applied sciences

MDPI

Article

Face Recognition and Tracking Framework for Human–Robot Interaction

Aly Khalifa *, Ahmed A. Abdelrahman, Dominykas Strazdas, Jan Hintz, Thorsten Hempel and Ayoub Al-Hamadi *

Neuro-Information Technology, Otto-von-Guericke-University Magdeburg, 39106 Magdeburg, Germany; ahmed.abdelrahman@ovgu.de (A.A.A.); dominykas.strazdas@ovgu.de (D.S.); jan.hintz@ovgu.de (J.H.); thorsten.hempel@ovgu.de (T.H.)

* Correspondence: aly.khalifa@ovgu.de (A.K.); ayoub.al-hamadi@ovgu.de (A.A.-H.)

Abstract: Recently, face recognition became a key element in social cognition which is used in various applications including human–robot interaction (HRI), pedestrian identification, and surveillance systems. Deep convolutional neural networks (CNNs) have achieved notable progress in recognizing faces. However, achieving accurate and real-time face recognition is still a challenging problem, especially in unconstrained environments due to occlusion, lighting conditions, and the diversity in head poses. In this paper, we present a robust face recognition and tracking framework in unconstrained settings. We developed our framework based on lightweight CNNs for all face recognition stages, including face detection, alignment and feature extraction, to achieve higher accuracies in these challenging circumstances while maintaining the real-time capabilities required for HRI systems. To maintain the accuracy, a single-shot multi-level face localization in the wild (RetinaFace) is utilized for face detection, and additive angular margin loss (ArcFace) is employed for recognition. For further enhancement, we introduce a face tracking algorithm that combines the information from tracked faces with the recognized identity to use in the further frames. This tracking algorithm improves the overall processing time and accuracy. The proposed system performance is tested in real-time experiments applied in an HRI study. Our proposed framework achieves real-time capabilities with an average of 99%, 95%, and 97% precision, recall, and F-score respectively. In addition, we implemented our system as a modular ROS package that makes it straightforward for integration in different real-world HRI systems.

Keywords: face recognition; face tracking; face detection; face alignment; person identification; human–robot interaction; intelligent robots; interactive systems

Citation: Khalifa, A.; Abdelrahman, A.A.; Strazdas, D.; Hintz, J.; Hempel, T.; Al-Hamadi, A. Face Recognition and Tracking Framework for Human–Robot Interaction. *Appl. Sci.* **2022**, *12*, 5568. https://doi.org/10.3390/app12115568

Academic Editors: Luis Gracia and Carlos Perez-Vidal

Received: 9 May 2022
Accepted: 27 May 2022
Published: 30 May 2022

1. Introduction

Robots have an increasing involvement in real-world contexts, such as homes, schools, hospitals, labs and workplaces. As a result, the field of human–robot Interaction (HRI) presents new challenges in security, automation, and recognition [1]. Robots need social intelligence to interact effectively with and assist humans. Furthermore, a reasonable difference between humans and robots is that humans can recognize and remember individuals by perceiving their facial features smoothly, while robots pose significant challenges in perception [2]. This is an essential part of social cognition and represents a key element for improving human–robot interaction. Moreover, the recent advances in face detection and face recognition (FR) through deep neural networks make it possible to make robots rapidly approach human-level performance and handle several challenging conditions, including large pose variations and occlusions, difficult lighting conditions, and poor-quality images with large motion blur [3,4]. However, there are still unresolved challenges for real-world applications to operate in unconstrained circumstances, including computing power limitations and the lack of training data for user-wise face identification.

As the field of HRI advances, the levels of interaction between humans and robotics become more complex. In order to better understand the critical aspects that influence the human–robot interaction behavior, we conducted a Wizard-of-Oz study [5] to analyze common communication intuitions of new human interaction partners. Figure 1 shows the different interactions between the subjects and the industrial robot.

Figure 1. Previous field study Wizard-of-Oz [5]. A video summary can be found here: https://youtu.be/JL409R7YQa0 (accessed on 29 May 2022).

Based on the key results and conclusions of the study, we implemented a multi-modal robotic system called "RoSA" (Robot System Assistant) [6]. This way "RoSA" tackles the challenge of intuitive and user-centered human–robot interaction by integrating different interaction streams such as speech, gesture, object, body, and face recognition.

During the interactions in RoSA, the subjects were not cooperating with the face recognition module as they needed to look down to do the required tasks efficiently as illustrated in Section 3, i.e., the face pitch angle is far away from the camera, preventing the upside camera from capturing the best face pose that fit with the face recognition module. Moreover, face recognition is viewpoint dependent for rotations about all axes (pitch, yaw, and roll) and had the worst accuracy for rotations in pitch [7] as shown in Figure 2. To make the interaction smooth and to increase recognition accuracy for this scenario, we propose a face recognition system that is improved with a tracking capability to handle the subjects' continuous changes in appearance and illumination, in addition, providing the robot with the capability to learn new faces features and recognize them in real-time to participate in social cognition.

In this paper, a typical face recognition framework enhanced with a tracking capability is built by integrating a light-weight RetinaFace-mobilenet [3] with Additive Angular Margin Loss (ArcFace) [4]. Furthermore, to improve the processing speed and accuracy, we propose a tracking algorithm that combines the tracked faces with the actual user identity to improve the recognition performance and accuracy. Finally, we packaged the proposed recognition framework as a real-time Robot Operating System (ROS) node for an easy plugin into other real-world HRI systems.

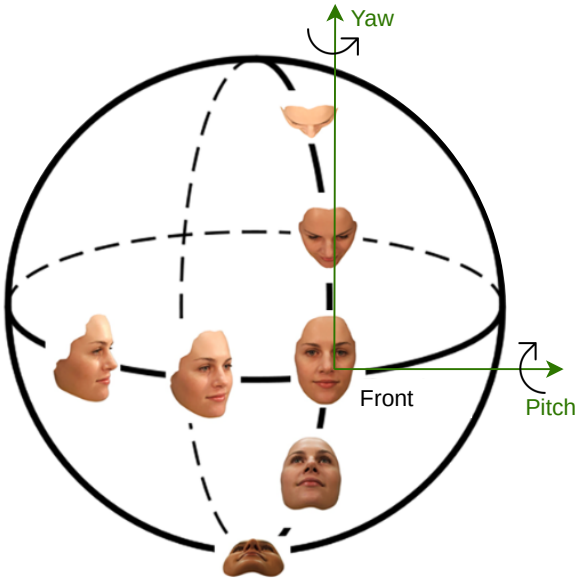

Figure 2. Influence of yaw and pitch angles variations on the head pose, showing that pitch angle have the great impact on the face features. Moving away from the front pose resultant on less distinctive features.

The remainder of the paper is organized as follows: Section 2 reviews recent related work on face detection, face alignment, face recognition, and face tracking algorithms. The RoSA system is illustrated in Section 3. Our proposed methodology and framework are presented in Section 4. Experiments and results are presented in Section 5. Finally, Section 7 concludes this paper.

2. Related Works

Most current deep face recognition systems can be decomposed into three main stages: *face detection*, where faces are localized in an input image, *face alignment*, where the detected faces are warped into a 2D or 3D canonical face model; and *face recognition*, where the aligned faces are classified into different identities. Each part has been actively studied in the field, and near-human performances have been achieved over many benchmark datasets [3,4,8]. In the following, we give a brief overview of recent works on each stage.

2.1. Face Detection Algorithms

Face detection algorithms aim to locate the main face area in input images or video frames. Furthermore, they help robots discriminate between humans and other objects in the scene.

Before the deep learning era, the cascade-based methods and deformable part models (DPM) dominated the face detection field with limitations in unconstrained face images due to considerable variations in resolutions, illumination, expression, skin color, pose, and occlusions [9].

