t})} = \boldsymbol{W}^{\boldsymbol{T}} \boldsymbol{\phi}_n^{(t)}$.

Loss function: In order to fulfill the preservation of speed in the feature space, we want to minimize the objective function:

$$\sum_{ij} \mathbb{E}_t \left[\left(\dot{\boldsymbol{y}}_i^{(t)} - \dot{\boldsymbol{y}}_j^{(t)}\right)^2 \dot{\Gamma}_{ij}^{(t)}\right.$$

under standard SFA oriented constrains, i.e., zero mean, unit variance and uncorrelated features for the new mapped node representations. The weight factor $\dot{\Gamma}_{ij}^{(t)}$ penalizes the distance between the new representations $\boldsymbol{y}_i^{(t)}$ and $\boldsymbol{y}_j^{(t)}$. The greater the weight factor, the greater the penalty the new mappings to be "close". A great value of $\dot{\Gamma}_{ij}^{(t)}$ denotes that the speed vectors $\dot{\boldsymbol{\phi}}_i^{(t)}$ and $\dot{\boldsymbol{\phi}}_j^{(t)}$ of the skeleton nodes $\boldsymbol{\phi}_i$ and $\boldsymbol{\phi}_j$ in the input space are "close". Thus minimizing the loss function above it is ensured that two speed vectors $\dot{\boldsymbol{\phi}}_i^{(t)}$ and $\dot{\boldsymbol{\phi}}_j^{(t)}$ that are "close" in the input space, their corresponding mappings $\boldsymbol{y}_{\boldsymbol{i}}^{(\boldsymbol{t})}$ and $\boldsymbol{y}_{\boldsymbol{j}}^{(\boldsymbol{t})}$ will also be "close" in the feature space. Provided the new

skeleton node representation of the n-th node $\boldsymbol{y}_n^{(t)} = \boldsymbol{W}^T \boldsymbol{\phi}_n^{(t)}$, the matrix notation of the objective function along with the constrains is given by:

$$\min_{\boldsymbol{W}} trace(\boldsymbol{W}^T \boldsymbol{\Phi} \mathrm{diag}(\boldsymbol{L}) \boldsymbol{\Phi}^T \mathbf{W})$$

$$\boldsymbol{W} \boldsymbol{\Phi} \boldsymbol{\Phi}^T \boldsymbol{W}^T = \boldsymbol{I}$$

3.4.4 Experimental evaluation

Data mapping: Given an input skeleton action sequence $\boldsymbol{\mathcal{V}} \in R^{I \times N \times T}$, and the learned mapping matrix $\boldsymbol{W} \in R^{I \times J}$ provided by the later optimization problem the new skeleton action representation is $\widetilde{\boldsymbol{\mathcal{V}}} = \boldsymbol{\mathcal{V}} \times_1 \boldsymbol{W}^T$ by applying n-mode multiplication [27]. Each mapped frontal slice $\widetilde{\boldsymbol{V}}_k$ [27] represents the new mapped posture representation that is going to be used later in in key-posture dictionary learning.

Data preprocessing: The input skeleton nodes $\boldsymbol{x}_n^{(t)}$ where transformed into the polynomial feature space by applying the polynomial feature expansion function $\boldsymbol{\phi}(\boldsymbol{x}^{(t)}) = [x_1^2,\ \mathrm{x}_1\mathrm{x}_2, \mathrm{x}_1\mathrm{x}_3, \mathrm{x}_2^2, \mathrm{x}_2\mathrm{x}_3, \mathrm{x}_3^2, \mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3]$. Next, in order to eliminate the effects of slightly different camera setups into the accuracy of the model, the new skeleton representations were transformed into a unified coordinate system [24]. Moreover, in order to eliminate the intra-class variance imposed by the same action execution from different actors, each video was separately normalized to have zero mean and unit standard deviation [28].

Parameter selection: The feature extraction part demands the specification of three types of parameters. Namely, the definition of skeleton graph $G(\cdot)$, the choice of the scales σ_i and σ_j in the Radial Basis Function (RBF) for the weight computation $\Gamma_{ij}^{(t)}$ and the output dimension of the mapped data, i.e., the parameter J. For the graph structure definition, the graph applied is the one defined by the human body skeleton structure plus an edge connection of some skeleton nodes with a node close to the center of the skeleton. For a pictorial representation see Figure 13. As far as the tuning of scale parameters is concerned, their values were determined by following the same approach in [29]. That is, the scale value is defined by $\sigma_i = d^2(\boldsymbol{\phi}_i^{(t)},\ \boldsymbol{\phi}_j^{(t)})$ where $\boldsymbol{\phi}_K^{(t)}$ is the K-th more distant neighbor skeleton node of the i-th node $\boldsymbol{\phi}_i^{(t)}$. This neighbor is defined by traversing the skeleton edges depicted in Figure 13. Finally, the parameter J that optimize the accuracy results was set $J = 8$.

