Robots Autonomously Detecting People: A Multimodal Deep Contrastive Learning Method Robust to Intraclass Variations

Angus Fung, Beno Benhabib, and Goldie Nejat, *Member, IEEE*

***Abstract*—Robotic detection of people in crowded and/or cluttered human-centered environments including hospitals, long-term care, stores and airports is challenging as people can become occluded by other people or objects, and deform due to clothing or pose variations. There can also be loss of discriminative visual features due to poor lighting. In this paper, we present a novel multimodal person detection architecture to address the mobile robot problem of person detection under intraclass variations. We present a two-stage training approach using 1) a unique pretraining method we define as Temporal Invariant Multimodal Contrastive Learning (TimCLR), and 2) a Multimodal YOLOv4 (MYOLOv4) detector for finetuning. TimCLR learns person representations that are invariant under intraclass variations through unsupervised learning. Our approach is unique in that it generates image pairs from natural variations within multimodal image sequences, in addition to synthetic data augmentation, and contrasts crossmodal features to transfer invariances between different modalities. These pretrained features are used by the MYOLOv4 detector for finetuning and person detection from RGB-D images. Extensive experiments validate the performance of our DL architecture in both human-centered crowded and cluttered environments. Results show that our method outperforms existing unimodal and multimodal person detection approaches in detection accuracy when considering body occlusions and pose deformations in different lighting.**

***Index Terms*— Robotic Person Detection, RGB-D Features, Deep Contrastive Learning, Intraclass Variations, Cluttered/Crowded Environments**

I. Introduction

Robots need to be able to autonomously detect multiple people in various human-centered environments in order to engage in effective human-robot interactions. Namely, person detection applications range from long-term care, retirement and private home settings, where interactive robots search for users to provide reminders and assistance with activities of daily living [1]–[3], product searches in big-box retail stores [4], [5], and direction guidance in airports [6] and hospitals [7], [8]. Furthermore, in disaster scenes, rescue robots can search to find trapped victims under rubble [9].

In general, human-centered environments can be crowded and cluttered with multiple dynamic people and objects, resulting in person and body part occlusions [9]. Furthermore, as people move or interact in the environment, they can undergo deformation due to both variations in clothing and body articulation [10]. These environments can also have variable illumination due to both natural and artificial lighting sources [11], which can result in appearance changes despite intrinsic properties of the person (*e.g.*, shape) not changing [11]. These variations can be defined as intraclass variations.

Classical learning approaches have been used to detect people by a mobile robot [12]–[15]. These approaches first extract a set of expert handcrafted features, such as HOG features [12]–[15], from upright people [16], and then train a supervised learning model to classify people. However, HOG features can only be used in applications where people maintain a fixed orientation to the robot [16]. Yet, people exhibit a variety of poses including sitting, lying down, etc.

Deep learning (DL) approaches address the limitations of classical learning methods by autonomously learning feature extraction, without having human experts extract handcrafted features. Thus, they can generalize to people in different poses and postures within varying environments [17]. DL methods use convolutional neural networks (CNN) to learn person feature representations in a hierarchical structure [18], [19].

To-date, robots use DL detection methods built on off-the-shelf *object detectors* to detect people in both indoor/outdoor environments. The methods include: 1) You Only Look Once (YOLOv3) [20] used in [21], 2) Single Shot MultiBox Detector (SSD) [22] used in [23], 3) RetinaNet [24] used in [9], and 4) Faster R-CNN (FRCNN) [25] used in [7], [8]. These methods use a *unimodal* CNN to extract features from RGB images taken from a single camera on a robot to detect people. However, they have difficulty in cluttered environments with varying illumination, as visual features necessary for discriminating people from their backgrounds become less prominent. Furthermore, these DL methods use *off-the-shelf pretrained RGB weights* to initialize the CNNs. This limits their application to unimodal CNNs as off-the-shelf *multimodal* pretrained weights do not exist [26], [27].

To address intraclass variations in DL person detection methods, data augmentation has been used to increase the training data [9]. However, this cannot capture the majority of variations due to low probability of occurrence [28]. Unsupervised contrastive learning (CL) can be used to address intraclass variations by pretraining a multimodal model to learn invariant features from scratch and from unlabeled data [29]. CL approaches have been used to learn representation invariances by contrasting between images of different viewpoints of the same static objects in constant lighting conditions [30]. Thus, they have the potential to learn representations which are invariant to intraclass variations. Unlabelled data is relatively inexpensive as a robot can be deployed to autonomously collect multimodal data directly from human-centered environments without the need for manual labelling. Recently, CL methods have been used in a

This research is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), HeRo NSERC CREATE program, AGE-WELL Inc., and the Canada Research Chairs (CRC) program. Authors are with the Autonomous Systems and Biomechatronics Lab in the Department of Mechanical and Industrial Engineering at the University of Toronto, 5 King's College Road, Toronto, ON, M5S 3G8 Canada. (Email: angus.fung@mail.utoronto.ca, {benhabib, nejat}@mie.utoronto.ca).

handful of robotic applications [30]–[32]. However, to-date, CL has not been applied to robotic person detection.

In this paper, we present a novel multimodal DL person detection architecture for mobile robots which uses a two-stage training approach consisting of: 1) CL for pretraining, and 2) Multimodal YOLOv4 (*MYOLOv4*) for finetuning. For prediction, the trained *MYOLOv4* detector is used for autonomous people detection from RGB-D data. We have developed a new pretraining method, Temporal Invariant Multimodal Contrastive Learning (*TimCLR*), to pretrain a multimodal CNN model from unlabelled RGB-D data in human-centered environments. *TimCLR* incorporates intraclass variations by generating multimodal image pairs from sampling video frames within a short temporal interval, and contrasting person representations within and between modalities, in addition to augmented data. This captures the natural variations in appearance (lighting, occlusions and pose deformations) as people move in their environments. Our overall approach is unique in that it uses CL to combine natural variations in the environment obtained from multimodal features, as well as incorporates a fusion backbone to contrast multimodal features. Thus, our approach does not require pretrained RGB weights or expert handcrafted mappings. We present extensive experiments to verify that our DL architecture outperforms existing DL detection methods in both human-centered crowded (with dynamic people) and cluttered (with objects) environments.

II. Related Works

In this section, we discuss the existing DL methods developed for robots to detect multiple dynamic people in human-centered environments and we further introduce CL methods and their current robotic applications.

A. *Person Detection by Robots*

Existing person detection methods for robotic applications consist of: 1) single-stage detectors including YOLO [9], [21], [33], [34], SSD [9], [23], and RetinaNet [9]; and 2) two-stage detectors including Faster R-CNN [7]–[9]. In single-stage detectors, every position in an image is considered as region proposals for containing potential people [9]. A CNN is used to predict the bounding boxes of people [9]. For example, in [21], [33], [34], YOLO detectors used RGB images to detect and follow a person by a robot in indoor environments to provide assistance. In [23], an SSD detector detected multiple people in RGB images for person following by a robot in indoor environments. All detectors were initialized using off-the-shelf RGB weights which were pretrained on ImageNet with images of general objects [19].

