Current Robotics Reports (2020) 1:105–115
https://doi.org/10.1007/s43154-020-00011-8

DEFENSE, MILITARY, AND SURVEILLANCE ROBOTICS (S FERRARI AND P ZHU, SECTION EDITORS)

Using Deep Learning to Find Victims in Unknown Cluttered Urban Search and Rescue Environments

Angus Fung[1] · Long Yu Wang[1] · Kaicheng Zhang[1] · Goldie Nejat[1] · Beno Benhabib[1]

Published online: 16 July 2020

Abstract

Purpose of Review We investigate the first use of deep networks for victim identification in Urban Search and Rescue (USAR). Moreover, we provide the first experimental comparison of single-stage and two-stage networks for body part detection, for cases of partial occlusions and varying illumination, on a RGB-D dataset obtained by a mobile robot navigating cluttered USAR-like environments.

Recent Findings We considered the single-stage detectors Single Shot Multi-box Detector, You Only Look Once, and RetinaNet and the two-stage Feature Pyramid Network detector. Experimental results show that RetinaNet has the highest mean average precision (77.66%) and recall (86.98%) for detecting victims with body part occlusions in different lighting conditions.

Summary End-to-end deep networks can be used for finding victims in USAR by autonomously extracting RGB-D image features from sensory data. We show that RetinaNet using RGB-D is robust to body part occlusions and low-lighting conditions and outperforms other detectors regardless of the image input type.

Keywords Urban search and rescue · Victim identification · Body part occlusion · Low-lighting conditions · Deep learning

Introduction

Autonomous victim identification in urban search and rescue (USAR) scenes is challenging due to the occlusion of body parts in cluttered environments, variations in body poses and sensory viewpoints, and sensor noise [1]. The majority of classical learning approaches that have been developed to detect human body parts in cluttered USAR environments have focused on first extracting a set of handcrafted features, such as human geometric and skin region features [1] or histograms of oriented gradients (HOG) [2], and, then, training a supervised learning model (e.g., support vector machines (SVM)) using these features. The manual design of the features often requires empirical selection and validation [3], which can be time-consuming and entail expert knowledge. Furthermore, these approaches also use pre-defined rules to analyze the grouping of human parts. However, in USAR scenes, due to occlusions, multiple body parts of a person may not be visible at the same time for such groupings to occur.

Deep networks have the potential to be used in USAR to autonomously extract features directly from sensory data. While they have been applied to human body part detection in structured environments, such as operating rooms [4], office buildings [5], and outdoor urban settings [6–9], they have not been considered for cluttered USAR environments. In USAR, victim identification needs to take place in environments that are unknown, without any a priori information available regarding victim locations. Furthermore, the entire body of a victim may not be visible due to occlusions and lighting conditions may vary significantly.

This article is part of the Topical Collection on *Defense, Military, and Surveillance Robotics*

✉ Angus Fung
angus.fung@mail.utoronto.ca

Long Yu Wang
longyu.wang@mail.utoronto.ca

Kaicheng Zhang
kc.zhang@mail.utoronto.ca

Goldie Nejat
nejat@mie.utoronto.ca

Beno Benhabib
beno@mie.utoronto.ca

[1] Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada

Our previous research has focused on developing rescue robots that use learning for exploration, navigation and victim identification tasks in USAR environments [1, 10–12], and identifying landmarks in USAR scenes for 3D mapping [13–15]. In this paper, we present the first feasibility study that investigates the use of deep learning to address the victim identification problem in robotic USAR. We propose an overall architecture that uses deep neural network detectors with RGB-D images to identify body parts. We provide a detailed investigation of these deep neural networks to determine which ones are robust to body part occlusion and low-lighting conditions.

Related Work

Person and Body Part Detection Using Learning

In this section, we discuss classical machine learning and deep learning methods that have been used to identify human bodies or body parts in varying environments.

Person Detection Using Classical Machine Learning Approaches

There exists a handful of papers that have specifically focused on finding victims in USAR environments using RGB and depth images with classical learning classifiers, e.g., [1, 2, 16]. For these detection methods, input features or body-part grouping templates were needed to be handcrafted.

For example, our own previous work in [1] focused on first segmenting potential bodies from depth images based on concave curvature information. Then, 2D ellipses were fit to the segmented regions and an elliptical shape factor was computed. A recursive algorithm grouped potential body parts that were spatially close. The grouping of body parts, the elliptical shape factor, and skin color extracted from corresponding RGB images were all used as features for an SVM classifier.