In recent years, deep learning methods have shown their power in computer vision and pattern recognition. As a result, many deep convolutional neural networks (CNN or DCNN)-based face detection methods have been proposed to overcome the limitations mentioned above [3,10–14]. The CNN-based face detection approaches generally have two stages: a feature extraction stage by utilizing a CNN-backbone network to generate the feature map, and a stage for predicting the bounding box locations [15]. They can be divided into two categories: (1) multi-stage; and (2) single-stage detection algorithms.

Two-stage algorithms: Most two-stage algorithms are typically based on Faster R-CNN [12] and generate several candidate boxes and then refine the candidates with a subsequent stage. The first stage utilizes a sliding window to propose the candidate bounding

boxes at a given scale, and the second stage rejects the false positives and refines the remaining boxes [16–18]. The advantage of this type of model is that they reach the highest accuracy rates, on the other hand they are typically slower.

Single-stage algorithms: Most single-stage algorithms are typically based on the single shot multi-box detector (SSD) [11]. These algorithms treat object detection as a simple regression problem by performing the candidate classification and bounding box regression from the feature maps directly in only one stage, without the dependence on an extra proposal stage [3,13]. The advantage of this type of model is that they are much faster than two-stage algorithms, but they have lower accuracy rates.

Among the many variants using the single-stage structure, state-of-the-art face detection performance was achieved by RetinaFace [3]. RetinaFace is the latest one-stage face detection model, which is based on the structure of RetinaNet [19] and uses deformable convolution and dense regression loss. We utilized the lightweight version of RetinaFace based on the mobilenet backbone to enhance the detection speed to achieve real-time performance.

2.2. Facial Landmarks and Face Alignment Algorithms

Face alignment plays a vital role in many computer vision applications. It is necessary to improve the robustness of face recognition against in-plane rotations and pose variations [20]. Meanwhile, facial landmarks are essential for most existing face alignment algorithms because they are involved in the similarity transformation for finding the closest shape of the face. So, facial landmark localization is a prerequisite for face alignment.

Face alignment aims to identify the geometric structure of the detected face and calibrate it to the canonical pose, i.e., determining the location and shape of the face elements, such as the mouth, nose, eyes, and eyebrows.

From an overall perspective, face alignment methods can be divided into model-based and regression-based methods [21]. However, the regression methods show superior accuracy, speed, and robustness when compared to model-based methods [22]. Furthermore, model-based methods show difficulties to express the very complex individual landmark appearance.

Trigeorgis et al. [23] further optimize regression-based methods by introducing a single convolutional recurrent neural network architecture that combines all stages' training through facilitating a memory unit that shares information across all levels. The importance of the initialization strategies for face alignment is demonstrated in [24]. Despite that, Valle et al. [25] handled the sensitivity problem of initialization strategies by introducing the Deeply-initialized Coarse-to-Fine Ensemble (DCFE) approach. DCFE refines a CNN-based initialization stage with Ensemble of Regression Trees (ERT) to estimate probability maps of landmarks' locations. Cascade of experts is used by Feng et al. in [26] to improve the face alignment accuracy versus the different face shape poses. Feng et al. proposed Random Cascaded Regression Copse (R-CR-C) method that utilizes three parallel cascaded regressions. Furthermore, Zhu et al. [27] used a probabilistic approach to adopt coarse-to-fine shape searching.

There have been significant improvements in face alignment using deep learning methods. As in [28], Kumar and Chellapa introduced a single dendritic CNN, termed the Pose Conditioned Dendritic Convolution Neural Network (PCD-CNN). Furthermore, they combine a classification network with a second and modular classification network to predict landmark points accurately. In addition, Wu et al. [28] proposed a boundary-aware face alignment algorithm that interpolates the geometric structure of a human face as boundary lines to improve landmark localization.

In a later work, a more efficient compact model has been recently proposed by Guo et al. named practical facial landmark detector (PFLD) [29]. They used a branch of the network to estimate the geometric information for each face sample to make the model more robust. PFLD achieved a size of 2.1 Mb and over 140 fps per face on a mobile phone with high accuracy against complex faces, including unconstrained poses, expressions, lighting, and occlusions, which makes it more suitable for HRI applications.

2.3. Face Recognition Algorithms

A face recognition system is a system that can identify or verify a person in an input image or a video frame. With the current advances in machine learning, the deep face recognition systems based on the CNN models have been the most common due to their remarkable results, and several deep face recognition models have been proposed [4,30–34]. These models work by localizing the face in the input image, extracting the face embeddings, and comparing them to other face embeddings pre-extracted and stored in a database. Every embedding creates a unique face signature and the identity of a specific human face.

Taigman et al. proposed a multi-stage approach called DeepFace [30] based on AlexNet architecture [35]. The faces are first aligned to a generic 3D shape model, and then facial representation is derived from a nine-layer deep neural network. In addition, the authors used a Siamese network trained by standard cross-entropy loss for face verification. Inspired by the work of DeepFace, Sun et al. introduced a high-performance deep convolutional neural network called DeepID2+ [36] for face recognition. DeepID2+ achieved a better performance by adding supervision to early convolutional layers and increasing the dimension of hidden representations. Schroff et al. proposed FaceNet [31] based on the GoogleNet architecture [37]. FaceNet directly optimizes the face embedding by a deep convolutional network trained using a triplet loss function at the final layer. He et al. proposed a Wasserstein convolutional neural network (WCNN) approach [38] that optimizes face recognition by learning invariant features between near-infrared and visual face images.

Recently, different loss functions for face recognition have been proposed [4,32,33,39,40] to enhance discriminative feature learning and representation. Sphereface presents the importance of the angular margin and its advantage in feature separation, but the training is unstable and hard to converge. CosFace defines the decision margin in the cosine space by directly adding the cosine margin penalty to the target logit, which results in better performance than SphereFace with easier implementation and stable training. The ArcFace or Additive Angular Margin Loss [4] is one of the most potent loss functions designed for deep face recognition [41–43]. It enhances discriminative learning by introducing an additive angular margin. In contrast with SphereFace and CosFace which have a nonlinear angular margin, ArcFace has a constant linear angular margin.

The evaluation of single face recognition requires high computational power. Furthermore, multiple faces in a single scene need to be recognized and identified in practice. This makes recognizing multiple faces another challenge, as it requires more computing power to process multiple faces per scene. The accuracy and processing time are the main criteria for any face recognition system. Nevertheless, especially for the HRI, accuracy and real-time recognition are a challenge in scenes with subjects that do not co-operate with the recognition system.

2.4. Face Tracking Algorithms

Visual object tracking has always been a research hotspot in computer vision, and face tracking is a special case. Face tracking is primarily a process of determining the position of the human face in a digital video or frame based on the detected face. This is challenging as the face is not the same during the time (video frames), but it may vary in pose and view. Moreover, other factors affected the face tracking in the actual scene and made it more complex, such as illumination, occlusion, and posture changes. On the other hand, face tracking has many advantages, such as counting the number of human faces in a digital video or camera feed and following a particular face as it moves in a video stream to predict the person's path or direction. Moreover, it can reduce the processing time needed for face detection and recognition.

Many visual object tracking algorithms have been presented; however, Kalman filter [44] and template matching [45] are the most popular methods. In [46], Bewley et al. proposed simple online and real-time tracking (SORT) for multiple object tracking. SORT is a simple approach that associates objects efficiently for online and real-time applications by utilizing the Kalman filter and the Hungarian method. It achieves a favorable performance

at high frame rates of 260 Hz. In [47], Wojke et al. integrates SORT with the appearance information by employing a trained CNN to discriminate pedestrians on a large-scale person re-identification dataset, and called it Deep-SORT. This technique has improved the performance and reduced the number of identity switches through longer periods of occlusions.

