SPRING

GRANT AGREEMENT N. 871245

Deliverable D4.1

Human Description in Realistic Environments

Due Date: 16/06/2021
Main Author: UNITN
Contributors: INRIA, CVUT, BIU
Dissemination: Public Deliverable

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

SPRING

DOCUMENT FACTSHEET

Deliverable no.	D4.1: Human Description in Realistic Environments
Responsible Partner	UNITN
Work Package	WP4: Multi-modal human behaviour understanding
Task	T4.1: Describing Humans: provide a description of the status of each person
Version & Date	V1, 16/06/2021
Dissemination level	[X] PU (public) [ ] CO (confidential)

CONTRIBUTORS AND HISTORY

Version	Editor	Date	Change Log
1	UNITN	16/06/2021	First Draft
Final	UNITN, CVUT	30/06/2021	Final draft with partner comments

APPROVALS

Authors/editors	UNITN, INRIA, CVUT, BIU
Task Leader	UNITN
WP Leader	UNITN

Contents

1 Introduction

This deliverable is part of **WP4** of the H2020 SPRING project. The objective of **WP4** is "*developing technologies for analysing human behaviours from multi-modal sensors robotic platforms.*" Three main software modules are required to enable the robot to understand human behavior:

- **T4.1** *Describing Humans*, where the goal is to provide a description of the status of each person;
- **T4.2** *Individual & Group Behaviour Recognition*, where the goal is to classify human behaviours both at individual and at group-level;
- **T4.3** *Affect & Robot Acceptance Analysis*, where the goal is to analyze the affective state of the user(s) interacting with the robot, and specifically the level of acceptance of the robot.

D4.1 describes the methods and the software used for the **T4.1** task, namely "describing humans." All the experimental results reported in this document have been obtained on large-scale public datasets in order to provide both qualitative and quantitative evaluation.

All software developed in the SPRING project is expected to run on the robotic platform ARI, whose visual perception capabilities have been enhanced specifically for SPRING. However, the COVID-19 pandemic still constrains the way of conducting experiments in our research laboratories. In particular, the ARI robot arrived later than what was originally planned and we still have limited access to it. As a consequence, we are unable now to report the performance of our software modules in realistic environments. In addition, we have not been able yet to integrate our modules with the ones from other workpackages. For instance, we have not assigned the tracking ID of the pedestrian to the detected faces yet (see Section 3).

However, we have made progress in other directions, in order to compensate as much as possible the overall delays in the **WP4** progress, and in the whole SPRING project. In this respect, in addition to the methodology and software used for **T4.1**, we describe here also some ongoing research. In particular, we discuss the techniques INRIA has developed for face frontalization, which will be used in our future work (see Section 4.1). Besides, we have started to work on some modules which are supposed to be implemented in the future to compensate for the delay, such as facial expression recognition, which is related to **T4.3** (see Section 4.2).

This document is organized as follows. Section 2 describes the overall architecture for describing humans, as well as each of the modules. Then, Section 3 discusses dependencies of **D4.1** with respect to other workpackages, and Section 4 presents ongoing research. The document ends drawing some conclusions and future work. The software is being updated in SPRING-WP4-Repository. As per European Commission requirements, the repository will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software to the project coordinator at spring-coord@inria.fr. The software packages will use ROS (Robotics Operating System) to communicate with each other and with the modules developed in the other workpackages.

2 Modular Architecture for Face Analysis

2.1 Overall Architecture

The goal of **T4.1** - *Describing Humans* (**M7 - M21**) is to devise approaches to provide a description of the status of each person. That is, to estimate the 3D pose of the user(s) interacting with the robot from visual data and to implement state-of-the-art solutions for face analysis (detection and recognition), head pose estimation, facial landmarks extraction and prediction of soft biometric patterns (age, gender, etc.). Transfer learning and domain adaptation techniques will be considered within a deep learning framework in order to exploit synthetic data from **T2.2** and publicly available datasets, thus avoiding the need of annotating novel data in the considered egocentric camera setting. The use of audio signals as well as the information derived from the 3D semantic maps of WP2 will contribute to improve the recognition accuracy of human pose estimates and of soft-biometric patterns. **WP4** has 4 main outcomes w.r.t **T4.1**:

- **Face Detection and Facial Keypoints Estimation**: from an RGB-Image, the task is to provide bounding boxes encompassing each human face in the scene. For each face, this module must also provide semantic keypoints to be used for facial expression analysis, biometric analysis and face verification.
- **Face Verification & Soft Biometric Analysis** (also called *state estimation*): this module needs to be able to determine whether two faces belong to the same person. In the meanwhile, it needs to infer soft biometrics, such as age and gender, based on RGB images. Face verification for partially occluded faces will be subject of future work.
- **Facial Mask Detection**: the objective of this module is to determine whether or not a face is wearing a mask.
- **Human Pose Estimation**: from an RGB-Image, this modules provides semantic keypoints corresponding to body and head pose for each human in the scene.

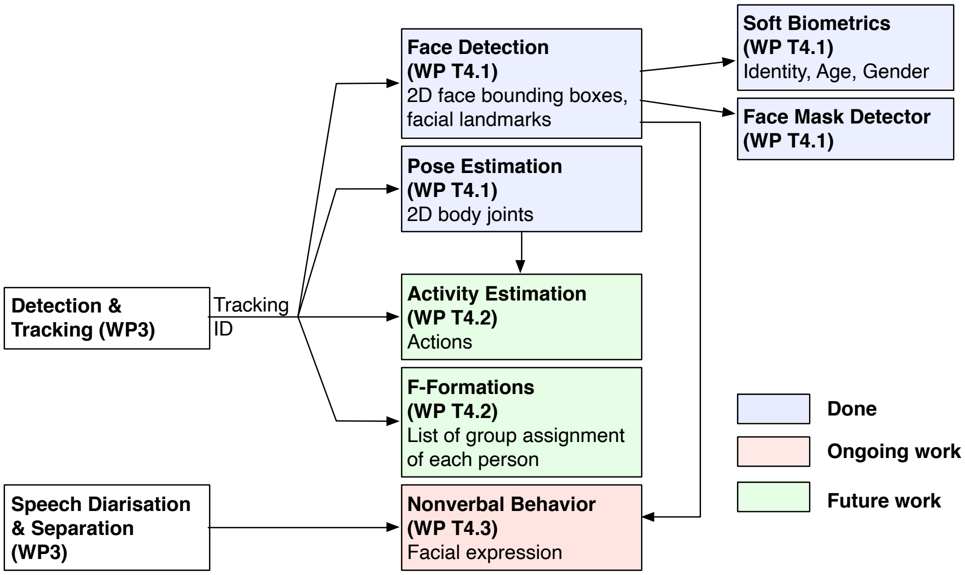


Figure 1: Diagram of the overall architecture. The tracking IDs are obtained from WP3. The IDs will be assigned to the detected faces. Besides, the tracking IDs will also be linked to the following tasks, *i.e.*, 3D pose estimation, activity estimation, formation estimation, and non-verbal behavior, *e.g.*, expression recognition.

The overall architecture for human behavior analysis is shown in Figure 1. For the time being the face analysis is independent from **WP3**, hence the tracking ID obtained from **WP3** will be assigned to the detected faces in the second stage. The blue boxes are the modules which have already been implemented, and the green boxes are the modules which will be implemented in the future. In the following subsections, we describe in more detail each module, highlighting the purpose, the input/output, the dependencies, the method we used and the current status of development.

2.2 Face detection

The purpose of face detection module is to detect/localize faces in a given image. Locating a face means finding the coordinates of the face, whereas localization refers to demarcating the extent of the face using a bounding box. Face detection also contributes to the soft biometric module, together with the face recognition module. This module is used to detect the face of a person who walks by. After detecting the faces, we will then calculate the gender, and age range of the face. It is also used as part of the facial expression/emotion inference module. Expression inference can be used to help us understand the feelings of the people around the robot. For instance, we can tell whether a person is smiling or has closed eyes. Besides, after localizing the face, we can crop out the face and use other softwares to compute the landmarks/coordinates of the eyes, ears, cheeks, nose, and mouth. Based on the detected facial landmarks, we can parse the face into different facial action units according to the facial action coding system which can be used to define the micro expressions.

To sum up, face detection is a non-trivial computer vision problem. Although detecting faces can be easily solved by humans, it is very challenging for computers given the dynamic nature of faces. For instances, faces in the images captured in the wild usually have different orientations or angles they are facing, illumination conditions, self occlusions by glasses, masks and hats, etc.

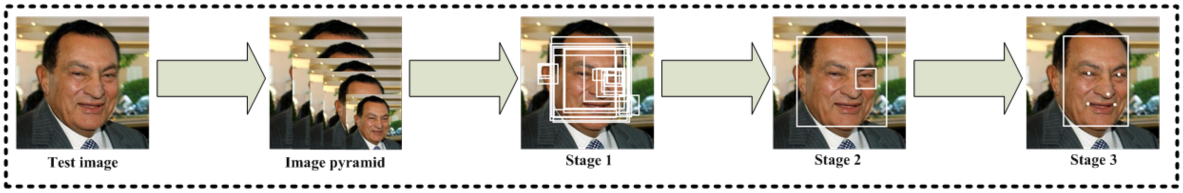


Figure 2: Pipeline of MTCNN [1].