In [2], infrared images were first used to detect human body temperature. In cases where a human body could not be detected using these images, a head detection technique was used. The head detection technique extracted Haar features from RGB images and HOG features from the infrared images. Adaboost was then used to classify the Haar features, while an SVM was used to classify the HOG features. The correspondence between the two sets of images was used to locate the head.

In [16], an infrared sensor was used to first detect a potential victim, and then trigger an RGB image capture of the scene. The RGB image was converted into grayscale and fed into a three-layer feed-forward neural network (NN) for body part classification. The input and hidden layers of the NN both contained 256 nodes, and the output layer contained 3 nodes representing a foot, hand, or body.

Other classical learning approaches have also been proposed for identifying human body parts in outdoor environments, such as parking lots and town centers [17], and indoor environments, such as retail stores and offices [18]. For example, in [18], a two-stage procedure was used for detecting the top of human heads using RGB and depth.

Person Detection Using Deep Learning Approaches with RGB Images

Recently, deep learning approaches have been used for body pose estimation by detecting individual body parts in RGB images [6–9]. In [6], Adapted Fast R-CNN (AFR-CNN) [19] and a dense CNN architecture were used to identify body parts as part of the pose estimator DeepCut. Training and evaluations were conducted on the Leeds Sports Poses [20] and MPII Human Pose [21] public datasets consisting of people doing sports or everyday activities such as eating, fishing, or typing, in both indoor and outdoor environments.

In [7], a sliding window detector with a 152-layer deep residual network (ResNet) [22] was used to detect body parts as part of the pose estimator DeeperCut. The model was trained on the same datasets as in [6].

In [9], a Faster R-CNN multi-person detector with a ResNet backbone was trained using the person category of the COCO public dataset [23] as part of the pose estimation process. This category contains adults and children doing sports or everyday activities in indoor or outdoor environments.

In [24], a Single Shot Multi-box Detector (SSD) [25] network was used to recognize body parts within a pose estimator. It was trained on both the MPII Human Pose and Leeds Sports Poses datasets containing annotations for the lower and upper legs, lower and upper arms, and head.

In [26], a You Only Look Once (YOLOv2) detection network was used to detect hands for a hand-pose estimator. The network was initialized using weights pre-trained on ImageNet [27], a public dataset with 14 million images consisting of humans, animals, and objects. It was then fine-tuned on an in-house RGB image dataset captured in different indoor environments.

In [28], a Feature Pyramid Network (FPN) was extended for body part instance segmentation. A multi-task loss was regressed to provide instance-level body part masks and surface patches, including left and right hands, feet, upper and lower legs, head, etc. The network was trained on the DensePose-COCO dataset.

In [29], a Detector-in-Detector network was proposed where the first detector (body detector) detects the body, and the second (parts detector) uses this information to detect hands and faces. The body detector uses Faster R-CNN with a ResNet-50 backbone while the parts detector builds on the body detector with two convolutional layers. A custom Human-Parts dataset consisting of 14,962 images and 106,879 annotations was used for training.

The availability of public datasets makes training of the aforementioned RGB image-based detectors very convenient. However, as these detectors are dependent on only RGB images, they have difficulty functioning in low-lighting USAR environments.

Person Detection Using Deep Learning Approaches with RGB and Depth Images

Only a few detectors have considered the use of both RGB and depth (RGB-D) images as inputs to their networks [4, 5], which are more robust against illumination and texture variations. In [4], a ResNet detector was used to detect upper body parts in an operating room. RGB-D information was used as inputs and a score map for upper body parts was the output of the network. The RGB-D data was captured by multiple cameras fixed around the operating room. The score map was then used by a random forest classifier to classify the overall human pose.

In [5], a long short-term memory (LSTM) network was used to detect head-tops. The first layer employed the head-top detection technique presented in [18], where for each possible head-top pixel, a set of bounding boxes were generated from both RGB and depth images. This set of boxes contained different ratios of potential human body proportions for a particular head-top. Each set of bounding boxes belonging to a head-top pixel was simultaneously fed into two LSTM chains, one for RGB images and one for depth images. A third LSTM fusion network used feature vectors from both LSTM chains at each link in the chain, and logistic regression was used at the end of the third LSTM chain to classify whether a person was detected.

