2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Deep Active Robotic Perception for Improving Face Recognition Under Occlusions

Valia Dimaridou, Nikolaos Passalis and Anastasios Tefas
Computational Intelligence and Deep Learning Group, AIIA Lab
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece
{dimaridou, passalis, tefas}@csd.auth.gr

***Abstract*—Recent studies have demonstrated that active perception can improve the perception abilities of deep learning (DL) models. However, there are challenges associated with using active perception in DL models, including the need for datasets and/or realistic simulations that can support the training process, along with the difficulty of predicting the final target position, which reduces planning efficiency. To address these challenges, this work presents a methodology for enhancing the perception abilities of DL models through active perception. The methodology proposes a way to create datasets for active perception by fusing existing large-scale datasets and decomposing the active perception problem into three sub-tasks for face recognition. The sub-tasks aim to determine the appropriateness of the current view for face recognition, the direction in which the robot should move for a better viewpoint, and the expected amount of movement required. A novel trial-based approach is introduced to estimate the final target position, making the method platform-agnostic and easily applicable to different robots. The proposed methodology is validated through experiments on two well-known face verification datasets that have been augmented with occlusions, demonstrating its effectiveness in enhancing the perception abilities of DL models through active perception.**

***Index Terms*—active perception, face recognition, robotics perception**

I. Introduction

In recent years, Deep Learning (DL) has revolutionized various complex robotics vision tasks, ranging from object detection and recognition, scene segmentation, face recognition, and more [1]. This enhanced perceptual ability has also opened up various opportunities, leading to the creation of impressive applications such as autonomous cars, autonomous drones, and robots that can collaborate with humans on various tasks. Despite these breakthroughs in DL for robotics, most existing methods have a major drawback - they rely on a static inference paradigm that is inherent in traditional computer vision pipelines. This means that DL models process fixed and static inputs, ignoring the fact that robots have the ability to interact with their environment to gather more information. For example, consider a robot equipped with a camera that performs face recognition [2]. A DL model may not be able to recognize a person if the robot has only captured a suboptimal view, e.g., a far profile view. However, by repositioning itself, a robot can acquire a better view, e.g., a frontal and closer view. Then, the same DL model could recognize the person. This approach, known as active perception [3]–[5], involves manipulating the robot or sensor to acquire a clearer view or signal, leading to improved situational awareness. This process is similar to how humans and animals interact with their environment, such as humans looking from different angles to process complex visual stimuli, or animals pointing their ears toward the source of an audio signal [6].

Recent, yet rudimentary, studies have shown that active perception can enhance the perception abilities of various models. For instance, a deep learning system that predicts the optimal next move for a robot using reinforcement learning, as demonstrated in [7], can significantly enhance the performance of object detection where factors such as viewing angle, occlusions, and object scale can greatly affect recognition accuracy. Similar findings have also been reported in more recent works [8]–[10]. It is worth noting that active perception approaches often result in the creation of faster and lighter DL models, as they are trained to solve a simplified problem, e.g., face recognition from specific viewpoints [11].

Even though active perception can indeed lead to improved perception capabilities in such cases, it comes with a challenge: it requires a significant change in training pipelines for DL models. Indeed, active perception-enabled DL models are usually trained either using datasets that can support active perception, e.g., datasets that can allow for simulating robot movements [11], or in simulation [12]. However, the former approaches require the creation of appropriate datasets, which can be costly, while the latter suffers from well-known problems, such as distribution shifts [13], requiring the use of appropriate sim-to-real methodologies before deployment. Furthermore, most of the existing approaches only try to solve a *direction prediction problem*, i.e., predict the direction towards which a robot should move in order to acquire a better viewpoint. However, this a) requires the constant involvement of active perception in order to refine the movement and decide when it should stop, as well as b) makes planning much harder, since there is no estimate of the final target.