(a) Examples of results on FDDB

(b) Examples of results on WIDER FACE

Figure 3: Examples of detected faces using MTCNN [1].

Method. Recently, with the development of artificial intelligence, especially in the domain of deep learning, big progress has been made for face detection. To apply face detection to videos, however, we need to process video frames in real time. State-of-the-art face detection can be achieved using a Multi-task Cascade CNN via the MTCNN library [1]. MTCNN is a deep cascaded multi-task framework which exploits the inherent correlation between detection and alignment to boost up its performance. As shown in Figure 2, this framework leverages a cascaded architecture with **three stages** of carefully designed deep convolutional networks to predict face and landmark location in a coarse-to-fine manner. In particular, Multi-task Cascaded Convolutional Networks (MTCNN) is a framework developed as a solution for both face detection and face alignment. The process consists of three stages of convolutional networks that are able to recognize faces and landmark location such as eyes, nose, and mouth. In the first stage, MTCNN uses a shallow CNN to quickly produce candidate windows. In the second stage it refines the proposed candidate windows through a more complex CNN. And lastly, in the third stage it uses a third CNN, more complex than the others, to further refine the result and output facial landmark positions.

Stage 1: The Proposal Network (P-Net)
This first stage is a fully convolutional network (FCN). This Proposal Network is used to obtain candidate windows and their bounding box regression vectors. Bounding box regression is a popular technique which is used in P-Net to predict the localization of face boxes. After obtaining the bounding box vectors, some refinement is done to combine overlapping regions. The final output of this stage is all candidate windows after refinement to downsize the volume of candidates.

Stage 2: The Refine Network (R-Net)
All candidates from the P-Net are fed into the Refine Network. Notice that this network is a CNN. The R-Net further reduces the number of candidates, performs calibration with bounding box regression and employs

non-maximum suppression (NMS) to merge overlapping candidates. The R-Net outputs whether the input is a face or not, a 4-element vector which is the bounding box for the face, and a 10-element vector for facial landmark localization.

Stage 3: The Output Network (O-Net)
This stage is similar to the R-Net, but this Output Network aims to describe the face in more detail and output the five facial landmarks' positions for eyes, nose and mouth.

Some detected faces are shown in Figure 2. MTCNN is also very fast on CPU. Given the good performance of MTCNN and its fast speed, we rely on MTCNN [1] library for face detection which can detect faces in videos in real time. The input of the face detection module are RGB images with bounding boxes of the detected persons. The output of the module is a list of bounding boxes with confidence scores. Now this module has been integrated with ROS.

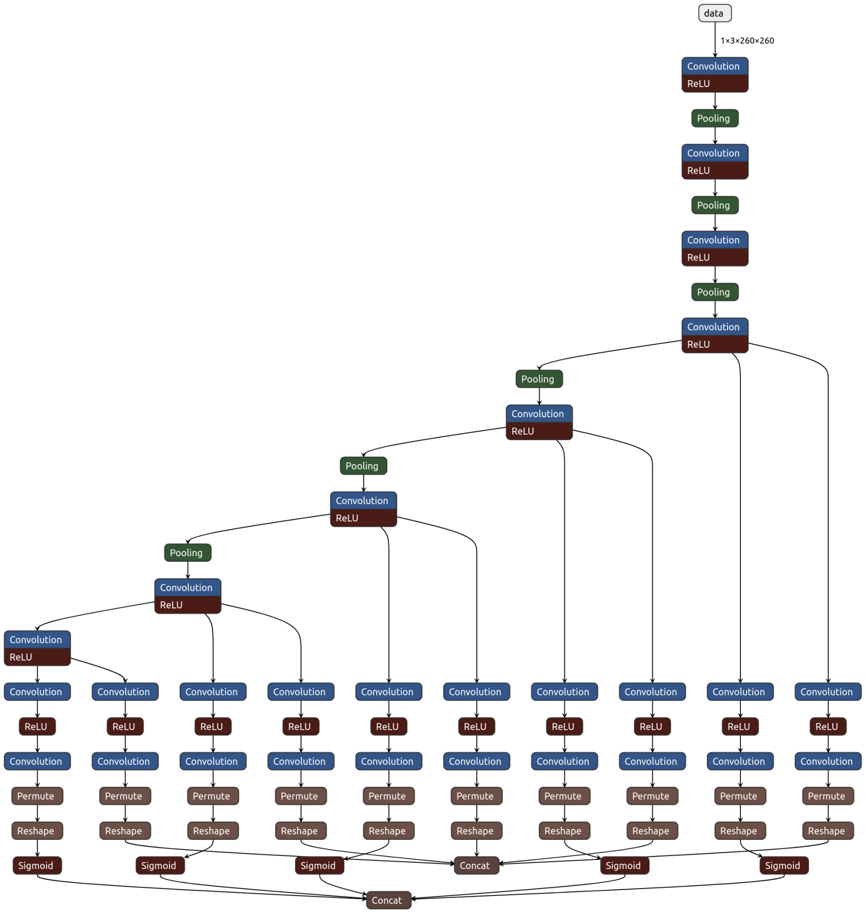


Figure 4: Mask detection model which is based on SSD [2]. The image is from https://github.com/AIZOOTech/FaceMaskDetection.

2.3 Face-mask detection

Now we are in the pandemic period of COVID-19 and it is usual for people to wear masks as a protection measure. Hence, there is growing need for mask detection, which in turn may have influence on the face recognition task. The purpose of face-mask detection module is to detect/localize faces and to establish whether a given face is wearing a protection mask.

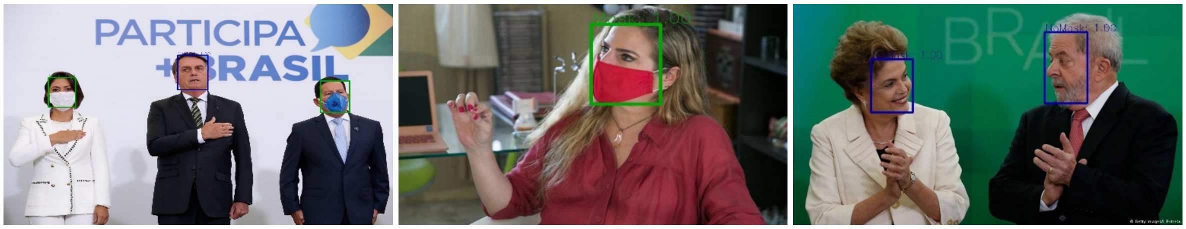


Figure 5: Face and Mask Detection

Method We started from the open source api provided by **AIZOOTech**, the link of their github implementation is `https://github.com/AIZOOTech/FaceMaskDetection` (The code is under MIT license). For the ros integration, A ROS wrapper was developed by us for integration. The prototype of mask detection model is Single Shot MultiBox Detector [2]. The model architecture is shown in Figure 4.

Single Shot MultiBox Detector (SSD) was originally designed for real-time object detection, as the traditional object detection algorithms, such as the Region-Convolutional Neural Network (R-CNN) [3], are very slow. Even though its successors fast R-CNN [4] and faster R-CNN [5] propose improvements to the seminal work to develop a faster network and the achievements are truly amazing, none of them manage to create a real-time object detector. To address the bottlenecks of R-CNN and its successors, two other models have been proposed which enable real-time object detection. One of the models is YOLO (You Only Look Once) [6] and the other one is SSD MultiBox (Single Shot Detector) [2].

Single Shot means that the tasks of object localization and classification are done in a single forward pass of the network. MultiBox is the name of a technique for bounding box regression. Detector represents the object detector which also classifies those detected objects. SSD's backbone is VGG-16 without the fully connected layers. VGG-16 was used because it has good performance in image classification tasks and its popularity for other problems in which transfer learning helps in improving results. Besides, some auxiliary convolutional layers (from conv6 onwards) were added, thus enabling to extract features at multiple scales and progressively decrease the size of the input to each subsequent layer. However, 80% of the time is spent on the base VGG-16 network: this means that with a faster and equally accurate network SSD's performance could be even better. For efficiency, the VGG-16 base has been replaced with a lighter network as shown in Figure 4.

Figure 5 shows some examples of the detected masks. The input is RGB image & bounding box of the detected faces. The output is a list of bounding boxes with detected faces and labels for faces with mask and without mask with the confidence scores. Now the algorithm has been integrated with ROS.

2.4 Soft biometric analysis

2.4.1 Identification: Face verification

The task of this module is to check whether the target person is known by the robot or not. This is done by matching the person's face to a dataset of known faces and determining whether a positive match is found. The state-of-the-art method is FaceNet [7]. It directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Once this space has been produced, tasks such as face recognition, verification and clustering can be easily implemented using standard techniques with FaceNet embeddings as feature vectors. FaceNet uses a deep convolutional network trained to directly optimize the embedding itself. To train, we use triplets of roughly aligned matching/non-matching face patches generated using an online triplet mining method. The benefit of this approach is much greater representational efficiency: it achieves state-of-the-art face recognition performance using only 128-bytes per face. On the widely used Labeled Faces in the Wild (LFW) dataset, FaceNet achieves the accuracy of 99.63%. On YouTube Faces DB it achieves 95.12%. FaceNet also introduces the concept of harmonic embeddings, and a harmonic triplet loss, which describe different versions of face embeddings (produced by different networks) that are compatible to each other and allow for direct comparison between each other.

