



# OpenDR — Open Deep Learning Toolkit for Robotics

Project Start Date: 01.01.2020

Duration: 48 months

Lead contractor: Aristotle University of Thessaloniki

**Deliverable D3.4: Final report on deep human centric active perception and cognition**

Date of delivery: 29 September 2023

Contributing Partners: AUTH, AU, TAU  
Version: v3.0



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449.

<b>Title</b>	<b>D3.4: Final report on deep human centric active perception and cognition</b>
<b>Project</b>	<b>OpenDR (ICT-10-2019-2020 RIA)</b>
<b>Nature</b>	Report
<b>Dissemination Level:</b>	<b>Public</b>
<b>Authors</b>	Avramelou Loukia (AUTH), Kakaletsis Efstratios (AUTH), Pas-salis Nikolaos (AUTH), Kirtas Emmanouil (AUTH), Manousis Theodoros (AUTH), Babis Emmanouil (AUTH), Nousi Paraskevi (AUTH), Tzelepi Maria (AUTH), Symeonidis Charalampos (AUTH), Spanos Dimitrios (AUTH), Tosidis Pavlos-Apostolos (AUTH), Tsampazis Konstantinos (AUTH), Tefas Anastasios (AUTH), Nikolaidis Nikolaos (AUTH), Quoc Nguyen (TAU), Jussi Taipalmaa (TAU), Kateryna Chumachenko (TAU), Anton Muravev (TAU), Moncef Gabbouj (TAU), Illia Oleksienko (AU), Alexandros Iosifidis (AU)
<b>Lead Beneficiary</b>	TAU (Tampere University)
<b>WP</b>	3
<b>Doc ID:</b>	OPENDR_D3.4.pdf

## Document History

<b>Version</b>	<b>Date</b>	<b>Reason of change</b>
v1.0	9/5/2023	Deliverable structure template ready
v2.0	25/09/2023	Contributions from partners finalized
v3.0	29/09/2023	Final version ready

## Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Deep person/face/body part active detection/recognition and pose estimation (T3.1)	6
1.2	Deep person/face/body part tracking, human activity recognition (T3.2)	6
1.3	Social signal (facial expression, gesture, posture, etc.) analysis and recognition (T3.3)	6
1.4	Deep speech and biosignals analysis and recognition (T3.4)	6
1.5	Multi-modal human centric perception and cognition (T3.5)	7
1.6	Connection to Project Objectives	7
<b>2</b>	<b>Deep person/face/body part active detection/recognition and pose estimation</b>	<b>9</b>
2.1	Using Synthesized Facial Views for Active Face Recognition	9
2.2	Multi-axes Control for Embedding-based Active Face Recognition	10
2.2.1	Introduction	10
2.2.2	Description of work performed so far	10
2.2.3	Performance evaluation	12
2.3	Active Perception for enabling Efficient High Resolution Pose Estimation	15
2.3.1	Introduction	15
2.3.2	Description of work performed so far	15
2.3.3	Performance evaluation	17
2.4	Deep Reinforcement Learning for Active Perception	18
2.4.1	Introduction	18
2.4.2	Description of work performed so far	18
2.4.3	Performance evaluation	21
2.5	Neural Attention driven Non-Maximum Suppression for Person Detection	24
2.5.1	Introduction, objectives and work performed so far	24
<b>3</b>	<b>Deep person/face/body part tracking, human activity recognition</b>	<b>25</b>
3.1	Variational Spatio-Temporal Graph Convolutional Networks for Skeleton-based Action Recognition	25
3.1.1	Introduction	25
3.1.2	Summary of the state of the art	26
3.1.3	Description of work performed so far	28
3.1.4	Performance evaluation	30
<b>4</b>	<b>Social signal (facial expression, gesture, posture, etc.) analysis and recognition</b>	<b>31</b>
4.1	RGB Hand Detection and Gesture Recognition	31
4.1.1	Introduction	31
4.1.2	Summary of state of the art	31
4.1.3	Description of the work performed so far	32
4.1.4	Performance evaluation	33
<b>5</b>	<b>Deep speech and biosignals analysis and recognition</b>	<b>33</b>
5.1	Integrating Whisper and Vosk in Speech Transcription	33
5.1.1	Introduction and summary of state of the art	33
5.1.2	Description of the work performed so far	34

5.1.3	Performance evaluation . . . . .	34
<b>6</b>	<b>Multi-modal human centric perception and cognition</b>	<b>36</b>
6.1	Improving Unimodal Inference with Multimodal Transformers . . . . .	36
6.1.1	Introduction . . . . .	36
6.1.2	Summary of state of the art . . . . .	36
6.1.3	Description of the work performed so far . . . . .	37
6.1.4	Performance evaluation . . . . .	38
6.2	Multi-frame person detection . . . . .	39
6.2.1	Introduction and summary of state of the art . . . . .	39
6.2.2	Description of work performed so far . . . . .	40
6.2.3	Performance evaluation . . . . .	40
<b>7</b>	<b>Conclusions</b>	<b>42</b>
<b>8</b>	<b>Appendix</b>	<b>48</b>
8.1	Using Synthesized Facial Views for Active Face Recognition . . . . .	48
8.2	Neural Attention-Driven Non-Maximum Suppression for Person Detection . . . . .	76
8.3	Improving Unimodal Inference with Multimodal Transformers . . . . .	91
8.4	Deep learning for active robotic perception . . . . .	97

## Executive Summary

This document presents the final update on the work performed for **WP3–Deep human centric active perception and cognition**. This work package contains five main tasks. These are *Task 3.1–Deep person/face/body part active detection/recognition and pose estimation*, *Task 3.2–Deep person/face/body part tracking, human activity recognition*, *Task 3.3–Social signal (facial expression, gesture, posture, etc.) analysis and recognition*, *Task 3.4–Deep speech and biosignals analysis and recognition*, and *Task 3.5–Multi-modal human centric perception and cognition*. The document starts with a general introduction, providing an overview of the individual chapters and linking them to the main objectives of the project. The introduction is followed by chapters dedicated to each of the tasks. Each chapter provides an overview on the state of the art for the individual topics, details of the partners’ current work as well as performance results (where available). Finally, the conclusion section provides a closing overview of the work and the total progress of the work package.

# 1 Introduction

This document describes the work done during the final year of the project in the five major research areas of WP3, namely:

- Deep person/face/body part active detection/recognition and pose estimation;
- Deep person/face/body part tracking, human activity recognition;
- Social signal (facial expression, gesture, posture, etc.) analysis and recognition;
- Deep speech and biosignals analysis and recognition;
- Multi-modal human centric perception and cognition.

## 1.1 Deep person/face/body part active detection/recognition and pose estimation (T3.1)

AUTH further extended high resolution pose estimation approach developed in OpenDR in order to more efficiently handle cases where multiple humans appear, as well as included a more challenging evaluation setup in Section 2.3. AUTH also further developed (Section 2.2) an embedding-based active perception approach for face recognition by leveraging a new dataset developed by AUTH to enable multi-axes control. Finally, AUTH finalized the work conducted on Non-Maximum Suppression (NMS) for person detection (Section 2.5), as well as on using synthesized facial views (Section 2.1) and deep reinforcement learning (Section 2.4) for active face recognition.

## 1.2 Deep person/face/body part tracking, human activity recognition (T3.2)

AU worked on incorporating uncertainty estimation in skeleton-based human action recognition by proposing variational version of ST-GCN and AGCN models (Section 3.1). The proposed variational perception methods improve the quality of human action recognition and estimate model uncertainties that can be further used for active perception.

## 1.3 Social signal (facial expression, gesture, posture, etc.) analysis and recognition (T3.3)

TAU worked on extending the hand gesture recognition functionality of the OpenDR toolkit (Section 4.1). The newly contributed tool achieves higher recognition accuracy, as well as performing the localization of hands on the image, making it more flexible and robust compared to the previously integrated solution.

## 1.4 Deep speech and biosignals analysis and recognition (T3.4)

Speech recognition is an essential component in human-robot interaction, allowing robots to understand and respond to vocal commands in noisy and dynamic environments. Various open toolkits and research projects have laid the groundwork for advancements in this field. TAU contributed by integrating existing speech recognition and transcription toolkits into OpenDR, ensuring their compatibility and viability for the use cases (Section 5.1).

## 1.5 Multi-modal human centric perception and cognition (T3.5)

TAU has developed a method to allow for better performance in a variety of unimodal deep learning tasks (such as hand gesture recognition, audiovisual emotion recognition, language sentiment analysis, etc.) by training the models in a multimodal fashion. A specific instance of this method, trained to perform text-based intent recognition, has been successfully integrated into the OpenDR toolkit (Section 6.1). Furthermore, TAU has also developed a method for multi-frame human detection (Section 6.2).

## 1.6 Connection to Project Objectives

The work performed within WP3, as summarized in the previous subsections, perfectly aligns with the project objectives. More specifically, the conducted work progressed the state-of-the-art towards meeting following objectives of the project:

O1 *To provide a modular, open and non-proprietary toolkit for core robotic functionalities enabled by lightweight deep learning*

O1a *To enhance the robotic autonomy exploiting lightweight deep learning for on-board deployment*

AUTH finalized the work conducted on Non-Maximum Suppression (NMS) for person detection (Section 2.5).

O1b *To provide real-time deep learning tools for robotics visual perception on high resolution data*

AUTH further extended high resolution pose estimation approach developed in OpenDR in order to more efficiently handle cases where multiple humans appear, as well as included a more challenging evaluation setup in Section 2.3.

O2 *To leverage AI and Cognition in robotics: from perception to action*

O2a *To propose, design, train and deploy models that go beyond static computer perception, towards active robot perception*

AUTH further developed (Section 2.2) an embedding-based active perception approach for face recognition by leveraging a new dataset developed by AUTH to enable multi-axes control. AUTH also finalized the method proposed for active face recognition using synthesized facial views (Section 2.1), as well as deep reinforcement learning (Section 2.4).

AU worked towards uncertainty estimation for skeleton-based human action recognition to create perception methods that provide valuable uncertainty information that can be used for active perception system to decide when to rely on the perception and when to perform action to improve perception certainty. AU implemented Variation Spatio-Temporal Graph Convolutional Networks, which provide prediction uncertainty and improve the quality of human action recognition models (Section 3.1).

O2b *To provide specific deep human-centric active robot perception tools*

TAU has extended the functionalities of the OpenDR toolkit by developing, implementing and integrating the following tools: speech recognition and transcription (Section 5.1),

hand gesture recognition (Section 4.1), as well as multimodally trained text-based intent recognition (Section 6.1) and human detection from high resolution frames (Section 6.2).



## 2 Deep person/face/body part active detection/recognition and pose estimation

### 2.1 Using Synthesized Facial Views for Active Face Recognition

During this period AUTH finalized the work conducted in active face recognition using synthesized facial views. As described in D3.3 and in previous deliverables, the proposed approach utilizes facial views synthesized by photorealistic facial image rendering. Essentially, the camera-equipped robot that performs the recognition in an appropriate environment (as shown in Figure 1) selects the best among a number of candidate physical movements around the face of interest by simulating their results through view synthesis. In other words, once the robot (that is at a certain location with respect to the subject) acquires an image, it feeds the face recognizer with this image as well as with synthesized views that differ by  $\pm\theta^\circ$  from the current view. Subsequently, it either stays in the current position or moves to the position that corresponds to one of the two synthesized views. The respective decision is based on the confidence of the three recognitions (on the real and the two synthesized views). In case of a "move" decision, it proceeds in acquiring a "real" image from its new location. The procedure repeats in the same manner, for this location, for one or more steps.

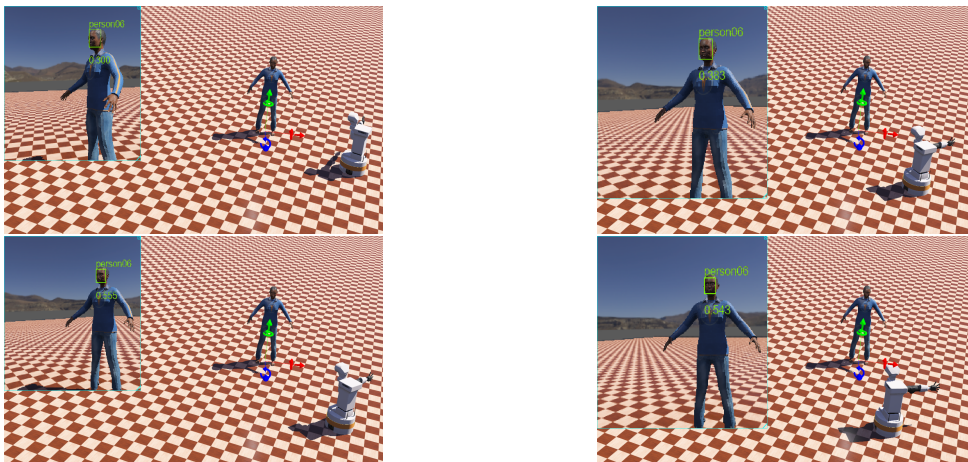


Figure 1: A simulation in Webots where a TIAGo mobile robot performs active face recognition on person 06 starting from two different initial robot positions (top row) and reaching its final position (respective image at the bottom row). The sub-image at the upper-left corner depicts the robot camera view along with the face detection bounding box, person label and recognition confidence.

Experimental evaluations that were conducted in the previous periods utilizing three datasets verified the superior performance of the proposed method compared to a state of the art active method as well as other baseline approaches.

Within this period, as part of the review procedure for the paper that was submitted in the previous period (see D3.3) in the Machine Vision and Applications journal, a number of additional experiments were performed. In more detail, the proposed approach was compared against a dummy active method that involves robot movement in random directions, and a method that employs physical robot movement towards a frontal location based on the view angle estimation provided by the view synthesis algorithm. The proposed approach surpassed

both approaches by a large margin. A proof of concept simulation in Webots was also created. After the revision process the paper was accepted for publication:

- Efstratios Kakaletsis, Nikos Nikolaidis, "Using Synthesized Facial Views for Active Face Recognition", *Machine Vision and Applications* 34, 62, Springer, (2023)

Moreover, a short version of this paper was accepted and presented in EUSIPCO 2023 conference:

- Efstratios Kakaletsis, Nikos Nikolaidis, "Active Face Recognition Through View Synthesis", 2023 EURASIP 31st European Signal Processing Conference (EUSIPCO 2023), Helsinki, Finland, September 4-8, 2023

The corresponding publications can be found in Appendix 8.1.

## 2.2 Multi-axes Control for Embedding-based Active Face Recognition

### 2.2.1 Introduction

Active vision aims to equip computer vision methods with the ability to dynamically adjust the capturing sensor's viewpoint, position, or parameters in real time. This dynamic capability allows for improving the accuracy of the perception process. However, this process largely depends on the availability of appropriate datasets and simulation environments. Previous approaches, such as [51], were limited due to the small number of available movement steps and number of data samples. In this Section, we demonstrate how our previous approach, proposed in [51], can be extended to handle additional control axes, when appropriate data and annotations are available. To this end, we introduce an embedding-based active perception approach for face recognition capable of performing 2-axis control, leveraging the additional information provided by ActiveFace dataset, developed by AUTH, to be detailed in D6.4.

### 2.2.2 Description of work performed so far

In this Section we detail an extension of the embedding-based active face recognition method presented in [51]. This method was shown to yield much better recognition results than the ones achieved when using a static perception approach, since it takes advantage of a robot's ability to interact with its environment in order to get a more informative view of the person's face. We demonstrate how the proposed extension can exploit rich annotations and the variety of data provided by the extended active perception dataset, enabling us to acquire even better results. This is achieved with the use of a trainable controller which, when given an image  $\mathbf{x}^{(t)}$  at a time  $t$ , dictates the robot to move towards a certain direction in order to acquire a new image which offers a better frontal view of the person. The new image is given by:

$$\mathbf{x}^{(t+1)} = v(a_t, t), \quad (1)$$

where  $v(\cdot)$  denotes the current environment. The trainable controller is represented as:

$$a_t = g_{\theta_c}(\mathbf{x}^{(t)}), \quad (2)$$

where  $\theta_c$  denotes a set of trainable action parameters.

The model is comprised of two modules, the feature extractor model  $f_{\theta_r}(\cdot)$ , which learns discriminative embeddings of a given face image, thus being able to separate the representations extracted from images that belong to different persons, and the controller model  $g_{\theta_c}(\cdot)$  which is responsible for learning the best possible action that the robotic system should take next in order to acquire a better view of a person's face.

When an unseen image is given as input during the evaluation process and the controller has given the appropriate control commands to the robotic system, the id of the person is obtained using the 1-nearest neighbor approach on a database that contains frontal and nearly frontal facial images for every person.

Instead of using reinforcement learning when training the controller, the model executes all possible control actions at the same time and calculates the recognition accuracy of each of the obtained images, improving learning efficiency [51]. The action that led to the lowest distance between the representation of the current face and the correct face is retained and used to train the controller. The optimal action when given an image  $x_i$  and a correct image  $x_p$  is given by:

$$d_i^{(a)} = \operatorname{argmin}_{k \in \{0,1,2,\dots,n\}} \|f(\mathbf{x}_{ik}) - f(\mathbf{x}_p)\|_2, \quad (3)$$

where  $n$  is the total number of possible actions that the controller can choose.

The loss function that the controller aims to minimize is given by:

$$L_g = \sum_{i=1}^N L_x(g_{\theta_c}(\mathbf{x}_i), d_i^{(a)}), \quad (4)$$

where  $L_x$  represents the cross-entropy loss function. The feature extractor, on the other hand, aims to minimize the following loss function:

$$L_f = \sum_{i=1}^N \sum_{j=1, j \neq i}^N L_e(f_{\theta_r}(\mathbf{x}_i), f_{\theta_r}(\mathbf{x}_j), d_{ij}), \quad (5)$$

where the binary variable  $d_{ij} \in \{0, 1\}$  denotes whether the  $i$ -th face image belongs to the same person as the one depicted in the  $j$ -th face image and  $L_e$  is a loss that encourages the separability of different face embeddings. In this work we use the contrastive loss, as suggested in [51], 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  $\sqrt{m}$ :

$$L_e(\mathbf{y}_i, \mathbf{y}_j, d_{ij}) = d_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 + (1 - d_{ij}) \max(0, m - \|\mathbf{y}_i - \mathbf{y}_j\|_2)^2, \quad (6)$$

where  $\mathbf{y}_i = f_{\theta_r}(\mathbf{x}_i)$  is the representation extracted from the face recognition model and  $\|\cdot\|_2$  refers to the  $l^2$  norm of a vector. The final loss of the model is given by the sum of (4) and (5):

$$L = L_g + L_f \quad (7)$$

The model uses the Adam optimization algorithm with initial learning rates  $\eta_r = \eta_c = 10^{-3}$  for the feature extractor and controller, respectively.

We appropriately modified the aforementioned approach to allow for an extra Front (i.e., towards the subject) movement/action of  $0.5m$  per move in order to take advantage of the range of camera-subject distances provided by the dataset generated in this work. The Left and Right

actions dictate the controller to move by 10 degrees either to the left or to the right, respectively, on a circle centered at the human subject. Since the classes involved in (4) are not balanced, different weights were used for different classes. More specifically, the Stay action was given a action weight of 0.01, while both the Left and Right ones were given a weight of 1 and the Front was weighted by 1.2.

Since the employed dataset does not contain, due to the existence of furniture, images from every camera/robot position, it was observed that the model was not always able to find an existing image for every available action. As each environment had missing images at different camera distances and angles (i.e., for the locations occupied by the furniture) and the model could learn to avoid collisions for environments that do not require such actions, it was decided to not train the model for any image where any of the left, right or front images are missing. During inference, the controller chooses the best action that leads to an image that exists. If for a given image there are no left, right or front images the controller dictates the robotic system to stay in place. In a real-world scenario, the controller would output different actions, from most optimal to less optimal, until the robotic system could move towards the best available spot.

### 2.2.3 Performance evaluation

The develop active vision model was evaluated on an active perception dataset generated by AUTH, called *ActiveFace* face image dataset, which contains two different evaluation sets. Set 1 contains the whole dataset, while Set 2 contains only facial images with a pan range of  $-90^\circ$  to  $90^\circ$  ( $0^\circ$  corresponds to frontal view). In both cases, the training set consisted of 22 subjects, while the remaining 11 were used to evaluate the trained model. For those 11 subjects, all the frontal and nearly frontal ( $-10^\circ$  to  $10^\circ$ ) images at  $1m$  distance away from the human, for every environment and lighting condition, were added to the recogniser database, while the remaining ones were used for testing the trained model, i.e., they were used as images captured at the starting location of the robot. All images were resized to  $96 \times 96$  and all experiments were conducted 5 times using different random seeds and the mean and standard deviation of their accuracy scores was recorded. For each of Sets 1 and 2 both a static and an active vision model were trained in order to evaluate the increase in accuracy when using the latter method. It is expected that the network will perform worse on the entire dataset (Set 1) compared to its accuracy score on the  $-90^\circ$  to  $90^\circ$  subset (Set 2), since the model may not even detect a face for extreme pan values and large distances. The active model was first pretrained without the control branch and then trained simultaneously on both the feature extractor and the control branch.

The static vision model was trained for 5, 10, 20 and 30 epochs for both Set 1 and Set 2. The active model was trained for 10 (5 for the feature extractor and 5 for both branches), 20 (10 for the feature extractor and 10 for both branches) and 30 (15 for the feature extractor and 15 for both branches) epochs for both subsets. Moreover, the active vision model was evaluated for 30 control steps, which would essentially allow the robotic system to move to any location in an environment. This way, the recognition accuracy ceiling of the model for both Sets 1 and 2 will be reached.

Finally, the active model was also trained and evaluated without the addition of the extra Front action in order to demonstrate how allowing the robotic system to move towards the subject can result in an increase in inference performance.

The evaluation results are shown in Tables 1, 2 and 3. As a reminder, **Set 1** represents the full ActiveFace dataset, while **Set 2** denotes the dataset with the reduced pan range.

Table 1: Static vision model evaluation: accuracy mean and standard deviation.

Model	Set 1	Set 2
Static (5 epochs)	$51.1 \pm 4.2\%$	$60.3 \pm 1.5\%$
Static (10 epochs)	$47.9 \pm 4.2\%$	$58.1 \pm 3.5\%$
Static (20 epochs)	$44.9 \pm 2.4\%$	$57.9 \pm 2.9\%$
Static (30 epochs)	$44.3 \pm 2.2\%$	$58.5 \pm 2.8\%$

Table 2: Active vision model evaluation **with** the additional Front movement/action: accuracy mean and standard deviation.

Model	Set 1	Set 2
Active (10 epochs)	$67.9 \pm 6.8\%$	$76.9 \pm 6.5\%$
Active (20 epochs)	$69.2 \pm 7.6\%$	$79.1 \pm 1.7\%$
Active (30 epochs)	$67.4 \pm 8.6\%$	$78.3 \pm 6.8\%$

Table 3: Active vision model evaluation **without** the additional Front movement/action: accuracy mean and standard deviation.

Model	Set 1	Set 2
Active (10 epochs)	$60.3 \pm 6.4\%$	$66.4 \pm 7.3\%$
Active (20 epochs)	$55.5 \pm 1.9\%$	$66.6 \pm 6.8\%$
Active (30 epochs)	$59.1 \pm 4.4\%$	$72.1 \pm 3.2\%$

Evaluation results for the static model are presented in Table 1. Clearly, the models perform best when trained for 5 epochs, reaching accuracy scores of  $\sim 51.1\%$  and  $\sim 60.3\%$  for **Set 1** and **Set 2**, respectively. Increasing the number of epochs seems to cause an overfit of the model on the training data.

Once the active approach is employed, a substantial increase in prediction accuracy can be observed for both datasets by a maximum of  $\sim 18.1\%$  and  $\sim 18.8\%$ , respectively, as seen in Table 2. Since we introduce more parameters, the models can be trained for more epochs and seem to overfit when the number of epochs is set to 30 (15 for the feature extractor and 15 for both branches). Evidently, the ability to train the robotic system to move within its environment in order to get a more informative view of the subject, namely a view which is closer to the frontal or nearly frontal views that the system has learned to recognize, yields much better face recognition results. Furthermore, once the controller's Front movement is removed (Table 3) the model is  $\sim 8.9\%$  and  $\sim 7\%$  less accurate than the one with the additional Front action, when comparing the highest recorded prediction accuracy scores of each respective conducted experiment.

This clearly demonstrates that allowing the model to move in more directions, i.e., not only around but also towards the subject, can further increase its ability to recognize faces.

Figure 2 depicts an example of how the control branch has learned to change its viewpoint in order to get a better (more closer and towards a frontal position) view of the person depicted in the original image, that is, the one obtained from the initial robot location. The original image is obtained from point **a** (starting position for the robot) and then the robot moves along the depicted path until it reaches a frontal view of the subject's face at a distance of  $1m$  (point

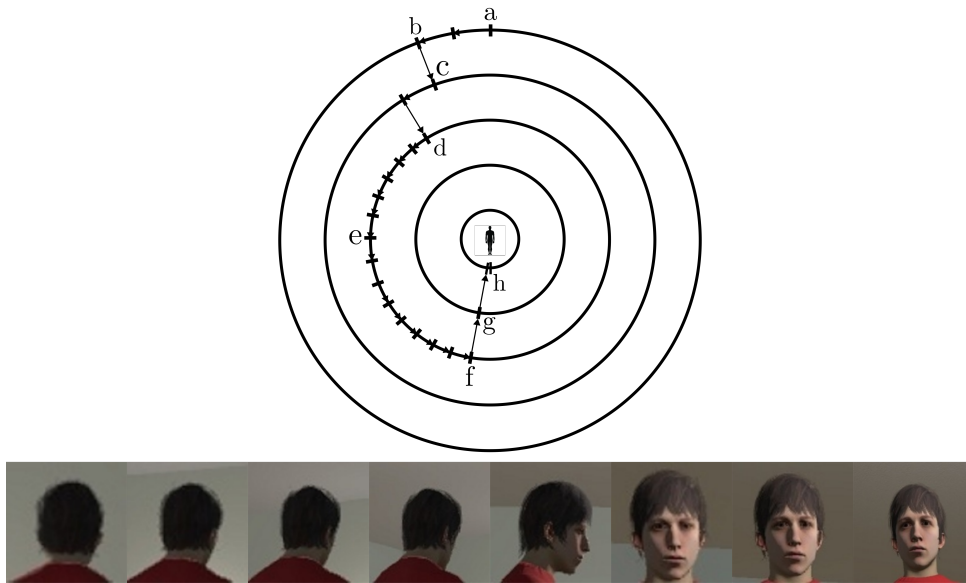
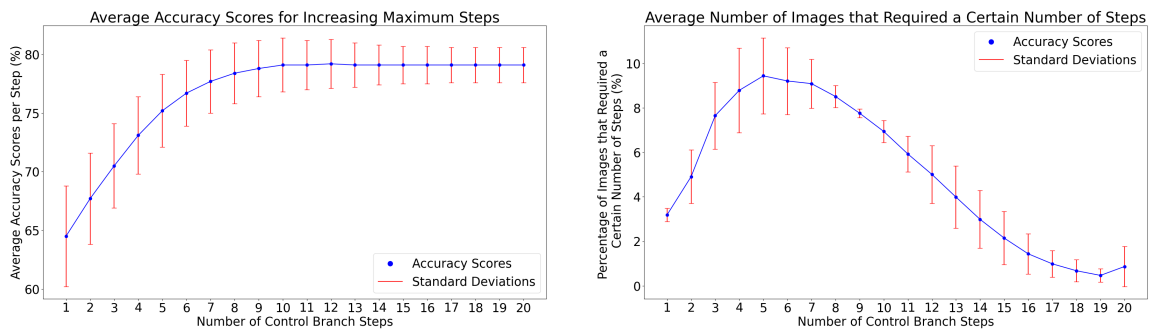


Figure 2: Example of control branch movements. Images from left to right correspond to the robot locations depicted on the diagram: (a) distance  $3m$ , angle  $180^\circ$ ; (b) distance  $3m$ , angle  $160^\circ$ ; (c) distance  $2.5m$ , angle  $160^\circ$ ; (d) distance  $2m$ , angle  $140^\circ$ ; (e) distance  $2m$ , angle  $90^\circ$ ; (f) distance  $2m$ , angle  $10^\circ$ ; (g) distance  $1.5m$ , angle  $10^\circ$ ; (h) distance  $1m$ , angle  $0^\circ$ .



(a) Average accuracy score when increasing maximum allowed control branch steps

(b) Average number of images that required  $n$  steps, where  $n = 1, 2, \dots, 20$ , before making a prediction.

Figure 3: Evaluating the impact of number of control steps on average accuracy score (Figure 3a), as well as showing the distribution of control steps needed in order to acquire the best possible view (Figure 3b).

**h).** We can observe that, generally, at larger distances the controller prefers to make a Front movement in order to move closer to the subject and, thus, increase the captured facial image resolution. Once the image is clear enough, it then makes either Left or Right movements in order to move in front of the subject.

Furthermore, we evaluated the prediction accuracy of each trained model as the number of allowed steps increased from 1 to 20 steps. Our hypothesis was that a well-trained model, neither underfitted nor overfitted, would reach a point where its performance stagnates. This suggests that the model does not need to take additional steps to obtain a better view of the subject's face. Fig 3a illustrates the average prediction accuracy of the model trained on Set 2 for 20 epochs, incorporating the Front movement command per maximum allowed number

of steps. We can observe that the accuracy increases as the number of steps increases until reaching a plateau at  $n = 12$ . This indicates that the active perception process converges and consistently produces better results with a higher number of steps up to a point, where the best view has been obtained.

Additionally, we recorded the average number of images that required a certain number of steps ( $n = 1, 2, \dots, 20$ ) before the active perception process stopped. Figure 3b represents the percentage of images that needed a specific number of steps to make a prediction for the same model. Most images required 5 steps, but the proportion of images requiring additional steps decreased gradually. Notably, at 20 steps, the recorded percentage appears to increase. We identified that this occurs in some cases where the controller reaches a frontal view of the subject's face but continues moving towards the left or right without stopping. This suggests that the agent may not be robust enough to consistently choose the Stay command when the robotic system achieves a frontal view of the subject.

## 2.3 Active Perception for enabling Efficient High Resolution Pose Estimation

### 2.3.1 Introduction

Pose estimation is one of the most essential challenges in the rapidly evolving field of computer vision and artificial intelligence, since it is relevant to many different applications, including healthcare, sports analysis, and autonomous vehicles. In this section, we further extend and evaluate the high resolution pose estimation approach developed in OpenDR, as initially introduced in D3.3. First, we provide a brief description of the updated proposed methodology and dataset creation process, followed by an experimental evaluation on an updated and more challenging setup.

### 2.3.2 Description of work performed so far

In D3.3 we provided a novel methodology for high resolution pose estimation that marries efficiency with accuracy, allowing a pose estimator to process high resolution images without sacrificing computational speed or pose precision. The main concept of this approach is to process an input image or frame in lower resolution to create a rough heatmap with the human figures. Then, using the positional information of that heatmap, we can focus on certain regions where humans probably appear, in order to further analyze them and detect the poses. In this section, we further extend the proposed method by including an additional step in the inference procedure that can manage to separate the areas of interest based on the extracted heatmap. The updated methodology is trying to eliminate the parts of the image that are between a number of people, which does not carry any useful information for pose estimation, in order to further accelerate the active perception process and enable processing high resolution images faster. According to our prior research, as also shown in Figure 4, we use the extracted heatmap to find the minimum enclosing bounding box that contains the human figures without taking into account the surroundings. Following this procedure can bring up a new problem when humans are scattered inside the image. The new approach we propose is trying to separate the areas of interest and minimize the parts of the image without useful information. To this end, we propose a “divide and conquer” approach where we continuously divide the heatmap into parts and keep only those that contain non-zero values, i.e., contain regions with potential defections. This can

be trivially implemented by following a binary search-like procedure on each of the axes of the heatmap. An example of this process is provided in Figure 5. Therefore, the proposed idea is following a simple but effective technique, to continuously divide the heatmap into parts until the enclosing bounding boxes contain only the useful parts of the heatmap. The newly updated methodology solves the problem of taking as input parts of images that include background information which is not useful in pose estimation. Since our methodology is based on the heatmaps produced from the first pass, the proposed method uses those heatmaps for cropping the areas of interest from the original image. The outcome is to apply the pose estimation model only on the newly cropped parts which we expect to accelerate the inference procedure and improve the average precision results. An example of this process is provided in Figure 6.

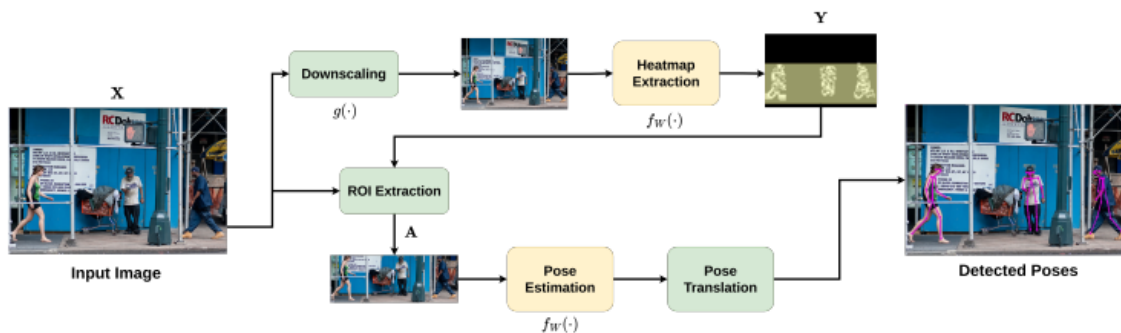


Figure 4: Active Perception for High Resolution Pose Estimation: The proposed approach replaces the ROI extractor with an advanced “divide and conquer” approach that can enable more accurate extraction of multiple smaller ROIs for analysis.



Figure 5: An example of applying the proposed ROI extraction approach. Note that instead of extracting one global ROI (red bounding box) - as done in the previous approach described in D3.3, we extract two smaller ROIs (cyan bounding boxes). This enables processing the input image faster since a smaller portion of it needs to be analyzed in high resolution.



### 2.3.3 Performance evaluation

Continuing the previous work in pose estimation, significant progress has been made in several key aspects of our work, each contributing to a deeper understanding and refinement of our active perception methodology for high resolution pose estimation. To this end, we have developed an enriched evaluation dataset, that is meticulously curated to encompass various scenarios, environments, and human poses, ensuring our methodology is rigorously tested under various conditions. A sample of the dataset is shown in Figure 6. This dataset was created by including two original images (selected from the COCO dataset) into the compiled high resolution frame, as well as by including a more complicated background. Furthermore, we have also combined the proposed approach with the DEKR [17] pose estimator, further highlighting the flexibility of the proposed method and its ability to work with other detectors, apart from OpenPose, to meet the needs of different applications.



Figure 6: A sample from the updated dataset used to evaluate the proposed high resolution pose estimation methodology.

The experimental evaluation is provided in Table 4. The following results show the application of the naive proposed method on the enhanced dataset, including results both on the lightweight OpenPose (LwOP) as well as on DEKR. The proposed method increases the average precision and the FPS of the inference both in the lightweight OpenPose and DEKR pose estimators in this more challenging dataset. The following experimental setups vary significantly in image resolution that the network takes as input, a critical parameter that plays a pivotal role in determining the performance and accuracy of our pose estimation algorithms. The first setup (Setup 1) uses two resizing steps, the first one is setting the image height to 360 pixels and the second to 512 pixels. The second experimental setup (Setup 2) resizes the input image in 512 pixels in the first pass and again in 512 pixels during the second pass.

Table 4: High Resolution Pose Estimation Evaluation

Method	Pose Estimation model	Average precision	FPS	Input resolution
Baseline	LwOP	0.165	32	$368 \times 368$
		0.371	2.8	Raw HR
Proposed	LwOP	0.384	29	Setup 1
		0.355	45	Setup 2
Baseline	DEKR	0.418	8	$512 \times 512$
		0.618	1.5	Raw HR
Proposed	DEKR	0.538	8.5	Setup 1
		0.517	9	Setup 2

## 2.4 Deep Reinforcement Learning for Active Perception

### 2.4.1 Introduction

Face recognition has been a major area of research in the field of computer vision and deep learning, with great success in various applications. However, despite the advances in face recognition, it still lacks the integration with robotic systems for real-world deployment. One of the biggest challenges in this integration is the issue of active perception. In real-world scenarios, the robot’s observation is limited and it needs to actively control its viewpoint to accurately recognize faces.

The problem of implementing active perception for face recognition can be addressed by using Reinforcement Learning (RL) as it provides a framework for learning control policies for autonomous systems. However, training RL agents for active perception tasks remains a challenging problem, mainly due to the difficulty in defining appropriate reward functions, data efficiency of RL algorithms, as well as, the sensitivity of RL algorithms on hyperparameter choices. To better understand challenges in the field, AUTH also conducted a review over existing active perception works (provided in Appendix 8.4).

In this section, we propose a Deep Reinforcement Learning (DRL) methodology for active perception, specifically for active face recognition, that allows DRL agents to be trained incrementally to solve the given task. More specific, we propose a curriculum-based learning method. As a curriculum, we define a structured and organized plan that outlines the learning objectives. A curriculum serves as a blueprint for what the agent should learn, how it should learn it and how it’s learning will be evaluated.

### 2.4.2 Description of work performed so far

In this section we describe the proposed method that has been developed so far. Our methodology is a curriculum based Deep Reinforcement Learning method, which decomposes the task of controlling a drone to correctly recognize a person in a room, into several subtasks and uses reward-based early stopping to advance to the next subtask. In order to solve the problem using DRL, we created an appropriate environment for our agents to interact with. The task to be solved is to maximize the confidence of a face recognition algorithm, when recognizing a human in a room. The confidence  $c$  of the face recognition algorithm is defined as:

$$c = 1 - \frac{\|\mathbf{y} - \mathbf{y}_l\|_2}{a}, \quad (8)$$

where  $\mathbf{y}$  is the embedding of the face image, and  $\mathbf{y}_l$  the embedding of the face matched in our database, following the embedding based face recognition approach outlined in the previous deliverables. Finally, we used a widely known DRL algorithm, PPO, to train our DRL agents.

Assuming  $\mathcal{T}$  as a single-task curriculum, the task  $m_{trg}$  describes the task of correctly recognizing a human subject in a room. While solving for  $m_{trg}$  the training of our agent is slow and often unstable. So, we can decompose  $m_{trg}$  to the following tasks  $m_i$  which we assume as a sequence curriculum, a curriculum where the agent needs to learn one task before moving to the next in a particular order:

- $m_1$  - center a human subject in the field of view.
- $m_2$  - close the distance to the human subject, while keeping the human subject centered in the field of view.
- $m_3$  - position accordingly to have the best view of the human subject.
- $m_4$  - correctly recognize the human subject.

To formulate the reward of our agent we use three unit vectors named  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{c}$ , where  $\mathbf{a}$  is the vector starting from the drone with direction towards the human model,  $\mathbf{b}$  the vector starting from the drone with direction towards where the drone's camera is facing and  $\mathbf{c}$  the vector starting from the human model with direction towards where the human model is facing. We denote the angle  $\angle(\overrightarrow{AB})$  as  $a$  and the angle  $\angle(\overrightarrow{BC})$  as  $b$ . We denote as  $h$  the current height of the drone in respect to a predefined target height  $h_{trg}$  and as  $d$  the distance between the drone and a desired distance to the human subject  $d_{trg}$ . As  $c \in [0, 1]$  we denote the confidence with which the face recognition system recognizes the human model in the scene.

