Online Recognition-by-Tracking with Deep Appearance and Facial Features in a Robotic Environment

Rui G. Figueiredo, Rui P. Figueiredo, Alexandre Bernardino, José Santos-Victor
Instituto Superior Tecnico, Lisboa

***Abstract*—In the past few years, the number of robots being deployed in society has been continuously increasing. Robots are coming to family houses as personal assistants in domestic tasks and entertainers (e.g. toys) as well as in elderly care, handicap assistance, and nursing centers. The new generation of service robots have now to interact with humans in uncertain environments. For this, the robot needs to localize, engage and identify the target subject. The identification of the target could be done in different ways. Image-based face recognition is one example. It is a well-studied problem and state-of-the-art solutions achieve remarkable performance. However, most of the proposed solutions are not adapted to the robot environment. In this work, we explore a new approach to the problem of online person recognition. We present the Recognition-by-Tracking framework that uses pedestrian tracking in order to accumulate evidence about the face identities what leads to more accurate predictions.**

I. INTRODUCTION

In the past few years, the number of robots being deployed in society has been continuously increasing. Robots have now many uses outside of industrial environments [8], [10] where they were extensively and successfully used in the past decades. Robots are coming to family houses [15] as personal assistants in domestic tasks [10] and entertainers (e.g. toys) as well as in elderly care [10], handicap assistance, and nursing centers. The new generation of service robots have now to interact with humans in complex and uncertain environments. For this, the robot needs to localise, engage and identify the target subject. One example is *Peper*[1], a robot designed to interact with people and understand their emotions. It is able to identify if the target person is joyful, sad, angry or surprised. If the robot was able to recognize the person, the task could be improved. The robot could access to data collected in previous interactions and would make the iteration more personalized. Like in human-human interaction, the more you know about the person, the better it is the interaction and the engagement.

The literature on person identification is vast, however, most solutions are unnatural and intrusive. One example is the identification by fingerprint [27], but it requires collaboration by the user and contact with the sensor. In a robot such as *Peper*, this kind of recognition will be not natural. The robot will not be able to interact with a group unless it has more than one fingerprint sensor. Another type of solutions is vision-based. Where physical contact is not needed. In particular, facial recognition has achieved accuracy close to the human levels [23]. Face-recognition technologies are being deployed in many *smartphones*[2] as an approach for authentication. In the human-robot interaction context the problem is not so much explored, the existing solutions are not adapted to this environment. In the robot environment person recognition is a dynamic online task.

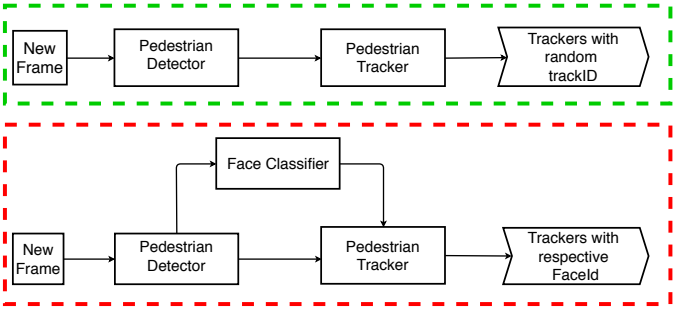

Figure 1. Standard architecture for Tracking-By-Detection (Green), New Architecture for Recognition-by-Tracking (Red)

The main objective of this work is to develop a system to provide the robot the ability to recognize people using RGB cameras as the main sensor. The usual approach to facial recognition problems is to run the recognition method in each frame [12], [17], that we call by Recognition-by-Detection. From frame to frame, the information is lost. We want the robot to be able to keep the identity of the persons over the time, even if the face becomes occluded. Pedestrian Tracking methods have succeeded in the task of keeping the same identity for each person along the time [29]. However, the identity given to each tracker is a randomly generated number that is not related to the real identity of the person. So we present a new architecture that changes the traditional Tracking-by-Detection framework [1], presented on green in Figure 1, replacing the randomly generated id for each person with their face identity. The new architecture is presented in red in Figure 1. We provide an implementation of our method that can be used in any robot that uses the Robot Operating System (ROS).

For the proposed system, we will need to be able to detect, track and recognize pedestrians. In section II we will revise the works in each one of this areas. In section III we present the method that we develop for Recognition-by-Tracking. In section IV we present the setup for our experiments. Section V presents the results. Section VI concludes the paper and present the future work.