The aforementioned detectors have been trained for structured indoor environments such as operating rooms, offices, and building corridors. People in such settings are less occluded and typically have common poses, such as standing, sitting, or lying down. Therefore, they do not generalize well to cluttered USAR environments in which people can be partially buried in a variety of different poses and with only small portions of their body visible. In this paper, we investigate the first use of deep learning networks to uniquely address these challenges for the victim identification problem in cluttered USAR scenes.

Deep Learning Networks for the Victim Identification Problem in USAR Environments

The proposed architecture for victim identification comprises three stages: data collection, training, and inference (Fig. 1). In the data collection stage, RGB and depth images are collected and used as inputs to the training stage, where features are extracted to produce a feature map used to train a detector for body part classification. In the inference stage, new RGB-D images are used as inputs for the trained detector for body part detection.

Two main approaches can be used when designing deep learning architectures for person detection. The first is a two-stage approach, which comprises a first stage that generates a set of region proposals indicating where target objects might be located, and a second stage classifies each proposed region as an object class or as background [30•]. In contrast, a single-stage detector performs object localization and classification concurrently [31]. When using such approaches, there is a trade-off between accuracy and speed. In this work, we investigate both these approaches for the victim identification problem in USAR environments. The two-stage detector we consider is Feature Pyramid Network (FPN) with Faster R-CNN [38]. It is more accurate than its predecessors such as Faster R-CNN [32•] and R-CNN [33]. The FPN with Faster R-CNN and its variations have been used in person and object detection applications, e.g. [34, 35]. However, they have not been used in cluttered USAR environments where body parts are occluded.

Single-stage detectors have the advantage of faster detection speed than the two-stage approaches, by removing the proposal generating stage. Their drawback is that they tend to have lower accuracy [30•]. The most popular single-stage detectors are SSD [25], YOLOv2 [36], YOLOv3 [37•], and RetinaNet [30•]. They have been adopted for real-time object detection in self-driving cars and environment monitoring applications [38–40]. However, they have not yet been applied to cluttered USAR scenes. Their faster detection speeds can be an advantage in time-critical search and rescue missions. The below sub-sections discuss how we have designed the network architectures for each of the aformentioned detectors to address the victim identification problem.

Two-Stage Detector

FPN with Faster R-CNN

The FPN with Faster R-CNN approach [32•] (Fig. 2a) uses an FPN to extract features from RGB-D images taken in USAR scenes, and outputs feature maps at different scales. The feature maps are generated by a backbone ResNet-50 model pretrained on the ImageNet dataset [27]. The feature maps are passed to the region proposal network (RPN) to generate bounding box proposals, which are used for the second-stage network for body part classification and bounding box refinement. The FPN network structure is designed to improve detection accuracy by extracting features at different scales while keeping computation cost low [32•]. In USAR, body