As shown in Figure 7, the triplet has three points, *i.e.*, anchor point, positive point, and negative point. The triplet loss encourages the positive pair to be close to each other, while the negative pair is encouraged to

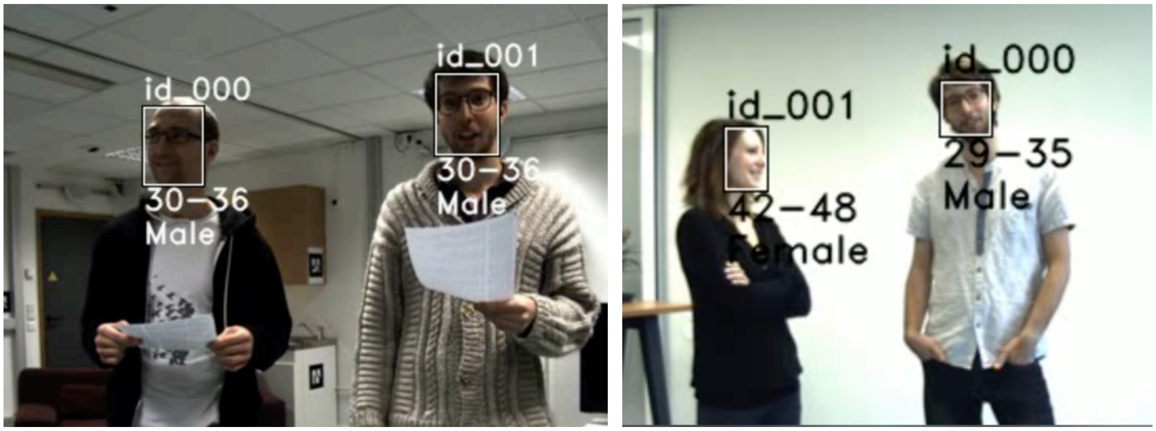


Figure 6: Face Verification. (testing images are from public benchmarks https://team.inria.fr/perception/avdiar.)

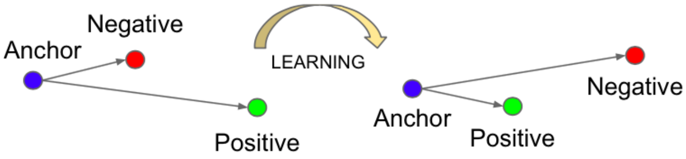


Figure 7: Triplet Loss used in FaceNet [7].

be far away from each other. This is achieved by the following constraint:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2 \tag{1}$$

in which x_i^a, x_i^p, x_i^n denote the anchor point, positive point, and negative point. This equation ensures that an image x_i^a (anchor) of a specific person is closer to all other images x_i^p (positive) of the same person than it is to any image x_i^n (negative) of any other person. This is visualized in Figure 7. Here α is a margin that is enforced between positive and negative pairs. However, the backbone of FaceNet is very large. To alleviate the computing burden of the robot, we use on Knowledge Distillation [8] method to learn a small network which has a similar performance as FaceNet.

Method We use the Knowledge Distillation [8] method to transfer the knowledge learned from the teacher network FaceNet [7] to the student network 'MobileNet V3 small sp' [9]. SP models are those that instead of having only the output layer set to match the number of classes of VGGFace2, they have a completely different classification module matching the one of FaceNet.

The teacher network is very large with 27.9 million parameters so that it is very time consuming to be executed. The student model is much lighter with 6.7 million parameters while has comparable performance with the teacher network. It is very fast. Therefore, the student network is more suitable for our task which has limited computing resources. The MobileNet takes as input the RGB image, and computes its embeddings (i.e, deep features), and then computes the distance between the embeddings and the ones of the reference images stored in the dataset. The identity is determined by checking whether the distance between itself and the input RGB image belows a certain threshold. If the distance is smaller than the threshold, we can say that they have the same identity. The training dataset is VGGFace2 [10].

- The match accuracy of the teacher on the test set is **99.9%** and the accuracy of our student is **98.1%**.
- The speed of the teacher on the CPU (Ryzen 7) is **36.0 ms/image** and the performance of our student is **9 ms/image**.

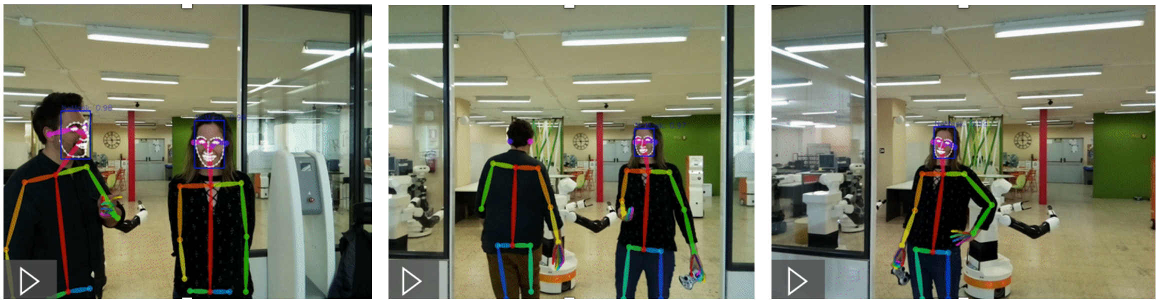


Figure 8: Pose Estimation

We have also tested MobileNetV2 sp [11] with 7.3 million parameters. It can reach the accuracy of 99.1%, but it consumes more time which is 24.8 ms/image. We have also tested other MobileNets, such as MobileNet V2, MobileNet V2 sp, MobileNet V3 large, MobileNet V3 large sp, etc. Finally, we select MobileNet V3 small sp [9] as our model as it balances well between the performance and speed. Last but not least, by increasing the size of the training set, we have further improved its match accuracy to **99.6%**.

The face verification results are shown in Figure 6 with the id number in the gallery. The input to the face verification module is a RGB image patch, and the output is the ID of the matched face or ‘unknown’. For the prepossessing (localization of the face), this module requires the output of face pose estimation. It has been integrated with ROS.

2.4.2 Age and Gender

Together with face recognition, we also need to extract other biometric information (approximated age and gender) of a target person. Even though there are other models for facial biometric information extraction, this will consume a lot of additional computing resources. To solve this problem, we use transfer learning technique. To be specific, we rely on the pretrained face verification model to extract features and add additional two small branches for two new tasks, namely, age and gender estimation. The reason is that face verification can extract discriminative features and these features may also reflect the age and gender information. Therefore, we add two extra classifiers to the FaceNet while its backbone remains the same. The details are as follows.

Method. Based on the model obtained from face verification, we have added another two classifiers at the end of the last convolutional layer of the deep network for the other two tasks, *i.e.*, age estimation and gender classification. The deep network is MobileNet V3 small. When training the network, the parameters of ‘MobileNet V3 small sp’ backbone remain the same and we only update the parameters in the classifiers based on the loss from gender and age. The dataset used for the training is AgeDB [12] dataset which contains both the age and gender label for the face. The input is a RGB image with facial key-points. The output is a list of biometric information (*i.e.*, age and gender).For the privacy issues, we will not show the experimetal results here, but the experimental results show that the age and gender estimator work well. This module has been integrated with ROS.

2.5 Pose estimation

The task of this module is to estimate the pose of each person detected by the robot, where a pose is represented through localization of semantic key-points of a person’s body (e.g., left hand, left elbow, left shoulder, etc). In particular, the pose representation with respect to an image consists of the pixel coordinates of each detected semantic keypoint. Realtime multi-person 2D pose estimation is a key component in enabling machines to have an understanding of people in images and videos. We rely on OpnePose [13] for this task. This method uses a nonparametric representation, which is referred to as Part Affinity Fields (PAFs), to learn to associate body parts with individuals in the image. This bottom-up system achieves high accuracy and realtime performance, regardless of the number of people in the image. In previous work, PAFs and body part location estimation were refined simultaneously across training stages. It is demonstrated that a PAF-only refinement rather than both PAF and body part location refinement results in a substantial increase in both runtime performance and accuracy. The combined detector not only reduces the inference time compared to running them sequentially, but also maintains the accuracy of each component individually.

Method. We use the open source API provided by OpenPose from Github. The link is `https://github.com/CMU-Perceptual-Computing-Lab/openpose`. Since it offers good performance already, we use it directly. The input is a RGB image and the output is a list of body joints and facial landmarks. The examples are shown in Figure 8. This module has been integrated with ROS. For the ROS integration we use the ros-wrapper implementation pointed by the official CMU repository: `https://github.com/ravijo/ros_openpose`. The code is under MIT license. For the details of the method, please check [13].

3 Dependency of Previous Works

3.1 Entity(person) tracking from WP3

The purpose of this module is to track the position of 2D entities among a sequence of frames. Given a bounding box on a reference frame and a sequence of following frames, the module attempts to re-locate the entity pointed by the bounding box at the following frames. If successful, the module returns the location of the bounding box in the new frame, otherwise it returns a flag indicating failure.

Method We have not received this module implemented from **WP3** yet. Therefore, as a substitute, we developed a wrapper for the opencv tracking API which is implemented based on [14]. The input includes a reference frame with a bounding box, and a sequence of frames. The output is the position of the bounding box for each frame and the success flag per frame. This module has been integrated with ROS.