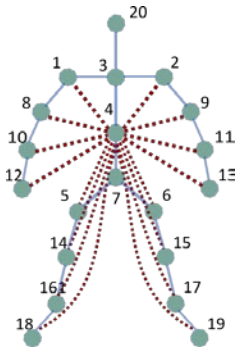

Figure 13: Skeleton structure of graph *G*

Video representation and classification: Given the mapped postures $\widetilde{\boldsymbol{V}}_k$, a dictionary of key-postures was constructed by applying k-means. The optimum number of key-postures K was selected among a range of K parameters. These key-postures were, used to extract a histogram representation of a given video. In order to capture the time dependency during the given action, a temporal pyramid of histograms was applied [28]. A similar approach was followed in [30]. Thus, the final skeleton action

representation is $7K$ dimensional vector. Those vectors were fed into a Support Vector Machine (SVM). The optimum SVM C parameter was chosen though a grid-search procedure whereas the $\mathcal{X}^2$-kernel was applied.

MSR-Action3D dataset [31]**:** A kinect-like depth sensor was used to obtain the recorded skeletons. It consists of 20 different recorded actions performed by 10 different subjects/actors. In addition, each subject repeated each recorded action two or three times. Namely, the actions are, *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side boxing*, *bend*, *forward kick, side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up \ throw*. For each skeleton the 3D joint locations through time were provided. In addition, the connections of the nodes that define the recorded skeleton were also given. Each recording was done in 15fps. Finally, because of the similarity of the actions this dataset is considered difficult. A variety of validation method exist [32]. Here we follow the one discussed here [31]. The whole dataset was split into three sub-sets namely, the AS1, AS2 and AS3. The data samples that correspond to the subjects with odd identification were used for training, i.e., 1, 3, 5, 7, 9, while the ones that correspond to the subjects with even identification, i.e., 2, 4, 6, 8, 10, were used for testing.

Table 5: Experimental results in MSR Action 3D

Method	AS1	AS2	AS3	Ave.
Li et. al. 2010 CVPRW [31]	72.9	71.9	79.2	74.7
Chen et. al. 2013 JRTIP [33]	96.2	83.2	92.0	90.47
Gowayyed et. al. 2013 IJCAI [34]	92.39	82.14	97.17	90.59
Vemulapalli et. al. 2014 CVPR [30]	95.29	83.87	95.50	94.49
Du et. al. 2015 CVPR [24]	93.33	**94.64**	95.50	94.49
Wang et. al. 2016 ICCV [35]	-	-	-	**96.9**
MaTHiSiS	**97.83**	91.96	**99.05**	96.28

In conclusion, a novel method for skeleton action recognition was introduced. In this method, the geometrical structure of the moving action skeletons leveraged in order to extract features that capture the context of an action. Good results were recorded on the MSR Action 3D publicly available dataset, comparable to the state of the art reported accuracies. As described in deliverable D4.1, these emotive actions are mapped to the three affective states (engagement, boredom, frustration), therefore the accuracy of detected actions in turn yield accurate results in affect recogntion.

3.5 Speech recognition and speech-based affect recognition

For our experiments we used four different audio datasets. Three of the datasets are publicly available (Emovo [36], Savee [37], German [38]) and the last one is a custom made dataset, which includes audio samples gathered from movies. For the custom made dataset the samples were annotated manually by several researchers in NCSR Demokritos. All the movies used for creation of our Movies-Dataset were in English except one that was in Portuguese. Statistics of the aforementioned datasets are reported in Table 6.

Table 6: Number of audio samples of the original audio databases for each class

	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Boredom
Emovo	84	84	84	84	84	84	84	-

Savee	60	60	60	60	120	60	60	-
German	127	46	69	71	79	62	-	81
Movies	367	-	80	63	413	117	-	-

Since not all datasets included samples for all the classes of Table 6, we decided to work only on their union. Thus, our final dataset consists only of the five common classes, namely anger, fear, happiness, neutral and sadness.