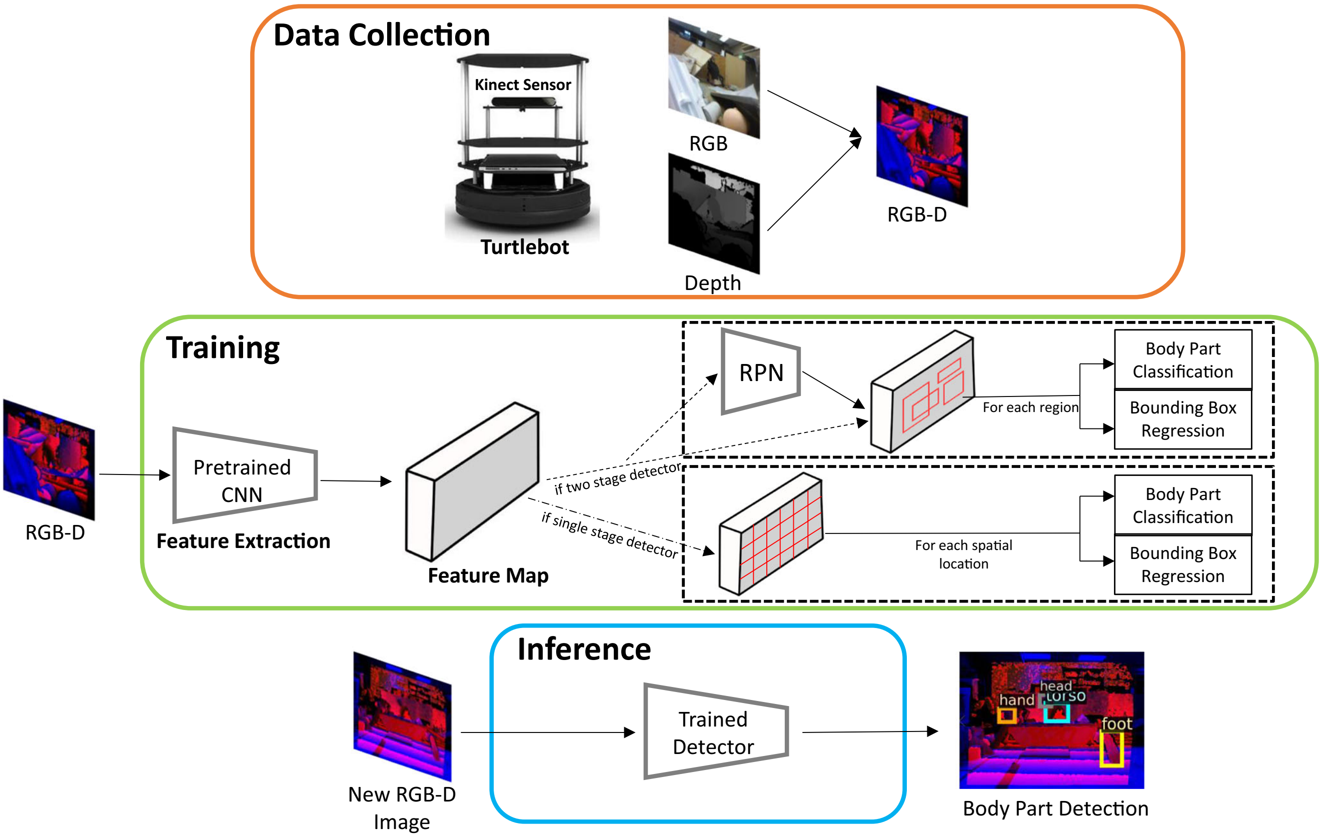

Fig. 1 Deep network architecture for body part detection

parts can appear at any scale based on their relative pose to the robot. As shown in Fig. 2a, the FPN structure consists of a multiple layer CNN (ResNet) that scales down an input image through convolution, and at the last layer scales it back up. Feature maps produced when scaled down are added element-wise to those produced when scaled up through lateral connections. While lower level feature maps have higher resolution and provide more details on small body parts, higher-level feature maps are processed through more convolution layers and gain more semantic understanding of the overall image. By combining the feature maps, the detector benefits from both aspects. With high-level semantic features in higher-resolution layers, the network becomes more robust to the detection of small body parts where occlusion is present. The number of output classes for the network is seven; six body parts (arm, foot, hand, head, leg, torso) and one for background.

Single-Stage Detectors

SSD

In SSD [25], RGB-D images are first processed by pretrained convolutional layers (VGG16 [41]) to output a feature map (Fig. 2b). The feature map goes through size reduction via a chain of convolution layers. The feature maps at different detection layers are processed independently by convolution filters to provide coordinates of victim body part bounding boxes and classification probabilities. Each cell in a feature map is associated with $k \times 4$ values representing the four coordinates of k bounding boxes centered at this cell [25]. The size and aspect ratio of the boxes are initialized using manually selected default values and then refined by the network, enabling the network to detect both small and large body parts. For each bounding box, the filters output one body part detection probability for each of the six classes of body parts, plus four additional scalars predicting the offset values to improve upon the bounding box coordinates [25]. The output, $6 + 4$ values, for each bounding box are compared with manually labeled ground truth to calculate losses.

YOLOv2

YOLO detectors use a single CNN [36, 37] for both body part localization and classification. The CNN is a Darknet-19 pretrained on the ImageNet dataset [27]. To apply YOLO detectors on our body part dataset from a cluttered USAR-like environment, the labels were annotated according to the Pascal VOC format [42]. An input RGB-D image is processed by the CNN shown in Fig. 2c [36] which directly outputs a