In two-stage detectors, a first stage region proposal network (RPN) is used to generate region proposals, followed by a second stage, where these proposals are classified and regressed [9]. In both [7], [8], a FRCNN detector was used to find people with mobility aids in populated environments (*e.g.*, hospitals, airports). People were detected from RGB or depth images using RGB and depth networks which were pretrained on ImageNet. As off-the-shelf pretrained RGB weights require 3 input channels, the depth images were preprocessed using ColorJet which distributes depth data based on distance. Both networks were trained on the Mobility Aids dataset [7] consisting of expert annotated RGB-D images of people with different aids in a hospital obtained from a mobile robot.

In our prior work [9], single/two-stage detectors were compared for person/body part detection in cluttered urban search and rescue (USAR) settings. The detectors were used to identify arm, foot, hand, head, leg, torso body parts from RGB-D images. All networks were trained on RGB, depth, and RGB-D data, using off-the-shelf RGB weights pretrained on ImageNet. The RGB-D network was trained by compressing the RGB-D image from 4 to 3 channels that consist of the grayscale RGB image, and the upper and lower 8 bits of the 16-bit depth image. The networks were trained on a RGB-D dataset consisting of fully visible/partially occluded body parts collected by a robot in an USAR-like environment.

B. *Contrastive Learning Methods*

Contrastive learning methods learn representations through similarities/dissimilarities between pairs of images without the need for expert labels by maximizing the agreement between two views of the same scene [35] through optimizing the contrastive loss [36]. CL is commonly used to pretrain general representations which are transferred via finetuning to tasks such as object detection [29] or robot manipulations [30]. CL methods can be image-based [29], [32], [35] or video-based [30], [31] input type. Image-based methods generate views by applying random data augmentation to the same image [36]. Video-based methods generate diverse views using natural transformations from frames in a video sequence [30] or from various video sequences [31].

CL methods have been used in a handful of robotic applications [30]–[32]. In [30], a video-based CL method learned robot manipulation behaviors of pouring liquids by imitating human interactions from smartphone videos. A network was pretrained to learn viewpoint-invariant features from RGB images by maximizing representations between different viewpoints of the same scene from different video sequences. Reinforcement learning (RL) was used to learn policies from the pretrained network. Experiments were conducted in an indoor room with constant lighting.

In [31], a video-based CL method trained a network for object discovery by a mobile robot to learn representations of unseen toys and appliances. This included recognizing and matching an unknown object seen previously to learn viewpoint invariant features (*e.g.*, texture, shape). The method extracted features from RGB image pairs of video sequences captured by a camera on a robot, and maximized the representation between each object and its neighbor**.** Experiments were conducted in a house and office with constant lighting to match static objects to reference objects.

In [32], an image-based CL method was used to train a RL architecture for robot navigation in a smoked filled environment. CL was applied to learn smoke-invariant representations from LIDAR and radar data (robust to smoke). Pairs were generated by taking one view from radar, and the other from LIDAR. CL was optimized with the RL policy. Experiments were conducted in a cardboard maze.

The image-based method MoCo v3 [29] has been applied to object detection, using synthetic data augmentation to learn representations during ImageNet pretraining. These features are transferred to a detection task, for example, on the COCO dataset [37]. MoCo v3 generates image pairs by applying data augmentations twice on the same RGB image [29]. The images were passed into the encoders to generate pairs of representations, used by the contrastive loss to maximize representation similarity [29]. MoCo v3 has higher detection accuracy than other image-based methods, including supervised pretraining methods [29], [35]. Thus, it has the potential to be applied to the robotic problem of person detection. However, it only considers data augmentation to generate views and does not incorporate temporal variations.

C. Summary of Limitations

Existing robotic person detection methods have used unimodal CNNs with off-the-shelf pretrained RGB weights trained on ImageNet, except in [9]. As these approaches only use RGB information, they have difficulty detecting people under poor lighting due to underexposure [38]. While [9] incorporates RGB-D data, it uses pretrained RGB weights and handcrafted heuristics to compress the RGB-D data from 4 channels to 3, resulting in information loss. To avoid this, off-the-shelf RGB-D weights are required, which do not currently exist. Although an alternative is to train from scratch using supervised DL methods, RGB-D robotic datasets with hand-labelled detection annotations are small, ranging from 572 to 17,000 images [8], [9], [39]. These are significantly smaller than ImageNet with one million RGB images used during pretraining, or MS COCO with 123,000 RGB images used to train from scratch RGB person detectors, which can result in overfitting [40]. Instead of using off-the-shelf weights, handcrafted heuristics, or large amounts of annotated data, CL can be used to pretrain RGB-D models from unlabeled data.

Image-based CL methods do not consider intraclass variations. Video-based CL methods, used in a handful of robotic applications, consider temporal variations within RGB images, but have been applied to static objects in constant lighting environments [30], [31]. Moreover, using temporal variations without data augmentation degrades representations quality due to feature suppression [41].

To address the above limitations, we present a modified MoCo v3 CL method to generate image pairs by uniquely combining both synthetic data augmentations and temporal variations from multimodal image frames within a short temporal interval. In additional to contrasting features between modalities to transfer invariances learned from one modality to the other, we use a fusion backbone to contrast multimodal features. We incorporate the above as our novel first stage *TimCLR* pretraining method, whose weights are then transferred to the second stage *MYOLOv4* detector for finetuning and detecting people in RGB-D images.

III. Person Detection Methodology

We propose a person detection architecture to detect multiple dynamic people or body parts from RGB-D images in human-centered environments. The proposed architecture, Fig. 1, consists of two training stages: 1) a *TimCLR* stage, using CL for unsupervised pretraining to learn RGB-D person and body part representations which are robust to intraclass variations, and 2) a *MYOLOv4* stage for supervised finetuning via the pretrained *TimCLR* model. For prediction, RGB-D images are passed into the trained *MYOLOv4* detector to detect multiple people/body parts. We selected the state-of-the-art YOLOv4 as it incorporates Path Aggregation Network (PAN) allowing for the detection of people/body parts at multiple scales [42]. This has the potential to improve detection under occlusions where only a portion of the person or body part is visible.

The *TimCLR* stage uses an unlabeled sequence of RGB-D images as inputs. RGB-D image pairs are generated by the *Sampling* module by sampling frames within a short temporal interval. These pairs capture natural variations in the environment by considering similar scenes under different conditions. In addition to natural variations, the *Augmentation* module applies synthetic data augmentation to each RGB-D image in the image pair. The *Multimodal Feature Extraction & Fusion* (*MFEF*) module passes RGB-D images into the encoders to extract RGB, depth, and RGB-D person representations. The *Crossmodal (CM)* module maximizes the contrastive loss of those representations generated by the unimodal and fusion backbones. *TimCLR* weights updated through backpropagation are transferred to the next stage.