We formulate the individual rewards  $r_i(t)$  at each episode timestep that correspond to each sub task as:

$$r_1(t) = \begin{cases} +j & , a_{t+1} - a_t > 0 \\ 0 & , a_{t+1} - a_t = 0 \\ -j & , a_{t+1} - a_t < 0 \end{cases} \quad (9)$$

$$r_2(t) = \begin{cases} +j & , d_{t+1} - d_t > 0 \\ 0 & , d_{t+1} - d_t = 0 \\ -j & , d_{t+1} - d_t < 0 \end{cases} \quad (10)$$

$$r_3(t) = \begin{cases} +j & , h_{t+1} - h_t > 0 \\ 0 & , h_{t+1} - h_t = 0 \\ -j & , h_{t+1} - h_t < 0 \end{cases} \quad (11)$$

$$r_4(t) = \begin{cases} +j & , b_{t+1} - b_t > 0 \\ 0 & , b_{t+1} - b_t = 0 \\ -j & , b_{t+1} - b_t < 0 \end{cases} \quad (12)$$

$$r_4(t) = \begin{cases} +j & , b_{t+1} - b_t > 0 \\ 0 & , b_{t+1} - b_t = 0, \\ -j & , b_{t+1} - b_t < 0 \end{cases} \quad (13)$$

where  $j$  is a small scalar reward.

When solving for  $m_{trg}$  the reward at each timestep  $t$  is given by:

$$r_{trg}(t) = r_1(t) + r_2(t) + r_3(t) + r_4(t) \quad (14)$$

When we decompose  $m_{trg}$  to a sequence curriculum  $C$  the reward at each timestep is formulated as:

$$r_{m_1}(t) = r_1(t) \quad (15)$$

$$r_{m_2}(t) = r_1(t) + r_2(t) \quad (16)$$

$$r_{m_3}(t) = r_1(t) + r_2(t) + r_3(t) \quad (17)$$

$$r_{m_4}(t) = r_1(t) + r_2(t) + r_3(t) + r_4(t) \quad (18)$$

Additionally, there is a minor penalty given to the agent, each time it decides to run the face recognition module, and the confidence is  $c < 0.5$ .

Aside from the reward decomposition, different terminal states and the corresponding terminal rewards need to be defined when solving for  $m_{trg}$  or for each task  $m_i$ .

When solving for  $m_{trg}$  the terminal state  $s_{T_{m_{trg}}}$  is defined as:

$$s_{T_{m_{trg}}} = c > c_{trg} \quad (19)$$

and the corresponding reward for reaching  $s_T$  is:

$$r_{m_{trg}}(s_T) = 4x + zc, \quad (20)$$

where  $x$  and  $z$  are constants used to scale the reward on a terminal state. When we decompose  $m_{trg}$  to a sequence curriculum  $C$  the terminal states  $s_{T_{m_i}}$  are defined as:

$$s_{T_{m_1}} = a_t < a_{trg} \quad (21)$$

$$s_{T_{m_2}} = a_t < a_{trg}, d_t = d_{trg} \quad (22)$$

$$s_{T_{m_3}} = a_t < a_{trg}, d_t = d_{trg}, h_t = h_{trg} \quad (23)$$

$$s_{T_{m_4}} = a_t < a_{trg}, b_t < b_{trg}, d_t = d_{trg}, h_t = h_{trg} \quad (24)$$

$$s_{T_{final}} = c > c_{trg} \quad (25)$$

and the corresponding rewards for reaching  $s_{T_{m_i}}$  are:

$$r(s_{T_{m_1}}) = x \quad (26)$$

$$r(s_{T_{m_2}}) = 2x \quad (27)$$

$$r(s_{T_{m_3}}) = 3x \quad (28)$$

$$r(s_{T_{m_4}}) = 4x \quad (29)$$

$$r(s_{T_{m_{final}}}) = 4x + zc \quad (30)$$

Additionally, we terminate an episode whenever the drone hits an object, or gets too close to the human model and penalize the agent with  $r(s_T) = -1$ .

A question that arises in the proposed curriculum-based approach, is the following. Based on which criteria should the agent advance on the next part of the curriculum. The trivial method would be to train on each subtask for a fixed amount of time  $f_t$ . But finding the correct amount

of time needed for each subtask is not trivial at all. The proposed method introduces a module that advances the curriculum, when the agent consistently solves a subtask and is described below: We evaluate the current learned policy every  $k$  timesteps. Let  $\bar{r}$  be the mean reward of the learned policy  $p$  over  $n$  evaluations.

$$\bar{r} = \frac{\sum_{i=1}^n r_i}{n} \quad (31)$$

, where  $r_i$  the total reward the agent received during an evaluation episode when solving for task  $m_i$ . We calculate the moving average  $ma$  of  $\bar{r}$  with a window of  $w$ :

$$ma(nk) = \frac{\sum_{i=nk-w+1}^{nk} \bar{r}_i}{w} \quad (32)$$

We introduce  $p$ , a patience counter, which we increase if  $\bar{r} \leq ma$  and reset it if  $\bar{r} > ma$ . If  $p = l$ , where  $l$  is a predefined patience limit, we stop training on task  $m_i$  and start training on task  $m_{i+1}$ .

---

**Algorithm 1** Proposed method for curriculum advancement
 

---

```

1:  $p = 0$ 
2: for  $iteration = 1, 2, \dots$  do
3:   Evaluate learned policy  $\pi_\theta$  for  $n$  evaluation episodes and calculate mean reward  $\bar{r}$ 
4:   Calculate the moving average of  $\bar{r}$  with window  $w$ .
5:   if  $\bar{r} \leq ma$  then
6:      $p+ = 1$ 
7:   else
8:      $p = 0$ 
9:   end if
10: end for

```

---

### 2.4.3 Performance evaluation

In this section, we introduce the design of the virtual environment with which the agent interacts, as well as, the agent and training configuration. We wrap our environment as a Gym environment, OpenAI’s toolkit for research on RL algorithms. Gym already provides a vast collection of environments, none of which were suitable for our task.

We built our environment in Webots, an open source robot simulator. We constructed a square room containing numerous different objects. We also built 40 different human models using MakeHuman, an open source program to create realistic 3D human models. At each episode one of the human models is chosen and placed in the room randomly. The agent controls a MavicPro2 camera mounted drone, which flies freely inside the room. The created environment can be seen in Figure7.

We wrap our environment as a Gym-like environment, in order to be compatible with most RL frameworks. A Gym-like environment exposes two critical methods, in order for an agent to be able to interact with it. These are the reset and step methods. The reset method will initialize the world and return the initial state of the world ( $s_0$ ) to the agent (e.g. location and orientation of every object inside the room, including the human model and the drone). The step method will translate the action the agent executes in the environment, will move the world

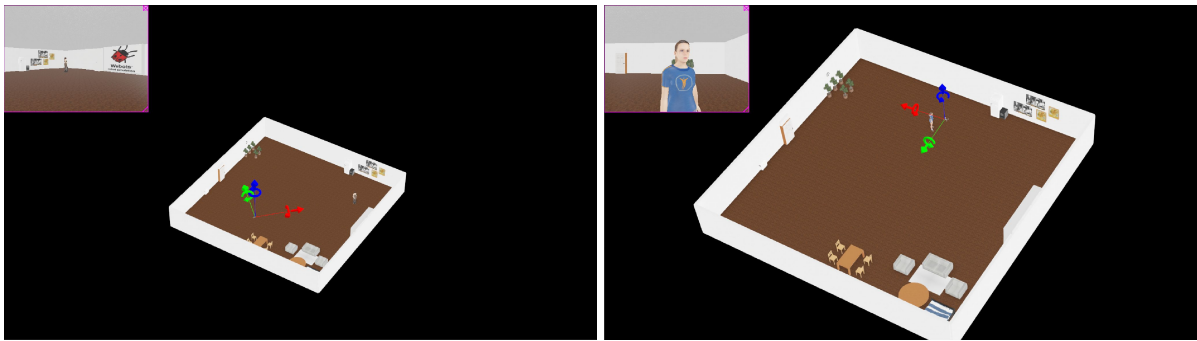


Figure 7: Created environment for active Face Recognition

and all objects within it forward by one timestep and will return the next state of the world, the reward for executing that action and a flag, indicating whether or not an episode has come to an end or not.

Next we describe the agent configuration. Let  $\mathbf{x} \in \mathbb{R}^{W \times H \times C}$  be the image captured by the drone at each timestep, where  $W$ ,  $H$  and  $C$  are the width, height and number of channels in the corresponding image. To train our agent we capture images of  $W = 400$ ,  $H = 300$  and  $C = 3$ . We feed this image to an actor-critic architecture, utilizing a Deep Convolutional Network. Specifically we employ the following lightweight architecture:

- A shared network containing three convolutional layers with 32, 64 and 64 filters respectively and a linear layer which outputs a 512-wide embedding.
- A linear layer with 512 output, and a linear layer with an output of as many neurons as the action space, each corresponding to one action. This layer outputs normalized probabilities for each action, given an input image  $\mathbf{x} \in \mathbb{R}^{W \times H \times C}$ .
- A linear layer with 512 output, and a linear layer with an output of 1, which is responsible to learn the advantage function.

As mentioned before, the critic network is responsible to output normalized probabilities for each action given an input image. In our experiments the available actions are discrete actions and are described as:

- Do nothing.
- Move forward for 10cm.
- Move backwards for 10cm.
- Move left for 10cm.
- Move right for 10cm.
- Rotate the drone clockwise for  $3^\circ$ .
- Rotate the drone counter-clockwise  $3^\circ$ .
- Deploy Face Detection and Recognition module
- Move upwards for 5cm.

- Move downwards for 5cm.

The action 'Deploy Face Detection and Recognition module' deploys a face recognition pipeline which consists of two parts: a face detection and alignment module and a face recognition module. In our experiments, we used RetinaFace as the face detection and alignment module and MobileFaceNet trained with the Arcface loss function as the face recognition module. When this action is executed, an image captured by the drone is processed by RetinaFace, which locates and crops the image around any detected faces. The cropped image is then fed into the face recognition module, which produces a feature vector for the face. This feature vector is compared to feature vectors of known identities, and the Euclidean distance between the input face and each known identity is calculated. If the distance between the input face and a known identity is below a predefined threshold, the identity is considered a match. The pipeline used was provided by OpenDR.

Our agent was trained using the Proximal Policy Optimization (PPO) algorithm. During our experiments, we used the following hyperparameters:

- `n_steps` = 6400
- `learning_rate` = 0.003
- `gamma` = 0.9
- `batch_size` = 64
- `target_kl` = 0.4
- `total_timesteps` = 5000000

We trained five agents using three different training approaches: a single-task curriculum, a sequence curriculum with fixed training timesteps ( $t_f = 250000$ ), and a sequence curriculum that advanced in next subtasks when convergence on the previous subtask was achieved. The training curves can be seen in Figure8 and Figure9. Additionally in Table.5 we demonstrate the average timesteps of the 5 agents needed in order to start solving the task consistently.

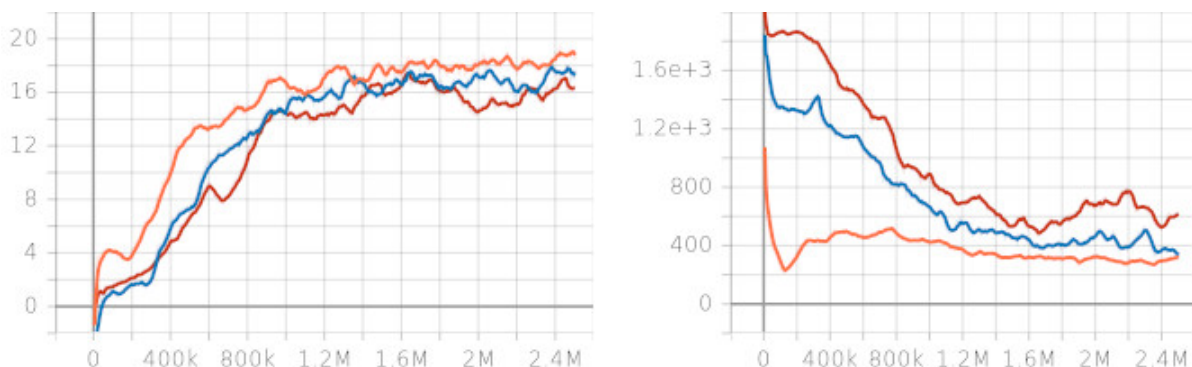


Figure 8: Mean reward and episode timesteps of 5 agents achieved during training. Orange: the proposed method, Blue: sequence curriculum with fixed training timesteps and Red: single-task curriculum

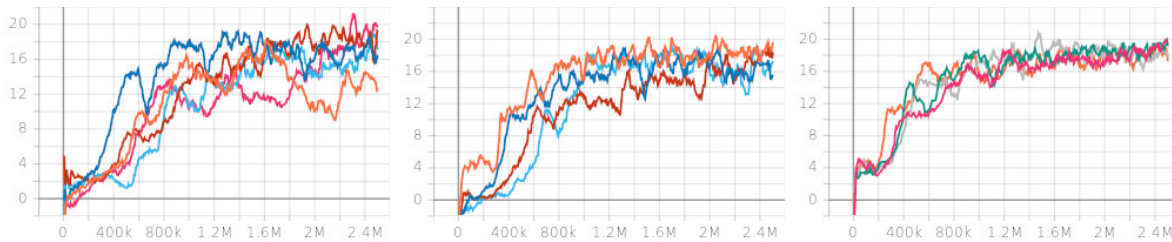


Figure 9: Episode reward of 5 agents achieved during training.

Left: single-task curriculum, Middle: sequence curriculum with fixed training timesteps and Right: the proposed method

Table 5: Active Face Recognition Reinforcement Learning Policies Evaluation

Method	Average Training steps for task solution
Single Task Curriculum	3.2M
Fixed-Timesteps Curriculum	2.4M
Proposed Curriculum	1.6M

## 2.5 Neural Attention driven Non-Maximum Suppression for Person Detection

### 2.5.1 Introduction, objectives and work performed so far

AUTH finalized the work conducted on Non-Maximum Suppression (NMS) for person detection. NMS is a post-processing step incorporated in almost every visual object detection framework, where detected rectangular Regions-of-Interest (RoIs) that spatially overlap are merged/filtered. The problem it attempts to solve arises from the tendency of many detectors to output multiple, neighbouring candidate object RoIs for a single given visible object, due to their implicit sliding-window nature. NMS methods typically rescore the raw candidate detections/RoIs outputted by the detector, before thresholding these modified scores so that, ideally, only a single RoI is finally retained for each visible object. Due to these limitations of traditional algorithms [12] [7], modern Deep Neural Network (DNN)-based methods [22] [25] for performing NMS have emerged during the past few years. Compared to the corresponding literature, AUTH proposed within OpenDR a novel NMS method, specifically for the person detection task, which offers: (i) a novel reformulation of the NMS task for object detection as a sequence-to-sequence problem, (ii) a novel deep neural architecture for NMS, relying on the Scaled Dot-Product Attention mechanism, called *Seq2Seq-NMS* and (iii) a new, fast, efficient and GPU-based neural implementation of the low-level Frame Moments Descriptor (FMoD) [38], which is employed for feeding the proposed DNN with appearance-based representations of detected candidate RoIs. An extensive quantitative evaluation using well-known metrics and public person detection datasets indicated favourable results in comparison to several competing NMS methods, both neural and non-neural. A version of this work was presented in [58] as well as in Section 2.3 on D3.2 (M24). A paper describing the final version of the corresponding method and the respective experiments was accepted and published during this period in IEEE Transactions on Image Processing. The paper can be found in Appendix 8.2:

- Charalampos Symeonidis, Ioannis Mademlis, Ioannis Pitas and Nikos Nikolaidis, "Neural Attention-Driven Non-Maximum Suppression for Person Detection," in IEEE Trans-



actions on Image Processing, vol. 32, pp. 2454-2467, 2023.

The overall work on *Seq2Seq-NMS* has been integrated in the OpenDR toolkit and is available at: [https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object\\_detection\\_2d/nms/seq2seq\\_nms](https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/nms/seq2seq_nms).

Moreover, in the previous period, AUTH proposed a new variant of Seq2Seq-NMS, which was named FSeq<sup>2</sup>-NMS. The proposed variant is able to harness the information-rich intermediate feature maps of DL-based object detectors. These intermediate feature maps are used to derive learned, high-level, semantically meaningful RoI representations, replacing the implemented version of [38] in the pipeline of Seq2Seq-NMS. Compared to Seq2Seq-NMS, the inference time of proposed variant is relatively shorter, since the corresponding RoI representations are being extracted during the detector's inference step. In addition, FSeq<sup>2</sup>-NMS can be easily plugged on top of any DL-based detector, and trained as a separate sub-module. Experiments conducted on two public person detection datasets, widely used for detecting humans in crowded scenes, confirmed that FSeq<sup>2</sup>-NMS is highly suitable for this scenario, achieving top accuracy.

The preliminary version of this work was presented in Section 2.5 of D3.3 (M36). A paper describing the final version of the corresponding method was accepted and presented during this period in ICASSP 2023:

- Charalampos Symeonidis, Ioannis Mademlis, Ioannis Pitas and Nikos Nikolaidis, "Efficient Feature Extraction for Non-Maximum Suppression in Visual Person Detection," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1-5, 2023.

Finally, during this period, the overall work on FSeq<sup>2</sup>-NMS has been integrated in the OpenDR toolkit and it is available at: [https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object\\_detection\\_2d/nms/fseq2\\_nms](https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/nms/fseq2_nms).

## 3 Deep person/face/body part tracking, human activity recognition

### 3.1 Variational Spatio-Temporal Graph Convolutional Networks for Skeleton-based Action Recognition

#### 3.1.1 Introduction

Skeleton-based human action recognition is the task of classifying human actions based on a person's poses. A human skeleton represents the key points of the human body, such as eyes, neck, feet, palms, knees, etc. These points have always the same adjacency structure which follows the structure of the body and can be reliably estimated by using tools such as OpenPose [8,49]. However, standard CNN-based approaches that work well on images and videos cannot be directly applied to a series of skeletons due to their irregular structure. For this reason, some skeleton-based methods use RNNs [33,42,69] to process sequential skeleton data or produce a pseudo-image by rearranging skeleton joints and processing it with CNNs [29,34,37]. To further improve the use of the domain structure, Graph Convolutional Networks (GCN) were proposed for skeleton-based human action recognition [20]. The Spatio-Temporal GCN (ST-GCN) [66]

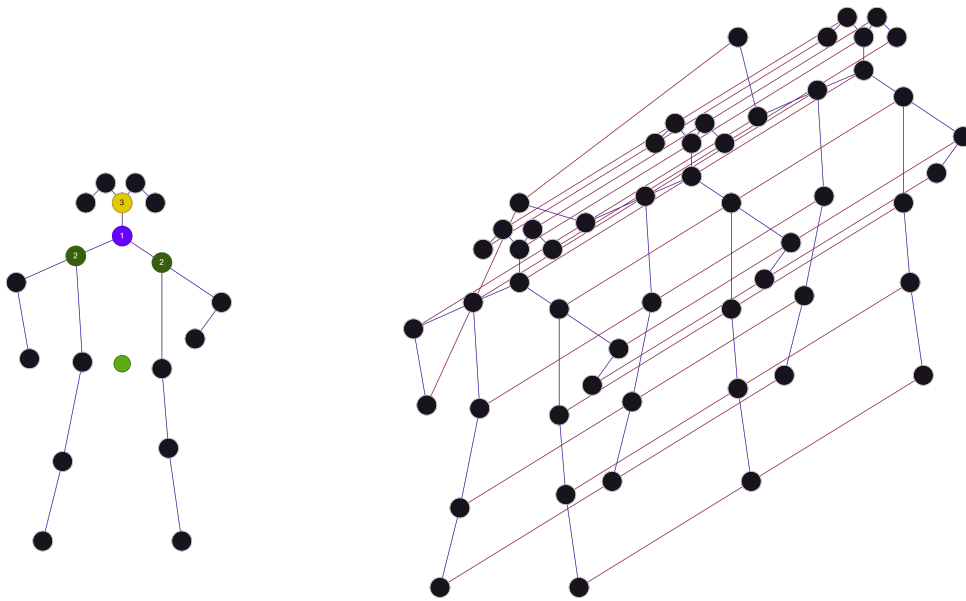


Figure 10: A single skeleton graph (left) with a center of mass represented by a green dot, and a temporal set of 3 skeletons (right) with purple spatial connections and magenta temporal connections.

pioneered this approach for skeleton-based action recognition by applying graph convolution to each skeleton frame and combining the results in the temporal dimension by following the (2+1)D convolution approach [60] with 2D spatial convolutions and 1D temporal convolutions to achieve pseudo-3D convolution. AGCN [57] diverges from the predefined graph structure by learning an additional adjacency matrix that represents connection between joints, which do not exist in the human body directly, but are important for an action analysis. Additionally, AGCN uses 2 streams of features by encoding a second-order feature set as a graph of joint connection vectors, which improves the performance of the model.

Applications of human action recognition in robotics include an important health assistance field, where a robot follows or observes a human and tries to identify if the human needs assistance from the robot or from health authorities. For this task, the accuracy and certainty in perception is critical, but usually the aleatoric uncertainty predicted by classical networks offers poor estimation of the actual model uncertainty, which should include both aleatoric and epistemic uncertainties. The accurate estimation of uncertainties can be used to determine whether the robot should perform an action based on the perception, or should it not rely on the perception and perform an action aimed at improving the certainty in perception, which may include coming closer to the person or changing the viewing angle.

### 3.1.2 Summary of the state of the art

Spatio-temporal graph convolutional networks [52, 57, 66] follow the (2+1)D convolution approach [60] where the spatial 2D convolution is implemented as a GCN or a transformer network and a temporal convolution is used to combine spatial features in the temporal dimension. The input skeleton sequence is stored as a 3D tensor  $S \in \mathbb{R}^{C \times T \times N}$ , where  $C$  is the number of channels for each joint,  $T$  is the number of frames and  $N$  represents the number of joints in each skeleton. The structure of a skeleton is stored in an adjacency matrix  $A \in \mathbb{R}^{N \times N}$ , which

represents connections between joints in a binary mode. Both ST-GCN [66] and AGCN [57] analyze graphs on 3 different levels. An adjacency matrix is defined for three different neighboring partitions. These partitions are presented in Fig. 10. For each node in the skeleton graph, the first partition includes the root node itself (1), the second partition contains the nodes that are closer to the center of mass (2) and the last partition contains all other neighbors (3). Each of the resulting adjacency matrices  $A_p, p \in 1, 2, 3$  are normalized as follows:

$$\hat{A}_p = D_p^{-0.5} A_p D_p^{-0.5}, \quad (33)$$

where  $D_p$  represents for a degree matrix for a partition  $p$ . Graph convolution  $\Gamma(\cdot)$  is a function that takes features of the previous layer or input features and transforms as follows:

$$\Gamma(S^i) = \rho(\Xi(S^i) + \text{BN}(\sum_p (\hat{A}_p \circ M_p) S^i W_p)), \quad (34)$$

where  $S^i$  is the current layer features,  $\rho$  is a ReLU activation function, BN is batch normalization,  $\circ$  is an element-wise multiplication,  $W_p$  is a learnable matrix that transforms features,  $M_p$  is a learnable attention matrix, and  $\Xi(\cdot)$  is a residual transformation function that linearly transforms the features of a previous layer to the same shape as the current layer's features:

$$\Xi(S^i) = \begin{cases} S^i, & \text{if } C^i = C^{i+1}, \\ S^i W_\xi, & \text{otherwise.} \end{cases} \quad (35)$$

Here  $W_\xi$  is a learnable transformation matrix that ensures the shape of the residual tensor is the same as the output of the other part of the layer and  $C^i$  and  $C^{i+1}$  denote the number of channels of the input and the output layers, respectively. The outputs of each GCN layer are followed by temporal 1D convolution to combine data from different time frames and the combination of batch normalization and residual connection, similarly to the GCN. AGCN [57] follows the same structure of layers as in GCN, but instead of using element-wise augmentation of the adjacency matrix, it adds an attention matrix that is designed to learn similarities between joints and provide an additional adjacency information based on the training data.

Gawlikowski et al. [16] classifies uncertainty estimation methods into four main categories, namely Single Deterministic Networks [56, 72], which either regress uncertainties using a separate model branch or analyze the performance of the parts of the model to estimate uncertainty, Bayesian Neural Networks (BNNs) [5, 39, 73], which consider a distribution over the model's weights and compute disparities between different samples of the resulting stochastic model to estimate uncertainty, Ensemble Methods [43, 47, 61], which consider a set of networks that are trained to achieve a set of weights with a Categorical distribution assigned to sample from them, and Test-Time Data Augmentation methods [26, 62, 63], which use a single network but apply different augmentation techniques to compute the difference between predictions on augmented inputs. Similarly to BNNs, Variational Neural Networks [45, 46] consider a distribution for a network, but place it not on the weights, but on the outputs of each layer. The parameters of the layer-wise Gaussian distributions are computed as outputs of the corresponding sub-layers.

Uncertainty quality estimation is a complex task due to the lack of ground-truth data for uncertainties in real-world data. For this reason, Epistemic Neural Networks (ENNs) [48] propose a framework to estimate the quality of uncertainty of different uncertainty estimation methods, which shows that the popular Monte Carlo Dropout (MCD) [13] and Bayes By Backprop [5] methods are outperformed by VNNs [46], Hypermodels [11], Deep Ensembles [47] and Layer

Ensembles [43]. The use of uncertainties for action recognition is explored in different ways. Guo et al. [19] propose a probabilistic transformer architecture based on combining query and key values to generate mean and variance values of a Gaussian distribution with an MLP layer. Zhao et al. [73] use a GCN + LSTM network for skeleton-based action recognition and introduce a Bayesian Neural Network to it by treating parameters of LSTM as random variables. A stochastic gradient Hamiltonian Monte Carlo method [10] is used to train the model. Zhang et al. [72] use a deterministic approach to estimate uncertainties for skeleton-based action recognition and utilize these uncertainties to lead an active-learning training.

### 3.1.3 Description of work performed so far

We propose Variational Spatio-Temporal Graph Convolutional Networks (VSTGCNs) to estimate and utilize uncertainty for skeleton-based human action recognition by combining Variational Neural Networks [46] with state-of-the art Spatio-Temporal Graph Convolutional Networks [57, 66]. Such an approach allows for a straightforward extension of uncertainty estimation to future improvements in the problem of skeleton-based human action recognition. We implement variational versions of ST-GCN [66] and AGCN [57] by replacing graph and temporal convolutions with variational versions of it. The structure of a layer in VSTGCN is presented in Fig. 11. The input spatio-temporal feature set is processed by independent graph convolutions  $\Gamma_\mu(\cdot)$  and  $\Gamma_\sigma(\cdot)$ , which process the same input, but are responsible for creating means and variances of a Gaussian distribution, respectively. Spatial features are sampled from the generated Gaussian distribution and are then used as inputs to a variational temporal convolution, which, following the same approach, generates stochastic outputs of the current VSTGCN layer. The variance in outputs of the last layer is the predicted model uncertainty. For VAGCN, we propose two models. The first model applies the same idea for VAGCN as for VSTGCN and replaces all layers inside AGCN by variational versions of it. The second model utilizes inner uncertainties by applying the learned similarity matrix not to the features of the previous layer, but to the corresponding variances, as follows:

$$\begin{aligned}\Gamma^V(S^i) &= \rho(\Xi(S^i) + \text{BN}(\sum_p (\hat{A}_p + M_p^V) S^i W_p)), \\ M_p^V &= B_p + C_p^V, \\ C_p^V &= \text{softmax}(\mathbb{V}[S^i]^T W_{\theta_p}^T W_{\phi_p} \mathbb{V}[S^i]),\end{aligned}\tag{36}$$

where  $\Gamma^V(\cdot)$  is the uncertainty-aware version of AGCN's graph convolution,  $M_p^V$  denotes the data-driven adjustment in adjacency matrix, which consists of a learned  $B_p$  adjacency matrix and an uncertainty-aware similarity matrix, applied to the variance in the input features  $\mathbb{V}[S^i]$ , instead of the features directly. This approach requires an aggregation of samples before the end of the network, and therefore it is practical to implement only a few evenly-spread layers this way and keep regular variational layers in-between.

Training of the variational models can be performed similarly to the original models, with the difference that we can select the number of samples used during training for variational models, which can impact the final model performance. For inference, we can determine the number of samples used to balance between inference speed and the quality of uncertainty. Following [44], we can initialize VSTGCNs with a corresponding classical model for mean weights and use small initial weights for variances to improve the accuracy of trained models.

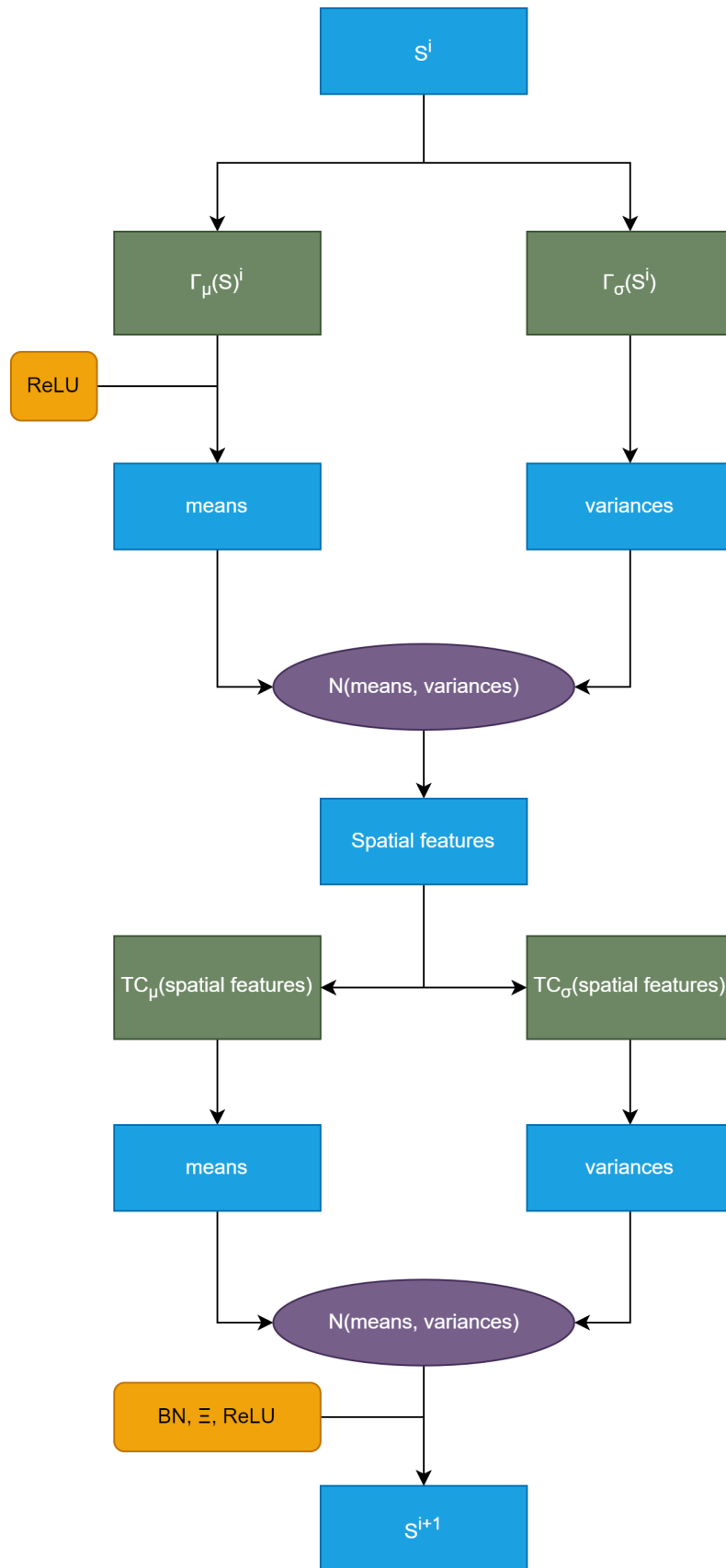


Figure 11: Structure of a single block in Variational ST-GCN.

Table 6: Evaluation of skeleton-based action recognition methods on the cross-view subset of NTU-RGBD-60 dataset. Results with an asterisk in the model name are taken from the corresponding papers.

Model	Samples		Accuracy
	Training	Inference	X-View (%)
Bayesian GC-LSTM* [73]	-	-	89.0
ST-GCN [66]	-	-	92.89
VSTGCN	1	10	93.03
IVSTGCN	2	10	<b>93.42</b>

Table 7: Evaluation of skeleton-based action recognition methods on the cross-subject subset of NTU-RGBD-60 dataset. Results with an asterisk in the model name are taken from the corresponding papers.

Model	Samples		Accuracy
	Training	Inference	X-Subject (%)
Bayesian GC-LSTM* [73]	-	-	81.8
ST-GCN [66]	-	-	<b>86.4</b>
VSTGCN	2	2	86.16
IVSTGCN	1	20	85.54
AGCN [57]	-	-	<b>86.27</b>
VAGCN	1	10	85.78
IVAGCN	1	10	86.03
IU-VAGCN	1	30	85.22

### 3.1.4 Performance evaluation

We performed experiments on large scale NTU-RGBD [32] and Kinetics [28] datasets. For NTU-RGBD, we use 120- and 60-class subsets and train on both cross-view and cross-subject scenarios.

We train original STGCN and AGCN models for 50 epoch on NTU-RGBD with batch size 64 and for 65 epochs with batch size 128 on Kinetics. We train VSTGCN and VAGCN models from scratch with the same parameters and also train IVSTGCN and IVAGCN models by initializing means of VSTGCN and VAGCN networks with a pretrained STGCN and AGCN models, respectively, and variances with Xavier uniform initialization. Then we train IVSTGCN and IVAGCN with 0.05 learning rate, batch size 64, 30 epochs with learning rate step at epochs 10, 15, 20. For Variational AGCN models, we create two versions. The first version, namely VAGCN and IVAGCN, uses the same uncertainty approach as in VSTGCN, and the second version, namely IU-VAGCN and IU-IVAGCN, utilizes the inner model uncertainties by placing uncertainty-aware GCN layers at different positions of the model. We select the number of samples for variational networks during training in range  $[1, \dots, 2]$  and during inference in range  $[1, \dots, 40]$ .

Table 6 presents evaluation results of the baseline ST-GCN model, the Bayesian GC-LSTM [73] model that utilizes uncertainty estimation, and the proposed variational versions of the baseline model on the joint data of a cross-view subset for 60-class version of NTU-RGBD and Table 7 presents the evaluation of the methods on the cross-set subset. On top of providing output uncertainty, variational STGCN models outperform STGCN on the cross-view subset, with IVSTGCN providing a bigger improvement in accuracy. On the cross-subject subset, the variational models are not outperforming the original models in terms of accuracy, with the initialized model providing an improvement compared to the regular variational model for AGCN, but not for STGCN. This leads to a conclusion that, depending on the dataset, a more accurate selection of training parameters is required to achieve an improvement in the model accuracy on top of the provided uncertainties, which is a direction for future work. The proposed models outperform an existing method for uncertainty estimation in skeleton-based called Bayesian GC-LSTM.

## **4 Social signal (facial expression, gesture, posture, etc.) analysis and recognition**

### **4.1 RGB Hand Detection and Gesture Recognition**

#### **4.1.1 Introduction**

Hand gestures have emerged as a prominent modality for facilitating communication between humans and robots, driving various contemporary human-robot collaborative applications. The efficacy of this approach can be attributed to several key factors. Firstly, hand gesture-based communication mitigates the need for humans to maintain close physical proximity to robots while conveying commands. This feature is especially advantageous in high-risk industrial settings, where safety concerns dictate the need for distant interactions, and in the context of assistive robots that have a requirement of remaining mobile. Secondly, hand gestures exhibit universality across diverse environments (in contrast to e.g. speech and language-based communication), which can be highly context-dependent and culturally bound. The universality of hand gestures ensures greater accessibility and comprehensibility in human-robot interactions. Moreover, hand gestures can serve as an additional non-verbal modality for interaction, enriching other communication channels, such as speech. This multimodal approach has a potential to improve the overall communication experience, enhancing the depth and clarity of conveyed intentions and emotions between humans and robots.

Automating hand gesture recognition with machine learning solutions has been a prominent topic in the recent years, and a multitude of approaches have been proposed for hand gesture recognition, each offering its own unique characteristics. To this end, we have worked towards extending OpenDR toolkit functionality in this regard.

#### **4.1.2 Summary of state of the art**

The existing automatic hand gesture recognition approaches proposed over the recent years vary across several factors, introducing a diverse range of possibilities. Firstly, the choice of modality for hand gesture recognition is crucial, whether it be RGB images, Depth images, or pre-extracted features like hand keypoints. Each modality has its advantages and drawbacks to consider. RGB sensors are readily available and easy to utilize, but their informativeness may

be limited. In contrast, Depth sensors or elaborate representations derived from RGB images (such as hand keypoints) can provide additional information, albeit with integration challenges and increased computational overhead. Additionally, utilization of various modalities simultaneously poses challenges related to the dataset creation, hence making it difficult to create sufficiently large datasets that would enable effective hand gesture recognition in-the-wild.

Furthermore, the nature of gestures themselves presents a dichotomy of static and dynamic representations, requiring different methodological approaches. Static gestures are captured as images, while dynamic gestures involve video sequences. This distinction influences the development of various techniques to address the gesture recognition problem. Additionally, the formulation of the gesture recognition problem varies across datasets and problem setups. Some methods treat it solely as an image or video classification task, while others incorporate localization components through bounding box detection or hand segmentation.

### 4.1.3 Description of the work performed so far

To enhance the functionality of the toolkit with respect to hand gesture recognition, TAU adopted the recent developments in the field of hand gesture recognition related to emergence of novel large-scale open datasets and have integrated a new RGB-based hand gesture recognition and localization tool. This tool excels in identifying and localizing 18 hand gestures, as shown in Figure 12, as well as ‘no gesture’ class, from RGB images.



Figure 12: Illustration of the gestures included in the Hagrid dataset

While a hand gesture recognition tool already exists within OpenDR toolkit, it poses certain limitations. Firstly, having formulated the task of hand gesture recognition as image classification, it lacks the localization components and is limited to a fixed number of hands (in this case, 1 or 2 depending on the gesture) present in the image simultaneously. In turn, we opt for integrating hand detection as part of the solution, enabling the development of more advanced use cases. In addition, the existing tool relies on multi-modal RGB and Depth inference, which offers benefits in terms of having higher recognition performance potential, but suffers from limitations in terms of availability of the large-scale in-the-wild multimodal datasets for the task. This leads to necessity of further effort for effective in-the-wild out of domain applications. On the other hand, unimodal RGB datasets are easier to obtain at scale, hence enabling unimodal models to provide sufficiently effective solutions to the hand gesture recognition task given that reasonably large datasets of images are available. Recently, HaGRID (HAND Gesture Recognition Image Dataset) [27] has been released, consisting of 500 000 hand gesture images collected in-the-wild from 30 000 unique people. We employ this dataset for development of OpenDR hand gesture recognition tools.



We opt for a lightweight Nanodet-based object detection model that we train on the large-scale Hagrid dataset, resulting in great speed vs performance ratio. Nanodet [55] is a lightweight anchor-free object detection model that is capable of achieving the performance suitable for deployment on edge devices, hence making it particularly beneficial for the robotics applications.

#### 4.1.4 Performance evaluation

We integrate a nanodet-m-1.5x model trained on Hagrid dataset and we compare its performance with the previously reported results in the literature for this dataset. To the best of our knowledge, the best reported result on this dataset is currently by Yolov7-tiny [27]. Table 8 shows the Mean Average Precision of several state-of-the-art object detectors, all of which underperform compared to our trained model that is integrated in the OpenDR toolkit. Note that the Nanodet model is also the lightest one compared to the others.

Table 8: Performance of different object detection models on Hagrid dataset.