Recently, deep learning-based face tracking algorithms have been dominant, where the face tracking problem is solved as a binary classification problem for predicting a face or a non-face. Lian et al. [48] proposed a multiple objects tracking algorithm that utilizes a multi-task CNN network (MTCNN) for face detection and fuses multiple features (appearance, motion, and shape features) for tracking. Despite the promising results achieved by deep learning-based face tracking algorithms, SORT has a higher frame rate with favorable accuracy due to its simplicity and ease of implementation.

3. Human–Robot Interaction System

We developed RoSA, a multi-modal system for contactless human–machine interaction based on speech, facial, and gesture recognition [6]. In order to make the interaction smooth and to increase recognition accuracy in RoSA, we propose a face recognition framework that is improved with a tracking capability to handle subjects' continuous changes in appearance and illumination.

The RoSA setup is illustrated in Figure 3, and has two workstations, workstations 1 (WS1) and workstation 2 (WS2), with different designs and purposes [6]. In addition to seven modules (face, speech, gesture, attention, robot, cube, and scene) were designed and implemented. The modules utilize the ROS, ROS network, and ROS messages for communications with the workstations and each other.

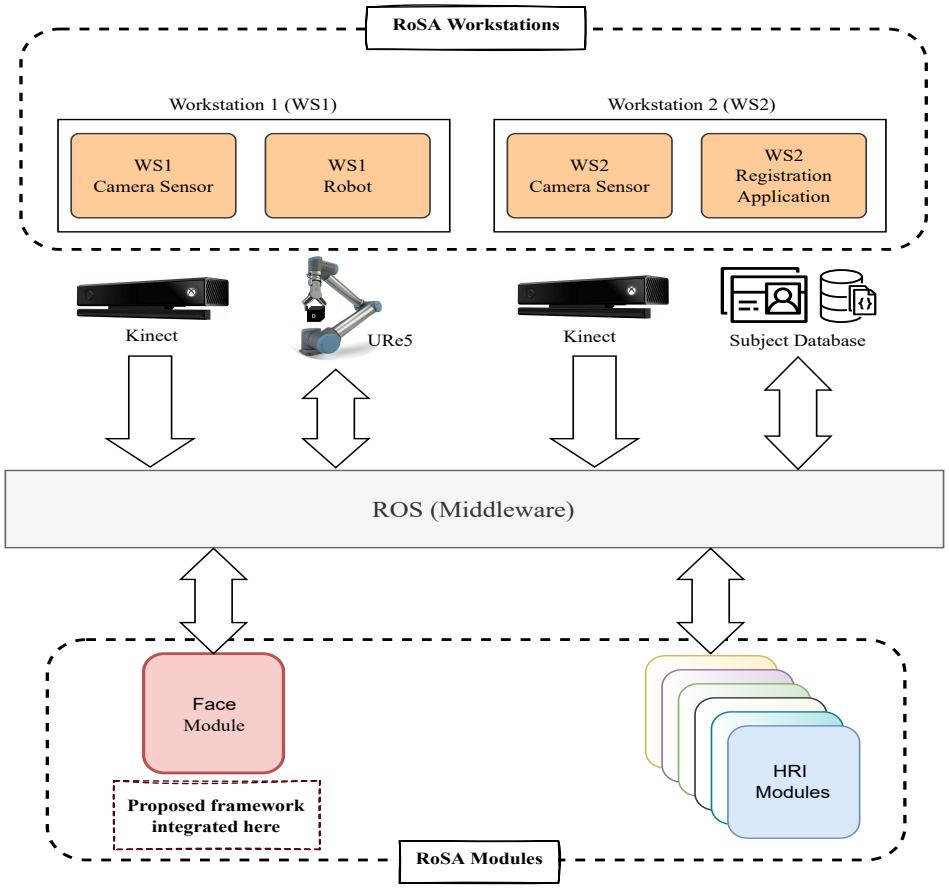

Figure 3. The system setup of the Robot System Assistant (RoSA) framework, showing the communication between the RoSA modules and workstations, is performed via the ROS, and the proposed framework is integrated as the face module.

WS1 is dedicated to all the human–robot interactions and collaborative tasks with the robot. It consists of an industrial robot *UR5e* provided with a gripper *RG6* for easy handling of the required tasks and securely fixed on a metal table. A top camera sensor is used for a live stream of all the human–robot interactions; a time of flight (ToF) Kinect V2 camera is selected for this task. A set of black and white cubes with letters are available for the tasks and under the robot's gripper control. For visual feedback, a projector was utilized to illuminate the cubes and the metal table. The primary purpose of WS2 is for subject registration, and it consists of a smart touch screen with built-in speakers.

In the experiments of the RoSA Study, the subjects enter the required information through a graphical user interface. At the same time, the face embedding is extracted by asking the subject to look at WS2 camera in frontal and profile postures. The collected information and embeddings are stored on the subject database. After the completion of the registration, RoSA asks the subject to go to WS1 to do the practical experiment and the collaborative tasks with the robot. Finally, RoSA asks the subject to answer the questionnaires at WS2. These questionnaires include evaluation questions about RoSA during the interaction. Furthermore, RoSA assists the subject to collect extra data for a module assessment and a benchmark.

An active session is required to enable the interaction between the current subject and the robot. This active session can only be achieved if the face module can effectively recognize and track the identity of the subject during the experiment. Regardless, due to the nature of the collaborative tasks and the unrestricted environment, face recognition is a challenging process and is required to handle the different lighting conditions, pose angles, partially occluded, and sometimes, completely hidden faces. This would sometimes lead to the loss of tracking and active session. The proposed face recognition system enables RoSA to recognize and track subjects robustly.

Using face tracking for user recognition and identification also improves on common problematic situations when implementing body tracking in multi-user scenarios: body tracking mix-up and false body detection in inanimate objects. While the coat hangers and office chairs do sometimes get detected as a person and assigned a body posture for further processing, it is very unlikely that the false body would also have a valid face that could also be detected. By fusing the detected faces to the detected bodies—to which we refer as "fused bodies"—we make sure that each body has a valid face for detection and thus a unique ID, determined by that face.

This approach also reduces the unintentional mix-up of tracked bodies, which occurs when two persons are standing close to each other or pass one another while restricting the view of the body tracker. After the loss of one of the tracked bodies due to occlusion or ambiguities, the body tracker estimation can jump over to the other subject and continue under the wrong ID. By constantly checking for integrity, between the user's skeleton and face with the help of the fused body, the mix-up can be detected right away and the error corrected. This way, it is sufficient to track only the face ID for interaction purposes and sort the detected bodies accordingly. After a mix-up, the information corresponding to the tracked body would be updated in the user's fused body entity, so the system would now be aware of which tracked body and its inputs correspond to the face ID.

4. Methodology and Proposed Framework

The proposed framework is a face recognition system improved with a tracking algorithm. Firstly, the current frame is fed to the face detection module to localize faces in each video frame. Then, a face tracker is created for each detected face across the video frame. Meanwhile, the detected faces are aligned to the canonical face using the detected landmarks and sent to the face recognition module. Finally, the face recognition module gets each detected face identity and associates this identity with the face tracker, and then publishes these identities to the other RoSA modules. The framework is illustrated in Figure 4 and consists of three main modules: *face detection and alignment*, *multi-face tracking*,

and *deep face recognition* modules. In the following sections, the details of each module will be discussed.

