# Real-Time Multi-Stage Deep Learning Pipeline for Facial Recognition by Service Robots

Nuno Pereira[1], Tiago Ribeiro[2], Gil Lopes[3], A. Fernando Ribeiro[4]

[1] University of Minho, Guimarães, Portugal, a72220@alunos.uminho.pt
[2] University of Minho, Guimarães, Portugal, id9402@alunos.uminho.pt
[3] University Institute of Maia – ISMAI, Maia, Portugal, alopes@ismai.pt
[4] University of Minho, Guimarães, Portugal, fernando@dei.uminho.pt
fernando@dei.uminho.pt

**Abstract.** Recent advances in generic service robots have shown their introduction in various novel environments such as domestic and healthcare facilities. In order to ensure an enhanced interaction between the robot and its users, a multi-stage deep learning pipeline for facial recognition algorithm is used. By detecting which of the pre-trained users the robot is interacting with, it can adapt its actions to best fit the user's needs. The multi-stage system is composed of three modules. An MTCNN to detect all the faces in the image, an Inception-Resnet that generates the feature vectors and provides an amplified network for facial recognition and an SVM classifier to categorize each of the faces recognized to the correct user. The combination of the three modules allows an end-to-end facial detection and recognition that can be used as a real-time identification method. The resulting method was implemented on the general service robot CHARMIE.

**Keywords:** Facial Recognition, Facial Detection, Multi-Network, Deep Learning, Service Robots, Inception-ResNet

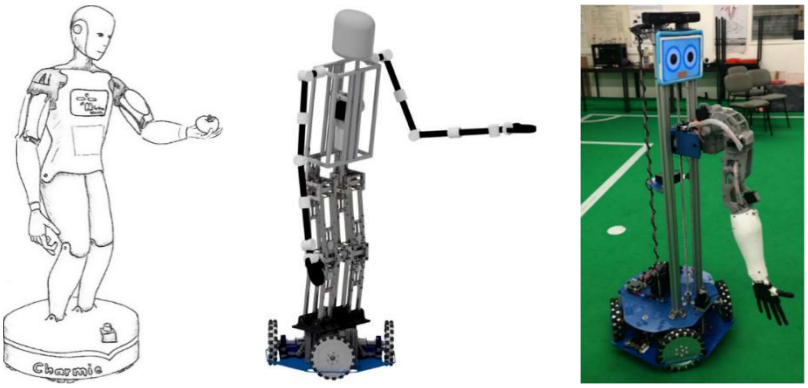
## 1 Introduction

The analysis of information collected from visual perception is a natural human behavior that allows humans to make structured decisions. By endowing robotic systems with the capability of observing different environments and recognizing valuable data, it is possible to create systems that can accurately evaluate and interact with a broad range of scenarios. Facial detection and recognition have been topics whose recent developments made it one of the most used bio-metric techniques for identity authentication.

This paper presents a real-time user-based facial detection and recognition system implemented on CHARMIE (Collaborative Home/Healthcare Assistant Robot by Minho Industrial Electronics) [1] shown in Fig. 1. CHARMIE is an anthropomorphic robot that performs generic service tasks in non-standardized environment settings using machine learning algorithms, which allow the robot to make rational decisions based directly on the surrounding environment.

The focus is to provide healthcare and domestic support in collaborative and cooperative tasks that involve interacting with specific workers/patients/users. In order to benchmark new technologies developed for CHARMIE, service and assistive robotics competitions [2] provide a common framework for a high-rigor benchmark of smart

and autonomous systems. RoboCup@Home is regarded as a top competition in the field of domestic service and assistive robotics [3], [4], where robots must perform a set of benchmark tests to aid in day-to-day realistic non-standardized home environment setting. The features described in this paper demonstrate a real-time user-based face detector and classifier used to create a more efficient recognition system for improved user recognition. It produces a more natural interaction between humans and service and assistive robots, which eases the performance in all tasks that are dependent of user recognition.




(a) (b) (c)

**Fig. 1.** CHARMIE (Collaborative Healthcare/Home Assistant Robot by Minho Industrial Electronics) different variations. (a) Conceptual sketch of the anthropomorphic robot. (b) Developed anthropomorphic design. (c) Primary prototype assembled

The proposed system detects and aligns faces with a deep cascaded multi-task framework, searching for facial landmarks, such as eyes, nose or mouth, predicting its bounding box coordinates, cropping the detected faces containing the least background possible. Next, the bounding boxes are encoded into another neural network that extracts the features, creating a vector representing the face. With that, the classifier predicts the identity.