Detector	mAP	AP_50	AP_75
SSDLiteMobileNetV3Large	0.7149	-	-
SSDLiteMobileNetV3Small	0.5338	-	-
FRCNNMobilenetV3LargeFPN	0.7805	-	-
Yolov7-tiny	0.8110	-	-
<b>nanodet-plus-m-1.5x_416 (integrated in OpenDR)</b>	<b>0.8259</b>	<b>0.9856</b>	<b>0.9544</b>

## 5 Deep speech and biosignals analysis and recognition

### 5.1 Integrating Whisper and Vosk in Speech Transcription

#### 5.1.1 Introduction and summary of state of the art

Recent advancements in speech recognition have been propelled by unsupervised pre-training techniques. For example, Wav2Vec 2.0 [4], [23] use self-supervised learning to learn speech representation followed by fine-tuning on transcribed speech. [65] combines unsupervised pre-training and pseudo-labeling to improve performance. [24] uses unlabeled target domain data during pre-training for fine-tuning out-of-domain data. These methods, which learn from raw audio without human labels, can utilize vast datasets of unlabeled speech, scaling up to 1,000,000 hours of training data [71]. However, while the encoder captures high-quality speech representations, they lack an equivalently performant decoder to transform these representations into usable outputs, necessitating a complex fine-tuning stage. This can lead to overfitting to specific dataset patterns and poor generalization to other datasets. The goal of a speech recognition system should operate reliably across diverse settings without the need for dataset-specific fine-tuning. Research demonstrated by [30], [41], and [9] have shown that systems pre-trained across multiple datasets are more robust and generalize more effectively to held-out datasets than models trained on a single source. But even combined datasets like SpeechStew [9], which contains 5,140 hours are tiny compared to vast amounts of unlabeled data used in previously mentioned 1,000,000 hours of unlabeled speech data utilized in [71].

Addressing the limitations of existing datasets, recent efforts have produced larger datasets for speech recognition by compromising on transcript quality. Researchers have proposed a transformer sequence-to-sequence model called Whisper [54], which scales weakly supervised speech recognition to 680,000 hours of labeled audio data. This approach eliminates the need for dataset-specific fine-tuning and expands to multilingual and multitask training, covering 96 different languages and including various speech processing tasks, including multilingual speech recognition, speech translation, spoken language identification, and voice activity detection.

Vosk [1] is an open-source speech recognition toolkit that provides offline, on-device voice recognition capabilities for a variety of platforms. Vosk [1] main features are supporting more than 20 languages and dialects, low-latency for small variants models, and providing a streaming API.

### 5.1.2 Description of the work performed so far

Both Whisper [54] and Vosk [1] come with open-source implementations and offer a diverse range of model variants suitable for offline, computation-constrained environments. Our integration of Whisper and Vosk into OpenDR takes advantage of these open-source offerings. The integration involves the creation of a learner class that encompasses basic functionalities such as downloading, loading, inference, and evaluation of models on speech datasets.

Whisper [54] offers transcription and translation functionality, but our integration is centered solely on its speech transcription capabilities and includes a streaming adaptation on top of the source code. Currently, Whisper lacks built-in functionality for end-of-phrase detection. To avoid adding additional dependencies to the toolkit, we introduce an extra forward pass specifically for detecting the end of phrases. This is implemented in both the ROS and ROS2 nodes for speech transcription when using Whisper as their backbone. It's worth noting that this additional implementation may introduce a large overhead to the ROS and ROS2 nodes when running on low-resource devices.

### 5.1.3 Performance evaluation

We evaluate the integration of Whisper [54] and Vosk [1] using five commonly used speech recognition datasets cited in the literature. These range from single-word commands like those in Google Speech Command [64], to phrase-based commands found in Fluent Speech Command [36], as well as longer speeches in Common Voice [3], LibriSpeech [50] and TEDLIUM [31]. In the following paragraphs, we introduce details about these datasets.

Google Speech Commands [64] comprises short, one-second sound clips recorded by thousands of people. The dataset contains 65,000 one-second-long utterances of 30 short words spoken by a diverse group. It includes simple commands like "yes," "no," "up," "down," "left," and "right." This dataset is primarily designed for simple command recognition rather than continuous speech recognition. We use version 0.0.2 of the dataset and use the test split for evaluation.

Fluent Speech Command [36] contains 30,043 utterances spanning 19.0 hours and featuring 97 speakers. Each audio file in the dataset includes a single spoken English command typically used for smart homes or virtual assistants, such as "put on the music" or "turn up the heat in the kitchen." The dataset was recorded under various noise conditions, making it suitable for real-world applications. Our evaluations use the test split, which contains 3,793 utterances spanning

2.4 hours and featuring 10 speakers.

Common Voice [3], part of Mozilla’s Common Voice project, is an open-source, multi-language dataset aimed at providing voice data for speech technology development in as many languages as possible. This dataset includes sentences spoken in various languages and provides details such as the speaker’s gender, age, and accent. It is ideal for training speech-to-text models, particularly for under-represented languages or accents. We use the test split of the English version of the Common Voice Corpus 5.1.

LibriSpeech [50] is a large-scale corpus containing approximately 1,000 hours of English speech. It is one of the most popular datasets for training large-scale speech-to-text models. This dataset features readings from a diverse array of speakers in terms of both accent and demographic information. Each audio file is accompanied by a corresponding transcription, which is word-aligned to indicate the start and end times of words in the audio. We evaluate models using the test-clean split, containing 2,620 utterances from 40 speakers, and totaling approximately 5.4 hours in duration.

TEDLIUM [21] originates from TED Talks audio recordings and is suitable for training and evaluating automatic speech recognition systems on lecture or presentation-style speech. The dataset has multiple versions, each typically containing more data than the last. We use TEDLIUM release 3, which includes 420 hours of speech test split for evaluations and filter out transcriptions labeled as *ignore\_time\_segment\_in\_scoring*.

The Word Error Rate (WER) is our chosen metric for evaluation. It is calculated by dividing the sum of word substitutions, deletions, and insertions by the total number of words in the reference transcription. WER is expressed as a percentage, where a lower value indicates higher accuracy. It ranges from 0%, signifying a perfect match, to potentially over 100%.

In our evaluation process, we focus exclusively on English-language samples. We choose two specific models for this purpose: Whisper’s ‘tiny.en’ [54] and Vosk’s ‘vosk-model-small-en-us-0.15’ [1]. The hyperparameters for these models are left at their default settings for the benchmark. These models were chosen because they offer compact English speech recognition capabilities, aligning closely with the objectives of the OpenDR project. For a uniform assessment, we employ a single setup configuration featuring an English normalizer provided by Whisper [54], along with lower-case conversion. This normalizer standardizes output transcriptions from both Vosk [1] and Whisper [54], as well as the reference transcription. This is particularly advantageous for short transcription datasets like Google Speech Command, where even minor variations in punctuation can significantly affect performance metrics. It’s important to note that the evaluation function integrated into OpenDR retains the raw output from both toolkits, with the only modification being the conversion of all transcriptions to lowercase. For further details on the employed normalizer, we refer readers to Appendix C in [54].

Table 9: English transcription WER (%) on different datasets.

<b>Dataset</b>	<b>Vosk small en-us-0.15 [1]</b>	<b>Whisper tiny.en [54]</b>
Google Speech Commands [64]	87.25	32.98
CommonVoice English 5.1 [3]	38.59	26.44
Fluent Speech Commands [36]	11.34	6.72
TEDLIUM release 3 [21]	13.09	5.95
LibriSpeech [50]	12.52	5.72

The benchmark results are presented in Table 9. We observe a significantly high WER

for the Vosk 'small en-us-0.15' model [1] when tested on the Google Speech Commands [64]. When inspecting the transcription results from this model in Google Speech Commands [64], we find many instances of empty transcriptions or transcriptions with a single word repeated multiple times. This leads us to suspect that there may be issues within the evaluation pipeline. The overall trend observed in the results indicates that the models demonstrate improved performance when provided with longer speech inputs. This can be attributed to longer inputs offering more contextual information, aiding the model in generating more accurate transcriptions. For more exhaustive benchmark results, readers are encouraged to consult the Whisper paper [54] and the Vosk website [1].

## 6 Multi-modal human centric perception and cognition

### 6.1 Improving Unimodal Inference with Multimodal Transformers

#### 6.1.1 Introduction

Research in the field of multimodal learning has primarily focused on tasks where all relevant modalities are assumed to be available during both training and inference stages. Works in this area have involved the development of novel feature fusion techniques, addressing multimodal alignment challenges, and more. However, it is not always realistic to rely on the assumption that all modalities will be present during inference in real-world applications. Various factors, such as transmission delays, media failures, or the nature of the specific application, can result in one or more modalities being unavailable during certain inference steps, even if they were available during training. In addition, usage of multimodal methods is associated with higher computational requirements as architectural solutions tend to be larger. Consequently, the use of unimodal models remains prevalent due to their simplicity and practicality in real-world scenarios. Nevertheless, unimodal models can benefit from multimodal training, which allows them to learn richer feature representations by relating them to other modalities and emphasizing the most relevant unimodal information for the task. To this end, we have aimed to introduce a generalized method for training such unimodal models with multimodal information. Importantly, our multimodal training approach does not increase the computational costs associated with the model, which is especially relevant for edge deployment in robotics applications.

In the context of OpenDR, we instantiate our approach in a form of a language-based intent recognition tool that is trained on language, speech, and vision data with the aim of improving language-based inference. Together with the speech transcription tool, this enables analysing speech intents of a person and predict one of the 20 intents: 'Complain', 'Praise', 'Apologise', 'Thank', 'Criticize', 'Agree', 'Taunt', 'Flaunt', 'Joke', 'Oppose', 'Comfort', 'Care', 'Inform', 'Advise', 'Arrange', 'Introduce', 'Leave', 'Prevent', 'Greet', 'Ask for help' to enable better human-robot interaction [68].

#### 6.1.2 Summary of state of the art

Over the recent years, a set of methods, however limited, has emerged that adopt the "multi-modal training unimodal testing" paradigm. These methods aim to enhance the performance of unimodal models by leveraging multimodal data during the training phase and can be broadly classified into several types. The first type focuses on reconstructing or hallucinating missing

modalities to supplement the training process [14, 15, 18, 59]. Another type of method optimizes alignment objectives between multiple modalities, utilizing techniques like contrastive learning [40] or spatiotemporal semantic alignment [2]. These alignment-based methods are typically well-suited for modalities that have a clear correspondence or pairing, such as RGB and Depth or RGB and Point Clouds. However, they have limited applicability when dealing with modalities that exhibit significant heterogeneity in data types, making their correspondence less evident, such as audio and RGB frames or text and RGB frames. Instead, we take a different approach to tackle this challenge. We propose a more general method that is applicable to various data modalities and unimodal architectures. Therefore, we aim to overcome the limitations of existing methods and provide a solution that is versatile and effective across different types of modalities.

### 6.1.3 Description of the work performed so far

To this end, TAU has proposed an approach to enhance the performance of an arbitrary given unimodal model through multimodal training. We consider the following problem scenario: given data representations from different modalities and corresponding architectures of unimodal models, our objective is to improve the performance of these unimodal models by leveraging multimodal information during the training process.

Our approach revolves around a general framework that unites the unimodal models within a joint architecture during training time, and decouples them during inference. This is accomplished by incorporating a multimodal Transformer-based branch, which is connected to the intermediate features of each modality’s unimodal model. In this setup, each unimodal model operates as a separate branch within the architecture. The multimodal branch is co-trained in conjunction with the resulting unimodal branches and shares early feature extraction layers with them. Additionally, knowledge transfer between the multimodal Transformer branch and the unimodal branches is facilitated through the optimization of a multi-task objective.

During inference, the multimodal branch and any branches corresponding to modalities that are not of interest are discarded, restoring the original architecture of the unimodal model, but now with the parameters of the unimodal model optimized through multimodal training.

More specifically, the data from each modality, denoted as  $X_i$ , is fed into a sequence of layers serving as the backbone for both the unimodal and multimodal branches. This results in a feature representation,  $\Phi_i$ , specific to each modality. The remaining portion of the unimodal branch, as well as the multimodal Transformer branch, independently process  $\Phi_i$ , each with its own task-specific head optimized for the task at hand (e.g., cross-entropy for classification tasks). Additionally, we optimize a knowledge transfer objective, denoted as  $\mathcal{L}_{kt}$ , which facilitates knowledge transfer from the stronger multimodal branch to the weaker unimodal branches. Various objective functions can be used to represent  $\mathcal{L}_{kt}$ . More concretely, we evaluate three alternatives: decision-level alignment, feature-level alignment, and attention-level alignment.

The unimodal and multimodal branches, along with their task-specific and knowledge transfer objectives, are jointly optimized for the task at hand. The shared feature layers receive gradient updates from the task-specific objectives of both the unimodal and multimodal branches, ensuring that they remain informative for both inference paths. This approach helps with extraction of relevant information from each modality, as it prevents the loss of modality-specific information while retaining relevant information necessary for modality fusion. Furthermore, the knowledge transfer objective encourages the remaining segment of the unimodal branch to learn in accordance with the multimodal Transformer, thereby enhancing its performance. Fol-

Table 10: Results on EgoGesture dataset.

Method	Acc-RGB	Acc-Depth	Acc-MM
MobileNetv2-RGB	86.07	-	-
MobileNetv2-Depth	-	86.87	-
MobileNetv2-MM	-	-	87.64
MobileNetv2- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>88.57</b>	<b>88.34</b>	<b>89.19</b>
I3d-RGB	90.69	-	-
I3d-Depth	-	90.64	-
I3d-MM	-	-	91.78
I3d- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>91.96</b>	<b>91.84</b>	<b>92.78</b>

Table 11: Results on RAVDESS dataset.

Method	Acc-Audio	Acc-Video	Acc-MM
Audio model	60.92	-	-
Vision model	-	60.00	-
Multimodal model	-	-	70.83
MM- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>63.16</b>	<b>63.16</b>	<b>73.0</b>

lowing this, a knowledge transfer objective from the multimodal branch to unimodal branches is optimized. Further details as well as a visual representation of the approach can be found in Appendix 8.3.

#### 6.1.4 Performance evaluation

We evaluate our approach using a set of unimodal methods presented in the recent literature on several different tasks: RGBD dynamic hand gesture recognition [70], audiovisual emotion recognition [35], audio-visual-language sentiment analysis [67] as well as audio-visual-language intent recognition [68]. Additionally, considering previous comments from the reviewers, we adopted tri-modal transformers in the multimodal segment of the sentiment analysis and intent recognition tasks, with audio modality represented by both prosodic and spectral speech features in the multimodal sentiment analysis model. Moreover, intent recognition with language-based inference and multimodal training is integrated as a tool into OpenDR toolkit. The results can be seen in the tables 10,11, 12a, and 12b. As can be seen, the proposed approach outperforms the unimodal inference baselines in vast majority of the cases. Interestingly, in certain cases training via the proposed framework also improves the multimodal performance, compared to training the corresponding architecture without unimodal feedback.

Table 12: Results on MOSEI and MIntRec datasets

Method	MAE	Corr	Acc.7
Audio	0.8146	0.2395	41.05
A- $\mathcal{L}_{kt}^{cos}$ (ours)	<u>0.8125</u>	<b>0.2812</b>	40.76
A- $\mathcal{L}_{kt}^{att}$ (ours)	<b>0.8111</b>	<u>0.2493</u>	<b>41.17</b>
Vision	0.8079	0.2313	42.18
V- $\mathcal{L}_{kt}^{cos}$ (ours)	<u>0.8028</u>	<b>0.2774</b>	42.18
V- $\mathcal{L}_{kt}^{att}$ (ours)	<b>0.7978</b>	<u>0.2680</u>	<b>42.73</b>
Text	0.6290	0.6481	48.72
T- $\mathcal{L}_{kt}^{cos}$ (ours)	<b>0.6199</b>	<b>0.6570</b>	<b>49.62</b>
T- $\mathcal{L}_{kt}^{att}$ (ours)	<u>0.6203</u>	<u>0.6537</u>	<u>49.02</u>

(a) Results on MOSEI dataset for multimodal sentiment analysis.

Method	Acc	F1	Prec	Rec
Binary				
Text	88.09	87.98	87.87	88.16
T- $\mathcal{L}_{kt}^{KL}$ (ours)	<u>89.29</u>	<u>89.18</u>	<u>89.14</u>	<u>89.35</u>
T- $\mathcal{L}_{kt}^{att}$ (ours)	<b>90.26</b>	<b>90.16</b>	<b>90.07</b>	<b>90.29</b>
20-class				
Text	71.76	68.18	69.66	67.98
T- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>72.21</b>	<b>69.03</b>	<b>69.71</b>	<b>69.71</b>
T- $\mathcal{L}_{kt}^{att}$ (ours)	<u>71.24</u>	<u>67.92</u>	<u>69.23</u>	<u>67.56</u>

(b) Results on MIntRec dataset for intent recognition.

## 6.2 Multi-frame person detection

### 6.2.1 Introduction and summary of state of the art

Detecting swimmers is a vital part of marine search-and-rescue (SAR) operations. Typical for SAR operations, everything must work fast and reliably since a person in water can stay only a limited time above surface. Having drones with automated perception would speed up detecting the targets in the water. Figure 13 shows that a drone can provide a good coverage of a water area, but detecting the swimmer may need some experience, especially from higher altitudes.



Figure 13: A set up marine SAR situation

The detector itself must be light enough to be fitted in an edge device mounted on a drone, but also it must be able to perform the detection on a high level. This means that there needs to be a compromise between the image quality and detection speed. Figure 14 shows how down sampling hides the details from images taken from different altitudes.



Figure 14: Effects of down sampling the images taken from different altitudes

Current studies have shown that drones can be used successfully for object detection such as detecting pedestrians in traffic surveillance applications, detecting people and objects in water, and also perform human detection in in-land SAR-operations. In marine SAR-operations, the altitude seems to be the biggest limiting factor in detection performance.

For this purpose, we are proposing a new method where we use multiple frames (videos) instead of still images for detecting swimmers, allowing from more reliable detecting humans in the water. We have observed that in many cases it is easier for the human observer to detect the swimmer from the videos while the target is moving. In many cases the human is the only moving object, while others remain still.

### 6.2.2 Description of work performed so far

In our previous studies [53], we have tested how object detector using YOLO architecture performs in marine SAR operation images. The detection from a single image works well in overall, but it has some problems when the drone altitude is increased, since the down sampling makes images to lose some detail. We have used the YOLOv4 [6] model to detect swimmers from single still images (research yet to be published), and here we use modified YOLOv4 for detecting objects from video, so the performance can be compared between single-frame and multi-frame object detection. The input of the model is modified in a way that it is scaled according to the number of input frames. The original YOLOv4 model has 3 input channels  $I_{image} = [R, G, B]$ , where R, G and B are the color channels of the image. When using multiple frames, we scale the number of input channels according to the number of frames we would like to use, as in  $I_{video} = [R_1, G_1, B_1, \dots, R_n, G_n, B_n]$ , where  $n$  denotes the number of frames used. Also illustrated in Figure 15.

When training the network, the bounding boxes from the selected frames are combined in a way that the new annotations have the maximum height and width of the overlapping bounding boxes. The network can be initialised with pre-defined weights for transfer learning, or initialized with randomized weights or zero-value weights.

### 6.2.3 Performance evaluation

Considering this task, avoiding false negative predictions is critically more important than avoiding false positive predictions, since a false negative prediction would mean that a per-



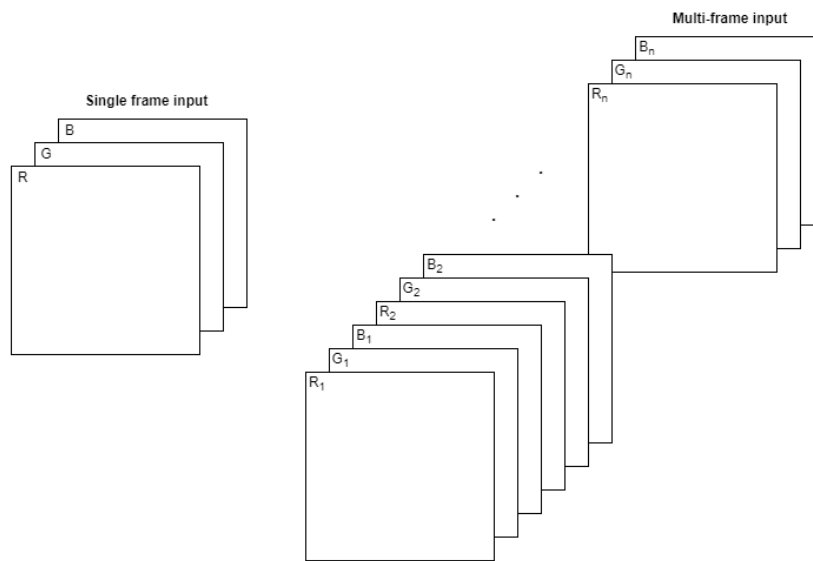


Figure 15: Illustration of different input types of the network

son in the need of help is missed. Performance will be evaluated using precision, recall and  $F_1$ -score as metrics, as well as inspecting the amounts of true positive, false positive and false negative detections. The goal is to increase the recall score, keeping the false negatives to minimum while not getting too much false positives. For this purpose, we are not interested in perfectly overlapping bounding boxes, but it is more important to make detections considering the nature of the task. Because of that, the IoU-threshold (Intersection over Union) has been set to different values, including as low as 10 percent, to make sure that we do not miss any targets because of strict thresholding.

## 7 Conclusions

TAU expanded on its previous work on O2b by extending the human-centric interaction tools, driven by the needs of the OpenDR use cases. Speech recognition functionality was improved to allow general recognition and transcription with models of varying complexity, not constrained by the limitations of the dictionary (Section 5.1). Hand gesture recognition was made more robust and less constrained in realistic scenarios (Section 4.1). Text-based intent recognition leverages multimodal training to achieve higher unimodal prediction rate (Section 6.1), while a method for more reliable human detection, taking into account motion apart from still images, was also developed in Section 6.2.

AUTH finalized its work on O1a on Non-Maximum Suppression (NMS) for person detection (Section 2.5), while also further extended high resolution pose estimation approach developed in OpenDR, contributing towards O1b, in order to more efficiently handle cases where multiple humans appear, as well as included a more challenging evaluation setup in Section 2.3. Finally, AUTH further contributed to O2a by developing an embedding-based active perception approach (Section 2.2) for face recognition by leveraging a new dataset developed by AUTH to enable multi-axes control, as well as finalized the method proposed for using synthesized facial views (Section 2.1) and deep reinforcement learning (Section 2.4) for active face recognition.

AU worked towards O2a by incorporating uncertainty estimation to human action recognition, which can serve as a valuable signal to decide whether an action can be reliably taken based on the perception results, or if the agent should actively improve perception to be certain in its outputs. AU implemented Variational Neural Networks versions of ST-GCN and AGCN methods for skeleton-based human action recognition (Section 3.1), which allow estimating output uncertainty and improve the quality of perception.

## References

- [1] Vosk: Offline speech recognition api, 2023. Available from: <https://github.com/alphacep/vosk-api>. 34, 35, 36
- [2] M. Abavisani, H. R. V. Joze, and V. M. Patel. Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1165–1174, 2019. 37
- [3] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215, 2020. 34, 35
- [4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020. 33
- [5] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight Uncertainty in Neural Networks. 2015. 27
- [6] A. Bochkovskiy, C.-Y. Wang, and H. Liao. YOLOv4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020. 40
- [7] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-NMS: Improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 24
- [8] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields, 2016. 25
- [9] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi. Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*, 2021. 33
- [10] T. Chen, E. B. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. *arXiv:1402.4102*, 2014. 28
- [11] V. Dwaracherla, X. Lu, M. Ibrahimi, I. Osband, Z. Wen, and B. V. Roy. Hypermodels for exploration. In *ICLR Proceedings*, volume 8, 2020. 27
- [12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 24
- [13] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *JMLR Workshop and Conference Proceedings*, volume 48, pages 1050–1059, 2016. 27

- [14] N. C. Garcia, P. Morerio, and V. Murino. Modality distillation with multiple stream networks for action recognition. In *Proceedings of the European Conference on Computer Vision*, pages 103–118, 2018. 37
- [15] N. C. Garcia, P. Morerio, and V. Murino. Learning with privileged information via adversarial discriminative modality distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2581–2593, 2019. 37
- [16] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. M. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu. A survey of uncertainty in deep neural networks. *arxiv:2107.03342*, 2021. 27
- [17] Z. Geng, K. Sun, B. Xiao, Z. Zhang, and J. Wang. Bottom-up human pose estimation via disentangled keypoint regression, 2021. 17
- [18] G. Giannone and B. Chidlovskii. Learning common representation from rgb and depth images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 37
- [19] H. Guo, H. Wang, and Q. Ji. Uncertainty-guided probabilistic transformer for complex action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20020–20029, 2022. 28
- [20] N. Heidari, L. Hedegaard, and A. Iosifidis. Chapter 4 - graph convolutional networks. In A. Iosifidis and A. Tefas, editors, *Deep Learning for Robot Perception and Cognition*, pages 71–99. Academic Press, 2022. 25
- [21] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve. Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*, pages 198–208. Springer, 2018. 35
- [22] J. Hosang, R. Benenson, and B. Schiele. Learning Non-Maximum Suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 24
- [23] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. 33
- [24] W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, et al. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*, 2021. 33
- [25] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 24
- [26] I. Kandel and M. Castelli. Improving convolutional neural networks performance for image classification using test time augmentation: a case study using MURA dataset. *Health Inf. Sci. Syst.*, 9(1):33, 2021. 27

- [27] A. Kapitanov, A. Makhlyarchuk, and K. Kvanchiani. Hagrid - hand gesture recognition image dataset. *arXiv preprint arXiv:2206.08219*, 2022. 32, 33
- [28] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman. The kinetics human action video dataset. *preprint, arXiv:1705.06950*, 2017. 30
- [29] T. S. Kim and A. Reiter. Interpretable 3D human action analysis with temporal convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1623–1631, 2017. 25
- [30] T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve. Rethinking evaluation in asr: Are our models robust enough? *arXiv preprint arXiv:2010.11745*, 2020. 33
- [31] C. Liu, M. K.-P. Ng, and T. Zeng. Weighted variational model for selective image segmentation with application to medical images. *Pattern Recognition*, 76:367–379, 2018. 34
- [32] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019. 30
- [33] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016. 25
- [34] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017. 25
- [35] S. R. Livingstone and F. A. Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018. 38
- [36] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio. Speech model pre-training for end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670*, 2019. 34, 35
- [37] N. M. and D. S. Deep ensemble network using distance maps and body part features for skeleton based action recognition. *Pattern Recognition*, 100:107125, 2020. 25
- [38] I. Mademlis, N. Nikolaidis, and I. Pitas. Stereoscopic video description for key-frame extraction in movie summarization. In *Proceedings of the EURASIP European Signal Processing Conference (EUSIPCO)*. IEEE, 2015. 24, 25
- [39] M. Magris and A. Iosifidis. Bayesian learning for neural networks: an algorithmic survey. *Artificial Intelligence Review*, 2023. 27
- [40] J. Meyer, A. Eitel, T. Brox, and W. Burgard. Improving unimodal object recognition with multimodal contrastive learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5656–5663, 2020. 37

- [41] A. Narayanan, A. Misra, K. C. Sim, G. Pundak, A. Tripathi, M. Elfeky, P. Haghani, T. Strohmman, and M. Bacchiani. Toward domain-invariant speech recognition via large scale training. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 441–447. IEEE, 2018. 33
- [42] B. Nikpour and N. Armanfard. Spatio-temporal hard attention learning for skeleton-based activity recognition. *Pattern Recognition*, 139:109428, 2023. 25
- [43] I. Oleksiienko and A. Iosifidis. Layer ensembles. *arxiv:2210.04882*, 2023. 27, 28
- [44] I. Oleksiienko and A. Iosifidis. Uncertainty-aware ab3dmot by variational 3d object detection. *arxiv:2302.05923*, 2023. 28
- [45] I. Oleksiienko, D. T. Tran, and A. Iosifidis. Variational neural networks implementation in pytorch and jax. *Software Impacts*, 14:100431, 2022. 27
- [46] I. Oleksiienko, D. T. Tran, and A. Iosifidis. Variational neural networks. *Procedia Computer Science*, 222C:104–113, 2023. 27, 28
- [47] I. Osband, J. Aslanides, and A. Cassirer. Randomized prior functions for deep reinforcement learning. In *NeurIPS*, volume 31, pages 8626–8638, 2018. 27
- [48] I. Osband, Z. Wen, M. Asghari, M. Ibrahimi, X. Lu, and B. V. Roy. Epistemic Neural Networks. 2021. 27
- [49] D. Osokin. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose, 2018. 25
- [50] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015. 34, 35
- [51] N. Passalis and A. Tefas. Leveraging active perception for improving embedding-based deep face recognition. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pages 1–6, 2020. 10, 11
- [52] C. Plizzari, M. Cannici, and M. Matteucci. Spatial temporal transformer network for skeleton-based action recognition. *arxiv:2008.07404*, 2020. 26
- [53] L. Qingqing, J. Taipalmaa, J. P. Queralta, T. N. Gia, M. Gabbouj, H. Tenhunen, J. Raitoharju, and T. Westerlund. Towards active vision with uavs in marine search and rescue: Analyzing human detection at variable altitudes. In *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 65–70, 2020. 40
- [54] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023. 34, 35, 36
- [55] RangiLyu. Nanodet-plus: Super fast and high accuracy lightweight anchor-free object detection model. <https://github.com/RangiLyu/nanodet>, 2021. 33

- [56] M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, page 3183–3193, 2018. 27
- [57] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 26, 27, 28, 30
- [58] C. Symeonidis, I. Mademlis, I. Pitas, and N. Nikolaidis. Neural Attention-driven Non-Maximum Suppression for Person Detection. *TechRxiv*, 2021. 24, 76
- [59] W. Teng and C. Bai. Unimodal face classification with multimodal training. In *Proceedings of the 16th International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–5. IEEE, 2021. 37
- [60] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 26
- [61] M. Valdenegro-Toro. Deep sub-ensembles for fast uncertainty estimation in image classification. *arxiv:1910.08168*, 2019. 27
- [62] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019. 27
- [63] G. Wang, W. Li, S. Ourselin, and T. Vercauteren. Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation. In *BrainLes*, volume 11384, pages 61–72. Springer, 2018. 27
- [64] P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018. 34, 35, 36
- [65] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034. IEEE, 2021. 33
- [66] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arxiv:1801.07455*, 2018. 25, 26, 27, 28, 30
- [67] A. Zadeh, A. Bagher, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency. Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2236–2246, 2018. 38
- [68] H. Zhang, H. Xu, X. Wang, Q. Zhou, S. Zhao, and J. Teng. Mintrec: A new dataset for multimodal intent recognition. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1688–1697, 2022. 36, 38
- [69] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *IEEE International Conference on Computer Vision*, pages 2117–2126, 2017. 25

- [70] Y. Zhang, C. Cao, J. Cheng, and H. Lu. Egogesture: a new dataset and benchmark for egocentric hand gesture recognition. *IEEE Transactions on Multimedia*, 20(5):1038–1050, 2018. 38
- [71] Y. Zhang, D. S. Park, W. Han, J. Qin, A. Gulati, J. Shor, A. Jansen, Y. Xu, Y. Huang, S. Wang, et al. Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1519–1532, 2022. 33
- [72] Y. Zhang, Z. Zhao, W. Li, and L. Duan. Multi-scale enhanced active learning for skeleton-based action recognition. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. 27, 28
- [73] R. Zhao, K. Wang, H. Su, and Q. Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6881–6891, 2019. 27, 28, 30, 31

## 8 Appendix

### 8.1 Using Synthesized Facial Views for Active Face Recognition

The appended papers follow.



# Using Synthesized Facial Views for Active Face Recognition

Efstratios Kakaletsis\* and Nikos Nikolaidis

Department of Informatics, Artificial Intelligence & Information Analysis Laboratory, Aristotle University of Thessaloniki, Thessaloniki, GR-54124, Greece.

\*Corresponding author(s). E-mail(s): [ekakalets@csd.auth.gr](mailto:ekakalets@csd.auth.gr);  
Contributing authors: [mnik@csd.auth.gr](mailto:mnik@csd.auth.gr);

## Abstract

Active perception / vision exploits the ability of robots to interact with their environment, for example move in space, towards increasing the quantity or quality of information obtained through their sensors and, thus, improving their performance in various perception tasks. Active face recognition is largely understudied in recent literature. Attempting to tackle this situation, in this paper, we propose an active approach that utilizes facial views produced by photorealistic facial image rendering. Essentially, the robot that performs the recognition selects the best among a number of candidate movements around the person of interest by simulating their results through view synthesis. This is accomplished by feeding the robot's face recognizer with a real world facial image acquired in the current position, generating synthesized views that differ by  $\pm\theta^\circ$  from the current view and deciding, based on the confidence of the recognizer, whether to stay in place or move to the position that corresponds to one of the two synthesized views, in order to acquire a new real image with its sensor. Experimental results in three datasets verify the superior performance of the proposed method compared to the respective "static" approach, approaches based on the same face recognizer that involve synthetic face frontalization and synthesized views, random direction robot movement, robot movement towards a frontal location based on view angle estimation, as well as a state of the art active method. Results from a proof of concept simulation in a robotic simulator are also provided.

**Keywords:** active vision ; active face recognition ; synthesized facial views; photorealistic facial synthesis

## 1 Introduction

In recent years, the robotics and vision communities have started researching more thoroughly the field of active vision / perception and exploration. Active perception methods try to obtain more, or better quality, information from the environment by actively choosing from where and how to observe it using a camera (or other sensors), in order to accomplish more effectively tasks such as 3D reconstruction [1, 2], [3], [4], [5] or object recognition [6], [7]. This could be achieved, for example, by moving a camera-equipped mobile robot, e.g. a wheeled robot or a UAV, in positions which offer different (and hopefully better) views of the object of interest. Although active 3D object reconstruction has attracted considerable interest, mainly towards solving the "next-best-view" problem (i.e. choosing the next viewing position in order to obtain a detailed and complete 3D object model), active approaches for recognition tasks, especially for face recognition, are less frequent in the literature. Deep Learning has lately dominated face recognition research due to the superior performance achieved. However the vast majority of recognition methods adopt a static approach i.e., an approach that is based on an image acquired from a specific viewpoint, even in setups where an active approach could have been used. Indeed, face recognition can be combined with an active approach for controlling the movement of a camera-equipped robot towards capturing the face from more informative views and thus obtaining more robust results, at the expense of energy consumption and additional time needed. Synthesized views of faces, whose images were acquired through a camera, can be used for robot movement guidance in an active face recognition setup. Instead of having the robot move in a physical way for capturing a novel (and better) view, one can use a synthesized view as an aid towards choosing a new viewpoint and improving recognition through an acquisition procedure.

In this paper, we propose an active face recognition approach that utilizes facial views synthesized by photorealistic facial image rendering. Essentially, the camera-equipped robot that performs the recognition selects the best among a number of candidate physical movements around the face of interest by simulating their results through view synthesis. In other words, once the robot (that is at a certain location with respect to the subject) acquires an image, it feeds the face recognizer with this image as well as with synthesized views that differ by  $\pm\theta^\circ$  from the current view. Subsequently, it either stays in the current position or moves to the position that corresponds to one of the two synthesized views. The respective decision is based on the confidence of the three recognitions (on the real and the two synthesized views). In case of a "move" decision, it proceeds in acquiring a "real" image from its new location. The procedure repeats in the same manner, for this location, for one or more

steps. Using synthesized facial views facilitates decision-making by providing estimates of what is to be expected (in terms of recognition accuracy) in a new robot position. The proposed method involves a face recognizer that is trained to recognize frontal or nearly frontal faces, while having to deal with input facial images obtained from an arbitrary view point. This fact makes recognition challenging, but at the same time more easily applicable in a real-world scenario, since it does not require the existence of facial images acquired from different viewpoints in order to train a view-independent face recognizer.

The remainder of this paper is organized as follows. In Section II related work is presented, whereas in Section III we describe the details of the proposed method. In Section IV experiments conducted to measure the algorithm's performance are presented. Finally, Section V provides a discussion and conclusions.

## 2 Related Work

### 2.1 Active Computer Vision

A few recent active approaches for tasks such as object detection, recognition, 3-D reconstruction and manipulation are presented in this section. Additional methods can be found in the review paper [8] that deals in particular with the problem of view planning in robot active vision.

In [9], a robotic arm equipped with a depth camera captures information for a scene from several poses, towards understanding the environment and performing multiple object detection. Boundary Representation Models (B-Reps) are used to represent the objects. The world representation is initialized and, after generating a first set of object detection hypotheses, the approach tries to perform exploration in order to generate new hypotheses or validate existing ones. This is accomplished by finding regions of interest (regions to be inspected) and suitable new views, acquired by appropriate poses of the arm. A proof of concept using a KUKA LWR 4 arm is provided. As expected, the object recognition rate increases as the number of views increases.

In [10] the authors deal with the problem of reconstructing a scene while also identifying the objects in it using 3D scans and a dataset of 3D shapes. Towards this end, a 3D attention model is developed that selects the best views to scan from, as well as the most informative regions within in each view, so as to achieve object recognition. The region-level attention mechanism generates features which are fairly robust against occlusion. Temporal dependencies among consecutive views are encoded with deep recurrent networks.

A new approach, called 3D ShapeNets, for representing a 3D shape as a probability distribution of binary variables on a voxel grid, using a Convolutional Deep Belief Network is proposed in [11]. This representation supports joint object recognition and shape completion from depth maps and enables active object recognition through view planning. The model, learns the distribution of 3D shapes from different object categories and various poses from raw CAD data, while also discovering hierarchical compositional part representations.

Moreover, in [12], the authors present a novel methodology for optimizing a robot's vision sensor viewpoint and apply it in the tasks of object recognition and manipulation (grasping synthesis) in unstructured environments. The algorithm uses extremum seeking control (ESC), which utilizes a task success criterion in a continuous optimization loop. In the case of object recognition, an image is captured by the robot's camera and supplied to the recognition algorithm. The algorithm generates a success rate value (probability of recognizing an object) that forms the main component of the objective function, which is to be maximized by the neural-network based ESC algorithm, towards generating velocity commands for the robot camera. The camera moves on a sphere (viewsphere) around the object, i.e., it points to the object all the time while keeping the distance fixed. The algorithm requires neither a task model nor training on offline image data for viewpoint optimization and is shown to be robust to occlusions.

In [13] another active vision-based object recognition approach is presented, among other contributions. More specifically, a CNN-based approach is described that allows object recognition over arbitrary camera trajectories, (which generate multi-view image sequences) without requiring explicit training over the potentially infinite number of camera paths and lengths. This is done by decomposing an image sequence into a set of image pairs, classifying each pair independently, and then learning an object classifier by weighting the contribution of each pair. The method is then extended to the next-best-view problem in an active recognition framework. This is accomplished by training a second CNN to map from an observed image to the next viewpoint and incorporating it into a trajectory optimisation task.

In [14] a method for active object recognition that involves a deep CNN for the simultaneous prediction of the object label and the next action to be performed by the sensor so as to improve recognition performance is presented. The task is treated as a reinforcement learning problem and a generative model of object similarities is embedded in the network for encoding the state of the system. Other, older, active object recognition approaches, are reviewed in [15–17].

[1] deals with the problem of active object reconstruction. In there, a next-best-view planning scheme based on supervised deep learning is proposed. A properly trained three-dimensional convolutional neural network (3D-CNN) is used to predict the next-best-view position, given the current view.

Finally, in [18] a viewpoint planning strategy for 3D reconstruction with application in the reconstruction of blades is presented. The algorithm focuses on controlling surface overlap for the various views so as to allow for successful registration. OctoMaps are used towards this end and the method is tested in both simulation and real blade reconstruction.