The *MYOLOv4* stage uses labelled RGB-D images and the pretrained weights from *TimCLR* as inputs for finetuning. *MYOLOv4* adopts the YOLOv4 structure of a PAN and *YOLOv3 head* [42], Fig. 1. The RGB backbone is replaced with a subset of the *TimCLR* backbone layers. The weights from *TimCLR* are used to initialize *MYOLOv4* for training. The *MFEF* module extracts and fuses RGB and depth person features to detect multiple people (or body parts) as they move in a human-centered environment. *MYOLOv4* outputs a bounding box for each detected person within the image.

A. Temporal Invariant Multimodal Contrastive Learning

TimCLR extends MoCo v3, to provide different views by combining both synthetic data augmentation and natural variations in the environment obtained from a sequence of RGB and depth images. By combining these, we minimize the degradation of learned representations due to feature suppression [41]. The 4 main modules of the *TimCLR* stage are discussed below.

1) Sampling

The *Sampling* module samples pairs of RGB and depth images from a multimodal dataset $\boldsymbol{\mathcal{D}}$**,** consisting of sequences of unlabeled images containing people performing activities in a human-centered environment. These sequences capture people (and body parts) undergoing natural variations in occlusion, pose deformation, and lighting. Image pairs are sampled within a short temporal interval Δ_t. In practice, these pairs are sampled either from short video sequences of length Δ_t or from short segments of length Δ_t from longer duration full videos. Let $(\mathbf{x}_1^{RGB}, \mathbf{x}_1^{D})$ and $(\mathbf{x}_2^{RGB}, \mathbf{x}_2^{D})$ be RGB-D images sampled at times t_1 and t_2, an image pair is represented as:

$$\left((\mathbf{x}_1^{RGB}, \mathbf{x}_1^{D}), (\mathbf{x}_2^{RGB}, \mathbf{x}_2^{D})\right) \sim \boldsymbol{\mathcal{D}} \times \boldsymbol{\mathcal{D}}. \quad (1)$$

2) Augmentation

The *Augmentation* module applies a set of transformations to each sample pair of images from Eq. (1), which relate two multimodal views representing the same people under different conditions. Namely, the following MoCo v3 transformations are randomly applied [29]: cropping, color jittering, grayscaling, gaussian blurring, and horizontal flipping. Let $\mathbf{T_1}, \mathbf{T_2} \sim \boldsymbol{\mathcal{T}}$ be the composite of those transformations, and $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ be the transformed images:

$$(\hat{\mathbf{x}}_1^{RGB}, \hat{\mathbf{x}}_1^{D}) = \mathbf{T_1}(\mathbf{x}_1^{RGB}, \mathbf{x}_1^{D}), \tag{2}$$
$$(\hat{\mathbf{x}}_2^{RGB}, \hat{\mathbf{x}}_2^{D}) = \mathbf{T_2}(\mathbf{x}_2^{RGB}, \mathbf{x}_2^{D}). \tag{3}$$

The output RGB-D image pairs $((\hat{\mathbf{x}}_1^{RGB}, \hat{\mathbf{x}}_1^{D}), (\hat{\mathbf{x}}_2^{RGB}, \hat{\mathbf{x}}_2^{D}))$ represent two augmented views of people under different natural variations.

3) *Multimodal Feature Extraction & Fusion*

The *MFEF* module extracts and fuses features from the pairs of RGB-D transformed views to produce RGB, depth, and RGB-D person representations. It consists of an encoder $\mathbf{f}_q$, and momentum encoder $\mathbf{f}_k$ [29]. Each RGB-D image pair is passed into both networks to extract three representations:

$$\boldsymbol{q}_i^{RGBD}, \boldsymbol{q}_i^{RGB}, \boldsymbol{q}_i^{D} = \mathbf{f}_q(\hat{\mathbf{x}}_i^{RGB}, \hat{\mathbf{x}}_i^{D}; \boldsymbol{\theta}_q), \qquad i \in \{1,2\} \tag{4}$$
$$\boldsymbol{k}_i^{RGBD}, \boldsymbol{k}_i^{RGB}, \boldsymbol{k}_i^{D} = \mathbf{f}_k(\hat{\mathbf{x}}_i^{RGB}, \hat{\mathbf{x}}_i^{D}; \boldsymbol{\theta}_k), \qquad i \in \{1,2\} \tag{5}$$

where $\boldsymbol{\theta_q}$ and $\boldsymbol{\theta_k}$ are the weights of the network, and $\boldsymbol{q}_i^{RGBD}, \boldsymbol{q}_i^{RGB}, \boldsymbol{q}_i^{D}$ and $\boldsymbol{k}_i^{RGBD}, \boldsymbol{k}_i^{RGB}, \boldsymbol{k}_i^{D}$ are the feature representations of view i of each modality for the encoder, and momentum encoder, respectively [29]. The encoder weights $\boldsymbol{\theta_q}$ are updated by back-propagation [29]. The momentum weights $\boldsymbol{\theta_k}$ are updated by a weighted average of $\boldsymbol{\theta_q}$ and $\boldsymbol{\theta_k}$ [29], where m is the momentum coefficient:

$$\boldsymbol{\theta_k} \leftarrow m\,\boldsymbol{\theta_k} + (1-m)\,\boldsymbol{\theta_q}. \tag{6}$$

Each encoder consists of separate RGB and depth backbones, fusion backbones, and multilayer perceptrons (MLPs), with weights $\boldsymbol{\theta_l^{RGB}}, \boldsymbol{\theta_l^{D}}, \boldsymbol{\theta_l^{RGBD}}, \boldsymbol{\theta_l^{MLP}} \in \boldsymbol{\theta_l}, \boldsymbol{l} \in \{\boldsymbol{q}, \boldsymbol{k}\}$, respectively. Each backbone uses a modified ResNet-18 model. The standard RGB ResNet-18 consists of 5 convolution blocks (C1-C5), fully connected (FC), and a SoftMax layer [43]. Our RGB and depth backbones extract RGB and depth features each consisting of C1-C3 blocks, Fig. 1. A fusion backbone is used to concatenate the feature maps of each of the C3 blocks, followed by a 1x1 convolution layer, and C4-C5, Fig. 1. MLPs, consisting of two FC layers, are added to the output of the fusion backbone to extract RGB-D representations. Additional MLPs are added to the output of the RGB and depth backbone layers to extract unimodal representations, Fig. 1. The output representations from the encoders $(\boldsymbol{q}_i^{RGBD}, \boldsymbol{q}_i^{RGB}, \boldsymbol{q}_i^{D})$, and $(\boldsymbol{k}_i^{RGBD}, \boldsymbol{k}_i^{RGB}, \boldsymbol{k}_i^{D})$, $i \in \{1,2\}$ are passed to the *Crossmodal* module.