[1]https://www.softbankrobotics.com/emea/en/robots/pepper/find-out-more-about-pepper

[2]https://www.apple.com/ca/business-docs/FaceID_Security_Guide.pdf

II. Related Work

A. Pedestrian Detection / Multi-Person Key-Point Detection

Pedestrian detection is a fundamental task in any intelligent video surveillance system. It is the task of detecting persons in an image. Nowadays, it is a well-studied problem with satisfactory results. Typically the approach for the traditional pedestrian detector methods is to extract some features from the image and build a classifier. Features like the Haar-like features [28] or the histogram of oriented gradients, HOG [6], a descriptor based on the counts of the occurrences of the gradient orientation in localized portions of an image. The method described in [7] combines HOG features with Aggregated Channel Features (ACF) that consist in the concatenation of the normalized gradient magnitude and the histogram of the orientation for the LUV color channels.

Multi-Person key-point detection is the task of detecting the position of the interesting body key-points (e.g Nose, Right Shoulder) for all persons in the image. The state-of-the-art is Open Pose [5]. The solution is based on a multistage CNN architecture. It receives a 2D image from the scene and outputs N confidence maps for the 2D locations of the N anatomical key-points, and a set of M part affinity maps, that represents the way as the key-points are connected. Afterward, it parses the output maps in order to associate the different key-points and construct the skeleton for each person in the image.

B. Face Recognition

Image-based face recognition is a well-studied problem and state-of-the-art solutions achieve remarkable performance [18], [22]–[24]. For example, [23] achieves 99.63% accuracy in one of the most well-known data-sets [11]. The standard approach for the problem is to find a transformation of the face image in a space with lower dimension than the image space. Usually, the obtained representation is called an embedding. After we obtain the embedding it becomes a standard machine learning problem where we have a set of vector samples (the set of embeddings) a set of labels (the person's names).

In the beginning, the methods to compute embeddings were frequently based on the principal component analysis (PCA) [13]. The most famous is called Eigen Faces [26]. Given a data set of images the principal direction along these images is computed. A base is formed with the k principal directions, the directions with more variance. The face embedding is built by the projection of the images in this base.

Other approaches were based on the extraction of human engineered features, such as SIFT [19] from some face regions typically the left eye region, right eye region, and nose-mouth region [9]. The concatenation of these extracted features generates the embedding.

In last years, the increase of computation power led to solutions based on Deep Convolution Neural Networks [18], [22]–[24]. Deep learning based methodologies differ in the architecture of the network, however, the biggest differences are verified in the way that the errors are calculated during the training. Typically the last layer of this networks is a fully connected layer and the activation function in this layer is the soft-max. The error function is typically cross-entropy loss function (1), where y_i is the label for the sample i and the $\hat{y_i}$ is the prediction . In the work presented in [23] this layer is not used and the error in each iteration is directly calculated in the embeddings. The error is computed by an equation called *triple loss* (2), $f(x)$ is the calculated embedding for a given image region x, ${x_i}^a$ is the anchor image, if there are N different images in the data-set there will have N different anchors, ${x_i}^p$ is the farthest image of the image ${x_i}^a$ within the images of the same class as ${x_i}^a$, ${x_i}^n$ is the closest image within the images that do not belong to the same class of ${x_i}^a$, α is the margin we want to enforce between positive samples and negative samples. The idea of using this function is to ensure that all the positive samples are closer than the nearest negative sample. With this, we ensure that the classes are separable.

The previously discussed methods don't explore the aspect of online recognition. Some works have already tried to do online recognition with the accumulation of information by tracking [3], [16], [21]. However, they only do face tracking, so they cannot keep the identity of the person if the person turns around and the face becomes not visible.

$$-\sum_i \hat{y_i} \log(y_i) \tag{1}$$

$$\sum_i^N [||f({x_i}^a) - f({x_i}^p)||_2^2 - ||f({x_i}^a) - f({x_i}^n)||_2^2 + \alpha]_+ \tag{2}$$

C. Pedestrian Tracking

Several pedestrian tracking algorithms exist. Due to recent advances in object detection techniques and the increase in computational resources, Tracking-by-Detection [1], has become increasingly popular. It is an appearance-based method. It uses as input the image regions that look like people provided by a pedestrian detection. The pipeline is simple. When a pedestrian is detected, it needs to be associated to its corresponding tracker. If the detection is new on the scene, we need to create a new tracker. In order to find the correct associations a similarity metric is defined.