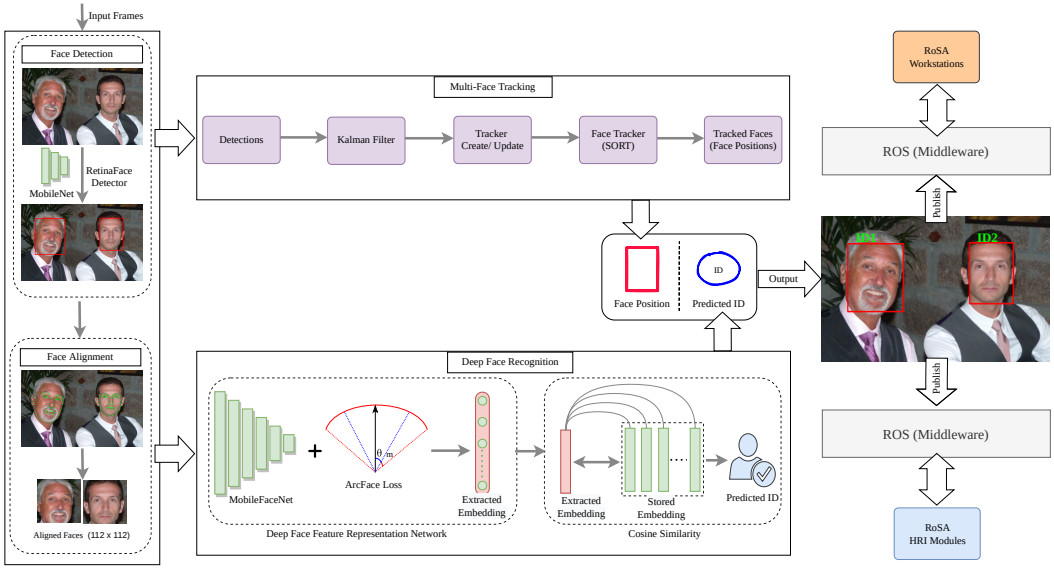

Figure 4. An overview of the proposed face recognition and tracking framework. The predicting face locations and identities are published to the ROS network for broadcasting to RoSA workstations and modules.

4.1. Face Detection and Alignment

For face detection tasks, we use a deep CNN-based face detector by employing a single-shot, multi-level face localization method, called RetinaFace [3]. RetinaFace unifies three different face localization tasks together under one single shot framework: face box prediction, 2D facial landmark localization, and 3D vertices regression. Additionally, all points for these three tasks are regressed on the image plane. RetinaFace proposes a single-shot, multi-level face localization model, which consists of three components: the feature pyramid network, the context head module, and the cascade multi-task loss. First, the feature pyramid network generates five feature maps of different scales. Then, the feature map of a particular scale is fed to the context head module to compute the multi-task loss, i.e., the first context head module predicts the bounding box from the regular anchor. Afterward, the second context head module predicts a more accurate bounding box using the regressed anchor generated by the first context head module. Finally, the anchors are matched to ground-truth boxes if the Intersection over Union (IoU) is greater than 0.7 and 0.5 for the first and second context head respectively, and are matched to the background if IoU is less than 0.3 and 0.4 for the first and second context head, respectively. Furthermore, the unmatched anchors are ignored during training. For any training anchor i, RetinaFace minimizes the following multi-task loss [3]:

$$\mathcal{L} = \mathcal{L}_{cls}(p_i, p_i^*) + \lambda_1 p_i^* \mathcal{L}_{box}(t_i, t_i^*) + \lambda_2 p_i^* \mathcal{L}_{pts}(l_i, l_i^*) + \lambda_3 p_i^* \mathcal{L}_{mesh}(v_i, v_i^*), \quad (1)$$

where t_i, l_i, v_i are box, five landmarks and 1k vertices predictions, t_i^*, l_i^*, v_i^* is the corresponding ground-truth, p_i is the predicted probability of anchor i being a face, and p_i^* is 1 for the positive anchor and 0 for the negative anchor. The classification loss $\mathcal{L}_{cls}$ is the softmax loss for binary classes (face/not face). The loss-balancing parameters λ_1 and λ_2 are set to 0.25 and 0.1, respectively.

For the face landmarks and alignment task, we use a deep CNN-based network by utilizing a practical facial landmark detector (PFLD) by Gue et al. [29]. PFLD employs a branch of the network to estimate the geometric information for each face in order to regularize the landmark localization. Moreover, it adds a multi-scale fully connected (MS-FC) layer to enlarge the receptive field, catch the global structure, and precisely localize

landmarks on faces. For predicting landmark coordinates, it utilizes the MobileNet network as a backbone to enhance the processing speed and model size. As a result, it achieved a size of 2.1 Mb and over 140 fps per face on a mobile phone with high accuracy against complex faces, including unconstrained poses, expressions, lighting, and occlusions.

In the face detection and alignment module, all the faces in the images or the video frames are detected with RetinaFace. RetinaFace outputs bounding boxes and five landmarks (2 eyes, nose, and mouth) with a confidence score. For real-time constraints, we select MobileNet-0.25 [49] as a lightweight backbone network, which achieves the real-time speed of 40 fps at GPU for 4K images (4096 × 2160) with outstanding performance.

Next, the filtered faces, i.e., the detection boxes with high confidence scores are sent to face alignment for calibrating to the canonical view and for cropping it to a size of 112 × 112 to be suitable for the subsequent task of face feature extraction. For the face landmarks and alignment task, we used the compact model of the PFLD.

4.2. Face Recognition

For the face recognition task, we utilize the additive angular margin loss (ArcFace) model by Deng et al. [4] to extract the feature embeddings of the faces. ArcFace introduces an additive angular margin penalty m between the deep feature x_i and the target weight W_{y_i} to simultaneously enhance the intra-class compactness and inter-class discrepancy. It provides a more clear geometric interpretation due to its exact correspondence to geodesic distance on a hypersphere. ArcFace is inherited from the most common loss function, Softmax, and is defined as follows [4]:

$$L_{arc} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(cos\,(\theta_{y_i}+m))}}{e^{s(cos\,\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\,cos\,\theta_j}}. \quad (2)$$

In Equation (2), n denotes the number of classes in the training database, while N denotes the batch size. ArcFace model starts with extracting the face features x_i by utilizing a DCNN backbone. The backbone network is the bottleneck in terms of processing speed and model size; as in the testing, only this branch is involved so we selected the lightweight MobileFaceNet network [50] as a backbone. Then, based on the feature x_i and weight W normalization, we obtain the logit $cos\,\theta_j$ for each class as $W_j^T\,x_i$, and get the angle between the feature x_i and the ground truth weight W_{y_i} as $arccos\,\theta_{y_i}$. After that, the angular margin penalty m is added to the target angle θ_{y_i}. Finally, we calculate $cos(\theta_{y_i}+m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross-entropy loss. The results of the ablation study by Deng et al. [4] showed that the performance comparison on the LFW, CALFW, and CPLFW datasets for the Arcface loss function outperformed others with 99.82%, 95.45%, and 92.08% accuracies respectively. It was performed against 11 other loss functions, including Softmax, Center Loss, SphereFace, and CosFace. This is the main reason why we selected Arcface as a loss function for the face recognition module.

In the face recognition module, after the filtered faces are aligned, a deep face feature representation network transforms the aligned faces into a feature space. MobileFaceNet [50] was selected as a backbone for this task to handle the real-time constraints. Loss function optimization is challenging for large-scale face classification, as it is needed to strengthen the intra-class compactness and inter-class discrepancy for highly similar individual faces. For that, we used ArcFace as it outperforms the state-of-the-art functions. In addition, it enhances the discriminative power for learning deep features and maximizes the separability between face classes.

Finally, the face recognition module outputs a 512-dimensional feature embedding, and then the predicted identity is calculated by comparing the generated embedding against the stored embeddings by calculating the cosine similarity [51]. The ArcFace model is trained on the MS1M database [14]. Given a face image, the image is aligned, scaled,

and cropped before being passed to one of the models. This preprocessing is performed as described in [13] for ArcFace.

4.3. Improved Face Recognition Using Face Tracking

For the face tracking task, we build the face tracking algorithm based on a simple online and real-time tracking algorithm (SORT) [46]. SORT uses a Kalman filter for estimating the location of the face in the current frame given the location in the previous frame. It starts with detecting the target face in the initial frame i. After that, predict the future location $i+1$ of the target face from the initial frame using the Kalman filter. Noting that the Kalman filter just approximates the face's new location, which needs to be optimized. Finally, the Hungarian algorithm is used for face location optimization and association.