We used pre-segmented audio samples, randomly cropped from the original audio signal, each one belonging to any of the 5 classes (happiness, fear, sadness, anger, neutral) and with fixed duration equal to 2 sec. More specifically, we trained four different models, each time using samples from a single dataset. Each model was trained using the 80% of the samples from each class of the dataset. For evaluation purposes we performed four different experiments for each trained model (i.e., 16 experiments in total). We tested each trained model on the remaining 20% of samples from each class of the training dataset (note that those samples were used only for testing). Then we performed three additional experiments using each time all samples of each one of the other datasets.

Since we wanted to experiment with deep learning techniques which require huge amounts of training data, in order to achieve satisfactory classification performance rates and avoid over-fitting we performed data augmentation, to overcome the data scarcity problem. In general, data augmentation may be considered as a series of deformations that are applied on the annotated training samples. This process results in new additional training data. Since we wanted to work with spectrograms (i.e., visual representations of samples in the frequency domain), one option was through image reformations such as horizontally flipping, random crops and color jittering. Although this may work in computer vision problems, in our case, we opted to add a background sound (playing the role of noise) before extracting the spectrogram of each training sample. This procedure was repeated for three different SNRs (i.e., 5, 4 and 3) for the crop of the original audio sample. By including the original (i.e., without noise) training sample, data augmentation achieves a 3× dataset increase.

For recognising the five-target emotion labels, using a deep learning approach, we utilized a custom CNN classifier (CNN_EM) that operates upon the aforementioned spectrogram images. As recent literature has shown, deep hierarchical visual feature extractors can significantly outperform shallow classifiers trained on hand-crafted features and are more robust and generalizable when countering problems that include significant levels of inherent noise. The architecture of our deep CNN structure was finalized after a very extensive experimentation process on different layer combinations and parameter tuning. Our goal was to build a model, that could depict robust feature representations for recognizing speech-emotion across all the datasets, in a language independent manner.

The architecture consists of 4 convolution layers, with a stride of 2. The kernel sizes of the convolutional layers are of size 7, 5, 5 and 3 respectively. After every convolution and before the application of the non-linearity function we normalize the input batch using the Batch-Normalization transformation. In addition, in-between the initial 3 convolutional layers and after the last one, a pooling layer followed by a normalization layer is interposed. Normalization layers adopt the LRN normalization method and all max-pooling layers have a kernel with size equal to 3 and a stride of 2. The last two layers of the network are fully connected layers with dropout, followed by a softmax classifier, that shapes the final probability distribution. For all the layers we used the ReLu as our activation function and weights are always initialized using the xavier initialization. For the learning algorithm we decided to use the standard SGD, as it lead to superior results compared to other learning algorithms. The output of the network is a distribution on the five target classes. We have adopted a 5000-iterations fine-tuning procedure, with an initial learning rate of 0.001, which decreases after 600 iterations by a factor of 10. The input to the network corresponds to images of size 250×250 and organized in batches of 64 samples.

For comparison purposes we have evaluated the following two methods:

- audio-based classification (see D4.1): we extracted mid-term audio feature statistics. Classification has been achieved using SVM classifiers directly on the audio domain and using a plethora of features.
- image-based SVM: an SVM classifier applied on hand-crafted image features has also been evaluated. In particular the following visual features have been used to represent the spectrogram images: HoG, LBP and color histograms. Training data were the same as the ones used for CNN.

Experimental results in terms of the achieved F1 score within the testing data of the proposed emotion classification approach, compared to the audio-based classification and the image-based classification with hand-crafted features are presented in Table 7.

Table 7: Experimental results indicating the testing error of tested methodologies. Each row indicates the training and each column the testing set used. We mainly focus on the average F1 measure as the final evaluation metric, due to its ability to be robust against unbalanced datasets.

		Audio-Based SVM				Image-Based SVM				CNN_EM			
		Test Dataset											
		Emovo	Savee	German	Movies	Emovo	Savee	German	Movies	Emovo	Savee	German	Movies
Training Dataset	Emovo	0.48	0.22	0.49	0.28	0.42	0.14	0.42	0.20	0.57	0.16	0.42	0.27
	Savee	0.29	0.57	0.34	0.26	0.21	0.32	0.28	0.26	0.30	0.60	0.33	0.31
	German	0.41	0.26	0.64	0.32	0.43	0.25	0.28	0.29	0.41	0.24	0.67	0.35
	Movies	0.26	0.22	0.33	0.29	0.27	0.23	0.33	0.30	0.29	0.24	0.42	0.23
Average F1		**0.35**				**0.31**				**0.36**			

By observing these results, we may conclude the following:

- CNN_EM is the best method with respect to the average cross-dataset F1 measure. Audio-based classification is 1% lower, while the SVM classifier on hand-crafted visual features achieves almost 5% average F1 measure.
- CNN_EM is the best method for 9 out of 16 in total classification tasks, while audio-based classification is the best method in 5 of the classification tasks.
- CNN_EM, which operates directly in the raw data, is more robust across different domains and languages and can be used as an initialization point and/or knowledge transferring mechanism to train more sophisticated models.