4) *Crossmodal*

The *Crossmodal* module computes representation similarity scores using the contrastive loss, $\mathcal{L}_{\mathrm{CL}}$, based on InfoNCE [29]:

$$\mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}, \boldsymbol{k}) = \mathbb{E}_Q\left[\log\frac{\exp(\boldsymbol{q_i}\cdot\boldsymbol{k}^+/\tau)}{\exp(\boldsymbol{q_i}\cdot\boldsymbol{k}_i^+/\tau) + \sum_{k^-}\exp(\boldsymbol{q_i}\cdot\boldsymbol{k}_i^-/\tau)}\right], \tag{7}$$

where $Q = \{\boldsymbol{q_1}, \ldots, \boldsymbol{q_N}\}$ is the set of representations from the mini-batch; $\{\boldsymbol{k}_1^+, \ldots, \boldsymbol{k}_N^+\}$ and $\{\boldsymbol{k}_1^-, \ldots, \boldsymbol{k}_N^-\}$ are the set of representations corresponding to positive and negative image pairs; and τ is the temperature [29]. The RGB-D contrastive loss to measure similarity between representations is:

$$\mathcal{L}_{RGBD} = \mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}_1^{RGBD}, \boldsymbol{k}_2^{RGBD}) + \mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}_2^{RGBD}, \boldsymbol{k}_1^{RGBD}). \tag{8}$$

The crossmodal contrastive losses to measure similarity between unimodal RGB and depth representations are:

$$\mathcal{L}_{RGB,D} = \mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}_1^{RGB}, \boldsymbol{k}_2^{D}) + \mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}_2^{RGB}, \boldsymbol{k}_1^{D}), \tag{9}$$
$$\mathcal{L}_{D,RGB} = \mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}_1^{D}, \boldsymbol{k}_2^{RGB}) + \mathcal{L}_{\mathrm{CL}}\,(\boldsymbol{q}_2^{D}, \boldsymbol{k}_1^{RGB}). \tag{10}$$

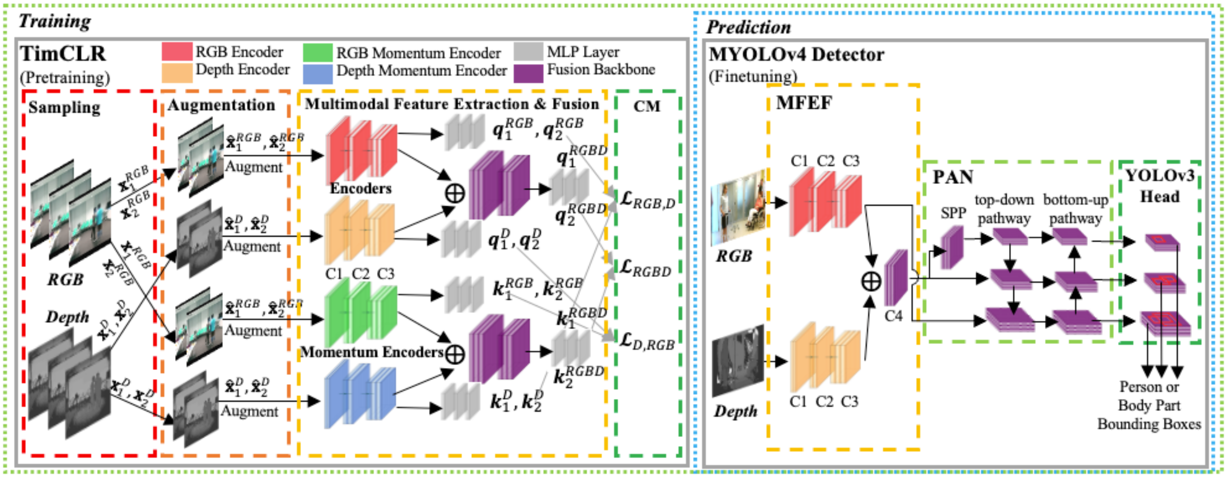
The full contrastive loss which is the combination of all the aforementioned losses is defined as:

$$\mathcal{L}_{\mathrm{MCL}} = \lambda_{RGBD}\mathcal{L}_{RGBD} + \lambda_{RGB,D}\mathcal{L}_{RGB,D} + \lambda_{D,RGB}\mathcal{L}_{D,RGB}, \tag{11}$$

where $\lambda_{RGBD}, \lambda_{RGB,D}$, and $\lambda_{D,RGB}$ are the weighting factors. The encoder weights $\boldsymbol{\theta_q}$ are passed to the *MYOLOv4* stage.

B. *Multimodal YOLOv4*

MYOLOv4, Fig. 1, adopts its structure from YOLOv4 [42]. It consists of *MFEF*, *PAN*, and *YOLOv4 head* modules. *MFEF* extracts feature maps from RGB and depth images using separate RGB and depth backbones each consisting of C1-C3 blocks which are initialized by $\boldsymbol{\theta_q^{RGB}}$ and $\boldsymbol{\theta_q^{D}}$, the weights of the unimodal backbones in *TimCLR*. These backbones are fused by channel-wise concatenation with a 1x1 convolution layer, followed by a C4 block, which is initialized by $\boldsymbol{\theta_q^{RGBD}}$, the weights of the fusion backbone in *TimCLR*. The *PAN* module uses C3 and C4, and consists of a Spatial Pyramid Pooling (SPP) block [42], and top-down and bottom-up pathways to aggregate features at different scales. SPP applies max pooling with kernels sizes 1×1, 5×5, 9×9, and 13×13, and concatenates the outputs. The top-down pathway consists of convolution and upsampling layers to



Training
TimCLR
(Pretraining)
RGB Encoder
Depth Encoder
RGB Momentum Encoder
Depth Momentum Encoder
MLP Layer
Fusion Backbone
Sampling
Augmentation
Multimodal Feature Extraction & Fusion
CM
RGB
Depth
$\mathbf{x}_1^{RGB}$
$\mathbf{x}_2^{RGB}$
$\mathbf{x}_1^{RGB}$
$\mathbf{x}_2^{RGB}$
$\mathbf{x}_1^{D}, \mathbf{x}_2^{D}$
$\mathbf{x}_1^{D}, \mathbf{x}_2^{D}$
$\hat{\mathbf{x}}_1^{RGB}, \hat{\mathbf{x}}_2^{RGB}$
Augment
$\hat{\mathbf{x}}_1^{D}, \hat{\mathbf{x}}_2^{D}$
Augment
$\hat{\mathbf{x}}_1^{RGB}, \hat{\mathbf{x}}_2^{RGB}$
Augment
$\hat{\mathbf{x}}_1^{D}, \hat{\mathbf{x}}_2^{D}$
Augment
Encoders
C1 C2 C3
Momentum Encoders
$\oplus$
$\boldsymbol{q}_1^{RGB}, \boldsymbol{q}_2^{RGB}$
$\boldsymbol{q}_1^{RGBD}$
$\boldsymbol{q}_2^{RGBD}$
$\boldsymbol{q}_1^{D}, \boldsymbol{q}_2^{D}$
$\boldsymbol{k}_1^{RGB}, \boldsymbol{k}_2^{RGB}$
$\boldsymbol{k}_1^{RGBD}$
$\boldsymbol{k}_2^{RGBD}$
$\boldsymbol{k}_1^{D}, \boldsymbol{k}_2^{D}$
$\mathcal{L}_{RGB,D}$
$\mathcal{L}_{RGBD}$
$\mathcal{L}_{D,RGB}$
Prediction
MYOLOv4 Detector
(Finetuning)
MFEF
RGB
Depth
C1 C2 C3
C1 C2 C3
C4
PAN
SPP
top-down pathway
bottom-up pathway
YOLOv3 Head
Person or Body Part Bounding Boxes

Fig. 1: Proposed multimodal DL detection architecture with first stage *TimCLR* and second stage *MYOLOv4* detector.

produce feature maps at three scale resolutions (1/8, 1/16, 1/32) of the input image. SPP, C4, and C3 are concatenated after each upsampling layer, respectively. The bottom-up pathway consists of convolution layers to produce feature maps of the same scales, with lateral connections to concatenate the corresponding top-down feature map. The *YOLOv3 head* module uses these feature maps to predict people/body parts at each scale. *MYOLOv4* is first trained through supervised learning and used during prediction to output a bounding box for each detected person or body parts within an image.