## 2.2 Active Face Recognition

Despite the fact that active object recognition has attracted considerable interest in the computer vision and robotics communities, active face recognition

has been scarcely studied. Such a simple method is described in [6] and comprises of a neural network-based face recognizer along with a decision making controller that decides for the viewpoint changes. More specifically, a pre-trained VGG-Face CNN is used by the recognition module in order to extract facial image features and it is combined with a nearest-neighbor identity recognition criterion. The simple controller module can make three different decisions based on the uncertainty of the current result (i.e., the distance  $d$  between the input image and the closest image in the database of known persons): a) recognize the individual, if  $d$  is below a threshold  $t_1$  b) disregard the individual as unknown, if  $d$  is above a threshold  $t_2$  or c) reassess the subject by moving to a different viewpoint, if  $t_1 < d < t_2$ . The direction towards which the movement shall be performed in order to increase the probability of correct recognition is not studied by the authors.

The authors in [7] propose a deep learning-based active perception method for embedding-based face recognition and examine its behavior on a real multi-view face image dataset. The proposed approach can simultaneously extract discriminative embeddings, as well as predict the action that the robot must take (stay in place, move left or right by a certain amount, on a circle centered at the person) in order to get a more discriminative view.

### 2.3 Multi-view Facial Image Synthesis

A significant number of techniques for synthesizing facial images in novel views appeared in the last years since such images can have a number of applications, e.g., in improving face recognition accuracy. For example, since profile faces usually provide inferior recognition results compared to frontal faces, generative adversarial networks (GANs) based methods for the frontalization of profile facial images [19] or generation of other facial views [20] have been proposed for improving face recognition results.

A method for the generation of frontal views from any input view that utilizes a novel generative adversarial architecture called the Attention Selective Network (ASN) is described in [21]. Towards improving single-sample face recognition by both generating additional samples and eliminating the influence of external factors (illumination, pose), [22] presents an end-to-end network for the estimation of intrinsic properties of a facial image with recovery of albedo UV map and 3D facial shape. In [23], a facial image rendering technique is used both in the training and testing stages of a face recognition approach.

A method that produces photorealistic facial image views is described in [24]. The basic idea of this approach is that rotating faces in the 3D space and re-rendering them to the 2D plane can serve as a strong self-supervision. A 3D head model (obtained by utilizing the 3D-fitting network 3DDFA [25–27]), accompanied by the projected facial texture of a single view, is being rotated and multi-view images of the face are rendered using the Neural 3D Differential Renderer [28] along with 2D-to-3D style transfer and image-to-image translation with GANs to fill in invisible parts. This last state-of-the-art

method was selected due to its robustness and photorealistic quality for the generation of the synthetic facial images required by the method proposed in this paper.

Although facial view synthesis can improve face recognition performance, active perception methods can be expected to provide better results, in cases where acquisition of additional "real" facial views is possible due to the existence of e.g. a wheeled robot.

## 3 Proposed Active Face Recognition Algorithm

### 3.1 Face Recognition

Let us denote as database subset  $G$  a set of training facial images for the persons that shall be recognized. Similarly, the facial images to feed the face recognizer are included in the query (test) set  $T$ . The face recognition library face.evoLve [29] which contains many state-of-the-art deep face recognition models, is used. More specifically, an implementation of a certain face recognition approach of face.evoLve from the OpenDr Toolkit [30, 31] was used. IR-50 (50 layers) [32] trained on MS-CELEB-1M using an ArcFace [33] loss head was used as the 512-dimensional feature extraction backbone.

For the database subset  $G$ , face detection, facial landmark extraction and face alignment was based on the face.evoLve module that is based on MTCNN [34], whereas for the query images in  $T$ , these processing steps were based on RetinaFace [35, 36]. Face recognition is performed by a nearest-neighbor classifier that uses Euclidean distance in the 512-dimensional feature space to find the database facial image that best matches the query image.

Face recognition confidence  $FRC \in [0, 1]$ , is also evaluated based on the distance between the input query image and the nearest image in the database  $G$ . The  $FRC$  is given by the following formula:

$$FRC = 1 - \frac{distance}{threshold} \quad (1)$$

where  $distance$  is the euclidean distance of query facial image from the nearest neighbor image in the database  $G$  and  $threshold$  is the optimal threshold found by running a pairwise matching experiment on LFW [37].

### 3.2 Active Face Recognition Using Synthesized Views

The proposed active face recognition algorithm uses the face recognition confidence  $FRC$  and facial images synthesized for view angles around the current robot view, in order to select the next robot movement, towards performing a successful recognition. Starting from an initial position, the robot can take one of the following three decisions: stay at the current position, move by  $\theta^\circ$  to the right or move by  $\theta^\circ$  to the left, in order to acquire a new image. Depending on the achieved recognition confidence, one or more additional movements, towards the same direction as the first one, might be decided.

**Algorithm 1** Active Face Recognition Algorithm on Pseudocode**Input:**  $I_r$ , *threshold*,  $\theta^\circ$ **Result:**  $Person_{ID}(I_r)$ 


---

```

1:  $\alpha = Estimate\_View\_Angle(I_r)$ 
2:  $I_s^- = Render(\alpha - \theta^\circ, I_r)$ 
3:  $I_s^+ = Render(\alpha + \theta^\circ, I_r)$ 
4:  $I = argmax(FRC(x))$ 
    $x \in \{I_r, I_s^-, I_s^+\}$ 
5: if  $I = I_r$  then
6:    $I_{ID} = I_r$ 
7:   go to 32
8: else
9:   if  $I = I_s^+$  then
10:     $\theta_{incr} = +\theta^\circ$ 
11:   else
12:     $\theta_{incr} = -\theta^\circ$ 
13:   end if
14: end if
15:
16:  $I_r^{1step} = Move\_and\_Capture(\alpha + \theta_{incr})$ 
17: if  $FRC(I_r^{1step}) > threshold$  then
18:    $I_{ID} = argmax(FRC(x))$ 
    $x \in \{I_r, I_r^{1step}\}$ 
19:   go to 32
20: else
21:    $I_s^{2step} = Render(\alpha + 2 * \theta_{incr}, I_r^{1step})$ 
22:   if  $FRC(I_s^{2step}) < FRC(I_r^{1step})$  then
23:      $I_{ID} = argmax(FRC(x))$ 
    $x \in \{I_r, I_r^{1step}\}$ 
24:     go to 32
25:   else
26:      $I_r^{2step} = Move\_and\_Capture(\alpha + 2 * \theta_{incr})$ 
27:      $I_{ID} = argmax(FRC(x))$ 
    $x \in \{I_r, I_r^{1step}, I_r^{2step}\}$ 
28:     go to 32
29:   end if
30: end if
31:
32:  $Person_{ID}(I_r) = Recognize(I_{ID})$ 

```

---

More specifically, given a facial query image  $I_r$  (subscript  $r$  stands for real), captured by the robot camera at the robot starting position, the face synthesis algorithm [24] is utilized to estimate the robot view angle  $\alpha$  and then render/generate facial views in 2 different view angles i.e.  $-\theta^\circ$  and  $+\theta^\circ$  in pan with respect to the pan of  $I_r$  (and the same tilt as  $I_r$ ). These two images are

denoted by  $I_s^-$  and  $I_s^+$  respectively (subscript  $s$  stands for synthetic). Then, the face recognizer is fed with these three images  $I_r$ ,  $I_s^-$ ,  $I_s^+$  (one real, two synthetic ones). Depending on the image that obtained the biggest face recognition confidence  $FRC$ , the robot stays in its current position (if  $FRC$  was maximum in  $I_r$ ) or physically moves  $-\theta^\circ$  (or  $+\theta^\circ$ ) (if  $FRC$  was maximum in  $I_s^-$  (or  $I_s^+$ )) and acquires through its camera a new real image  $I_r^-$  (or  $I_r^+$ ). If a "stay" decision was taken, the algorithm outputs the ID of the person it recognized in  $I_r$  and terminates. If the robot moved, face recognition is performed again in  $I_r^-$  (or  $I_r^+$ ) and the obtained  $FRC$  is compared to an experimentally evaluated threshold  $t$ . In case a high enough confidence was observed, the algorithm outputs the ID of the person it recognized in  $I_r^-$  (or  $I_r^+$ ) and terminates. If not, it tries additional  $+\theta^\circ$  steps (movements) in pan, in the same direction as the first step. In more detail, in this second step, it generates/synthesizes a facial view  $-\theta^\circ$  (or  $+\theta^\circ$ ) in pan from the current pan value (and the same tilt), denoted as  $I_s^{--}$  (or  $I_s^{++}$ ), and evaluates (by calling the face recogniser)  $FRC$  on this synthetic image. If  $FRC(I_r^-) > FRC(I_s^{--})$  (or  $FRC(I_r^+) > FRC(I_s^{++})$ ) the algorithm decides that the robot shall stay in its current position, outputs the ID of the person it recognized in  $I_r^-$  (or  $I_r^+$ ) and terminates. Otherwise, the robot physically moves  $-\theta^\circ$  ( $+\theta^\circ$ ) from its current position, acquires a new image  $I_r^{--}$  ( $I_r^{++}$ ) and the algorithm outputs the ID of the person it recognized in this image. The procedure can be repeated for a number of additional steps (movements), until the predefined maximum number of steps is reached. The performance of the proposed procedure obviously depends on whether the synthesis algorithm [24] estimates with sufficient accuracy the view angle of the query image  $I_r$  and also on whether the synthesized views are of good quality. In order to limit the possibly negative effect of these factors on the performance of the algorithm (e.g. by leading it to move towards the wrong direction), the algorithm does not actually take a decision based on the last real image it has visited but does so based on the real image where it has obtained the maximum  $FRC$  value. In more detail, if the algorithm took one step of  $-\theta^\circ$ , it takes a decision using the real image  $I$  given by:

$$I = \underset{x \in \{I_r^-, I_r\}}{\operatorname{argmax}} (FRC(x)) \quad (2)$$

or the equivalent expression that involves  $I_r^+$ ,  $I_r$ , if a step of  $+\theta^\circ$  has been taken. Similarly, if two steps of  $-\theta^\circ$  each have been performed, the algorithm decides on the person ID using the real image  $I$  given by:

$$I = \underset{x \in \{I_r^{--}, I_r^-, I_r\}}{\operatorname{argmax}} (FRC(x)) \quad (3)$$

or the equivalent expression that involves  $I_r^{++}$ ,  $I_r^+$ ,  $I_r$ , if two steps, of  $+\theta^\circ$  each, have been taken. The pseudocode for the proposed method, when two steps are allowed, is presented in algorithm 1.



It should be noted that the actual recognition is always performed on a real image, i.e., an image captured by the robot camera. The synthesized views are only used to aid the robot in deciding whether to move in a new position (and acquire a new image there) or stay in the current position. The rationale behind the proposed approach is that in case the initial robot position is far from a frontal or nearly frontal one, the algorithm will hopefully direct it to move towards a position which is closer to a frontal one. Obviously, the procedure can work, in the same way, for tilt.

## 4 Performance Evaluation

For the evaluation of the proposed active face recognition approach, a number of experiments were conducted using the HPID dataset [38], the Queen Mary University of London Multi-view Face Dataset (QMUL)[39] and a Synthetic Dataset (SD) [40]. In all three datasets, images of all subjects were divided into two non-overlapping subsets: a database subset  $G$  (images that the face recognizer uses to decide upon the ID of the query image through the nearest neighbor classifier) and a query (test) subset  $T$  (these are meant to be the images captured by the robot camera in its initial position). This was done by choosing images with different pan ranges for  $G$  and  $T$ . With this setup we are simulating active recognition where the robot is moving only in the pan direction. Short descriptions of the three datasets are provided below.

### 4.1 HPID Dataset

The HPID dataset [38] is a head pose image database consisting of 2790 face images of 15 subjects captured by varying the pan and tilt angles from  $-90^\circ$  to  $+90^\circ$ , in  $15^\circ$  increments. Two series of images were captured for each person, (93 images in each series).

The database subset  $G$  (Figure 1) contains facial images with tilt in angles  $[-30^\circ, -15^\circ, 0^\circ, +15^\circ, +30^\circ]$  and pans  $[-15^\circ, 0^\circ]$ , i.e. only nearly frontal images. The query subset (Figure 2) contains face images with tilts  $[-30^\circ, -15^\circ, 0^\circ, +15^\circ, +30^\circ]$  and pans  $[-90^\circ, -75^\circ, -60^\circ, -45^\circ, -30^\circ]$ . The selection of the range  $[-90^\circ \dots -30^\circ]$  in pan, instead of the full ( $[-90^\circ \dots -30^\circ]$  and  $[+30^\circ \dots +90^\circ]$ ) semi-circle, in this and the other two datasets, was just for simplicity. Similar results were obtained in experiments involving the full semi-circle.

Synthetic images generated from the "real" query images for use from our algorithm are depicted in Figure 3.

### 4.2 QMUL Dataset

Queen Mary University of London Multi-view Face Dataset (QMUL) [39] consists of automatically aligned, cropped and normalised face images of 48 persons. Images of 37 persons are in greyscale (dimensions: 100x100 pixels) whereas those of the remaining 11 subjects are in colour and of dimensions

56x56. For each person 133 facial images exist, covering a viewsphere of  $-90^\circ \dots +90^\circ$  in pan and  $-30^\circ \dots +30^\circ$  in tilt in  $10^\circ$  increments. For the Database split G, the facial images with pan in angles  $[-10^\circ, 0^\circ]$  and tilt in angles  $[-30^\circ, \dots, +30^\circ]$  were used. The Query split T (test) includes images with pan in angles  $[-90^\circ, \dots, -20^\circ]$  and tilt in the range  $[-30^\circ, \dots, +30^\circ]$ .

### 4.3 Synthetic Dataset

The Synthetic Dataset (SD) was generated using Unity's Perception package. It consists of 175422 cropped face images of 33 subjects taken at different environments, lighting conditions, camera distances and angles. In total, the dataset contains images for 8 environments, 4 lighting conditions, 7 camera distances (1m-4m) and 36 camera angles ( $0 - 360^\circ$  at  $10^\circ$  intervals). A subset of the dataset was used in the experiments. The subset included all 33 subjects in all environments and 1 lighting condition, at a camera distance of 1.0 m. For the Database split G, facial images with pan  $[0^\circ, +10^\circ]$  and tilt  $0^\circ$  were used. The Query (test) split T included images with pan in the range  $[+20^\circ, \dots, +90^\circ]$  and tilt  $0^\circ$ .



**Fig. 1** Samples from the database subset  $G$  of the HPID dataset, depicting real facial images of a subject with tilt angles  $[-30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ]$  and pans  $[-15^\circ, 0^\circ, 15^\circ]$ .

### 4.4 Results

Results (in terms of recognition accuracy) are presented in Table 1. The line marked "Static" in this Table presents the result of the static equivalent of our approach, in which only the initial query facial image is used by the same recogniser involved in the active approach. As can be seen, the proposed active method (lines "Proposed (Active), 2 steps" and "Proposed (Active), 4 steps", referring to the cases where the robot can move up to 2 or 4 times from its initial position in  $\theta^\circ$  increments) outperforms its static counterpart for both 2 and 4 algorithm steps, at the obvious expense of additional robot movements and time required to perform them. For the HPID and SD datasets the best performance is obtained for 4 steps of the algorithm and the absolute increase of accuracy with respect to the static version is 15.61% and 13.05% respectively, whereas for the QMUL dataset the best performance is obtained for 2 steps (increase of 15.69% compared to the static approach).



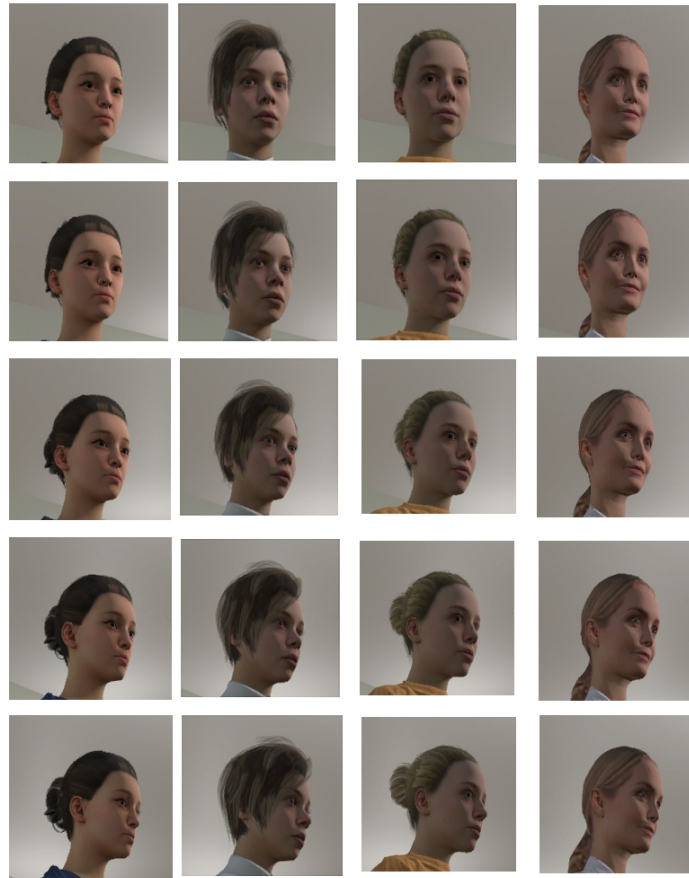
**Fig. 2** Samples from the query subset  $T$  depicting real facial images of a subject with tilts  $[-30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ]$  and pans  $[-90^\circ, -75^\circ, -60^\circ, -45^\circ, -30^\circ]$ .



**Fig. 3** Samples of synthetic facial images generated from the query subset  $T$  of the HPID dataset. Each row depicts the two synthetic images generated in pan angles  $pan - 15^\circ$ ,  $pan + 15^\circ$  from real images with pans  $[-75^\circ, -60^\circ, -45^\circ, -30^\circ]$ . Each row corresponds to a different tilt value of the real image, in the range  $[-30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ]$ .



**Fig. 4** Samples from the database subset  $G$  of the SD dataset, depicting facial images of four subjects with tilt angle  $0^\circ$  and pans  $[0^\circ, +10^\circ]$ .



**Fig. 5** Samples from the query subset  $T$  of SD Dataset depicting facial images of four subjects with tilt  $0^\circ$  and pans  $[+20^\circ, +30^\circ, +40^\circ, +50^\circ, +60^\circ]$ .

**Table 1** Face recognition accuracy results and comparison with the static approach and other variants

Method	HPID[38]	QMUL [39]	Synthetic(SD) [40]
<i>Static (only Queries)</i>	72.49 %	69.88%	66.95%
<i>Proposed (Active) (2 steps)</i>	82.12%	<b>85.57%</b>	68.35 %
<i>Proposed (Active) (4 steps)</i>	<b>88.10%</b>	82.85%	<b>80%</b>
<i>Random direction movement (2 steps)</i>	71.31%	72.68%	63.54%
<i>Frontalization by physical movement (real frontal views)</i>	74.82%	62.83%	56.33%
<i>Frontalization (synthetic frontal views)</i>	80.75%	75.95%	66.10%
<i>Static &amp; Synthetic (real &amp; synthetic views) (4 steps)</i>	72.22%	66.35%	62.28%

**Table 2** Comparison with [7]

Method	HPID [38]	QMUL [39]	Synthetic (SD) [40]
<i>Proposed (Active) (2 steps)</i>	82.88%	82.47%	85.00%
<i>Proposed (Active) (4 steps)</i>	<b>87.78%</b>	<b>84.59%</b>	<b>88.81%</b>
[7] (Active) (2 steps)	60.96%	69.94%	67.63%
[7] (Active) (4 steps)	61.30%	68.11%	70.41%

The proposed approach was also compared to the frontalization approach that is often used in face recognition when the recognizer is trained only on

frontal views. In this case, the facial view synthesis algorithm [24] is used in order to generate a frontal ( $0^\circ$  in pan) view from the input (query) image. This image is then provided to the recognizer. The results (line "Frontalization (synthetic frontal views)") show that although frontalization achieves improved performance with respect to the static approach in HPID and QMUL datasets and similar performance in SD, it is superseded by the proposed active approach. Indeed the best achieved results of the proposed approach correspond to an absolute increase in accuracy (with respect to the frontalization approach) of 7.35%, 9.62% and 13.9% for the HPID, QMUL and SD datasets respectively.

One can naturally wonder what is the benefit of introducing an active approach, that involves actual robot movement, over the use of synthetic images only. To answer this question we set up another experiment where for each (real) query image, captured at a view angle  $\alpha$  we generate (where possible) 8 synthetic images at angles  $\alpha \pm \theta^\circ, \dots, \alpha \pm 4\theta^\circ$  around the query and feed them to the recognizer along with the query image. The result with the highest FRC is then adopted as the final decision. Results are presented in line "Static & Synthetic (real & synthetic views) (4 steps)". Obviously this approach is not viable, providing results inferior to those of the static case.

One could also argue that, instead of using the synthesized views as proposed in this paper, it would suffice to estimate the view angle of the robot with respect to the person and instruct it to move directly (i.e., without intermediate steps) to the position that would allow it to obtain a frontal view ( $0^\circ$  in pan). However, there are certain difficulties that would make such an approach hard to implement in practice. Indeed, we observed during the experiments that view angle estimates (at least those provided by the view synthesis algorithm used in this paper) although accurate enough for the purposes of view synthesis, are quite far from the ground truth values, thus deeming this approach problematic. Experiments were performed to quantitatively verify this claim. The experimental evaluation was conducted on all three datasets, and involved obtaining the view angle estimate  $\theta^\circ$  and instructing the robot to physically move by  $-\theta^\circ$  and recognise the subject from its -supposedly-frontal new position. The respective recognition accuracy figures are provided in the line "Frontalization by physical movement (real frontal views)" of Table 1. The obtained results show that this approach is significantly inferior to the proposed one and also worse than the frontalization approach that is based on view synthesis. As a matter of fact, this approach is inferior to even the static one in two out of three datasets.

Another set of experiments were conducted in order to prove that the guidance provided by the synthesized views with respect to the direction the robot shall move is beneficial for the proposed algorithm. Towards this end, the proposed approach was compared to a two-step random direction movement approach that was implemented as follows: starting from its initial position, the robot chooses a random direction (positive or negative rotation, i.e., right or left movement) and then performs two  $\theta^\circ$  steps ( $\theta^\circ = 10^\circ$  or  $15^\circ$  depending on

the dataset) towards this direction, capturing the respective (real) images. The decision on the ID of the depicted person is then taken based on the real image (one of the three available) where the maximum *FRC* value was observed. This approach is similar to the 2 step version of the proposed algorithm, the difference being that the movement direction is not decided on the basis of the utilized synthetic views but is chosen randomly. The recognition accuracy results for this approach are presented in row "Random direction movement (2 steps)" of Table 1. By observing this Table, one can notice that results are close to those of the static approach but clearly inferior to those obtained by the 2 step version of the proposed algorithm. This verifies that the guidance provided by the synthetic images is indeed beneficial for the recognition.

Finally, the proposed method was compared to the recent embedding-based active deep face recognition technique [7]. The experimental setup followed in [7] for the HPID dataset, was used in all three datasets. More specifically, 75% of the subjects contained in each dataset was used to train the models of [7], while the remaining 25% were used for evaluating the trained models (test set). Since our approach requires no training, only the test set data were utilized in the experiments that involved it. Images in the test set were used to form the Database split G and the Query split T, in the same way (same range of pan and tilt angles) mentioned in Sections 4.1 to 4.3. Results are presented in Table 2. One can observe that the proposed method outperforms the method in [7] in both the 2 and 4 steps setups, achieving (in the 4 steps setup) an absolute increase in accuracy of 26.48%, 16.48% and 18.40% for the HPID, QMUL and SD datasets respectively.

Statistics regarding the steps taken by the proposed approach (4 steps) are presented in Table 3 for the SD dataset. Each row in this Table corresponds to the type of real image that the algorithm reached in its course, i.e., the number of steps it has taken towards the right or the left direction. These types are mentioned in the first column and follow the same naming conventions used in Section 3.2. For example, the row marked  $I_r^+$  includes statistics for cases where the algorithm (robot) moved by  $+10^\circ$  from its initial position (the one represented by the input query image). The pan angle increment from the initial position, the number of images and the percentage they represent over the total are presented for each case. The presented statistics show that in 24.34% of the cases the robot decided to stay in its initial position whereas in the remaining 75.66% it moved by  $\pm 10^\circ, \dots, \pm 40^\circ$  (one to four steps). It shall be noted however that the decision on the ID of the depicted person is not necessarily obtained from the last position the robot has visited, due to the fact that the image with the maximum recognition confidence (FRC) is used for this purpose.

The average number of movements that the algorithm instructs the robot to perform can be easily evaluated from statistics such as the ones presented in Table 3. The respective figures are presented in Table 4. Note that in case the robot decides to perform no movement (stay decision) the number of movements is obviously zero. As can be seen, when 4 steps are allowed, the

**Table 3** Active Face Recognition Statistics (4 Steps, SD dataset): steps performed by the algorithm.

Image type	Angle	# Images	Percentage
$I_r$	0°	28	24.34%
$I_r^+$	+10°	5	4.347%
$I_r^{++}$	+20°	7	6.086%
$I_r^{+++}$	+30°	5	4.347%
$I_r^{++++}$	+40°	3	2.608%
$I_r^-$	-10°	52	45.217%
$I_r^{--}$	-20°	8	6.956%
$I_r^{---}$	-30°	4	3.478%
$I_r^{----}$	-40°	3	2.608%
<b>Total</b>	–	115	100%

algorithm instructs the robot to make, on average, from 0.76 to 1.17 movements, a fact that denotes that the time required for active recognition (time for the computations as well as the time for the robot to move) is relatively low and can be further lowered if only 2 steps are allowed.

**Table 4** Active Face Recognition Statistics: average number of steps.

Method	HPID [38]	QMUL [39]	SD [40]
<i>Proposed (Active)</i> 2 steps	0.82	0.6689	1.14
<i>Proposed (Active)</i> 4 steps	<b>0.89</b>	<b>0.7623</b>	<b>1.17</b>

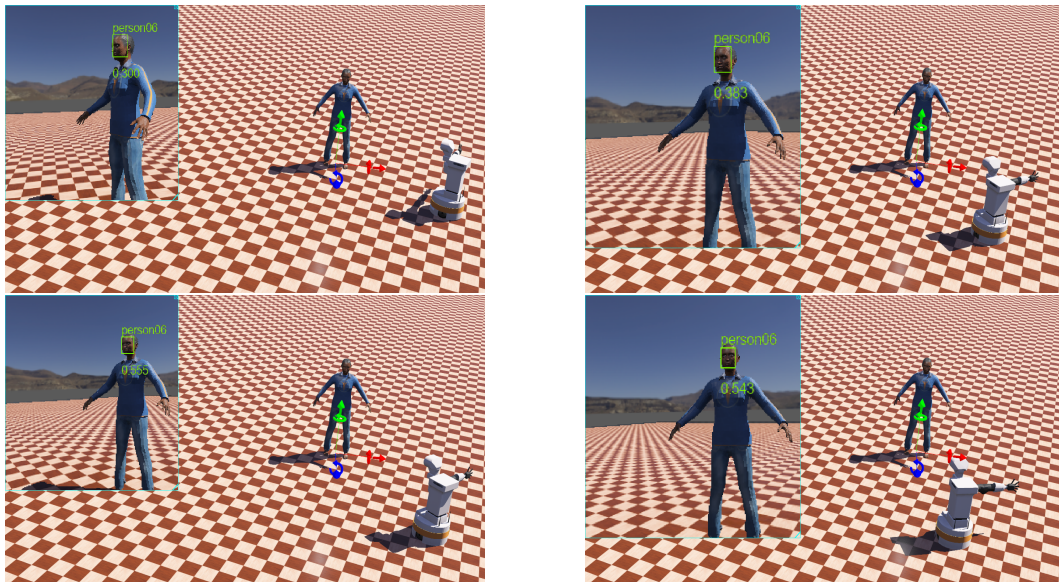
In addition, Table 5 presents statistics regarding the moves that the algorithm (equivalently the robot in a real situation) performs and whether these lead towards a frontal view, i.e., 0° in pan (which is something that might be expected since the recognised is trained on near frontal images) or away from such a view. The statistics for the HPID, show that in most cases (59.6%) the algorithm moves the robot towards a frontal view. However, in another 20.6% of the cases the robot moves away from the frontal position, which indicates that either the estimate for the view angle of the input (query) image provided by the view synthesis algorithm is rather inaccurate or that the generated synthetic views are in some cases of poor quality, causing the algorithm to err with respect to the direction it shall move the robot. A similar behavior can be observed in the SD dataset, whereas in QMUL in the majority of cases 49.35% the algorithm decides to stay in the initial position whereas it moves away from the frontal direction in 45.73% of the cases (Table 5). Despite these issues, the algorithm manages to achieve good results in most cases.

**Table 5** Active Face Recognition Statistics: move type (4 steps)

Move Type	HPID [38]	QMUL [39]	SD [40]
<i>towards frontal</i>	<b>447</b> (59.67%)	57(4.9%)	<b>67</b> (58.26%)
<i>stay</i>	117(15.62%)	<b>573</b> (49.35%)	28(24.34%)
<i>away from frontal</i>	155(20.69%)	531(45.73%)	20(17.39%)
<i>total</i>	749	1161	115

## 4.5 Simulation Results

In order to provide simulation-based evidence that the proposed active vision method is indeed effective, we created a simple simulation environment using the open source and widely used Webots [41],[42] robotic simulator. The environment implements<sup>1</sup> a face recognition scenario that involves a TIAGo mobile manipulator robot<sup>2</sup> and its RGB camera. The TIAGo model<sup>2</sup> provided with the simulator moves in a circle around a static human model and performs face recognition using the proposed active method (2 steps approach). The method involves the same face detector and recogniser used in the experiments performed on the three datasets. The Database split G of the face recogniser contains facial images from 10 subjects with pan  $[0^\circ, \pm 15^\circ]$  and tilt  $0^\circ$ . In the implemented scenario, the robot is placed (initialized) at a random location approximately 2m away from the subject and performs active recognition on the face detected in the frames of its camera.

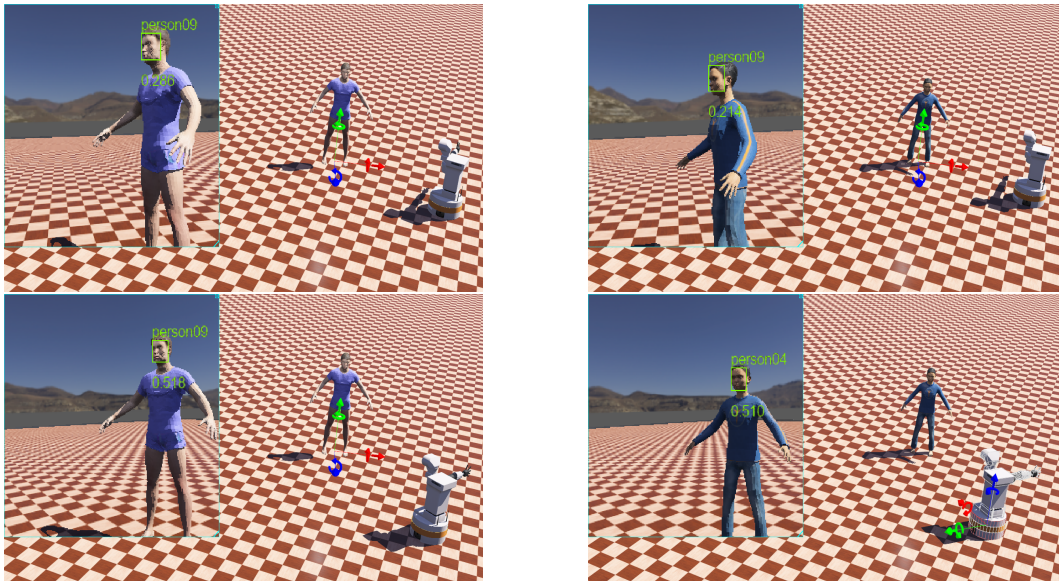


**Fig. 6** A simulation in Webots where a TIAGo mobile robot performs active face recognition on person 06 starting from two different initial robot positions (top row) and reaching its final position (respective image at the bottom row). The sub-image at the upper-left corner depicts the robot camera view along with the face detection bounding box, person label and recognition confidence.

<sup>1</sup><https://pal-robotics.com/robots/tiago/>

<sup>2</sup><https://cyberbotics.com/doc/guide/tiago-steel>





**Fig. 7** Additional simulation examples involving persons 09 (left) and 04 (right). Top row: initial robot positions. Bottom row: final robot position. Notice that in the case of person 04 the robot recognizes a different one (person 09) in its initial position, however this is corrected in its final position.

As illustrated in Figures 6 and 7 the robot moves towards more frontal views, increasing the recognition confidence and, in one case (Figure 7 right), changes its decision regarding the person's identity, towards the correct one.

## 5 Discussion and Conclusions

An active face recognition approach that utilizes facial views produced by facial image synthesis was presented in this paper. The camera-equipped robot that performs the recognition selects the best among a number of candidate physical movements around the person of interest by simulating their results through view synthesis. Experimental results show that the proposed method is superior to both its static version and face recognition that involves synthetically generated images. Moreover, it achieves significantly better results than the method in [7].

It shall be noted that certain assumptions were adopted in this paper and, furthermore, a number of issues were not fully addressed. First of all, the actual control of the robot in order to move in  $\theta^\circ$  increments around the person is not dealt with, being outside the scope of the paper. However, a rough estimate of the person position with respect to the robot would suffice to enable robot control. Also, it was assumed that the person being recognized remains relatively static during the recognition process, which can be a fair assumption if this process is brief or in situations when the person is sitting or lying on a bed. In case the person moves during this process, this shall be taken into account by the algorithm.

Moreover, it was assumed that there are no obstacles in the robot path. If this is not the case, these obstacles shall be detected (e.g. by a depth sensors)

and taken into account when planning the next move. Furthermore, obstacles in the space between the robot and the person might occlude the person for certain robot-person relative positions. However, since the algorithm decides on the person's identity based on the acquired image where the recognizer obtained the largest recognition confidence, it is rather safe to assume that, in most such cases, the algorithm might not face serious problems, even if it has instructed the robot to move in positions where occlusions occur.

Regarding algorithm performance, as mentioned in the previous section, there is a significant number of cases where the algorithm instructs the robot to move in a direction that is not towards a more frontal view. This might be attributed to errors of the view angle estimation and view synthesis algorithms. Using a better algorithm of this type might possibly lead to even bigger improvements with respect to the static approach. Another useful observation is that, giving the robot the freedom to move for additional steps (4 instead of 2) does, in two of the three datasets, significantly improve the recognition accuracy.

In the future, we plan to evaluate the proposed algorithm in larger datasets and extend the Webots simulation in order to investigate some of the issues mentioned above (occlusions, objects that hinder robot motion etc). Comparison of our approach to additional methods is also planned.

## Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 871449 (OpenDR). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

## References

- [1] Miguel Mendoza, J Irving Vasquez-Gomez, Hind Taud, L Enrique Sucar, and Carolina Reta. Supervised learning of the next-best-view for 3D object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020.
- [2] Jeffrey Delmerico, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza. A comparison of volumetric information gain metrics for active 3D object reconstruction. *Autonomous Robots*, 42(2):197–208, 2018.
- [3] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3D reconstruction. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE, 2016.
- [4] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Appearance-based active, monocular, dense reconstruction for micro aerial vehicles. In

*Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

- [5] J Irving Vasquez-Gomez, David Troncoso, Israel Becerra, Enrique Sucar, and Rafael Murrieta-Cid. Next-best-view regression using a 3d convolutional neural network. *Machine Vision and Applications*, 32(2):1–14, 2021.
- [6] Masaki Nakada, Han Wang, and Demetri Terzopoulos. Acfr: Active face recognition using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 35–40, 2017.
- [7] Nikolaos Passalis and Anastasios Tefas. Leveraging active perception for improving embedding-based deep face recognition. In *Proceedings of IEEE 22nd International Workshop on Multimedia Signal Processing (MMSp)*, pages 1–6. IEEE, 2020.
- [8] Rui Zeng, Yuhui Wen, Wang Zhao, and Yong-Jin Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 6(3):225–245, 2020.
- [9] Dorian Rohner and Dominik Henrich. Using active vision for enhancing an surface-based object recognition approach. In *Proceedings of Fourth IEEE International Conference on Robotic Computing (IRC)*, pages 375–382. IEEE, 2020.
- [10] Kai Xu, Yifei Shi, Lintao Zheng, Junyu Zhang, Min Liu, Hui Huang, Hao Su, Daniel Cohen-Or, and Baoquan Chen. 3D attention-driven depth acquisition for object identification. *ACM Transactions on Graphics (TOG)*, 35(6):1–14, 2016.
- [11] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.
- [12] Berk Calli, Wouter Caarls, Martijn Wisse, and Pieter P Jonker. Active vision via extremum seeking for robots in unstructured environments: Applications in object recognition and manipulation. *IEEE Transactions on Automation Science and Engineering*, 15(4):1810–1822, 2018.
- [13] Edward Johns, Stefan Leutenegger, and Andrew J Davison. Pairwise decomposition of image sequences for active multi-view recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3813–3822, 2016.

- [14] Mohsen Malmir, Karan Sikka, Deborah Forster, Ian Fasel, Javier R Movellan, and Garrison W Cottrell. Deep active object recognition by joint label and action prediction. *Computer Vision and Image Understanding*, 156:128–137, 2017.
- [15] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011.
- [16] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429–446, 2004.
- [17] GCHE de Croon, Ida G Sprinkhuizen-Kuyper, and Eric O Postma. Comparing active vision models. *Image and Vision Computing*, 27(4):374–384, 2009.
- [18] Weixing Peng, Yaonan Wang, Zhiqiang Miao, Mingtao Feng, and Yongpeng Tang. Viewpoints planning for active 3-d reconstruction of profiled blades using estimated occupancy probabilities (EOP). *IEEE Transactions on Industrial Electronics*, 68(5):4109–4119, 2020.
- [19] Qingyan Duan and Lei Zhang. Look more into occlusion: Realistic face frontalization and recognition with boostgan. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):214 – 228, January 2021.
- [20] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision, (ICCV)*, pages 2439–2448, 2017.
- [21] Jiashu Liao, Alex Kot, Tanaya Guha, and Victor Sanchez. Attention selective network for face synthesis and pose-invariant face recognition. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 748–752. IEEE, 2020.
- [22] Huan Tu, Gesang Duoqi, Qijun Zhao, and Shuang Wu. Improved single sample per person face recognition via enriching intra-variation and invariant features. *Applied Sciences*, 10(2):601, 2020.
- [23] Iacopo Masi, Tal Hassner, Anh Tuân Tran, and Gérard Medioni. Rapid synthesis of massive face sets for improved face recognition. In *Proceedings of 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 604–611. IEEE, 2017.
- [24] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-Render: Unsupervised photorealistic face rotation from single-view

- images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5911–5920, 2020.
- [25] Jianzhu Guo, Xiangyu Zhu, and Zhen Lei. 3DDFA. <https://github.com/cleardusk/3DDFA>, 2018.
- [26] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3D dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 152–168. Springer International Publishing, 2020.
- [27] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):78–92, 2017.
- [28] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3907–3916, 2018.
- [29] Qingzhong Wang, Pengfei Zhang, Haoyi Xiong, and Jian Zhao. Face. evolve: A high-performance face recognition library. *arXiv preprint arXiv:2107.08621*, 2021.
- [30] Nikolaos Passalis, Stefania Pedrazzi, Robert Babuska, Wolfram Burgard, Daniel Dias, Francesco Ferro, Moncef Gabbouj, Ole Green, Alexandros Iosifidis, Erdal Kayacan, Jens Kober, Olivier Michel, Nikos Nikolaidis, Paraskevi Nousi, Roel Pieters, Maria Tzelepi, Abhinav Valada, and Anastasios Tefas. OpenDR: An Open Toolkit for Enabling High Performance, Low Footprint Deep Learning for Robotics. *arXiv preprint arXiv:2203.00403*, 2022.
- [31] OpenDR: A modular, open and non-proprietary toolkit for core robotic functionalities by harnessing deep learning. <https://github.com/opendr-eu/opendr>. Accessed: 2022-06-27.
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [33] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019.
- [34] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.