For the driver pilots, the audio-based SVM was used, since it was the first solution to be implemented for the speech-based affect recognition. Since, however, CNN_EM has been proven as the most efficient solution in evaluations with public datasets, the deployment of this solution will be opted for the assisted pilots. In any case, with the availability of the collected MaTHiSiS dataset, evaluation of both methods with real-world data with regard to each MaTHiSiS use case will follow, in order to establish the optimal solution for the MaTHiSiS domain and purposes.

4. Sensorial Components communication API

The SC is the hub between the sensorial cues provided by the various MaTHiSiS Platform Agents (PAs) and the MaTHiSiS CLS (Cloud-based Learner's Space). The overall architecture of WP4 and its place within the MaTHiSiS ecosystem is highlighted in orange in Figure 14, where the SC's positioning can be viewed, residing between the PAs and the Affect and Intent Recognition library (AIR lib), which handles the collection and synchronization of the sensorial and transactional learner cues.

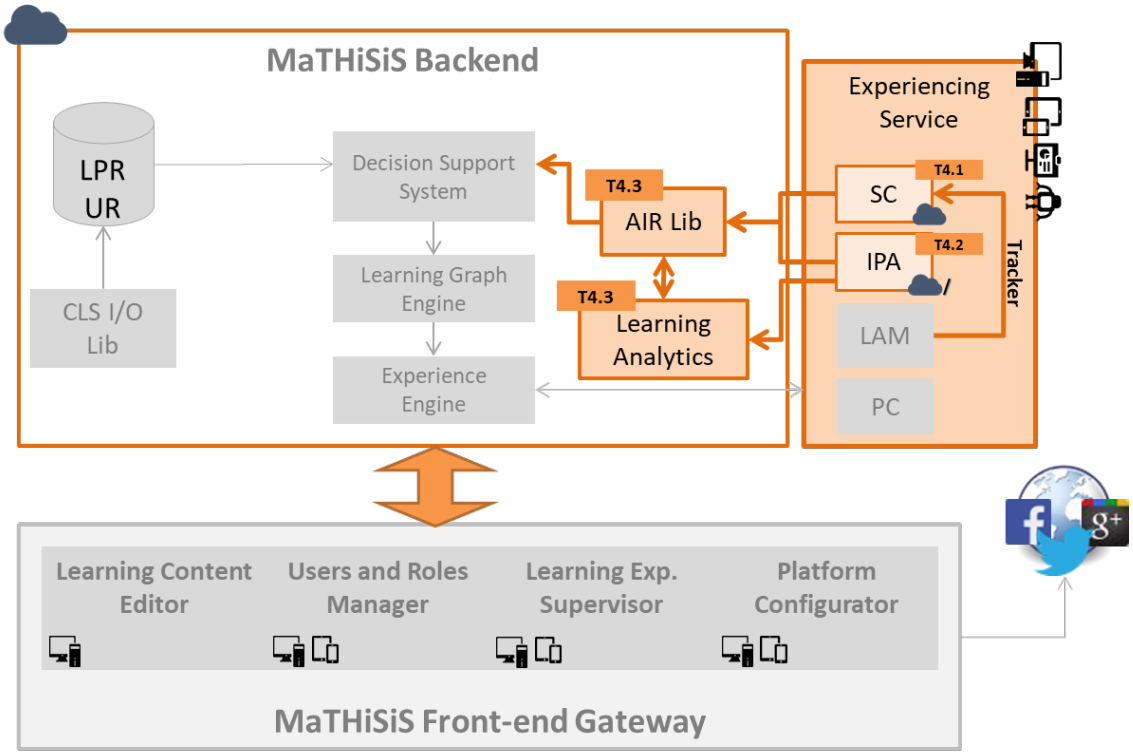

Figure 14: The WP4 tools' placement in the MaTHiSiS ecosystem, including the connection of the SC with the cloud and the Platform Agents

4.1 PA-side communication

The Sensorial Component is a part of the Experiencing Service (ES) space that is located in the Platform agent layer. It involves the implementation of all algorithms intended to capture information about the affective state of a learner. Among its main responsibilities is the gathering process of raw data from different PA sensors (e.g. cameras, microphones etc.). The SC communicates with the entire MaTHiSiS system through the Experience Service layer, including receiving requirements about when to transmit information to the cloud and the preferred sensors to be used in a Learning Session, as well as transmitting the features about the affect state of the learners to the AIR lib.