The main problem we are targeting is the speed/accuracy trade-offs. Continuous face detection and face recognition processing are time-consuming. Moreover, the quality of the face features depends on the face pose, where the frontal face pose generates the best facial features and degrades in a departure from the frontal pose. Therefore, instead of detecting all faces around all input video frames, we assign only each newly detected face a tracker and start the tracking instead of detection. Furthermore, for each new tracker only, the face embedding will be inferred and compared against the stored embeddings by calculating the cosine similarity to generate the user identity (ID), then add the ID to the tracker metadata for fast recognition, i.e., retrieve the ID from the tracker in the successive frames without the need for recognition. These will improve the processing time, recognition rate, and reduce the recognition errors caused by variations from frontal face poses.

In the proposed tracking Algorithm 1, for each input frame, we are detecting faces using face detection and alignment in Section 4.1. Initially, a new tracker for each detector box will be created by applying SORT [46]. SORT analyzes previous and current frames and predicts face locations on the fly by utilizing the Kalman filter and Hungarian algorithm. Then, the user ID will be obtained using face recognition in Section 4.2 and assigned with the face tracker for use in fast recognition in the further frames. Finally, the tracker will be associated with the detected faces and maintained throughout tracking, and the user ID is assigned for each face tracker. We update the tracker in each frame to validate if a face is there inside the box to improve the tracking quality. If not, we are deleting the tracker to prevent unbounded growth in the number of trackers. Moreover, the actual user identity is attached to the face tracker instead of a unique face tracker ID to improve the face recognition speed.

Algorithm 1: The Proposed Face Tracking Algorithm.

Inputs : Video, Detections, KalmanFilter, HeadJoints, SubjectIDs
Output: Recognized Tracked Faces
Initialize $KalmanFilterTracker$;
foreach *frame* $f_i \in Video$ **do**
  $Trackers \leftarrow Predict()$;
  $Trackers \leftarrow Assign(Detections, Trackers)$;
  $TrackersID \leftarrow Attach(Trackers, SubjectIDs)$;
  $TrackersID \leftarrow Assign(TrackersID, HeadJoints)$;
  Update $KalmanFilterTracker$;
  foreach *tracker* $t_i \in TrackersID$ **do**
    $ROS \leftarrow Publish(t_i)$;
  end
end

To improve the proposed face tracking algorithm and minimize the tracking error, we obtain the head joint from the tracked skeleton provided by the WS1 Kinect V2 camera and try to assign it with the face center. If the assignment is successful, we update the face

tracker with the fine location. Otherwise, the tracker will be deleted. Further, this reduces the number of identity switches through longer periods of occlusions.

5. Experiments and Analysis

The most effective parts of the face recognition and tracking framework are the face detection and face recognition models. In order to well evaluate the effectiveness of the introduced tracking approach, we trained and evaluate the two models separately.

5.1. Face Detection

For the face detection, the RetinaFace is trained on the WIDER FACE dataset [52]. It contains 32,203 images and 393,703 face bounding boxes with a high degree of variability in scale, pose, expression, occlusion, and illumination. The evaluation is performed on the WIDER FACE validation set, with Average Precision (AP) of 0.83 for the hard subset.

5.2. Face Recognition

For the face recognition network, the ArcFace is trained on the MS1MV2 dataset [4,53] for 30 epochs with a batch size of 512, feature scale s of 64, and angular margin m of 0.5. MS1MV2 is a semi-automatic refined version of the MS-Celeb-1M dataset [53] which contains about 100k identities with 10 million images. The evaluation is performed on large-pose CPLFW and large-age CALFW datasets and achieved performance of 95.45% and 92.07% respectively.

5.3. Results

The metrics used to measure the overall system performance are precision, recall, F-score, and recognition rate. We classify the predictions into True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). A *True Positive* can be obtained in recognition when the model correctly predicted the subject class (i.e., subject ID), which means that it matches the ground truth. Otherwise, the prediction is considered a *False Positive*.

A *True Negative* can be obtained in recognition when the model is not supposed to predict a subject that is not in the database. Otherwise, the prediction is considered a *False Negative*.

Precision is the matching probability of the predicted subject identity relative to the ground truth identity, which shows the results of a correctly recognized subject. It can be calculated as follows:

$$Precision = \frac{TP}{TP + FP}. \tag{3}$$

Recall measures the probability of the subjects that were correctly recognized among ground truth subjects, which is the total number of true positives relative to the sum of true positives and false negatives, as follows:

$$Recall = \frac{TP}{TP + FN}. \tag{4}$$

F-score is evaluated as the harmonic mean of precision and recall to see which model best performs. It can be calculated as follows:

$$F\text{-}score = \frac{Precision * Recall}{Precision + Recall} * 2. \tag{5}$$

The recognition performance can be obtained by the face recognition rate FR_R, and it is the ratio between the total number of correctly recognized faces and the total detected/tracked faces. It can be calculated as follows:

$$FR_R = \frac{TP}{Total faces} * 100. \tag{6}$$

In order to evaluate the proposed framework, we tested it for two different evaluations: dataset, and online evaluations.

5.3.1. Dataset Evaluation

We use the ChokePoint dataset [54] to evaluate the proposed framework. This dataset is a video dataset that was collected and designed for experiments on person identification/verification under real-world surveillance conditions. It contains videos of 25 subjects (six female and 19 male). In total, the dataset consists of 48 video sequences and 64,204 face images with variations in terms of illumination conditions, pose, sharpness, as well as misalignment due to automatic face localization/detection.

The experimental results show the performance of tracking for 25 subjects of the ChokePoint dataset. To show recognition refinements, we have tested the proposed face recognition framework with tracker-assisted and without. The average results are shown in Table 1. Furthermore, the Receiver Operating Characteristic (ROC) curve is obtained in Figure 5, which shows that the tracking approach improves the recognition rate for high false positive rates and reduces the false classification rate.

Table 1. The average results of precision, recall, and F-score on ChokePoint dataset.

Tracking	Precision	Recall	F-Score
No	0.83	0.79	0.81
Yes	0.96	0.93	0.94

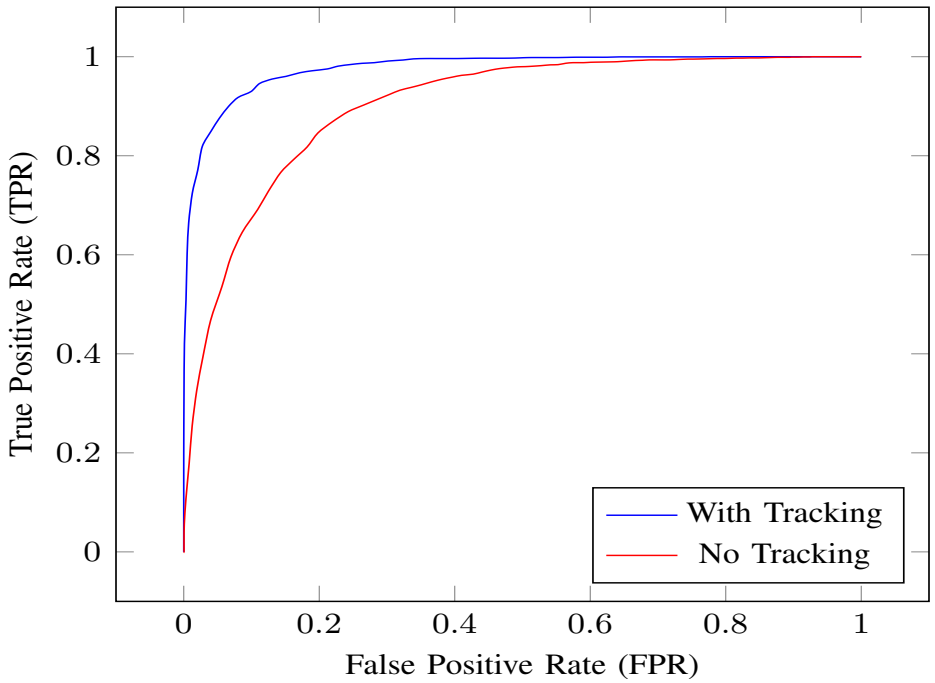

Figure 5. ROC Curve of ChokePoint Dataset for the Proposed Framework.

