UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DEPARTMENT OF INFORMATION ENGINEERING

MASTER DEGREE IN COMPUTER ENGINEERING

"A real-time service for face re-identification based on deep learning and online clustering"

Supervisor: Prof. Emanuele Menegatti

Candidate: Leonardo Codato

Co-Supervisor: Dott. Andrea Molon

ACCADEMIC YEAR 2021 / 2022

Graduation date 13/12/2022

Index

Abstract

A face re-identification service aims to verify if detected faces have already been seen from the system. The ability of identify new and returning persons and distinguish them can be useful in social robots in order to adapt their behavior according to who they have in front. The reidentification problem is particular challenging since we start with no previous information about the persons to recognize and the database upon which we do recognition must be built in real-time with no control over the conditions and quality of the gathered faces. The proposed system is based on deep neural network to be able to extract the features from a detected face and a clustering system to correctly group and store the faces of the same person together. In particular, since the recognition is performed on such clusters, is crucial to keep them of the higher quality possible. To do so every face is tied with a quality score, which will play a vital role in choosing which faces keep in the sets, which not, and in general in the cluster update and creation operations. The system has achieved an accuracy of 0.78 on faces detected by a mobile robot in an unconstrained environment. With a False Acceptance Rate of 0.15 and a False Rejection Rate of 0.07 mainly due to misclassification of sideview of the faces.

Abstract (Italian)

Un servizio di re-identificazione facciale, una volta rilevata una faccia, si occupa di determinare se è già stata vista dal sistema e se è nuova. Poter distinguere persone nuove da quelle già viste, può essere utile nei robot sociali per poter adattare il loro comportamento a seconda di chi sta interagendo con loro. La re-identificazione è un compito particolarmente impegnativo poiché si parte senza avere alcun tipo di informazione su chi bisognerà identificare. Il database sui cui si basa il riconoscimento deve essere aggiornato in tempo reale senza alcun controllo sulle condizioni in cui sono state raccolte le facce e la loro qualità. Il sistema proposto si basa su una rete neurale per poter estrarre le caratteristiche di una faccia rilevata e su un sistema di clustering per raggruppare correttamente i volti appartenenti alla stessa persona. Poiché il riconoscimento si basa sui cluster creati dal sistema, è particolarmente importante aggiornarli con facce di buona qualità. Per fare ciò ogni volto è associato con quality score che risulta cruciale per decide che facce mantenere, quali eliminare e più in generale nelle operazioni di creazione ed aggiornamento dei cluster. Il sistema proposto ha raggiunto una precisione di 0.78 su facce rilevate da un robot mobile in un ambiente non controllato. Un False Acceptance Rate di 0.15 e un False Rejection Rate of 0.07, principalmente a causa di errori di classificazione dovuti a viste laterali dei volti.

1. Introduction

In computer vision one of the most studied and active research field has always been face recognition. The first studies can be traced back to the mid-1960s, with the first example of facial recognition technology presented in 1991 by Pentland et. al [1]. Such systems have seen prosperous growth in recent years. This can be traced back to two main reasons: the recent technologies such as machine learning and deep learning that have facilitated their development and to the numerous commercial uses that can be derived from them. Think about, for example, the most common smartphone unlocking mechanism to the rarer social robots that are expected to become increasingly popular in the future [2].

Use of the face to recognize a person is considered extremely attractive compared to other biometric features such as the fingerprint, because it is easier and more intuitive for the end user to use. Facial images can be easily acquired simply standing in front of a camera, there is no need to touch any sensor and require less assistance from the end user compared to other systems. However, this ease of use come with some drawbacks. Identifying facial images captured in unconstrained environments that involves changes in lighting, posture, facial expressions, partial occlusion, camera movement etc... still poses several challenges.

A face recognition task can be classified in two main categories: face verification, which involves comparison of the features of two faces, one of which can be stored in a DB, stating if they belong to the same person and face identification, that refers to the process of finding the identity of a face image given a database of known faces [3]. However, a third task can be identified: face re-identification. Face re-identification require a system to state if a face has already been seen, and if so, determine its identity. The main characteristic of such task is that the system starts with no prior knowledge about the faces to recognize, and it must add new people to its database while these are detected.

Consider a mobile social robot in an office. The ability to distinguish between those who live the office on a daily basis from those who enter it occasionally can lead to a use case where the robot, previously informed about an incoming meeting and the invited persons, can recognize and guide the guests to the room where the meeting will be held. Another use case could be in a hospital or nursing home where people must be

guided to the correct room to visit. The robot or set of robots could store the information about the last places visited by a guest and lead them to the correct room. Given that, the only information accessible to the robot, are the room's number and the frequency of the visitor visits without retaining any information about their and room's guest identity. These can be some use case where a re-identification system can be extremely useful, as it does not require populating the database that can be created and updated automatically by the robot while new people are encountered.
A face re-identification system must address challenges that are common to face recognition, like:

- Pose variation, which causes detection problem other than identifying problem.
- Illumination changes, which can significantly change the appearance of a face
- Occlusion, hidden parts of the face can make recognition difficult
- Blur, noise, and resolution difference can complicate the task

But there are also challenges specific of the re-identification task, like:

- The creation and maintenance of a high-quality faces database for the recognition without prior knowledge about the persons.

Moreover, considering the use for social robot and in particular in human robot interaction we need a system that:

- Must achieve high processing speed in order to maintain real time interaction with the final user
- Should achieve a discrete accuracy for images taken while the robot is moving
- Must use monocular RGB camera

The presented project is a real-time unsupervised service for face re-identification that can works on mobile robot in an unconstrained environment. This system allows the robot to autonomously create and populate a database with the different persons faces detected, grouping them, and assigning a unique alphanumeric ID to distinguish them. The faces

set associated to each ID can be updated over time with the new detected faces whether they represent an improvement over those already present.
We can summarize the entire process as follows. Firstly, every image passed to the service from the main image HUB goes through some basic preprocessing operations. The face detection step is performed using a pre-trained CNN as well as the face encoding step which produces a 128-dim encoding vector for each face. To correctly group the faces of the same person is used a simple online clustering algorithm that creates and maintains the clusters. It is used to add the faces and choose which ones remove when the maximum allowed number is reached. Since these clusters are used to perform the recognition, their quality is of paramount importance. To ensure that the faces stored in the clusters are of high quality is used a face quality assessment neural network. Using this information, we can choose which faces store, which not and which to remove. In addition, there is also the possibility to add some known person to the database passing a set of images of their faces, such cluster will not be updated during execution.

The thesis is organized in the following way. In Chapter II is presented in detail the re-identification problem. Chapter III analyzes the proposed solution deepening, in each section, respectively: the preprocessing techniques, the face analysis operations, the cluster management, the exceeding face and the expired person removal and lastly the operations for the known person management. Chapter IV presents how the service has been implemented and how communicates with the robot. Chapter V analyzes the performances of the proposed solution. In Chapter VI are presented some methods to improve the service, and in chapter VII there is the conclusion.

2. Re-Identification problem definition

The face re-identification problem is a reduction of the person re-identification problem defined as a process of establishing correspondence between images of a person taken from different cameras. It is used to determine whether instances captured by different cameras belong to the same person, in other words, assign a stable ID to different instances of the person. Such problem has been widely studied over the years with a number of solutions and implementations proposed [4] [5] [6]. Person re-identification systems found their main use for surveillance purpose where multiple cameras are used, and they utilize the entirety of the person's body to perform recognition.
Such solutions in a scenario involving a social robot are not suitable since in many cases the person will be close to the robot, to interact with it, and a full view of the body will not be available. So, to re-identify a person we must rely on the face. However, the face is not a so powerful biometric feature compared to others such as fingerprints or retina recognition. A face can potentially be influenced by cosmetics, disguises, and lighting but it is used because it is by far the easiest to obtain without intrusive interaction. In contrast to fingerprint or iris images, facial images can quickly be obtained without physical contact and with less assistance from the end user. Also, the modality required to capture a face image is well known by all people since every one of use has been in front of a camera before. The face re-identification task is mainly concerned with accomplishing face recognition for people who return after a while in front of the camera, while the person re-identification aims to consistently identify a person starting from instances captured from different cameras. The case for face recognition with multiple cameras can be addressed for example in a scenario where a fleet of robots refers to the same persons database and multiple robots have in their field of view the same face. In such case we have to consistently associate the same ID across all the views.

Face re-identification is also a task of the broader face recognition problem that requires, given a still image or a video, to identify or verify the face of one or more persons in the scene using a stored database of faces. In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face in a one-to-one match up with a stored face. In such cases the database on which the recognition is performed is prebuilt and closed, with a predefined number of identity and the system can only say if the face belongs to such identities or not. However, a third task can be identified: face re-identification. Face re-identification requires a system to determine if a face has already been seen, and if so, determine its identity. The main characteristic of such task is that the system starts with no prior knowledge about the faces, and it must add new people to its database while these are detected.

The face re-identification problem has been addressed over the years with a number of solutions proposed. In [7] Aryananda implement an online and unsupervised face recognition system, where the robot opportunistically collects, labels, and learns various faces while interacting with people (without any staged introduction session), starting from an empty database. Their use eigenfaces for recognition, which require to take faces picture in a high-constrained environment with same light conditions, frontal and aligned exclude the use case in real scenario. In [8] Farinella et al. Propose a reidentification system employing a frontal face detector and Locally Ternary Pattern for the re-identification purpose. However, cannot be used on mobile robot since require the person to stand in front of the camera for the recognition time, also works only on frontal faces. In [9] Shen Khang Teoh et al develop a re identification system based on a single shot face detector and a CNN model based on Mobilenet V2 to generate 256 facial embedding vectors. The recognition is performed using cosine distance. The system obtain a 99.01% per face recognition accuracy with an average processing time of 60ms. However, the faces used to populate their DB for recognition were ideal, with clear frontal

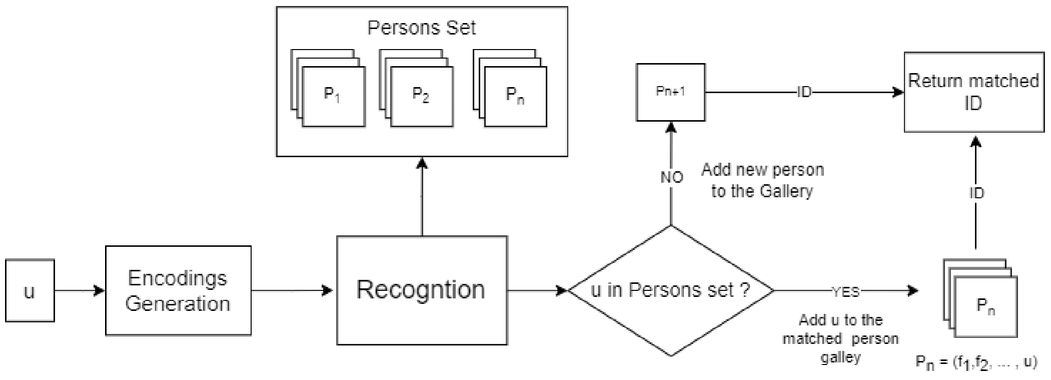


Figure 1 - Representation of a Re-Identification system

faces. Also claimed to work only on frontal faces. Dantcheva et al in [10] specifically addresses the full frontal-to-side facial recognition using, other than facial features, other soft biometric traits, specifically color–, texture– and intensity– based traits taken from patches of hair, skin, and clothes. The working scenario consider faces with near to 90 degrees pose difference, which, in an unconstrained scenario, is not always the case.

A face re-identification (Fig. 1) system is similar to a face recognition system which is composed of a gallery set, a faces' database of known people associated with their unique identifier, and a probe, an unknown face on which the recognition has to be performed.

Let the database be represented as $D = (P_1, P_{2,}P_3, \dots, P_n)$ composed by n different persons. Each person is associated with a set of m faces belonging to the same person $P_i = (f_{i,1}, f_{i,2}, \dots, f_{i,m})$ where m can differ for each person. The set of known ID's is given by $id(D) = (id(P_1), id(P_2), id(P_3), \dots, id(P_n))$ where the $id(\cdot)$ function specifies the unique ID assigned to its argument. Let U= $(u_1, u_{2,}u_3, \dots, u_n)$ be the unknown probe set, with the set of unknown IDs given by $id(U) = (id(u_1), id(u_2), id(u_3), \dots, id(u_n))$. In a recognition system when the probe is presented to the system, it is compared to each person and some similarity measures are computed. The gallery is ranked using the similarity in order to determine the probe ID. The same setup applies to the problem of Re-ID. We can distinguish the re-identification problem in two categories: the closed set and open set re-identification.