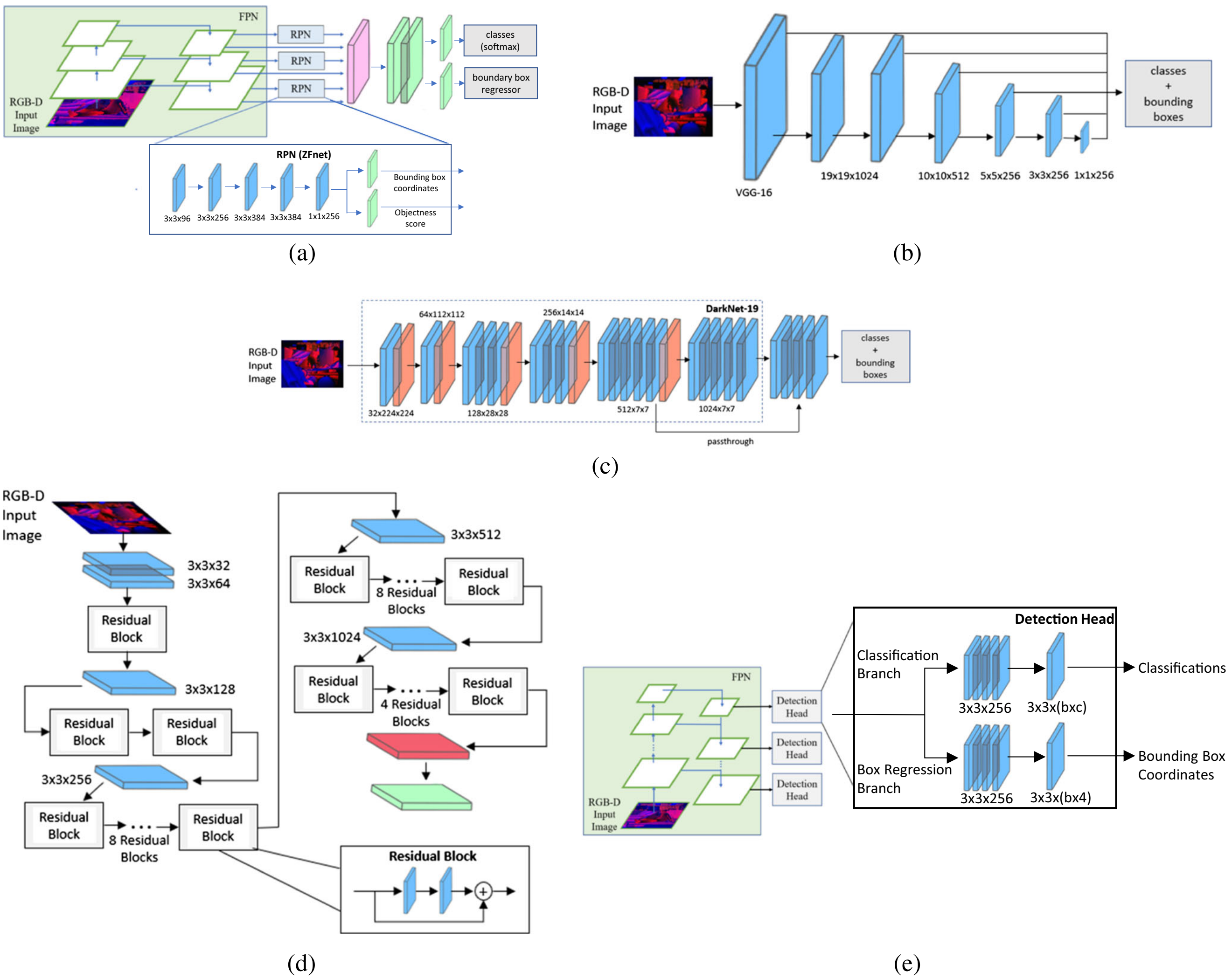

Fig. 2 Two stage detector: **a** FPN with Faster R-CNN Network Flow. Single-stage detectors: **b** SSD architecture, **c** YOLOv2 Architecture, **d** Darknet-53, the feature extraction layers used in YOLOv3, and **e** RetinaNet

$S \times S \times (b \times (c+5))$ tensor for bounding box localization and victim body part classification, where $c = 6$ is the number of body part classes. The number of grid cells that an input is divided into is $S \times S = 13 \times 13$. For each cell, b bounding boxes are initialized, and for each bounding box, $6 + 5$ scalar values are predicted. With respect to the five scalar values, four are for localization and one is for confidence, defined as $\Pr(class|object) \times IOU_p^t$. $\Pr(class|object)$ is the probability of whether a body part belongs to a specific class, conditioned on the grid cell containing a victim body part. The four localization values are the horizontal and vertical offsets against the grid cell, and the height and width are normalized against the size of the entire image, respectively. IOU_p^t is the Intersection Over Union calculated using the predicted body part bounding box, p, and the hand labeled ground truth bounding box, t, by dividing the area of overlap by the area of union.