- [35] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5203–5212, 2020.
- [36] OpenDR Face Detection module: RetinaFace. <https://github.com/opendr-eu/opendr/blob/master/docs/reference/face-detection-2d-retinaface.md>. Accessed: 2022-06-27.
- [37] Gary B. Huang Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. Technical Report UM-CS-2014-003, University of Massachusetts, Amherst, May 2014.
- [38] Nicolas Gourier, Daniela Hall, and James L Crowley. Estimating face orientation from robust detection of salient facial structures. In *Proceedings of Workshop on Visual Observation of Deictic Gestures*, volume 6, page 7. FGnet (IST–2000–26434) Cambridge, UK, 2004.
- [39] Jamie Sherrah and Shaogang Gong. Fusion of perceptual cues for robust tracking of head pose and position. *Pattern Recognition*, 34(8):1565–1572, 2001.
- [40] Charalampos Georgiadis. Generation of a synthetic annotated dataset for training and evaluating active perception methods. *BSc Thesis, Aristotle University of Thessaloniki*, 2022, doi : 10.13140/RG.2.2.21002.34248.
- [41] Webots. <http://www.cyberbotics.com>. Open-source Mobile Robot Simulation Software.
- [42] O. Michel. Webots: Professional mobile robot simulation. *Journal of Advanced Robotics Systems*, 1(1):39–42, 2004.

# Active Face Recognition through View Synthesis

Efstratios Kakaletsis, Nikos Nikolaidis

*Department of Informatics, AIIA Laboratory, Aristotle University of Thessaloniki*

*Thessaloniki, Greece, GR-54124*

*Email: {ekakalets, nnik}@csd.auth.gr*

**Abstract**—Active vision exploits the ability of robots to interact with their environment, towards increasing the quantity / quality of information obtained through their sensors and, therefore, improving their performance in perception tasks. Active face recognition is largely understudied in recent literature. In this paper, we propose an active approach that utilizes facial views produced by facial image rendering. The robot that performs face recognition selects the best candidate rotation around the person of interest by simulating the results of such movements through view synthesis. This is achieved by passing to the robot’s face recognizer a real world facial image acquired in the current position, generating synthesized views that differ by  $\pm\theta^\circ$  from the current view. Then, it decides, on the basis of the confidence of the recognizer, whether to stay in place or move to the position that corresponds to one of the two synthesized views, so as to acquire a new real image. Experimental results in two datasets verify the superior performance of the proposed method compared to the respective static approach and an approach based on the same face recognizer that involves face frontalization with synthesized views.

**Index Terms**—active vision, active face recognition, synthesized facial views, photorealistic facial synthesis

## I. INTRODUCTION

Recently the robotics and computer vision communities have started researching more thoroughly the field of active vision / perception and exploration [1]. Active perception methods try to obtain more, or better quality, information from the environment by actively choosing from where, when and how to observe it using a camera (or other sensors), in order to accomplish more effectively tasks such as 3D reconstruction [2], [3], [4], [5], [6] or object recognition [7], [8]. This could be achieved, for example, by moving a camera-equipped mobile robot, e.g. a wheeled robot or a UAV, in positions which provide different, hopefully better, views of the object of interest. Although active 3D object reconstruction has attracted considerable interest, mainly towards tackling the “next-best-view” problem (choosing the next viewing position so as to obtain a detailed and complete 3D object model), active approaches for recognition tasks, particularly for face recognition, are much less frequent in the literature. Deep Learning dominates face recognition research due to its superior performance. However the vast majority of recognition approaches adopt a static approach i.e., an approach that is based on an image acquired from a certain viewpoint, even in

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement number 871449 (OpenDR). This publication reflects only the authors views. The European Union is not liable for any use that may be made of the information contained therein.

setups where an active approach could have been used. Indeed, face recognition can be combined with an active approach for directing the movement of a robot towards capturing the face from more informative views and thus obtain more robust results, at the expense of energy consumption and additional decision time. Synthesized views of faces, whose images were captured through a camera, can be used for robot movement guidance in an active face recognition setup. Instead of having the robot move in a physical way for capturing a novel view, one can use a synthesized view as an aid towards choosing a new viewpoint and improving recognition.

In this paper, we propose an active face recognition approach that utilizes facial views synthesized by photorealistic facial image rendering. Essentially, the camera-equipped robot that performs the recognition selects the best among a number of candidate physical movements (rotations) around the face of interest by simulating their results through view synthesis. In other words, once the robot (that is at a certain location with respect to the subject) acquires an image, it provides the face recognizer with this image as well as with synthesized views that differ by  $\pm\theta^\circ$  from the current view. Subsequently, it either stays in the current position or moves to the position that corresponds to one of the two synthesized views. The respective decision is based on the confidence of the three recognitions (on the real and the two synthesized views). In the case of a “move” decision, it proceeds to acquire a “real” image from its new location. The procedure repeats in the same manner, for this location, for one or more steps. Using synthesized facial views facilitates the decision-making procedure by providing estimates of what is to be expected in a new robot position. The proposed method involves a face recognizer that is trained to recognize frontal or nearly frontal faces, a fact that facilitates its real-world application. Despite this, it can recognize successfully input facial images obtained from an arbitrary view point, since it utilizes the ability of a robot to move in order to capture more informative views of the subject. This gives it advantage over static approaches. Indeed, although (static) face recognition is a very mature technology, such approaches can operate successfully only if they have been trained to recognize the subject from view angles similar to the one of the input image. This requires that such methods are trained with images captured from a large number of view angles. In contrast, the proposed approach needs to “know” only frontal or nearly frontal views.

The contributions of the paper can be summarized as follows: a) to our knowledge this is one of a very few (2-

3) works that deal with active face recognition; b) as far as we are aware, this is the first time facial image synthesis is utilized in an active face recognition setup; c) the method requires no training and the involved face recognizer needs to "know" only frontal images of the subjects, and d) the presented results show that the proposed approach performs better than the respective static approach and an approach that involves face frontalization.

## II. RELATED WORK

Despite the fact that active object recognition has attracted considerable interest in the computer vision and robotics communities, active face recognition has been scarcely studied. Such a simple method is described in [7] and comprises of a neural network-based face recognizer along with a decision making controller that decides for the viewpoint changes. The authors in [8] propose a deep learning-based active perception method for embedding-based face recognition and examine its behavior on a real multi-view face image dataset. The proposed approach can simultaneously extract discriminative embeddings, as well as predict the action that the robot must take (stay in place, move left or right by a certain amount, on a circle centered at the person) in order to get a more discriminative view.

A significant number of techniques for synthesizing facial images in novel views appeared in the last years since such images can have a number of applications, e.g., in improving face recognition accuracy. For example, since profile faces usually provide inferior recognition results compared to frontal faces, generative adversarial networks (GANs) based methods for the frontalization of profile facial images [9] or generation of other facial views [10] have been proposed for improving face recognition results. A method for the generation of frontal views from any input view that utilizes a novel generative adversarial architecture (ASN) is described in [11]. Towards improving single-sample face recognition by both generating additional samples and eliminating the influence of external factors (illumination, pose), [12] presents an end-to-end network for the estimation of intrinsic properties of a facial image. In [13], a facial image rendering technique is used both in the training and testing stages of a face recognition approach. A method that produces photorealistic facial image views is described in [14]. The basic idea of this approach is that rotating faces in the 3D space and re-rendering them to the 2D plane can serve as a strong self-supervision. A 3D head model (obtained by utilizing the 3D-fitting network 3DDFA [15] accompanied by the projected facial texture of a single view, is being rotated and multi-view images of the face are rendered using the Neural 3D Differential Renderer [16] along with 2D-to-3D style transfer and image-to-image translation with GANs to fill in invisible parts. This last state-of-the-art method was selected due to its robustness and photorealistic quality for the generation of the synthetic facial images required by the method proposed in this paper.

Although facial view synthesis can improve face recognition performance, active perception methods can be expected to provide better results, in cases where acquisition of additional

real world facial views is possible due to the existence of e.g. a wheeled robot.

## III. PROPOSED ALGORITHM

### A. Face Recognition

Let us denote as database subset  $G$  a set of training facial images for the persons that shall be recognized. Similarly, the facial images to feed the face recognizer are included in the query (test) set  $T$ . The face recognition library face.evoLve [17] which contains many state of the art deep face recognition models, is used. More specifically, an implementation of a certain face recognition approach of face.evoLve from the OpenDR Toolkit<sup>1</sup> [18] was used. IR-50 (50 layers) [19] trained on MS-CELEB-1M using an ArcFace [20] loss head was used as the 512-dimensional feature extraction backbone. For the database subset  $G$ , face detection, facial landmark extraction and face alignment was based on the face.evoLve module that is based on MTCNN [21], whereas for the query images in  $T$ , these processing steps were based on RetinaFace [22]. Face recognition is performed by a nearest-neighbor classifier that uses Euclidean distance in the 512-dimensional feature space to find the database facial image that best matches the query image. Face recognition confidence  $FRC \in [0, 1]$ , is also evaluated based on the distance between the input query image and the nearest image in the database  $G$ . The  $FRC$  is given by the following formula:

$$FRC = 1 - \frac{distance}{max\_distance} \quad (1)$$

where  $distance$  is the Euclidean distance of query facial image from the nearest neighbor image in the database  $G$  and  $max\_distance$  is the maximum such distance.

### B. Active Face Recognition Through Synthesized Views

The proposed active face recognition algorithm uses the face recognition confidence  $FRC$  and facial images synthesized for view angles around the current robot view, in order to select the next robot movement, towards performing a successful recognition. Starting from an initial position, the robot can take one of the following three decisions: stay at the current position, move by  $\theta^\circ$  to the right or move by  $\theta^\circ$  to the left, on a circle centered at the person that is to be recognized, in order to acquire a new image. Depending on the achieved recognition confidence, an additional movement, towards the same direction as the first one, might be decided. More specifically, given a facial query image  $I_r$  (subscript  $r$  stands for real), captured by the robot camera at the robot starting position, the face synthesis algorithm [14] is utilized to estimate the view angle and then render/generate facial views in 2 different view angles i.e.  $-15^\circ$  and  $+15^\circ$  in pan with respect to the pan of  $I_r$  (and the same tilt as  $I_r$ ). These two images are denoted by  $I_s^-$  and  $I_s^+$  respectively (subscript  $s$  stands for synthetic). Then, the face recognizer is fed with these three images  $I_r, I_s^-, I_s^+$  (one real, two synthetic ones). Depending on the image that obtained the biggest face recognition confidence  $FRC$ , the robot stays in its current position (if  $FRC$  was maximum

<sup>1</sup>OpenDR Toolkit: <https://github.com/opendr-eu/opendr>



in  $I_r$ ) or physically moves  $-15^\circ$  (or  $+15^\circ$ ) (if  $FRC$  was maximum in  $I_s^-$  (or  $I_s^+$ )) and acquires through its camera a new real image  $I_r^-$  (or  $I_r^+$ ). If a "stay" decision was taken, the algorithm outputs the ID of the person it recognized in  $I_r$  and terminates. If the robot moved, face recognition is performed again in  $I_r^-$  (or  $I_r^+$ ) and the obtained  $FRC$  is compared to an experimentally evaluated threshold  $t$ . In case a high enough confidence was observed, the algorithm outputs the ID of the person it recognized in  $I_r^-$  (or  $I_r^+$ ) and terminates. If not, it tries yet another  $15^\circ$  step (movement) in pan, in the same direction as the first step. In more detail, in this second step, it generates/synthesizes a facial view  $-15^\circ$  (or  $+15^\circ$ ) in pan from the current pan value (and the same tilt), denoted as  $I_s^{--}$  (or  $I_s^{++}$ ), and evaluates (by calling the face recogniser)  $FRC$  on this synthetic image. If  $FRC(I_r^-) > FRC(I_s^{--})$  (or  $FRC(I_r^+) > FRC(I_s^{++})$ ) the algorithm decides that the robot shall stay in its current position, outputs the ID of the person it recognized in  $I_r^-$  (or  $I_r^+$ ) and terminates. Otherwise, the robot physically moves  $-15^\circ$  ( $+15^\circ$ ) from its current position, acquires a new image  $I_r^{--}$  ( $I_r^{++}$ ) and the algorithm outputs the ID of the person it recognized in this image.

The performance of the proposed procedure obviously depends on whether the synthesis algorithm [14] estimates with sufficient accuracy the view angle of the query image  $I_r$  and also on whether the synthesized views are of good quality. In order to limit the possibly negative effect of these factors on the performance of the algorithm (e.g. by leading it to move towards the wrong direction), the algorithm does not actually take a decision based on the last real image it has visited but does so based on the real image where it has obtained the maximum  $FRC$  value. In more detail, if the algorithm took one step of  $-15^\circ$ , it takes a decision using the real image  $I$  given by:

$$I = \underset{x \in \{I_r^-, I_r\}}{\operatorname{argmax}} (FRC(x)) \quad (2)$$

or the equivalent expression that involves  $I_r^+$ ,  $I_r$ , if a step of  $+15^\circ$  has been taken. Similarly, if two steps of  $-15^\circ$  each have been performed, the algorithm decides on the person ID using the real image  $I$  given by:

$$I = \underset{x \in \{I_r^{--}, I_r^-, I_r\}}{\operatorname{argmax}} (FRC(x)) \quad (3)$$

or the equivalent expression that involves  $I_r^{++}$ ,  $I_r^+$ ,  $I_r$ , if two steps, of  $+15^\circ$  each, have been taken. The pseudocode is presented in algorithm 1.

It should be noted that the actual recognition is always performed on a real image, i.e., an image captured by the robot camera. The synthesized views are only used to aid the robot in deciding whether to move in a new position (and acquire a new image there) or stay in the current position. The rationale behind the proposed approach is that in case the initial robot position is far from a frontal or nearly frontal one, the algorithm will hopefully direct it to move towards a position which is closer to a frontal one. Obviously, the procedure can be generalized to include additional steps (movements), i.e., more than the two movements it currently has. It can also work, in the same way, for tilt.

---

#### Algorithm 1 Active Face Recognition Algorithm (2 steps) on Pseudocode

---

**Input:**  $I_r$ ,  $threshold$ ,  $\theta^\circ$

**Result:**  $Person_{ID}(I_r)$

```

1:  $\alpha = Estimate\_View\_Angle(I_r)$ 
2:  $I_s^- = Render(\alpha - \theta^\circ, I_r)$ 
3:  $I_s^+ = Render(\alpha + \theta^\circ, I_r)$ 
4:  $I = \operatorname{argmax}(FRC(x))$ 
    $x \in \{I_r, I_s^-, I_s^+\}$ 
5: if  $I = I_r$  then
6:    $I_{ID} = I_r$ 
7:   go to 28
8: else
9:   if  $I = I_s^+$  then
10:     $\theta_{incr} = +\theta^\circ$ 
11:   else
12:     $\theta_{incr} = -\theta^\circ$ 
13:
14:  $I_r^{1step} = Move\_and\_Capture(\alpha + \theta_{incr})$ 
15: if  $FRC(I_r^{1step}) > threshold$  then
16:    $I_{ID} = \operatorname{argmax}(FRC(x))$ 
      $x \in \{I_r, I_r^{1step}\}$ 
17:   go to 28
18: else
19:    $I_s^{2step} = Render(\alpha + 2 * \theta_{incr}, I_r^{1step})$ 
20:   if  $FRC(I_s^{2step}) < FRC(I_r^{1step})$  then
21:     $I_{ID} = \operatorname{argmax}(FRC(x))$ 
      $x \in \{I_r, I_r^{1step}\}$ 
22:    go to 28
23:   else
24:     $I_r^{2step} = Move\_and\_Capture(\alpha + 2 * \theta_{incr})$ 
25:     $I_{ID} = \operatorname{argmax}(FRC(x))$ 
      $x \in \{I_r, I_r^{1step}, I_r^{2step}\}$ 
26:    go to 28
27:
28:  $Person_{ID}(I_r) = Recognize(I_{ID})$ 

```

---

## IV. EXPERIMENTAL EVALUATION

For the evaluation of the proposed active approach experiments were conducted using the HPID dataset [23] and the Queen Mary University of London Multi-view Face Dataset (QMUL) [24]. In the two datasets, images of all subjects were divided into two non-overlapping subsets: a database subset  $G$  (images that the face recognizer uses to decide the ID of the query image through the nearest neighbor classifier) and a query (test) subset  $T$  (which includes the images captured by the robot camera in its initial position). Obviously  $G$  and  $T$  contained images from different pan ranges. This setup was adopted in order to simulate active recognition where the robot is moving only in the pan direction. Concise descriptions of the two datasets are provided below.

### A. Datasets

The HPID dataset [23] is a head pose image dataset that consists of 2790 face images of 15 subjects captured by varying the pan and tilt from  $-90^\circ$  to  $+90^\circ$ , in increments of  $\theta = 15^\circ$ . Two sets of images were captured for

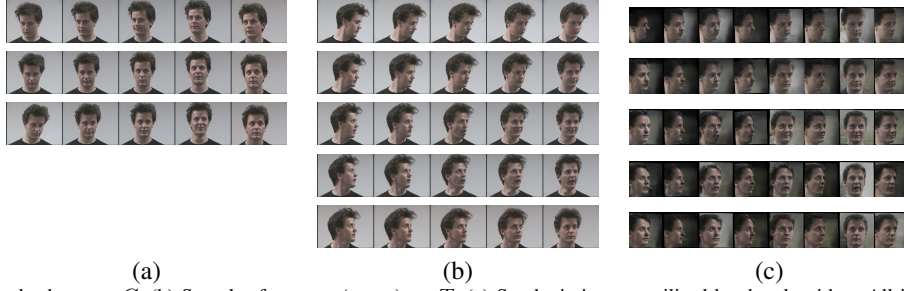


Fig. 1. (a) Samples from database set  $G$ , (b) Samples from test (query) set  $T$ , (c) Synthetic images utilised by the algorithm. All images are from the HPID dataset.

each person (93 images in each set). The database subset  $G$  (Figure 1.a) contains facial images with tilt in angles  $[-30^\circ, -15^\circ, 0^\circ, +15^\circ, +30^\circ]$  and pans  $[-15^\circ, 0^\circ]$ , i.e., only nearly frontal images. The query subset  $T$  (Figure 1.b) contains face images with tilts  $[-30^\circ, -15^\circ, 0^\circ, +15^\circ, +30^\circ]$  and pans  $[-90^\circ, -75^\circ, -60^\circ, -45^\circ, -30^\circ]$ . The selection of the range  $[-90^\circ \dots -30^\circ]$  in pan, instead of the full (i.e.,  $[-90^\circ \dots -30^\circ]$  and  $[+30^\circ \dots +90^\circ]$ ) semi-circle, in this and the QMUL dataset, was just for simplicity. Similar results were obtained when the experiments involved the entire semi-circle.

Queen Mary University of London Multi-view Face Dataset (QMUL) [24] consists of automatically aligned, cropped and normalised face images of 48 persons. Images of 37 persons are in greyscale (100x100 pixels) whereas those of the remaining 11 persons are in colour and of dimensions 56x56 pixels. For each person 133 facial images exist, populating a viewsphere of  $-90^\circ \dots +90^\circ$  in pan and  $-30^\circ \dots +30^\circ$  in tilt in  $\theta = 10^\circ$  increments. For the Database split  $G$ , images with pan in angles  $[-10^\circ, 0^\circ]$  and tilt in the range  $[-30^\circ, \dots, +30^\circ]$  were used. The Query split  $T$  includes images with pan in angles  $[-90^\circ, \dots, -20^\circ]$  and tilt in the range  $[-30^\circ, \dots, +30^\circ]$ .

### B. Experimental Results

The results (in terms of recognition accuracy) are presented in Table I. The line marked "Static" presents the result of the static equivalent of our approach, in which only the initial query facial image is used by the same recogniser involved in the active approach. As can be seen, the proposed active method, implemented to perform up to 4 steps (line "Proposed (Active) (4 steps)") outperforms its static counterpart, increasing the recognition accuracy by 15.61% and 12.97% (absolute increase) in HPID and QMUL datasets, respectively.

The proposed approach was also compared to the frontalization approach that is often used in face recognition when the recognizer is trained only on frontal views. In this case, the facial view synthesis algorithm [14] is used in order to generate a frontal ( $0^\circ$  in pan) view from the input (query) image. This image is then provided to the recognizer. The results (line "Frontalization (synthetic frontal views)") show that although frontalization achieves improved performance with respect to the static approach, it is clearly superseded by the proposed active approach.

Statistics regarding the steps taken by the proposed approach were also evaluated and are presented in Table II for HPID dataset. These statistics show that in 26.48% of the cases the

TABLE I  
FACE RECOGNITION ACCURACY RESULTS AND COMPARISON WITH THE STATIC APPROACH AND OTHER VARIANTS

Method	HPID [23]	QMUL [24]
Static (non-active, only queries)	72.49 %	69.88%
Proposed (Active) (4 steps)	<b>88.10%</b>	<b>82.85%</b>
Frontalization (synthetic frontal views)	80.75%	75.95%

TABLE II  
ACTIVE FACE RECOGNITION STATISTICS (4 STEPS, HPID DATASET): STEPS PERFORMED BY THE ALGORITHM.

Image type	Angle	# Images	Percentage
$I_r$	$0^\circ$	397	26.48%
$I_r^+$	$+15^\circ$	368	24.54%
$I_r^{++}$	$+30^\circ$	84	5.603%
$I_r^{+++}$	$+45^\circ$	0	0%
$I_r^{++++}$	$+60^\circ$	1	0.066%
$I_r^-$	$-15^\circ$	515	34.35%
$I_r^{--}$	$-30^\circ$	121	8.07%
$I_r^{---}$	$-45^\circ$	9	0.600%
$I_r^{----}$	$-60^\circ$	5	0.333%
Total	-	1500	100%

robot decided to stay in its initial position whereas in the remaining 73.64% it moved by  $\pm 15^\circ, \dots, \pm 60^\circ$  (one to four steps). It shall be noted however that the decision on the ID of the depicted person is not necessarily obtained from the last position the robot has visited, due to the fact that the image with the maximum recognition confidence ( $FRC$ ) is used for this purpose (equations (2) and (3)).

The average number of movements that the algorithm instructs the robot to perform can be easily evaluated from statistics such as the ones presented in Table II. Based on these calculations, the algorithm instructs the robot to make, on average, 0.76 (HPID) or 0.89 (QMUL) movements, a fact that signifies that the time required for active recognition (time for the computations as well as the time for the robot to move) is relatively low. Note that in case the robot decides to perform no movement (stay decision) the number of movements is obviously zero.

### V. DISCUSSION AND CONCLUSIONS

An active approach for face recognition that utilizes facial views produced by facial image synthesis was presented in this paper. The robot that performs the recognition selects the best among a number of candidate physical movements around the person of interest by simulating their results through view synthesis. Experimental evaluation showed that the method supersedes both its static version and face recognition that involves frontalization through synthesis of frontal images.

It must be stressed that certain assumptions were adopted in this paper, whereas a number of issues were not fully addressed. First, the actual control of the robot so as to move in

$\theta^\circ$  increments on a circle around the person was not dealt with, since it falls outside the scope of the paper. However, a rough estimate of the person position with respect to the robot would suffice to enable robot control. Also, it was assumed that the person being recognized remains relatively static during the recognition process, which can be an acceptable assumption if the process is brief. However, if the person moves, this shall be taken into account by the algorithm. It shall be also noted that the (mild) requirement for a static face is indeed satisfied in certain cases that include sitting or lying persons, as in a healthcare environment, or an elderly care establishment, where a service robot operates in order to aid the inhabitants.

In addition, it was assumed that there are no obstacles in the robot's path. If this is not the case, these obstacles shall be detected (by e.g. depth sensors) and taken into account. Furthermore, obstacles in the space between the robot and the person might occlude the person for certain robot positions. However, since the algorithm decides on the person's identity based on the acquired image where the recognizer obtained the largest recognition confidence, it is rather safe to assume that, in most such cases, the algorithm might not face serious problems, even if it has instructed the robot to move in positions where occlusions occur.

One could also consider, instead of using the synthesized views as proposed in this paper, to estimate the view angle of the robot camera with respect to the person and instruct it to move directly (namely, without intermediate steps) to the position that would allow it to capture a frontal view ( $0^\circ$  in pan). However, there are certain issues that make this approach difficult in practice. Indeed, we observed in the experiments that view angle estimates (i.e., those provided by the view synthesis algorithm used) although accurate enough for the purposes of view synthesis, are quite far from the ground truth values, thus rendering this approach problematic. Experiments, which are omitted due to lack of space, verified that such an approach indeed leads to inferior results.

Future plans include evaluation of the algorithm in additional datasets and creation of a realistic simulation so as to investigate some of the issues mentioned above (occlusions, actual robot control, objects that hinder robot motion etc). Employing a more sophisticated face recognizer and comparing it to additional methods, are also planned.

## REFERENCES

- [1] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.
- [2] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta, "Supervised learning of the next-best-view for 3D object reconstruction," *Pattern Recognition Letters*, 2020.
- [3] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.
- [4] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3477–3484.
- [5] C. Forster, M. Pizzoli, and D. Scaramuzza, "Appearance-based active, monocular, dense reconstruction for micro aerial vehicles," *Robotics: Science and Systems Conference, University of California, Berkeley, USA, July 12-16*, 2014.
- [6] J. I. Vasquez-Gomez, D. Troncoso, I. Becerra, E. Sucar, and R. Murrieta-Cid, "Next-best-view regression using a 3D convolutional neural network," *Machine Vision and Applications*, vol. 32, no. 2, pp. 1–14, 2021.
- [7] M. Nakada, H. Wang, and D. Terzopoulos, "AcFR: Active face recognition using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 35–40.
- [8] N. Passalis and A. Tefas, "Leveraging active perception for improving embedding-based deep face recognition," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSp)*. IEEE, 2020, pp. 1–6.
- [9] Q. Duan and L. Zhang, "Look More Into Occlusion: Realistic Face Frontalization and Recognition with BoostGAN," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [10] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis," in *IEEE International Conference on Computer Vision*, 2017, pp. 2439–2448.
- [11] J. Liao, A. Kot, T. Guha, and V. Sanchez, "Attention selective network for face synthesis and pose-invariant face recognition," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 748–752.
- [12] H. Tu, G. Duoji, Q. Zhao, and S. Wu, "Improved single sample per person face recognition via enriching intra-variation and invariant features," *Applied Sciences*, vol. 10, no. 2, p. 601, 2020.
- [13] I. Masi, T. Hassner, A. T. Tran, and G. Medioni, "Rapid synthesis of massive face sets for improved face recognition," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 604–611.
- [14] H. Zhou, J. Liu, Z. Liu, Y. Liu, and X. Wang, "Rotate-and-Render: Unsupervised Photorealistic Face Rotation from Single-View Images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5911–5920.
- [15] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3D dense face alignment," in *European Conference on Computer Vision (ECCV)*, 2020.
- [16] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D mesh renderer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3907–3916.
- [17] Q. Wang, P. Zhang, H. Xiong, and J. Zhao, "Face. evolVe: A high-performance face recognition library," *arXiv preprint arXiv:2107.08621*, 2021.
- [18] N. Passalis, S. Pedrazzi, R. Babuska, W. Burgard, D. Dias, F. Ferro, M. Gabbouj, O. Green, A. Iosifidis, E. Kayacan, J. Kober, O. Michel, N. Nikolaidis, P. Nousi, R. Pieters, M. Tzelepi, A. Valada, and A. Tefas, "OpenDR: An Open Toolkit for Enabling High Performance, Low Footprint Deep Learning for Robotics," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12479–12484, 2022.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [20] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *IEEE/CVF conference on Computer Vision and Pattern Recognition, (CVPR)*, 2019, pp. 4690–4699.
- [21] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [22] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2020, pp. 5203–5212.
- [23] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial structures," in *FG Net workshop on visual observation of deictic gestures*, vol. 6. FGnet (IST-2000-26434) Cambridge, UK, 2004, p. 7.
- [24] J. Sherrah and S. Gong, "Fusion of perceptual cues for robust tracking of head pose and position," *Pattern Recognition*, vol. 34, no. 8, pp. 1565–1572, 2001.

## **8.2 Neural Attention-Driven Non-Maximum Suppression for Person Detection**

The appended paper [58] follows.

# Neural Attention-driven Non-Maximum Suppression for Person Detection

Charalampos Symeonidis, Ioannis Mademlis, Ioannis Pitas and Nikos Nikolaidis  
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

**Abstract**—Non-maximum suppression (NMS) is a post-processing step in almost every visual object detector. NMS aims to prune the number of overlapping detected candidate regions-of-interest (RoIs) on an image, in order to assign a single and spatially accurate detection to each object. The default NMS algorithm (GreedyNMS) is fairly simple and suffers from severe drawbacks, due to its need for manual tuning. A typical case of failure with high application relevance is pedestrian/person detection in the presence of occlusions, where GreedyNMS doesn't provide accurate results. This paper proposes an efficient deep neural architecture for NMS in the person detection scenario, by capturing relations of neighboring RoIs and aiming to ideally assign precisely one detection per person. The presented Seq2Seq-NMS architecture assumes a sequence-to-sequence formulation of the NMS problem, exploits the Multihead Scale-Dot Product Attention mechanism and jointly processes both geometric and visual properties of the input candidate RoIs. Thorough experimental evaluation on three public person detection datasets shows favourable results against competing methods, with acceptable inference runtime requirements.

**Index Terms**—Non-Maximum Suppression, Object Detection, Scaled-Dot Product Attention, Sequence-to-Sequence Learning, Person Detection, Deep Neural Networks

## I. INTRODUCTION

Non-Maximum Suppression (NMS) is a final refinement step incorporated to almost every visual object detection framework, assigned the duty of merging/filtering any spatially overlapping detected Regions-of-Interest (RoIs), i.e., bounding boxes, which correspond to the same visible object on an image. The problem it attempts to solve arises from the tendency of many detectors to output multiple, neighbouring candidate object RoIs for a single visible object, due to their implicit sliding-window nature. Thus, an NMS algorithm processes the raw object detector outputs identified on an input image and attempts to filter out the duplicate RoIs.

The de facto dominant NMS method for object detection is GreedyNMS. It selects high-scoring detections and deletes less confident neighbours, since they most likely cover the same object. Its simplicity, speed and unexpectedly good behaviour in most cases make it competitive against proposed alternatives, since rapid execution is very important for NMS. An Intersection-over-Union (IoU) threshold determines which less-confident neighbors are suppressed by a detection. This fixed IoU threshold leads GreedyNMS to failure in certain cases. Too powerful a suppression, using a low threshold, may

remove detections that cover different spatially overlapped objects, while a too high threshold may be unable to suppress duplicate detections.

Due to these limitations of traditional algorithms, modern Deep Neural Network (DNN)-based methods for performing NMS have emerged during the past few years. While some DNNs are assigned with auxiliary tasks complementing the original NMS scheme (e.g., estimate target density maps in order to apply dynamic suppression thresholding [1]), others provide a more straightforward solution (e.g., outputting a score for each candidate detection, thus indicating whether it corresponds to a “duplicate” detection or not [2]). The latter type of methods relies on building representations for each candidate detection, typically based on their corresponding geometric/spatial relations [2], while ignoring RoI visual appearance. This is either because CNN-based features can blur the boundaries between highly overlapping true positives and duplicates, or due to the difficulties DNNs are faced with when trying to extract accurate representations for highly occluded objects. However, evidence has recently surfaced indicating that appearance-based input may improve the performance of DNN-based NMS methods [3] [4], if that information is properly fused with the geometry-based input.

An additional issue stems from the fact that the NMS problem for object detection purposes is essentially sequential in nature. The output RoIs are sequentially processed by the common object detection evaluation protocols [5] [6], ordered according to the scalar confidence scores assigned to them by the NMS method. Similarly, the input candidate RoIs, i.e., the raw output of the object detector which is fed as input to the NMS algorithm, must also be ordered according to the initial confidence scores assigned to them by the detector. Thus, essentially, an NMS method actually decides whether a candidate RoI is duplicate, or not, based on the decisions it has previously taken for the preceding, higher-scoring candidate RoIs along the input sequence. However, to the best of our knowledge, NMS has not been previously explicitly formulated as a problem of processing sequences, thus related algorithms have not been applied to solving it.

Motivated by such issues of existing neural NMS approaches, this paper offers the following contributions:

- a novel reformulation of the NMS task for object detection as a sequence-to-sequence problem.
- a novel deep neural architecture for NMS, relying on the Scaled Dot-Product Attention mechanism, called *Seq2Seq-NMS*.

The source code is publicly available at: [https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object\\_detection\\_2d/nms/seq2seq\\_nms](https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/nms/seq2seq_nms).

- a new, fast, efficient and GPU-based neural implementation of the low-level Frame Moments Descriptor (FMoD) [7], which is employed for feeding the proposed DNN with appearance-based representations of detected candidate RoIs.

The proposed method is highly applicable to the person/pedestrian detection task, where most NMS algorithms face difficulties in identifying individuals in the presence of occlusions. The majority of existing NMS methods target fast execution, but person detection requires a high degree of accuracy; this is critical for ensuring human safety in domains such as autonomous systems [8] [9] [10] [11] [12] [13]. Moreover, the visual appearance representation approach adopted by Seq2Seq-NMS, i.e., FMoD descriptors computed on edge maps of cropped candidate RoIs, is most accurate in cases where the visible silhouette of the target object class remains approximately identical in shape across the training and test images. This is true in the person detection case, bar abnormally extensive viewpoint variations across the employed dataset. Adopting FMoD, which has already proven its worth in NMS for person detection from aerial viewpoints [3], renders the applicability of the proposed method focused to similar scenarios.

Extensive quantitative evaluation using well-known metrics and public person detection datasets indicates favourable results in comparison to several competing NMS methods, both neural and non-neural, leading to state-of-the-art results. The source code is publicly available at: [https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object\\_detection\\_2d/nms/seq2seq\\_nms](https://github.com/opendr-eu/opendr/tree/master/src/opendr/perception/object_detection_2d/nms/seq2seq_nms).

## II. RELATED WORK

NMS is the final step of typical object detection pipelines, thus this Section first briefly reviews state-of-the-art detectors. Subsequently, NMS algorithms and related loss functions are presented. Finally, the motivation behind the proposed method is discussed in the context of the existing approaches to NMS.

### A. Object Detection

Object detection is a long-standing, fundamental problem in computer vision. Its task is to generate bounding boxes (in 2D pixel coordinates) for objects detected on an image that belong to prespecified object classes and to assign classification scores to them. Most of the early object detection algorithms [14] [15] relied mainly on local handcrafted descriptors and discriminative classifiers. The Deformable Part-based Model (DPM) [16] is a special case, where an object is represented by its component parts arranged in a deformable configuration. In [17], the authors designed a joint person detector, based on the DPM architecture, which overcomes the limitations imposed by frequent occlusions in real-world street scenes.

Object detection has been tremendously improved thanks to Deep Neural Networks (DNNs), with Convolutional Neural Networks (CNNs) being the most relevant architectures. DNN-based object detectors are usually grouped into two categories: two-stage and one-stage object detectors. Typically,

the former ones (e.g., [18]) first create object proposals from input images, using a method such as selective search or a separate DNN, and then extract features from these proposals using CNNs. These features are then fed to a classifier that determines the existence and the class of any object in each proposal. Although two-stage detectors achieve state-of-the-art performance, their running speed is typically slow. One-stage object detectors, such as [19] [20] and [21] perform region proposal and object classification in a single, unified DNN. Initial regions are predefined bounding boxes with various scales and ratios placed densely on the image, which are generally referenced as anchors. From the initial anchors, the detectors find those that likely contain objects. Compared to two-stage detectors, their one-stage competitors are usually much faster, but less accurate.

### B. Non-Maximum Suppression

The de facto standard in NMS for object detection is GreedyNMS [22]. It selects high-scoring detections and deletes less confident neighbours, since they most likely cover the same object. An Intersection-over-Union (IOU) threshold determines which less-confident neighboring detections are suppressed. It is a simple, well-known, but limited method, leading to several attempts for replacing it with much improved alternatives.

In Soft-NMS [23], a rescaling function decreases the score of neighboring less-confident detections, instead of completely eliminating them, achieving better precision and recall rates compared to GreedyNMS. The authors experiment with Gaussian and linear weighting functions, which both require a hyper-parameter tuning similar to GreedyNMS. In [24], the final coordinates of a detection are being reformulated as the weighted-average of the coordinates of all neighboring detections, given an IoU threshold. GossipNet [2] is a DNN designed to perform NMS, by processing the coordinates and scores of the detections. Overall, it jointly analyzes all detections in the image, so as not to directly prune them, but to rescore them. In [25], the authors replace the classification scores of candidate detections, used in GreedyNMS, with learned localization confidences to guide NMS towards preserving more accurately localized bounding boxes. In [4], an attention module is applied with the task to exploit relations between the input detections, in order to classify them as duplicate or not. [1] proposes Adaptive-NMS, a dynamic thresholding version of GreedyNMS. A relatively shallow neural network predicts a density map and sets adaptive IoU thresholds in NMS for different detections according to the predicted density. An accelerated NMS method has been proposed in [26], allowing higher inference times in exchange for a small performance drop, due to the large number of boxes that are likely to be over-suppressed.

GossipNet was modified in [3], for the specific case of person detection from aerial views, so as to jointly process visual appearance and geometric properties of candidate RoIs. The method exploited handcrafted descriptors encoding statistical RoI appearance characteristics, which were computed on the

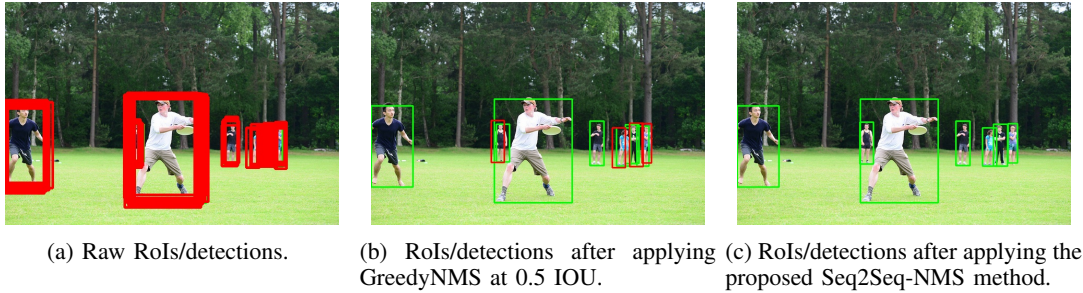


Fig. 1: Candidate RoIs/detections from Faster-RCNN in an image from the COCO dataset. Detections matched successfully to humans are colored green, while “incorrect” detections are colored red.

spatial distribution of edges detected within each RoI. These distributions acted as a discriminant factor for identifying complete vs partial object silhouettes, since the silhouette of any person seen from an aerial view is rather similar in shape.

More recently, [27] proposed *Distance-IoU* (DIOU), a new metric which can replace the typical IoU metric in GreedyNMS. This work suggested that the suppression procedure should take into account not only the overlap of two neighboring detections, but also the distances between their centers. Alternatively, Cluster-NMS was proposed in [28], i.e., a technique where NMS is performed by implicitly clustering candidate detections. Cluster-NMS can incorporate geometric factors to improve both precision and recall rates and can efficiently run on a GPU, achieving very fast inference runtimes.