In the *closed set re-identification* scenario, the probe is a subset of the gallery with the gallery size fixed. It means that all the persons upon which the recognition can be performed are available at start time and will not be changed overtime. In the end the probe identity exists in the gallery and the objective is to determine the true ID of the probe. Thus, given that $id(U) \subseteq id(D)$, the true probe ID for a given probe u_j is $id(u_j) = id(P_{i^*})$, such that,

$$i^* = \underset{i \in 1,\dots,n}{\operatorname{argmax}} \; p\,(P_i | \, u_j) \qquad [1]$$

where $p(P_i|u_j)$ is the likelihood that $id(u_j) = id(P_i)$ and is most often represented by a similarity measure. This implies that the top ranked gallery ID is assigned to the probe.

In the *open set re-identification*, the probe may or may not be a subset of the gallery and the gallery can evolve over time. The objective is to first establish if the probe ID is a part of the gallery, and if so, determine the true probe ID. Thus, in order to find the true ID, in addition to ranking the gallery elements and determining i^*, we have to also satisfy the following condition:

$$p(P_{i^*}|u_j) > \varepsilon \qquad [2]$$

here ε is an acceptable level of certainty beyond which we can state that $id(u_j) \subseteq id(D)$. If this condition is not satisfied, we can say that the probe is not part of the gallery, and the probe ID must be enrolled into the database. The process of determining a previously unknown ID is called *novelty detection*. With this the recognition database evolves over time increasing the number of people for the recognition.

In our scenario the users' ID are not available in advance, which means that there is no gallery set available a-priori and it must be created by the system itself. Therefore, we are dealing with an open set re-identification problem where the database dynamically changes over time. Moreover, we can classify our scenario as a multiple Re-ID problem since we allow multiple persons to be on camera for the recognition at the same time.

We have said that the face re-identification problem can be easily reconducted to the face recognition problem, and therefore we have to deal with many of the inherent challenges of such task. The main issue is how facial appearance can change from shot to shot making the recognition difficult. Factors leading to such variation can be categorized into intrinsic and extrinsic [11].

Intrinsic factors are the ones that depend on the physical structure of the face.

- Expression: Humans are very efficient in communication. There is a whole palette of expressions to communicate even without a world. Some of the prominent expressions are neutral, laugh, smile, disgust, anger, fear, surprise etc... Expression could be characterized by facial actions including closing eyes and/or mouth, modifying the geometry and/or texture etc.…

- Age: Being made of a biological tissue, the face is also bound to change with time. The texture and appearance of a human face varies with aging which may create an issue during the face recognition.
- Occlusion: User could cover some part of his face due to weather conditions or fashion. Appearance or removal of hair/beard, wearing glasses/accessories, application of makeup etc. may lead to occlusion of the face.
- Doctored images and image falsification: Spoofed facial image could be used to attack an automatic face recognition system. A common practice is to present an artificially constructed facial image that is a mixture of two faces to the system. A secure system should be provisioned to detect spoofs and protect its templates.

Extrinsic factors are the external factors that affect the image capturing process and thus, alter the appearance of the face.

- Illumination: Lighting has a magical effect on the appearance of any object. There is a very high degree of freedom for lighting variation to appear. These variations in many of the cases result in a huge change that is even more than the identity change.
- Scale/Resolution: Image can be captured from different distances. This not only affects the resolution but also the attention in the image. These variations are broadly called scale variation.
- Pose: It is characterized by rotation of the head with respect to the image capturing plane. One can observe that it could not only hide some part of the face but also could be present all together with lot different perspective to the appearance.
- Noise: It is one of the fundamental properties of the acquisition device that cannot be avoided. Therefore, some noise compensation methods could be deployed in face recognition.
- Blur: It is quite common in an image and can appear due to various reasons including unstable camera, motion of object, high camera exposure time etc. in our case this can be due to the robot movement.

All of these problems are unavoidable when face images are acquired in an uncontrolled, uncooperative environment and can lead to misclassification, assigning to a face the wrong ID or to the creation of new ID even if the person is already in the database. These two situations lead to a massive problem for an open set Re-ID system.
The database is not given, and we have no prior information about the persons to recognize. It can be updated over time adding fresh faces to a set's person if such faces satisfy some criteria or adding new persons to the gallery.
Consider the first scenario where we have a recognition and want to update the corresponding set. So given a person P_1 in the database, a probe u_1 such that $id(P_1) = id(u_1)$ and our recognition algorithm that assigns the IDs, $\widetilde{id}$. The objective is to develop a service such that the $\widetilde{id}(u_i) = id(u_i)$, where $id(u_i)$ is the true identity of u_i. However, since the service is working in an unconstrained scenario where multiple factors can affect the recognition we can fall in the situation where $\widetilde{id}(u_1) \neq id(u_1)$, the assigned identities differ from the real one. Namely $\widetilde{id}(u_1) = id(P_2)$, the face is wrongly associated to the stored person P_2. In such case the detected face could be wrongly added to the faces set of the person P_2 such that $P_2 = (f_{2,1}, f_{2,2}, \dots, u_{1,m})$ where all the faces $f_{2,j}$ belong to P_2 except for $u_{1,m}$. This can corrupt the set of the person leading to other possible misclassification in the future.
The other case is when a person already in the database is not recognized and a new ID (person) will be created. Which means that, given a person P_1 in the database and a probe u_1, such that $id(P_1) = id(u_1)$, the condition stated by equation [2] is, wrongly, not satisfied. With this, since the face is not recognized, a new person P_k will be created such that $P_k = (u_{1,1})$. When another probe u_g such that $id(P_1) = id(u_g)$ the service given that also the only face $u_{1,1}$ of P_k is such that $id(u_{1,1}) = id(P_1)$ can return one of the two results $\widetilde{id}(u_g) = id(P_k)$ or $\widetilde{id}(u_g) = id(P_1)$ since both of them are technically true. This means that for the system the same person will be associated with two different identities. The database creation and maintenance are critical tasks for such re-identification system. We have to make sure that when a new person is created (new ID) the face detected is really new and not a misclassification. Moreover, we have to make sure that the faces used to update a person's set belong to such set and their quality is high enough to improve successive recognition and not make it worse.

3. Proposed solution

The solution must deal with an open set face re-identification problem. The general pipeline requires to 1) preprocess the given image 2) detect all the faces that are present if any 3) align the faces 4) evaluate their quality 5) compute the encodings 6) perform the re-identification. This last step can result in an effective re-identification that can lead to an update of the faces set of the person or to a novelty detection that leads to the creation of a new person in the database.

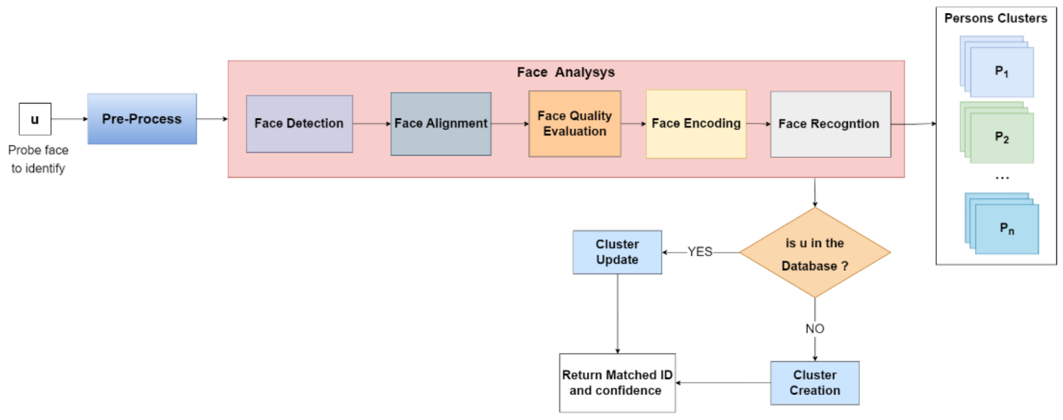


Figure 2 - Representation of the re-identification pipeline of a probe image given in input

We have also to deal, by design, with two more constraints which are:

1. The number of faces for each set is limited. We have to choose which one keeps and which one removes. Every new detected face is eligible to be part of the set if it respects some quality conditions and can potentially improve the quality of the cluster.
2. The person can stay saved in the database for a limited time. After a user-defined period of time the person must be removed from the database. If their return after the elimination, we fall in the case of a novelty detection and will be saved again with a new ID

Moreover, we have implemented a number of functions to manage a set of static clusters. Such sets are created and updated manually by the end user since they store faces of known person, and they are handled differently than sets created autonomously by the robot.

All those steps are presented in the following sections describing how they have been implemented.

3.1 Preprocess

We have said that the service will work with images taken in an unconstrained environment. Those implies a set of situations we have to deal with:

- We can have strong illumination changes from face to face that have to be recognized.
- Since the robot is mobile and also the person can move around, we have to deal with blurred images.
- We have to consider the case where the robot's camera is covered because is very close to a wall or a person stand in front of it.
- Noise is always a factor to be considered.

To deal with all of this before proceeding with the proper Re-ID we apply to the image some preprocessing.

Usability detector

To check if the image is not uniform, which means that the camera is totally occluded, and perform the detection only if it is useful, since a face could be present, we implemented a usability detector. It uses a canny edge detector to obtain a binary image that highlights the edges. The process of Canny edge detection algorithm can be broken down to five different steps:

1. Apply Gaussian filter
2. Find the intensity gradients of the image
3. Apply gradient magnitude thresholding or lower bound cut-off suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

To state if it is uniform or not, we pick the edge to pixels image ratio of the canny image:

$$edgeRatio = \sum_{i,j}^{w.h} canny[i][j] \Big/ w * h \qquad [3]$$

Where $\sum_{i,j}^{w.h} canny[i][j]$ is the summation of the pixels value of the canny image and $w * h$ is the number of all the pixels. The *edgeRatio* gives us the value of how many edges

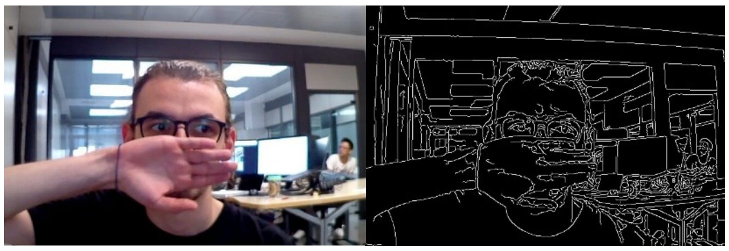


Figure 3 - Example of image that pass the usability test. On the right the canny image

are in the image. If it is under a threshold, we consider the image as uniform, and it does not proceed further in the evaluation. This is done since the steps to find and recognize a face are expensive computational operations and we did not want to spend resources on images that carry no information.

Histogram equalization

To deal with bad illumination and contrast situations, where a face can be over/underexposed or with a general bad illumination that can make the recognition harder, we apply a basic histogram equalization in order or improve the condition for the successive steps. The CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm has been applied. CLAHE divides the image into small blocks called "tiles" and each of these blocks are histogram equalized. To avoid noise amplification, it uses contrast limiting by clipping the histogram bins at a predefined value. Clipped pixels are distributed uniformly to other bins before applying histogram equalization. After equalization, to remove artifacts in tile borders, bilinear interpolation is applied.

Denoising

To deal with noise we applied a Non-Local Means Denoising algorithm [12]. Classic denoising methods want to replace the color of a pixel with an average of the colors of nearby pixels. The variance law in probability theory ensures that if nine pixels are averaged, the noise standard deviation of the average is divided by three. But the most similar pixels to a given pixel have no reason to be close at all. NLM aims to find in the image a set of pixels that resemble the one we want to denoise. Denoising is then done by computing the average color of these most resembling pixels. The resemblance is evaluated by comparing a whole window around each pixel, and not just the color.

3.2 Face analysis

In this section are presented all the steps that are performed over a face to arrive at the re-identification. These include the classic key steps prior to the recognition which includes face detection, which require to find the faces present in the image. Face encoding, which represents the detected face. Such encoding can be obtained starting from a set of detected features of the face. Finally, the recognition step, which requires to match the probe face to the ones of the persons in the database, if any, or the creation of a new one. Beside those steps, two other operations are carried out. Face alignment, which takes care of aligns the detected face so to minimize the differences among the detections, and face quality evaluation, which evaluates the quality of the face with a score. Such score will be used in different occasions like compute the match confidence and choose which faces eliminate from a set.

3.2.1 Face detection

Face detection is a specific case of object-class detection, where the task is to find the locations and sizes of all objects in an image that belong to a given class. Face detection algorithms focus on the detection of human faces answering two questions: are there any faces in the image? Where are they located?
By design we require that the face detector must:

- Detect multiple faces in an image since we want to cover the case where there are multiple people in front of the robot, and we want to be able to recognize them all.
- Detect faces that are far away from the camera to the extent possible. Want to recognize people that are not directly interacting with the robot, and so

are close to it, but also people that are approaching or passing by and that can be farther.

- Detects partially occluded faces. We have no control over the environment where the images are captured, and it may happen that the faces are partially covered.
- Deal well with harsh illumination conditions. Always because of the unconstrained environment the illumination might not be optimal.