5.3.2. Online Evaluation

We employ the proposed framework on a real HRI study [6], to further evaluate the framework in real-time HRI and show its robustness. During the experiments in the study, the data for evaluation were collected from 11 subjects (two female and nine male) aged between 20 and 34 years.

The experimental results show the performance of tracking and recognition rate for 11 subjects during the interactions with RoSA [6]. To show recognition refinements, we have tested the proposed face recognition framework with tracker-assisted and without. The proposed framework achieved a face recognition rate of 94% and 76% with tracking

and without tracking, respectively. Figure 6 shows the impact of tracking on the *Precision* of the proposed framework, and the impact of tracking on *Recall* of the proposed framework is shown in Figure 7. Furthermore, Figure 8 shows the *F-score* results of the proposed framework with tracker-assisted and without tracking.

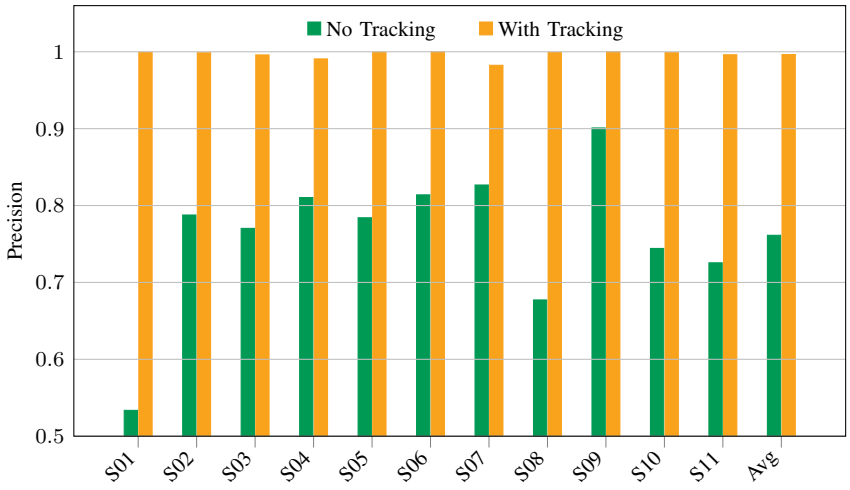

Figure 6. Impact of Tracking on Precision of Face Recognition.

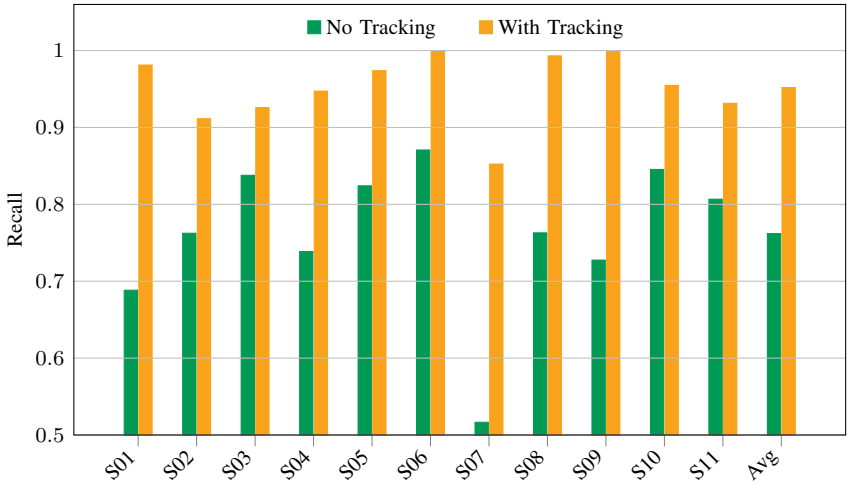

Figure 7. Impact of Tracking on Recall of Face Recognition.

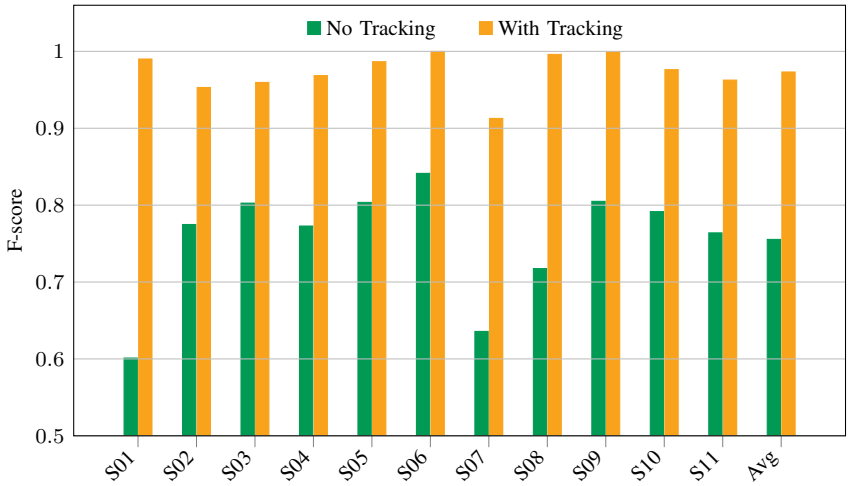

Figure 8. F-score results of the proposed framework with tracker-assisted and without tracking.

Compared to the standard face recognition framework, the proposed framework performance is faster in terms of processing time with frame rates of 25–40 fps. Some results of the proposed framework during the real HRI in our RoSA system [6] are shown in Figure 9.