4 Ongoing Works

4.1 Face Frontalization

Consider a typical HRI scenario that involves a robot and a group of persons. Participants inherently move their head, e.g. nodding. Consequently, non-rigid facial movement analysis is perturbed by rigid head movements. It is therefore important to separate facial movements from head movements. This can be achieved via face frontalization which consists of synthesizing, over time, a frontal view of a face from an arbitrary view.

Recently, it has been shown that face recognition from frontal views yields better performance than face recognition from unconstrained views [15, 16, 17, 18, 19, 20]. It is worth noticing that face recognition requires *expression-free frontalization*, while face expression recognition and visual speech recognition (or lip reading), e.g. [21, 22, 23, 24] require *expression-preserving frontalization*.

We address face frontalization as follows (please consult [25] for details). We detect a face and we extract 3D face landmarks. The latter is also referred to as 3D face alignment (3DFA) and a number of DNN architectures have been recently proposed. Such a 3DFA-DNN predicts the image-centred 3D coordinates of $J = 68$ landmarks, $\boldsymbol{X}_{1:J} = \{\boldsymbol{X}_j\}_{j=1}^{J} \subset \mathbb{R}^3$. We also consider a frontal 3D deformable face model which consists of a 3D triangulated mesh whose vertices are conditioned by the parameters of a shape model. Let this model be

$$\hat{\boldsymbol{V}}_n = \overline{\boldsymbol{V}}_n + \mathbf{W}_n \boldsymbol{s}, \quad \forall n \in \{1 \dots N\}, \tag{2}$$

where $\overline{\boldsymbol{V}}_{1:N} \subset \mathbb{R}^3$ are the vertices of a mean (neutral) shape, $\mathbf{W}_{1:N} \subset \mathbb{R}^{3\times K}$ are reconstruction matrices, and $\boldsymbol{s} \in \mathbb{R}^K$ is a low dimensional embedding of the vertex set, with $K \ll 3N$. $\boldsymbol{s}$ is the vector of parameters that control the shape deformations.

In order to estimate the scale σ, 3D rotation matrix $\mathbf{R}$, and 3D translation vector $\boldsymbol{t}$, between the input (arbitrarily viewed) face and a frontal view of the same face, we align the predicted set of landmarks, $\boldsymbol{X}_{1:J}$ with a corresponding set $\overline{\boldsymbol{V}}_{1:J}$ which is a subset of $\overline{\boldsymbol{V}}_{1:N}$ just defined. Clearly, the mapping between these two point sets doesn't hold exactly. We have

$$\overline{\boldsymbol{V}}_j = \sigma \mathbf{R} \boldsymbol{X}_j + \boldsymbol{t} + \boldsymbol{e}_j, \quad \forall j \in \{1 \dots J\}. \tag{3}$$

The errors $\boldsymbol{e}_{1:J}$ embed the fact that the landmarks $\boldsymbol{X}_{1:J}$ are affected by non-rigid facial deformations as well as by localization errors. Rather than attempting to simultaneously estimate the rigid and non-rigid parameters, we propose to start by estimating the rigid parameters using robust statistics. In practice we assume that $\boldsymbol{e}_{1:J}$ are samples of a random variable $\boldsymbol{e}$ drawn from a robust probability distribution function (pdf). Then the problem is cast into maximum likelihood estimation (MLE), or equivalently into the minimization of the negative of the log-likelihood, $\min_\theta \mathcal{L}(\theta)$, with:

$$\mathcal{L}(\boldsymbol{\theta} | \boldsymbol{X}_{1:J}, \overline{\boldsymbol{V}}_j) = -\frac{1}{2} \sum_{j=1}^{J} \log P(\boldsymbol{e}_j; \boldsymbol{\theta}). \tag{4}$$

In practice, we propose to use the generalized Student-t distribution [26]:

$$P(\boldsymbol{e}; \boldsymbol{\theta}) = \int_0^\infty \mathcal{N}(\boldsymbol{e}; 0, w^{-1} \boldsymbol{\Sigma}) \mathcal{G}(w; \mu, \nu) dw \tag{5}$$

where $\mathcal{N}(\cdot; 0, \boldsymbol{\Sigma})$ is the zero-mean normal distribution and $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times 3}$ is a covariance matrix. The random latent variable $w \in \mathbb{R}^+$ is drawn from the gamma distribution $\mathcal{G}(\cdot; \mu, \nu)$, and it plays the role of a *precision*. Therefore, the samples $w_{1:J}$ of this variable characterize the landmark locations: the higher their values the more reliable the associated landmarks. The model's rigid and statistical parameters are

$$\boldsymbol{\theta} = \{\sigma, \mathbf{R}, \boldsymbol{t}, \boldsymbol{\Sigma}, \mu, \nu\}. \tag{6}$$

Direct minimization of (4) is intractable. Expectation-maximization (EM) is therefore adopted, namely the negative log-likelihood (4) is replaced with the *expected complete-data negative log-likelihood*:

$$\mathrm{E}_W[-\log P(\boldsymbol{X}_{1:J}, \overline{\boldsymbol{V}}_j, w_{1:J} | \boldsymbol{X}_{1:J}, \overline{\boldsymbol{V}}_j; \boldsymbol{\theta})]. \tag{7}$$

In practice, EM alternates between the estimation of the posteriors of the precision means, $\overline{w}_{1:J}$ and the estimation of the parameters (6) via minimization of (7) which yields [25]:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{2} \sum_{j=1}^{J} \left(\overline{w}_j \| \overline{\boldsymbol{V}}_j - \sigma \mathbf{R} \boldsymbol{X}_j - \boldsymbol{t} \|^2_{\boldsymbol{\Sigma}} + \log |\boldsymbol{\Sigma}| \right). \tag{8}$$

While it iterates, the algorithm assigns (i) high precision values to landmarks that obey a rigid transformation and (ii) low values to landmarks affected by detection errors or by non-rigid motion: therefore the contribution to (8) of the former is stronger than the contribution of the latter. Once the parameters are thus estimated, the rigid transformation is applied to $\boldsymbol{X}_{1:J}$ in order to obtain *expression-preserving* frontalized landmarks, whose coordinates are denoted $\boldsymbol{Y}_{1:J} \subset \mathbb{R}^3$, namely:

$$\boldsymbol{Y}_j = \sigma \mathbf{R} \boldsymbol{X}_j + \boldsymbol{t}, \quad \forall j \in \{1 \dots J\}. \tag{9}$$

It is now possible to fit the deformable shape model (2) to the *weighted* frontalized landmarks in order to obtain optimal values for the shape parameters, namely:

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}}{\operatorname{argmin}} \frac{1}{2} \sum_{j=1}^{J} \overline{w}_j \| \boldsymbol{Y}_j - (\overline{\boldsymbol{V}}_j + \mathbf{W}_j \boldsymbol{s}) \|^2 \tag{10}$$

This enables us to build a frontal dense map associated with the 3D mesh and to warp the pixel intensities from the input image to the frontal one. Since the shape vertices form a triangulated mesh, so do their image (2D) projections. We compute the barycentric coordinates of each pixel that lies inside a projected triangle and use these coordinates to interpolate the depth. Thus, there is a depth value associated with each face pixel, and let (a_1, a_2, A_3) be the image coordinates and the depth of 3D face point $\boldsymbol{A}$, respectively. The final step consists of warping the input-face pixel values onto the frontal view. For that purpose, we assume a scaled orthographic camera model. Each 3D face point $\boldsymbol{A}$ is rotated, scaled, translated and projected onto the input image. Once it passes a visibility check, the corresponding pixel-intensity value is copied from the input face to the frontal face. The face frontalization pipeline is illustrated with an example in Fig. 9.

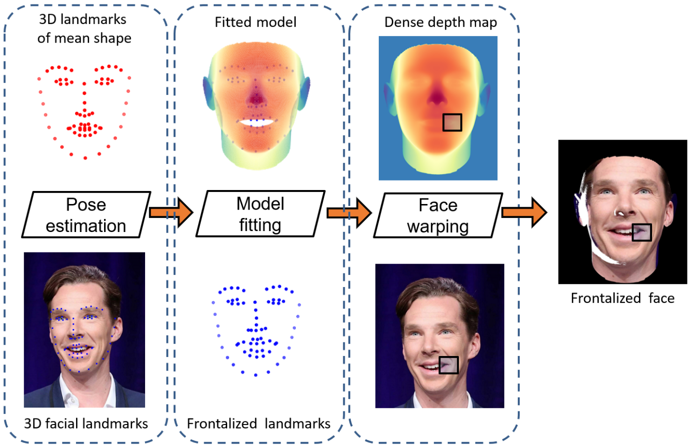


Figure 9: Overview of the proposed method. 3D landmarks extracted from a face (bottom-left) are aligned with 3D vertices associated with a frontal model (top-left). This deformable model is fitted to the frontalized landmarks (bottom-middle), yielding a deformed model aligned with the landmarks (top-middle). A dense depth map is computed by interpolating the 3D vertices of the triangulated mesh of the deformed model (top-right). This depth map is combined with the input face which is warped onto the frontal view (bottom-right). The occluded face regions are displayed in white.