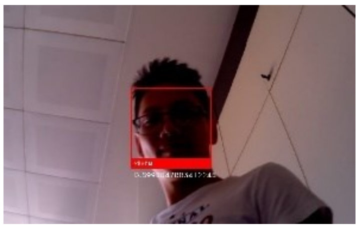


Figure 4 - Detected face in bad lighting conditions

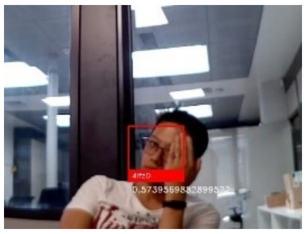


Figure 5 - Detected partially occluded face

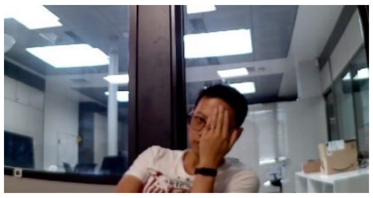


Figure 6 - Occluded face not detected

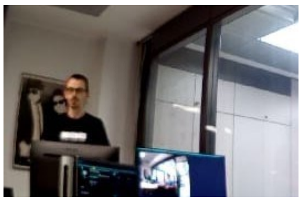


Figure 7 - distant face not detected

The pre-trained model used is the Max Margin Object Detection [13] model that is part of the *dlib* library [14] developed by Davis King. It has been trained starting from faces image gathered from publicly available dataset like ImageNet, AFLW, the VGG dataset and others.
Such detector can detect and distinguish, in the sense of not grouping them in a single box, multiple faces in the image returning their location. It is able to detect faces in quite harsh conditions as long as there are some distinguishable features. Has been observed that even low light faces are detected in most of the case (Fig. 4). Is quite robust to partial facial occlusion as long as some features of the faces are still in sight. Can be seen that in Fig. 5 even if parts of the face are covered it still be detected while in Fig. 6 where the face is more covered it is not. It also reliably detect side views of a face. Obviously has its limit and if a face is too far from the camera it is not detected (Fig. 7). To deal with

the 2D rotation problem, where a face can appear upside down or with other rotation, simply the image is rotated until a face is found.

3.2.2 Face Alignment

For each face detected in the input image, it is cropped and passed to the face alignment component. The main objective of face alignment is to obtain a normalized rotation, translation, and scale representation of the face. This is done to improve the quality of the successive recognition step since it can be seen as data normalization, helpful when dealing with machine learning techniques.
The alignment is obtained starting from a set of facial landmarks that in our case are:

- left eye corner, outside part of eye.
- left eye corner, inside part of eye.
- right eye corner, outside part of eye.
- right eye corner, inside part of eye.
- immediately under the nose, right at the top of the philtrum.

These are used to obtain faces that are:

- Centered in the image
- Rotated such that the eyes lie on horizontal line
- Scaled such that the size of the faces is the same

3.2.3 Face Quality Evaluation

Has been said that the quality of the face clusters used to perform the recognition is particularly important. They are the foundation on which the system is based for the re-identification. Since there are no constraints over which a face can be captured and evaluated, there are also no guarantees that a face is suitable for the recognition, that is: there are no guarantess that the detected faces are frontal, well illuminated, and big enough. So, we want to have some additional information about their quality to be sure that the faces stored in the clusters, which will power the recognition in the future, are good faces, in the sense that are useful for the recognition task. This is translated into the concept referred to as biometric quality. Fundamentally, the simple underlying basis to

biometric quality is that, if the biometric samples given as input to an automated recognition system are of low quality, unreliable inaccurate results will be generated. And the other way around, if the acquired biometric samples are of high quality, low error rates will be achieved.
Quality assessment techniques can be classified with respect to the amount of information they employ in order to obtain the quality measures. The classes are:

- Full reference approach (FR) , a gallery sample with high quality is supposed to be available. The system compares the features from the probe samples with the ones from the high-quality reference.
- Reduced-Reference methods (RR) just partial information of a high-quality sample is available.
- No-Reference methods (NR) do not use any reference information to compare with the probe sample.

So given that we start with no prior information about which faces we have to store and recognize, our case falls under the no-reference approach class. We have to obtain a quality score of the probe face solely based on the face itself.
To evaluate the quality of the aligned face is used the FaceQNetV1 model [15], a No-Reference, end-to-end Quality Assessment (QA) system for face recognition based on deep learning. The system consists of a Convolutional Neural Network that is able to predict the suitability of a specific input image for face recognition purposes returning a value between 0 and 1. The model is an improvement of the V0 version [16].
The main take of the model is how the ground truth information for the training has been created, that is a quality measure of the input face that can be used to train the neural network. Some work employs human perception to give a score to a face as groundtruth. Instead, to train the faceQnet model they used a performance-based groundtruth, which will result in a quality metric that represents the correlation between the input image and the expected face recognition performance of automatic systems.
So given a high-quality image A, a second image of the same subject B and a recognition system, the similarity between images A and B is strictly related to their quality. A high similarity score means that also B must be on the same quality level of A, while a low score means that B must be a worser image. In their case the high-quality image is chosen

from every person set as the one that maximizes the ICAO factors [17]. This way, by comparing an image B with a perfect image A of the same subject, they use the resulting comparison score as the groundtruth quality measure for the image B, and then use both image B and its groundtruth quality to train the model. In particular for the V1 model the similarity score is obtained as the average of the results of three different models in order to be less system dependent as possible. Since using only one the resulting quality measure would be highly accurate when estimating the recognition performance of that training matcher, but it might not be useful for predicting the accuracy of recognizers never seen before.

The model is based on the ResNet50 architecture, where the last classification layers have been replaced with two new ones designed for quality regression. In V1 also a dropout layer has been added.

Given a newly seen face to the model, it returns a quality score, between 0 and 1, that can be read as the propensity of the face to the recognition task, higher the score better will be the recognition. The score is used in by the system in various occasions to:

- Choose if a face is eligible for the recognition step. If the quality score is under a threshold the face is directly discarded without trying the recognition.
- In case of novelty detection, choose if the face can be used to create a new person or not. Since the first face is the one that will be used, at least for the first times, to perform the recognition we set an even higher quality threshold on the creation of a new person.
- Is used as a discriminant to choose the best face in a set. The best face is used for the first step of the recognition as well as part of the criteria to select which face to eliminate from the set once this has reached the maximum number.
- Is part of the score used to choose which face to eliminate from the set.
- Is part of the recognition confidence that the service reports in output.

3.2.4 Face encodings

Once we have a face aligned, we need to represent it in a way that is comparable with other faces to perform the recognition step. To do so we generate a face encoding that is a way to represent the face using a set of 128 computer-generated measurements. Those encodings are computed starting from a set of 68 face landmarks that are detected all over the face. Those landmarks are:

- chin
- left eyebrow
- right eyebrow
- nose bridge
- nose tip
- left eye
- right eye
- top lip
- bottom lip

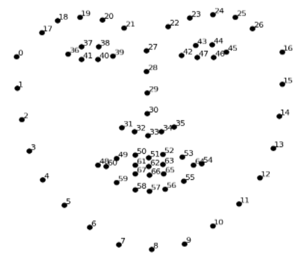


Figure 8 - Face landmarks locations

They can be seen represented in the figure. The model to detect those features has been trained on the ibug300-W dataset [18].
The detected landmarks and the corresponding face are then fed to a deep neural network to obtain the 128-dim encoding vector of the face.
This pretrained model, as well as the one for the detection, also come from the well-known *dlib* library. It is a ResNet network with 29 conv layers. It is a version of the ResNet-34 network from [19], with a few layers removed and the number of filters per layer reduced by half. The network was trained from scratch on a dataset of about 3 million faces and a total of 7485 individual identities.
Deep residual network addresses the degradation problem that arises as the depth of the net increases, which causes the accuracy to drop. A residual network is a network where shortcut connections between the layers are added, in this case the shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked

layers (Fig.9). Identity shortcut connections add neither extra parameter nor computational complexity and the entire network can still be trained end-to-end by SGD with backpropagation. A residual network shows the following improvement with respect to a plain network:

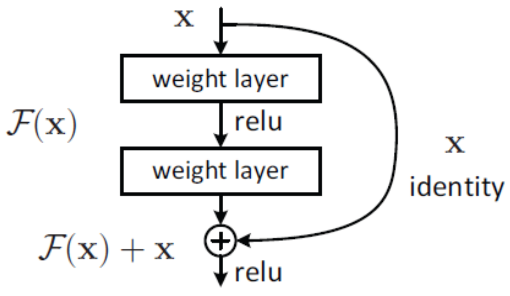


Figure 9 - Shortcut representation between two layers of a residual network

1. extremely deep ResNet are easy to optimize, while the counterpart “plain” nets (that simply stack layers) exhibit higher training error when the depth increases.
2. deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks.

Face encodings will be used as representation of the face in the following step, in fact we will not store the face image in the dataset but only the encodings tied with their quality score. The choice behind not storing the entire image face comes from the fact that, to perform recognition, we only need their encoding representation.
The encodings are used to state the similarity between two faces. Two different pictures of the same person would have similar encoding and two different people would have totally different encoding. The similarity can be stated using some distance function between the encodings, where a small distance means that the faces are similar and most likely belong to the same person and the other way around.

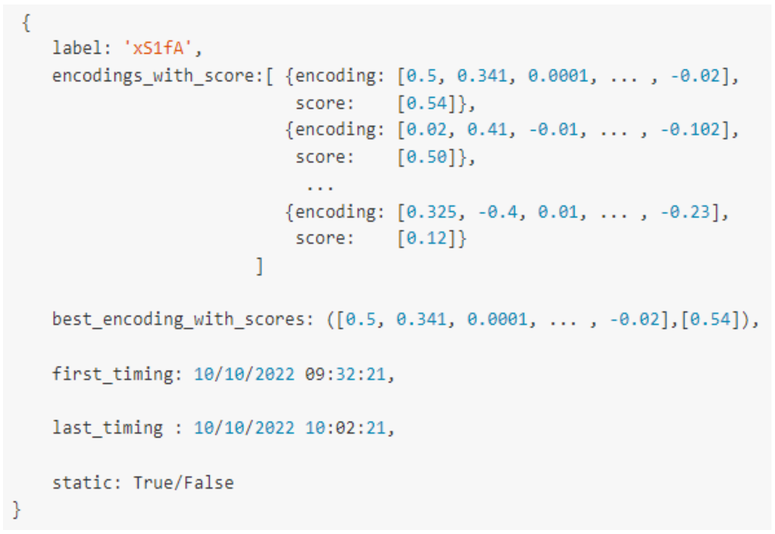
3.2.5 Face recognition

Once a face has been detected before starting the next step, we have to make sure that such face is eligible for the recognition. This is done gatekeeping all the faces with quality lower than a predefined threshold (Eq. 4).

$$Q(u) > T_{QR} \qquad [4]$$

If a face u has a low quality, it will lead to a bad recognition in the sense that there are higher chances for errors to be made. To avoid such errors a bad face which quality $Q(u)$ does not satisfy Eq. [4], where T_{QR} is the quality threshold for the recognition, is simply rejected by the service. This is done because it is better to miss some recognitions than to corrupt the databases and ruin all the successive ones. After this last check, the actual identification can start.
Has been said that the similarity between two encodings can be stated using a distance function. In this case the Euclidean distance between two 128-dim vectors has been used. Whether the distance is under a fixed recognition threshold we mark that as a re-identification and, most likely, the two encoding belong to faces of the same person.

Before talking about the actual recognition procedure, we have to explain where the encodings cluster are stored. Those are saved in a NoSQL document-oriented database.



```
{
   label: 'xS1fA',
   encodings_with_score:[ {encoding: [0.5, 0.341, 0.0001, ... , -0.02],
                            score:    [0.54]},
                          {encoding: [0.02, 0.41, -0.01, ... , -0.102],
                            score:    [0.50]},
                             ...
                          {encoding: [0.325, -0.4, 0.01, ... , -0.23],
                            score:    [0.12]}
                        ]

   best_encoding_with_scores: ([0.5, 0.341, 0.0001, ... , -0.02],[0.54]),

   first_timing: 10/10/2022 09:32:21,

   last_timing : 10/10/2022 10:02:21,

   static: True/False
}
```

Figure 10 - Representation of how the information are saved in the DB in JSON format.

The DB is a collection of documents where a document represents a person and is identified by its unique label (Fig. 10). Every document also stores:

1. encodings_with_score: The list of encodings of the detected faces upon which the recognition is performed, tied with their scores. When a new encoding is eligible for updating the dataset, it is simply added to this list or substitutes another encoding.

2. best_encoding_with_score: is the best stored encoding that is considered to be the one that better represents the face of the given person. This is the stored encoding with the highest quality score.
3. first_timing: The first time the person has been seen by the service.
4. last_timing: The last time the person has been seen by the service.
5. static: A flag that indicates if the person is a known person and has been loaded by the service owner (True) or has been detected and loaded by the robot itself (False). This will become clearer in section 3.6 that will talk about the static cluster manager.