The main contribution of this work is a methodology that can tackle these two challenges. To this end, we first propose a methodology that allows for the creation of datasets that can be used for active perception approaches by fusing existing large-scale datasets in an appropriate way. This enables reusing large-scale datasets with minimal cost, providing a practical way to train active perception-enabled DL models. Furthermore, we propose decomposing the problem of active perception for face recognition into three appropriate subtasks qualitatively answering the following questions: a) Is the current view appropriate for face recognition?, b) If not, towards which direction the robot should move?, and c) After

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

deciding the direction, how much the robot it expected to move?. Note that answering the last question is not trivial, since it required robot-specific knowledge (e.g., the parameters of the camera, require the potential use of depth sensors, etc). To overcome this limitation, in this work we propose a novel trial-based approach, which enables estimating the final target position after performing a small trial movement. This enables the proposed method to be platform-agnostic and be very easily applied to different robots with no modifications. The proposed method is validated using two well-known face verification datasets, demonstrating its effectiveness.

The rest of the paper is structured as follows. First, Section II introduces the proposed method. Then, Section III provides the experimental evaluation. Finally, Section IV concludes the paper.

II. Proposed Method

Background Let $\mathbf{x}_t$ denote an observation acquired from a robot at timestep t. In this work, we will focus on face recognition tasks and we assume that this observation has been appropriately pre-processed, e.g., cropped using face detection models. Also, let $f_f(\mathbf{x}_t) \in \mathbb{R}^D$ be a face recognition model that extracts a discriminative representation that can be used for face verification or retrieval [14], [15], where D is the dimensionality of the extracted representation. Typically, we require this representation to bring images that belong to the same person closer, i.e.,

$$||f_f(\mathbf{x}_t) - f_f(\mathbf{x})||_2 < r \ \forall \mathbf{x} \in \mathcal{X}_k, \tag{1}$$

where $\mathcal{X}_k$ denotes the set of images of a person k, $||\cdot||_2$ denotes the l_2 norm, and r is a face recognition threshold, i.e., the maximum distance between two face representations in order to consider that these belong to the same person. Similarly, for images that belong to different persons it holds:

$$||f_f(\mathbf{x}_t) - f_f(\mathbf{x})||_2 \geq r \ \forall \mathbf{x} \in \bar{\mathcal{X}_k}, \tag{2}$$

where $\bar{\mathcal{X}_k}$ denotes the set of images that do not belong to the person k. The aim of this work is to learn how a robot should manipulate the camera sensor in order to improve its confidence, i.e., reduce the distance for representations that belong to the same persons:

$$||f_f(\mathbf{x}_{t+T}) - f_f(\mathbf{x}_R)||_2 < ||f_f(\mathbf{x}_t) - f_f(\mathbf{x}_R)||_2, \tag{3}$$

where $\mathbf{x}_{t+T}$ denotes the face representation obtained after T timesteps (i.e., movements of the robot) and $\mathbf{x}_R \in \mathcal{X}_k$. Note that it is often enough to improve the confidence just for one image of a person, since $\mathcal{X}_R$ may contain different views of the same person. Therefore, it is enough to reduce the distance with respect to just one of these images in order to correctly recognize the person.

Active Perception Networks Active perception approaches aim to appropriately control a robot/robot's sensors in order to obtain such a better representation. In this work, we decompose the employed active perception pipeline into three discrete steps. First, we need to decide whether the current input $\mathbf{x}_t$ is good enough or whether we need to actually perform any active perception step in order to improve (note that typically active perception is used to just improve perception, it is not a goal *per se*). Then, we need to decide the direction of the movement. In this work, we only consider two movement axes, i.e., up/down and left/right. However, this is without loss of generality, since the same approach can be applied to any number of axes. Finally, we need to decide the amount of movement that needs to be performed. Of course, complex occlusions might require a combination of such movements that need to be applied in sequence in order to get the best result. In such cases, the robot should perform them sequentially.