## 2 Related Work

Facial detection and recognition are an active computer vision field of study that dates more than twenty years in research. Dimensional reduction algorithms were popularized by the Eigenface method [5], where eigenvectors were calculated using principal components analysis (PCA) and then compared to known individuals. This approach proved to be efficient but has low accuracy with different scales of images and orientation invariance of the head and lighting variation problems.

With recent advances in deep learning convolutional networks, DeepFace [6] achieved state-of-the-art accuracy (97.35%) on LFW [7] benchmark, approaching human performance (97.53%) by training four million facial images on the AlexNet [8]

model. After this breakthrough, various proposals achieved state-of-the-art regarding classification. [9] proposes different models categorized by Euclidean-distance-based loss, angular/cosine-margin-based loss and SoftMax Loss and its variations.

The Euclidean-distance-based loss is a metric learning method that embeds images into a Euclidean space that maximizes the inter-variance (between classes) distance and minimizes the intra-variance (within class). Contrastive loss uses face image pairs, pulling positive pairs and pushing negative ones. DeepID3 [10] achieved 99.53% introducing VGGNet [11] and GoogleNet [12] architectures to previous DeepID approaches.

Those methods combined face identification with SoftMax and verification with con-trastive loss. FaceNet [13] introduced the concept of triplet loss, which directly optimizes the embedding itself, rather than using an intermediate bottleneck. Using the Inception architecture [12] provides the capability to concatenate different filter sizes in the same processing layer. Contrary to contrastive loss that considers the absolute distances of the matching pairs and non-matching pairs, triplet loss takes into consideration the relative difference of the distances between them.

Regarding previous RoboCup@Home competitions, popular facial detection techniques deployed include OpenCV, Viola-Jones algorithm and Haar-based algorithms, according to [14] and [15]. However, as stated by [16], those methods proved to still be limited on multiple variations of faces like scale, pose or illumination. More recent approaches include the use of OpenFace, having better performances when finding facial landmarks. [14] also overviews the diverse topologies competitors deployed for facial recognition. Overall, the methods used still have limited deep learning frameworks, due to the need of libraries and cloud services based Deep Neural Networks. Although those present good robustness, cloud services are often unreliable due to connectivity problems so many teams prefer their own offline solutions.

## 3 Methodologies

The developed system is composed of three main modules. Initially, to perform face detection and alignment, MTCNN [17] method is used. The outcome of the first module are all the detected crops of isolated faces in the image. The second module, regarding face recognition, uses an Inception-ResNet architecture that combines both Inception and ResNet algorithms [18] to provide an enhanced network that best suits the facial recognition purpose. The cropped detected faces ultimately serve as inputs to the network to classify the faces, transforming them into an image representation vector. The classification is made with SoftMax Loss function or Support Vector Machine (SVM). Three different datasets were used for training, one custom made with the laboratory researchers working on CHARMIE (named LAR dataset) and two standard datasets: CASIA-Webface [19] and LFW [7].

### 3.1 MTCNN (Multi-task Cascaded Convolutional Networks)

MTCNN [17] proposes a framework to integrate detection and alignment tasks using a 3-stage (P-Net, R-Net and O-Net) unified cascaded CNNs by multi-task learning.

The usage of cascaded networks as a face detector, such as MTCNN allowed an accurate performance of facial bounding box regression as well as keypoint estimation, cropping and aligning the detected face proposals. In addition, it provided a robust system for real world situations, detecting faces in non-standard situations with an acceptable frame rate.

### 3.2 Inception-ResNet

The deep neural network used for faces feature extraction is the Inception-Resnet version 1 [18]. This model results in a combination of two different deep learning methodologies: residual connections introduced by Resnet [20] and the Inception modules introduced with GoogleNet [12].