During the driver pilots, two versions of the Experience Service (ES) were implemented: One for mobile PA, which runs on mobile devices and one for all other PAs (desktop, IWB, robots), running on desktop.

The main part of SC is a standalone software implemented for Windows, given the restrictions of the sensor and the algorithms. This particular software communicates with the ES using sockets

communication protocol[2]. Namely, the SC involves a socket server, which waits for ES messages (when to respond, which sensors to use) and responds back predicted labels for all SC modalities. In detail, when the SC is launched, it waits for the configuration message, which has the following format:

```
{
"command": "CONFIGURE",
"store_data_filepath": "C:\\MaTHiSiS_Driver_Pilots\\",
"webcamera": {
        "intrisics":
"C:\\Experiencing_Service_1.9\\ApplicationClient\\GUIntrinsics\\intrinsics\\webcamera.xml",
        "device_id": 0
},
"kinect": {
        "intrisics":
"C:\\Experiencing_Service_1.9\\ApplicationClient\\GUIntrinsics\\intrinsics\\kinect.xml",
        "device_id ": 1
},
"nao": {
        "intrisics":
"C:\\Experiencing_Service_1.9\\ApplicationClient\\GUIntrinsics\\intrinsics\\nao.xml",
        "ip": "192.168.1.2"
}
}
```

This message is responsible to provide the following configuration information to the SC:

- **store_data_filepath** is the local path where recorded data will be stored, if data collection is opted for the Learning Session (for further enrichment of the MaTHiSiS dataset).
- **"webcamera", "kinect", "nao"** Informs the SC about the availability of the corresponding sensor (by NAO the entirety of sensors on the robot is implied) in the particular Learning Session. If one of these parameters are not included in the message sent from the ES to the SC, then no such sensor will be used by the SC in the particular session.
- **intrisics** is the file path for an xml record of the intrinsic information (e.g. camera focal length) of each corresponding available sensor – intrinsics are captured through a calibration step on the ES tool.
- **device_id** is the id of the sensor (concerning web cameras and Kinects) to be used in the Learning Session.
- **ip** is the current IP of the NAO robot.

According to the aforementioned information, the SC sets up and launches the corresponding available sensors for the particular Learning Session. Then it expects the messages that will inform it when it should start/stop the affect state extraction process.

In essence, the SC is not aware of the rest of the activities of the Learning Session, e.g. when the ES receives the LM, when an LM is launched, or when the LM is finished. The ES informs the SC through

[2] https://en.wikipedia.org/wiki/WebSocket

messages when it needs to receive the SC's outputs and sends the information back to the cloud. A typical message has the following format:

```
{
"command": "START",
"learner": "5821f8e11d56e4334ad48ed1",
"session": "593699c0c12d3168adb77017",
"record_session": "True",
"audio_modality_source": "kinect",
"facial_modality_source": "webcamera",
"gaze_modality_source": "webcamera",
"motion_modality_source": "kinect",
"timestamp": "000015",
}
```

The fields in this message include:

- **command:** takes two values, START or STOP, and indicates that this messages is the start point or the termination point of a time period that the system is interested in having the affect state of the learner.
- **learner:** the ID of the learner (as stored in the cloud)
- **session:** the ID of the Learning Session (as stored in the cloud)
- **record_session**: this field involves two options (true, false) and declares if the current session should be recorded or not for data collection.
- **timestamp:** the value of this field is used in order to synchronize the data gathered by the sensors connected to the desktop ES with data from mobile sensors (the mobile ES), in case where a mobile PA was used in the session.

4.2 Platform-side communication (AIR lib)

The Affect and Intent Recognition library (AIR lib) is the component through which the Sensorial Component communicates the data acquired locally in the Platform Agents to the cloud. This component is also in charge of the synchronization of the SC data and the key moments (which prominent user interactions with the PAs and Learning Materials). Using these two inputs, the adaptation mechanism is triggered. The data sent by the SC is formatted using the JavaScript Object Notation (JSON) as follows:

```
{
"affect_label_probs": [{vector of probabilities (3 values in [0, 1] denoting probability of boredom, engagement and frustration) per Sensorial Component, if available}],
"affect_labels": [{label of the most likely affect state detected per SC), if available}],
"features": [{vector of features per SC, non-fixed sizes, if available}],
"learner_id": {learner identifier},
"sensor_num": {number of SC modalities sent in the statement},
```

"sensors_type": [*{labels to identify the corresponding SC, initially defined as: "audio", "face", "mobile", "skeleton", "gaze"}*],

"session_id": *{session identifier}*,

"timestamp": *{timestamp of the generation of the statement}*

}

The data sent by the SC could use two different formats. If the SC has the ability to directly infer affective states, this information is sent using the parameter *"affect_label"*. Moreover, in this case, the probabilities of each of the possible affective states, based on the theory of flow (namely engagement, boredom and frustration), need to be provided. These probabilities must be included in the parameter *"affect_label_probs"*. This parameter contains several groups of values, one per each of the modalities transmitted.