In this work, we propose to modify the existing face recognition approaches by adding an additional classification head on top of the representation extracted by existing face representation backbones. This enables us to quickly make a decision concerning whether the current face is occluded. Note that the first step of the aforementioned pipeline is always executed, while the other two are executed only when we have decided that we need to perform active perception during the first step. Therefore, it is critical that a lightweight approach is used for making this decision to avoid slowing down the face recognition pipeline when faces that are good enough are obtained. This decision can be made by a simple linear classifier on top of the extracted representation as:

$$\mathbf{y}_a = \mathbf{W}_a^T f_f(\mathbf{x}_t) \in \mathbb{R}, \tag{4}$$

where $\mathbf{W}_a$ are the weights of the employed binary classifier. Experimentally, we have found out it is possible to merge the first two steps using the same classifier, which can further accelerate the inference process. Therefore, we propose employing a *Active Perception Direction Classifier* (APDC), which is formulated as:

$$\mathbf{y}_{APDC} = \mathbf{W}_{APDC}^T f_f(\mathbf{x}_t) \in \mathbb{R}^5, \tag{5}$$

where each of the output neurons of the classifier corresponds to the following actions: a) do nothing (active perception is not needed), b) move to the left, c) move to the right), d) move up and e) move down. This classifier is trained using the regular cross entropy loss and the ground truth annotation generated along with the dataset. The process that generates the ground truth annotations will be explained in *Dataset Generation* subsection.

The third part of the proposed active perception pipeline is to predict the amount of movement that needs to be performed in order to obtain the best possible view. This allows for performing optimal planning instead of incrementally repeating the active perception steps, while it also enables us to perform more fine-grained control. To this end, after deciding the direction of movement using the APDC, the robot performs a trial movement (of predefined duration), and a new image, $\mathbf{x}_{t+1}$ is acquired. Then, another network, the *Movement Regressor* (MR), is to estimate how many times this movement should be repeated in order to get the best possible view. This network received both the original and the new images and outputs a number that expresses the ratio between the current movement and the optimal estimated movement:

$$\mathbf{y}_R = f_{MR}(\mathbf{x}_t, \mathbf{x}_{t+1}) \in \mathbb{R}. \tag{6}$$

Note that we do not directly use the representations extracted from $(\mathbf{x}_t)$ to this end since such features are invariant to small perpetuations due to the way face recognizers are trained.

Therefore, we opted for using a separate network for this end. Even though this comes with an additional cost, it should be noted that this network is only used when the APDC decides that a movement should be performed. Similar to APDC, this network is trained using ground truth annotations obtained through a simple simulation environment described in *Dataset Generation* subsection.

Dataset Generation The employed data and ground truth annotation pipeline employ two components: a *2D Occlusion Generator* (2DOG) and b) a *3D simulator* (3DS) module. The 2DOG picks a facial image (from a face recognition dataset) and a segmented object (from a dataset that contains objects and segmentation masks) and generates an occluded facial image. Note that, as described in Section III, there are many large-scale datasets that can be directly used to this end. After selecting a random occlusion percentage, a resize percentage for the selected object, and an occlusion direction, a synthetic image is generated by superimposing the selected object and facial image. Since we know the occlusion direction, the ground truth annotation for the APDC is readily available. Then, during the training, the 3DOG module accepts an occluded image produced by 3DOG and the direction produced by the APDC and produces the new observation $\mathbf{x}_{t+1}$ (using a randomly selected distance). The same module then calculates the remaining movement needed for complete object removal, as a ratio of the movement impact. This ratio is the target for the regression model optimization process. Note that 3DOG is used both during the training (in order to generate the new observation images), as well as during the validation (to simulate the effect of active perception).

III. Experimental Evaluation

Evaluation Datasets and Experimental Setup As we described in Section II, the proposed method requires two different datasets for training, i.e., a face dataset and a generic object dataset. To this end, in this work, we employed the MS-Celeb dataset [16] as the face dataset, which is one of the largest publicly available databases used for face recognition and verification. It consists of a total of 10M images, from 1M individuals. The individuals depicted in this database are people that have received public attention, mostly due to their profession. The dataset includes diversity both in terms of age and race. The subset that is used in the current pipeline is constructed by selecting the top 100K celebrities, based on the frequency of their appearance. The final training dataset consists of 5,084,127 unique images.