IV. Experiments

Our proposed two-stage person detection architecture is investigated in two sets of experiments: 1) a comparison study with existing robot person detection methods to evaluate detection accuracy, and 2) an ablation study to validate the design choices of our architecture. Two environments with varying levels of person occlusion, illumination, and deformation were considered; one crowded with people and the other cluttered with objects, from which RGB-D images have been taken by a mobile robot. All experiments were conducted on a workstation with two RTX 3070 GPUs, an AMD Ryzen Threadripper 3960X, and 128GB of memory.

A. Datasets

For the pretraining stage *TimCLR*, the unlabelled **NTU RGB+D 120 Action Recognition (NTU) Dataset** [44] is used. The dataset consists of multiple people performing 120 different actions (*e.g.,* standing, eating, jumping) in indoor environments with 114,480 RGB-D video samples collected by a Kinect sensor. NTU naturally captures the intraclass variations that would be expected in a real-world human-centered environment. For the finetuning stage, the following datasets are used, as discussed in Section IV.C:

1) **MS COCO dataset** [37] which consists of 123,000 RGB images of general objects with 250,000 person instances in indoor/outdoor environments. The entire dataset was used for training to learn semantically rich RGB person features.

2) **MA dataset** [8] which consists of 17,000 annotated RGB-D images of multiple dynamic people undergoing frequent occlusions in a crowded real-world hospital environment, collected by a Kinect sensor on a robot. The dataset contains 5 classes of people with different aids, which we combined into 1 person class to detect people. The dataset was split into 65/35 for training/testing. The two test sets (TS): 1) TS1 (few people occlusions), and 2) TS2 (frequent people occlusions).

3) **USAR dataset** [9] which consists of 570 RGB-D images of human/mannequin body parts in a real-world cluttered environment. The images were obtained from a Kinect sensor on a Turtlebot 2 robot. The dataset contains 6 classes: arm, foot, hand, head, leg, and torso; and is split using an 80/20 rule, with 3 separate test datasets collected on different days: 1) TS1 (contains fully visible people), 2) TS2 (contains partial person or body part occlusions and deformations), and 3) TS3 (contains people under low lighting conditions).

B. Performance Metrics

The mean average precision (mAP) was chosen as the performance metric for detection accuracy. The Intersection over Union (IoU) measures the area overlap between the ground truth and predicted bounding box. AP_{50} is the mAP where predictions with an IoU > 0.5 are considered true positives. $AP_{0.5:0.95}$, or AP, is the averaged mAP over IoU = $\{0.5, 0.55, \ldots, 0.95\}$. Averaging across IoUs more accurately measures localization [37], and is the primary accuracy metric. We also measured the memory usage and inference speed in frames per second (FPS) on an Nvidia Jetson AGX Xavier platform using the TensorRT framework, respectively.

C. Training

The proposed person detection architecture is trained in two stages: 1) pretraining (*TimCLR*), and 2) finetuning (*MYOLOv4*). *TimCLR* was pretrained on a subset of the NTU dataset, consisting of 1 million RGB-D images, generated by randomly selecting from the 114,480 videos in the dataset. Image pairs were sampled at Δ_t= 50 frames. *TimCLR* used the default hyperparameters from MoCo v3, and trained for 100 epochs. The RGB branch of *MYOLOv4* was first finetuned using the pretrained *TimCLR* weights on the MS COCO dataset. For evaluation on the MA test sets, *MYOLOv4* was additionally finetuned on the MA RGB-D training set, representing a crowded environment. Similarly, for evaluation on the USAR test sets, *MYOLOv4* was finetuned on the RGB-D USAR training set, representing a cluttered environment. Finetuning used stochastic gradient descent with a learning rate (LR) of 0.01 for 26 epochs.

D. Comparison Methods

Our person detection method was compared against existing DL RGB person detection methods, including RGB: 1) YOLOv2-416 [45], 2) YOLOv3-416 [20], 3) SSD-300 [22], 4) RetinaNet-FPN [24], and 5) FRCNN [25]; 6) depth ColorJet (CJ) FRCNN [7], [8] which uses CJ to preprocess depth images into 3-channels; and 7) RGB-D Compression (C) FRCNN [9] which uses compression to preprocess RGB-D images. We additionally compared against more recent DL RGB methods: 8) YOLOv4 [42], and 9) EfficientDet-D0 [46]. We also designed the following RGB-D strong baselines: 10) CJ-MYOLOv4, 11) CJ Multimodal FRCNN (CJ-MFRCNN), and 12) CJ Multimodal EfficientDet (CJ-MEfficientDet); which all use CJ preprocessing on the depth images [8]. We also compare against RGB-D TimCLR + MFRCNN, which uses our *TimCLR* method with the two-stage detector FRCNN which has been found to be accurate in detecting people in both crowded and cluttered environments [47], [48]. By comparing with RGB-D TimCLR + MFRCNN, we investigate the ability of *TimCLR* to learn invariant person features regardless of the specific finetuning method used.

We pretrained the networks ourselves rather than using the published pretrained weights to ensure fairness in comparisons by using the same backbone implementation and training procedure. Namely, we pretrained on ImageNet, as these methods require pretraining using classification labels (with the exception TimCLR + MFRCNN). In contrast, *TimCLR* is unsupervised and does not require such labels. Thus, TimCLR + MFRCNN is trained using the same procedure in Section IV.C. All networks were pretrained using a ResNet-18 backbone for 100 epochs, except for EfficientDet and RGB-D CJ-EfficientDet which used a

EfficientNet-B0 backbone [46]. We used the smaller ResNet model with 18 layers to meet the fast inference requirements of robotic applications. For the RGB-D CJ-based methods, we used the same RGB weights for the RGB and depth encoders, following [8], as ImageNet does not have depth images for pretraining. The networks were finetuned using the procedure from Section IV.C, with the default LRs for these methods.

E. People Detection Comparison Results

The detection accuracy results for our proposed method and all comparison methods are presented in Table I. Our proposed *TimCLR + MYOLOv4* detector outperformed the other methods with respect to AP and AP$_{50}$ on all test sets (TS), including on 1) partial occlusions (MA/USAR TS2), 2) deformations (USAR TS2), and 3) varying illuminations (USAR TS3). The results show *TimCLR*'s ability to learn invariant person features. Namely, our method outperformed all the unimodal RGB networks. While the RGB approaches generally outperformed the depth-only CJ-FRCNN method, they performed worse under varying lighting (USAR TS3).