The works in this area differ in the tracker representation. Typically a tracker is represented by a Kalman state [14] of constant velocity, and a set of embeddings describing the appearance of its bounding box. In the work presented in [4], the tracker is represented only by its state of constant velocity. The similarity metric is intersection-over-union (IOU) between the detected bounding boxes and the predicted bounding boxes. The predicted bounding boxes are estimated with a Kalman Filter [14] recurring on the defined model of constant velocity.

The work presented in [29], Deep Sort, extends [4] by adding a set of K embeddings describing the bounding box appearance. These embeddings are extracted with a Neural Network that contains 2,800,864 parameters. A forward pass of 32 bounding boxes takes approximately 30 ms on a Nvidia GeForce GTX 1050 mobile GPU. This neural network was

Table I
2D KEY-POINTS REPORTED BY OPEN POSE [5]

Id	Name	Id	Name
0	Nose	10	Right Ankle
1	Neck	9	Right Knee
2	Right Shoulder	11	Left Hip
3	Right Elbow	12	Left Knee
4	Right Wrist	13	Left Ankle
5	Left Shoulder	14	Right Eye
6	Left Elbow	15	Left Eye
7	Left Wrist	16	Right Ear
8	Right Hip	17	Left Ear

trained on a large-scale person re-identification dataset [30] that contains over 1,100,000 images of 1,261 pedestrians. They improve the similarity metric by adding a term that considers the body appearance.

III. METHOD

As we saw in Figure 1 our method adds to the traditional architecture of Tracking-by-Detection the identity of each person tracked. Recognition-by-Tracking is a new approach to online recognition problem, we collect several samples from the identity of each person before taking a decision. We are able to keep the real identity along the time.

The method comprises three stages. In the first stage, *Detection Stage*, we take the image as input and we output the bounding box for the face, if visible, and for the body. In second stage, *Feature Extraction and Recognition Stage*, we take as input the bounding boxes. From the body bounding box, we extract the embedding that describe this region. From the face bounding box we extract the probability for each class. In third stage, *Tracking & Identity Set Stage*, we manage the trackers and their identity.

A. Detection Stage

For the proposed system, for each person in the image we will need one bounding box for the face and one bounding box for the body. The steps for this stage are presented in Figure 2. It could be achieved by merging the detections from face and pedestrian detectors. This method is not so efficient, we need to run two detectors for each frame, we need to find a metric in order to establish the match between faces and bodies. Instead, we used joint face and body detector method Open Pose [5]. It achieves a remarkable performance on body key-point detection. It is very robust at illumination changes, occlusions, and poses. Open Pose takes as input an RGB image and outputs the 2D location for the body key-points presented in Table I. All the points are in form $p_{i,j} = (x_{i,j}, y_{i,j})$, person i, part j, in pixels. For example $p_{i,0}$ represents the position of the nose for the person i. Let S_F be the set of point indices belonging to the face, $S_F = \{0, 1, 14, 15, 16, 17\}$ and S_B be the set of point indices belonging to the body, $S_B = \{0, 1, 2, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$. We exclude the points from the arms in order to prevent larger bounding boxes for the case when the person has the arms open.

Figure 2. Diagram of Detection Stage, 1 - Original Image, 2- Key-points detected by OpenPose, 3- Face Bounding Box (Blue), Body Bounding Box (Red)

Let us define the face bounding box for the i person in the image as $B_i^F = (x_i^F, y_i^F, w_i^F, h_i^F)$, where (x_i^F, y_i^F) represent the top left corner of the bounding box and w_i^F,h_i^F represent the width and the height of the bounding box, respectively. In the same way let us define the body bounding box for the person i in the image as $B_i^B = (x_i^B, y_i^B, w_i^B, h_i^B)$.