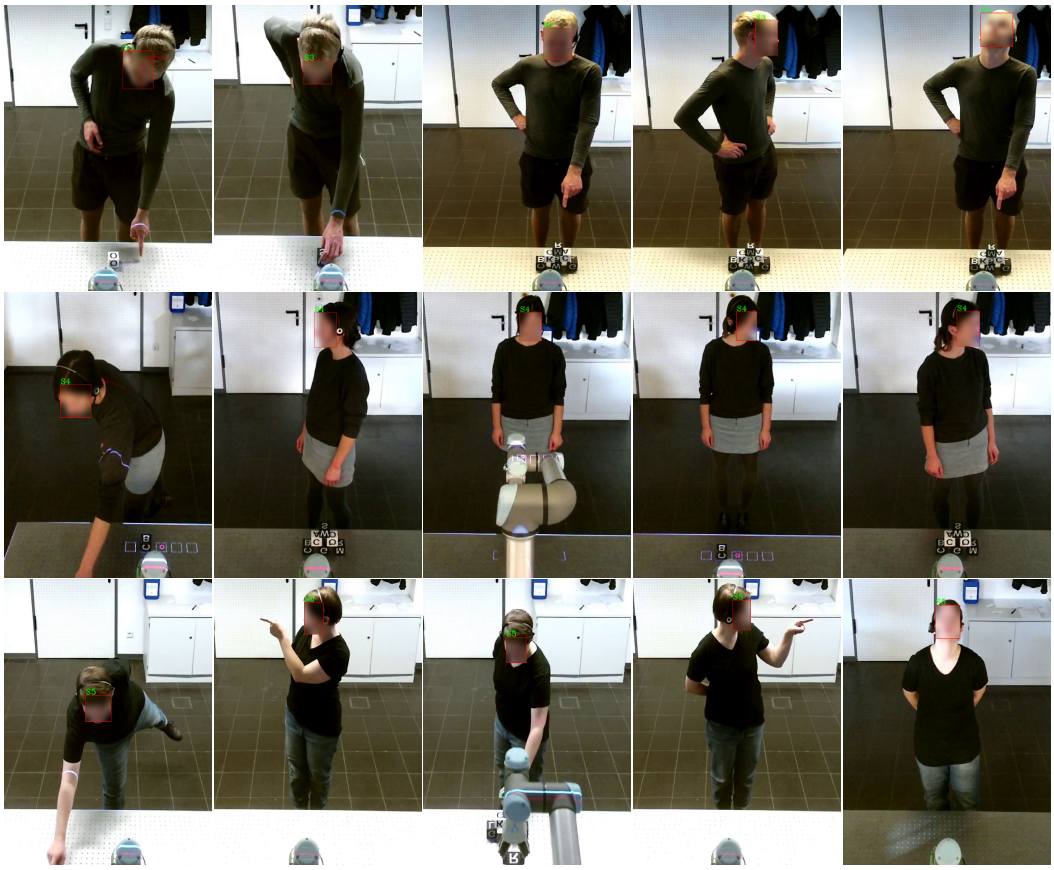

Figure 9. Experimental results of the proposed framework that shows the robustness of the framework against various head posture and illumination conditions.

To confirm the obtained results, we run the experiments again on the recorded videos from the Wizard-of-Oz study [5] with the same results. It contains videos of 36 subjects doing the same tasks on the RoSA study, which were collected on different days with different lighting conditions. For every subject (video), we selected three exemplar face images with different poses and added the extracted embedding to the database to match with video faces. Table 2 shows the precision and recall results for 37 subjects separated by the top ten results.

Table 2. Result of precision and recall for the proposed framework.

No	ID	Precision		Recall	
		Tracking	No Tracking	Tracking	No Tracking
1	4	0.97	0.76	0.81	0.70
2	7	1	0.84	0.92	0.59
3	11	1	0.68	1	0.73
4	16	1	1	1	0.66
5	18	0.96	0.88	1	0.81
6	24	0.98	0.65	0.95	0.77

Table 2. *Cont.*

No	ID	Precision		Recall	
		Tracking	No Tracking	Tracking	No Tracking
7	25	0.89	0.53	0.92	0.63
8	29	0.98	0.83	0.98	0.79
9	32	0.99	0.68	0.98	0.60
10	36	0.95	0.76	0.97	0.75

5.4. Computational Efficiency Assessment

In general, lightweight face networks provide promising results for face recognition. They are able to perform comparably to state-of-the-art very deep face models in most face recognition scenarios. In particular, ResNet100-ArcFace by Deng et al. [4] is one of the best performing state-of-the-art models in the different evaluated scenarios, however, it demands high computational resources. For example, the biggest difference in accuracy between ResNet100-ArcFace and MobileFaceNet (our used network), is 8% in the very large-scale DeepGlint-Image dataset (one of the most challenging databases), while in the remaining databases it is less than 3%. However, regarding the computational complexity, ResNet100-ArcFace requires 19X more storage space and involves 26X more FLOPs and 32X more parameters than MobileFaceNet.

Applying face tracking provides us the advantage of no need to apply face detection and recognition for all input frames. However, to increase the accuracy of our framework and minimize the tracking error, we apply the whole recognition process in each fifth frame.

To calculate the computational efficiency assessment of the proposed framework, we tested it on the collected videos (total of 47 videos) during the RoSA study [6] and the Wizard-of-Oz study [5] and obtained the average processing time for each face recognition module. The hardware setting used was a NVIDIA GeForce GTX 1080 Ti Desktop GPU (12 Gb GDDR5, 3584 CUDA cores). Table 3 shows the average execution time of individual methods used in the proposed framework. To summarize, the average execution time per frame for the whole process takes about 6.7 ms, and the average number of frames per second is ∼35 frames.

Table 3. Average execution time of individual methods used in the proposed framework.

Method	Average Time (ms)
Detection	3.2
Alignment	1.4
Tracking	0.8
Recognition (Embedding Inference)	1.3
Identification (Similarity)	0.08
Visualization & Delays	7.5

6. Limitations, and Future Work

The study conducted has a complex setting that contains two workstations (WS1 and WS2) synchronized together using the ROS operating system. In addition, extracting face features during the experiments is a challenging task due to the illumination conditions, extreme deviation in head pose angles, and occlusion. However, the aforementioned performance evaluation showed the effectiveness of the proposed framework in recognizing the subject's identity in a multi-person environment.

Few subjects caused a wrong identification during the experiments due to the lack of the registration process and good face feature embedding, which lead to the re-registration of the mentioned subjects.

The advantage of our framework is that it depends on lightweight CNNs for all face recognition stages, including face detection, alignment and feature extraction, to meet

the real-time requirements in HRI systems. Furthermore, the developed framework can simultaneously recognize the faces of the cooperating subjects in various poses, face expressions, illumination, and other outdoor-related factors. Although two of the subjects were wearing face masks for the whole experiment, our model succeeded to recognize their identity with reasonable confidence.

Future work would involve a new study with a large number of subjects with different human–robot interaction scenarios to effectively assess the performance of the framework and overcome the limited number of subjects in the RoSA study. In addition future work would involve designing an end-to-end trainable convolutional network framework for all the face recognition stages.

7. Conclusions

We propose a face recognition system for human–robot interaction (HRI) boosted by face tracking based on deep convolutional neural networks (CNNs). To ensure that our framework can work in real-time HRI systems, we developed our framework based on lightweight CNNs for all face recognition stages, including face detection, alignment, tracking, and feature extraction. Furthermore, we implemented our approach as a modular ROS package that makes it straightforward for integration in different HRI systems. Our results suggest that the use of face tracking alongside face recognition increases the recognition rate.

We utilize the state-of-the-art loss function *ArcFace* for the face recognition task and the *RetinaFace* method for face detection combined with a simple online and real-time face tracker. Furthermore, we propose a face tracker to tackle the challenges faced by the existing face recognition methods including various illumination conditions, continuous changes in head posture, and occlusion.

The face tracker is designed to fuse the tracking information with the recognized identity and associate it with the faces once they are detected for the first time. For the updated trackers, the last recognized identity will be kept alongside the tracker. Despite what preceded, a new identity prediction is required for the new trackers. This method improved the overall processing time and face recognition accuracy and precision for the unconstrained face.

The proposed framework is tested in real-time experiments applied in our real HRI system "RoSA" with 11 participants interacting with the robot to accomplish different tasks. Furthermore, to confirm the obtained results, we tested it on the recorded videos from the Wizard-of-Oz study, which contains videos of 36 subjects doing the same tasks on "RoSA" with the same results. The results showed that the framework can improve the robustness of face recognition effectively and boost the overall accuracy by an average of 25% in real-time. It achieves an average of 99%, 95%, and 97% precision, recall, and F-score respectively.

Author Contributions: Conceptualization, A.K. and A.A.-H.; methodology, A.K., A.A.A., D.S., J.H. and T.H.; software, A.K., D.S., J.H. and T.H.; validation, A.K., D.S., J.H. and A.A.A.; investigation, A.K., D.S., J.H. and A.A.A.; resources, A.A.-H.; writing—original draft preparation, A.K., A.A.A., D.S., J.H. and T.H.; writing—review and editing, A.K., A.A.A., D.S., J.H., T.H. and A.A.-H.; visualization, A.K., A.A.A., D.S., J.H. and T.H.; supervision, A.A.-H.; project administration, A.K. and A.A.-H.; funding acquisition, A.A.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the Federal Ministry of Education and Research of Germany (BMBF) RoboAssist No. 03ZZ0448L and Robo-Lab No. 03ZZ04X02B within the Zwanzig20 Alliance 3Dsensation.