We now evaluate the performance of face frontalization for the task of lip reading. The evaluation is twofold. First, we use a dataset that contains pairs of frontal and profile videos of speaking participants for a large number of subjects. The evaluation consists of computing a metric between an image obtained by face frontalization of a profile view of a speaker, with an image containing a frontally-viewed face of the same speaker. It is important that the profile and frontal images are recorded with synchronized cameras in order to capture the same expression. Consequently, the proposed evaluation is based on image-to-image comparison. Several metrics were developed in the past for comparing two images, e.g. feature-based and pixel-based metrics. In this work we use the *zero-mean normalized cross correlation* (ZNCC) coefficient between two image regions, a measure that has successfully been used for stereo matching, e.g. [27]. ZNCC is invariant to differences in brightness and contrast between the two images, due to the normalization with respect to mean and standard deviation. Second we use a lip-reading network in conjunction with a dataset that contains short videos of speakers that utter a single word, together with the ground-truth annotations (word label). We devise an experimental protocol that measures the effect of face frontalization on the word classification score.

In order to evaluate the performance of the proposed frontalization method and to compare it with state-of-the-art methods, we used a publicly available dataset, namely the OuluVS2 dataset [28]. This dataset targets the understanding of speech perception, more precisely, the analysis of non-rigid lip motions that are associated with speech production. The dataset was recorded in an office with ordinary (artificial and natural) lighting conditions. The recording setup consists of five synchronized cameras (2 MP, 30 FPS) placed in different points of view: 0°, 30°, 45°, 60°, 90°.

Method	Principle	ZNCC
Hassner et al. [29]	2D-to-3D fitting + symmetry	0.780
Banerjee et al. [17]	2D-to-3D fitting + symmetry	0.739
Zhou et al. [20]	2D-to-3D fitting + GAN	0.801
Yin et al. [30]	2D-to-2D mapping using GAN	0.773
Proposed	3D-to-3D robust fitting	**0.841**

Table 1: Mean ZNCC coefficients for 15 participants from the OuluVS2 dataset. ZNCC lies in the interval $[0, 1]$.

The dataset contains 5 × 780 videos recorded with 53 participants. Each participant was instructed to read loudly several text sequences displayed on a computer monitor placed slightly to the left and behind the 0° (frontal) camera. The displayed text consists of digit sequences, e.g. "one, seven, three, zero, two, nine", of phrases, e.g. "thank you", "have a good time", and "you are welcome", as well as of sequences from the TIMIT dataset, e.g. "agricultural products are unevenly distributed". While participants were asked to keep their head still, natural uncontrolled head movements and body position changes were inevitable. As a consequence the actual head pose varies from one participant to another and there is no exact match between the head and camera orientations.

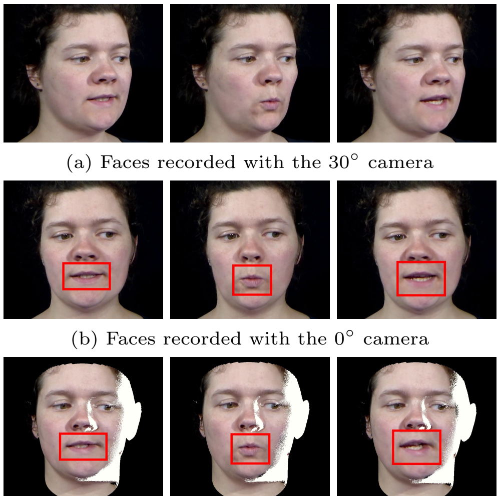
Part.	Yaw	[29]	[17]	[20]	[30]	Prop.
#31	19.1	***0.905***	0.856	0.822	0.875	**0.927**
#01	23.5	***0.915***	0.893	0.884	**0.921**	0.909
#02	24.9	0.888	0.878	***0.929***	0.881	**0.956**
#10	29.0	0.805	***0.812***	**0.873**	0.792	***0.812***
#23	30.0	0.810	**0.857**	0.819	0.817	***0.847***
#27	32.9	0.685	**0.852**	***0.824***	0.772	0.787
#19	37.8	***0.752***	0.650	0.662	0.677	**0.755**
#12	38.5	0.731	0.713	***0.755***	0.683	**0.770**
#21	40.6	0.632	***0.743***	0.653	0.673	**0.766**
Mean		0.791	0.801	***0.802***	0.787	**0.836**

Table 2: ZNCC scores for nine participants as a function of estimated yaw angle (in degrees) that corresponds to the horizontal head orientation computed with the proposed 3D head-pose estimator. For each participant, the best scores are in **bold** and the second best are in ***slanted bold***.

In practice, we evaluated the performance of the proposed method and we compared it with four state-of-the-art methods for which the code is publicly available, [29, 17, 20, 30]. We applied the frontalization to images extracted from the videos recorded with the 30° camera (I_p) and compared the results with the "ground-truth", namely the corresponding images extracted from the videos recorded with the 0° camera (I_t). Notice that videos recorded with higher viewing angles, i.e. 45°, 60° and 90°, can be hardly exploited by a frontalization algorithm because half of the face is occluded. For each frontalized image I_f we extract the mouth region R_f and we search in the associated ground-truth image I_t for the best-matching region R_t. This provides a ZNCC coefficient for each query image I_p. Notice that ZNCC only cares about the horizontal and vertical shifts in the image plane and assumes that the frontalized face and the corresponding ground-truth frontal face share the same scale. In practice, different frontalization algorithms output faces at different scales. For this reason and for the sake of fairness, prior to applying to estimating the ZNCC, we extract facial landmarks from both the frontalized and ground-truth faces and we use a subset of this set of landmarks to estimate the scale factor between the two faces.

We randomly selected 30 video pairs, recorded with the 30° and 0° cameras, respectively, associated with 15 participants from the OuluVS2 dataset. Each video contains 160 images, hence there are $30 \times 160 = 4800$ image pairs in our benchmark. The mean ZNCC coefficients obtained with two state-of-the-art methods and with the proposed method are displayed in Table 1. [29] uses soft symmetry (occluded pixels are replaced with mirror-symmetric ones).

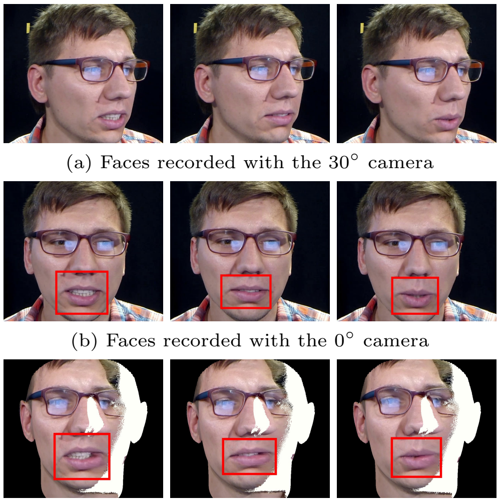
We noticed that there were important discrepancies in method performance across participants. In order to better understand this phenomenon, we computed the mean ZNCC coefficients for nine participants and displayed these means as a function of the yaw angle, i.e. horizontal head rotation estimated with the proposed method, Table 2. One may notice that there is a wide range of yaw angles, from 19° to 40°, and that the performance gracefully decreases as the yaw angle increases. The proposed method yields results that are more consistent than the other methods, as the yaw angle increases.Examples of face frontalization obtained with our method are shown on Figure 10 and Figure 11. The ZNCC correlation scores correspond the mouth region, shown in red.


(a) Faces recorded with the 30° camera

(b) Faces recorded with the 0° camera

(c) Proposed method (self-occluded facial features are displayed in white)

Figure 10: Frontalization examples for participant #02 from the OuluVS2 dataset. The ZNCC scores correspond to the mouth bounding boxes shown in red.


(a) Faces recorded with the 30° camera

(b) Faces recorded with the 0° camera

(c) Proposed method (self-occluded facial features are displayed in white)

Figure 11: Frontalization examples for participant #21 from the OuluVS2 dataset.. The estimated horizontal head orientation (yaw angle) is of 40° in this case.

Testing Training	[29]	[20]	[30]	[31]	Prop.
Pre-trained model [31]	68	60	20	88	81
Fine-tuned model [31]	83	79	37	94	93

Table 3: The effect of frontalization on the word classification score (in %) for the 100-IWR task. *First row:* word recognition scores obtained with several frontalization methods incorporated in the lip-reading model of [31]. *Second row:* word recognition scores obtained with several frontalization methods incorporated in the lip-reading model of [31] which was previously fine-tuned with frontalized faces obtained with the proposed method.

We also evaluated the ability of our method to improve the performance of lip reading. For this purpose, we used an isolated word recognition (IWR) task. The LRW (lip reading in the wild) dataset [32] consists of half a million videos of 500 English words uttered by 1000 different speakers. Each video has 29 frames and each target word is surrounded by context words. There are large inter-speaker variations in head motion. At the best of our knowledge, the best performing methods for this 500-IWR task are based on the temporal convolutional network (TCN) model of [23, 31, 33], with a word classification score of 88.5. These lip-reading model variants use the same built-in frontalizer for training and for testing, namely a 3D-to-3D affine transformation that maps the input face onto a frontal view of a generic face model. In the experiments described below we use the implementation of [31], available online.