The body bounding box for the person i, B_i^B, is compute following (3). We simply take the max and min for x and y within the $(x_{i,j}, y_{i,j})$ pairs where $j \in S_B$.

$$\begin{cases} x_i^B = \min\limits_{j \in S_B} x_{i,j} \\ y_i^B = \min\limits_{j \in S_B} y_{i,j} \\ w_i^B = \max\limits_{j \in S_B} x_{i,j} - x_i^B \\ h_i^B = \max\limits_{j \in S_B} y_{i,j} - y_i^B \end{cases} \tag{3}$$

The face bounding box is computed following (4). We take a reference for the position from the nose location. For the size we set up a proportion, between the distance between the eyes and the face size. So, we suppose a linear model $y = \rho x$ where x is the distance between the eyes and y is the bounding box size. In order to find the parameter ρ we have collected a set of (x_i,y_i) pairs from [2], a data set that provide the annotation for the facial key-points for 202,599 faces. With this set of pairs, using least squares method, we obtain $\rho = 3,73$ with an coefficient of determination, $R^2 = 0,96$.

$$\begin{cases} w_i^F = (x_{i,14} - x_{i,15}) \times \rho \\ h_i^F = (x_{i,14} - x_{i,15}) \times \rho \\ x_i^F = \max(0, x_{i,0} - \frac{w_i^F}{2}) \\ y_i^F = \max(0, x_{i,0} - \frac{h_i^F}{2}) \end{cases} \tag{4}$$

B. Feature Extraction and Recognition Stage

In this stage, from the body bounding box we extract one embedding that describes the the body, E_i^B, that will be used on the tracker. From the face bounding box, we extract one descriptor for the face, E_i^F, that will be used in the face classifier. The steps for this stage are presented in Figure 3.

In order to compute E_B, we have used the Deep Sort Network described in [29]. The network takes as input an image, $I_i^B \in \mathbb{R}^{128 \times 64 \times 3}$, where the first two dimensions correspond to the height and the width of the image, and the last dimension to the number of color channels (RGB). The network outputs an embedding $E_i^B \in \mathbb{R}^{128}$.

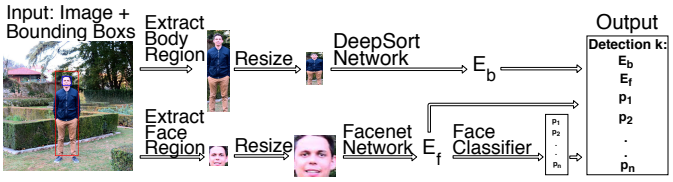

Figure 3. Diagram of Feature Extraction and Recognition Stage.

As the bounding boxes do not always have the same size, we need to resize them using a bi-linear interpolation in order to fit in the input of the network.

For extraction of E_i^F, we use the Inception network [23]. The architecture that we are using takes as input an image $I_i^F \in \mathbb{R}^{168\times168\times3}$. It produces as output an embedding $E_i^F \in \mathbb{R}^{128}$.

After we get an embedding representing the face, the problem can be solved with the standard Machine Learning Algorithms. We use a Multilayer Percepton (soft-max layer). The face classifier is trained using images collected offline. The classifier outputs a set of probabilities $P_i = (p_{i,1}, p_{i,2}..., p_{i,K})$, where K is the number of persons registered in the system.

C. Tracking and Identity Set Stage

The structure of our tracker is the same as the one described in [29]. Each person tracked is described by a state of constant velocity and a set of K embeddings describing the bounding box appearance. In order to set an identity for each tracker, we follow a Bayesian inference approach. Each detection comes with a classification y provided by the face classifier. This classification follows a categorical distribution, $y \sim Cat(p_1, ..., p_K)$. We use as conjugate prior the Dirichlet distribution $p_1, ..., p_K \sim Dir(\alpha_1, ..., \alpha_K)$, where K is the number of classes. For each tracker, we initialize the distribution with all $\alpha_i = 1$. Each time that there is a new association between one tracker and one detection, if the face is visible, we need to update the value for all α_i for the respective tracker. The update is made following (5). The derivation of (5) could be found in [25].

In order to set an identity for each tracker, the system uses the set of α_i and the rules showed in Figure 4. The system only provides predictions about the identity of the person after getting N samples of the person face. We set a parameter θ (probability threshold) in order to distinguish unknown persons from known persons.

$$\alpha'_i = \alpha_i + p_i \tag{5}$$

IV. Experimental Setup

In order to evaluate our method, we set up two experiments using the robot Vizzy [20]. Vizzy is a Humanoid on Wheels for Assistive Robotics. In the first experiment, Vizzy as Coach, we simulated the task where the robot is stopped and a person comes to play with him. The persons start at 3,5 meters away from Vizzy, walks in his direction and stops 1 meter away