On the other hand, if the SC only extracts, or is decided to provide only affect-related features, as input to the multimodal fusion, this information is transmitted to the cloud per se in order to be processed and classified. In that case, the parameter *"features"* contains a vector of features per each modality recorded and sent. The format of the value in this field varies and depends on the potential outcome of each modality's algorithm, as mentioned in Section 3.1 of Deliverable D4.1 [5].

As aforementioned, for the driver pilots, due to the selection of late fusion as the initial approach for multimodal fusion, all SC modalities followed the first approach, sending affect labels and respective probabilities and no feature data.

5. Sensorial Component in MaTHiSiS

This chapter will present the first results and/or observations of the in situ use of the SC modalities in the MaTHiSiS platform in the driver pilots.

5.1 Sensorial Component in the driver pilots

The Sensorial Component has already been employed in the MaTHiSiS driver pilots. During this time, it demonstrated its capacity to run in real time and in real life educational settings, and to yield meaningful (observed) output for the overall fused affect recognition of the learners during their Learning Experiences with the MaTHiSiS platform. Another significant finding is that the presence of sensors (i.e. cameras, Kinect) devices did not invoke any inconvenience to the users of the MaTHiSiS system.

The MaTHiSiS system was able to make meaningful and accurate decisions about the learner's affect. Also, the ability of the system to perform reliably for a reduced set of modalities in some cases (i.e. instead of using all the information from the entirety of sensors that the SC supports), e.g. using the NAO camera only, was a positive achievement of a required user story, as defined in Deliverable D7.1 [39].

On the other hand, the SC also met some complications during the sessions. The first was the position of the sensors during the sessions when the learner was interacting with LMs in his mobile device. Most users did not face the camera(s), since they were looking down to mobile device's screen, therefore obscuring their face and thus, facial expression and gaze estimation algorithms were difficult to predict their emotion. As a consequence, many sessions on mobile devices yielded unrepresentative results (constant non-engagement), which affected the quality of the platform's adaptation methodology performance. Similar problems were detected in sites were NAO robots were used with the robot's "autonomous life" mode on. This mode enforces the robot to turn and lock its field of view on the face of the nearest detected person speaking. During the execution of learning materials where the learner did not need to speak, this was often the tutor, residing somewhere close-by in the room, which gave explanations to the learner.

In addition, issues similar to the ones observed in the data acquisition phase and reported in Section 2.3, i.e. ambient noise and malpositioning of the robots towards the sensors (e.g. too close to cameras so that the skeleton could not be detected by the Kinect or the face was cropped out of the field of view) also yielded unrepresentative results of the SC components. In the ASC case, for example, learners were spontaneously moving significantly during the session as a result to be out of (web or NAO) camera's and the Kinect's field of view. However, such conditions were foreseen to a certain extent and also did not damage critically the SC's performance, as in the cases above.

Finally, there were some restrictions set by the nature of the sensors and the algorithms. The use of Microsoft Kinect camera limits the choices of operating systems that could run the SC. Kinect 2 is compatible only on window 8.1 and above. Furthermore, the SC consists of resource-demanding algorithms. Considering also that the Kinect camera by itself is a resource-demanding piece of hardware, the non-mobile part of the SC can run in its full extent only on powerful devices (mainly PCs).

As a conclusion, it is worth noticing that the SC modalities were the only source of information for the adaptation of the Learning Experience tailored on the user's needs for cases where learning materials that could not provide information about the learners on-activity performance (e.g. score) were used in the driver pilots, with reliable results when the sensor set-up was not incorrect.