The Inception architecture uses multiple Inception modules with different convolutional layers with different spatial kernel sizes operating in parallel. These filter the same level layer in the architecture concatenating into the next level, thus, finding various features with fewer convolutional layers. The selected model is a broader and deeper version of [12]. Therefore, the efficiency of the network benefits by replacing the filter concatenation stage of the Inception with residual connections, retaining computer efficiency. That resulted in more straightforward blocks to be used, followed by 1x1 convolution without activations before the residual connections. Those convolutions were implemented in order to compensate the dimensionality reduction induced by the Inception part, given that residual addictions only work if the input and output matched the same depth dimension.

This pipeline was selected considering the tradeoff between performance and computational cost of the overall architecture. Given the robot's resources, it was crucial to have a less computationally expensive framework without substantially sacrificing the accuracy and confidence performances. Inception modules maintained state-of-the-art accuracy with a modest increase of computational cost compared to deeper networks, whereas ResNet's residual blocks allowed a reduced training time.

### 3.3 Training and Validation

To perform the neural network training and fine-tuning, two different datasets were used. Initially the CASIA-Webface [19] dataset was used to train the neural network, containing 494,414 images of 10,575 people. It is then split into three sub-sets: training dataset (80%), validation dataset (10%) and test dataset (10%), and further verified on the LFW [7] dataset benchmark.

Additionally, a customized dataset was created for classification. It contains image data from researchers and professors from the Laboratory of Automation and Robotics (LAR) of Minho University. It is composed of 19 different people, and the image data was fetched using frame samples of multiple videos with different face positions. Moreover, when extracting the images from the videos, crops, horizontal flips and rotations on random frames were performed when creating the LAR dataset. Those data augmentation techniques were employed to simulate an additional pose variation an overall variety of the image data. The dataset had 300 images per identity, 250 images for

classifier training and 50 for validation, over people of ages between 18 and 56 of both female and male genders.

As previously stated, CHARMIE performs generic service tasks in non-standardized environment, focusing in providing healthcare and domestic support, performing collaboratively and cooperatively. When presented with an unknown person, the learning process of a new identity needs to be efficient and fast. Training a convolutional neural network as a multi-class classifier, introducing a new class means an end-to-end retraining and re-evaluation of the network, being an extended and expensive procedure. By using an external SVM classifier, adding more people to the dataset for training and classification proved to be an efficient method with great accuracy, without being computationally and time expensive.

# 4 Results

## 4.1 MTCNN (Multi-task Cascaded Convolutional Networks)

The MTCNN framework was implemented in real-time processing using CHARMIE's camera. It uses an image-pyramid scale factor of 0.709, as well as a [0.6,0.7,0.7] detector threshold array for the bounding box IoU on the 3 stages.

The P-Net flags all face proposals, with different kernel sizes generating numerous bounding boxes. It resizes the original photo to check for various face sizes that may appear. The P-Net final output is shown in Fig. 2. (a). It has already been filtered by non-maximum suppression so it can be fed into the next stage. Next, the R-Net focuses

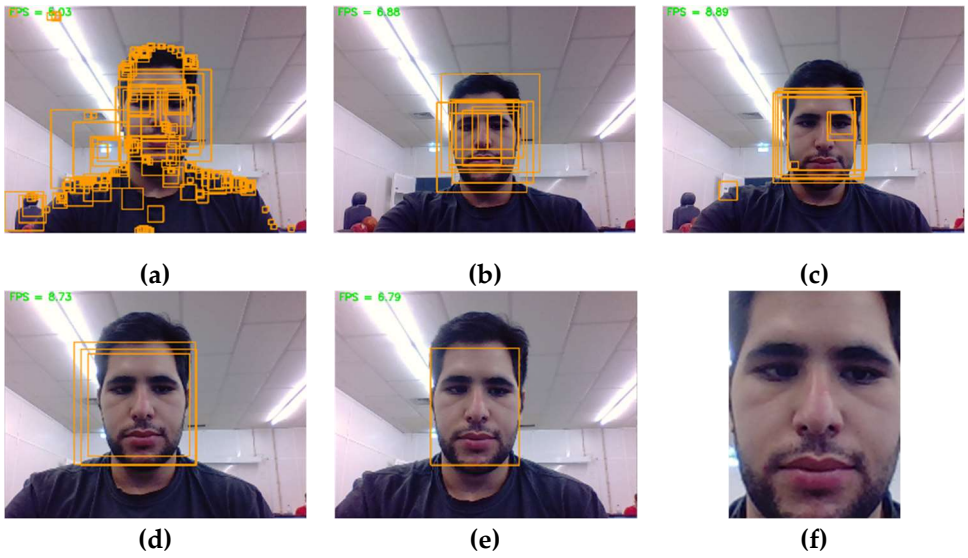



**Fig. 2.** Output bounding boxes of MTCNN framework. (a) P-Net output with NMS. (b) R-Net output. (c) R-Net output after NMS (d) O-Net output (e) O-Net output after NMS (f) Final Cropped face.