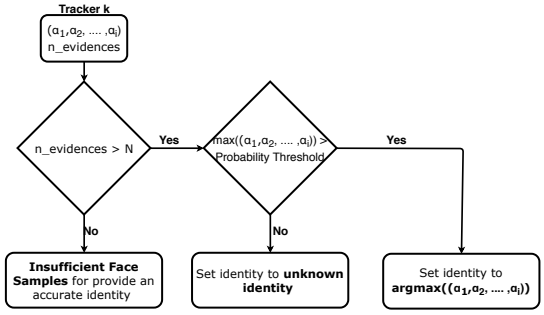

Figure 4. Flowchart of the set identity process for each Tracker k.

from. In the second experiment, we simulated the task of Vizzy walking and crossing with people in the corridor. We ran the two experiments with 20 people where 2 of them are unknown for the robot, i.e, they are not used to train the classifier. For these experiments, we have used the camera located at the chest of the robot, Logitech HD Pro C920 1080p. We have recorded 40 videos, one for each person for each experiment. In order to train the softmax classifier, we collected 10 images per know person, in total 180 images. These images are collected offline.

The evaluation is done measuring the accuracy in the recognition between the frames N and $N + P$ according to Figure 5. We have compared the accuracy between the labels given by the face classifier in each detection, following Recognition-by-Detection approach, with the accuracy in the labels given by the process Recognition-by-Tracking. The label given for Recognition-by-Detection is $r = argmax((p_1, p_2, ..., p_K))$, where $(p_1, ..., p_K)$ is the set of probabilities reported by the face classifier in each frame. The person is classified as unknown person if $p_r < \theta$, θ is the probability threshold that we had defined in order to distinguish between known and unknown persons.

We have set the parameter P to 40, i.e, we evaluate the performance in 40 frames. This was the max value that we could choose for P, since the videos have a limited duration and we want to test different values for the parameter N and remain the same conditions for all videos, evaluate in the same number of frames. We have tested the influence of the parameter N and the parameter θ.

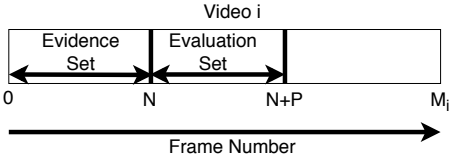

Figure 5. Diagram of the set for evidence and for evaluation. N is the parameter of the Recognition-by-Tracking Framework, the number of evidence frames, M_i is the number of frames in video i, P is the number of frames used to measure the accuracy.

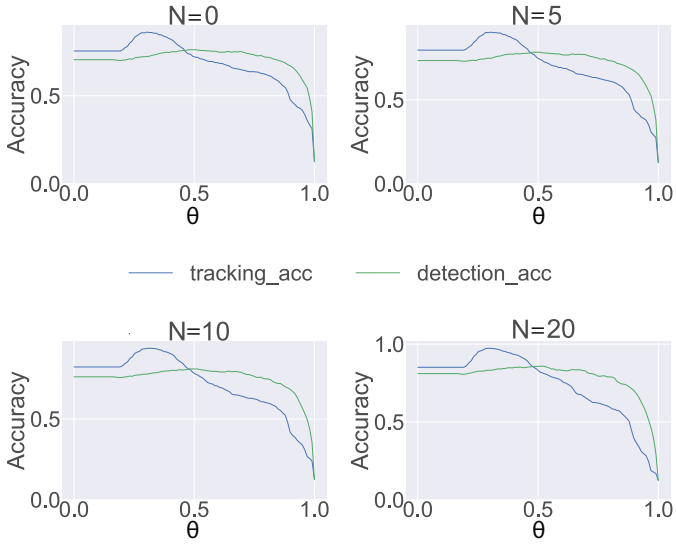

Figure 6. Accuracy vs θ for different values of N for the experiment 1 using the Softmax Classifier.

V. Results

A. Experiment 1 - Vizzy as Coach

In table II we present the results obtained for different values of θ. We set the N parameter to 20. We choose 20 since we are acquiring and processing frames at 20Hz, found an identity to a new person takes one second, a reasonable value.

Table II
Mean accuracy for the experiment 1, setting N equal to 20.

θ	Recognition-by-Detection	Recognition-by-Tracking
0	0.65	0.90
0.25	0.67	0.95
0.5	0.72	**0.98**
0.75	0.69	0.74

From Table II, we could observe that at the best configuration we obtain a mean accuracy 98%. In Figure 6 we show the accuracy for different values of the parameters N and θ. In Figure 7 we present the α values along the time.