YOLOv3

YOLOv3 [37•] further improves upon YOLOv2 by incorporating elements used in other state-of-the-art detection algorithms such as residual blocks [22] and feature pyramids [32•]. The feature extraction layers of YOLOv2 are replaced by a pretrained Darknet-53 (Fig. 2d), which consists of 53 layers mainly composed of 3 × 3 and 1 × 1 convolutions with residual blocks. The output feature map is passed through another 53 layers for detection. Detection is done at three different scales using a similar concept to feature pyramid networks [37•] to improve small body part detection. Namely, body parts are detected on three different-size feature maps, output by different layers. The larger dimension grids are responsible for detecting smaller body parts, and vice versa.

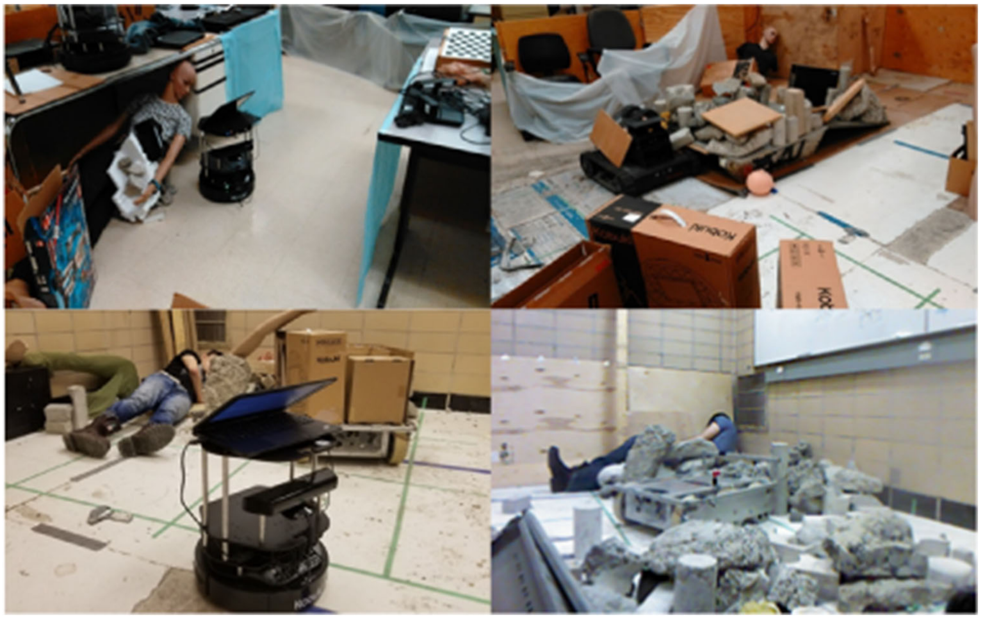

Fig. 3 USAR-like environment layout (top two and bottom two panels consist of mannequin and human victims, respectively)

RetinaNet

The RetinaNet architecture [30•] uses FPN for multi-scale feature extraction from RGB-D images, followed by two parallel branches of convolutional networks for body part classification and bounding box regression (Fig. 2e). Similar to FPN with faster R-CNN, the feature maps are generated by a backbone ResNet-50 model pretrained on the ImageNet dataset. RetinaNet uses the feature pyramid levels P3 to P7. At each level, $b = 9$ anchor boxes are selected for each spatial location of the feature map grid. Each box is associated with class prediction for all c classes (6 body parts + 1 background) and four coordinates. From a structural perspective, the feature map at each level of the pyramid is passed to two branches of convolutional networks in parallel. The classification branch consists of four $3 \times 3 \times 256$ convolution layers with rectified linear unit (ReLU) activation. This is followed by a $3 \times 3 \times (b \times c)$ convolution layer that outputs a $gridsize \times b \times c$ sized tensor, predicting the victim body part classifications for each anchor box. The box regression branch also consists of four $3 \times 3 \times 256$ convolution layers with ReLU activation, followed by a $3 \times 3 \times (b \times 4)$ layer that predicts the location coordinates of all bounding boxes (Fig. 2e).