### C. Loss Functions for Bounding Box Regression

In DNN-based methods for visual object detection, prediction of spatially accurate RoIs/bounding boxes is enforced by an additional loss term during model training. The regressed RoI parameters are position, shape and scale, in terms of 2D pixel coordinates. These parameters are predicted either directly, or as offsets relative to “anchor boxes”, in the case of anchor-based detectors. It is common to use the  $\mathcal{L}_n$ -norm for calculating the corresponding loss term (e.g., [18] [19] [20]). However, [29] indicates that the correlation between training with such  $\mathcal{L}_n$ -norm loss terms and improving test accuracy, as measured by the Intersection-over-Union (IoU) metric, is not strong at all. On the other hand, directly incorporating the IoU metric in a loss function would implicitly force the detector itself to also perform a rudimentary degree of NMS, but this is unsuitable for cases where two bounding boxes are non-overlapping, due to their zero loss gradient. Thus, [29] proposes the *Generalized-IoU* (GIoU) loss term, which handles similar scenarios but suffers from slow convergence and inaccurate regression. Thus, in [27], a loss term relying on the DIOU metric was formulated, by adding to the IoU loss a penalty based on the 2D center point coordinates of two bounding boxes. This was shown to converge faster than GIoU. [27] also proposed the *Complete-IoU* (CIoU) loss, an extension of DIOU with an additional term which can be tuned so as to impose aspect ratio consistency between two bounding boxes, thus leading to further increases in test accuracy.

### D. Limitations of Existing Methods

State-of-the-art object detectors continue to require NMS as a final step [21], even when they use sophisticated loss functions for bounding box regression during training. A typical scenario showcasing the indispensability of a reliable NMS method is when object detection is performed on images with high levels of occlusions [1] [30]; ironically, this constitutes a challenge even to state-of-the-art NMS algorithms.

Although the geometric properties of candidate RoIs have been considerably exploited by various NMS approaches [2] [27] [28] [26], only a couple of methods [1] [4] [3] have attempted to take advantage of both visual appearance and geometric/spatial RoI information. Therefore, joint exploitation of appearance and geometry for NMS in object detection is underexplored. In addition, despite a vast amount of effort expended towards achieving short inference times [26] [27], since fast execution is an important aspect of NMS, one can easily identify real-world scenarios where a potential improvement in accuracy may equally matter (e.g., pedestrian/person detection in human safety-centric applications).

Despite the sequential nature of the NMS task in object detection, since at least the input candidate RoIs are always ordered according to their confidence score, no previous method has relied on formulating the problem as a sequence-to-sequence task. Thus, the recent rise of self-attention neural modules [31], capable of efficiently capturing interrelations within a sequence, has not yet significantly affected NMS algorithms. To the best of our knowledge, the only relevant method employing self-attention mechanisms is [4], tailored for the duplicate removal task and not for pure NMS. Thus, it does not perform free rescoring: an input candidate RoI which was assigned a low confidence score by the detector (e.g., due to occlusion) cannot be rescored higher by the duplicate removal DNN; only lower. An unconstrained NMS method exploiting the powerful self-attention neural mechanism has yet to emerge.

Out of the existing literature, the proposed method is most related to [2] [3] and [4]. Like GossipNet in [2], Seq2Seq-NMS approaches NMS as a rescoring problem. However, an optimized geometric representation for each candidate RoI is proposed here, slightly similar, but different and enriched compared to the GossipNet input descriptor. Like [3], Seq2Seq-NMS jointly processes visual and geometric representations

of the input candidate RoIs, using the FMoD descriptor [7] computed on edge maps of cropped detections. However, in this paper, the FMoD descriptor has been re-implemented neurally, leading to significant runtime gains thanks to GP-GPU-based parallel processing, while a novel deep neural architecture is proposed here, so as to exploit the sequence-to-sequence formulation, instead of relying on GossipNet. Finally, similarly to [4], the Seq2Seq-NMS architecture employs the powerful self-attention neural mechanism, but since the proposed method is a complete, free rescoring NMS DNN it is able to search for and fully exploit interrelations between the candidate RoI representations, without being constrained by the original confidence score assigned by the object detector.

### III. ATTENTION-DRIVEN NON-MAXIMUM SUPPRESSION

In this paper, NMS for object detection is first reformulated as a sequence-to-sequence task. This approach is highly related to the evaluation criteria established in object detection [5] [6], where the candidate RoIs identified on an input image are assumed to indirectly form a sequence, based on the scalar confidence score assigned to each of them by the detector (in descending order). Traditionally, evaluating a detector’s accuracy on a known dataset involves an analysis of this sequence. At each step, a candidate RoI is processed and matched to a ground-truth object, if and only if: (a) their IoU is higher than a predefined threshold, and (b) that ground-truth object hasn’t been previously matched to a higher-scoring candidate detection. In the case where both (a) and (b) are fulfilled, the candidate RoI is marked as “correct”, otherwise it is marked as “false”. In the special case where only (a) is fulfilled, the candidate detection is marked as “false”, due to it being a “duplicate” detection. Thus, the position of a candidate RoI in the sequence can be a significant factor when taking the decision to classify it as a “duplicate” or not.

This emphasis in the ordering is shared with problems traditionally viewed as sequence-to-sequence ones. For instance, in machine translation, a sequence of words from one language must be transformed into a sequence of words in another language. The order of each word (*token*) in the sentence is crucial and can modify its meaning (*context*). Similarly, in object detection evaluation, although a candidate RoI (token) can be successfully matched to a ground-truth object, it can be classified as “duplicate” and therefore as “false”, instead of being classified as “correct”, due to the fact that a higher-scoring candidate detection, which has been positioned earlier in the sequence, has already been matched with the same ground-truth object.

Motivated by these notions, this paper explicitly formulates the NMS task as a mapping from an input sequence of candidate RoIs to a corresponding output sequence with identical length. Let  $\mathbf{R}^{in}$  be the input sequence of candidate RoIs, in descending order with respect to detector confidence scores:

$$\mathbf{R}^{in} = [\mathbf{r}_1^{in}, \dots, \mathbf{r}_N^{in} | r_i^{score_{det}} \geq r_{i+1}^{score_{det}}] \quad (1)$$

where  $\mathbf{r}_i^{in} = [r_i^{x_{min}}, r_i^{y_{min}}, r_i^{x_{max}}, r_i^{y_{max}}, r_i^{score_{det}}]$  is an input candidate RoI expressed through its 2D image coordinates,

along with its corresponding score assigned by the detector, and  $N$  is the number of candidate detections. Let  $\mathbf{R}^{out}$  be the output sequence of candidate RoIs, in descending order based on the scores assigned by the NMS method:

$$\mathbf{R}^{out} = [\mathbf{r}_1^{out}, \dots, \mathbf{r}_N^{out} | r_i^{score_{NMS}} \geq r_{i+1}^{score_{NMS}}] \quad (2)$$

where  $\mathbf{r}_i^{out} = [r_i^{x_{min}}, r_i^{y_{min}}, r_i^{x_{max}}, r_i^{y_{max}}, r_i^{score_{NMS}}]$  is an NMS-rescored candidate RoI. The proposed formulation of the NMS task can be expressed as:

$$\mathbf{R}^{out} = NMS(\mathbf{R}^{in}) \quad (3)$$

Building upon this novel view of the NMS task, the method proposed in this paper, which we call *Seq2Seq-NMS*, receives as input a sequence of candidate RoIs, generated by an object detector, and extracts rich representations regarding their appearance and geometry. Subsequently, these representations are fed to a DNN which processes them in parallel, while mainly paying attention to spatially neighboring, higher-scoring candidates when analyzing each RoI. Finally, it outputs a sequence of scalar scores, each one defining the context of a candidate detection. This is essentially information that determines the final decision of whether the respective RoI should be classified as “correct” or as “potentially suppressed”, after the NMS task has been completed. In the proposed formulation, the context of the  $i^{th}$  candidate detection is expressed through the corresponding output score, which is a classification probability  $p_i : \{p_i \in \mathbb{R} | 0 \leq p_i \leq 1\}$  (1/0 means “correct”/“potentially suppressed”, respectively). After the inference stage, simple thresholding can be applied on these output probabilities/scores, in order to decide which candidate detections should be retained. This formulation avoids hard discarding/pruning of RoIs at the inference phase itself, thus allowing us to find a balance in the trade-off between False Positive Rate (FPR) and True Negative Rate (TNR), depending on the application (e.g., using a low threshold in human safety-centric applications such as pedestrian detection).

Seq2Seq-NMS relies on building rich representations for each candidate detection, based on their visual appearance, their geometry and their interrelations. Abstractly, it consists of the following three steps:

- Appearance-based RoI representations extraction.
- Geometry-based RoI representations extraction.
- Detections rescoring through the attention-driven NMS DNN.

These steps are detailed below.

#### A. Appearance-based RoI Representations Extraction

This step can be considered optional, since RoI representations that have been already computed at the intermediate feature extraction layers of the DNN-based object detector itself can be used instead. However, the use of RoI representations computed solely for the NMS procedure makes the NMS DNN less detector-specific and more robust against variations in the effectiveness and the performance of the deployed detector. In [3], where the goal was person detection



from aerial views, representations consisting of statistical RoI appearance properties, computed on the spatial distribution of edges detected within each RoI, were used. These distributions acted as a discriminant factor for identifying complete vs partial object silhouettes, since the aerial view of persons silhouettes are similar in shape. However, the same argument can be made for people seen from a ground perspective (e.g., pedestrians perceived by an autonomous car), therefore this is a solution applicable to most person detection scenarios.

---

**Algorithm 1:** Appearance-based RoI representations extraction using FMoD

---

**Input:** (a) an RGB image  $\mathbf{I}$   
 (b) a set of  $N$  RoIs expressed in 2D pixel coordinates  $\mathbf{B} = [\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_N] \in \mathbb{R}^{N \times 4}$   
 (c) FMoD pyramid levels  $L$ ,  $L \geq 1$   
**Output:** Appearance-based representations  $\mathbf{A} \in \mathbb{R}^{N \times 5(4^L - 1)}$

---

```

1 begin
2   Resize image  $\mathbf{I}$  to a fixed size of  $W_f \times H_f$ .
3    $E(\mathbf{I}) \leftarrow$  Compute the edge map of image  $\mathbf{I}$ .
4   Extract in parallel the  $0^{th}$ -level RoI maps
      $\mathbf{M}^0 = [\mathbf{M}_0^0, \mathbf{M}_1^0, \dots, \mathbf{M}_N^0]$ , where  $\mathbf{M}_i^0 \in \mathbb{R}^{1 \times W_0 \times H_0}$ ,
     through the ROIAlign operator on  $E(\mathbf{I})$ .
5   Compute in parallel the  $0^{th}$ -level FMoD
     representations  $\mathbf{A}^0 = [\mathbf{A}_0^0, \mathbf{A}_1^0, \dots, \mathbf{A}_N^0]$  of  $\mathbf{M}^0$ ,
     where  $\mathbf{A}_i^0 \in \mathbb{R}^{15 \times 1}$ .
6   for  $j \leftarrow 1$  to  $(L - 1)$  do
7     Extract in parallel the  $j^{th}$ -level RoI maps
        $\mathbf{M}^j = [\mathbf{M}_0^j, \mathbf{M}_1^j, \dots, \mathbf{M}_N^j]$ , where
        $\mathbf{M}_i^j \in \mathbb{R}^{4^j \times \frac{W_0}{2^j} \times \frac{H_0}{2^j}}$ , through subdivision of
        $\mathbf{M}^0$  RoI maps into four quadrants for  $j$  times,
       using the ROIAlign operator.
8     Compute in parallel the  $j^{th}$ -level FMoD
       representations  $\mathbf{A}^j = [\mathbf{A}_0^j, \mathbf{A}_1^j, \dots, \mathbf{A}_N^j]$  of  $\mathbf{M}^j$ ,
       where  $\mathbf{A}_i^j \in \mathbb{R}^{15 \times 4^j}$ .
9   end
10  Concatenate FMoD representations across all
     pyramid levels  $\mathbf{A} \in \mathbb{R}^{N \times 5(4^L - 1)}$ , where
      $\mathbf{A}_i = [\mathbf{A}_i^0, \dots, \mathbf{A}_i^L]$ .
11 end
  
```

---

In [3], a CPU implementation of the low-level FMoD visual descriptor was employed for representing candidate RoIs. FMoD was originally devised in a global [7] and in a local [32] variant (LMoD), respectively applied to movie [33] and activity video [34] [35] [36] summarization via key-frame extraction. Typically, FMoD and LMoD capture informative image statistics from various available image channels (e.g., luminance, color/hue, optical flow magnitude, edge map, and/or stereoscopic disparity), both in a global and in various local scales, under a spatial pyramid video frame partitioning scheme. Following [3], only the edge map of an image's luminance channel is used here as input channel for the FMoD algorithm, with the latter one applied separately

at each candidate RoI. The intent is to compactly capture the spatial distribution of the edges within each RoI in a single description vector. However, in [3] RoIs were processed sequentially and not simultaneously, thus demanding very long inference times. To tackle this limitation, in this paper FMoD was re-implemented neurally so that it runs very fast and in parallel on modern GPUs. Given as input an image and a set of candidate RoIs (in pixel coordinates) of different shape and scale, it extracts all corresponding regions of the luminance edge map by cropping it along the boundaries of the respective RoIs. This is done separately for each candidate RoI, but in parallel for all of them (at a single step). Subsequently, the FMoD descriptors/representations of all these cropped edge maps/RoIs are also computed separately but in parallel.

The appearance-based RoI representations extraction process can be divided into three operations. The first one involves the computation of the edge map of the input image, which is a relatively fast and efficient process. The second step is the use of the *ROIAlign* [37] operator to extract, in parallel, fixed-size regions across one or multiple maps. Finally, deriving the FMoD representations of these fixed-size maps involves in-parallel computation of the following 15 scalar statistical attributes:

- (1-3) horizontal/vertical/vectorized-block mean values.
- (4-6) horizontal/vertical/vectorized-block standard deviation values.
- (7-9) horizontal/vertical/vectorized-block skew values.
- (10-12) horizontal/vertical/vectorized-block kurtosis values.
- (13-15) horizontal/vertical/vectorized-block signal power values.

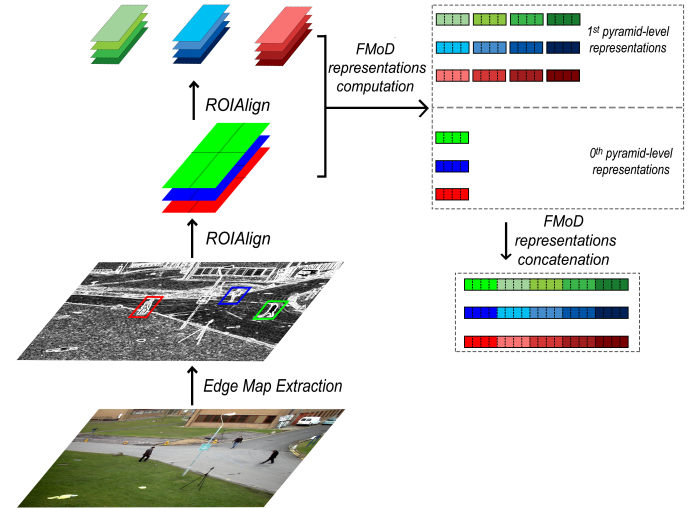


Fig. 2: Computation of the visual appearance-based candidate RoI representations, by applying the fast FMoD implementation to an image with 3 RoIs and using 2 pyramid levels.

The corresponding procedure is described in Algorithm 1. Initially, the RGB input image  $\mathbf{I}$ , of an arbitrary resolution, is resized to a fixed resolution of  $W_f \times H_f$  and its corresponding

edge map  $E(\mathbf{I})$  is computed. To make actual inference times even shorter, this operation is carried out here in parallel with the corresponding detector's inference phase. Similarly to [3], the FMoD representations of all RoIs are computed under a spatial pyramid partitioning scheme [38]. At the pyramid base, the  $0^{th}$ -level RoI maps  $\mathbf{M}^0 = [\mathbf{M}_0^0, \mathbf{M}_1^0, \dots, \mathbf{M}_N^0]$ ,  $\mathbf{M}_i^0 \in \mathbb{R}^{1 \times W_0 \times H_0}$  are extracted in parallel by applying the ROIAlign operator on  $E(\mathbf{I})$ , assuming that  $N$  candidate RoIs have been identified by the object detector for input  $\mathbf{I}$ . Using  $\mathbf{M}^0$ , the  $0^{th}$ -level FMoD representations  $\mathbf{A}^0 = [\mathbf{A}_0^0, \mathbf{A}_1^0, \dots, \mathbf{A}_N^0]$ ,  $\mathbf{A}_i^0 \in \mathbb{R}^{15 \times 1}$  are computed in parallel. Subsequently, the representations at the remaining spatial pyramid levels are computed iteratively, by the in-parallel computation first of  $\mathbf{M}^j$  and then of the corresponding partial FMoD descriptors  $\mathbf{A}^j$ . Once the latter ones have been computed for all (predefined and fixed)  $L$  pyramid levels, they are concatenated along them. For example, in an image with  $N = 3$  candidate RoIs and  $L = 2$  pyramid levels,  $\mathbf{A} \in \mathbb{R}^{3 \times 75}$ . This example is illustrated in Figure 2.

### B. Geometry-based RoI Representations Extraction

The spatial/geometric interrelations between the various candidate RoIs, based only on their 2D pixel coordinates and not on their visual appearance, is crucial for solving the NMS problem. Such a set of purely geometric attributes has previously proven effective as an input descriptor, in the context of GossipNet [2]. Thus, in this paper, a slightly similar, but enriched set of attributes has been devised, serving as an additional representation for each RoI.

Given a set of  $N$  candidate RoIs, along with their corresponding detection scores, the tensor  $\mathbf{G} \in \mathbb{R}^{N \times N \times 14}$  is computed, where each entry  $\mathbf{G}^{ij} \in \mathbb{R}^{14}$  contains the following attributes:

- (1-3) the normalized horizontal/vertical/euclidean distances<sup>1</sup> between the centers of the  $j^{th}$  and the  $i^{th}$  RoI.
- (4-7) the normalized width/height/area/aspect-ratio of the  $j^{th}$  RoI.
- (8-11) the ratios between the  $j^{th}$  and the  $i^{th}$  RoIs width/height/area/aspect-ratio (e.g.,  $\frac{w_j}{w_i}$ ).
- (12) the detector's confidence score for the  $j^{th}$  RoI.
- (13) the detector's confidence score differences between the  $j^{th}$  and the  $i^{th}$  RoI (e.g.,  $s_j - s_i$ ).
- (14) the IoU between the  $j^{th}$  and the  $i^{th}$  RoI.

Therefore, each diagonal entry  $\mathbf{G}^{ii} \in \mathbb{R}^{14}$  contains the geometric representation of the  $i$ -th input candidate RoI/detection.

### C. Detections rescoring through the attention-driven NMS DNN

The goal of the proposed DNN architecture is to perform one-class Non-Maximum Suppression on a set of candidate RoIs/detections through rescoring rather than pruning them. For a given set of  $N$  such RoIs, the DNN receives as input a sequence of corresponding representations ( $\mathbf{A}$  and  $\mathbf{G}$ , encoding the appearance and geometry of all RoIs in the sequence),

<sup>1</sup>Horizontal and vertical distances are signed distances.

sorted in a descending order based on the respective scalar detection confidence score.

During inference, these two types of information are fused and each candidate RoI refines its representation by attending to the representations of all detections in the set. The Scaled Dot-Product Attention mechanism [31], originally proposed for machine translation tasks, is employed, since it has been proven effective in various applications, such as image classification [39], or generation [40]. The mechanism is briefly described below. In the context of the proposed DNN, the candidate detections used as keys are represented in a relative-to-each-query manner within this attention mechanism. Although this choice leads to slightly increased computational and memory costs, it allows the DNN to more effectively capture the interrelations between the candidate detections.

Finally, the model predicts a new scalar score for each RoI, indicating whether it should be suppressed or not. The output sequence is formed by sorting the candidate RoIs, based on their new scores in descending order.

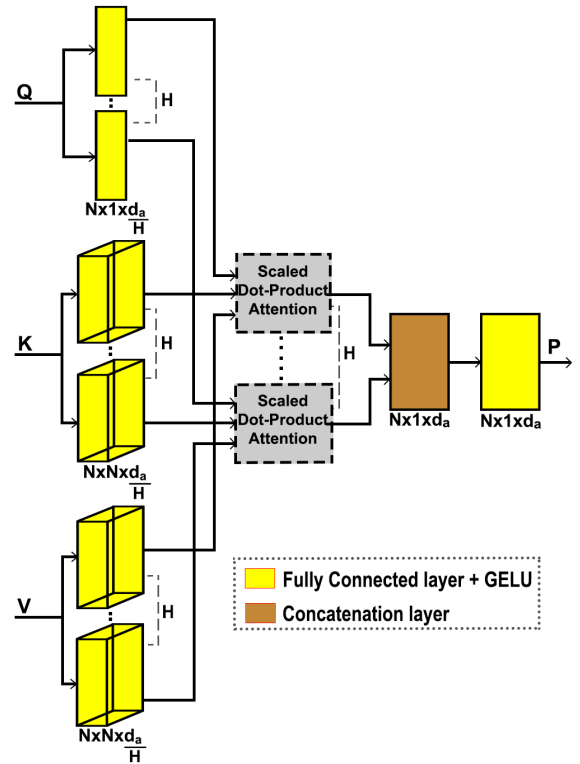


Fig. 3: Illustration of the Multihead Self-Attention Module.

**Multihead Self-Attention Module:** The Scaled Dot-Product Attention, also known as self-attention, was presented in [31] and formulated as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \quad (4)$$

where  $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$  are the queries,  $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$  are the keys and  $\mathbf{V} \in \mathbb{R}^{N_k \times d_v}$  are the values. Each query and each key has

a dimension of  $d_k$ , while each value has a dimension of  $d_v$ . Multihead Attention was also proposed in [31], as a module which allows various attention mechanisms, including self-attention, to run in parallel. This module can be formulated as:

$$\text{Multihead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{h}_1, \dots, \mathbf{h}_H] \mathbf{W}^O, \quad (5)$$

where

$$\mathbf{h}_i = \text{Attention}(\mathbf{Q} \mathbf{W}_i^Q, \mathbf{K} \mathbf{W}_i^K, \mathbf{V} \mathbf{W}_i^V). \quad (6)$$

In this formulation,  $\mathbf{W}_i^Q \in \mathbb{R}^{d_a \times d_k}$ ,  $\mathbf{W}_i^K \in \mathbb{R}^{d_a \times d_k}$ ,  $\mathbf{W}_i^V \in \mathbb{R}^{d_a \times d_v}$ ,  $\mathbf{W}_i^O \in \mathbb{R}^{H d_v \times d_a}$  are projection parameter matrices,  $H$  is the number of heads,  $d_k = d_v = \frac{d_a}{H}$ , and the operator [...] implies concatenation.

The proposed DNN architecture relies on these mechanisms in order to identify relations between candidate detections, based both on their visual appearance and their geometric properties. Such relations can help the model in determining whether a detection should be suppressed or not. For example, the DNN can decide that a higher-scoring candidate RoI should possibly suppress other less-scoring ones having similar appearance and geometric representations.

In [31] the authors introduced *positional encoding* for Natural Language Processing (NLP) tasks, which uses a combination of sines and cosines at multiple frequencies, in order to encode the position of a word in a sequence. In theory, this approach could also be adopted for encoding RoI geometry (e.g., the position of RoI centers along a certain axis). However, this may fail to capture the interrelations of candidate RoIs in a relative manner, as the encoded information in the NMS task is far more complex compared to [31]. As an alternative, we approached the task by encoding all the representations of the input candidate detections in a relative-to-each-RoI manner. Thus, the keys and values of the Scale Dot-Product Attention are represented in a relative-to-each-query representation scheme. For example, the  $j^{\text{th}}$  key may be represented differently for the  $i^{\text{th}}$  query, compared to its representation for the  $(i+1)^{\text{th}}$  query. Although this increases the method's memory complexity, each query is allowed to represent the keys and the values relatively to itself. Thus, for  $N$  detections,  $\mathbf{Q} \in \mathbb{R}^{N \times 1 \times d_a}$ ,  $\mathbf{K} \in \mathbb{R}^{N \times N \times d_a}$  and  $\mathbf{V} \in \mathbb{R}^{N \times N \times d_a}$ , the output is  $\mathbf{P} \in \mathbb{R}^{N \times 1 \times d_a}$ .

Due to the increased number of dimensions of  $\mathbf{Q}$ ,  $\mathbf{K}$  and  $\mathbf{V}$ , batch matrix multiplication is employed in Eq. (4) to speed up the process. The architecture of this module is illustrated in Figure 3.

**Joint Processing Module (JPM):** In this module, the representations of the detections are jointly and simultaneously refined, mainly through the Multihead Self-Attention mechanism. The JPM receives as its input  $\mathbf{F}_t^Q \in \mathbb{R}^{N \times 1 \times d_m}$ , which holds the current representations of all candidate detections, as well as  $\mathbf{F}_t^K \in \mathbb{R}^{N \times N \times d_a}$ , which holds the current relative-to-each-detection representations, for all  $N$  candidate detection.

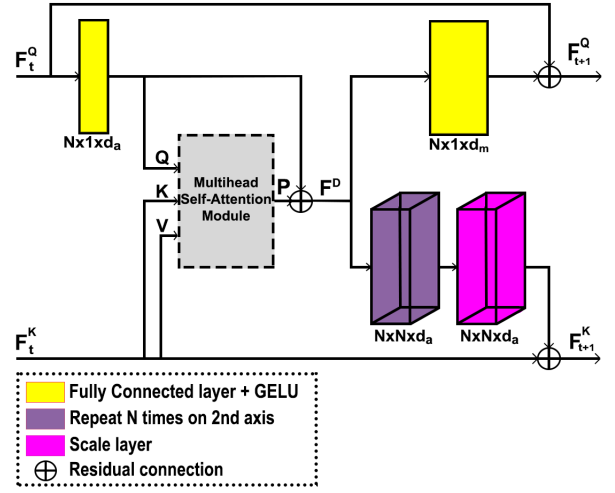


Fig. 4: Illustration of the Joint Processing Module (JPM).

The architecture of the JPM is shown in Fig. 4. The queries and keys are formed as:

$$\begin{aligned} \mathbf{Q} &= \mathbf{F}_t^Q \mathbf{C}^Q, \\ \mathbf{K} &= \mathbf{F}_t^K, \\ \mathbf{V} &= \mathbf{K}, \end{aligned} \quad (7)$$

where  $\mathbf{C}^Q \in \mathbb{R}^{d_m \times d_a}$  stands for the weights of a fully connected layer. The new representations of the candidate detections, which is the output of this module, are formed as:

$$\begin{aligned} \mathbf{F}_{t+1}^Q &= \mathbf{F}^D \mathbf{C}^D + \mathbf{F}_t^Q, \\ \mathbf{F}^D &= \mathbf{P} + \mathbf{Q}, \end{aligned} \quad (8)$$

where  $\mathbf{C}^D \in \mathbb{R}^{d_a \times d_m}$  also denotes the weights of a fully connected layer. In addition, residual connections [41] are applied between  $\mathbf{Q}$  and  $\mathbf{P}$  as well as between  $\mathbf{F}_{t+1}^Q$  and  $\mathbf{F}_t^Q$ .

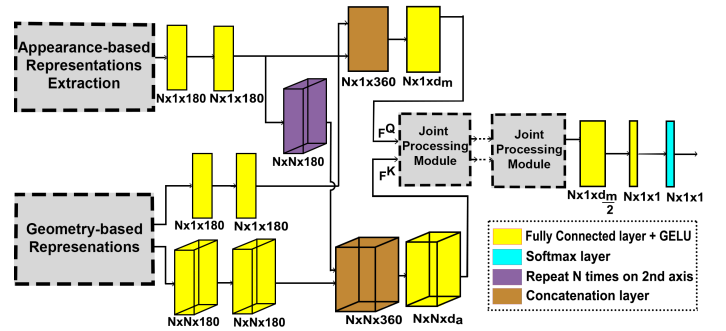


Fig. 5: Seq2Seq-NMS architecture.  $N$  is the number of input candidate RoIs/detections.

Finally, the relative-to-each-candidate-detection representations  $\mathbf{F}^K$  are refined as:

$$\mathbf{F}_{t+1}^K = \mathbf{F}_t^K + \mathbf{F}^S \otimes \mathbf{C}^K, \quad (9)$$

where  $\mathbf{F}^S$  is derived from  $\mathbf{F}^D$ , by repeating it  $N$  times along its second dimension, and  $\mathbf{C}^K$  are learned weights of

a *Scale Layer* that we introduce, performing an element-wise multiplication between its weights and an input representation. Its purpose is to select the degree of information which will flow from  $\mathbf{F}^S$  to  $\mathbf{F}_{t+1}^K$  in each JPM.

**Masking:** A masking approach has been integrated into the self-attention mechanism of the proposed architecture. For  $N$  sorted candidate detections, we mask the values of the input of the softmax function in Eq. (4). Without loss of generality, masking is detailed below for the simplest case, where  $H = 1$ .

Given a candidate RoI  $\mathbf{r}_i^{in}$ , an its associate RoI  $\mathbf{r}_j^{in}$  and  $\mathbf{S} = \frac{\mathbf{QK}^T}{\sqrt{d_k}}$ , masking is defined as:

$$S_{ij} = \begin{cases} -\infty, & \text{if } IoU(\mathbf{r}_i^{in}, \mathbf{r}_j^{in}) < 0.2 \\ 0.1 \cdot S_{ij}, & \text{if } IoU(\mathbf{r}_i^{in}, \mathbf{r}_j^{in}) \geq 0.2 \text{ and } j > i \\ S_{ij}, & \text{otherwise} \end{cases} \quad (10)$$

Masking is employed for two reasons. First, each RoI must be prevented from attending to spatially distant detections. The overlap of RoIs is used to determine whether  $S_{ij}$  should be set to  $-\infty$ , before applying the softmax function. If yes, the attention weight linking  $\mathbf{r}_i^{in}$  to  $\mathbf{r}_j^{in}$  (after applying softmax) will be zeroed out. Second, we attempt to replicate the behaviour of Greedy NMS, where a detection is characterized as duplicate, thus marked for suppression, when another, higher-scoring detection spatially covers the same object. In the proposed architecture this can be accomplished by forcing (through masking) the internal representation of a candidate detection to be modified by attending mainly to representations that correspond to RoIs higher-scoring than itself.

**Network Architecture:** For a set of  $N$  candidate sorted detections, the proposed DNN uses as input their corresponding appearance-based  $\mathbf{A}$  and geometry-based representations  $\mathbf{G}$ . FMoD representations of 3 pyramid levels are employed as  $\mathbf{A} \in \mathbb{R}^{N \times 1 \times 315}$ . The extracted geometry-based RoI representations, namely  $\mathbf{G} \in \mathbb{R}^{N \times N \times 14}$ , are assigned to  $\mathbf{G}^K$  as it contains the relative-to-each-candidate-detection representations. Its diagonal, derived from the first two dimensions, forms  $\mathbf{G}^Q \in \mathbb{R}^{N \times 1 \times 14}$ . The representations derived from a fusion between  $\mathbf{A}$  and  $\mathbf{G}^Q$  form  $\mathbf{F}^Q \in \mathbb{R}^{N \times 1 \times d_m}$ . This fusion is mainly accomplished by concatenating and applying fully-connected layers between the two types of representations. In addition, the representations derived from a fusion between  $\mathbf{A}$  and  $\mathbf{G}^K$  form  $\mathbf{F}^K \in \mathbb{R}^{N \times N \times d_a}$ . Both  $\mathbf{F}^Q$  and  $\mathbf{F}^K$  are used as input to the first JPM.

A stack of JPMs, sequentially connected, are in charge of refining representations  $\mathbf{F}^Q$  and  $\mathbf{F}^K$ . Finally, after applying two fully connected layers on  $\mathbf{F}^Q$ , the DNN uses a softmax function to output the final NMS scores. The model architecture is depicted in Fig. 5. The Gaussian Error Linear Unit (GELU) is used as activation function. Layer normalization [42] is applied on the output of residual connections and dropout [43] is used for regularization, similarly to [31].

**Training:** The weighted binary cross entropy was selected as the training objective of the proposed neural architecture. In

particular, the loss function is defined as:

$$L = -\sum_{i=1}^N (w_1 y_i \log(r_i^{scores_{NMS}}) + w_0 (1 - y_i) \log(1 - r_i^{scores_{NMS}})), \quad (11)$$

where  $N$  is the number of candidate detections,  $\mathbf{r}^{scores_{NMS}}$  are the output NMS scores,  $\mathbf{w}$  are class weights and  $\mathbf{y}$  are the labels derived from a matching function, given a specific IoU value. In particular,  $y_i \in \{1, 0\}$  indicates whether the  $i^{th}$  detection was successfully matched to an object or not. A detection is matched successfully to an object, when the IoU between its RoI and an object's 2D bounding box is higher or equal to a matching threshold, and that specific object hasn't been matched to any higher scoring detection. In this paper, this matching IoU threshold was set to 0.5. A strategy similar to the one in [2], is used for the class weights computation.

#### IV. EXPERIMENTAL EVALUATION

The performance of Seq2Seq-NMS was evaluated on three separate datasets for the person detection task. In all datasets, candidate RoIs from the *Single Shot Detector* (SSD) [19] were provided as input to the proposed NMS method. In the implemented version of the detector, VGG16 with atrous convolutions was selected as the backbone CNN. The input images were resized to a resolution of  $512 \times 512$  pixels, while the detector was trained from scratch for each dataset<sup>2</sup>. Independently from this set of experiments, a complementary evaluation scheme was also conducted by employing a different detector per dataset. These detectors were selected in order: a) to facilitate direct comparisons with previously published NMS methods, and b) to compare the proposed NMS approach against competing ones in conjunction with different detectors with different behaviour. In this complementary set of experiments, the following detectors were employed: (a) a non-neural detector [17], (b) a two-stage DNN-based detector [18] and (c) a one-stage DNN-based detector [21].

The employed Seq2Seq-NMS architecture consists of 4 Joint Processing Modules. We set  $d_m = 256$ , and  $d_a = \frac{d_m}{2} = 128$ . The Multihead Self-Attention module uses  $H = 2$  attention heads and thus  $d_q = d_k = d_v = \frac{128}{H} = 64$ . Appearance-based RoI representations computed from 3-level FMoD were used, with  $0^{th}$  level RoI maps extracted at resolution  $W_0 \times H_0 = 160 \times 160$  pixels. In each evaluation setup, the proposed method was trained using the ADAM [44] optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$  and  $\epsilon = 10^{-9}$ . Given that the number of RoIs may be extremely large, we first applied TorchVision NMS with the relaxed 0.8 IoU threshold as a preprocessing step (common in NMS literature [2]). To achieve a fair comparison, this preprocessing step is applied in all deployed methods. Finally, Seq2Seq-NMS is trained using only the 720 highest-scoring candidate detections as an input sequence, due to practical memory limitations.

In all cases, Seq2Seq-NMS was compared against both neural and non-neural NMS algorithms. The first competing method is a baseline Greedy NMS approach running on GPU.

<sup>2</sup>The employed SSD implementation was adopted from [https://github.com/opencv/opencv/tree/master/src/opencv/perception/object\\_detection\\_2d/ssd](https://github.com/opencv/opencv/tree/master/src/opencv/perception/object_detection_2d/ssd)

TABLE I: COMPARISON OF DIFFERENT NMS METHODS ON THE PETS DATASET, USING DETECTIONS FROM [17]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	Pre-NMS max dets. = 600			Pre-NMS max dets. = 1200			Pre-NMS max dets. = 1500		
		AP <sub>0.5</sub>	AP <sub>0.5</sub> <sup>0.95</sup>	Average Inference Time (ms)	AP <sub>0.5</sub>	AP <sub>0.5</sub> <sup>0.95</sup>	Average Inference Time (ms)	AP <sub>0.5</sub>	AP <sub>0.5</sub> <sup>0.95</sup>	Average Inference Time (ms)
Original NMS IoU>0.4	GPU	76.7%	32.2%	2.1	77.2%	32.1%	3.5	77.3%	32.0%	5.0
Original NMS IoU>0.5	GPU	74.2%	31.7%	2.8	74.7%	31.7%	6.4	74.8%	31.7%	8.1
Original NMS IoU>0.6	GPU	66.9%	29.6%	4.2	67.2%	29.7%	10.1	67.3%	29.7%	13.6
TorchVision NMS IoU>0.4	GPU	76.8%	32.2%	0.4	77.3%	32.1%	0.6	77.3%	32.1%	0.6
TorchVision NMS IoU>0.5	GPU	73.9%	31.7%	0.4	74.4%	31.6%	0.6	74.4%	31.6%	0.5
TorchVision NMS IoU>0.6	GPU	66.4%	29.5%	0.4	66.6%	29.6%	0.5	66.7%	29.6%	0.6
Soft-NMS <sub>L</sub>	CPU	77.6%	32.5%	50.3	77.6%	32.3%	98.5	77.6%	32.1%	143.3
Soft-NMS <sub>G</sub>	CPU	78.2%	33.4%	39.2	77.6%	32.9%	89.5	77.2%	32.6%	154.7
Fast-NMS	GPU	75.3%	31.9%	1.4	75.2%	31.6%	2.2	75.2%	31.5%	3.2
Cluster-NMS	GPU	76.8%	32.2%	3.2	77.2%	32.1%	5.1	77.3%	32.1%	7.5
Cluster-NMS <sub>S</sub>	GPU	75.7%	32.3%	2.7	74.0%	31.3%	4.2	74.7%	31.6%	6.6
Cluster-NMS <sub>D</sub>	GPU	77.0%	32.3%	3.8	77.6%	32.1%	7.3	77.6%	32.1%	9.1
Cluster-NMS <sub>S+D</sub>	GPU	77.2%	32.6%	4.0	76.5%	32.0%	8.0	76.5%	32.0%	11.2
Cluster-NMS <sub>S+D+W</sub>	GPU	77.2%	32.6%	47.6	76.5%	32.0%	154.8	76.5%	32.0%	276.1
GossipNet	GPU	81.9%	36.3%	27.2	84.3%	37.2%	64.2	84.6%	37.2%	95.8
Seq2Seq-NMS	GPU	83.6%	37.8%	11.0	85.4%	38.4%	13.8	<b>85.5%</b>	<b>38.4%</b>	15.4
<b>Seq2Seq-NMS Gains</b> (The best performance of each method is used for comparison)		AP <sub>0.5</sub> +0.9%			AP <sub>0.5</sub> <sup>0.95</sup> +1.2%					

The second is TorchVision’s<sup>3</sup> GreedyNMS implemented to run very fast on GPUs. Soft-NMS [23], i.e., a non-neural NMS method widely used as a more accurate replacement for Greedy NMS, was also tested. Evaluation was conducted using both the linear and the Gaussian weighting functions (referred to as Soft-NMS<sub>L</sub> and Soft-NMS<sub>G</sub>, respectively), with on-CPU execution. Another competing algorithm is Fast-NMS [26]: a generally faster, non-neural replacement for standard NMS, executed on GPU but suffering a marginal penalty regarding accuracy. Additionally, several variants of Cluster-NMS [28], a more recent non-neural method, were also used for comparisons. Below, the term Cluster-NMS<sub>S</sub> is used to imply the use of the score penalty mechanism, while Cluster-NMS<sub>D</sub> implies the addition of the normalized central point distance. In the latter case, the method is equivalent to DIoU-NMS [27]. The term Cluster-NMS<sub>S+D</sub> is used when both of these mechanisms are utilized. Finally, Cluster-NMS<sub>S+D+W</sub> indicates a weighted strategy similar to [24]. More details regarding these variations can be found in [28]. The last approach selected for comparison purposes is GossipNet [2], a neural NMS method achieving state-of-the-art accuracy.

