Leveraging Active Perception for Improving Embedding-based Deep Face Recognition

Nikolaos Passalis, Anastasios Tefas

Leveraging Active Perception for Improving Embedding-based Deep Face Recognition

Nikolaos Passalis and Anastasios Tefas
Artificial Intelligence Information Analysis Laboratory
Department of Informatics, Faculty of Sciences
Aristotle University of Thessaloniki, Thessaloniki, Greece
E-mail: {passalis, tefas}@csd.auth.gr

***Abstract*—Even though recent advances in deep learning (DL) led to tremendous improvements for various computer and robotic vision tasks, existing DL approaches suffer from a significant limitation: they typically ignore that robots and cyber-physical systems are capable of interacting with the environment in order to better sense their surroundings. In this work we argue that perceiving the world through physical interaction, i.e., employing active perception, allows for both increasing the accuracy of DL models, as well as for deploying smaller and faster models. To this end, we propose an active perception-based face recognition approach, which is capable of simultaneously extracting discriminative embeddings, as well as predicting in which direction the robot must move in order to get a more discriminative view. To the best of our knowledge, we provide the first embedding-based active perception method for deep face recognition. As we experimentally demonstrate, the proposed method can indeed lead to significant improvements, increasing the face recognition accuracy up to 9%, as well as allowing for using overall smaller and faster models, reducing the number of parameters by over one order of magnitude.**

I. Introduction

Deep Learning (DL) has led to tremendous improvements in recent years for various challenging computer vision tasks [1], including, but not limited to, object detection and recognition [2], scene segmentation [3], face recognition [4], and others. The advanced perception capabilities enabled by DL also provided powerful tools for various robotics tasks, leading to the development of spectacular applications, such as autonomous cars [5], drones [6], [7], and robots that can seamlessly interact with humans, e.g., collaborative manufacturing [8].

However, despite these recent achievements of DL in these areas, most of the existing methods suffer from a significant drawback: they follow a static inference paradigm, as inherited by the traditional computer vision pipeline. More specifically, DL models perform inference on a fixed and static input, ignoring that robots, as well as many cyber-physical systems [9], [10], have the ability of *interacting* with the environment in order to better sense their surroundings. For example, consider the task of face recognition, where a robot has acquired a sub-optimal profile view of a subject. An existing static perception-based DL model might fail to recognize the subject from this view, especially if it has never been trained on profile face images. However, it is usually possible that the robot can acquire a better and more discriminative view by more appropriately repositioning itself with respect to the human subject. Therefore, in this case, the exact same DL model, will probably be able to recognize the subject, after the robot repositions itself in a more appropriate angle with respect to the subject. This approach, which is called *active perception* [11], [12], [13], allows for manipulating the robot/sensor in order to acquire a better and more clean view/signal, leading to improved situational awareness. It is worth noting that this process is very similar to the way humans and various animals interact and understand their environment. For example, humans tend to look from different angles when trying to process complex visual stimuli, while many mammals have specialized muscles that rotate their ears toward the source of an audio signal [14].

A number of recent, yet quite primitive approaches, demonstrated that active perception can indeed increase the perception capabilities of various models. For example, in [15] it is demonstrated that developing a deep learning system that also predicts the next best move for a robot, using reinforcement learning, can significantly improve the performance of object detection, where the viewing angle, occlusions and the scale of each object can have a significant effect on the object recognition accuracy. Similar observations were also reported by more recent works [16], [17], [18]. At the same time, it is worth noting that active perception approaches often allow for developing faster and more lightweight DL models, since models are trained in order to solve a simpler problem. For example, in the case of object detection [15], a simpler model can be trained just for recognizing the objects from a limited number of angles, since a robot can usually acquire a more appropriate view that allows for accurately recognizing the corresponding object. To the best of our knowledge, despite these encouraging results in these areas, there have not been any thoroughly study on developing deep learning-based active perception models for human-centric robot perception, such face recognition.