We experimented with [32] and with [31] in the following way (please refer to the results reported in

Table 3). We randomly selected 100 words and associated videos from LRW dataset. We modified the linear transformation layer of the TCN architecture and the softmax layer to build a 100-IWR classifier. Using the model trained with [31], we tested the performance of various frontalization methods, namely [29, 20, 30], the frontalization used in [31] for training, as well as the one proposed in this paper. We also fine-tuned [31], where we replaced their affine frontalization with the proposed expression-preserving frontalization, such as to synthesize frontal faces for each speaker from arbitrarily-viewed faces. We consider the same subset of 100 words as above. For each word category we used 150 videos for training, 20 videos for validation and 20 videos for testing. The fine tuning uses the Adam optimizer and a cosine scheduler for the learning rate. This fine-tuned model was then used to test the frontalization methods mentioned above. The results in Table 3 show that the fine-tuned model using the proposed face frontalization increases the state-of-the-art performance, e.g. from 88% to 94%.

We proposed a face frontalization method that preserves non-rigid facial deformations. This stays in contrast with several state-of-the-art frontalization methods that are designed to boost the performance of face recognition by predicting as-neutral-as-possible frontal faces. We conducted a series of experiments in order to analyze the effect of frontalization on the task of visual speech recognition, whose success heavily relies on the analysis of non-rigid mouth motions, i.e. lip reading. For this purpose, we used two datasets.

The first dataset, OuluVS2, consists of multiple-view videos of speakers collected in a controlled laboratory environment. We designed an evaluation pipeline that consists of measuring the zero-mean normalized cross-correlation (ZNCC) score between a frontalized face and a frontal view of the same face. To this end, the mouth regions of the two images to be compared are aligned such that ZNCC is maximized. We compared our method with four state-of-the-art methods that use various geometric and DNN models. This benchmark reveals that the proposed method better preserves the shape of the mouth by a significant margin.

The second dataset, LRW, consists of videos collected from TV programs, that contain persons uttering speech from a catalog of 500 English words. Unlike the OuluVS2 participants, who are instructed to keep their heads still, the LRW dataset contains in the wild recordings: the speakers have large and unexpected head motions. We plugged our frontalization model into a DNN-based lip reading framework and we thoroughly analyzed its effect on the word classification scores. In the light of these experiments, we concluded that these scores are improved significantly. For example, augmenting the LRW training set with frontalized videos predicted by our method, and fine-tuning the lip-reading network with these videos, increases the classification scores of all tested methods, Table 3.

In the future, we plan to add a temporal model to our frontalization pipeline and to extend its use to other tasks such as emotion recognition, visual speech reconstruction and audio-visual speech enhancement. We believe that visually-augmented speech technologies could be extremely useful in noisy and reverberant acoustic environments.

4.2 Multi-modal Expression Recognition

In a hospital environment, every patient continuosly goes under medical care and her/his emotional state can vary a lot during the day. Therefore, it is necessary to correctly understand the patient's emotional state, such that robots can infer and interpret human emotions in a more effective way when interacting with people. The objective of task **T4.3** (*Multi-modal Affect & Robot Acceptance Analysis*) is to develop technologies for analyzing the affective state of the user(s) interacting with the robot, and specifically the level of acceptance of the robot. In particular, a state-of-the-art algorithm for emotion recognition from the extracted sources needs to be implemented to meet the requirements of the target scenario. The emotional state of the user, in combination with other features (e.g., user distance from the robot, gender/age of the user, etc.), will then be used to develop and implement an approach for automatically predicting in real-time the level of acceptance of the robot for the target user.

In this section we describe our ongoing research on emotion recognition, where the objective is to infer the human emotional state. We follow a categorical model, in which emotions consist of discrete entities associated with labels. In particular, we consider 7 basic expressions, namely "happy", "sad", "anger", "surprise", "disgust", "fear", and "neutral" (see Figure 12). Other models – which are not explored here – are also possible, such as dimensional models, in which emotions are defined by continuous values of their describing features, usually represented on axes (see the recent survey [34] for more details). Considering that humans do not show their emotions only through visual or audio modalities, we decided to follow a multi-modal approach. Specifically, the features considered are:

- audio
- video
- text
- facial landmarks.

Hence, the task here is to recognize 7 basic expressions from a multi-modal input.

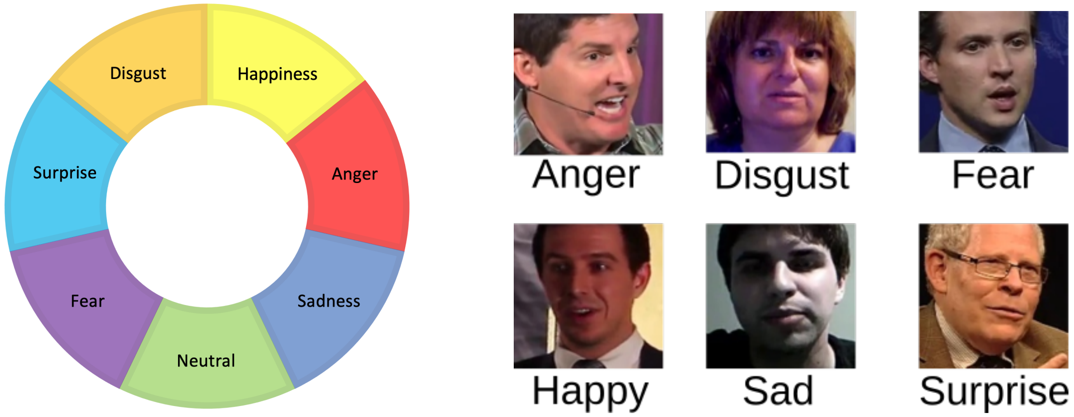


Figure 12: Left: categorical model for emotion description. Right: images showing examples of emotions.

Emotion recognition is a challenging task, in particular when performed in actual hospital environment, where the scenario may differ from the controlled environment in which most experiments are usually performed (e.g., a laboratory with stable illumination conditions). Currently, due to the lack of data that is specific to the SPRING scenario, we are evaluating our method on two publicly available data sets. The first one is RAVDESS [35], which consists of 1444 video clips acted by 24 professional actors (12 female and 12 male), where each sequence is annotated with a categorical emotion and the actors are repeating the sentence *"Kids are talking by the door"* and *"Dogs are sitting by the door"*, simulating the different emotions. The second one is CMU-MOSEI [36], which is the largest in the wild dataset (see Figure 13 for some examples). CMU-MOSEI contains more than 23,500 sentence utterance videos from more than 1000 online YouTube speakers. Each sequence is annotated with the intensity for each emotion, therefore it is possible that multiple emotions are present at the same time.

Current methods in multi-modal emotion recognition try to explore the relations between features from different modalities using different approaches. For example, in [37] an attention mechanism between modalities is used to further explore correlations, whereas the [37] exploits the modality relation by combining them together in multiple ways and then fusing them together again. Another approach that is worth mentioning is by Ghaleb et al. [38], where a specific *MSE* loss between the modalities embedding is used to better align the extracted features.

Inspired by the recent success in image classification made by [39] and [40], we decided to address the emotion recognition problem using a decoder-encoder architecture. Hence, as shown in Figure 14, each modality is processed independently trough a separate backbone:

- the video frames are processed with an R(2+1)D [41] pretrained on Kinetics 400 [42];
- the audio signal is processed to extract the mel-spectogram and then processed using a TasNet [43];
- the text is pre-processed to extract the GloVe embeddings [44] and the sequence is then analyzed trough a Transformer [45];

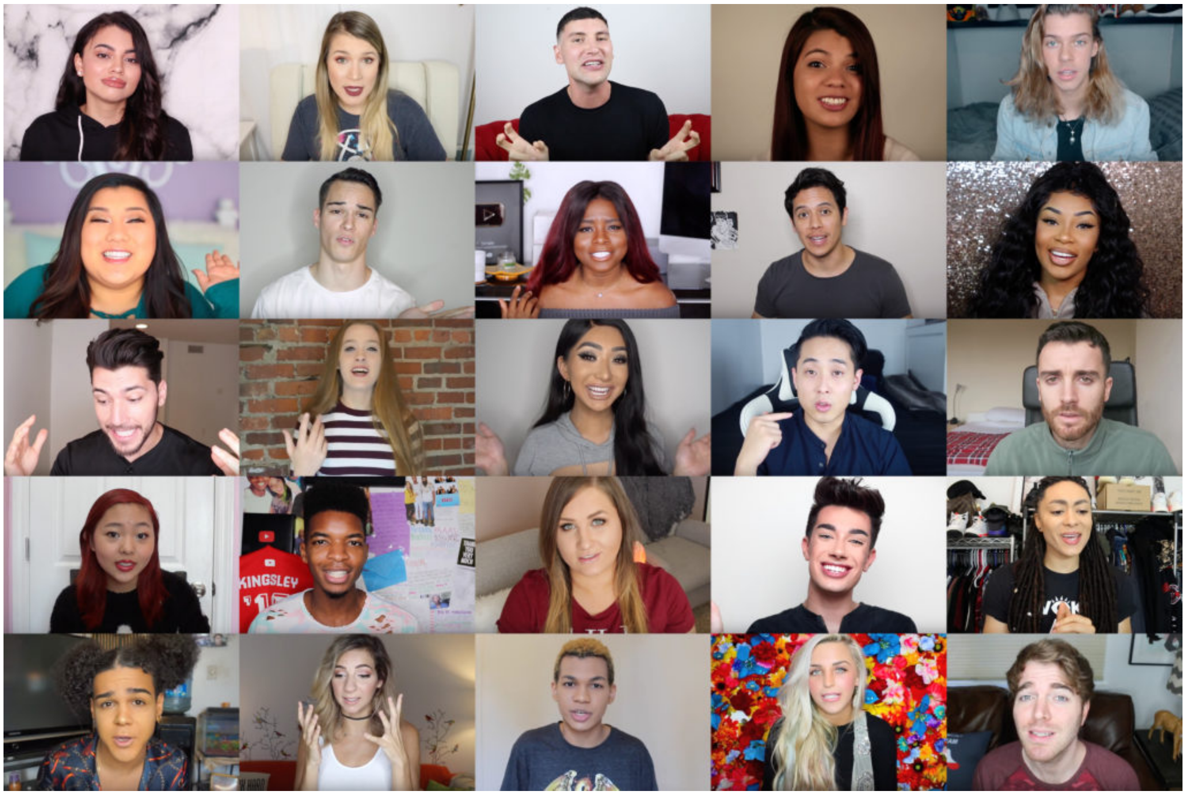


Figure 13: Samples images from the CMU-MOSEI dataset [36].