on filtering a high number of false positives from the previous network. It uses the initial data to generate more precise bounding boxes, with all the lower confident proposals being eliminated, Fig. 2. (b). By using NMS, the bounding boxes are reduced and padded into squares, merging overlapped proposals, as shown in Fig. 2 (c). Finally, the O-Net standardizes both the bounding box and facial landmarks coordinates, Fig. 2. (d). After the NMS, only the bounding box prediction with the highest confidence level is provided, as shown in Fig. 2. (e). The final output of the MTCNN is the cropped image from the final bounding box prediction represented in Fig. 2. (f). This image is a representation of the data used to train the Inception-ResNet.

### 4.2 Inception-ResNet

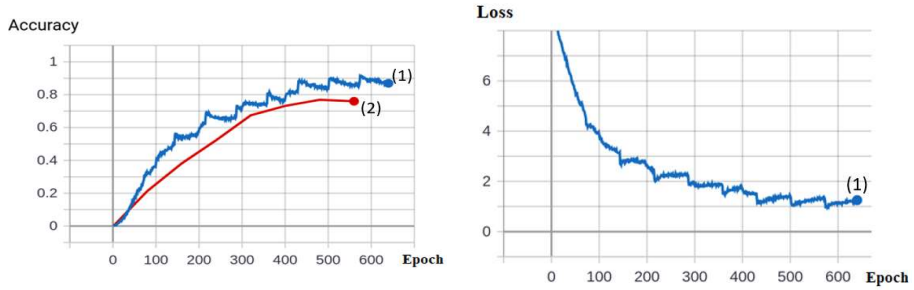
After tackling the face detection problem, the next method deals with the face recognition problem. Inception-Resnet v1 was trained using the CASIA-Webface dataset with two different topologies. After some testing, the best performing network reached 90%, 78% and 77.5% accuracy on the train, validation and test datasets, respectively, after 510 epochs. The hyperparameters of the best performing Inception-Resnet model are shown in Table 1.

**Table 1.** Hyperparameter settings of the Inception-Resnet network.

| Batch Size | Number of epochs | Initial lr | Lr Decay | Dropout | Train Accuracy | Validation accuracy | Train time |
|---|---|---|---|---|---|---|---|
| 5000 | 510 | 0.06 | 0.005* | 20% | 90% | 78% | 14 hours |

* for every 100 epochs

Additionally, regulator parameters were employed to prevent overfitting. The L2 weight decay and L1 norm loss activation were set to 0.0005. The learning rate started at 0.06, decaying until 0.03. The network's input had a 50 batch size in 100 mini-batches per epoch, making it a total of 5000 images per batch. The model's convergence over time can be seen Fig. 3.



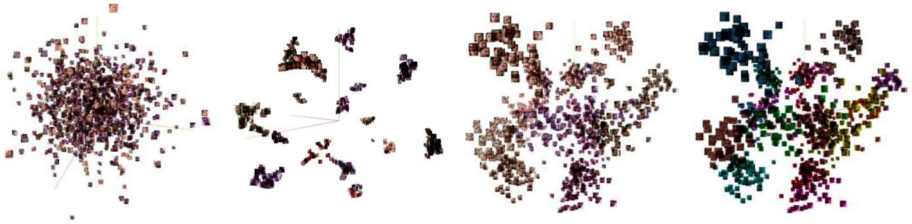

**(a)** **(b)**

**Fig. 3.** Best performing model's convergence over time during 510 epochs. (a) Accuracy. (1). Train dataset. (2). Cross-Validation dataset. (b) Loss. (1). Train dataset.

### 4.3 SVM classifier (Support Vector Machine)

Using the TensorFlow framework. It uses the feature vectors extracted from the Inception-Resnet v1 network to associate the detected faces with trained ones.