Motivated by the aforementioned observations, we examine two main hypotheses in this work. First, we argue that the recognition accuracy of DL models can be improved by

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.

acquiring a more appropriate view, after manipulating the position of a robot inside the world. For example, moving a robot closer to a human is expected to improve the confidence of the robot when recognizing a human, as well as reduce the recognition errors. At the same time, we hypothesize that a significant part of the complexity in modern DL models arise from their ability to perform view invariant inference. Therefore, we argue that significantly smaller models can be used when active perception approaches are employed, without reducing recognition accuracy.

The main contribution of this work is proposing a DL-based active perception method for embedding-based face recognition, as well as examining the behavior of such approach on a real multi-view face image dataset, shedding light on the aforementioned research questions. The proposed method is capable of simultaneously learning discriminative embeddings, that can disentangle the representations extracted from facial images that belong to different persons, as well as learning which should be the next control action by a robot carrying a camera in order to improve the face recognition confidence, as shown in Fig. 1. The proposed method does not rely on prior knowledge, such as that frontal views might lead to better recognition accuracy, and it is capable of autonomously learning how to acquire the best view in order to facilitate the task at hand. Therefore, the proposed formulation is generic and task agnostic and can be applied to virtually any DL-based recognition model, given that the appropriate simulation environment have been developed and/or the appropriate dataset have been collected.

Furthermore, the proposed method is computationally efficient, since it utilizes the same main network backbone both for extracting a discriminative embedding, as well as for predicting the next action that must be performed in order to increase the recongition confidence, as shown in Fig. 1. To this end, two different branches, an embedding branch and a control action branch, are employed, as further described in Section II. Additionally, instead of using a computationally intensive reinforcement learning-based approach for the optimization, similar to other active perception methods [15], [17], the proposed method employs a purely supervised learning approach, which can significantly accelerate the convergence of the method. It is worth noting that the proposed method is task agnostic and can be trivially implemented in the typical batch-based setting used for training DL models. Therefore, it can be directly used for most classification/regression tasks, extending beyond the face analysis applications presented in this paper, paving the way for providing generic active perception-enable DL models. Finally, to the best of our knowledge, this is the first deep learning-based active perception approach that allows for efficiently optimizing one unified architecture towards both learning discriminative embeddings, as well as performing control.

The rest of the paper is structured as follows. First, the proposed method is introduced and analytically derived in Section II, while an extensive experimental evaluation is provided in Section III. Finally, conclusions are drawn and further research directions are discussed in Section IV.

II. Proposed Method

The proposed method is presented in this Section. First, the necessary notation and a brief introduction to representation learning for face recognition is provided. Next, the proposed approach is presented and discussed in detail.

A. Notation and Representation Learning

Let $\mathbf{x}_i \in \mathbb{R}^{W \times H \times C}$ denote a (cropped) face image, where W, H and C are the width, height and number of channels of the corresponding image. Also, let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots \mathbf{x}_N\}$ be a collection of N training images, while the binary variable $d_{ij} \in \{0, 1\}$ is introduced to denote whether the i-th face image belongs to the same person as the one depicted in the j-th face image. Most recent deep face recognition methods, e.g., [19], aim at learning an appropriate model $\mathbf{y} = f_{\theta_r}(\mathbf{x})$ that will extract a discriminative identify-oriented representation from each face image by solving the following optimization problem:

$$\theta_r = \arg\min_{\theta} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \mathcal{L}(f_\theta(\mathbf{x}_i), f_\theta(\mathbf{x}_j), d_{ij}) \tag{1}$$