- for the facial landmarks, the sequence is processed with a Spatio-Temporal Graph Neural Network [46].

Then, to effectively combine the features, each modality is projected using an MLP and contrasted with the other modalities using the unsupervised contrastive loss:

$$\mathcal{L} = -\sum_{i \in I} log \frac{exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in A(i)} exp(z_i \cdot z_a/\tau)} \tag{11}$$

where z denotes the embedding of the projection layers, $\cdot$ is the inner dot product, τ is the temperature (scalar), $A(i)$ represents all the positive and negative embeddings, I is the set of the augmented samples and $j(i)$ is the index of the of the other samples and modalities that are generated from the same source.

The idea behind our approach is to push together the embeddings of the same emotion while pushing away the other emotions, and at the same time obtain an inter-modality feature alignment thanks to the contrastive loss performed between different modalities. As done in [46], the loss is performed over the modality projection layers, whereas the features extracted for the emotion recognition come from previous layers: this allows the network to produce features that are still aligned between modalities but are more significant from a feature information perspective. Consequentially, the features extracted are concatenated and processed through a prediction layer that outputs the emotion probabilities. The last layer is trained either with a *Cross Entropy Loss* (for RAVDESS [35]) or a *BCE Loss* (for CMU-MOSEI [36]).

In order to evaluate on CMU-MOSEI, we report both the weighted accuracy (WAcc) and the weighted F1-score for each separate emotion, as proposed in [47]. The weighted accuracy is defined as follows:

$$\text{WAcc} = \frac{TP * N/P + TN}{2N} \tag{12}$$

where TP (resp. TN) denotes the true positive (resp. true negative) predictions, and P (resp.N) denotes the total number of positive (resp. negative) examples. In order to evaluate on RAVDESS, we only report the

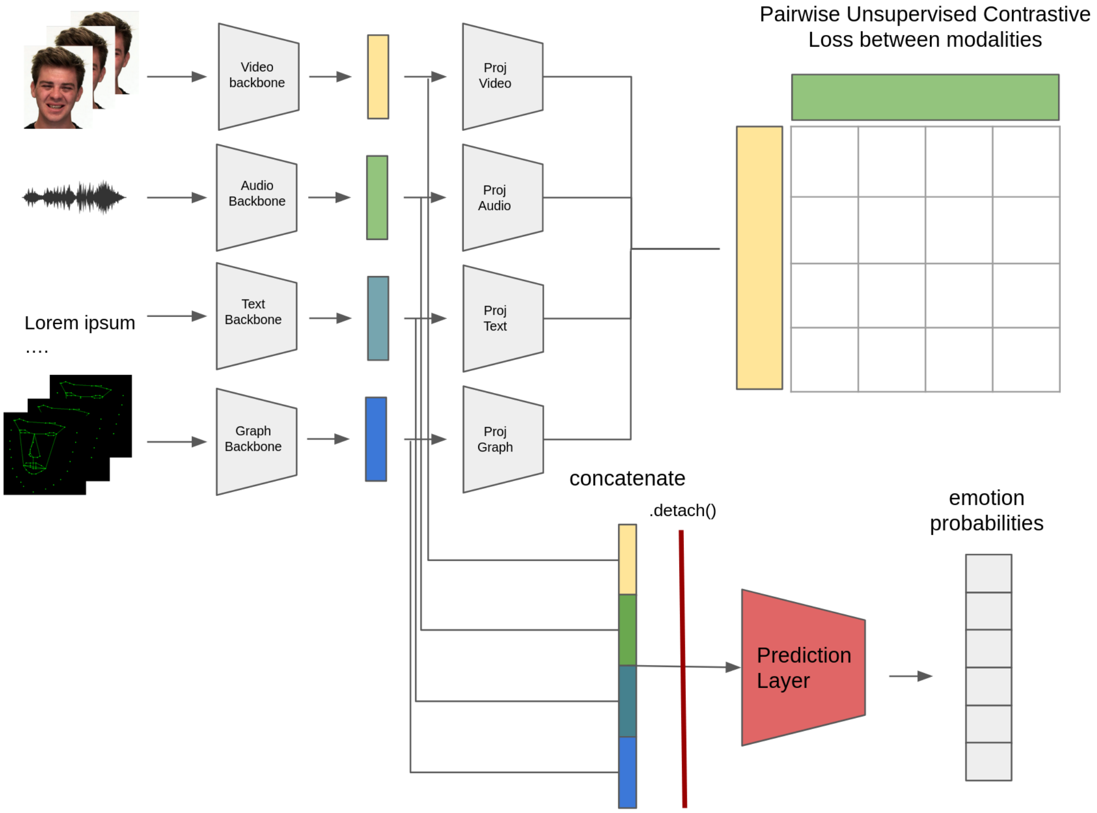


Figure 14: Proposed architecture for expression Recognition.

overall accuracy as done in [38],[48]. Considering that such a dataset has no standard splits for training and testing, we split the dataset in two ways: the first leaves out some actors for the training set, while the other performs a non actor dependent split between training and evaluation, where both splits are done using 90% for training and 10% for testing as in [48].

	Happy		Sad		Anger		Surprise		Disgust		Fear	
	WAcc	F1	WAcc	F1	WAcc	F1	WAcc	F1	WAcc	F1	WAcc	F1
Graph-MFN[36]	66,3	66,3	60,4	66,9	62,6	72,8	53,7	85,5	69,1	76,6	62	89,9
LF+MHA	61,27	61,61	55,797	36,09	54,92	37,06	50,34	5,66	55,84	44,15	57,25	43,71
MESM [37]	64,1	-	63,00	-	66,8	-	65,70	-	75,6	-	65,8	-
ours	68,43	68,81	62,3	55,84	66,17	69,93	60,64	67,79	71,76	73,27	63,51	75,33

Table 4: Performance of several methods on CMU-MOSEI [36].

Table 4 shows the performance of several methods on CMU-MOSEI. We consider the following baselines: Graph-MFN [36], which uses a Dynamic Fusion Graph (DFG) to fuse the different modalities studying also the cross-modal interactions; MESM [37], which uses sparse CNN and cross attention modules to predict the emotions; and LF+MHA which is a late fusion approach with a multi-head attention module over the fused features. In Table 5 performances on RAVDESS are reported. Concerning our method, we do not consider text features (since actors are repeating the same sentence), but audio and landmarks only. We plan to investigate the usage of video features as well in the future. Here we consider the method by Ghaleb et al. [38], which uses a separate network for processing audio and video, and then regularizes the two modalities with a temporal loss that acts over the sequence. Another method that we report is Lr-Grin [48] which usse a Spatio-Temporal Graph Network that is capable of learning the optimal adjacency matrix to better exploit the data. Moreover, we report some experiments that we have conducted using an i3d [49] on facial images and also performance

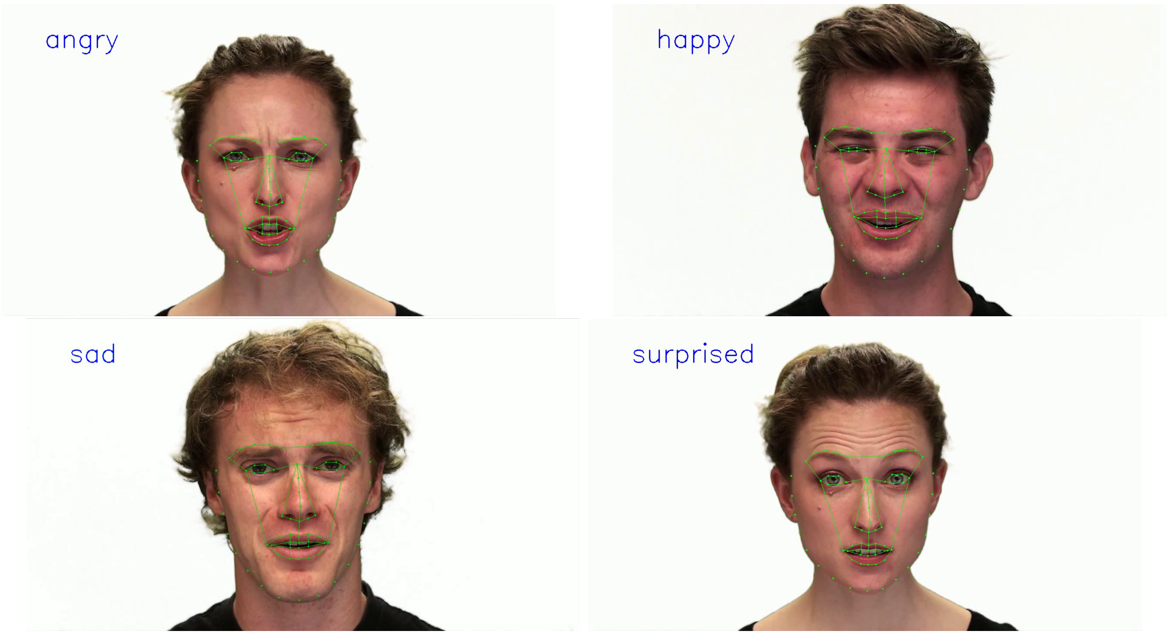


Figure 15: Examples of emotions predicted on RAVDESS

	Actor split	Video	Audio	Graph	Acc (%)
Ghaleb et al [38]	✓	✓	✓		67.7
i3d [49]	✓	✓			60.8
TCN [43]	✓		✓		58,5
ours	✓		✓	✓	**75.1**
Lr-Grin [48]				✓	85.65
STGCN [46]				✓	76.35
TCN [43]			✓		75,9
ours			✓	✓	**87,15**

Table 5: Performance of several methods on RAVDESS [35].

evaluation of models that use only the audio (TCN) [43] or graph (STGCN) [46] as input. Some anecdotal example of RAVDESS can be seen in Figure 15.