So given that a number of persons has been saved in the DB with a number of faces stored for each of them, and a new probe is subjected to the recognition system first we have to identify the best clusters for the recognition. They are the clusters that have higher chances of representing the person which the probe face belongs to. This is done by comparing the probe with the best encoding of each cluster. The k persons with the smallest distance from the probe are kept as eligible for the recognition.

$$\left\{d\left(u, f_{b,j}\right)_1, d\left(u, f_{b,j}\right)_2, \dots, d\left(u, f_{b,j}\right)_k, \dots, d\left(u, f_{b,j}\right)_n \mid d(\cdot)_{i-1} \leq d(\cdot)_i, \forall P_j \in D\right\} \quad [5]$$

Where $d\left(u, f_{b,j}\right)$ is the distance between the probe encoding u and the best encoding $f_{b,j}$ of the person P_j. The list is ordered in ascending order.

The choice of starting the recognition comparing the probe with only one encoding for each person come from the fact that the system must work in real time and if a high number of people are saved in the database with a high number of faces per person, compare the probe with all the encodings become a very time-consuming task. Given that n persons are saved and for each person there are m faces, comparing all has complexity $\Omega(nm)$. Starting from the best encodings and going on with only the best k clusters, have complexity $\Omega(n + km)$ where in our case k is fixed to 3 such that $O(n + 3m) \rightarrow O(n + m)$.

So, after identifying the best clusters, the probe is compared with all the encodings of such sets. For each candidate, the average distance, between the probe and the stored faces, is computed (Eq. 6).

$$avgDist_j = \frac{\sum_i^m dist(u,\ f_{i,j})}{m} \qquad j \in \{P_1, \dots, P_k\}, i \in P_j \qquad [6]$$

Where $avgDist_j$ is the average distance of the probe from the faces of the person P_j belonging to the set of k candidates, and $, m$ is the number of faces in the dataset of the person P_j. Notice that m can be different for each person P_j.

If $avgDist_j$ is lower than the recognition threshold ε there is a match and the face is assigned to the cluster P_j, otherwise we fall in the novelty detection case and a new person, under certain conditions, could be created.

Other than the recognition method based on the average distance we also implemented another technique inspired by the DBSCAN clustering algorithm [20]. DBSCAN is a density-based clustering algorithm that, given a set of points, groups together points that are tightly packed. Consider ε be the radius of a neighborhood with respect to some point and $minPts$ the minimum number of points in the ε-neighborhood of a point to form a cluster. DBSCAN algorithm classify the points as:

- a point p is a core point if at least $minPts$ points are within distance ε of it
- a point q is directly reachable if is within distance ε of a core point
- a point q is reachable if there is a path that starts from a core point of directly reachable points.
- All non- reachable points are marked as outliers

If a point p is a core point, then it forms a cluster together with all points (core or non-core) that are reachable from it.

So given a probe u, a set of faces $P_j = (f_{j,1}, f_{j,2}, \dots, f_{j,m})$, a recognition threshold ε, a distance function $d(f_i, f_z)$ and T_p the percentage of points in P_j that must be in the ε-

neighborhood of u such that it is considered a part of the cluster P_j. We say that the probe u is recognized as part of the cluster P_j if

$$R_{p,j} \geq T_p \qquad R_{p,j} = \frac{\left|\{d(u, f_i) \leq \varepsilon,\ \forall\, f_i \in P_j\}\right|}{\left|P_j\right|} \qquad [7]$$

Where $R_{p,j}$ represent the ratio between the points in the ε-neighborhood of u and the point inside P_j. T_p plays the role of $minPts$ of the DBSCAN algorithm. The reason behind usage of a percentage of points and not a fixed value is that the size of the faces set grows with time and starts with no face inside. The idea behind using this method is that we can define a recognition even if not all the faces in the set are under recognition threshold but only a considerable amount. However, performance wise, the previous method was quite more robust. Also, it has been observed that if a face were recognized the percentage of points in the cluster under the recognition threshold were around 90% while dropping incredibly low in the other case, so using a percentage threshold of 30% or 40% like tested were quite useless.

However, the concept of the percentage of points under the recognition threshold ε ($R_{p,j}$) has been kept concurring to the calculation of the recognition confidence. It represents how confident the system is in assigning the point to a group. The idea behind the confidence is that a final user might want a quantitative indication of the result that the system reports in output, in order to evaluate the goodness of the operation. The confidence is calculated by weighing four statistics:

1. $Q(u)$, probe face quality. The score given in the face evaluation step concur in the confidence computation. A higher score helps increase the confidence.
2. $R_{p,j}$, ratio of points in the ε-neighborhood of the probe. If a high number of faces in the set are close to the probe there is a higher chance that the face belongs to such set.
3. $Q(P_j)$, average quality of the faces in the set. The quality of the faces database upon which the recognition is performed is crucial, a high-quality face set leads to a higher confidence.

4. $|P_j|/M$, number of faces in the set over the maximum allowed. We have said that the face set can grow over time by adding new detected faces in a person dataset. The number of faces in the set at recognition time play a role in the effectiveness of the recognition.

$$Q(P_j) = \frac{\sum_{i=1}^{|P_j|} Q(f_{i,j})}{|P_j|} \qquad [8]$$

$$C_u = Q(u)w_1 + R_{p,j}w_2 + Q(P_j)w_3 + \frac{|P_j|}{M}w_4 \qquad [9]$$

Where $Q(\cdot)$ is the quality function that return the quality score of a face, $Q(u)$ is the probe quality, $Q(P_j)$ is the average quality of the set P_j, M is the maximum number of faces per set, w_i are the weights and $C_u$is the resulting confidence.
Has been chosen to give a higher weight, respect to the other statistics, to the number of points in the ε-neighboorhood of u since is what defines a recognition. We also gave more importance to the quality of the faces in the dataset over their quantity, since having a smaller number of points of higher quality is better than having a lot, but of worse quality. The confidence is reported in output together with the label assigned to the face.

3.3 Cluster management

Has been said that the clusters representing the persons must be created autonomously by the service upon novelty detection. Also, since they are created on the fly, they need to be updated with new faces to strengthen the recognition over time, so we need to define the criteria with which a cluster is created and updated. These two operations, creation, and update are of crucial importance to the proper functioning of the service since the clusters are the basis for performing a quality facial re-identification.

3.3.1 Cluster Creation

We are in the situation where the recognition system does not match the probe to any set saved in the database. It is the case where there are high chances that a new person has been detected. In one of the first builds of the service every novelty detection led to the creation of a new person in the database. Doing so the system was pretty weak, in particular, on the first recognitions after the creation there was a tendency in misclassification. This was due to faces far away from the camera, rotated on a side, or blurred that had passed the quality check to be identified as new ones but their quality was not high enough to handle a reliable recognition solely based on them. So, they could have been good enough to be recognized over a previous existing database but not good enough to create a new cluster.

To avoid such situations, we added a stronger quality check upon creation of a new person (Eq. 10), so that the first face that must handle the first recognitions can do it in a reliable way. If a face should create a new person but do not pass the check it is rejected by the service.

$$Q(u) > T_C \qquad [10]$$

Where, $Q(u)$ measure the quality of the probe that should create a new person and T_C is the quality threshold required to create it.

When a new cluster is created the following information are saved in the database:

- Its unique label used to identify it
- It is stored the face encoding that has led to the creation, which is also marked as the best one.
- The flag static is marked as False since the set must be updated over time
- It is stored the time of the detection. It is the first time at which the person has been seen from the system and an information that can be useful to the service owner.

3.3.2 Cluster update

The pivotal point of the re-identification system is that the faces set for the recognition are updated with the fresh faces of the person when those are detected and recognized. The idea behind keeps updating the set come from the fact that the images are taken in the wild and their quality might not be optimal. So, there is always a chance that a new face is better than the ones already stored in the database. We want to create a dataset that improves over time its quality by leaving room for the best images instead of those that may perform worse.

After a face has been recognized we have to consider whether to add it to the set or not. This is done checking (Eq. 11) the recognition confidence C_u computed before. If the confidence is higher than an update threshold T_U , we can be reasonably sure that the face belongs to the set and is eligible for the set update.

$$C_u \geq T_U \qquad [11]$$

There is also a check on the static field of the cluster to update, if it is set to True the operation is not performed, the reason behind this choice is explained in section 3.6 Static cluster manager. One more thing we have to consider is that every person set has an upper limit on the number of faces that can be stored. This is done for space limit since would be infeasible to store all the detected faces, the recognition would take forever since should go through all the encodings and it would be also useless since to perform a good recognition are needed only a small number of good face images.

When a new encoding has passed the confidence check it is added to the set. The set is ordered by quality score where in position zero of the list is placed the best_encoding. When a new one is added it is checked if it can substitute the previous best encoding otherwise it is placed in the list with respect to its quality. We also go to update the field last_timing, which stores the info about when a person has been seen the last time from the service.

Now, given that the number of encodings is limited, upon a new insertion we can fall in two situations:

- The set has not already reached the upper limit and we can store the new face freely. In this case the encoding that has passed the confidence check is simply added to the set, and we proceed with a new recognition.
- The set has already reached its limit and we have to remove an encoding. In this case we have to deal with the *exceeding face removal.*

3.4 Exceeding face removal

Has been said that when we add a new encoding to a set and this has already reached the maximum capacity, we have to choose which encoding to keep and which one to remove. In this situation can happen that the new encoding replace another one in the set, or it will not be added to the cluster. To make this choice we use all the information stored in the set, which are:

1. The best encoding, which is the most representative sample of the person's face.
2. The distance of all the other encodings from the best one
3. The quality score of each saved sample.

The objective is to have a metric that measures how likely is an encoding to stay in the set. It has to measure both the quality of the encoding, since the overall quality of the dataset must improve with the recognitions, and the similarity of a face with the one that better represents the person. To do we compute a keep score Ks_i for each encoding as follows :

$$Ks_i = Sim(f_{i,j}, F_{b,j})w_1 + Q(f_{i,j})w_2 \qquad f_i \in P_j \qquad [12]$$

Where $Sim(\cdot)$ compute the similarity between two encodings and $F_{b,j}$ is the best encoding of the set P_j.

The weights are distributed to give more importance to encodings that are more similar to the best encoding rather than with a higher quality score. This comes from the fact that a higher quality face, but far away from the best one, might belong to another person and may have ended up in the set because of an error. Also, a high similarity with the best encoding implies that also the quality score is high. The safety of using the best encoding as best representation of the person that the set identifies comes from the previous checks performed on sets creation and updates. The score is computed for each encoding in the set as well as the new encode. The one with the lowest score is removed. This operation, once the set is filled, is performed every time a face is recognized as belonging to such set. This implies that the newly added probe has a chance to stay in the set and substitute another encoding as well as to be the one chosen to be eliminated since it will not improve the overall quality of the cluster.

3.5 Expired face removal

One of the constraints that we have to meet by design is that a person cannot be stored in the database for longer than the user-defined time. Such time is calculated starting from the last time the person has been seen from the service, this information is stored in the field last_timing of each face set.
The check is a sperate and independent process from the whole re-identification routine seen so far, it runs at scheduled time and inspect the entire database removing all the information of the person to eliminate. If the person return “in sight” of the service will be treated as a novelty detection and a non-static cluster will be created with its automatically generated label.

3.6 Static clusters

The main point of the re-identification system is that, on novelty detection, it can autonomously create a new person in its database and update it over time. However, there may be instances where you want to save known persons in the system. Think for example of a robot wandering in an office where there are a group of well-known persons that live the office daily. In such cases there might be the desire to create datasets for these people, who will be assigned their names as labels, with a set of face images of particularly good quality so as to ensure a consistent reliable optimal recognition.
Those datasets are called *static clusters.* The name comes from the fact that since they are generated from a set of well-known high-quality face’s images taken in good condition, there is no need to update the cluster with newly recognized faces, since the dataset is

already optimal. So, for such a cluster the autonomous update operation is not performed. Another difference with the non-static cluster is that upon creation we can choose their label. There are a set of operations to manage those clusters, they are: Cluster creation, update, and removal.

Cluster creation

Create a new static cluster starting from the face image given in input. The steps taken are the same as the re-identification routine where the image is preprocessed, goes through the face analysis, and is saved in a new cluster. In this case the quality checks are not performed since it is assumed that the face image passed in input is taken in good conditions and it is of high quality. When a new static cluster is created the label is chosen by the user.