In Figure 6 we could notice the influence of the parameter N. For higher values of the parameter, the max of the accuracy is higher. We could observe that the dependency on θ is smaller for higher values of N, if we observe the plot for $N = 20$ in Figure 6 we could see that the accuracy does not change for $\theta \in [0.3, 0.5]$

We could observe in Figure 7 the filtering effect provided by the Bayesian Inference approach, the values for the probabilities on the tracking approach are much more smooth than the ones observed on detection approach. We could observe that the α values on the tracker with the increase of the number of evidence (number of frames) trends to go to lower values in the case of an unknown person and to higher values in the case of know persons. This is the reason why we chose to add the parameter N to our framework.

We notice that the accuracy following the approach of Recognition-by-Tracking is significantly better than the accuracy on framework Recognition-by-Detection. When θ goes to higher values, the performance of Recognition-by-Detection becomes better than the performance of our framework. It could be explained observing Figure 7. The probability values on the Recognition-by-Tracking framework stabilizes around some value, typically smaller than one. For the probability on the Recognition-by-Detection, we have peaks going to a higher value than the values where the probability of the tracker stabilizes. If we observe the example for person 8 in Figure 7 we could see that setting $\theta = 0.75$, the tracking mechanism will be always wrong. The detection mechanism will be right in some frames, in the peaks. But we don't need a significantly higher value for θ, we just need to choose one value that prevents to classify unknown persons as known persons. In Figure 6, we could notice the values of θ at which the unknown persons begin to be classified correctly. The accuracy starts to decrease when we are classifying known persons as unknown. In the plot for $N = 20$ in Figure 7 we could notice that the mean accuracy starts to increase at $\theta \approx 0.2$ and starts to decrease at thresholds $\theta \approx 0.5$. So if, we choose to set up $N = 20$ we need to choose a threshold in this interval.

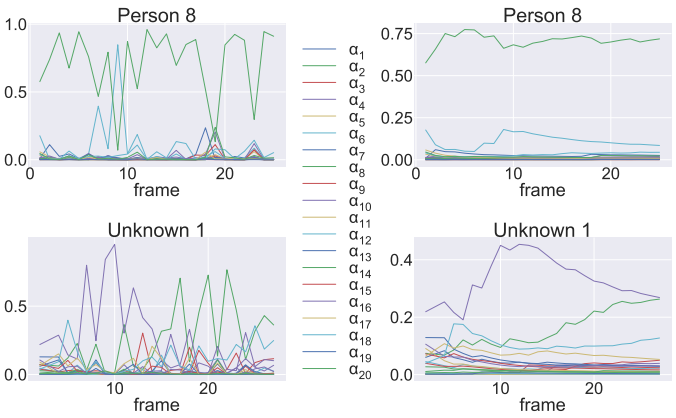

Figure 7. α values for Categorical Distribution (left) and normalized α values for Dirichlet Distribution (right) in the experiment 1.

B. Experiment 2 - Vizzy Walking

The results obtained for the second experiment are reported in Table III. We follow the same approach, setting N to 20. In Figure 8 we present the accuracy for different values of θ and N. The results for the second experiment are worse than the ones obtained for the first one. In the first experiment the pose of the person is always frontal to the robot. When the robot is moving it is not true, so it degrades the observations that the robot get from the person face.

VI. Conclusions and Future Work

We could demonstrate in this work that the approach of Recognition-by-Tracking improves the accuracy on Face Recognition when compared with the traditional approach. For the first experiment, where the robot is stopped, we archive

Table III
MEAN ACCURACY FOR THE EXPERIMENT 2, SETTING N TO 20.

θ	Recognition-by-Detection	Recognition-by-Tracking
0	0.37	0.79
0.25	0.38	**0.80**
0.5	0.40	0.68
0.75	0.42	0.42

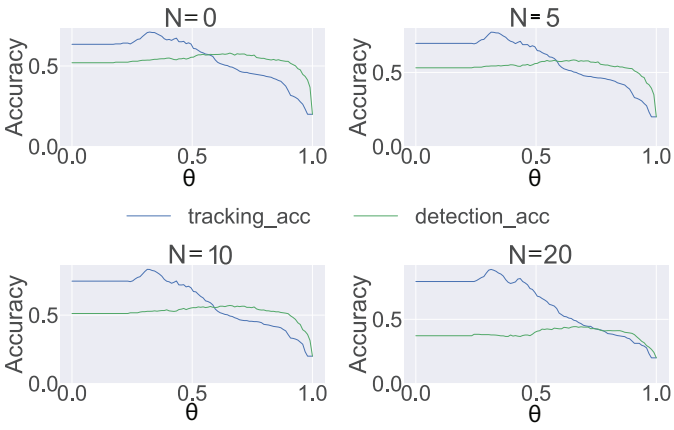

Figure 8. Accuracy vs θ for different values of N for the experiment 2

accuracy near to 98 % at the best configuration. For the second experiment, the accuracy drops to 80 %. The method fails mostly because of the position of the face relative to the robot. In the future we will control the position and the pose of the robot relative to the person, using Active Vision Mechanisms.