We also employed the Common Objects in Context [17] dataset as the source for objects that can be used for generating occlusions. COCO dataset includes highly accurate annotations for instance segmentation tasks, which makes it ideal for object extraction. More specifically, we use the instance segmentation masks to accurately crop all the annotated objects included in each sample image. We save the cropped objects as RGBA samples, in order to seamlessly apply the occlusions on face images. We later apply 3 post-processing steps to further improve the quality and realism of the occlusions: (1) we remove images that their axis aligned bounding box is smaller than 50x50 pixels; (2) we apply contour detection and exclude the images that contain more than one blob (this happens due to object split, caused by occlusion with a different instance); (3) we further crop the objects so that the object coverage in the RGBA image is higher than 70%. We specifically perform the last two steps, because we want the object to occlude as much as possible and not leave face features visible in the occlusion areas. Our final object dataset includes 129K and 65K objects split into training and validation sets respectively.

Finally, for evaluating the performance of the proposed method we used two well-known face verification datasets: a) the Labeled Faces in the Wild dataset (LFW) [18], which contains 13,233 images, collected from the web, as well as b) the Cross-Pose Labeled Faces in the Wild (CPLFW) dataset, which contains 11,652 images.

Model Architecture The employed face recognition model is composed of a ResNet-50 backbone, while the ArcFace loss is used for learning face discriminative representations [15], combined with focal loss [19]. The employed model is trained on MS-Celeb-1M dataset [20]. Note that during the training of the active perception components of the proposed method, we do not back-propagate gradients to the backbone in order to avoid altering the representations learned for face recognition, which would require re-training the face recognition head. The direction decision network is implemented as an extra classification head that predicts direction data parallel to the classification head. It operates on the $512 \times 7 \times 7$ feature maps produced by the face recognition model, using two linear layers, where the last linear module outputs five probabilities, one for each possible direction. For the MR we use an EfficientNet B7 backbone [21], followed by a global pooling layer and two fully connected layers with 2048 and 1 neuron(s) respectively. The swish activation function is used for the first layer, while the sigmoid activation function is used for the last layer in order to constrain the output to values between 0 and 1. Both of the variations use sigmoid activation so that we can limit the robot's movement to non-extreme values.

We use a two-step pipeline to train the APDC and MR. The first part of the training pipeline includes the training of the APDC. We load pre-trained weights both for the face feature extractor and face recognition head and keep them frozen in the optimization step. The second part of the training pipeline includes the training of MR. The previously trained (from step 1) APDC remains frozen in this step, and EfficientNet B7 backbone is being optimized. We initialize the parameters of the backbone with ImageNet pre-trained weights and normalize the input images accordingly. For all the experiments reported in this paper, the models are trained for 250K iterations. The Adam optimizer is used [22] with a learning rate of $1e{-3}$.

Experimental Evaluation First, we evaluate the baseline face verification accuracy under no occlusions, along with the effect of different occlusion ratios in Table I. The CPLFW dataset is harder than the LFW dataset leading to a baseline verification accuracy of about 92% instead of about 99%. Applying increasingly larger amounts of occlusions further reduces these figures. Indeed, for 60% occlusion, the accuracy drops to below 65% for the CPLFW dataset and to less than 85% for the LFW dataset, demonstrating the significant

TABLE I: Face verification accuracy when occlusion objects are applied to face images.

Dataset	CPLFW	LFW
Baseline	92.23%	99.81%
30% occlusion	90.28%	99.49%
40% occlusion	85.64%	97.68%
50% occlusion	72.46%	89.75%
60% occlusion	63.50%	81.68%

TABLE II: Effect of employing active perception on face verification accuracy. Using the proposed method allows for achieving higher accuracy in a smaller number of steps.

Dataset	CPLFW	LFW
Baseline	81.61%	95.04%
APDC (1 step)	87.43%	98.23%
APDC (2 steps)	89.93%	99.41%
APDC (3 steps)	90.91%	99.59%
Proposed (APDC+MR)	92.05%	99.8%

impact of occlusions on face recognition accuracy.