Table 1 Training parameters

Network	Backbone	Batch	α	Iterations
FPN w. Faster R-CNN	ResNet-50	2	0.025	20,000
YOLOv2	Darknet-19	64	0.0001	5000
YOLOv3	Darknet-53	64	0.001	5000
SSD	VGG-16	4	0.00004	30,000
RetinaNet	ResNet-50	2	0.025	30,000

Training

In order to train all the designed detectors, we created a dataset consisting of 570 corresponding RGB-D images of both human and mannequin body parts in a cluttered USAR-like environment (Fig. 3). The images were obtained from a Kinect sensor onboard a mobile Turtlebot 2 platform. The images were manually labeled into six classes for training purposes: arm, foot, hand, head, leg, and torso. To account for different lighting conditions, we applied a random distribution of noise to the RGB images during preprocessing by using gamma correction. First, image pixel intensities were scaled from [0, 255] to [0, 1.0]. A gamma corrected image was then obtained using

$$O = I^{(1/G)} \tag{1}$$

where I is the scaled input image and G is the gamma value. The corrected image O is then converted back to the range [0, 255]. We distributed our image dataset to five possible gamma values: 0.1, 0.2, 0.4, 0.8, and 1.0, where $G = 1$ has no effect on the image and $G = 0.1$ is the darkest setting. We trained each network on RGB-D images consisting of both partially and fully visible parts. For the training process, k-fold cross validation ($k = 5$) was used to partition our dataset into training and validation images. Training took place on a Nvidia Titan V GPU. The learning parameters were initialized according to Table 1 and fine-tuned empirically for each network. The maximum training iterations required for all runs of a network are also reported in Table 1 for each network. The reported batch size is the number of images used to compute the

Table 2 Victim body part detection results for the networks on varying body part visibilities

Network	Type	Dataset images	Overall		Arm		Foot		Hand		Head		Leg		Torso	
			mAP	Recall	mAP	Recall	mAP	Recall	mAP	Recall	mAP	Recall	mAP	Recall	mAP	Recall
YOLOv2	RGB-D	Fully visible	74.99	71.60	61.88	53.93	68.02	62.50	54.92	53.34	96.23	92.75	76.42	73.73	92.46	91.35
		Partially occluded	66.93	64.41	56.83	52.17	64.72	66.67	64.92	62.50	81.08	75.93	67.53	68.33	66.51	60.87
	RGB	Fully visible	67.17	66.94	56.72	54.24	59.42	62.50	50.07	48.89	94.19	92.75	49.77	50.00	92.81	93.27
		Partially occluded	64.20	63.06	60.54	54.28	58.24	63.15	55.81	50.00	81.96	79.03	61.09	60.93	67.56	71.00
	Depth	Fully visible	72.96	69.97	56.59	55.94	68.47	65.18	55.39	50.00	91.22	88.71	73.75	68.65	92.34	91.35
		Partially occluded	68.78	62.91	56.18	50.00	64.87	66.67	74.29	68.75	74.04	68.52	63.10	58.33	80.22	65.22
YOLOv3	RGB-D	Fully visible	78.74	73.94	68.80	62.72	74.92	66.97	59.72	58.89	95.11	92.75	79.15	72.88	94.72	89.43
		Partially occluded	72.26	68.30	68.24	58.70	74.11	70.84	76.38	75.00	75.32	75.93	74.80	75.00	64.72	54.35
	RGB	Fully visible	69.95	64.77	54.03	48.31	62.58	57.15	63.72	53.34	95.37	91.94	55.86	52.31	88.10	85.58
		Partially occluded	64.79	55.42	57.18	41.31	64.75	64.59	62.19	53.13	77.72	68.52	65.37	55.00	61.54	50.00
	Depth	Fully visible	76.47	67.01	64.12	52.55	69.47	58.04	62.99	51.11	92.72	86.29	79.92	69.49	89.59	84.62
		Partially occluded	72.49	59.91	63.40	54.35	65.56	58.33	82.45	75.00	71.52	61.12	74.44	65.00	76.11	45.66
FPN with Faster R-CNN	RGB-D	Fully visible	79.55	88.51	78.32	91.07	67.44	79.31	61.29	72.45	94.34	96.77	81.33	92.37	94.62	99.07
		Partially occluded	73.06	85.51	68.38	69.59	69.59	80.36	55.72	70.00	89.59	92.42	76.01	91.41	79.04	91.13
	RGB	Fully visible	75.50	84.60	73.11	59.24	59.24	69.30	68.98	68.98	87.68	89.68	69.71	83.33	94.28	96.08
		Partially occluded	67.52	78.6	64.78	55.74	55.74	64.66	62.46	62.46	83.11	85.29	60.15	74.14	78.90	87.50
	Depth	Fully visible	74.12	83.39	68.25	65.85	65.85	73.27	47.99	47.99	91.61	94.93	78.06	86.07	92.95	97.12
		Partially occluded	69.06	80.48	64.47	57.08	57.08	71.82	50.18	50.18	89.97	92.95	71.00	82.54	81.68	88.89
RetinaNet	RGB-D	Fully visible	82.46	90.23	76.50	91.96	76.03	83.62	70.89	77.55	96.13	98.39	76.03	89.83	99.21	100.00
		Partially occluded	77.66	86.98	67.20	86.79	75.84	81.25	72.51	81.25	90.05	93.18	72.01	87.50	88.40	91.94
	RGB	Fully visible	79.33	90.70	74.63	94.83	71.10	79.83	79.48	89.36	88.66	90.48	68.48	91.67	94.65	98.04
		Partially occluded	72.56	86.63	69.46	90.35	65.53	77.59	71.63	83.70	84.93	88.24	62.93	87.93	80.89	91.96
	Depth	Fully visible	75.18	88.55	61.72	89.42	75.43	88.79	56.05	66.28	92.32	96.38	73.22	94.26	92.32	96.15
		Partially occluded	72.74	87.45	60.96	91.13	69.63	85.45	62.02	72.45	88.87	92.31	71.20	92.06	83.77	91.27
SSD	RGB-D	Fully visible	71.64	70.78	66.71	60.53	64.00	63.95	39.35	32.43	89.92	93.64	72.98	76.60	96.88	97.56
		Partially occluded	68.65	69.24	61.17	54.88	54.05	57.89	47.39	39.53	89.56	91.35	69.16	76.19	90.56	95.59
	RGB	Fully visible	69.01	68.99	50.55	52.22	69.77	63.41	55.90	45.45	89.51	93.28	57.41	65.00	90.57	94.59
		Partially occluded	67.20	67.14	48.61	51.16	60.26	60.53	52.64	40.90	89.20	91.54	63.53	67.04	88.95	91.66
	Depth	Fully visible	71.01	71.07	53.00	47.27	71.82	72.03	49.03	40.00	85.30	90.35	72.45	79.82	94.43	96.97
		Partially occluded	65.35	67.24	43.14	40.91	65.99	67.02	51.14	46.00	84.42	83.96	66.40	73.22	87.00	92.31