Different loss functions $\mathcal{L}(\cdot)$ have been proposed to this end. In this work, we employ the *contrastive* loss [20], [21], which is minimized when embeddings that belong to the same identity are as close as possible, while the representations of face images that do not belong to the same person maintain at least a distance of $\sqrt{m}$:

$$\begin{aligned} \mathcal{L}_c(\mathbf{y}_i, \mathbf{y}_j, d_{ij}) = d_{ij} ||\mathbf{y}_i - \mathbf{y}_j||_2^2 + \\ (1 - d_{ij}) \max(0, m - ||\mathbf{y}_i - \mathbf{y}_j||_2^2), \end{aligned} \tag{2}$$

where $||\cdot||_2$ refers to the l^2 norm of a vector. After training the model $\mathbf{y} = f_{\theta_r}(\mathbf{x})$, the identity of a person depicted in a previously unseen image $\mathbf{x}$ can be obtained simply by performing nearest neighbor search on a database that contains images $\mathbf{x}_i$ of known identities, i.e., $\mathcal{X}_d = \{(\mathbf{x}_i, l_i)\}$, where l_i is the identity of the person depicted in the i-th image. Therefore, during inference the identity l of a person appearing in a novel image $\mathbf{x}$ is obtained as:

$$l = l_i, \text{where} \;\; i = \arg\min_i ||f(\mathbf{x}_i) - f(\mathbf{x})||_2 \;\; (\forall (\mathbf{x}_i, l_i) \in \mathcal{X}_d). \tag{3}$$

B. Active Perception for Face Recognition

Even though this static approach presented in the previous subsection allows for achieving quite impressive face recognition results, as well as for easily training the model using a collection of static images, it comes with an important drawback: it ignores the ability of robotic systems to interact with the environment in order to get a more discrimintive view for the task at hand. For example, a drone carrying a camera can fly to the appropriate direction in order to acquire a more clean frontal view of a person, allowing for analyzing the input with greater confidence. To this end, we introduce

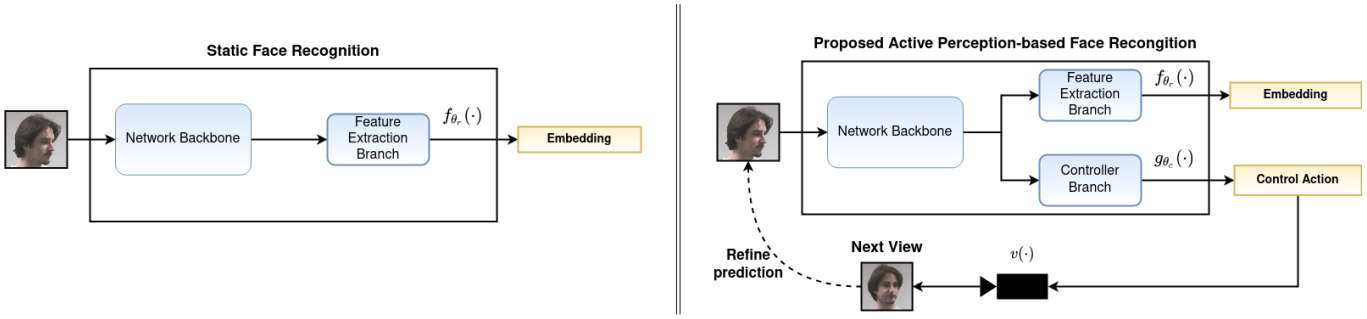

Fig. 1: Comparing the proposed active perception-based approach to static face recognition. We simultaneously train a DL model to predict both a discriminative feature vector, which is used for face recognition, as well as a high-level description for the next best action, which can be used for acquiring a better and more discriminative view of the input. Note that the model is *not* trained for facial pose estimation, yet it implicitly learns the control actions that will lead to the view that will provide the best recognition results.

a trainable *controller* $\mathbf{a}_t = g_{\theta_c}(\mathbf{x}^{(t)})$, where θ_c is a set of trainable parameters for the controller model, that receives an observation (image) $\mathbf{x}^{(t)}$ from the environment at time t and provides an appropriate control command $\mathbf{a}_t$ to the robot. Then, the updated observation is obtained by *executing* the corresponding action $\mathbf{a}_t$ as:

$$\mathbf{x}^{(t+1)} = v(\mathbf{a}_t, t), \tag{4}$$

where $v(\cdot)$ is either a model of the environment that returns the result of a simulated action $\mathbf{a}_t$ at time t, or the real environment, in the case of deploying the model into a real system, where we execute the corresponding action and get the updated observation. In this way, the controller $g_{\theta_c}(\cdot)$ provides a way to actively interact with the environment in order to get updated sensory stimuli, that will, in turn, lead to more accurate predictions for the embedding extractor $f_{\theta_r}(\cdot)$. For the rest of the paper, we will refer to the environment $v(\mathbf{a}_t, t)$ as $v(\mathbf{a}_t)$ to avoid cluttering the used notation.