Then, in Table II we report the results of applying the proposed method on a dataset that contains occluded images (equally distributed occlusions from 20% to 60%). First, note that similarly to the previous results, when no active perception is employed, the accuracy drops significantly. Employing the APDC network allows for improving the obtained results, but even after three steps, we haven't yet reached the original accuracy. Note that the use of APDC only is equivalent to the method proposed in [23]. On the other hand, when the full proposed pipeline is employed in just two steps (trial + active perception) we are able to get almost to the point of the original images that contained no occlusions.

IV. Conclusion

This paper presented a methodology for enhancing the perception abilities of DL models through active perception. The methodology addressed the challenges associated with using active perception in DL models, including the need for specific datasets or simulations, and the difficulty of predicting the final target position. Experimental results on two well-known face verification datasets demonstrated the effectiveness of the proposed methodology in enhancing the perception abilities of DL models through active perception. The results showed that the proposed method was able to provide improved recognition accuracy compared to traditional passive perception approaches. These findings suggest that the proposed methodology has the potential to be widely adopted in real-world applications where active perception can enhance the perception abilities of DL models, providing a practical and effective solution. Future research direction include the application of this methodology for other tasks, e.g,. object detection, as well as the evaluation of the developed models using real scenarios, e.g., in healthcare robotics [24].

Acknowledgment

This work was supported by the European Union's Horizon 2020 Research and Innovation Program (OpenDR) under Grant 871449. This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. European Conf. on Computer Vision*, pp. 499–515, 2016.

[3] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[4] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, pp. 177–196, 2018.

[5] N. Passalis, S. Pedrazzi, R. Babuska, W. Burgard, D. Dias, F. Ferro, M. Gabbouj, O. Green, A. Iosifidis, E. Kayacan, *et al.*, "Opendr: An open toolkit for enabling high performance, low footprint deep learning for robotics," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 12479–12484, 2022.

[6] R. S. Heffner and H. E. Heffner, "Evolution of sound localization in mammals," *The evolutionary biology of hearing*, pp. 691–715, 1992.

[7] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 1378–1385, 2017.

[8] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 9068–9079, 2018.

[9] X. Han, H. Liu, F. Sun, and X. Zhang, "Active object detection with multistep action prediction using deep q-network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3723–3731, 2019.

[10] S. K. Ramakrishnan and K. Grauman, "Sidekick policy learning for active visual exploration," in *Proc. European Conf. on Computer Vision*, pp. 413–430, 2018.

[11] N. Passalis and A. Tefas, "Leveraging active perception for improving embedding-based deep face recognition," in *Proc. IEEE 22nd Intl. Workshop on Multimedia Signal Processing*, pp. 1–6, 2020.

[12] P. Tosidis, N. Passalis, and A. Tefas, "Active vision control policies for face recognition using deep reinforcement learning," in *Proc. European Signal Processing Conf. (EUSIPCO)*, pp. 1087–1091, 2022.

[13] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 3803–3810, 2018.

[14] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2018.

[15] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, June 2019.

[16] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European Conf. on computer vision*, pp. 87–102, 2016.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. European Conf. on Computer Vision*, pp. 740–755, Springer, 2014.

[18] 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," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Intl. Conf. on computer vision*, pp. 2980–2988, 2017.

[20] Q. Wang, P. Zhang, H. Xiong, and J. Zhao, "Face. evolve: A high-performance face recognition library," *arXiv preprint arXiv:2107.08621*, 2021.

[21] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Intl. Conf. on Machine Learning*, pp. 6105–6114, 2019.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] N. Passalis, A. Tefas, and I. Pitas, "Efficient camera control using 2d visual information for unmanned aerial vehicle-based cinematography," in *Proc. IEEE Intl. Symposium on Circuits and Systems*, pp. 1–5, 2018.

[24] L. D. Riek, "Healthcare robotics," *Communications of the ACM*, vol. 60, no. 11, pp. 68–78, 2017.