Cluster update

For such type of cluster, the automatic update performed in the re-identification routine is not allowed. It is so, since it is assumed that the faces used to create the cluster are of high quality and the images taken by the robot will hardly match them, so to avoid ruining the set the update is interdicted. When a face is recognized as belonging to such cluster, all the information regarding the label and the confidence are still reported in output, but the update step is skipped.
There is, however, the possibility of updating the cluster using a manual update similar to the static cluster creation. This is done passing in input new images that the user wants to add in the dataset of a person. Other than the face image to insert is also given the label of the set to update, if such label does not exist in the database a new person is created.

Cluster removal

Simple operation to remove a static cluster from the database given its label.

4. Service design

In the previous section has been presented the proposed solution to the re-identification problem with particular focus on the steps taken in the re-identification pipeline describing how they work and how they have been implemented. However, it was a high-level presentation of the implanted service while in this section we want to focus more on the actual design of the service. What other solutions are present in the market, how the different implemented libraries are used to solve the different tasks, how they are structured, how the robot communicates with the service and how has been managed the server design.

4.1 Available service analysis

Before starting to implement the service a number of third parties services for face recognition have been analyzed to understand how the problem was handled, which kind of services were already available and if any of them could help solving the reidentification problem. Among the others the following services has been analyzed:

- Amazon Rekognition, service of the Amazon Web Services that provides an API to perform facial detection and recognition on a given image. Such images can be passed in .jpg or .png image format or stored in their database (Amazon S3 bucket). The face identification is performed over a set of faces collections, where each collection represents a person, previously stored.
- Azure face, a Microsoft's service. It provides an API to access to AI algorithms that detect, recognize, and analyze human faces in images. Allow for the creation of a Person group to store the faces of the people to match. After detecting a face in a given image can perform Identification based on the groups stored.

Other services such as Kairos, Face ++ or Betaface provide, in various form, the possibility of creating a DB of people. For each person can be uploaded different images of the face, for example taken from different angles, to make the recognition more accurate. Then an image containing a face can be uploaded to the service and matched against the people in the DB to state if it is present or not.

These services solve specifically the recognition problem, to deal with the re-identification a possible implementation is to evaluate the recognition confidence returned by the services and if it is considered too low create a new person in the saved collection.

After this first analysis of the available services, the research focused on open-source libraries for the face recognition problem that could be used to implement our re-identification system. The libraries analyzed, in many cases, were a wrapper for different face detection and encoding pre-trained neural network models. We focused on libraries such as:

- Deepface, which offer a high number of pre-trained models for both face detection and embeddings.
- CompreFace, offers ready to use docker image. However, the fact that comes as a docker image nullifies any possibility of using it to solve our tasks. Moreover, it is not optimized to work with a huge number of faces stored in as DB.

The drawback of such libraries is that in most of the cases they are created to deal with the recognition problem in an end-to-end manner, which means that they accept a given face image and a dataset and return if such face is present or not, without returning any other information. So, applying them to another problem was difficult.

In the end the choice fell on the FaceRecogntion library, which is an elegant and ready to use wrapper of many functions and models of the already cited *dlib* library. It gives us the freedom to create and manage the already presented functions to solve the reidentification problem, moreover the fact that it uses functions from *dlib*, allows us to use functions from the latter seamlessly.

4.2 System Structure

After giving a description of the available services offered by the implemented system and their pipeline in section 3, here we want to discuss the actual structure of the system,

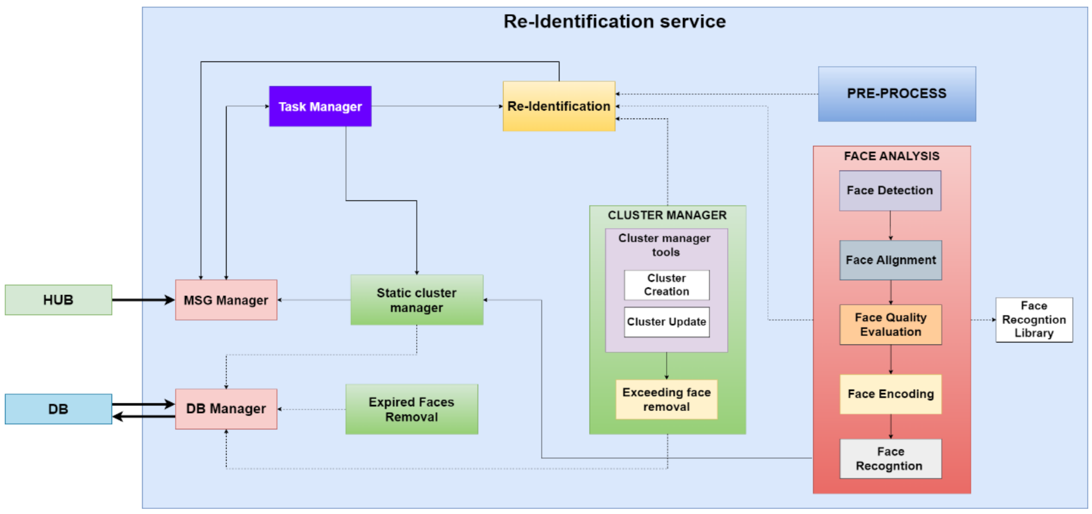


Figure 11 - Structure of the re-identification service exposing the relation between the libraries, the services, the managers, and the external components (HUB, DB)

describing briefly which libraries have been defined, how and by which services are used. The two main services offered are the re-identification of one or more faces in an image and the side service for the management of the static person. These are based on three libraries, and each one manages a different aspect of the service. The libraries are:

Preprocess library

It defines all the functions used to preprocess the image received. This includes the histogram equalization, the denoising, the usability check. Moreover, it defines the function used to decode the received image from the base64 encoding, that is used to send images to each other between different services and the robot, to a format compatible with the service.

Face Analysis library

It is the library that operates directly on the face and defines the functions in charge of the face analysis. Some functions like the face detection or the face encodings, are based on the FaceRecognition library. Other defined functions are the recognition function, which return whether the face is recognized or not, the confidence computation and other utilities tools such as the one to get the boxes of the face locations or to compute the encodings distances. It is also in charge of loading the models, so that this is done one time when the service is started and avoids loading and dismissing the models every time they are invoked. Such kind of operation would take a lot of time, which is not compatible with the requirement of a real-time service.

Cluster manager library

This library is in charge of managing the people sets stored in the database. Other than the functions to create and update such clusters, as well as the one to remove the exceeding faces already presented, it defines all those utility functions to proper manage the database like:

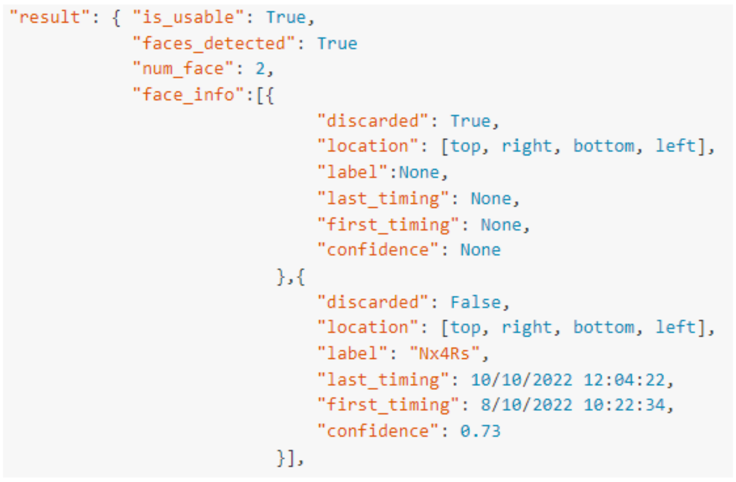
- Get the list of people stored in the database
- Get the number of encodings for a specific cluster
- Get only the last_timing associated with the label of all the people
- Get the best_encoding of each set

Many of those can be useful also to the final user, which might want to know the situation of its database, so functions like getting the list of people stored or only the list of static persons, are exposed such that they can be used also by the user. This library as well as the *static cluster* manager and the *expired face removal* function are linked to a DB manager and to the external component DB. This is why these are the functions that are actively going to write and read into the database. The DB manager is simply an API that allows the functions to access the documents of a specific collection in the DB.

These three libraries are used by the two main operations defined: the re-identification and the static cluster management. Those are the main operations that a user can request to the service.

Re-Identification

Is the main service provided to the user and it implements the pipeline seen in Section 3. It takes in input an image, preprocesses it, and proceeds the with the face detection. For each face detected in the image is evaluated its quality, to decide whether to discard it or proceed with the operation, it is encoded and passed through the identification process. The service uses the cluster manager tools to create or update the person set when needed.



```
"result": { "is_usable": True,
            "faces_detected": True
            "num_face": 2,
            "face_info":[{
                              "discarded": True,
                              "location": [top, right, bottom, left],
                              "label":None,
                              "last_timing": None,
                              "first_timing": None,
                              "confidence": None
                          },{
                              "discarded": False,
                              "location": [top, right, bottom, left],
                              "label": "Nx4Rs",
                              "last_timing": 10/10/2022 12:04:22,
                              "first_timing": 8/10/2022 10:22:34,
                              "confidence": 0.73
                          }],
```

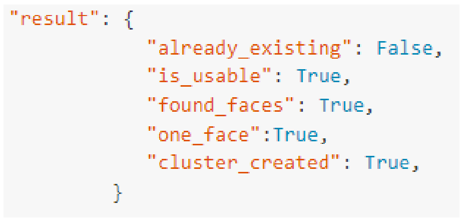
Figure 12 - Result message of the re-identification service

Once all the faces have been evaluated the service return a message (Fig. 12) containing the results of the re-dentification:

- is_usable, contain the result of the usability check performed in the preprocess, if it fails the image does not go through the re-identify process
- faces_detected, tells if there are faces in the image. In case no face has been detected the image does not go through the re-identify process.
- num_face, report the number of faces detected in the image
- face_info is a list of the size of the detected face, every member of the list reports the information about a single face.
- discarded, tells if the face has been discarded due to low quality.
- location, report the location of the face in the image
- label, the label associated with the identified face.
- last_timing, report the last time the person has been seen before this identification
- first_timing, the first time the person has been identified by the service.
- confidence, the recognition confidence

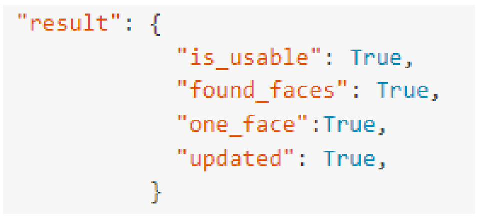
Static cluster manager

This is the other service offered to the user, it is more of a side service with respect to the re-identification since is used only manages a specific use case. It takes in input an image and a label in case we want to create or update a cluster or only the label of the person to remove. It is linked (Fig. 11) to the face analysis and preprocess libraries because the input images must go through many of the steps that are also involved in the recognition, like the quality evaluation or the encoding, in order to store them in the DB. All the operation returns a message with their result.



```
"result": {
              "already_existing": False,
              "is_usable": True,
              "found_faces": True,
              "one_face":True,
              "cluster_created": True,
          }
```

Figure 13 - Result message of the create operation



```
"result": {
              "is_usable": True,
              "found_faces": True,
              "one_face":True,
              "updated": True,
          }
```

Figure 14 - Result message of the update operation

Considering the creation of a new static person the output message has the following structure (Fig.13):

- already_existing, tell if the person we want to create already exist, the check is performed only checking on the label given to create the cluster
- is_usable, is the usability check
- found_faces, tells id there are faces in the image
- one_face , check if there is only one face in the image. In case more than one face Is detected the image is rejected since the system could not know on which face create the new person.
- cluster_created, tell the operation has been successful

The update operation checks if the person exists, if it does not invoke the create operation or otherwise proceed with the update, the result message (Fig. 14) is similar to the one for the creation. For the removal is simply reported in output the label of the person removed and a message of success. Two more operations that can be invoked by the client are the ones to obtain the list of all the persons or only of the static persons stored in the DB. In such cases it is simply a message with a list of the people's labels.

4.3 Server structure and communication

So far, we have discussed about how the services and the libraries are linked inside the system and which kind of output we can expect from them. However, we still have to explain how the system actually communicates with a client that wants to request a specific service among the ones offered, which protocol is used to handle the communication, how the input and the output are formatted, which is the server organization and how the different requests are managed by the system.

WebSocket Protocol

The communication between the system, hosted on a server machine, and the client is managed using WebSocket. It is bidirectional, a full-duplex protocol that is used in the scenario of client-server communication. The reason it was chosen over other protocols such as for example HTTP is that it is a stateful protocol, which means the connection between client and server will keep alive until it is terminated by either party (client or server). The handshake between the two parties is performed only one time when the connection is established, from this point message exchange will take place in bidirectional mode until connection persists between client-server. This is one of the main differences with the HTTP protocol which instead must establish a connection performing a three-way handshaking between client and server every time a new request is made. This significantly slows down the communication speed where WebSocket ensures a low-latency communication which is very well suited for applications that must work in real-time like ours.