Worth noting is that our proposed method outperformed the strong multimodal baselines that we designed which all used the ImageNet pretrained weights, as is typical of existing methods, for finetuning instead of our *TimCLR* weights. This further highlights the direct performance benefits due to *TimCLR*. For example, *TimCLR + MYOLOv4* outperformed RGB-D CJ-MYOLOv4, the best performing RGB-D baseline method, with 12% and 19% improvements in AP under partial occlusions on the MA and USAR datasets, respectively. These results are statistically significant, Table II. In general, *TimCLR* was more effective than CJ at capturing depth features for differentiating people/body parts from background clutter (TS2) and under poor lighting (TS3). Compared to RGB-D C-FRCNN, our *TimCLR + MYOLOv4* had improvements of 13-44% and 1-30% on AP and AP$_{50}$, respectively, across all test sets. Thus, RGB-D C-FRCNN predicts many correct bounding boxes of people with a 50% overlap with the ground truth, but, it performs poorer when considering larger overlaps, which the AP metric captures. We postulate that it had difficulties in estimating the extent of a person/body part under partial occlusions, as visual features may be lost during compression. Thus, our proposed method can accurately localize multiple people/body parts with a greater overlap between the predictions and ground truth.

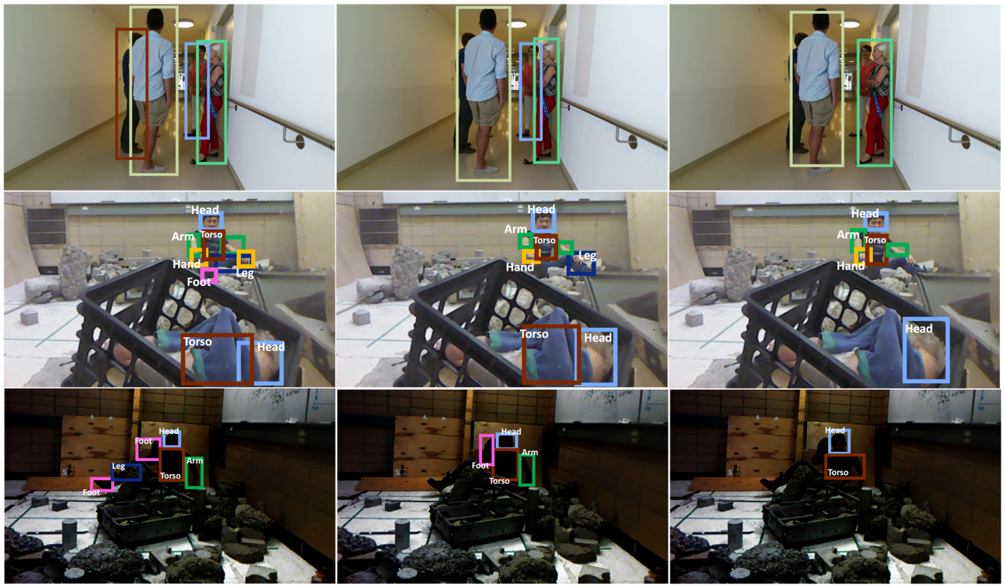
Furthermore, our method outperformed RGB-D TimCLR + MFRCNN with a statistically significant higher AP across all test sets with the exception of USAR TS1, despite the latter incorporating a two-stage detector. We postulate that this is a result of *PAN* in YOLOv4 which allows for the detection of body parts at multiple scales. Thus, our method is more effective at detecting body parts when discriminative features occupy smaller regions due to partial occlusions or poor lighting, as seen by larger improvements in AP on TS2-3 compared to TS1. Overall, *TimCLR + MYOLOv4* and TimCLR + MFRCNN which were finetuned using *TimCLR* features were more accurate and robust in cluttered/crowded environments. Thus, *TimCLR* can be directly applied to pretrain any state-of-the-art detector with high accuracy.

Fig. 2 presents example detections of our *TimCLR + MYOLOv4* method (Fig. 2(a)) compared to RGB-D CJ-MYOLOv4 ((Fig. 2(b)), the best RGB-D baseline, and RGB-D C-FRCNN, the best robotic person detection baseline (Fig. 2(c)). Fig. 2 shows these methods under partial occlusions (rows 1-3), pose deformation (rows 2-3), and poor lighting (row 3), on the MA (row 1) and USAR (rows 2-3) datasets. For example, row 2 shows a person partially occluded by rubble and a mannequin in the fetal position, and row 3 shows a person partially occluded by rubble in low lighting. Our method identified all four people in Fig. 2(a), especially the person on the left (missed by the other detectors); and the only method to detect the partially occluded right foot in the second scene and partially occluded right foot and leg in the third scene. While RGB-D CJ-MYOLOv4 detected a part of the left leg in Fig. 2(b), it did not capture the articulated portion of the leg as our method did. Under low lighting, only ours detected the right foot and articulated leg in the third scene.

A non-parametric Kruskal-Wallis test was performed on both datasets, showing a statistically significant difference in AP between the multimodal methods, Table II. A post-hoc Dunn test with Bonferroni correction showed our *TimCLR + MYOLOv4* had a statistically significant higher AP than the alternatives, Table II, with the exception of RGB-D TimCLR + MFRCNN on the USAR TS1 dataset as previously mentioned. In general, the YOLOv4-based detectors had the lowest memory usage, with the fastest inference rates of 41-42 FPS. Thus, our *TimCLR + MYOLOv4* is most suited for mobile robot detection of people in human-centered cluttered and crowded environments as it achieved the highest AP scores while maintaining real-time inference rates and lowest