Both the feature extractor model $f_{\theta_r}(\cdot)$, as well as the controller model $g_{\theta_c}(\cdot)$ must be appropriately trained for the task at hand. That is, the feature extractor model must be trained to extract discriminative embeddings, while the controller model must be trained in order to provide the appropriate control commands that will allow for getting a view that will maximize the face recognition accuracy. To this end, the optimization problem provided in (1) is updated following the proposed active perception setting:

$$\theta_r, \theta_c = \arg\min_{\theta_1, \theta_2} \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \mathcal{L}(f_{\theta_1}(v(g_{\theta_2}(\mathbf{x}_i))), f_{\theta_1}(\mathbf{x}_j), d_{ij}). \tag{5}$$

Note that the active perception controller is allowed to manipulate only the first input to the loss function, since the other one corresponds to the fixed version that is stored in the database. Ideally, the controller should be aware of the remaining images contained in the database, since this information can be exploited in order to acquire a view that matches the view of the person that is stored in the database. However, in this paper we assume that all images in the database are acquired under similar conditions, allowing for ignoring possible view variations of the database images.

It is also worth noting that (5) provides (at least) two different ways to minimize the employed loss: a) either by learning a powerful view-invariant feature extractor $f_{\theta_r}(\cdot)$, or b) by jointly learning a feature extractor along with an appropriate controller that is capable of acquiring images that makes the recognition problem easier. For the deployment, and after training both models simultaneously, the robot can either output the most probable prediction according to (3), or first appropriately control its camera (e.g., by moving itself or controlling a gimbal) in order to acquire an updated view. It is also possible that the controller will not provide any suggested action, as further explained below. In this case, we assume that the robot has already obtained an optimal view and no action should be performed.

Without loss of generality, in this paper we assume that control will be performed in discrete steps on one (horizontal) axis, which lies along a sphere centered on the subject's face. The proposed approach can be directly extended to handle multiple axes as well. Therefore, the controller $g_{\theta_c}(\cdot) \in \mathbb{R}^3$ support three possible actions:

1) "stay" ($\mathbf{a}_t$ = (1, 0, 0)),
2) "left" ($\mathbf{a}_t$ = (0, 1, 0)),
3) "right" ($\mathbf{a}_t$ = (0, 0, 1)),

with each of the three output neurons corresponding to the confidence of the controller for performing each of the three possible actions. Note that the controller $g_{\theta_c}(\cdot)$ only provides high-level control commands, which must be appropriately translated into actual control commands by using an appropriate controller, e.g., an PID controller [6], [22].

Despite the updated formulation provided in (5), it is still not straightforward to directly optimize $g_{\theta_c}(\cdot)$, since the model $v(\cdot)$ is usually not fully known and it is not differentiable.

Instead of using reinforcement learning (RL), which is typically used to solve such control problems [15], [17], in this work we opt for a simpler, yet more efficient approach, which arises from the following assumption/observation: the recognition confidence is expected to monotonically increase or decrease when moving towards the same direction (at least for small continuous intervals). We call this *smooth control manifold* assumption. Even though it is possible that this assumption does not hold in practice, this approach allows for significantly simplifying the optimization process, as well as increasing the face recognition accuracy, as we further demonstrate in Section III. Therefore, for each face image sampled during the training process, we propose executing all the three possible actions simultaneously (using the appropriate simulation environment/dataset) and observing the effect on the recognition confidence. Let $\mathbf{x}_{i0}$, $\mathbf{x}_{i1}$, and $\mathbf{x}_{i2}$ denote the updated facial image obtained after moving the robot to the left, right, or performing no action. Then, the training target for the controller can be trivially acquired by choosing the action that minimizes the distance between the representation of the correct face and the current face. Therefore, the controller target $d_i^{(a)}$ for an image $\mathbf{x}_i$ and a positive example $\mathbf{x}_p$ is acquired as:

$$d_i^{(a)} = \arg \min_{k \in \{0,1,2\}} ||\mathbf{x}_{ik} - f(\mathbf{x}_p)||_2. \tag{6}$$