Image HUB

We have said that the system is hosted on a server machine and that the clients can do their request with the operation to perform on the given image. However, between the client that actually captures the images, like robots, camera systems etc.… and the re-identification service there is a broker: the *image HUB*. The image HUB (Fig. 15) is an intermediary between the clients and the server, in fact starting from an image taken from a robot we might want to carry out multiple operations since it might contain a lot of information, we might perform pose recognition of a person in front of the robot while we are also identifying it. The HUB, received an image from a client, distributes it to all the services from which the client wants to get a result.

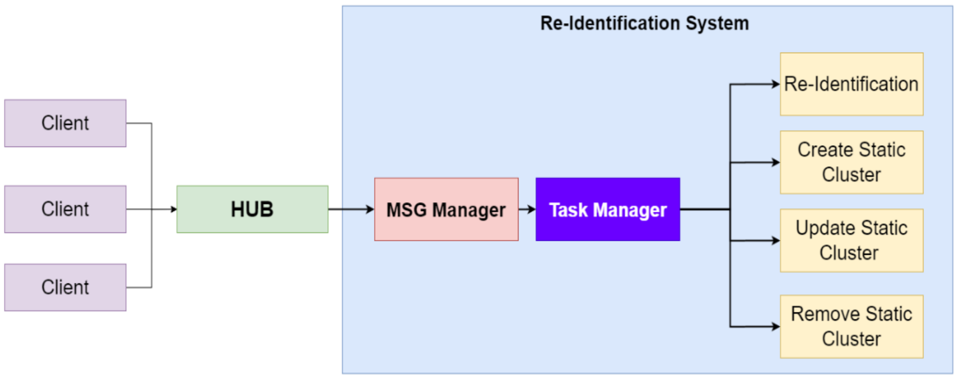
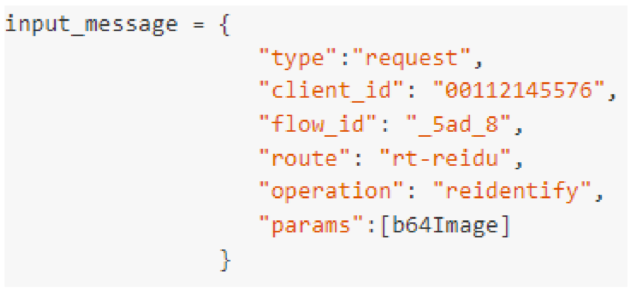


Figure 15 - Server structure of the re-identification system

Message manager

The HUB directly communicates with the message manager. The message manager is the actual server component of the system, and it is in charge of managing the client connection. The client, in our case the HUB, sends a message containing all the information to request a specific action from the system and to identify the client who requested it. The input message received by the message manager can be seen in Fig. 16, where:

- type, describe the type of the message received. In the work environment of the service exits three types of messages:
    - *Request*, it request as service to perform a specific required function



```
input_message = {
                    "type":"request",
                    "client_id": "00112145576",
                    "flow_id": "_5ad_8",
                    "route": "rt-reidu",
                    "operation": "reidentify",
                    "params":[b64Image]
                }
```

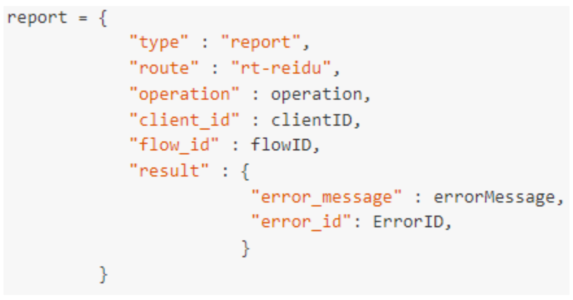
Figure 16 - Input message received by the message manager from the HUB to perform the re-identify operation

    - *Order*, mainly used to give a task to a robot, like moving to a specific place
    - *Report*, used to return an outcome of the previous type of messages. It can contain the output of the service or the robot's state at the end of the action performed.
- client_id, uniquely identify the client who made the request
- flow_id, this is used to identify the message sent, it could be used to follow the message among the service through it which passes
- route, identify the context of the request, in case of system the offer a number of services the route can be used to identify to which of them the message Is refer to and direct it correctly. In the case of the re-identification, the route it is named "rt-reidu" that is the acronym by which the service is referred to.
- operation, define the specific operation to perform the operations that can be performed are:
    - reidentify, starting from the given image
    - createStaticCluster, to add a new static person in the DB
    - updateStaticCluster, to update a static cluster adding a new face
    - removeStaticCluster, to remove a cluster given the label
    - getPersonList/StaticPersonList, to obtain a list of labels of the stored persons.
- params, is a list containing all the input parameters required by the service to perform the operation. In the case of re-identification only an image encoded in base64 is requested, in the case of creation of a static cluster both an image and the label are requested.

The message manager performs a first test about the correctness of the message checking if it has been specified an operation to perform and if the field is not empty. In such cases it reports an error message to the client, otherwise the message is sent to the Task manager.

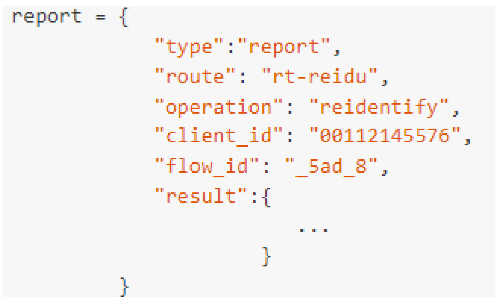
Task manager

This component addresses the messages to the corresponding service in charge of performing the requested operation. The message manager and the task manager are kept separate mainly for scalability reasons. In case we want to expand the system by adding new services we can have different tasks managers while working with the same message manager. It still receives and directs the messages to the correct one, in this case the route field of the input message will be used.
The task manager performs a series of specific checks on the message integrity. First of all, it is checked that the requested operation is among those that the service can offer. The images are exchanged between the parties encoded in a base64 string. In case there is one in the parameters list it is checked that such string is not empty before passing it to the service. In case one of these checks fail, an error message is returned to the client (Fig. 17). In case the requested operation is successful a message of report type is sent to the client containing the result of the operation (Fig. 18).



```
report = {
            "type" : "report",
            "route" : "rt-reidu",
            "operation" : operation,
            "client_id" : clientID,
            "flow_id" : flowID,
            "result" : {
                          "error_message" : errorMessage,
                          "error_id": ErrorID,
                        }
        }
```

Figure 17 - Report message when an error occurred



```
report = {
              "type":"report",
              "route": "rt-reidu",
              "operation": "reidentify",
              "client_id": "00112145576",
              "flow_id": "_5ad_8",
              "result":{
                          ...
                       }
          }
```

Figure 18 - Standard format of an output message. The result field is filled with the output of the specific operation invoked

The information contained is the same as the input message, but where the type is defined as a report since it returns the result of the operation. The field result, in fact, contains the outcome of the service as presented in section 4.2 (i.e., Fig. 12) or the error information, in case an error occurs. Such information are:

- error_message, which contains the message that explains which error has occurred.
- error_id, an identifier of the error.

5. Performance analysis

Now it is time to test the re-identification service and analyze its performances. Before talking about the tests carried out and how the data are analyzed, we have to present the parameters that control the quality of the operation. Those are the ones that will be tuned during the tests to obtain a performing system. Such parameters are:

- RECOGNITON_THRESHOLD: represent the distance threshold between two encodings to consider them similar and so belonging to the same person. If the distance is below such threshold we consider it as a match, otherwise they are different person. It is the main parameter that control the recognition system, a fine tune on it implies better performances over the entire system
- FACE_QUALITY_THRESHOLD: is the minimum quality score required to a face to be analyzed by the system and not be discarded. A higher value implies that there will be a higher number of not analyzed detected faces, but the DB quality and the re-identification result would be better.
- FACE_QUALITY_CREATION: is the minimum quality score required to a face that has not been identified in any other faces set, to create a new person in the database. Higher values implies that a new face is rejected by the system until its quality is high enough to start the re-identification.
- UPDATE_CONFIDENCE: is the minimum recognition confidence required by the system to use the probe face to update the set with which it is matched.

These parameters have been already introduced in section 3, however is important to report them again to have a clear idea of their purpose in the following.
Has been conducted two distinct types of tests:

1. First a test on some videos with a known number of identities to identify a good starting value for the RECOGNITON_THRESHOLD to use in the following.

2. Then a series of tests was carried out with images coming from a robot. During such tests, different combinations of the parameters value have been tried to identify those that guarantee the best results.

The first set of tests has been done before the system was connected to the image HUB and to the robot. Here the objective was simply to identify a good starting value for the RECOGNITION_THRESHOLD. To do so, the system has been evaluated on a series of videos with a known number of faces and identity, with different values for the said parameter. At the start the database is empty and is populated at running time while fresh faces are discovered. Here are reported the results obtained from two videos, one where two faces of a male and a female are present (Fig. 19). And the other with six different persons, both male and female (Fig.20 -21).

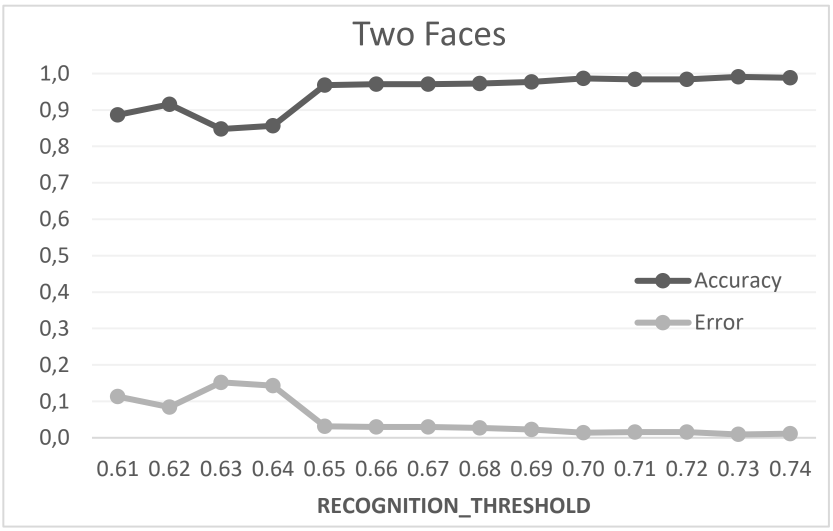


Figure 19 - Accuracy and error for a set of values of the recognition parameter

The accuracy is obtained considering only when a face is successfully matched with the label with which has been identified for the first time by the service. While in the error has been grouped all the misclassification in other labels and also the creation of a new person for a face already present in the database. Can be seen from the graphic (Fig. 19) that for values smaller than 0.65 there is a low accuracy. This happens because a low recognition threshold implies that two faces must be very similar to be matched. If we consider slightly rotated faces or with a hugely different illumination can happen that are not recognized, and new set is created in the DB. For higher values, the situation stabilizes with a higher accuracy. We have to keep in mind that the two faces considered belong to quite different persons, so the same behavior does not persist if we expand the set of subjects.