TABLE I COMPARISON OF DETECTION ACCURACY OF OUR PROPOSED DETECTION METHOD VERSUS EXISTING DETECTION METHODS

Dataset	Mobility Aids (MA)				USAR						Memory (GB)	Inference (FPS)
	Test Set 1		Test Set 2 Occlusion		Test Set 1		Test Set 2 Occlusion+Deformation		Test Set 3 Illumination			
Method	AP	AP$_{50}$	AP	AP$_{50}$	AP	AP$_{50}$	AP	AP$_{50}$	AP	AP$_{50}$		
RGB-D *TimCLR + MYOLOv4*	60.10	94.30	49.20	75.30	22.30	45.80	21.10	49.60	20.20	44.40	1.4	41
RGB-D TimCLR + MFRCNN	58.81	94.05	47.23	75.41	22.02	45.80	17.95	40.29	18.60	37.07	2.1	4
RGB YOLOv4 [42]	56.20	90.10	44.10	70.80	15.10	35.20	14.40	31.60	14.60	33.60	1.4	42
RGB FRCNN [25]	55.91	92.79	45.44	75.20	14.62	37.12	15.75	35.76	10.00	29.83	2.1	4
RGB EfficientDet [46]	57.00	90.20	43.40	76.30	15.00	37.20	13.30	34.40	15.70	35.40	2.0	25
RGB YOLOv2 [45]	34.76	83.84	25.10	67.51	7.83	29.16	7.57	31.29	8.07	30.29	1.5	24
RGB YOLOv3 [20]	39.28	92.96	36.52	75.00	16.96	33.45	14.77	34.11	15.53	31.39	1.5	18
RGB SSD [22]	35.24	87.80	26.62	71.22	10.51	29.70	9.96	32.74	8.46	30.66	2.0	14
RGB RetinaNet [24]	51.72	93.69	42.60	75.48	13.55	34.77	14.21	32.95	13.56	27.70	1.5	19
Depth CJ-FRCNN [7.8]	42.22	84.24	34.42	70.37	10.72	25.47	9.59	24.63	17.23	34.51	2.1	19
RGB-D C-FRCNN [9]	48.44	93.66	39.54	74.54	15.52	38.38	15.21	38.23	17.86	34.95	2.1	4
RGB-D CJ-MFRCNN	57.90	92.72	44.70	73.15	20.03	43.89	16.66	38.59	17.84	35.40	2.1	4
RGB-D CJ-MYOLOv4	57.10	86.60	44.10	70.40	20.10	43.50	17.80	38.80	18.10	41.30	1.4	41
RGB-D CJ-MEfficientDet	57.40	90.30	44.60	75.50	19.70	44.10	16.60	37.50	17.70	39.80	2.0	24



(a) *TimCLR* + *MYOLOv4* **(our method)** (b) RGB-D CJ-MYOLOv4 (c) RGB-D C-FRCNN

Fig. 2: Multimodal detection results from: (a) *TimCLR* + *MYOLOv4* (**ours**), (b) RGB-D CJ-MYOLOv4, (c) RGB-D C-FRCNN; overlaid on RGB images.

memory usage on an embedded computing platform.

F. Ablation Study

We performed an ablation study to evaluate the design of our proposed detection architecture. We investigated *TimCLR* with respect to: 1) image pair generation, 2) fusion design, and 3) crossmodal loss. We evaluated each design choice based on the detection accuracy of *MYOLOv4* using the pretrained weights from *TimCLR*, presented in Table III.

In experiment 1, we investigated generating positive image pairs, using: 1) synthetic data augmentation, 2) natural variations, and 3) a combination. We noticed that pretraining using only data augmentation outperforms using only natural variations on TS1 as a result of feature suppression. However, they perform similarly under intraclass variations (TS2-3). During feature suppression, the network may ignore texture or shape features as other cues such as color distributions which are similar between frames captured within a short interval can be used to differentiate positive pairs from negative pairs. The combined approach avoids this as it applies color jittering, while still incorporating natural variations. Thus, it achieves higher detection accuracy under intraclass variations. In experiment 2, we investigated the fusion of RGB and depth features at C3, C4, or C5 blocks of the encoder. We found that C3 was the optimal layer to fuse. Finally, we investigated the crossmodal contrastive loss which contrasts representations between RGB and depth for evaluating the transfer of knowledge between modalities. We performed runs with and without this loss. The run with the loss performed similarly on TS1, but substantially better under intraclass variations (TS2-3). One possible reason is that contrasting of features between modalities encourages representations between them to be similar, thus transferring learned feature invariances from one modality to the other.

TABLE II KRUSKAL-WALLIS AND DUNN TEST

Kruskal-Wallis Test				
H statistic for AP between these methods (*TimCLR+MYOLOv4*, TimCLR+MFRCNN, RGB-D CJ-MFRCNN, RGB-D C-FRCNN, RGB-D CJ-MYOLOv4, RGB-D CJ-MEfficientDet), $p < 0.001$				
Mobility Aids		**USAR**		
Test 1 $n = 10795$	**Test 2** $n = 6238$	**Test 1** $n = 913$	**Test 2** $n = 905$	**Test 3** $n = 315$
545	612	731	12533	6017
Dunn Test with Bonferroni Correction				
Z statistic for AP between ours and the alternatives, $p < 0.001$				
Mobility Aids		**USAR**		
Test 1	**Test 2**	**Test 1**	**Test 2**	**Test 3**
2.88	5.08	0.72, *p*=1	45.55	6.88
6.25	13.20	3.09	52.21	17.78
10.60	27.75	46.76	57.47	18.15
8.75	15.09	3.46	45.85	8.17
8.09	14.98	12.15	55.47	15.34

V. CONCLUSIONS

In this paper, we present a novel multimodal person detection architecture for mobile robots to address the robotic problem of person detection under intraclass variations. We introduce a new pretraining method *TimCLR* which learns person features which are invariant to natural variations in the environment, such as person and body part occlusions, pose deformations, and varying lighting. Our *TimCLR* generates contrastive image pairs by sampling natural variations from multimodal image sequences within a short temporal interval, in addition to data augmentation. These invariant person features are used by the *MYOLOv4* detector for robust detection of people under intraclass variations. Extensive experiments verified that our *TimCLR* + *MYOLOv4* outperformed the existing detection methods in finding people in both crowded hospital and cluttered USAR environments. Our ablation study validated the design choices of *TimCLR*. Future work includes integrating our detection architecture within a mobile robot for real-time person search and detection in varying human-centered environments.

REFERENCES

[1] S. C. Mohamed, S. Rajaratnam, S. T. Hong, and G. Nejat, "Person Finding: An Autonomous Robot Search Method for Finding Multiple Dynamic Users in Human-Centered Environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 433–449, Jan. 2020.

[2] S. F. R. Alves, M. Shao, and G. Nejat, "A Socially Assistive Robot to Facilitate and Assess Exercise Goals," *IEEE Int. Conf. on Robot. and Automation Workshop on Mobile Robot Assistants for the Elderly*

TABLE III ABLATION STUDY

Dataset	Mobility Aids (MA)				USAR					
	Test Set 1		Test Set 2 Occlusion		Test Set 1		Test Set 2 Occlusion+Deformation		Test Set 3 Illumination	
Method	**AP**	**AP$_{50}$**	**AP**	**AP$_{50}$**	**AP**	**AP$_{50}$**	**AP**	**AP$_{50}$**	**AP**	**AP$_{50}$**
TimCLR, data augmentation only	58.10	90.40	45.30	66.40	19.80	43.90	16.90	45.10	18.00	40.00
TimCLR, natural variations only	48.70	90.30	40.40	64.90	18.10	41.80	17.40	42.80	16.20	37.50
TimCLR, combined	60.10	94.30	49.20	75.30	22.30	45.80	21.10	49.60	20.20	44.40
TimCLR, C3 Fusion	60.10	94.30	49.20	75.30	22.30	45.80	21.10	49.60	20.20	44.40
TimCLR, C4 Fusion	57.40	90.10	46.00	65.70	20.20	43.70	19.40	48.70	18.30	41.10
TimCLR, C5 Fusion	54.20	87.40	44.40	65.10	18.30	40.90	18.20	43.50	17.50	40.00
TimCLR, no crossmodal loss	57.80	90.30	47.70	71.10	18.40	41.70	17.50	42.70	16.20	37.80
TimCLR, with crossmodal loss	60.10	94.30	49.20	75.30	22.30	45.80	21.10	49.60	20.20	44.40

(MoRobAE), pp. 1-5, 2019.