The created dataset images were inputted into the classifier. 250 training images for each of the 19 subjects were used, and 50 test images for validation. The SVM classifier correctly predicted all of the test images, where the confusion matrix displays a clear certainty on the predicted labels and ground truths. The feature vectors extracted were plotted into TensorBoard's embedding projector using the PCA and T-SNE (Fig. 4).



(**a**) (**b**) (c) (d)

**Fig. 4**. T-distributed stochastic neighbour embedding (a) 1st interaction - joint probability distribution not calculated. (b) 1000th interaction with joint probabilistic distribution. PCA – Principal Component Analysis with three first PCs (c) non labelled embedding visualizer (d) labelled embeddings.

The final result, shown in Fig. 5 (a) and (b), already uses the robot's camera to detect different pre-trained users. It shows online detection of three different users detected with significant accuracy percentage plus one non trained user, shown as "unknown". It uses the multi-stage pipeline-built inference. MTCNN for face detection and bounding box generation where the faces detected are encoded into the Inception-ResNet that generates the feature vectors. Lastly, the feature vectors are classified by the trained SVM where if a certain threshold prediction value is reached, it plots the predicted person information.


(**a**) (**b**)

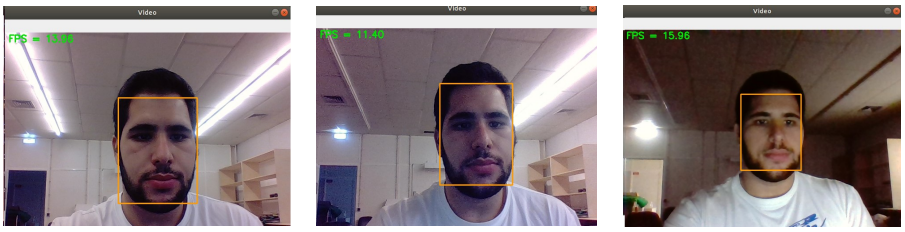
**Fig. 5.** Real-time user-based facial detection and recognition (**a**) Two different users detected and classified (**b**) Two different users detected, on the left, the user is part of the pre-trained users and is classified correctly, on the right, the user is not part of the pre-trained users, and so is labelled as 'Unknown'.

## 5 Discussion

The facial detector MTCNN is a compelling framework that is able to detect faces of various sizes in different pose scenarios. However, the concatenation of three different neural networks working at a range of 8 to 14 fps is computationally expensive and thus require high computational power. The most impactful configuration hyperparameter of the MTCNN is the scalar factor that defines the size range of searches and the bounding box thresholds. The scalar factor that yielded the more refined results surrounds the 0.7 value, where any value between 0.5 and 0.75 allowed the method to converge. Also, the detector threshold array for the bounding box IoU on the 3 stages was set at [0.6,0.7,0.7]. For more challenging proposals, lowering the values in the array would have better detection, but the appearance of false positives was more frequent. Fig. 6 shows an image from RoboParty 2019 with a high density of people with faraway faces, that come up as very small. The methodology presented detects almost every face with few exceptions where people are facing other directions. However, as Fig. 7 shows that, faces closer to the robot can successfully be detected even when looking away from the camera. A real concern over the detection pipeline was how different lighting settings would affect its performance. Hence, Fig. 8 presents an overview of different light conditions tests. As one can see, the MTCNN correctly predicts the user's bounding boxes.


**Fig. 6.** MTCNN output bounding boxes in a highly populated scenario. (Image from RoboParty 2019 Robotics Educational Event)

**Fig. 7.** MTCNN detecting user with different face poses, facing different directions.




**(a)** **(b)** **(c)**

**Fig. 8.** MTCNN bounding box detection with different light conditions.

Regarding the face recognition network, the Inception-ResNet v1 proved to have high accuracy but challenging to optimize, given the number of gradients and parameters presented in its architecture. The ideal batch size rounded between 4000 and 5000 images per batch, divided into mini-batches. The batches were divided into a set of mini-batches, for two reasons: computational resources and mini-batch gradient descent. Since the computational resources were limited, the GPU had difficulty allocating enough memory to train big batches of data. Mini-batch gradient descent reduced the variance of the gradients when calculating the error and updating the coefficients, which turned out to be very important for convergence. The maximum number of images the computer managed to process in a batch was 6000. However, when compared to smaller batches, like the standard 5000-image batch, proven efficient, the accuracy performance would slightly increase, but producing a less rewarding time-consuming framework.