6. Conclusion

This document reported on the implementation and results of the modalities that comprise the MaTHiSiS Sensorial Component. For the first version of the platform, the SC transmitted a prediction of affect state labels per modality to the multimodal fusion algorithm of Task 4.3, which were used for late fusion and final prediction of an overall affective state of the user in critical moments of their Learning Sessions, with good in vivo results in their MaTHiSiS-facilitated Learning Experience. In subsequent versions of the platform and in the scope of Task 4.3, the direct transmission of the raw features that were used per each modality to predict affect may be opted, were the benefits (or not) of early fusion will be explored for the purposes of MaTHiSiS. While in the first MaTHiSiS pilots, the goal was to make crucial empirical observations over the performance of the SC, quantitave evaluations involving each sensorial component separately and also user-material interactions, will be pursued in subsequent piloting phases and will be reported in Deliverables D4.3 and D4.4 (Affect Understanding in MaTHiSiS, M21 and M33 respectively). Multimodality will be reported in D4.5 and D4.6 (Multimodal learning analytics, M21 and M33).

Furthermore, the SC algorithms will be re-purposed to better address the particularities of the MaTHiSiS use cases, through training and testing on the dataset collected within the MaTHiSiS project. To this end, an extensive data acquisition task was organised within WP4, in order to collect controlled data for all WP4 modules. This task, although ultimately not free of hampers and problems, resulted to a first comprehensive dataset for all SC modalities, but also for T4.2 and T4.3 methods. This dataset will be used to train WP4 algorithms, which will be subsequently used (and tested) during the MaTHiSiS assisted and real-life pilots. This first set of data will be further enriched, with a high focus on (re)collecting data for the data-scarce mobile-based affect recognition modality, with data collected in subsequent pilots and will solidify an educational-pertinent affect recognition methodology in the second release of the MaTHiSiS platform.

An important and highly anticipated outcome of the data acquisition phase and of the first real-life implementation of the SC in the driver pilots consisted of insights about the reliability of different SC modalities per MaTHiSiS use case and/or per learning setting. This included for example the consideration that affect recognition in mobiles might only reliably run with the mobile-based inertia sensor recognition modality and with sound, otherwise a setting that facilitates the positioning of the user in a camera's field of view while interacting with a mobile device needs to be considered. On the other hand, some unreliability results are proved surpassable through continuous familiarization of piloting partners with the MaTHiSiS platform during the course of the pilots, coupled with added technical guidance. For example, the SC performance will be dramatically enhanced if piloting partners make sure that cameras fully face the learner and that ambient noise and tutor-learner visual and audio overlaps are reduced as much as possible.

Overall, a first positive qualitative evaluation of the SC in the MaTHiSiS ecosystem yielded an affirmation of the strategies followed and algorithms developed in Task 4.1 and of the value of sensory-based affect state recognition in a technology-enhanced Learning Experience. To this end, the SC modalities will keep being enhanced in order to align with future requirements of the multimodal fusion approach of WP4.

7. References

[1] Nottingham Trent University (ed.), "Adaptation and Personalization principles based on MaTHiSiS findings.," Deliverable for the MaTHiSiS project., 2017.

[2] T. Kanade, J. F. Cohn, P. Lucey, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression," San Francisco, USA, 2010.

[3] P. Ekman and W. Friesen, "Facial Action Coding System: A Technique for the Measurement of Facial Movement," in *Consulting Psychologists Press*, Palo Alto, 1978.

[4] R. S. J. d. Baker, S. K. D'Mello, M. T. Rodrigo and A. C. Graesser, "Better to Be Frustrated Than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-affective States During Interactions with Three Different Computer-based Learning Environments," *Int. J. Hum.-Comput. Stud.,* vol. 68, pp. 223-241, 2010.

[5] Centre For Research and Technology Hellas (ed.), "D4.1: MaTHiSiS sensorial component M12," Deliverable of the MaTHiSiS project, 2017.

[6] N. Shaker, S. Asteriadis, G. Yannakakis and K. Karpouzis, "Fusing Visual and Behavioral Cues for Modeling User Experience in Games," *IEEE transactions one Cybernetics,* vol. 43, no. 6, 2013.

[7] N. Shaker, S. Asteriadis, G. Yannakakis and K. Karpouzis, "A Game-based Corpus for Analyzing the Interplay between Game Context and Player Experience," 2012.

[8] X. Zhang, Y. Sugano, M. Fritz and A. Bulling, "Appearance-based gaze estimation in the wild," 2015.

[9] J. Li, T. Wang and Y. Zhang, "Face detection using surf cascade," in *Computer Vision Workshops (ICCV Workshops), IEEE International Conference on IEEE*, 2011.

[10] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[11] D. F. Dementhon and D. S. Larry, "Model-based object pose in 25 lines of code," 1995.