For negative examples there are two options: a) either not training the controller with them, or b) training the controller in order to again minimize the distance between the embedding vectors (despite belonging to different persons). The motivation for the latter option is that the controller should always perform control in order to find the view that provides the best matching between face embeddings. In this work, the first option was selected, since it was experimentally shown to lead to slightly better recognition results.

Therefore, the loss to be minimized when optimizing the controller is defined as:

$$\mathcal{L}_g = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} d_{ij} \mathcal{L}_x(g_{\theta_c}(\mathbf{x}_i),\ d_i^{(a)}), \tag{7}$$

where $\mathcal{L}_x$ denotes the categorical cross-entropy loss. The feature extractor can be still trained as before, i.e., by minimizing the loss:

$$\mathcal{L}_f = \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \mathcal{L}(f_{\theta_r}(\mathbf{x}_i), f_\theta(\mathbf{x}_j),\ d_{ij}), \tag{8}$$

as provided in (1). Therefore, the final loss is obtained as:

$$\mathcal{L} = \mathcal{L}_g + \mathcal{L}_f \tag{9}$$

Gradient descent is employed for optimizing both models as:

$$\Delta\theta_r = -\eta_r \frac{\partial \mathcal{L}}{\partial \theta_r},\ \text{and}\ \Delta\theta_c = -\eta_c \frac{\partial \mathcal{L}}{\partial \theta_c}, \tag{10}$$

where η_r and η_c are the learning rates for the feature extraction and controller models respectively. For all the experiments conducted in this paper we set $\eta_r = \eta_c = 10^{-3}$, while a common backbone network with convolutional layers is shared between $f_{\theta_r}(\cdot)$ and $g_{\theta_c}(\cdot)$, with two different branches used for implementing these two function, as further described in Section III. The structure of the employed network architecture is depicted in Fig. 1.

III. Experimental Evaluation

The experimental evaluation is provided in this Section. First, the employed datasets, experimental setup and neural network architectures are presented. Then, the proposed method is extensively evaluated and discussed.

A. Dataset and Experimental Setup

The proposed method was evaluated using the Head Pose Image Dataset (HPID) [23], which contains facial images of several persons at various pans and tilts, ranging from $-90°$ to $90°$. The HPID dataset was selected for evaluating the proposed method, since it contains real images at various angles (instead of simulated faces), as also depicted in Fig. 2, and it provides the full range of pans (from $-90°$ to $90°$ with $15°$ steps). The small number of identities and face images per pose contained in the dataset renders this evaluation setup especially challenging, since it corresponds to a realistic few-shot learning scenario, which is often encountered in various robotics applications [24].

The 75% of the persons contained in the dataset was used to train the models, while the remaining 25% persons were used for evaluating the trained models. All experiments were conducted 5 times and the mean and standard deviation of the recognition accuracy is reported. Two evaluation setups were used to examine the performance of the proposed method under two different settings: a) "Set 1", where the recognition database contains face images with pans between $-15°$ to $15°$, and a) "Set 2", where face images with pans between $30°$ to $60°$ were used. Images with tilt between $-30°$ and $30°$ were used for all the conducted experiments.

The backbone network used for the conducted experiments consists of four 3×3 convolutional layers with 8, 16, 32 and 64 filters, respectively. The ReLU activation function was used for all the convolutions layer [25], while 2×2 max pooling was employed after each layer. A fully connected layer with 256 neurons follows the last convolutional layer of the backbone. The feature extraction branch consists of a fully connected layer with 64 neurons, while the controller branch is composed of a fully connected layer with 32 hidden neurons and a final layer with 3 neurons (one for each action). Images of 88×88 pixels were fed to the network, while the Adam optimizer was employed for the optimization [26]. The proposed method was pre-trained on the training dataset for 100 epochs (by training only the feature extraction branch), followed by 50 training epochs, where both branches were simultaneously optimized. Finally, note that the cross entropy loss was weighted with class weights, which were equal to 1 for the left and right actions and to 0.01 for the stay action, since the model tended to more frequently select this action.