[3] C. Thompson, S. Mohamed, W.-Y. G. Louie, J. C. He, J. Li, and G. Nejat, "The robot Tangy facilitating Trivia games: A team-based user-study with long-term care residents," in *IEEE Int. Symp. on Robot. and Intell. Sensors (IRIS)*, 2017, pp. 173–178.

[4] D. Dworakowski, C. Thompson, M. Pham-Hung, G. Nejat, "A Robot Architecture Using ContextSLAM to Find Products in Unknown Crowded Retail Environments," *Robot.*, vol. 10, no. 4, 2021.

[5] T. Wengefeld, S. Muller, B. Lewandowski, and H.-M. Gross, "A Multi Modal People Tracker for Real Time Human Robot Interaction," *IEEE Int. Conf. on Robot and Human Interactive Commun.*, 2019, pp. 1–8.

[6] R. Triebel *et al.*, "SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports," in *Field and Service Robotics: Results of the 10th Int. Conf.*, 2016, pp. 607-622.

[7] A. Vasquez, M. Kollmitz, A. Eitel, and W. Burgard, "Deep Detection of People and their Mobility Aids for a Hospital Robot," *ArXiv,* 2017.

[8] M. Kollmitz, A. Eitel, A. Vasquez, and W. Burgard, "Deep 3D perception of people and their mobility aids," *Robot. Auton. Syst.*, vol. 114, pp. 29–40, Apr. 2019.

[9] A. Fung, L. Y. Wang, K. Zhang, G. Nejat, and B. Benhabib, "Using Deep Learning to Find Victims in Unknown Cluttered Urban Search and Rescue Environments," *Curr. Robot. Rep.*, vol. 1, pp. 1-11, 2020.

[10] W. Ouyang *et al.*, "DeepID-Net: Deformable deep convolutional neural networks for object detection," *IEEE Conf. on Comput. Vis. Pattern Recognit.*, Boston, 2015, pp. 2403–2412.

[11] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 12, pp. 1219–1227, Dec. 1994.

[12] C.-S. Fahn, C.-P. Lee, and Y.-S. Yeh, "A real-time pedestrian legs detection and tracking system used for autonomous mobile robots," *Int. Conf. on Appl. Syst. Innov.*, 2017, pp. 1122–1125.

[13] D. Sanz, A. Ahmad, and P. Lima, "Onboard robust person detection and tracking for domestic service robots," *Iberian Robotics Conference*, Cham, Switzerland, 2015, pp. 547–559.

[14] Z. Yuan, Y. Zhang, and R. Duan, "RGB-D People Detection and Tracking from Small-Footprint Ground Robots," in *Int. Conf. on Control and Robot. (ICCR)*, Hong Kong, Sep. 2018, pp. 25–29.

[15] W. Huang, B. Zhou, K. Qian, F. Fang, and X. Ma, "Real-Time Multi-Modal People Detection and Tracking of Mobile Robots with A RGB-D Sensor," *Robot. Mechatron.*, p. 6, 2019.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[17] C. Zheng *et al.*, "Deep Learning-Based Human Pose Estimation: A Survey," *ArXiv*, 2022.

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

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[20] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." *ArXiv*, 2018.

[21] K. Agrawal and R. Lal, "Person Following Mobile Robot Using Multiplexed Detection and Tracking," in *Advances in Mechanical Engineering*, Singapore, 2021, pp. 815–822.

[22] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV*, Cham, 2016, pp. 21–37.

[23] R. Algabri and M.-T. Choi, "Target Recovery for Robust Deep Learning-Based Person Following in Mobile Robots: Online Trajectory Prediction," *Appl. Sci.*, vol. 11, no. 9, Art. no. 9, Jan. 2021.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection." *ArXiv*, 2018.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *ArXiv*, 2016.

[26] D. Du, L. Wang, H. Wang, K. Zhao, and G. Wu, "Translate-to-Recognize Networks for RGB-D Scene Recognition," *ArXiv*, 2019.

[27] S. Zia, B. Yüksel, D. Yüret, and Y. Yemez, "RGB-D Object Recognition Using Deep Convolutional Neural Networks," in *IEEE Int. Conf. on Comput. Vis. Workshops*, 2017, pp. 887–894.

[28] X. Wang, A. Shrivastava, and A. Gupta, "A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection,"*ArXiv*,2017.

[29] X. Chen, S. Xie, and K. He, "An Empirical Study of Training Self-Supervised Vision Transformers," *ArXiv*, 2021.

[30] P. Sermanet *et al.*, "Time-Contrastive Networks: Self-Supervised Learning from Video," *ArXiv*, 2018.

[31] S. Pirk, M. Khansari, Y. Bai, C. Lynch, and P. Sermanet, "Online Object Representations with Contrastive Learning," *ArXiv*, 2019.

[32] J.-T. Huang *et al.*, "Cross-Modal Contrastive Learning of Representations for Navigation using Lightweight, Low-Cost Millimeter Wave Radar for Adverse Environmental Conditions," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3333–3340, Apr. 2021.

[33] X. Wang, L. Zhang, D. Wang, and X. Hu, "Person detection, tracking and following using stereo camera," in *Int. Conf. on Graphic and Image Processing*, 2018, vol. 10615, pp. 83–91.

[34] L. Pang, Z. Cao, J. Yu, P. Guan, X. Chen, and W. Zhang, "A Robust Visual Person-Following Approach for Mobile Robots in Disturbing Environments," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2965–2968, 2020.

[35] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction," *ArXiv*, 2021.

[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," *ArXiv*, 2020.

[37] T. Lin *et al.*, "Microsoft COCO: Common Objects in Context" *ArXiv*, 2015.

[38] S. Kruthiventi, P. Sahay, and R. Biswal, "Low-light pedestrian detection from RGB images using multi-modal knowledge distillation," *IEEE Int. Conf. Image Process.*, 2017, pp. 4207–4211.

[39] L. Spinello and K. Arras, "People Detection in RGB-D Data," *IEEE Int. Conf. Intell. Robots Sys.*, pp. 3838-3843, 2011.

[40] L. Brigato and L. Iocchi, "A Close Look at Deep Learning with Small Data," *ArXiv,* 2020.

[41] T. Chen, C. Luo, and L. Li, "Intriguing Properties of Contrastive Losses," *ArXiv,* 2021.

[42] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection." *ArXiv*, 2020.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *ArXiv*, 2015.

[44] A. Shahroudy, J. Liu, T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in *IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1010–1019.

[45] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger." *ArXiv*, 2016.

[46] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection." *ArXiv*, 2020.

[47] C. E. Kim, M. M. D. Oghaz, J. Fajtl, V. Argyriou, and P. Remagnino, "A Comparison of Embedded Deep Learning Methods for Person Detection." *ArXiv*, 2019.

[48] G. Khan, Z. Tariq, and M. Khan, *Multi-Person Tracking Based on Faster R-CNN and Deep Appearance Features*. IntechOpen, 2019.