The ideal learning rate to initiate the training is between 0.03 and 0.06 (Fig. 9. **(**a) and (b**))**. These allowed the model to convergence whereas bigger values could not do so. Smaller values would work but would produce worse performance results. Learning rate decay was essential for the model's execution (Fig. 9. **(**c) and (d**))**. By slowly decaying the learn metric, the model was slowly adjusting by its patterns , whereas with a stable lr the Inception-ResNet topologies would have a very difficult time generalizing the learning process. Overfitting of the model turned out to not be a significant issue over the large dataset. By tuning down the dropout percentage, the model would have a lower accuracy and validation(Fig. 9.(e and f)). As previously stated, the best performing Inception ResNet v1 model reached almost 90% on training accuracy and 77.5% on validation accuracy after 500 epochs (Fig. 3). Since the dataset produced such a challenging classification framework, reaching that accuracy performance was ensuring that the model could work as a feature extractor for the final pipeline.

The classifiers training from feature vectors using a linear kernel support vector machine proved to be a very efficient solution since service robots must add and remove users from their memory with a considerably high frequency while also providing consistent classification results.

In real-time, previously implemented modules could successfully work in conjunction with new data instances (Fig. 5). Using the pre-trained members of LAR as test labels, the system correctly detect faces and predicted the user's identities on different configurations with very high accuracy and confidence. The model reached a maximum of 12 frames per second when only one person was detected but dropped to 6 or 7 frames per second when more people started to appear on the image. Additionally, in

real-time, some face poses such as looking deeply left or right would show a lower accuracy that could successfully be tackled with a more diverse custom dataset.

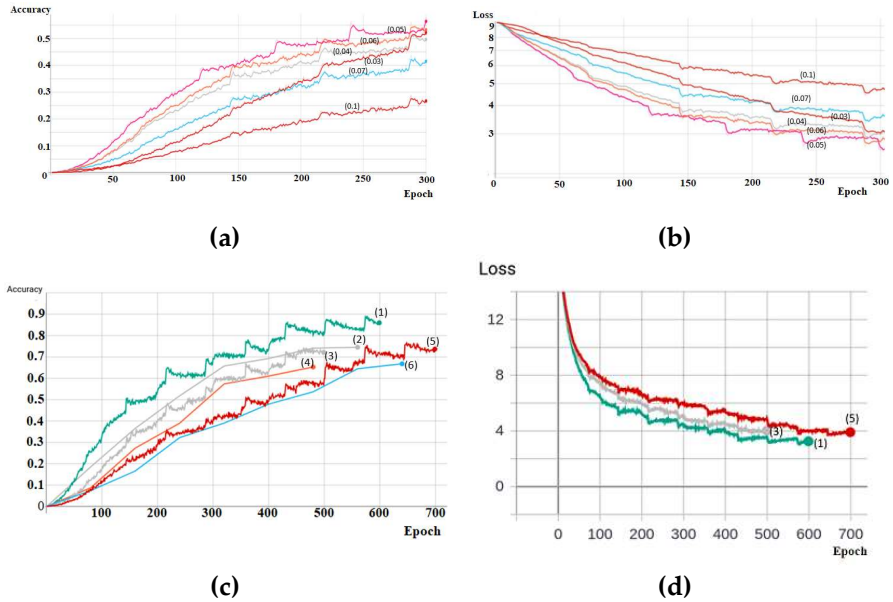



**Fig. 9.** Inception-Resnet hyperparameter tunning research using the CASIA-Webface dataset. (a)- Train accuracy with different Learning rates on the same topology. (b) Loss values of the different Learning Rates. (c) Dropout percentages convergence on the same topology. (1) 40% Dropout train dataset accuracy. (2) 40% Dropout validation dataset accuracy. (3) 70% Dropout train dataset accuracy. (4) 70% Dropout validation dataset accuracy. (5) 80% Dropout train da-taset accuracy. (6) 80% Dropout validation dataset accuracy. (d) Loss values of the train dataset of different Dropout percentages. (1) 40% Dropout train dataset accuracy. (3) 70% Dropout train dataset accuracy.(5) 80% Dropout train dataset accuracy.