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki. Ethical approval was done by Ethik Kommision der Otto-von-Guericke Universtiät (IRB00006099, Office for Human Research) 157/20 on 23 October 2020.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Zacharaki, A.; Kostavelis, I.; Gasteratos, A.; Dokas, I. Safety bounds in human robot interaction: A survey. *Saf. Sci.* **2020**, *127*, 104667. [CrossRef]
2. Mukherjee, D.; Gupta, K.; Chang, L.H.; Najjaran, H. A survey of robot learning strategies for human–robot collaboration in industrial settings. *Robot. Comput. Integr. Manuf.* **2022**, *73*, 102231. [CrossRef]
3. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
4. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4690–4699.
5. Strazdas, D.; Hintz, J.; Felßberg, A.M.; Al-Hamadi, A. Robots and Wizards: An Investigation Into Natural Human–Robot Interaction. *IEEE Access* **2020**, *8*, 207635–207642. [CrossRef]
6. Strazdas, D.; Hintz, J.; Khalifa, A.; Abdelrahman, A.A.; Hempel, T.; Al-Hamadi, A. Robot System Assistant (RoSA): Towards Intuitive Multi-Modal and Multi-Device human–robot Interaction. *Sensors* **2022**, *22*, 923. [CrossRef] [PubMed]
7. Favelle, S.; Palmisano, S. View specific generalisation effects in face recognition: Front and yaw comparison views are better than pitch. *PLoS ONE* **2018**, *13*, e0209927. [CrossRef] [PubMed]
8. Albiero, V.; Chen, X.; Yin, X.; Pang, G.; Hassner, T. img2pose: Face alignment and detection via 6dof, face pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7617–7627.
9. Minaee, S.; Luo, P.; Lin, Z.; Bowyer, K. Going deeper into face detection: A survey. *arXiv* **2021**, arXiv:2103.14983.
10. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
13. Najibi, M.; Samangouei, P.; Chellappa, R.; Davis, L.S. Ssh: Single stage headless face detector. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4875–4884.
14. Fiedler, M.A.; Werner, P.; Khalifa, A.; Al-Hamadi, A. SFPD: Simultaneous Face and Person Detection in Real-Time for human–robot Interaction. *Sensors* **2021**, *21*, 5918. [CrossRef]
15. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A survey of deep learning-based object detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
16. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
17. Zhang, C.; Xu, X.; Tu, D. Face detection using improved faster rcnn. *arXiv* **2018**, arXiv:1802.02142.
18. Najibi, M.; Singh, B.; Davis, L.S. Fa-rpn: Floating region proposals for face detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–17 June 2019; pp. 7723–7732.
19. Zhang, H.; Chang, H.; Ma, B.; Shan, S.; Chen, X. Cascade retinanet: Maintaining consistency for single-stage object detection. *arXiv* **2019**, arXiv:1907.06881.
20. Huang, G.B.; Mattar, M.; Berg, T.; Learned-Miller, E. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Proceedings of the Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Marseille, France, 17 October 2008.
21. Wu, Y.; Ji, Q. Facial landmark detection: A literature survey. *Int. J. Comput. Vis.* **2019**, *127*, 115–142. [CrossRef]
22. Gogić, I.; Ahlberg, J.; Pandžić, I.S. Regression-based methods for face alignment: A survey. *Signal Process.* **2021**, *178*, 107755. [CrossRef]
23. Trigeorgis, G.; Snape, P.; Nicolaou, M.A.; Antonakos, E.; Zafeiriou, S. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4177–4187.
24. Zhu, H.; Sheng, B.; Shao, Z.; Hao, Y.; Hou, X.; Ma, L. Better initialization for regression-based face alignment. *Comput. Graph.* **2018**, *70*, 261–269. [CrossRef]
25. Valle, R.; Buenaposada, J.M.; Valdes, A.; Baumela, L. A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 585–601.

26. Feng, Z.H.; Huber, P.; Kittler, J.; Christmas, W.; Wu, X.J. Random cascaded-regression copse for robust facial landmark detection. *IEEE Signal Process. Lett.* **2014**, *22*, 76–80. [CrossRef]
27. Zhu, S.; Li, C.; Loy, C.C.; Tang, X. Face alignment by coarse-to-fine shape searching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4998–5006.
28. Kumar, A.; Chellappa, R. Disentangling 3d pose in a dendritic cnn for unconstrained 2d face alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 430–439.
29. Guo, X.; Li, S.; Yu, J.; Zhang, J.; Ma, J.; Ma, L.; Liu, W.; Ling, H. PFLD: A practical facial landmark detector. *arXiv* **2019**, arXiv:1902.10859.
30. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
31. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
32. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
33. Zhong, Y.; Deng, W.; Hu, J.; Zhao, D.; Li, X.; Wen, D. SFace: Sigmoid-constrained Hypersphere Loss for Robust Face Recognition. *IEEE Trans. Image Process.* **2021**, *30*, 2587–2598. [CrossRef]
34. Li, L.; Mu, X.; Li, S.; Peng, H. A Review of Face Recognition Technology. *IEEE Access* **2020**, *8*, 139110–139120. [CrossRef]
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
36. Sun, Y.; Wang, X.; Tang, X. Deeply learned face representations are sparse, selective, and robust. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2892–2900.
37. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
38. He, R.; Wu, X.; Sun, Z.; Tan, T. Wasserstein cnn: Learning invariant features for nir-vis face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1761–1773. [CrossRef] [PubMed]
39. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 212–220.
40. Deng, J.; Zhou, Y.; Zafeiriou, S. Marginal loss for deep face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 June 2017; pp. 60–68.
41. Khalifa, A.; Al-Hamadi, A. A Survey on Loss Functions for Deep Face Recognition Network. In Proceedings of the 2021 IEEE 2nd International Conference on human–machine Systems (ICHMS), Magdeburg, Germany, 8–10 September 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
42. Fuad, M.T.H.; Fime, A.A.; Sikder, D.; Iftee, M.A.R.; Rabbi, J.; Al-Rakhami, M.S.; Gumaei, A.; Sen, O.; Fuad, M.; Islam, M.N. Recent Advances in Deep Learning Techniques for Face Recognition. *IEEE Access* **2021**, *9*, 99112–99142. [CrossRef]
43. Hsu, G.S.J.; Wu, H.Y.; Yap, M.H. A comprehensive study on loss functions for cross-factor face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 826–827.
44. Hu, W.C.; Chen, C.H.; Chen, T.Y.; Huang, D.Y.; Wu, Z.C. Moving object detection and tracking from video captured by moving camera. *J. Vis. Commun. Image Represent.* **2015**, *30*, 164–180. [CrossRef]
45. Liu, F.; Gong, C.; Huang, X.; Zhou, T.; Yang, J.; Tao, D. Robust visual tracking revisited: From correlation filter to template matching. *IEEE Trans. Image Process.* **2018**, *27*, 2777–2790. [CrossRef] [PubMed]
46. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 3464–3468.
47. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 3645–3649.
48. Lian, Z.; Shao, S.; Huang, C. A real time face tracking system based on multiple information fusion. *Multimed. Tools Appl.* **2020**, *79*, 16751–16769. [CrossRef]
49. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
50. Chen, S.; Liu, Y.; Gao, X.; Han, Z. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In Proceedings of the Chinese Conference on Biometric Recognition, Urumchi, China, 11–12 August 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 428–438.
51. Nguyen, H.V.; Bai, L. Cosine similarity metric learning for face verification. In Proceedings of the Asian Conference on Computer Vision, Queenstown, New Zealand, 8–12 November 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 709–720.

52. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. Wider face: A face detection benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5525–5533.
53. Guo, Y.; Zhang, L.; Hu, Y.; He, X.; Gao, J. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
54. Wong, Y.; Chen, S.; Mau, S.; Sanderson, C.; Lovell, B.C. Patch-based Probabilistic Image Quality Assessment for Face Selection and Improved Video-based Face Recognition. In Proceedings of the IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 81–88.