To test the system in this second case we have decided to constraint the values of the RECOGTION_THRESHOLD between 0.64 and 0.72. As before the accuracy considers

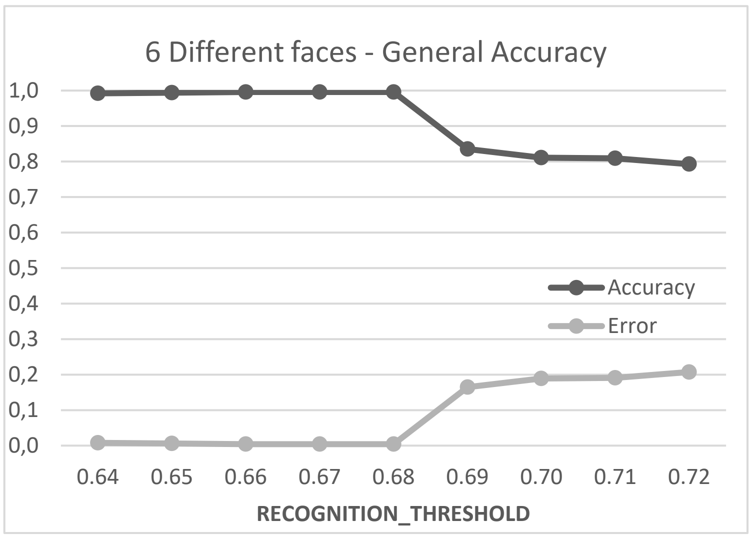
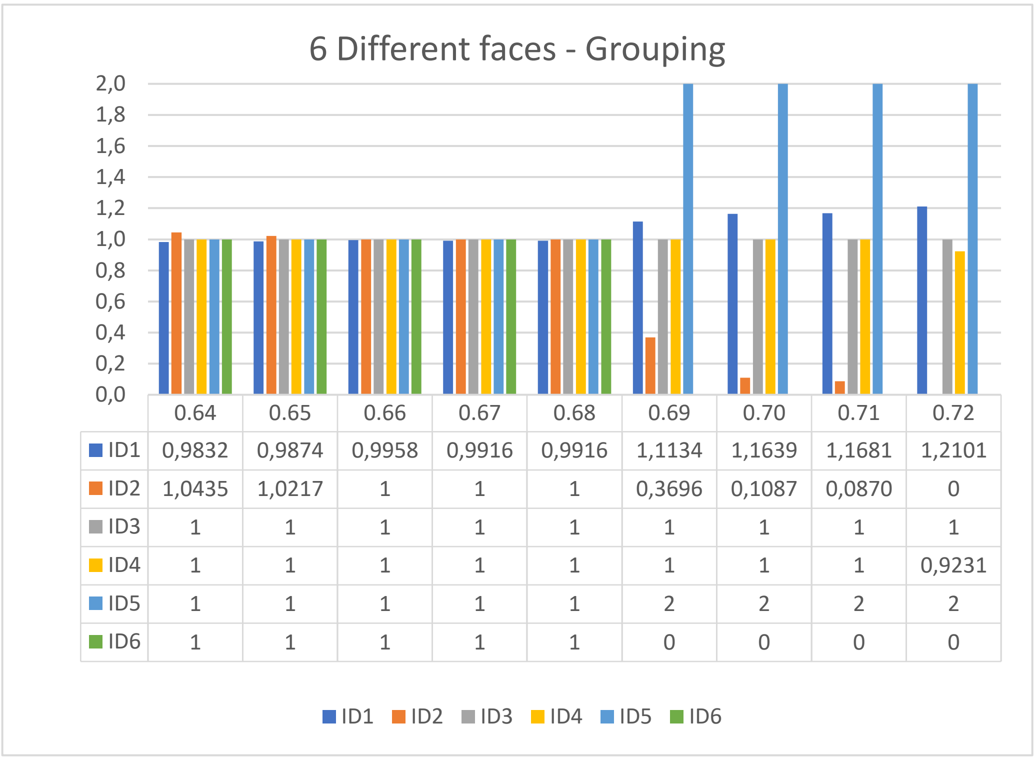


Figure 20 - General Accuracy and error with 6 different subjects considered.

only the true positives over all the association and the error considers all kinds of misclassification that the system can do. From Fig. 20 can be seen an opposite behavior to what we have seen before, with higher values of the parameter the error goes up. This can be seen more clearly in Fig. 21 where the percentage of face associated with each ID is depicted for the different values of the recognition threshold. A value higher than 1 means that faces belonging to another person have been associated with the wrong ID. Can be seen that for threshold around 0.66 to 0.68, most of the faces are correctly associated to their ID, aside for some faces of the ID1 that are wrongly associated to a newly created ID but can be seen that the accuracy is still high. On the contrary for values higher than 0.69 can be seen that there are a lot of misclassifications. ID5 embeds the faces belonging to ID6, while ID1 starts to be matched with faces belonging to ID2 and later even to ID4.

It happens because the recognition threshold is too high. This implies that even if the probe encoding and the encodings in the face set are quite distant, their distances still are under the threshold and so it is recognized as belonging to the given set. For too high thresholds we have the opposite problem of what has been observed in Fig. 19.



	0.64	0.65	0.66	0.67	0.68	0.69	0.70	0.71	0.72
ID1	0,9832	0,9874	0,9958	0,9916	0,9916	1,1134	1,1639	1,1681	1,2101
ID2	1,0435	1,0217	1	1	1	0,3696	0,1087	0,0870	0
ID3	1	1	1	1	1	1	1	1	1
ID4	1	1	1	1	1	1	1	1	0,9231
ID5	1	1	1	1	1	2	2	2	2
ID6	1	1	1	1	1	0	0	0	0

Figure 21 - Specific face - ID association when 6 different persons are considered

Here different faces are considered belonging to the same person and the number of clusters is lower with respect to the number of different persons detected.

So, depending on the value chosen for the recognition threshold the system has two different behaviors:

1. With low values, the threshold is too strict to perform a correct matching. Even if two faces belong to the same person, they might not be recognized due to the high similarity required by the parameter. This leads to the creation of a new cluster. Low recognition threshold implies having more clusters associated with the same person.
2. With high values, the threshold is too loose to perform a correct matching. Two faces that belong to different people might be associated with the

same ID since the distance threshold is too high. High recognition threshold values imply to have less cluster than the number of different persons detected, because multiple persons are associated to the same cluster.

These are two behaviors to avoid in our system since they can create cascading problems that cannot be resolved at running time. If we have multiple sets in the DB associated to the same person, the system can associate the person to each one of the sets losing consistency. On the contrary, too loose threshold implies that different persons are associated to the same cluster, with such set that will contain a number of faces of random people instead of only the correct one.
To avoid such problems can be observed from both Fig. 19 and 20 that a good set of values for the recognition threshold are the ones that go from 0.66 to 0.68, where the re-identification is consistent.
The other tests have been carried out connecting the system to the image HUB, with the images sent by a mobile robot every 0.5 seconds, this is for limitation of the robot used which is a Temi V2 equipped with a wide-angle monocular RGB camera with FOV 120 degrees. For the test a resolution of 640x480 has been used since the use case of the robot. The videos on which the tests were done came from a robot in an office environment with a series of people entering the camera's field of view. The ability to consistently re-identify people and update the database is tested.

Figure 22 - Temi V2

To do so is used a Consistency Confusion Matrix (CCM) [21] which measures the consistency of the predicted results. Where the true ID of a face is considered the one associated first to the person. This approach is actually evaluating the Re-ID system's abilities of consistently re-identifying the same face as the same ID. Table 1 is an example of a CCM. Each column represents a different face detected by the system while each row has a predicted ID. In CCM, the sequence of the row-column is sorted so that the True Positives (TP) can be found in cells $\{i , i\}$. For instance, for the i^{th} column (the ID) of a

P row by Q column CCM, the i^{th} cell contains the True Positive (TP), while the other cells from 1 to Q excluding i^{th} contain the False Acceptance (FA) where the current face is associated with an ID of another face. The $(Q+1)^{th}$ row up to P corresponds to the False Rejection (FR) where the probe is given a new ID although its correct ID already exists in the gallery. In Table 1 the ID5 represents a new ID for Face1 even if its ID was already present in the system.

	Face1	Face2	Face3	Face4
ID1	239	0	59	3
ID2	0	26	10	0
ID3	0	0	85	0
ID4	0	0	0	24
ID5	34	0	0	0

Table 1 - Example of CCM. Each column corresponds to a different face and each raw to a predicted ID

So given that the sum of all the CCM is given by:

$$sum(CCM) = \sum_{i=1}^{P}\sum_{j=1}^{Q} CCM(i,j) \qquad [13]$$

The overall True Positives can be found by:

$$TP = \sum_{i}^{Q} CCM(i,i) \qquad [14]$$

And the accuracy of the system is computed as:

$$Accuracy = TP/sum(CCM) \qquad [15]$$

the Re-ID accuracy is the main evaluation approach for the Re-ID system's performance. In addition to the Re-ID accuracy, False Acceptance Rate (FAR) and False Rejection Rate (FRR) are also important evaluation protocols and can be obtained from the CCM. The number of False Acceptance can be obtained as:

$$FA = \sum_{i=1}^{Q}\sum_{j=1}^{Q} CCM(i,j)\,, \quad j \neq i \qquad [16]$$

And the False Reject obtained as:

$$FR = \sum_{i=1}^{Q} \sum_{j=Q+1}^{P} CCM(i,j) \qquad [17]$$

While both rates are obtained as:

$$FRR = FR/sum(CCM) \qquad [18] \qquad FAR = FA/sum(CCM) \qquad [19]$$

We have to point out that the first bunch of tests with the robot have been carried out while the checks on the creation (Eq. 10) and the update (Eq. 11) of the clusters were not implemented yet. This led to a mediocre performance of the service with a low accuracy and a lot of both False Acceptance and False Reject. Many of the sets were degenerated with a mixture of faces associated to each ID with the system matching multiple people to the same ID. To improve the quality of the re-identification the two checks have been implemented.
Have been tested out different values of the three parameters governing the performance of the system. For the different tests has not been used the same video captured by the robot simply changing the values, but every time the robot captured a different video of the test environment. Every test started with the empty database. We report in particular some combination of values to analyze the behavior of the service.
With the following parameters:

- RECOGNITON_THRESHOLD = 0.675
- FACE_QUALITY_THRESHOLD = 0.08
- FACE_QUALITY_CREATION = 0.22
- UPDATE_CONFIDENCE = 0.65

We reached an $Accuracy = 0.502$ with FAR = 0.498, while there was no False Reject. However, can be observed from Table 2 that were presented to the system three different faces where only 2 IDs has been created. Also, there are a lot of False acceptance. This can be imputed to the

	Face1	Face2	Face3
ID1	177	295	58
ID2	0	180	0

Table 2 - CCM

recognition threshold to high. Already from this test can be observed a characteristic behavior of the system and one of its main problems, the fact that there are high chances to associate two different labels to the frontal view and the lateral view of the face. This is what happened to Face2 and ID1, mostly of the face associated to ID1 were sideview of Face2 while the frontal face is assigned to ID2. In the end the faces set representing ID1 degenerated since is updated with faces belonging to all three persons. Observing that the quality of the faces saved in the database in the top positions were around 0.40, has been decided to higher the FACE_QUALITY_CREATION. To improve the re-identification performances, we strict all the threshold, lowering the RECOGNITION_THRESHOLD to try to get rid of the False Acceptance problem and increasing both FACE_QUALITY_THRESHOLD and UPDATE_CONFIDENCE. The reason to higher the confidence required to the update is to strengthen the sets and try to update them only with faces we are sure belong to the person associated.
The best results (Table 3) have been obtained with:

- RECOGNITON_THRESHOLD = 0.67
- FACE_QUALITY_THRESHOLD = 0.1
- FACE_QUALITY_CREATION = 0.25
- UPDATE_CONFIDENCE = 0.67

We reached an $Accuracy = 0.78$ with FAR $= 0.15$ and $FRR = 0.07$. The reason the UPDATE_CONFIDECE seems low come from the fact that the highest confidence achieved for the recognition are around 0.72/3, this is due to the fact that the quality of the faces hardly passes the 0.50 threshold, and this lowers the confidence. With respect to the previous scenario the number of False acceptances drastically decreases, while a new problem manifests in fact for the Face1 has been created a new ID, ID5, while there was already one in the DB. Looking at the faces associated with such ID can be noted that ID1 mainly matches with side views of Face1 while ID5 has been created for frontal close faces of the person.

	Face1	Face2	Face3	Face4
ID1	239	0	59	3
ID2	0	26	10	0
ID3	0	0	85	0
ID4	0	0	0	24
ID5	34	0	0	0

Table 3 - CCM

Also, we can notice that the same happens for Face4 whose side views are associated with ID1 while the frontal is associated with ID3 (Fig. 23). Another thing that can be noted looking at the labelled data is that distant faces are more easily mixed than those that are closer. This can be due to the low resolution of the camera used that does not allow features to be extracted clearly as could be done with a closer sample.

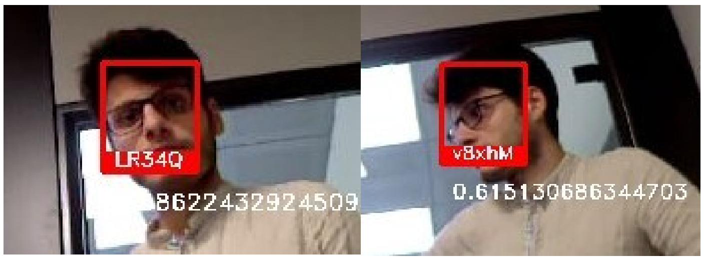


Figure 23 - Example of different association for a side and a frontal view

After increasing the UPDATE_CONFIDENCE can be observed that the sets of the faces in the DB do not update so frequently as before. In fact, reaching a confidence of 0.67 in the firsts re-identifications is quite hard because all the parameters influencing such metric are low: there are only a few faces in the set over the maximum allowed, in our case 25, and the quality, has already said, does not reach high values due to the camera resolution. So, updating the sets is not immediate. Regarding the confidences is observed that, as expected, are lower for the first re-identifications and start increasing as the set gets filled.

As expected, if a set is initialized in a clever way, which means the first time a person enters the robot's field of view, images covering all sides of the face are captured, the performance increases. This can be seen in Table 3 where the set corresponding to the ID2 has been created in such way unlike the others. Can be seen that there are a small number of other faces assigned to such ID as well there no Face2 assigned to others ID. Can also be observed that when Face3 is associated with ID2 the confidence was low, so this does not contribute to updating such set. At test time it can be observed that increasing both FACE_QUALITY_THRESHOLD and FACE_QUALITY_CREATION the number of discarded faces increases. We value more to avoid analyzing some faces rather than risk corrupting the sets already created with inferior quality faces.