## 6 Conclusion

A real-time user-based face recognition system using multi-stage deep learning methods is proposed for service robots domain, with the capability to detect and identify people. A custom dataset of users from the Laboratory of Automation and Robotics (LAR) from the University of Minho was trained for facial detection and recognition implementation on a domestic and healthcare service robot, CHARMIE. A multi-task cascaded framework, MTCNN, performed the detection and alignment module that isolated the detected faces. These serve as input to an extraction model, the Inception-Resnet model, that creates a feature vector of each face. Next, an SVM classifier is trained with the laboratory member and predicts their identities. All the networks end-to-end proved to be able to work on real-time applications and managed to detect users

with different face poses and external variations such as illumination. The Inception-ResNet model can differentiate classes into Euclidean space, as seen by the PCA and t-SNE plots, extracting the features of the faces. Furthermore, the classifier using SVM is able to predict new instances of the labels. Overall, the system was able to detect and predict every student with an above 90% accuracy.

For a service and assistive robot such as CHARMIE, face detection and user recognition are essential for a more user-oriented interaction. Therefore, CHARMIE can adapt its approach as well as how it performs different tasks depending on who the interacting user is. Additional information regarding its users can be associated with a specific user, for example, if it is a child, an adult or a senior, whether that person has any mobility issues or if it is allowed or not to be in a specific area of an environment. All this information helps create a more personalized experience for all user interactions since CHARMIE can use this technology to directly adapt its behaviours to positively influence whom it is interacting with.

This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020 and funded through a doctoral scholarship from the Portuguese Foundation for Science and Technology (Fundação para a Ciência e a Tecnologia) [grant number SFRH/BD/06944/2020], with funds from the Portuguese Ministry of Science, Technology and Higher Education and the European Social Fund through the Programa Operacional do Capital Humano (POCH).

## References

[1] R. A. Ribeiro T, Gonçalves F, Garcia IS, Lopes G, "CHARMIE: A Collaborative Healthcare and Home Service and Assistant Robot for Elderly Care," *Appl. Sci.*, vol. 11, no. 16, p. 7248, 2021.

[2] M. Basiri, E. Piazza, M. Matteucci, and P. Lima, "Benchmarking Functionalities of Domestic Service Robots Through Scientific Competitions," *KI - Kunstl. Intelligenz*, 2019.

[3] D. Holz and L. Iocchi, "Benchmarking Intelligent Service Robots through Scientific Competitions : The RoboCup @ Home Approach," in *AAAI Spring Symposium - Designing Intelligent Robots: Reintegrating AI II*, 2013, pp. 27–32.

[4] L. Iocchi, D. Holz, J. Ruiz-Del-Solar, K. Sugiura, and T. Van Der Zant, "RoboCup@Home: Analysis and results of evolving competitions for domestic and service robots," *Artif. Intell.*, 2015.

[5] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, 1991.

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

[7] G. B. Huang, M. Mattar, T. Berg, E. L. Labeled, R. Images, and E. Learned-miller, "Labeled Faces in the Wild : A Database for Studying Face Recognition in Unconstrained Environments," *Work. faces in'Real-Life'Images Detect. alignment,*

*Recognit.*, vol. 07–49, 2008.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, 2017.

[9] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, 2021.

[10] Y. Sun, D. Liang, X. Wang, and X. Tang, "DeepID3: Face Recognition with Very Deep Neural Networks," pp. 2–6, 2015.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[12] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[14] M. Matamoros, V. Seib, and D. Paulus, "Trends, Challenges and Adopted Strategies in RoboCup@Home," in *19th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2019*, 2019.

[15] M. Matamoros, V. Seib, R. Memmesheimer, and D. Paulus, "RoboCup@Home: Summarizing achievements in over eleven years of competition," *18th IEEE Int. Conf. Auton. Robot Syst. Compet. ICARSC 2018*, pp. 186–191, 2018.

[16] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, "Deep Face Recognition: A Survey," *Proc. - 31st Conf. Graph. Patterns Images, SIBGRAPI 2018*, pp. 471–478, 2019.

[17] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Process. Lett.*, 2016.

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.

[19] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning Face Representation from Scratch," 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.