ACKNOWLEDGEMENTS

This work was partially supported by FCT projects UID/EEA/50009/2019 and AHA – CMUP-ERI/HCI/0046/2013.

REFERENCES

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[2] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *CoRR*, abs/1611.01484, 2016.

[3] M. Baykara and R. Daş. Real time face recognition and tracking system. In *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, pages 159–163, Nov 2013.

[4] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple Online and Realtime Tracking. *ArXiv e-prints*, Feb. 2016.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[7] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545, Aug 2014.

[8] D. Fischinger, P. Einramhof, K. Papoutsakis, W. Wohlkinger, P. Mayer, P. Panek, S. Hofmann, T. Koertner, A. Weiss, A. Argyros, and M. Vincze. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems*, 75(Part A):60 – 78, 2016. Assistance and Service Robotics in a Human Environment.

[9] C. Geng and X. Jiang. SIFT features for face recognition. *Proceedings - 2009 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009*, pages 598–602, 2009.

[10] B. Graf, U. Reiser, M. Hägele, K. Mauz, and P. Klein. Robotic home assistant care-o-bot - product vision and innovation platform. In *2009 IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 139–144, Nov 2009.

[11] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts Amherst Technical Report*, 1:07–49, 2007.

[12] W. Jiang and W. Wang. Face detection and recognition for home service robots with end-to-end deep neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2232–2236, March 2017.

[13] I. T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. Springer New York, New York, NY, 1986.

[14] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[15] C. D. Kidd and C. Breazeal. Robots at home: Understanding long-term human-robot interaction. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3230–3235, Sept 2008.

[16] K. Koide, E. Menegatti, M. Carraro, M. Munaro, and J. Miura. People tracking and re-identification by face recognition for rgb-d camera networks. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–7, Sept 2017.

[17] T. Linder, S. Wehner, and K. O. Arras. Real-time full-body human gender recognition in (rgb)-d data. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3039–3045, May 2015.

[18] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[19] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, Sept 1999.

[20] P. Moreno, R. Nunes, R. Figueiredo, R. Ferreira, A. Bernardino, J. Santos-Victor, R. Beira, L. Vargas, D. Aragão, and M. Aragão. *Vizzy: A Humanoid on Wheels for Assistive Robotics*, pages 17–28. Springer International Publishing, Cham, 2016.

[21] B. M. Nair, J. Foytik, R. Tompkins, Y. Diskin, T. Aspiras, and V. Asari. Multi-pose face recognition and tracking system. *Procedia Computer Science*, 6:381 – 386, 2011. Complex adaptive sysytems.

[22] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An All-In-One Convolutional Neural Network for Face Analysis. *Proceedings - 12th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2017 - 1st International Workshop on Adaptive Shot Learning for Gesture Understanding and Production, ASL4GUP 2017, Biometrics in the Wild, Bwild 2017, Heteroge*, pages 17–24, 2017.

[23] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:815–823, 2015.

[24] Y. Taigman, M. Yang, and M. Ranzato. Deepface: Closing the gap to humal-level performance in face verification. *CVPR IEEE Conference*, pages 1701–1708, 2014.

[25] S. Tu. The dirichlet-multinomial and dirichlet-categorical models for bayesian inference. *Computer Science Division, UC Berkeley*, 2014.

[26] M. Turk and A. Pentland. Face recognition using eigenfaces. *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[27] T. van der Putte and J. Keuning. Biometrical fingerprint recognition: Don't get your fingers burned. In *Proceedings of the Fourth Working Conference on Smart Card Research and Advanced Applications on Smart Card Research and Advanced Applications*, pages 289–303, Norwell, MA, USA, 2001. Kluwer Academic Publishers.

[28] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–511–I–518, 2001.

[29] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[30] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*, pages 868–884. Springer, 2016.