Fig. 2: Sample images contained in the HPID dataset. Note that a complete set of poses are included in the dataset, allowing for evaluating the proposed method using real data instead of using face images generated from simulators.

TABLE I: Experimental Evaluation using the HPID dataset

Method	Accuracy (Set 1)	Accuracy (Set 2)
Static Perception	$54.1 \pm 3.4\%$	$49.9 \pm 4.1\%$
Static Perception (finet.)	$52.7 \pm 4.2\%$	$51.4 \pm 4.6\%$
Proposed (1 step)	$61.6 \pm 5.1\%$	$58.8 \pm 7.0\%$
Proposed (3 steps)	$\mathbf{62.2 \pm 5.9}\%$	$\mathbf{58.9 \pm 6.5}\%$

TABLE II: Evaluating the effect of model size on face recognition accuracy

Method	Network	Accuracy	# Param.
Static Perception	$0.25\times$	$38.8 \pm 6.6\%$	12k
Static Perception	$0.5\times$	$49.0 \pm 5.5\%$	47k
Static Perception	$1\times$	$54.1 \pm 3.4\%$	189k
Proposed	$0.25\times$	$\mathbf{57.5 \pm 5.8}\%$	14k
Proposed	$0.5\times$	$\mathbf{60.2 \pm 6.9}\%$	52k
Proposed	$1\times$	$\mathbf{62.2 \pm 5.9}\%$	197k

B. Experimental Evaluation

The evaluation results are presented in Table I. The proposed method is compared to a static perception approach ("Static Perception"), where the exact same DL model is used, but without employing the control branch. This baseline was trained for 100 epochs. To ensure a fair comparison with the proposed method, we also report evaluation results for the same model, further finetuned for 50 additional training epochs. The proposed method manages to increase the recognition accuracy by more than 7%, just after one control step, which can lead in a view change of at most 15°. This demonstrates the effectiveness of the proposed active perception approach and indicates the importance of exploiting the ability of a robotic system to interact with the environment in order to acquire a better and more discriminative view for the task at hand, confirming the first hypothesis posed in Section I. It is worth noting that significant improvements are obtained regardless the type of images contained in the database (Setup 1 and Setup 2). When allowed to perform additional control steps (3 steps instead of just 1), the proposed method again further increases the recognition accuracy for both setups. These results suggest that using more fine-grained control approaches, that can further adjust the control steps according the current state, can potentially lead to even greater accuracy improvements.

To evaluate the second hypothesis, i.e., that using active perception allows for using smaller and faster models with only a small impact on the final recognition accuracy, an additional set of experiments was conducted. The experimental results are reported in Table II. Two different networks were employed to this end. The same architecture as in the previous experiment is used, but the number of neurons per layer is reduced by $0.5\times$ and $0.25\times$ (respectively) compared to the original architecture. Indeed, active perception can exceed the performance of traditional static perception models, using one order of magnitude less parameters, leading to faster and more accurate models.

IV. Conclusions

In this work, we presented a DL method for face recognition that utilizes active perception. The proposed method employs a hybrid architecture with two output branches, allowing for simultaneously learning discriminative embeddings, as well as providing the appropriate high-level control commands. As it was experimentally demonstrated, the proposed method indeed allows for improving the face recognition accuracy, while, at the same time, allows for using simpler and more efficient models to perform face recognition, leveraging the ability of the system to acquire a view that is more suitable for the task at hand. Finally, it is worth noting that the proposed method

is generic and task-agnostic and it provides a straightforward way of enabling active-perception capabilities for many of the existing static perception DL models.