Considering the occlusion problem, has already stated in 3.2.1, the detector is able to detect a partially occluded face if the occlusion is not too strong. As can be seen in Fig. 24 where even if some the face is partially coverd the system is still able to re-identify it.

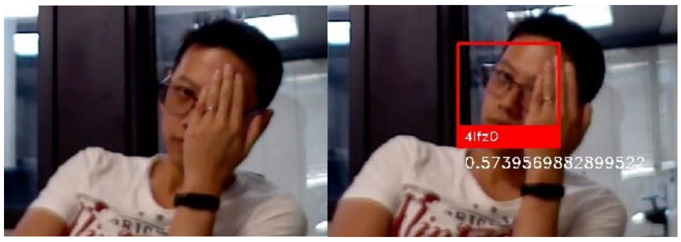


Figure 24 - Partially occluded face not detected and detected and identified

Also deals well with multiple people in its field of view and is able to distinguish them all (Fig. 25).

The computation time to analyze an image is in the order of milliseconds where, to obtain the result to consider the system operating in real time, the requested computation time was at most a second. The threshold of a second has been chosen since, in this case, the robot should interact with a person and a second is considered low enough to achieve human-robot interaction.

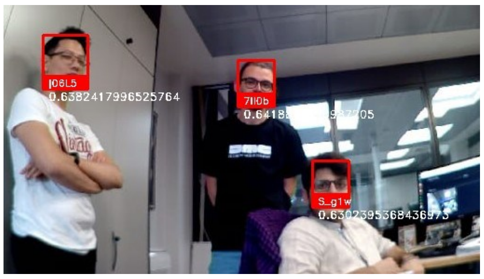


Figure 25 - 3 Way Re-Identification

Playing around with the values of the parameters we have to deal with the same major problem that is the creation of a "side-view set" where a lot of the side faces are grouped. While creating another set for each person that contains their frontal view. The creation of this big dataset is also true for the most distant detected faces. Probably due to the low quality used, the feature extracted does not carry enough information to distinguish distant faces. The system performs quite well considering closest face to the robot.

Another aspect of the system we want to point out is the evolution of the datasets saved in the DB. As already said every encoding is tied with its quality, that can be interpreted as its propensity to the recognition task. This metric is used to evaluate the overall quality of the database where, with higher values, we can expect better performances of the system. Can be observed that their average quality tends to increase with the number of the successful re-identification. While as long as the set is not filled up to its maximum every encoding that pass the confidence check is added (Eq. 11), so the overall quality might not be optimal since the first objective is to fill the set. When the set is full can be observed that the faces with worse quality, as explained in section 3.5, are removed leaving room to the better ones with the overall quality benefitting. While right after the creation of a set we can also expect to accept encodings with their estimated quality of 0.15/0.20, when the number of re-identifications performed on the same person increases, we leave room to encodings around 0.35/0.40 and that better represents the person. Such considerations on the quality obtained must be analyzed considering the instruments used to gather the faces as well as keeping in mind that there is a strong dependence on the way in which the person stands before the camera. If the person puts their face clearly in front of the camera, we can expect to obtain an excellent set right at the start.

6. Future work

We think that there is a lot of room for improvement in the re-identification system proposed as well as in the research field. While working on the development of the project, a number of complications were encountered that had to be dealt with and some of them could not be addressed. In particular can be observed that the accuracy achieved is not satisfactory. From the analysis carried out it appears that a major problem that affects such low performances is that the system tends to create a face set containing the sideview of the faces detected in its field of view and is unable to associate them to the different persons. In the end we have a big dataset associated to most of the faces sideview while another dataset for each person associated with its frontal view. The inability to links the sideview to the frontal view of a face heavily affect the performances of the service since most of the misclassification come from this problem.

To solve this, a possible solution could be to use a head pose detector in order to associate to each detected face the information about its pose. This is done to create encodings sets that not simply store the encodings of the face but can group them based on the detected orientation. Doing so, ideally, we have a set that cover all the orientation of the head. The detected face, knowing its orientation, can then be matched against the encodings in the set with the same orientation. Doing so the chances of misclassification should diminish.

Another situation in which the system struggled involved faces farther from the camera. Such cropped faces can have a low resolution that can affect the feature extraction and the computation of the encodings. To cope with this problem can be investigated upsampling techniques to obtain a higher resolution version of the face without losing information.

All of these new techniques that could be used to improve the service must be carefully analyzed and tested to not affect too much the recognition time. In fact, adding a new deep learning model, as could be the one for the head orientation detection, on top of the ones already present can significatively affect the processing time.

The face detection and recognition topics are an evergreen field of research with new models and algorithms developed that improve the performances of the previous ones. Trying new models can help increase the accuracy of the service. Also, more research can be done on other clustering techniques that could be used to associate the probe face to the faces sets as well as deepen the development of the algorithms used to update the database. Refining these two aspects can significatively improve the overall quality of the service.

7. Conclusion

The main objective of the research is to implement a face re-identification system that can update its database, initially empty, with new individuals as they are identified. It was also intended that such face sets would improve with the number of re-identifications made on each person, where the higher quality faces, if any, replace the lower quality ones. All of this with the final goal of running the system using images captured from a robot and achieving low processing time to enable real-time human-robot interaction. The reason behind investigating the re-identification task comes from the interest in the discipline and in particular from its application for social robots. With such a system, the robot can have the flexibility to adapt its knowledge about the person encountered. For example, using the timestamp saved, it can adapt its behavior with respect to how much a person live the place where the robot acts, offering a different amount of information or guiding the person through the place.

Overall, we have implemented a re-identification system that achieves real-time performances and can be used on mobile robots. The final accuracy obtained is equal to 0.78. The low value is mainly due to the major problem encountered at test time that is the creation of a "side-view faces set" where a lot of the side faces are grouped in, while also creating another set for each person that contains their frontal view. This heavily affects the quality of the service and is responsible for most of the misclassification. While we have achieved a successful face re-identification for persons relatively close to the robot, the performances for distant faces worsen. The reason can be found in the difficulty of extracting meaningful features from very low-resolution cropped faces. To cope with such problems can be investigated a head pose detector and tie such information to every face to enhance the recognition, as well as some upsampling technique for the cropped faces. We have also observed a successful re-identification while multiple persons stand in the field of view of the camera, which was one of the most interesting use cases of the service.

We have implemented a first concept of a self-improving database. Using the quality score associated with each analyzed face, the system is able to choose whether to update or not its set and to choose which encoding should be replaced to improve the quality of the successive identification. The choice of implementing such system comes from the fact that the images do not come from a controlled environment and their quality is not guaranteed. There is always a chance that in a re-identification a face captured in excellent condition will show up. We did not want to be forced to use only the first n images captured the first time, but we wanted to be able to take the best advantage from the new detections. This is also one of the most interesting points of the research. Developing a re-identification system, and in particular its database which is the main pillar on which the recognition is based, that is able in some way to improve over time with the newly detected faces can significatively improve the service performances.

Considering a real case scenario to deploy our application, while it is hard to see the system used to identify every kind of face detected, for example a person that passes by in front of the robot without performing any kind of interaction with it. It can instead be used to manage the re-identification constrained to the case of a person that wants to interact with the robot and so in a more accommodating environment for the system to function properly, since the person would be closer and probably looking at it.

In the end, addressing the problem of facial re-identification poses several challenges to the developer. Some of them are well known while others like the growing dataset and its management so as to achieve acceptable performances on the identification at each of its stages, from its creation up to when it is full, are more specific. While we have been able to develop a first version of such re-identification system, we realize that there is a lot of room for improvement to enable the application to operate in a higher number of contexts and improve its performances.

8. Bibliography

[1] Adjabi, I.; Ouahabi, A.; Benzaoui, A.; Taleb-Ahmed, A. Past, Present, and Future of Face Recognition: A Review. Electronics 2020, 9, 1188. https://doi.org/10.3390/electronics9081188

[2] Alexis Lambert, Nahal Norouzi , Gerd Bruder & Gregory Welch (2020): A Systematic Review of Ten Years of Research on Human Interaction with Social Robots, International Journal of Human–Computer Interaction, DOI: 10.1080/10447318.2020.1801172

[3] Umarani Jayaraman, Phalguni Gupta, Sandesh Gupta, Geetika Arora, Kamlesh Tiwari, Recent development in face recognition, Neurocomputing, Volume 408, 2020, Pages 231-245, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.08.110.

[4] Apurva Bedagkar-Gala, Shishir K. Shah, A survey of approaches and trends in person re-identification, Image and Vision Computing, Volume 32, Issue 4, 2014, Pages 270-286, ISSN 0262-8856, https://doi.org/10.1016/j.imavis.2014.02.001.

[5] Di Wu, Si-Jia Zheng, Xiao-Ping Zhang, Chang-An Yuan, Fei Cheng, Yang Zhao, Yong-Jun Lin, Zhong-Qiu Zhao, Yong-Li Jiang, De-Shuang Huang, Deep learning-based methods for person re-identification: A comprehensive review, Neurocomputing, Volume 337, 2019, Pages 354-371, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.01.079.

[6] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao and S. C. H. Hoi, "Deep Learning for Person Re-Identification: A Survey and Outlook," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 6, pp. 2872-2893, 1 June 2022, doi: 10.1109/TPAMI.2021.3054775.

[7] L. Aryananda, "Recognizing and remembering individuals: online and unsupervised face recognition for humanoid robot," IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 1202-1207 vol.2, doi: 10.1109/IRDS.2002.1043897.

[8] Farinella, G.M., Farioli, G., Battiato, S., Leonardi, S., Gallo, G. (2014). Face Re-Identification for Digital Signage Applications. In: Distante, C., Battiato, S., Cavallaro, A. (eds) Video Analytics for Audience Measurement. VAAM 2014. Lecture Notes in Computer Science(), vol 8811. Springer, Cham. https://doi.org/10.1007/978-3-319-12811-5_3

[9] S. K. Teoh, Y. H. Wong, C. F. Leong and L. Y. Tan, "Face Detection and Face Re-identification System Using Deep Learning and OpenVINO," 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS), 2021, pp. 1-5, doi: 10.1109/AiDAS53897.2021.9574201.

[10] A. Dantcheva and J. -L. Dugelay, "Frontal-to-side face re-identification based on hair, skin and clothes patches," 2011 8th IEEE International Conference on

Advanced Video and Signal Based Surveillance (AVSS), 2011, pp. 309-313, doi: 10.1109/AVSS.2011.6027342.

[11] Umarani Jayaraman, Phalguni Gupta, Sandesh Gupta, Geetika Arora, Kamlesh Tiwari, Recent development in face recognition, Neurocomputing, Volume 408, 2020, Pages 231-245, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.08.110.

[12] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, Non-Local Means Denoising, Image Processing On Line, 1 (2011), pp. 208–212. https://doi.org/10.5201/ipol.2011.bcm_nlm

[13] Davis E. King, Max-Margin Object Detection, 2015, https://doi.org/10.48550/arXiv.1502.00046

[14] Dlib, http://dlib.net

[15] Javier Hernandez-Ortega, Javier Galbally, Julian Fierrez, Laurent Beslay, Biometric Quality: Review and Application to Face Recognition with FaceQnet, 2020, https://doi.org/10.48550/arXiv.2006.03298

[16] J. Hernandez-Ortega, J. Galbally, J. Fierrez, R. Haraksim and L. Beslay, "FaceQnet: Quality Assessment for Face Recognition based on Deep Learning," 2019 International Conference on Biometrics (ICB), 2019, pp. 1-8, doi: 10.1109/ICB45273.2019.8987255.

[17] Portrait quality (reference facial images for MRTD), Version: 1.0 ICAO, Published by authority of the Secretary General (2018).

[18] ibug300-W dataset, https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, 2015, https://doi.org/10.48550/arXiv.1512.03385

[20] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.

[21] Wang, Yujiang & Shen, Jie & Petridis, Stavros & Pantic, Maja. (2018). A real-time and unsupervised face Re-Identification system for Human-Robot Interaction. Pattern Recognition Letters. 128. 10.1016/j.patrec.2018.04.009.

Acknowledgement

I would like to thank my supervisor, Professor Emanuele Menegatti, for providing guidance and feedback throughout the writing of the thesis.

Thanks to my tutor Andrea, to Matteo, David, Giacomo, Fabio and to all the staff of Omitech srl where I did my internship, for their mentorship during the development of the project.

Thanks to my grandmother, Maria, and to my aunt, Caterina, for always carrying me in their thoughts and prayers.

Thanks to my friends for giving me moments of lightheartedness in distant lands during these years.

Thanks to my parents, Michela and Mauro, for their constant support and for always believing in me.

In the end my deepest thanks to my beloved sister Arianna, for sharing the worst and best times of this journey and for pushing me when I need it. Reaching the end of this journey without your support would not have been possible.