[12] K. Mora, A. Funes, F. Monay and J.-M. Odobez, "Eyediap: A database for the development and evaluation of gaze estimation algorithms from rgb and rgb-d cameras.," in *Proceedings of the Symposium on Eye Tracking Research and Applications. ACM*, 2014.

[13] B. Tadas, P. Robinson and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit.," Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on IEEE, 2016.

[14] B. Tadas, P. Robinson and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild.," Proceedings of the IEEE International Conference on Computer Vision Workshops., 2013.

[15] Simonyan, Karen and A. Zisserman, "Two-stream convolutional networks for action recognition in videos.," 2014.

[16] L. Feng et al, “Gaze estimation from eye appearance: a head pose-free method via eye image synthesis.,” in *IEEE Transactions on Image Processing*, 2015.

[17] G. Leifman and et al, “Learning Gaze Transitions from Depth to Improve Video Saliency Estimation.,” in *arXiv preprint arXiv:1603.03669*, 2016.

[18] T. D. Rikert and M. J. Jones, “Gaze estimation using morphable models.,” in *Automatic Face and Gesture Recognition, Proceedings. Third IEEE International Conference on. IEEE*, 1998.

[19] T. Brox et al., “High accuracy optical flow estimation based on a theory for warping.,” 2004.

[20] C. Coutrix et al, “Identifying emotions expressed by mobile users through 2D surface and 3D motion gestures,” in *in Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012.

[21] H. J. Kim and Y. S. Choi, “Exploring emotional preference for smartphone applications,” in *In 2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 2012.

[22] Kim, Mira et al., “A touch based affective user interface for smartphone,” in *in IEEE International Conference on Consumer Electronics (ICCE)*, 2013.

[23] Amelynck, Denis et al, “Toward e-motion-based music retrieval a study of affective gesture recognition.,” in *in IEEE transactions on affective computing 3.2*, 2012.

[24] Y. Du, W. Wang and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in *Proceedings of the IEEE Internatiojnal Conference on Computer Vision and Pattern Recognition(CVPR)*, 2015.

[25] L. Wiskott and T. J. Sejnowski, “Slow Feature Analysis: Unsupervised Learning of Invariances,” *Neural Computations,* vol. 14, pp. 715--770, 2002.

[26] E. Kokiopoulou, J. Chen and Y. Saad, “Trace optimization and eigenproblems in dimension reduction methods,” *Numerical Linear Algebra with Applications,* vol. 18, pp. 565--602, 2011.

[27] T. G. Kolda and B. W. Bader, “Tensor Decompositions and Applications,” *SIAM Review,* vol. 51, pp. 455--500, 2009.

[28] Y. Shan, Z. Zhang and K. Huang, “Learning Skeleton Stream Patterns with Slow Feature,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2015.

[29] L. Zelnik-manor and P. Perona, “Self-tuning spectral clustering,” in *Advances in Neural Information Processing Systems* , 2004.

[30] R. Vemulapalli, F. Arrate and R. Chellappa, “Human action recognition by representing 3d skeletons as points in a lie group,” in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[31] W. Li, Z. Zhang and Z. Liu, “Action recognition based on a bag of 3d points,” in *Proceedings of IEEE International Conference in Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2010.

[32] J. R. Padilla Lopez, A. A. Chaaraoui and F. Florez Revuelta, “A discussion on the validation tests employed to compare human action recognition methods using the {MSR} Action3D dataset,” *Computing Research Repository(CoRR),* vol. abs/1407.7390, 2014.

[33] C. Chen, K. Liu and N. Kehtarnavaz, "Real-time human action recognition based on depth motion," *Journal of Real-Time Image Processing (JRTIP),* pp. 1-9, 2013.

[34] M. A. Gowayyed, M. Torki, M. E. Hussein and M. El-Saban, "Histogram of Oriented Displacements (HOD): Describing Trajectories of Human Joints for Action Recognition," in *In International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[35] L. Wang, J. Zhang, L. Zhou, C. Tang and W. Li, "Beyond Covariance: Feature Representation with Nonlinear Kernel Matrices," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[36] G. Costantini, I. Iaderola, A. Paoloni and M. Todisco, "M. EMOVO Corpus: An Italian Emotional Speech Database," in *LREC 2014*, 2014.

[37] P. Jackson and S. Haq, "Surrey Audio-Visual Expressed Emotion (SAVEE) Database," in *University of Surrey*, UK, 2014.

[38] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier and B. Weiss, "A database of German emotional speech," in *Interspeech 2005*, 2005.

[39] ATOS (ed.), "D7.1: MaTHiSiS Integration Strategy and planning M6.," Deliverable for the MaTHiSiS project., 2016.