The hyperparameters of all non-neural methods were tuned so as to report the best achieved results on 0.5 IoU matching threshold. Evaluation was performed on a PC using an Intel Core i7-7700 CPU and an NVIDIA GeForce RTX 2080 GPU with 11GB of memory, both for training and inference. The employed evaluation metrics are AP<sub>0.5</sub>, AP<sub>0.5</sub><sup>0.95</sup> and inference times. AP<sub>0.5</sub> corresponds to the average precision for 0.5 IoU, while AP<sub>0.5</sub><sup>0.95</sup> to the mean average precision for IoU ranging from 0.5 to 0.95 with a step size of 0.05.

In the evaluation of all methods, the number of maximum

candidate detections prior to the NMS procedure was set to 1500. All RoIs outputted by the NMS algorithms were utilized for evaluation, without any thresholding.

#### A. PETS

PETS [45] is a relatively small dataset, whose images were collected from static surveillance cameras and provide diverse levels of occlusion. The average number of people depicted in an image is approximately 14. Apart from [19], [17], a non-neural person detection method designed to handle occlusions, was selected as the corresponding detector for providing raw candidate RoIs as input to the NMS methods.

The proposed NMS architecture was trained for 8 epochs. The learning rate was set to  $10^{-4}/10^{-5}/10^{-6}$  for epochs 1-4/5-7/8, respectively. GossipNet’s architecture and training followed [2]. Final parameters of all methods were selected according to the best achieved accuracy in the validation set.

Table I reports the results of the proposed and the competing NMS methods, using candidate detections from [17] as input. This object detector outputs a large number of candidate RoIs, thus leading to increased GPU memory consumption for both the proposed method and GossipNet. Typically, most candidate detections that can be successfully matched to ground-truth objects are assigned higher confidence scores by the detector, compared to RoIs with lower scores (e.g.,  $< 0.05$ ) which are mostly false positive samples. Thus, in this experiment, we attempt to evaluate whether the lowest scoring detections have an impact on the performance of the proposed and the competing NMS methods. Table I reports the results of each NMS approach using  $N$  candidate detections as input, for different values of  $N$ . As it can be seen, the performance of several non-neural methods, such as Soft-NMS<sub>L</sub> and Cluster-NMS<sub>D</sub>, does not improve when the lowest-scoring detections

<sup>3</sup><https://pytorch.org/vision/stable/ops.html#torchvision.ops.nms>

(e.g.,  $> 1200$ ) are used. In contrast, both neural methods achieve more accurate results for longer input sequences (more candidate RoIs per image). In this setup, the proposed method achieved both the best  $AP_{0.5}$  and the best  $AP_{0.5}^{0.95}$ , against all competing approaches, even in the case where only the highest 1200 candidate input detections were used. The obtained  $AP_{0.5}$  was 85.5%, which is a +7.3% improvement against Soft-NMS<sub>L</sub> and Cluster-NMS<sub>D</sub>, the non-neural method with the best  $AP_{0.5}$ , and a +0.9% improvement against GossipNet. In addition, the obtained  $AP_{0.5}^{0.95}$  was 38.4%, which is an +1.2% gain over the competing methods. Notably, when using only a small number of the highest-scoring candidate detections (e.g.,  $N = 600$ ), the proposed method still achieves better results compared to all non-neural NMS algorithms. Regarding inference runtimes, it needs 15.4 ms to run per image when  $N = 1500$ , since the required edge maps are computed in parallel with the object detector’s inference. Thus, it is faster than GossipNet, as well as far less affected (with respect to runtime) by the number of candidate detections used as input. Indeed the GossipNet inference runtime drastically increases with  $N$  but this is not the case for the proposed approach. However, Seq2Seq-NMS is slower than most non-neural methods running on GPU.

TABLE II: COMPARISON OF DIFFERENT NMS METHODS ON THE TEST SET OF THE PETS DATASET, USING DETECTIONS FROM [19]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	$AP_{0.5}$	$AP_{0.5}^{0.95}$	Average Inference Time (ms)
Original NMS IoU>0.4	GPU	87.6%	35.0%	12.7
Original NMS IoU>0.5	GPU	89.9%	36.3%	13.1
Original NMS IoU>0.6	GPU	89.8%	37.1%	13.4
TorchVision NMS IoU>0.4	GPU	88.0%	35.1%	0.3
TorchVision NMS IoU>0.5	GPU	90.0%	36.4%	0.2
TorchVision NMS IoU>0.6	GPU	89.8%	37.2%	0.3
Soft-NMS <sub>L</sub>	CPU	90.0%	38.2%	134.4
Soft-NMS <sub>G</sub>	CPU	89.6%	38.6%	108.1
Fast-NMS	GPU	87.6%	36.8%	6.0
Cluster-NMS	GPU	90.2%	36.9%	13.4
Cluster-NMS <sub>S</sub>	GPU	90.1%	38.0%	13.8
Cluster-NMS <sub>D</sub>	GPU	90.2%	36.6%	17.9
Cluster-NMS <sub>S+D</sub>	GPU	90.6%	38.3%	22.4
Cluster-NMS <sub>S+D+W</sub>	GPU	90.6%	38.3%	38.2
GossipNet	GPU	90.7%	<b>38.8%</b>	24.5
Seq2Seq-NMS	GPU	<b>90.9%</b>	38.6%	19.7
<b>Seq2Seq-NMS Gains</b>		+0.2%	-0.2%	-

Table II reports the results using candidate detections from [19]. The proposed method achieved an  $AP_{0.5}$  of 90.9%, thus attaining a gain of +0.2% over GossipNet. In terms of  $AP_{0.5}^{0.95}$ , the proposed method was outperformed only by GossipNet (-0.2%) and was on par with Soft-NMS<sub>G</sub>. Regarding inference runtimes, Seq2Seq-NMS needed on average 19.7 ms to run per image, since the required edge maps are computed in parallel with the object detector’s inference stage. Though this is faster than GossipNet, it is again slower than non-neural methods running on GPU.

## B. COCO Person

COCO 2014 is a large dataset consisting of 82,783 images for training and 40,504 images for validation/testing. Although it contains 80 labeled classes, only the “person” class was used for evaluating the proposed method. Its images depict people in various viewing angles, scales and poses. The average ground-truth number of persons depicted in an image is 2.17. When considering only the images that actually contain visible people, this number increases to 4.01. Candidate detections were extracted from SSD and Faster R-CNN [18], in separate experiments, while the validation set splits were adopted from [2]. The first data subset, referred to as “minival”, contains 5000 images, while the second subset, referred to as “minitest”, contains 35000 images.

The proposed method was trained for 12 epochs. The learning rate was set to  $10^{-4}/10^{-5}/10^{-6}$  for epochs 1-8/9-11/12, respectively. GossipNet’s architecture and training again followed [2]. The final hyperparameters of all methods were selected according to the best achieved accuracy in the minival (validation) set. Table III reports the results of all competing NMS approaches.

TABLE III: COMPARISON OF DIFFERENT NMS METHODS ON THE MINITEST SET OF THE COCO DATASET, USING DETECTIONS FROM [18] AND [19]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	Input dets. from [18]		Input dets. from [19]		Average Inference Time (ms)
		$AP_{0.5}$	$AP_{0.5}^{0.95}$	$AP_{0.5}$	$AP_{0.5}^{0.95}$	
Original NMS IoU>0.4	GPU	65.4%	35.6%	56.3%	31.6%	4.3
Original NMS IoU>0.5	GPU	65.3%	35.8%	56.1%	31.6%	5.4
Original NMS IoU>0.6	GPU	63.3%	35.6%	55.5%	31.7%	6.9
TorchVision NMS IoU>0.4	GPU	65.4%	35.5%	56.3%	31.6%	0.3
TorchVision NMS IoU>0.5	GPU	65.3%	35.8%	56.1%	31.7%	0.3
TorchVision NMS IoU>0.6	GPU	63.1%	35.5%	55.5%	31.7%	0.4
Soft-NMS <sub>L</sub>	CPU	66.6%	37.0%	57.0%	32.1%	11.6
Soft-NMS <sub>G</sub>	CPU	66.3%	36.7%	57.2%	32.5%	11.7
Fast-NMS	GPU	64.3%	35.4%	55.8%	31.5%	1.6
Cluster-NMS	GPU	65.4%	35.5%	56.3%	31.6%	3.1
Cluster-NMS <sub>S</sub>	GPU	65.3%	36.1%	57.1%	31.9%	3.7
Cluster-NMS <sub>D</sub>	GPU	65.5%	35.6%	56.3%	31.6%	5.1
Cluster-NMS <sub>S+D</sub>	GPU	65.9%	36.6%	57.3%	32.1%	5.3
Cluster-NMS <sub>S+D+W</sub>	GPU	66.0%	<b>37.7%</b>	57.3%	32.1%	7.3
GossipNet	GPU	66.9%	36.1%	67.7%	36.7%	5.1
Seq2Seq-NMS	GPU	<b>67.4%</b>	37.0%	<b>68.7%</b>	<b>37.8%</b>	7.2
<b>Seq2Seq-NMS Gains</b>		+0.5%	-0.7%	+1.0%	+1.1%	-

When candidate detections from [18] were used as input, the proposed method achieves the best  $AP_{0.5}$ , equal to 67.4%, which is an improvement of +0.8% against Soft-NMS<sub>L</sub> and +0.5% against GossipNet. In terms of  $AP_{0.5}^{0.95}$ , Seq2Seq-NMS is outperformed by Cluster-NMS<sub>S+D+W</sub> and is on par with Soft-NMS<sub>L</sub>, achieving a value of 37.0%.

Using candidate detections from [19], the proposed Seq2Seq-NMS architecture achieved more significant gains: an  $AP_{0.5}$  of 68.7% and an  $AP_{0.5}^{0.95}$  of 37.8%, thus reaching

gains of +1.0% and +1.1% respectively over the second best approach.

Regarding inference time, Seq2Seq-NMS is close to that of Cluster-NMS<sub>S+D+W</sub>, but somewhat slower than GossipNet. The reported values are obtained by averaging the inference times of each method over the two separate cases (different employed detectors). Notably, the joint processing of input candidate RoIs by the neural NMS methods, compared to the non-neural ones, accomplishes more significant improvements when given inputs from the one-stage detector [19] than those from the two-stage detector [18]. In a sense, the neural NMS approaches seem to compensate for the inferior accuracy of one-stage detectors compared to the two-stage ones.

### C. CrowdHuman

The CrowdHuman dataset has been recently released to specifically target human detection in crowded areas. Crowded scenes are particularly challenging for person detectors, due to heavy visual occlusion of individual humans. The dataset contains 15000 images for training, 4370 images for validation and 5000 images for testing. The average number of persons in an image is 22.64, with various types of occlusions. Candidate detections were extracted from SSD [19] and YOLOv4 [21]. The images fed to the latter were rescaled to a resolution of  $608 \times 608$  pixels.

TABLE IV: COMPARISON OF DIFFERENT NMS METHODS ON THE CROWDHUMAN DATASET, USING DETECTIONS FROM [21] AND [19]. THE BOTTOM LINE REPORTS THE GAINS ACHIEVED WITH THE PROPOSED METHOD.

Method	Device	Input dets. from [21]		Input dets. from [19]		Average Inference Time (ms)
		AP <sub>0.5</sub>	AP <sub>0.5</sub> <sup>0.95</sup>	AP <sub>0.5</sub>	AP <sub>0.5</sub> <sup>0.95</sup>	
Original NMS IoU>0.4	GPU	78.8%	45.6%	62.6%	29.9%	8.3
Original NMS IoU>0.5	GPU	83.3%	48.2%	66.3%	31.5%	8.6
Original NMS IoU>0.6	GPU	85.3%	49.8%	67.0%	32.4%	9.8
TorchVision NMS IoU>0.4	GPU	79.1%	45.7%	62.8%	30.0%	0.3
TorchVision NMS IoU>0.5	GPU	83.5%	48.3%	66.4%	31.6%	0.3
TorchVision NMS IoU>0.6	GPU	85.3%	49.9%	66.9%	32.4%	0.4
Soft-NMS <sub>L</sub>	CPU	85.8%	51.1%	66.5%	32.3%	54.2
Soft-NMS <sub>G</sub>	CPU	84.9%	50.4%	67.1%	33.0%	58.1
Fast-NMS	GPU	84.3%	49.7%	64.8%	31.4%	2.2
Cluster-NMS	GPU	85.3%	49.9%	67.1%	32.1%	5.0
Cluster-NMS <sub>S</sub>	GPU	83.6%	49.2%	64.0%	31.0%	5.2
Cluster-NMS <sub>D</sub>	GPU	85.5%	50.4%	67.1%	32.2%	6.5
Cluster-NMS <sub>S+D</sub>	GPU	84.7%	50.1%	65.7%	31.8%	8.0
Cluster-NMS <sub>S+D+W</sub>	GPU	84.7%	50.1%	65.7%	31.9%	32.3
GossipNet	GPU	87.2%	51.0%	72.4%	35.0%	10.0
Seq2Seq-NMS	GPU	<b>87.3%</b>	<b>51.2%</b>	<b>73.9%</b>	<b>35.9%</b>	9.4
Seq2Seq-NMS Gains		+0.1%	+0.1%	+1.5%	+0.9%	-

The proposed NMS method was trained for 14 epochs. The learning rate was set to  $10^{-4}/10^{-5}/10^{-6}$  for epochs 1-8/9-12/13-14, respectively. GossipNet was trained for  $10^6$  iterations, with a learning rate set to  $10^{-4}$  and decreased by 0.1 at the  $6 \times 10^5$ -th and the  $8 \times 10^5$ -th iterations.

Table IV shows that the proposed method achieves minimal gains, in terms of AP<sub>0.5</sub> and AP<sub>0.5</sub><sup>0.95</sup>, when input candidate detections are provided by [21]. Indeed, Seq2Seq-NMS achieves an AP<sub>0.5</sub> of 87.3%, which is a +1.5% improvement against Soft-NMS<sub>L</sub> but corresponds to a minor +0.1% improvement over GossipNet. Similarly, the proposed method achieved AP<sub>0.5</sub><sup>0.95</sup> = 51.2% which corresponds to only a minor +0.1% improvement against the best competitor. However, when candidate detections are provided by [19] the proposed method achieves an AP<sub>0.5</sub> of 73.9% and AP<sub>0.5</sub><sup>0.95</sup> = 35.9%. The gains in both metrics are quite significant compared to the second-best GossipNet, achieving improvements of +1.5% and of +0.9% respectively.

Regarding inference runtime, the proposed method requires on average 9.4 ms; thus, it is faster than all non-GPU approaches and slightly faster than GossipNet. The reported values are obtained by averaging the inference times of each method over the two separate cases (different employed detectors).

### D. FMoD Ablation Study

This Subsection examines the effect of the appearance-based features extracted by FMoD on the performance of Seq2Seq-NMS. Moreover, alternative appearance-based descriptors which could replace FMoD in the overall pipeline are investigated. Experiments were performed on the CrowdHuman dataset, using [19] for providing the input raw candidate detections.

TABLE V: PERFORMANCE EVALUATION OF THE PROPOSED METHOD USING APPEARANCE-BASED ROI REPRESENTATIONS OBTAINED BY DIFFERENT FMOD VARIANTS.

Resolution of RoIs (in pixels)	Num. of Pyramid Layers	AP <sub>0.5</sub>	AP <sub>0.5</sub> <sup>0.95</sup>	Average Inference Time (ms)
20 × 20	1	73.2%	35.5%	7.4
160 × 160	1	73.3%	35.5%	7.9
20 × 20	2	73.3%	35.5%	8.3
160 × 160	2	73.4%	35.7%	8.5
20 × 20	3	73.7%	35.8%	8.9
160 × 160	3	<b>73.9%</b>	<b>35.9%</b>	9.0

The following aspects of FMoD were examined:

- the scale of RoIs used for computing the FMoD descriptors. To do so, the edge map RoIs obtained by the ROIAlign operator were extracted in a fixed resolution of either: a)  $20 \times 20$  pixels, or b)  $160 \times 160$  pixels, before computing the respective FMoD descriptors on them.
- the optimal number of FMoD spatial pyramid levels. Experiments were carried out for pyramid levels  $L$  equal to 1, 2 and 3.

As shown in Table V similar performance is attained for 1 or 2 FMoD pyramid levels, but the accuracy of Seq2Seq-NMS is improved with 3 FMoD pyramid levels. The scale of RoIs extracted by the ROIAlign operator seems to have a minimal impact on the accuracy. The reported inference

times amount to the overall time needed for computing the corresponding edge maps and extracting their appearance-based RoI representations using FMoD.

Moreover, candidate detections from [19] in the CrowdHuman dataset were also utilized in order to compare the following three variants of Seq2Seq-NMS:

- Seq2Seq-NMS that utilizes only geometry-based RoI representations. To achieve this, the DNN was fed with dummy zero-vectors as appearance-based representations.
- An extension of Seq2Seq-NMS where learnt convolutional features are employed as appearance-based RoI representations, instead of FMoD descriptors: in practice, already computed feature maps from the corresponding detector’s backbone CNN are exploited. Two variants were examined by employing the feature maps from the initial layers of VGG16 during inference. Early layers were preferred in order to retain as much spatial information as possible. The size of the selected maps, defined as tensors, were  $64 \times 64 \times 512$ , with the last dimension being the depth of the corresponding convolutional layer. In the first variant, the maps were properly resized and RoI maps were extracted using the ROIAlign operator in a  $20 \times 20$  resolution. In the second variant, a convolutional layer, with window= $1 \times 1$ , stride= $1 \times 1$ , and 32 filters followed by ReLU as activation function was employed before the ROIAlign operator. In this variant, the memory requirements induced by the ROIAlign operator were heavily reduced compared to the first variant. It must be highlighted that the ROIAlign operator is fully differentiable. A simple deep neural module, depicted in Table VI was implemented in order to compute the final appearance-based RoI representations. Seq2Seq-NMS was trained jointly with this module.
- The default Seq2Seq-NMS which uses FMoD descriptors as appearance-based RoI representations.

TABLE VI: IMPLEMENTED DEEP NEURAL MODULE IN SEQ2SEQ-NMS, TASKED WITH EXTRACTING APPEARANCE-BASED ROI DESCRIPTIONS.

<b>Conv2D + ReLU</b> , window= $3 \times 3$ , stride= $1 \times 1$ , filters=20
<b>Conv2D + ReLU</b> , window= $3 \times 3$ , stride= $1 \times 1$ , filters=4
<b>Max-Pooling</b> , window= $2 \times 2$ , stride= $2 \times 2$
<b>Flatten</b>
<b>Fully-Connected Layer + ReLU</b>

The relevant evaluation results are reported in Table VII. Default Seq2Seq-NMS with FMoD descriptors as appearance-based RoI representations improves  $AP_{0.5}$  by +0.8% and  $AP_{0.5}^{0.95}$  by +0.3%, compared to geometry-only RoI representations. A more notable improvement is demonstrated with convolutional RoI representations derived by the deep neural module: in the base case, this variant improved  $AP_{0.5}$  by +2.2% and  $AP_{0.5}^{0.95}$  by +1.3%, compared to the geometry-only Seq2Seq-NMS. The more memory-efficient variant achieved

+1.0 and +0.5% in the respective metrics. Regarding inference times, FMoD requires 2.7 ms in order to extract the corresponding appearance-based RoI representations from edge maps. If one includes the edge map computation, the corresponding inference time rises to 9.0 ms since, in our implementation, edge maps were computed in CPU; GPU alternatives may be much less time-demanding, thus significantly reducing overall inference requirements. In addition, the first variant of deep neural appearance-based RoI representations extraction requires 0.8 ms, while the more time- and memory-efficient variant requires 0.5 ms. The time needed by VGG16, in order to compute the raw feature maps is not included in the reported times.

TABLE VII: PERFORMANCE OF SEQ2SEQ-NMS ON THE CROWDHUMAN DATASET, USING DIFFERENT APPROACHES TO APPEARANCE-BASED ROI REPRESENTATION.

Type of the Appearance-based RoI Representations	$AP_{0.5}$	$AP_{0.5}^{0.95}$	Average Inference Time (ms)
Geometry-based RoI representations only (Using zero vectors as dummy appearance representations)	73.1%	35.6%	0.0
Deep neural RoI representations extracted from raw VGG16 feature maps at size $64 \times 64 \times 512$	<b>75.3%</b>	<b>36.9%</b>	0.8
Deep neural RoI representations extracted from VGG16 feature maps at size $64 \times 64 \times 32$	74.1%	36.1%	0.5
FMoD RoI representations	73.9%	35.9%	2.7 (9.0)

### E. Discussion

Overall, the proposed Seq2Seq-NMS DNN achieves top accuracy on the  $AP_{0.5}$  metric in all three datasets. The results show that Seq2Seq-NMS can successfully capture interrelations between candidate detections for the person detection task, based both on their visual appearance and their geometry. The three datasets used for evaluation contain images with a great variety of visible persons density, ranging from images of individual people to photographs of large crowds, indicating that Seq2Seq is suitable for generic person detection.

Regarding the  $AP_{0.5}^{0.95}$  metric, Seq2Seq-NMS achieves top accuracy in most cases. The main exception is COCO dataset, when using candidate detections from [19]. This behaviour can be explained by the fact that our method was specifically enforced during training to match candidate RoIs to ground-truth RoIs, in case their in-between IoU is more than 0.5, instead of doing so for various IoU thresholds in the  $[0.5, 0.95]$  range. More details about the matching strategy procedure, adopted in training, can be found in Section III.

Moving on to inference running time, the proposed method is relatively slower than non-neural, mostly less accurate, GPU-executed algorithms. However, when compared against DNN architectures for NMS, such as GossipNet, Seq2Seq-NMS achieves faster inference, with the exception of COCO (Table III). In addition, the inference runtime of Seq2Seq-NMS seems less affected by the input sequence length (number of candidate detections  $N$ ), thus achieving faster inference when processing longer sequences, as shown in, e.g., Table I.



Another observation stemming from the presented experimental results is that Seq2Seq-NMS fits well with person detectors of various types: it achieves improved  $AP_{0.5}$  performance against several competing NMS methods when combined with detectors of any nature (non-neural, one- and two-stage DNN-based). In the default Seq2Seq-NMS architecture, the use of FMoD for describing the visual appearance of the cropped candidate RoIs reinforces such a behaviour, since FMoD descriptors are independent of the employed person detector.

In addition, as shown in the ablation study presented in Section IV-D, the use of appearance-based ROI representations from FMoD indeed improves the performance of Seq2Seq-NMS, compared to the case where only geometry-based representations are used. The same study showed that the best accuracy is achieved when the appearance-based features are computed using three FMoD pyramid levels, whereas the scale of RoIs has minimal impact on accuracy. Finally, a simple variant of Seq2Seq-NMS that exploits deep neural appearance-based ROI representations from internal feature maps of the employed detector, instead of FMoD descriptors, further improves accuracy as shown in Table VI.

Besides the results depicted in Tables I, II III and IV, an ablation study was also performed regarding the proposed masking operation (described in Section III-C) of the self-attention mechanism. Omitting masking led to reduced accuracy rates, or even training convergence failures in cases with huge numbers of candidate RoIs per image. The importance of masking stems from the fact that it enforces an ordering constraint on how the internal representation of each candidate detection is shaped: thanks to masking, its form is finalized by attending mainly to representations that correspond to RoIs higher-scoring than itself, using the Scaled Dot-Product Attention mechanism. Thus, in our view, this finding supports the validity of the sequence-to-sequence formulation of the NMS task.

## V. CONCLUSIONS

Detecting humans accurately is crucial for human safety-centric applications, but also extremely challenging. Large variations in human poses and high levels of occlusions negatively affect person detection accuracy. Non-Maximum Suppression (NMS) is the last step in a typical object detection system, which is also affected by such challenges. This paper presented Seq2Seq-NMS, a novel deep neural architecture for performing NMS in similar hard cases, relying on a reformulation of NMS as a sequence-to-sequence problem. The proposed method utilises the Multihead Scaled Dot-Product Attention mechanism, in order to efficiently capture interrelations across the sequence of candidate detections, while also jointly exploiting visual appearance and geometric properties of the input RoIs in order to better represent them. Quantitative evaluation on three public person detection datasets showed that Seq2Seq-NMS can provide state-of-the-art results at the IoU threshold used for annotating its training dataset, with acceptable inference runtime requirements. Future extensions

may focus on a training strategy suitable for various IoU thresholds and on adapting the proposed method to multiclass object detection.

## VI. ACKNOWLEDGMENTS

This project has received funding from the European Union’s Horizon 2020 Research and Innovation programme under grant agreement No. 871449 (OpenDR). 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] S. Liu, D. Huang, and Y. Wang, “Adaptive NMS: Refining pedestrian detection in a crowd,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [2] J. Hosang, R. Benenson, and B. Schiele, “Learning Non-Maximum Suppression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [3] C. Symeonidis, I. Mademlis, N. Nikolaidis, and I. Pitas, “Improving neural Non-Maximum Suppression for object detection by exploiting interest-point detectors,” in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019.
- [4] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, “Relation networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [5] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.
- [6] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2009.
- [7] I. Mademlis, N. Nikolaidis, and I. Pitas, “Stereoscopic video description for key-frame extraction in movie summarization,” in *Proceedings of the EURASIP European Signal Processing Conference (EUSIPCO)*, 2015.
- [8] I. Mademlis, N. Nikolaidis, A. Tefas, I. Pitas, T. Wagner, and A. Messina, “Autonomous unmanned aerial vehicles filming in dynamic unstructured outdoor environments,” *IEEE Signal Processing Magazine*, vol. 36, pp. 147–153, 2018.
- [9] C. Symeonidis, E. Kakaletsis, I. Mademlis, N. Nikolaidis, A. Tefas, and I. Pitas, “Vision-based UAV safe landing exploiting lightweight deep neural networks,” in *Proceedings of the International Conference on Image and Graphics Processing (ICIGP)*, 2021.
- [10] C. Papaioannidis, I. Mademlis, and I. Pitas, “Autonomous UAV safety by visual human crowd detection using multi-task deep neural networks,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [11] E. Kakaletsis, E. Symeonidis, M. Tzelepi, I. Mademlis, T. A., N. Nikolaidis, and I. Pitas, “Computer vision for autonomous UAV flight safety: An overview and a vision-based safe landing pipeline example,” *ACM Computing Surveys*, 2021.
- [12] I. Mademlis, V. Mygdalis, N. Nikolaidis, and I. Pitas, “Challenges in autonomous UAV cinematography: an overview,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2018.
- [13] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, “Embedded UAV real-time visual object detection and tracking,” in *Proceedings of the IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2019.
- [14] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [15] P. Viola and M. Jones, “Robust real-time face detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001.
- [16] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [17] S. Tang, M. Andriluka, A. Milan, K. Schindler, S. Roth, and B. Schiele, “Learning people detectors for tracking in crowded scenes,” in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2013, pp. 1049–1056.

- [18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [20] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.
- [21] A. Bochkovskiy, C.-Y. Wang, and H. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *ArXiv*, vol. abs/2004.10934, 2020.
- [22] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
- [23] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS: Improving object detection with one line of code," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [24] H. Zhou, Z. Li, C. Ning, and J. Tang, "CAD: Scale invariant framework for real-time object detection," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017.
- [25] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [26] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [27] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 34, no. 07, 2020.
- [28] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Transactions on Cybernetics*, pp. 1–13, 2021.
- [29] S. H. Rezaatofghi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [30] X. Huang, Z. Ge, Z. Jie, and O. Yoshie, "NMS by representative region: Towards crowded pedestrian detection by proposal pairing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS)*, 2017.
- [32] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, "Compact video description and representation for automated summarization of human activities," in *Proceedings of the INNS Conference on Big Data*, 2016.
- [33] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas, "Multimodal stereoscopic movie summarization conforming to narrative characteristics," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5828–5840, 2016.
- [34] I. Mademlis, A. Tefas, and I. Pitas, "A salient dictionary learning framework for activity video summarization via key-frame extraction," *Information Sciences*, vol. 432, pp. 319 – 331, 2018.
- [35] I. Mademlis, A. Tefas, and I. Pitas, "Regularized SVD-based video frame saliency for unsupervised activity video summarization," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.
- [36] —, "Greedy salient dictionary learning with optimal point reconstruction for activity video summarization," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018.
- [37] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [38] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [39] C.-F. Chen, Q. Fan, and R. Panda, "CrossVit: Cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 357–366.
- [40] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. S. A. Ku, and D. Tran, "Image transformer," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.
- [41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [42] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.
- [43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [45] J. M. Ferryman and A. Ellis, "PETS2010: Dataset and challenge," in *Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2010.

### **8.3 Improving Unimodal Inference with Multimodal Transformers**

The appended paper follows.

# IMPROVING UNIMODAL INFERENCE WITH MULTIMODAL TRANSFORMERS

Kateryna Chumachenko<sup>†</sup>, Alexandros Iosifidis<sup>\*</sup>, Moncef Gabbouj<sup>†</sup>

<sup>†</sup>Tampere University, Faculty of Information Technology and Communication Sciences, Finland

<sup>\*</sup>Aarhus University, Department of Electrical and Computer Engineering, Denmark

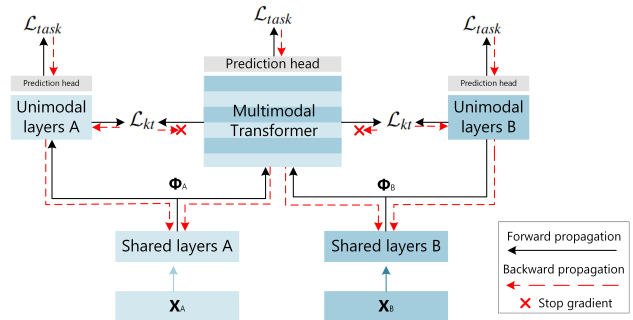
## ABSTRACT

This paper proposes an approach for improving performance of unimodal models with multimodal training. Our approach involves a multi-branch architecture that incorporates unimodal models with a multimodal transformer-based branch. By co-training these branches, the stronger multimodal branch can transfer its knowledge to the weaker unimodal branches through a multi-task objective, thereby improving the performance of the resulting unimodal models. We evaluate our approach on tasks of dynamic hand gesture recognition based on RGB and Depth, audiovisual emotion recognition based on speech and facial video, and audio-video-text based sentiment analysis. Our approach outperforms the conventionally trained unimodal counterparts. Interestingly, we also observe that optimization of the unimodal branches improves the multimodal branch, compared to a similar multimodal model trained from scratch.

## 1. INTRODUCTION

The availability of an abundance of data in the modern world has driven the development of machine learning methods exploiting such data to their fullest. Recently, there has been an increase in emergence of novel approaches utilizing multiple data modalities simultaneously, such as video, audio, text, or other sensor data, for solving a variety of tasks [1, 2]. Such methods are referred to as multimodal methods and they have been proven successful in a plethora of application fields, including emotion recognition [3], hand gesture recognition [4], human activity recognition [5], and others. Leveraging multiple data sources concurrently can lead to improved performance of the learning model as data of different modalities can complement and enrich each other.

Research within the field of multimodal methods has been largely focused on tasks where all modalities of interest are assumed to be present both during training and test stages, and has involved development of novel feature fusion methods [5], solving multimodal alignment problems [6], etc. Nevertheless, it is not always desirable to rely on the assumption of all modalities of interest being present at inference time. In real-world applications, data of one or multiple modalities might be unavailable at arbitrary inference steps due to, e.g., transmission delays and media failures, or simply the application at hand might not be suitable for utilizing certain modalities, while they might be available during training. Utilization of unimodal models therefore remains widely adopted due to their simplicity and easier applicability to real-world tasks. Nevertheless, models relying only on unimodal data at inference time can benefit from multimodal training. Such approach can aid in learning richer



**Fig. 1:** Description of the proposed framework. For a two modality case A and B, the architecture is comprised of two unimodal branches and a joint multimodal Transformer branch. Early feature extraction layers are shared between multimodal Transformer and corresponding unimodal branches, and both uni- and multimodal branches have their own task-specific heads. Additionally, unimodal branches optimize knowledge transfer criteria from multimodal Transformer, while multimodal branch is not updated based on this criterion. At inference time, multimodal branch is dropped and each of the unimodal branches can be used as a standard unimodal model (alternatively, multimodal branch can be used on its own, too).

feature representations from single modality by relating it with other modalities, and help highlight unimodal information that is most relevant for the task. At the same time, the computational costs associated with the model are not increased.

In this work, we propose an approach for improving performance of unimodal models with multimodal training, and employ a multi-branch architecture with both unimodal, and multimodal Transformer-based branches. Unimodal and multimodal branches are co-trained and knowledge from the stronger multimodal branch is continuously transferred to the unimodal branches via a multi-task objective, hence improving the performance of resulting unimodal models. We perform experiments on three multimodal tasks and observe consistent improvements in the performance of the models. At the same time, we also observe that our approach not only improves the performance of unimodal models, but also that of the multimodal teacher model, compared to the similar model trained from scratch. Our contributions can be summarized as follows:

- We propose an approach for improving the performance of arbitrary unimodal models with multimodal training, with no additional computational cost incurred by unimodal model at inference time;
- The proposed framework is agnostic of the underlying modalities or unimodal architecture types, while in the experiments we showcase various architectures, including 3D-CNNs, 2D+1D-CNNs, and transformer-based ones;
- We validate our approach on three multimodal tasks and ob-

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 871449 (OpenDR).

serve consistent improvements, with different modalities, architectures, and loss functions.

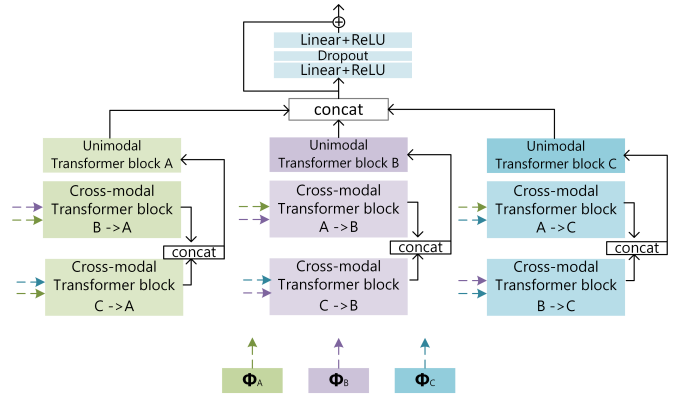
## 2. RELATED WORK

Modern research directions in the field of multimodal learning have largely focused on advanced modality fusion methods [2, 1, 7] and include a variety of approaches, ranging from CNN-based cross-modal Squeeze-and-Excitation blocks [5], to translation based approaches [8]. Within the field of multimodal fusion, perhaps the most notable recent development is the adoption of multimodal Transformers that allow to capture global correspondences between modalities, hence making them an especially favorable choice for temporal sequence modelling tasks where alignment between modalities is an important challenge [3, 9, 10]. The idea behind cross-modal Transformers lies in adoption of self-attention mechanism [11] with queries and key-value pairs originating from different modalities, and one of the most notable instantiations of such approach is the Multimodal Transformer (MULT) [6].

Nevertheless, the above-mentioned approaches have their limitations. Primarily, they all rely on the assumption that the same set of sensors/modalities are available at both training and inference, while such expectation is idealistic and is an especially relevant limitation for real-world applications where flexibility is required. A set of methods aim to solve this issue by introducing the multimodal training unimodal testing paradigm, aiming at improving unimodal models by utilizing multimodal data during training. Such methods can be broadly categorized into a few types, with the first type being the methods aiming to reconstruct or otherwise hallucinate a missing modality [12, 13, 14, 15]. Other methods optimize certain alignment objectives between multiple modalities, e.g., by contrastive learning [16], or by spatiotemporal semantic alignment [17]. Nevertheless, such methods are mostly suited for well-paired modality types, such as RGB and Depth, or RGB and Point Clouds, while having limited suitability for modalities where data types are drastically different and their correspondence is not immediately obvious, e.g., audio and RGB frames, or text and RGB frames. In our work, we take aim to overcome this issue, and propose a generalized framework suitable for various data modalities and unimodal architectures.

## 3. PROPOSED APPROACH

This section describes the proposed approach for improving the performance of an arbitrary unimodal model with multimodal training. We consider the following problem setup: given a set of data representations of arbitrary modalities and corresponding unimodal model architectures, we seek to improve performance of said unimodal models by exploiting multimodal information during training. Concretely, our approach relies on a general framework in which unimodal models are united in a joint architecture by a multimodal Transformer-based branch attached to intermediate features of unimodal models of each modality, hence each unimodal model becomes a separate branch. The multimodal branch is jointly co-trained with resulting unimodal branches, and shares early feature extraction layers with the unimodal branches. Additionally, knowledge transfer between the multimodal Transformer and the unimodal branches is achieved by optimizing a multi-task objective. During inference, the multimodal branch as well as branches corresponding to modalities that are not of interest are dropped, restoring the original architecture of the unimodal model, but with parameters optimized during multimodal training. Overall, a schematic represen-



**Fig. 2:** Example of a multimodal Transformer with three modalities A, B, and C.

tation of the proposed approach, with two example modalities  $A$  and  $B$ , is outlined in Figure 1.

As can be seen, data of each modality  $i$ ,  $\mathbf{X}_i$ , is input to a sequence of layers serving as backbone for both unimodal and multimodal branches, resulting in feature representation  $\Phi_i$  for modality  $i$ . Further,  $\Phi_i$  is processed with the remaining part of the unimodal branch, as well as the multimodal Transformer branch (as described further) independently, where each branch has its own task-specific head that optimizes the task-specific objective  $\mathcal{L}_{task}$  (e.g., cross-entropy for classification tasks). Additionally, a knowledge transfer objective from stronger multimodal branch to weaker unimodal branches  $\mathcal{L}_{kt}$  is optimized, where  $\mathcal{L}_{kt}$  can be represented by a variety of different objective functions, as will be discussed further.

Unimodal and multimodal branches as well as task-specific and knowledge transfer objectives are optimized jointly. Shared feature layers receive gradient updates from task-specific objectives of both uni- and multimodal branches, hence forcing them to remain informative for both inference paths and avoiding the loss of modality-specific information, while retaining information useful for modality fusion. In turn, knowledge transfer objective encourages the remaining segment of unimodal branch to learn in accordance with the multimodal transformer, hence improving its performance.

### 3.1. Multimodal Transformer

Here, we describe the multimodal Transformer branch. Given feature representations of two modalities  $\Phi_A$  and  $\Phi_B$ , cross-modal attention that fuses modality  $B$  into modality  $A$  is defined as:

$$\hat{\Phi}_{AB} = softmax \left( \frac{\mathbf{W}_q \Phi_A \Phi_B^T \mathbf{W}_k^T}{\sqrt{d}} \right) \mathbf{W}_v \Phi_B, \quad (1)$$

followed by another linear projection layer, where  $\mathbf{W}_q$ ,  $\mathbf{W}_v$ , and  $\mathbf{W}_k$  are learnable projection matrices,  $d$  is the feature dimensionality, and  $\Phi_A$  and  $\Phi_B$  are features of modalities  $A$  and  $B$ . This is generally referred to as cross-attention and it is a generalization of the self-attention mechanism [11] where queries originate from modality  $A$  and key-value pairs originate from modality  $B$ . Similarly, fusion of modality  $B$  into modality  $A$  is achieved by learning queries from modality  $B$  and key-value pairs from modality  $A$ .