From Tables 4 and 5 we notice that our model is capable of performing on a par with the state of the art. In particular, on RAVDESS there is a solid performance improvement. On the other hand, in more challenging scenarios like CMU-MOSEI, our method is better than the baselines in some situations but improvement is not as evident as in RAVDESS.

Our preliminary experiments showed that the proposed approach is promising on standard datasets, especially considering we are working in an unsupervised setting. Future work will continue to explore the advantages of our unsupervised contrastive fusion method. In particular, we believe that an important aspect of our method is that it allows each modality to be deeply exploited without being influenced by other modalities. Therefore, our model is expected to be more robust to the problem of non relevant modalities in a real world scenario, and to be capable of obtaining good performance with less input modalities.

5 Conclusion

In this document we described the current status of the "describing humans" task (**T4.1**). In particular, we implemented face detection module, mask detection module, face verification & facial biometric (age and gender) estimation module, and human pose estimation module. For face detection, we rely on MTCNN [1] library given its good performance and fast speed, as it can detect faces in videos in real time. Besides, we have also included a face mask detection module derived from SSD [2] model, but with a lighter backbone. This task has recently gained great relevance due to the Covid-19 pandemic, which demands people to wear face masks as a protection measure. For the face verification module, we used knowledge distillation technique to learn a light student network from the teacher network (*i.e.*, FaceNet [7]). Based on the learned student network, we also added two extra classifiers for age and gender estimation. Therefore, face verification, age and gender estimation are implemented using one single network. For human pose estimation, we rely on [13] which uses a nonparametric representation referred to as Part Affinity Fields (PAFs) which has a good balance between inference time and the accuracy. To sum up, we provided a first version of several functional modules which have been integrated with ROS. We plan to further elaborate these modules during next months, with the aim of improving their performance.

Due to delays in data collection and ARI robot delivery, we have not started yet to test our algorithms on real-time platforms in relevant environments. Instead, we tested the proposed methods on publicly available datasets only. In the future we will conduct experiments on the robotic platform in scenarios which are relevant for this project. In this context, we plan to use domain adaptation techniques in order to exploit both synthetic data from **T2.2** and publicly available datasets, thus avoiding the need of annotating novel data. The use of audio signals as well as the information derived from the 3D semantic maps of **WP2** will be investigated in order to improve the recognition accuracy of human pose estimates and of soft-biometric patterns. Besides, even though we have integrated the algorithm into ROS, we can not run it now as our dependent modules from our partner, such as the tracking module from Section 3.2, is not ready yet.

In order to compensate for delays in the experiments, we started working in advance on task **T4.3**, which is supposed to start on **M22**. In particular, we presented in this document our ongoing research on emotion recognition via contrastive learning. In the meanwhile, we are also communicating with other partners and planning to integrate different modules together, such as the face frontalization module from INRIA. In particular, we will investigate whether face frontalization could help, e.g., age&gender estimation or emotion recognition. We will also integrate our modules with the tracking module from **WP3**.

References

[1] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[4] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[7] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[10] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[12] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017.

[13] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[14] Lukežič Alan, Tomáš Vojíř, Luka Čehovin, Jiří Matas, and Matej Kristan. Discriminative correlation filter tracker with channel and spatial reliability. *International Journal of Computer Vision*, 126(7):671–688, 2018.

[15] Junho Yim, Heechul Jung, ByungIn Yoo, Changkyu Choi, Dusik Park, and Junmo Kim. Rotating your face using multi-task deep neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 676–684, 2015.

[16] Xiangyu Zhu, Zhen Lei, Junjie Yan, Dong Yi, and Stan Z Li. High-fidelity pose and expression normalization for face recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 787–796, 2015.

[17] Sandipan Banerjee, Joel Brogan, Janez Krizaj, Aparna Bharati, Brandon Richard Webster, Vitomir Struc, Patrick J Flynn, and Walter J Scheirer. To frontalize or not to frontalize: Do we really need elaborate pre-processing to improve face recognition? In *IEEE Winter Conference on Applications of Computer Vision*, pages 20–29, 2018.

[18] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen, Junliang Xing, et al. Towards pose invariant face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216, 2018.

[19] Erjin Zhou, Zhimin Cao, and Jian Sun. Gridface: Face rectification via learning local homography transformations. In *Proceedings of the European Conference on Computer Vision*, 2018.

[20] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5911–5920, 2020.

[21] Adriana Fernandez-Lopez and Federico M Sukno. Survey on automatic lip-reading in the era of deep learning. *Image and Vision Computing*, 78:53–72, 2018.

[22] Ahsan Adeel, Mandar Gogate, Amir Hussain, and William M Whitmer. Lip-reading driven deep learning approach for speech enhancement. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2019.

[23] Brais Martinez, Pingchuan Ma, Stavros Petridis, and Maja Pantic. Lipreading using temporal convolutional networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6319–6323, 2020.

[24] Shiyang Cheng, Pingchuan Ma, Georgios Tzimiropoulos, Stavros Petridis, Adrian Bulat, Jie Shen, and Maja Pantic. Towards pose-invariant lip-reading. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4357–4361, 2020.

[25] Zhiqi Kang, Mostafa Sadeghi, and Radu Horaud. Face frontalization based on robustly fitting a deformable shape model to 3d landmarks, 2021. IEEE Transactions on Multimedia, under review.

[26] Florence Forbes and Darren Wraith. A new family of multivariate heavy-tailed distributions with variable marginal amounts of tailweight: application to robust clustering. *Statistics and Computing*, 24(6):971–984, 2014.

[27] Changming Sun. Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47(1-3):99–117, 2002.

[28] Iryna Anina, Ziheng Zhou, Guoying Zhao, and Matti Pietikäinen. OuluVS2: a multi-view audiovisual database for non-rigid mouth motion analysis. In *International Conference on Automatic Face and Gesture Recognition*, volume 1, pages 1–5. IEEE, 2015.

[29] Tal Hassner, Shai Harel, Eran Paz, and Roee Enbar. Effective face frontalization in unconstrained images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4295–4304, 2015.

[30] Yu Yin, Songyao Jiang, Joseph P Robinson, and Yun Fu. Dual-attention GAN for large-pose face frontalization. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 24–31. IEEE Computer Society, 2020.

[31] Pingchuan Ma, Brais Martinez, Stavros Petridis, and Maja Pantic. Towards practical lipreading with distilled and efficient models. *arXiv preprint arXiv:2007.06504*, 2020.

[32] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, pages 87–103, 2016.

[33] P. Ma, Y. Wang, J. Shen, S. Petridis, and M. Pantic. Lip-reading with densely connected temporal convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, 2021.

[34] Matteo Spezialetti, Giuseppe Placidi, and Silvia Rossi. Emotion recognition for human-robot interaction: Recent advances and future perspectives. *Frontiers in Robotics and AI*, 7:145, 2020.

[35] Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), April 2018. Funding Information Natural Sciences and Engineering Research Council of Canada: 2012-341583 Hear the world research chair in music and emotional speech from Phonak.

[36] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[37] Wenliang Dai, Samuel Cahyawijaya, Zihan Liu, and Pascale Fung. Multimodal end-to-end sparse model for emotion recognition. *CoRR*, abs/2103.09666, 2021.

[38] Esam Ghaleb, Mirela Popa, and Stylianos Asteriadis. Multimodal and temporal perception of audio-visual cues for emotion recognition. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 552–558, 2019.

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[40] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[41] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *CoRR*, abs/1711.11248, 2017.

[42] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[43] Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. *CoRR*, abs/1711.00541, 2017.

[44] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[46] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting. *CoRR*, abs/1709.04875, 2017.

[47] Edmund Tong, Amir Zadeh, Cara Jones, and Louis-Philippe Morency. Combating human trafficking with multimodal deep models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1547–1556, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[48] Amir Shirian, Subarna Tripathi, and Tanaya Guha. Dynamic emotion modeling with learnable graphs and graph inception network. *IEEE Transactions on Multimedia*, pages 1–1, 2021.

[49] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. *CoRR*, abs/1705.07750, 2017.