The proposed method paves the way for developing generic active perception approaches for a wide variety of tasks, since it provides an efficient task-agnostic way for training DL methods that can actively interact with the environment to obtain a more discrimnative view. The most important obstacle for developing such approaches is the lack of appropriate datasets, needed for obtaining realistic views of the environments. Apart from collecting such datasets, which is an expensive and tedious task, several interesting alternatives exist. For example, Generative Adversarial Networks can be used to perform image-to-image translation to obtain different views for existing datasets [27], [28], 3D models can be created from real images and the various views can be directly rendered [29], both real and simulated data can be combined using highly realistic simulations [30], [31], while knowledge distillation methods could be used to further reduce the gap between real and simulated data [32], [33].

REFERENCES

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

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[4] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. European Conference on Computer Vision*, 2016, pp. 499–515.

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] Nikolaos Passalis, Anastasios Tefas, and Ioannis Pitas, "Efficient camera control using 2d visual information for unmanned aerial vehicle-based cinematography," in *Proc. IEEE International Symposium on Circuits and Systems*, 2018, pp. 1–5.

[7] Nikolaos Passalis and Anastasios Tefas, "Deep reinforcement learning for controlling frontal person close-up shooting," *Neurocomputing*, vol. 335, pp. 37–47, 2019.

[8] Quan Liu, Zhihao Liu, Wenjun Xu, Quan Tang, Zude Zhou, and Duc Truong Pham, "Human-robot collaboration in disassembly for sustainable manufacturing," *International Journal of Production Research*, vol. 57, no. 12, pp. 4027–4044, 2019.

[9] Jian-hua Li, "Cyber security meets artificial intelligence: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 12, pp. 1462–1474, 2018.

[10] George Loukas, Tuan Vuong, Ryan Heartfield, Georgia Sakellari, Yongpil Yoon, and Diane Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491–3508, 2017.

[11] Yiannis Aloimonos, *Active perception*, Psychology Press, 2013.

[12] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[13] Macheng Shen and Jonathan P How, "Active perception in adversarial scenarios using maximum entropy deep reinforcement learning," in *Proc. International Conference on Robotics and Automation*. IEEE, 2019, pp. 3384–3390.

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

[15] Phil Ammirato, Patrick Poirson, Eunbyung Park, Jana Košecká, and Alexander C Berg, "A dataset for developing and benchmarking active vision," in *Proc. IEEE International Conference on Robotics and Automation*, 2017, pp. 1378–1385.

[16] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese, "Gibson env: Real-world perception for embodied agents," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9068–9079.

[17] Xiaoning Han, Huaping Liu, Fuchun Sun, and Xinyu 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.

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

[19] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 212–220.

[20] Raia Hadsell, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1735–1742.

[21] Zheng Lian, Ya Li, Jianhua Tao, and Jian Huang, "Speech emotion recognition via contrastive loss under siamese networks," in *Proc. Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, 2018, pp. 21–26.

[22] Connor Schenck and Dieter Fox, "Visual closed-loop control for pouring liquids," in *Proc. IEEE International Conference on Robotics and Automation*. IEEE, 2017, pp. 2629–2636.

[23] Nicolas Gourier, Daniela Hall, and James L Crowley, "Estimating face orientation from robust detection of salient facial features," in *Proc. ICPR International Workshop on Visual Observation of Deictic Gestures*, 2004.

[24] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[26] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] Rameen Abdal, Yipeng Qin, and Peter Wonka, "Image2stylegan: How to embed images into the stylegan latent space?," in *Proc. IEEE International Conference on Computer Vision*, 2019, pp. 4432–4441.

[28] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[29] Alexandros Lattas, Stylianos Moschoglou, Baris Gecer, Stylianos Ploumpis, Vasileios Triantafyllou, Abhijeet Ghosh, and Stefanos Zafeiriou, "Avatarme: Realistically renderable 3d facial reconstruction" in-the-wild"," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[30] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and service robotics*. Springer, 2018, pp. 621–635.

[31] Olivier Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 5, 2004.

[32] Nikolaos Passalis and Anastasios Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 268–284.

[33] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2339–2348.