The overall multimodal Transformer branch is similar to the one proposed in [6] and consists of the previously defined cross-attention blocks, optionally followed by unimodal self-attention blocks in each modality, as shown in Figure 2. That is, for fusion of two modalities  $A$  and  $B$ , two cross-attention blocks  $A \rightarrow B$  and  $B \rightarrow A$  are employed and their resulting features concatenated,

and in the case where the number of modalities is greater than two, pair-wise cross-attention blocks are calculated within each pair. The prediction head is unimodal model-specific.

### 3.2. Unimodal branches

The proposed approach is agnostic of underlying unimodal models and can be combined with an arbitrary architecture. For the sake of completeness, we describe several examples of architectures used in our experimental evaluation further. For the task of dynamic gesture recognition based on RGB and Depth modalities, each unimodal branch is either an I3D [18] or MobileNetv2 [19] architecture, primarily based on 3D convolutional layers. The multimodal branch in I3D variant is attached after “*MixedAf*” layer, and in the case of MobileNetv2, prior to the last two convolutional blocks. Hence, the majority of the layers is shared between the multimodal and unimodal branches. The extracted 3D convolutional features  $\Phi$  have the shape of  $B \times C \times T \times H \times W$ , on which we perform spatial mean pooling, resulting in  $B \times C \times T$  input tokens input to the multimodal Transformer. For the task of audiovisual emotion recognition, we adopt an architecture similar to [3], with vision branch being the EfficientNet backbone followed by blocks of 1D-Convolutional layers, and audio branch is also a set of 1D-Convolutional layers. Here, we add multimodal Transformer branch on the output of “Stage 1” convolutional block in both branches. This can be compared to ‘intermediate transformer’ fusion described in [3], where outputs of multimodal Transformers are not fused back to their corresponding branches, but instead connect to their own output layer.

### 3.3. Multi-task training objective

The overall training objective is given by

$$\mathcal{L} = \alpha \sum_{i=1}^M \mathcal{L}_{kt}^i + \beta \sum_{i=1}^M \mathcal{L}_{task}^i + \gamma \mathcal{L}_{task}^{mm}, \quad (2)$$

where  $i$  is the modality indicator,  $\mathcal{L}_{task}^i$  is task-specific objective for branch of modality  $i$ ,  $\mathcal{L}_{task}^{mm}$  is the task-specific objective of the multimodal branch, and  $\mathcal{L}_{kt}^i$  is the knowledge transfer loss from multimodal branch to unimodal branch  $i$ , and  $\alpha, \beta, \gamma$  are scaling coefficients. A multitude of objective functions can serve the purpose of knowledge transfer. Here, we consider three cases, which we refer to as *decision-level alignment*, *feature-level alignment*, and *attention-level alignment*.

In *decision-level alignment* objective, the goal is to transfer high-level information about predictions and class probability distributions from stronger multimodal branch to weaker unimodal branch. To achieve this, for standard classification tasks, we formulate knowledge transfer as knowledge distillation task [20] and optimize KL-divergence  $\mathcal{L}_{kt}^{KL}$  between soft pseudo-labels generated by multimodal branch and softmax outputs of unimodal branches. Soft probability distribution between classes is achieved by applying temperature  $T > 1$  to predicted class probabilities. Such knowledge transfer allows the unimodal model to capture fine-grained class boundaries from the stronger multimodal model.

In *feature-level alignment* objective, the goal is to transfer broader semantic feature-level information from multi- to unimodal branch. Such formulation can be more general and suitable for a wider variety of tasks. For this goal, we adopt cosine similarity  $\mathcal{L}_{kt}^{cos} = \frac{\phi_A \cdot \phi_B}{\|\phi_A\| \cdot \|\phi_B\|}$  between the final hidden layer output features of the multimodal and unimodal branches, hence promoting the transfer of feature-level semantic information, aimed at improving the performance of task at hand.

Lastly, when unimodal branch architectures are also Transformer-based, a mechanism that we refer to as *attention-level alignment* can be employed. Here, knowledge transfer can be achieved by aligning self-attention probability distributions over temporal tokens in unimodal and multimodal branches. Intuitively, tokens in multimodal Transformer attend to tokens of other modalities globally via self-attention in cross-modal Transformer blocks. Subsequently, unimodal Transformer blocks in multimodal Transformer operate over tokens that have already ‘seen’ corresponding tokens of other modalities. The softmax probabilities of unimodal self-attention in final stages of multimodal Transformer can then be distilled to the corresponding unimodal branches similarly to the first case, by calculating KL-divergence over soft pseudo-labels. We further refer to this approach and objective function as  $\mathcal{L}_{kt}^{att}$ .

## 4. EXPERIMENTAL EVALUATION

As described earlier, to the best of our knowledge the few existing methods aimed at unimodal inference with multimodal training are primarily suitable for well-paired modalities as they rely on fine-grained spatial information transfer or modality reconstruction/hallucination. This makes their application in more general scenarios and more heterogeneous modalities largely non-trivial if not impossible. On the other hand, our proposed approach is generalized and makes no assumption on the underlying data. Therefore, to show the effectiveness of our method, we compare the models trained within our framework to unimodal counterparts proposed in recent literature [3, 6, 18, 19] on a variety of tasks and modalities of different types, and show that our proposed approach improves their performance. We perform experiments on three tasks / datasets: egocentric dynamic gesture recognition using EgoGesture dataset [4], audiovisual emotion recognition using RAVDESS dataset [21], and multimodal sentiment analysis on CMU-MOSEI dataset [22]. We train independently unimodal models with available modalities; multimodal model comprised of shared layers and multimodal Transformer; and the proposed multimodal architecture with knowledge transfer trained jointly, where we evaluate each of the resulting unimodal and multimodal branches independently. In each dataset, we report the performance on the test set, with the model selected based on best performance on the validation set. Each modality model is selected independently from other modalities and knowledge transfer loss weight is a hyperparameter. Best result is highlighted in bold, and results outperforming the baseline are underlined.

Method	Acc-RGB	Acc-Depth	Acc-MM
MobileNetv2 [19]	86.07	86.67	87.64
MobileNetv2- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>88.57</b>	<b>88.34</b>	<b>89.19</b>
I3d [18]	90.69	90.64	91.78
I3d- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>91.96</b>	<b>91.84</b>	<b>92.78</b>
Ablation studies			
I3d- $\mathcal{L}_{kt}^{KL}$ , no know. trans.	90.54	90.32	92.32
I3d- $\mathcal{L}_{kt}^{KL}$ (ours) - frozen	91.74	91.82	92.73

**Table 1:** Results on EgoGesture dataset.

**Hand gesture recognition.** For egocentric dynamic hand gesture recognition, we use EgoGesture dataset [4, 23], which is a hand gesture recognition dataset comprised of RGB and Depth modalities and including 83 hand gesture classes depicted in 24,161 short hand gesture clips, performed by 50 subjects. Unimodal branches are as

described in Sec. 3.2, and multimodal branch is comprised of a multimodal Transformer attached to intermediate layers of Depth and RGB branches. As this task is formulated as a video classification problem, we adopt decision-level alignment for knowledge transfer, and minimize KL-divergence with  $T = 5$  between soft output probability distributions of multimodal and unimodal branches.

Table 1 shows the results of the proposed approach. As can be seen, the proposed training framework outperforms the unimodal counterparts on both modalities and both architectures, leading to up to 2.5% improvement in accuracy. Interestingly, we observe that the proposed approach also improves the performance of the multimodal branch when it is trained in conjunction with unimodal branches, compared to the multimodal branch trained independently. This shows that providing unimodal feedback during training forces the shared feature layers to retain more information specific to each independent modality, hence improving the multimodal performance.

Method	Acc-Audio	Acc-Video	Acc-MM
Unimodal models [3]	60.92	60.00	64.92
MM- $\mathcal{L}_{kt}^{KL}$ (ours)	<b>63.16</b>	<b>63.16</b>	<b>66.33</b>

**Table 2:** Results on RAVDESS dataset.

**Audiovisual emotion recognition.** For audiovisual emotion recognition we employ the RAVDESS dataset [21] which consists of face and speech recordings of 24 actors acting out 8 emotions and posing a classification task, with 60 video sequences recorded for each actor. The architecture follows the description in Section 3.2, with unimodal models trained from scratch. Knowledge transfer loss  $\mathcal{L}_{kt}$  is the KL-divergence between soft outputs with  $T = 5$  and the task-specific loss is standard cross-entropy. Table 2 shows the results obtained in audiovisual emotion recognition tasks. As can be seen, the findings are consistent with those obtained in previous task, and the proposed approach improves both unimodal counterparts by up to 3%. Similarly, the multimodal branch is improved as well.

Method	MAE	Corr	Acc_7
Audio [6]	0.8146	0.2395	41.05
A- $\mathcal{L}_{kt}^{cos}$ (ours)	<u>0.8125</u>	<b>0.2812</b>	40.76
A- $\mathcal{L}_{kt}^{att}$ (ours)	<b>0.8111</b>	<u>0.2493</u>	<b>41.17</b>
Vision [6]	0.8079	0.2313	42.18
V- $\mathcal{L}_{kt}^{cos}$ (ours)	<u>0.8028</u>	<b>0.2774</b>	<u>42.18</u>
V- $\mathcal{L}_{kt}^{att}$ (ours)	<b>0.7978</b>	<u>0.2680</u>	<b>42.73</b>
Text [6]	0.6290	0.6481	48.72
T- $\mathcal{L}_{kt}^{cos}$ (ours)	<b>0.6199</b>	<b>0.6570</b>	<b>49.62</b>
T- $\mathcal{L}_{kt}^{att}$ (ours)	<u>0.6203</u>	<u>0.6537</u>	<u>49.02</u>
Multimodal [6]	0.6407	0.6748	48.72
MM- $\mathcal{L}_{kt}^{cos}$ (ours)	<b>0.6273</b>	<b>0.6793</b>	<b>49.32</b>
MM- $\mathcal{L}_{kt}^{att}$ (ours)	<u>0.6331</u>	0.6625	<u>49.09</u>

**Table 3:** Results on MOSEI dataset.

**Multimodal sentiment analysis** Next, for the task of multimodal sentiment analysis, we perform experiments on the unaligned version of CMU-MOSEI dataset [22], which contains 23,454 utterances extracted from movie review video clips taken from YouTube. The dataset consists of audio, vision, and text modalities, where each utterance is labeled with a sentiment score in the range  $[-3, \dots, 3]$  by human annotators. Since the dataset poses the regression task, the model is optimized with  $L1$  loss as task-specific objective, and we evaluate both feature-level and attention-level alignment knowledge transfer objectives  $\mathcal{L}_{kt}^{cos}$  and  $\mathcal{L}_{kt}^{att}$ . We follow the standard protocol of the dataset and report mean average error, correlation with human annotations (annotations are obtained from multiple annotators), and 7-class accuracy. We report average results over 3 random

	RGB	Depth	MM
MobileNetv2 [19]	86.07	86.67	87.64
$\alpha=1$	87.24	87.78	88.87
$\alpha=5$	<b>88.57</b>	<b>88.34</b>	<b>89.19</b>
$\alpha=10$	87.80	87.82	88.35
$\alpha=20$	87.18	87.36	88.18

**Table 4:** Results with different  $\alpha$  on EgoGesture

seeds. Unimodal models are as described in Sec. 3.2 and follow the method of [6], and the multimodal branch is identical to Figure 2.

Table 3 shows the results on the CMU-MOSEI dataset. Firstly, we observe that in our baseline experiments, text-only model outperforms the multimodal one (which is rather consistent with previous works, where text modality performance often lies close to the multimodal one [6]), while the text model trained under our proposed framework outperforms both of them. In fact, the proposed approach outperforms the baselines on all the modalities compared to unimodal models, with especially big increase observed in correlation metric, and the multimodal branch also outperforms the multimodal model trained independently. We observe that feature-level loss is more beneficial for improving the stronger text modality, and subsequently the multimodal branch. In turn, attention-level alignment shows to be more beneficial for audio and vision modalities. This shows that multimodal branch is mainly driven by the text modality (judging by their performance), hence features of the final hidden layer are likely to be more easily transferable to unimodal text branch than audio or vision branches. Instead, audio and vision branches can benefit from softer attention-level alignment, which does not enforce strong similarity to other modality, but instead, to tokens of the same modality enriched with multimodal information.

**Ablation studies** We perform a few ablations on the EgoGesture dataset. First, as our primary goal is to improve the unimodal branches, we train an architecture identical to the one described earlier, but the shared weights are only updated from the uni-modal branch, and are frozen in the multimodal path. The results can be seen in Table 1, the freezing of the layers does not have a significant effect on the model, with unimodal models being marginally below the standard variant. Next, we investigate the effect of the knowledge transfer loss and train the identical model but without optimization of the knowledge transfer objective from the multi-modal to the uni-modal branch. As can be seen in Table 1, multi-modal branch still outperforms the one trained from scratch (showcasing again the benefits of unimodal gradient updates to the shared layers), but unimodal branches retain the unimodal performance, hence showing the effect of the knowledge transfer loss. We are additionally providing ablations on the  $\alpha$  (coefficient of the knowledge transfer loss), with  $\beta$  and  $\gamma$  (task-specific losses) fixed to 1, which can be seen in Table 4 using EgoGesture dataset and MobileNetV2. As can be seen, any  $\alpha$  outperforms the baseline, while the best result is achieved at  $\alpha=5$ .

## 5. CONCLUSION

We have presented a general framework for improving performance of an arbitrary unimodal model with multimodal training that involves co-training of the unimodal models with multimodal Transformer and multi-task objective aimed at knowledge transfer from multimodal to unimodal branches. The proposed approach shows improved performance on 3 tasks of different modalities and structures. We also found that providing unimodal feedback to early layers of multimodal model aids its performance in a multimodal setting. Future work may include research on higher adaptiveness of the co-training, such that not all unimodal models are co-trained in the same manner, but instead relatively to their capacity.

## 6. REFERENCES



- [1] Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman, “Self-supervised multimodal versatile networks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 25–37, 2020.
- [2] Ronghang Hu and Amanpreet Singh, “Unit: Multimodal multitask learning with a unified transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1439–1449.
- [3] Kateryna Chumachenko, Alexandros Iosifidis, and Moncef Gabbouj, “Self-attention fusion for audiovisual emotion recognition with incomplete data,” in *26th International Conference on Pattern Recognition*. IEEE, 2022, pp. 2822–2828.
- [4] Yifan Zhang, Congqi Cao, Jian Cheng, and Hanqing Lu, “Egogesture: a new dataset and benchmark for egocentric hand gesture recognition,” *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1038–1050, 2018.
- [5] Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L Iuzzolino, and Kazuhito Koishida, “Mmtm: Multimodal transfer module for cnn fusion,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13289–13299.
- [6] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov, “Multimodal transformer for unaligned multimodal language sequences,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019, p. 6558.
- [7] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency, “Multimodal machine learning: A survey and taxonomy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2018.
- [8] Jihyun Lee, Binod Bhattarai, and Tae-Kyun Kim, “Face parsing from rgb and depth using cross-domain mutual learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1501–1510.
- [9] Jian Huang, Jianhua Tao, Bin Liu, Zheng Lian, and Mingyue Niu, “Multimodal transformer fusion for continuous emotion recognition,” in *International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 3507–3511.
- [10] DN Krishna and Ankita Patil, “Multimodal emotion recognition using cross-modal attention and 1d convolutional neural networks,” in *Interspeech*, 2020, pp. 4243–4247.
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [12] Nuno C Garcia, Pietro Morerio, and Vittorio Murino, “Learning with privileged information via adversarial discriminative modality distillation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2581–2593, 2019.
- [13] Nuno C Garcia, Pietro Morerio, and Vittorio Murino, “Modality distillation with multiple stream networks for action recognition,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 103–118.
- [14] Wenbin Teng and Chongyang Bai, “Unimodal face classification with multimodal training,” in *Proceedings of the 16th International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE, 2021, pp. 1–5.
- [15] Giorgio Giannone and Boris Chidlovskii, “Learning common representation from rgb and depth images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [16] Johannes Meyer, Andreas Eitel, Thomas Brox, and Wolfram Burgard, “Improving unimodal object recognition with multimodal contrastive learning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5656–5663.
- [17] Mahdi Abavisani, Hamid Reza Vaezi Joze, and Vishal M Patel, “Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1165–1174.
- [18] Joao Carreira and Andrew Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.
- [19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.
- [20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [21] Steven R Livingstone and Frank A Russo, “The ryerson audiovisual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english,” *PLoS one*, vol. 13, no. 5, 2018.
- [22] Amir Zadeh, Ali Bagher, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency, “Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2236–2246.
- [23] Congqi Cao, Yifan Zhang, Yi Wu, Hanqing Lu, and Jian Cheng, “Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3763–3771.



## **8.4 Deep learning for active robotic perception**

The appended paper follows.

# Deep learning for active robotic perception

Nikolaos Passalis<sup>1</sup><sup>a</sup>, Pavlos Tosidis<sup>1</sup>, Theodoros Manousis<sup>1</sup>, and Anastasios Tefas<sup>1</sup><sup>b</sup>

<sup>1</sup>*Computational Intelligence and Deep Learning group, AIIA Lab,  
Department of Informatics, Aristotle University of Thessaloniki, Greece  
{passalis, ptosidis, tmanousis, tefas}@csd.auth.gr*

Keywords: active perception, deep learning, active vision, active robotic perception

Abstract: Deep Learning (DL) has brought significant advancements in recent years, greatly enhancing various challenging computer vision tasks. These tasks include but are not limited to object detection and recognition, scene segmentation, and face recognition, among others. DL's advanced perception capabilities have also paved the way for powerful tools in the realm of robotics, resulting in remarkable applications such as autonomous vehicles, drones, and robots capable of seamless interaction with humans, such as collaborative manufacturing. However, despite these remarkable achievements in DL within these domains, a significant limitation persists: most existing methods adhere to a static inference paradigm inherited from traditional computer vision pipelines. Indeed, DL models typically perform inference on a fixed and static input, ignoring the fact that robots possess the capability to interact with their environment to gain a better understanding of their surroundings. This process, known as "active perception", closely mirrors how humans and various animals interact and comprehend their environment. For instance, humans tend to examine objects from different angles, when being uncertain, while some animals have specialized muscles that allow them to orient their ears towards the source of an auditory signal. Active perception offers numerous advantages, enhancing both the accuracy and efficiency of the perception process. However, incorporating deep learning and active perception in robotics also comes with several challenges, e.g., the training process often requires interactive simulation environments and dictates the use of more advanced approaches, such as deep reinforcement learning, the deployment pipelines should be appropriately modified to enable control within the perception algorithms, etc. In this paper, we will go through recent breakthroughs in deep learning that facilitate active perception across various robotics applications, as well as provide key application examples. These applications span from face recognition and pose estimation to object detection and real-time high-resolution analysis.


## 1 INTRODUCTION


In recent years, Deep Learning (DL) has led to significant advancements in a range of challenging computer vision and robotic perception tasks (LeCun et al., 2015). These tasks encompass but are not restricted to object detection and recognition (Redmon et al., 2016), scene segmentation (Badrinarayanan et al., 2017), and face recognition (Wen et al., 2016), among others. DL's sophisticated perception capabilities have also yielded potent tools for diverse robotics applications, resulting in the emergence of impressive use cases, such as self-driving vehicles (Bojarski et al., 2016), unmanned aerial vehicles (drones) (Passalis et al., 2018), and robots capable of seamless interaction with humans, notably in collaborative man-

ufacturing scenarios (Liu et al., 2019).

Despite the recent accomplishments of DL in these domains, a notable limitation plagues most existing approaches since they adhere to a static inference paradigm, which follows the traditional computer vision pipeline. Therefore, DL models perform inference on a fixed and static input, ignoring the ability of robots, as well as cyber-physical systems (Li, 2018; Loukas et al., 2017), to *interact* with their environment in order to enhance their perception. For example, we can consider the task of face recognition, where a robot captures a suboptimal profile view of a subject to be recognized. A conventional static perception-based DL model may struggle to identify the subject from a specific angle, particularly if it lacks training on profile face images for such angles. However, it is often feasible for the robot to attain a better and more distinguishing view by adjusting its position relative to the human subject. Consequently,

---

<sup>a</sup> <https://orcid.org/0000-0003-1177-9139>

<sup>b</sup> <https://orcid.org/0000-0003-1288-3667>

in such scenarios, the same DL model will likely succeed in recognizing the subject after the robot repositions itself for a more suitable angle. This methodology, known as *active perception* (Aloimonos, 2013; Bajcsy et al., 2018; Shen and How, 2019), enables the manipulation of the robot or sensor to obtain a clearer and more informative view or signal, ultimately enhancing the perception capabilities and situational awareness of robotic systems. Note that this process closely mirrors how humans and various animals engage with and perceive their surroundings. For instance, humans tend to explore different perspectives when processing complex visual stimuli, while many mammals possess specialized ear muscles that pivot their ears toward the source of an auditory signal in order to acquire a clearer version of the signal (Heffner and Heffner, 1992).

A number of recent, although relatively basic, approaches have illustrated that active perception can indeed enhance the perceptual capabilities of various models. For example, works such as (Ammirato et al., 2017) and (Passalis and Tefas, 2020), demonstrated that developing a deep learning system that predicts the next best move for a robot can significantly improve the accuracy of various perception tasks, such as object detection and face recognition, where the viewing angle, occlusions and the scale of each object can have a significant effect on the perception accuracy. Similar findings have also been documented in more recent research spanning a variety of domains (Han et al., 2019; Tosidis et al., 2022; Kakaletsis and Nikolaidis, 2023). It is also important to emphasize that active perception methodologies can also enable the development of less computationally intensive deep learning models. This occurs because these models are trained to address a less complex problem. For example, in (Passalis and Tefas, 2020), it is demonstrated that more lightweight face recognition models can be used when DL models can actively interact with the environment in order to acquire a more informative frontal view of the subjects.

However, training active perception models differs significantly compared to traditional static perception approaches, since models must learn also the dynamics of the perception process in order to provide control feedback. For example, an active face recognition model should also learn how perception accuracy varies as the robot moves around a subject, as well as the direction in which a robot should move in order to improve the accuracy of face recognition. Therefore, it becomes clear that training active perception models introduces additional challenges, both with respect to acquiring the necessary data for train-

ing, as well as for extending the traditional (usually supervised) learning pipelines to support such setups.

The main aim of this paper is to introduce the main active perception approaches used for training DL-based active perception models for different applications. To this end, we will first present and discuss the different options for acquiring the necessary data used for training active perception models. Next, we will present different training approaches that extend traditional supervised learning methods for active perception or employ reinforcement learning methods to provide active perception feedback. Finally, we will discuss applications in various fields related to robotics, as well as discuss implications and practical issues.

The rest of this paper is structured as follows. First, in Section 2 we present the different methodologies for acquiring data for active perception, while in Section 3 we present different learning approaches that are used for training active perception DL models. Then, in Section 4, we provide an overview of applications for different perception applications. Finally, Section 5 concludes this paper.

## 2 Data for Active Perception

As discussed in Section 1, training active perception models requires a shift from traditional static perception methods, presenting a distinctive challenge. This distinction arises from the necessity for active perception models to not only grasp the static aspects of object recognition but also to encompass the dynamics inherent in the perception process, allowing them to generate control feedback. For instance, when considering an active face recognition model, the model should acquire knowledge concerning the optimal direction in which the robot should navigate to enhance the accuracy of face recognition. Unfortunately, a notable constraint emerges as a significant portion of the available datasets does not inherently facilitate the training of models for active perception tasks. Current literature can be roughly categorized into three distinct methodologies that can be used for getting data suitable for active perception: a) simulation-based training, b) multi-view dataset-based training, and c) on-demand (synthetic) data generation. An overview of the different approaches, along with benefits and drawbacks, is provided in Table 1.

Ideally, an active perception model would learn as it interacts with its environment. However, getting ground truth data in real-time is typically infeasible. Therefore, in most cases, active perception models are trained in an offline fashion. The first

Table 1: Comparing different approaches that can be used for acquiring data that can be used for training active perception models

Approach	Benefits	Drawbacks	Examples
Simulation-based training	flexible, any movement can be simulated	computationally-demanding, sim-to-real gap	(Ginargyros et al., 2023; Tzimas et al., 2020; Tosidis et al., 2022)
Multi-view dataset-based training	real data used, no sim-to-real gap	limited flexibility, limited number of control actions, missing data	(Passalis and Tefas, 2020; Georgiadis et al., 2023)
On-demand (synthetic) data generation through manipulation	less susceptible to sim-to-real gap, faster than simulation-based training	less accurate simulation of control actions, perception dynamics might not be accurately modeled	(Dimaridou et al., 2023; Passalis and Tefas, 2021; Kakaletsis and Nikolaidis, 2023; Manousis et al., 2023; Bozinis et al., 2021)

category of methods employs realistic simulation environments, e.g., as in (Tosidis et al., 2022; Ginargyros et al., 2023), in order to simulate the effect of various movements and allow the agent to learn how perception accuracy varies when performing different actions. This approach provides great flexibility since any action can be simulated and the effect of the movement of a robot can be easily obtained. However, such approaches are computationally demanding, since they rely on realistic simulation environments and graphics engines, such as Webots (Michel, 2004) and Unity (Haas, 2014), slowing down the training process. Furthermore, these approaches are also hindered by the so-called “sim-to-real” gap (Zhao et al., 2020), since the agents are trained using data generated by a simulator.

The second category of approaches, called “multi-view dataset-based training” in this paper, employs datasets that contain multiple views of the same scene. In this way, the effect of various movements can be quantified by fetching the view that would correspond to the result of the said movement. For example, in (Passalis and Tefas, 2020), the multiple views around a person are used to simulate the effect of an agent moving around, enabling training active perception models that learn how to maximize face recognition accuracy. Such approaches can overcome the issues of computational complexity and the “sim-to-real” gap. However, they are often too restrictive, since the datasets should already contain the images that can be used for every possible action an agent can perform. This often leads to huge datasets, as well as to agents that can be trained for a limited number of control actions. Furthermore, such approaches often have to handle missing data, since, in many cases, there are missing data in the corresponding multi-view datasets.

Then, methods that generate “on-demand” data

have also been proposed. Such approaches can try to simulate the effect of various movements starting from real data and then appropriately manipulating the data, e.g., simulating occlusions (Dimaridou et al., 2023). Another approach is to generate multiple views that can then be used either for deciding the best course of action or training the agent (Kakaletsis and Nikolaidis, 2023). These approaches fall in between simulation-based and multi-view dataset-based approaches since they employ real images that have been appropriately manipulated to simulate the effect of active perception. Therefore, even though they are less susceptible to the sim-to-real gap and they are typically faster, they often provide less accuracy in simulating the effect of active perception feedback, leading to models that might fail to capture all details of the dynamics of the active perception process.

### 3 Learning Methodologies for Active Perception

Training active perception models also departs from the typical supervised learning approach that is followed in many perception applications, such as face recognition (Wen et al., 2016), object detection (Redmon et al., 2016) and pose estimation (Zheng et al., 2023). Active perception models should not only analyze and understand their input but also provide some kind of control feedback, that can be then subsequently used for improving perception accuracy. Therefore, they tend to incorporate elements typically found in planning (Sun et al., 2021) and control (Tsounis et al., 2020) approaches used in robotics applications. The degree to which such elements are part of each model depends on the specific application requirements. In recent literature, two approaches are prevalent: a) deep reinforcement learn-

Table 2: Comparing different learning paradigms that can be used training active perception models

Approach	Benefits	Drawbacks	Examples
Deep Reinforcement Learning	directly optimizes the active perception model	slow convergence, low sample efficiency, (usually) requires simulation environments	(Bozinis et al., 2021; Tzimas et al., 2020; Tosidis et al., 2022)
Supervised Learning	can work with any kind of data, easier and faster to train	requires carefully design heuristics to construct ground truth data	(Passalis and Tefas, 2020; Ginargyros et al., 2023; Dimaridou et al., 2023; Manousis et al., 2023)

ing (DRL)-based training and b) supervised training through carefully designed ground truth. An overview of these two different approaches, along with benefits and drawbacks, is provided in Table 2.

DRL has achieved remarkable progress in recent years, providing beyond human performance in many cases (Mnih et al., 2013). Such approaches naturally fit active perception, since they enable models to learn how to provide control feedback to maximize perception accuracy through the interaction with an environment. Such approaches almost exclusively require the use of a simulation environment to be trained. Even though DRL methods enable discovering complex policies that can directly optimize the objective at hand, i.e., perception accuracy, they suffer from low sample efficiency, long training times, and unstable convergence (Buckman et al., 2018). On the other hand, the supervised method typically follows an “imitation” learning training paradigm (Hua et al., 2021), where the best actions to be performed are found through an extensive search in the action space. This is better understood with the following example. A DRL-agent training to perform active face recognition, e.g., (Tosidis et al., 2022), would learn using the reward signal from the environment, e.g., confidence in correctly recognizing a person. On the other hand, a supervised approach, such as (Passalis and Tefas, 2020), would first require simulating the effect of various movements/actions and then provide ground truth data on which action should the agent perform at each step. This also enables supervised approaches to work with any kind of data available, since the actions to be evaluated can be dictated by the capacity of the dataset to support the corresponding action. Therefore, even though supervised approaches can provide more stable and faster convergence and typically do not require a complex simulation environment, they rely on hand-crafted heuristic-based approaches to constructing the ground truth data.

## 4 Active Perception for Robotic Applications

Several recent active perception approaches have been proposed for a variety of different applications. In the rest of this Section, we briefly overview methods proposed for different applications, as well as discuss practical issues that often arise in robotics. Among the most prominent applications of active perception is face recognition. Indeed, early DL-based approaches extended embedding-based active perception methods into active ones by including an additional head that predicts the next best movement that a robot should perform in order to increase face recognition confidence (Passalis and Tefas, 2020). This approach assumes that the robot moves on a predefined trajectory around the target in order to be compatible with the multi-view dataset employed. Then, the model is trained to both maximize face recognition confidence, following a contrastive learning objective, as well as to regress the direction of movement leading to the best face recognition accuracy. Note that this direction is calculated by leveraging the multiple views available in the dataset and then selecting the one that maximizes the confidence for the next active perception step. The experimental evaluation demonstrated the effectiveness of this approach over static perception for a variety of different active perception steps. However, this approach used a dataset with a small number of individuals and a relatively small number of possible control movements. Later methods, such as (Dimaridou et al., 2023), build upon this approach by a) simulating the effect of various occlusions on large-scale face recognition datasets, and b) regressing both the direction and distance the robot should move.

A simple DRL approach for training a DRL agent to perform drone control in order to acquire frontal views that can be used for face recognition was initially proposed in (Tzimas et al., 2020), highlighting the potential of DRL methods for active perception tasks. A more sophisticated approach was also recently proposed building upon DRL in (Tosidis et al.,

2022). This approach leverages a realistic simulation environment, built using Webots (Michel, 2004), and directly trains a DRL agent to perform control in a drone that flies around humans in order to maximize face recognition confidence. The experimental evaluation demonstrated that the trained agent was able to perform control in a variety of different situations. However, the sim-to-real gap remains with DRL approaches, which can be a limiting factor in directly applying such approaches in real applications.

A supervised approach was also proposed for object detection in (Ginargyros et al., 2023), where a rich dataset for potential movements was built using a simulation environment. This approach enabled the models to learn the object detection confidence manifold for different types of objects, e.g., cars and humans, while taking into account possible occlusions, allowing them to perform control tailored to the unique characteristics of different cases. To this end, a separate navigation proposal network was trained according to the confidence manifold of each object, enabling the model to learn to propose trajectories that will maximize object detection confidence. At the same time, this paper also revealed limitations that are often intrinsic to the current state-of-the-art object detection models, since it provided a structured approach for revealing the confidence manifold of object detectors. A dataset that can support active vision for object detection was also proposed in (Ammirato et al., 2017), and a DRL agent was also trained and evaluated. This paper demonstrated that it is possible, given the appropriate dataset and annotations, to directly train DRL agents to perform control for active vision tasks.

Another line of research focuses on performing *virtual control*, i.e., not physically altering the position of a robot or the parameters of a physical sensor, but rather selectively analyzing specific parts of the input in order to improve perception accuracy, while reducing the computational load. Such approaches can be especially useful in cases where high-resolution input images must be analyzed, while the object of interest lies only in a small area within the input. An especially promising approach was presented in (Manousis et al., 2023), where the heat map extracted from a low-resolution version of a high-resolution image was used to drive the perception process. To this end, the proposed method first identified a region of interest in the original image by looking for potential activations (i.e., parts where the DL model detects something, but not necessarily with high confidence), in a low-resolution version of the input, and then performed targeted cropping into the high-resolution image in order to select

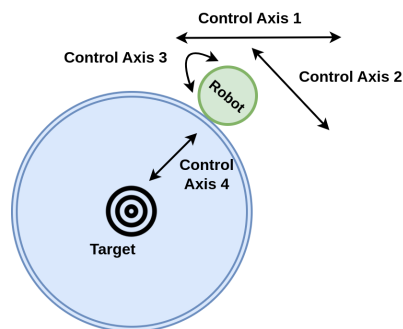


Figure 1: Active perception outputs can be represented in a homogeneous way using an application agnostic control specification defined by OpenDR (Passalis et al., 2022)

the area that needs to be analyzed. The experimental evaluation demonstrated that significant accuracy and speed improvements can be acquired using the proposed method. However, a limitation of such approaches is that as the size of the region of interest grows, the performance benefit obtained using active perception is becoming smaller. It is worth noting that such approaches can be also easily adjusted to perform control of the parameters of a camera, e.g., physical zoom, in order to acquire signals that are easier to analyze.

Another significant issue when implementing active perception models is the existence of a common way of expressing the outcomes of active perception. This is especially important, since in many robotics systems, different models might be employed for different perception tasks. Having to handle a completely different form of output for different models significantly complicates the development process. OpenDR toolkit (Passalis et al., 2022) has provided a common application agnostic control specification for standardizing such active perception outputs. This specification ensures that algorithms designed for active perception can effectively process the result. To this end, four control axes have been identified, as shown in Fig. 1. For all axes, it is assumed that the robot moves in a sphere and a real value from  $-1$  to  $1$  is provided for the movement on each axis. Using this way of expressing the output of active perception approaches holds the credentials for simplifying the development of active perception-enable robotics systems, by enabling the the efficient re-use of components related to handling and executing the feedback provided by active perception algorithms.

## 5 Conclusions

DL has revolutionized computer vision and robotics by enabling remarkable advancements in perception

tasks. However, as discussed in this paper, a significant limitation persists in many existing DL-based systems: the static inference paradigm. Most DL models operate on fixed, static inputs, neglecting the potential benefits of active perception – a process that mimics how humans and certain animals interact with their environment to better understand it. Active perception offers advantages in terms of accuracy and efficiency, making it a crucial area of exploration for enhancing robotic perception. While the incorporation of deep learning and active perception in robotics presents numerous opportunities, it also poses several challenges. Training often necessitates interactive simulation environments and more advanced approaches like deep reinforcement learning. Moreover, deployment pipelines need to be adapted to enable control within perception algorithms. These challenges highlight the importance of ongoing research and development in this field.

## ACKNOWLEDGMENTS

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

- Aloimonos, Y. (2013). *Active perception*. Psychology Press.
- Ammirato, P., Poirson, P., Park, E., Kořecká, J., and Berg, A. C. (2017). A dataset for developing and benchmarking active vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1378–1385.
- Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495.
- Bajcsy, R., Aloimonos, Y., and Tsotsos, J. K. (2018). Revisiting active perception. *Autonomous Robots*, 42(2):177–196.
- Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.
- Bozinis, T., Passalis, N., and Tefas, A. (2021). Improving visual question answering using active perception on static images. In *Proceedings of the International Conference on Pattern Recognition*, pages 879–884.
- Buckman, J., Hafner, D., Tucker, G., Brevdo, E., and Lee, H. (2018). Sample-efficient reinforcement learning with stochastic ensemble value expansion. *Proceedings of the Advances in Neural Information Processing Systems*, 31.
- Dimaridou, V., Passalis, N., and Tefas, A. (2023). Deep active robotic perception for improving face recognition under occlusions. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (accepted)*, page 1.
- Georgiadis, C., Passalis, N., and Nikolaidis, N. (2023). Activeface: A synthetic active perception dataset for face recognition. In *Proceedings of the International Workshop on Multimedia Signal Processing (accepted)*, page 1.
- Ginargyros, S., Passalis, N., and Tefas, A. (2023). Deep active perception for object detection using navigation proposals. In *Proceedings of the IEEE Symposium Series on Computational Intelligence (accepted)*, page 1.
- Haas, J. K. (2014). A history of the unity game engine.
- Han, X., Liu, H., Sun, F., and Zhang, X. (2019). Active object detection with multistep action prediction using deep q-network. *IEEE Transactions on Industrial Informatics*, 15(6):3723–3731.
- Heffner, R. S. and Heffner, H. E. (1992). Evolution of sound localization in mammals. In *The evolutionary biology of hearing*, pages 691–715.
- Hua, J., Zeng, L., Li, G., and Ju, Z. (2021). Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. *Sensors*, 21(4):1278.
- Kakaletsis, E. and Nikolaidis, N. (2023). Using synthesized facial views for active face recognition. *Machine Vision and Applications*, 34(4):62.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- Li, J.-h. (2018). Cyber security meets artificial intelligence: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(12):1462–1474.
- Liu, Q., Liu, Z., Xu, W., Tang, Q., Zhou, Z., and Pham, D. T. (2019). Human-robot collaboration in disassembly for sustainable manufacturing. *International Journal of Production Research*, 57(12):4027–4044.
- Loukas, G., Vuong, T., Heartfield, R., Sakellari, G., Yoon, Y., and Gan, D. (2017). Cloud-based cyber-physical intrusion detection for vehicles using deep learning. *IEEE Access*, 6:3491–3508.
- Manousis, T., Passalis, N., and Tefas, A. (2023). Enabling high-resolution pose estimation in real time using active perception. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2425–2429.
- Michel, O. (2004). Cyberbotics ltd. webots™: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1(1):5.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Passalis, N., Pedrazzi, S., Babuska, R., Burgard, W., Dias, D., Ferro, F., Gabbouj, M., Green, O., Iosifidis, A.,

- Kayacan, E., et al. (2022). OpenDR: An open toolkit for enabling high performance, low footprint deep learning for robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 12479–12484.
- Passalis, N. and Tefas, A. (2020). Leveraging active perception for improving embedding-based deep face recognition. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, pages 1–6.
- Passalis, N. and Tefas, A. (2021). Pseudo-active vision for improving deep visual perception through neural sensory refinement. In *Proceedings of the IEEE International Conference on Image Processing*, pages 2763–2767.
- Passalis, N., Tefas, A., and Pitas, I. (2018). Efficient camera control using 2d visual information for unmanned aerial vehicle-based cinematography. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 1–5.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788.
- Shen, M. and How, J. P. (2019). Active perception in adversarial scenarios using maximum entropy deep reinforcement learning. In *Proceedings of the International Conference on Robotics and Automation*, pages 3384–3390. IEEE.
- Sun, H., Zhang, W., Yu, R., and Zhang, Y. (2021). Motion planning for mobile robots—focusing on deep reinforcement learning: A systematic review. *IEEE Access*, 9:69061–69081.
- Tosidis, P., Passalis, N., and Tefas, A. (2022). Active vision control policies for face recognition using deep reinforcement learning. In *Proceedings of the 30th European Signal Processing Conference*, pages 1087–1091.
- Tsounis, V., Alge, M., Lee, J., Farshidian, F., and Hutter, M. (2020). Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3699–3706.
- Tzimas, A., Passalis, N., and Tefas, A. (2020). Leveraging deep reinforcement learning for active shooting under open-world setting. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1–6.
- Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *Proceedings of the European Conference on Computer Vision*, pages 499–515.
- Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pages 737–744.
- Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S., Shen, J., Kehtarnavaz, N., and Shah, M. (2023). Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37.