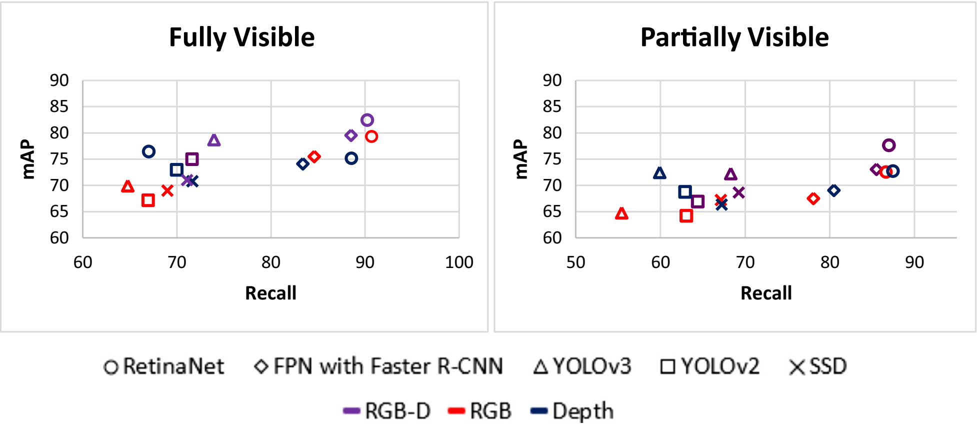

Fig. 4 Comparison of the precision-recall results for both the fully visible and partially occluded body parts for all networks, values are averaged across all body parts

gradient for backpropagation. For deeper networks, this is generally limited by GPU memory (e.g., for RetinaNet, the maximum is two images on our GPU). This same training procedure was also implemented separately on only RGB and depth images for comparison.

Experiments

Experiments were performed on a validation set of images from our dataset. All predicted regions equal to 0.5 with the manually labeled ground truth were accepted, which is common for object detection benchmarking [43]. Furthermore, repetitive detection of the same object in an image was minimized by using non-maximum suppression (NMS) [44] with the default threshold of 0.45 [36].

We chose 11-point mean average precision (mAP) [42] and recall as evaluation metrics. Recall was used to define the percentage of true victim body parts detected, and mAP measured the robustness of each network in maintaining high precision in tradeoff for higher recall. The precision-recall results for both the fully visible and partially occluded body parts for all networks are presented in Table 2 and Fig. 4. Furthermore,

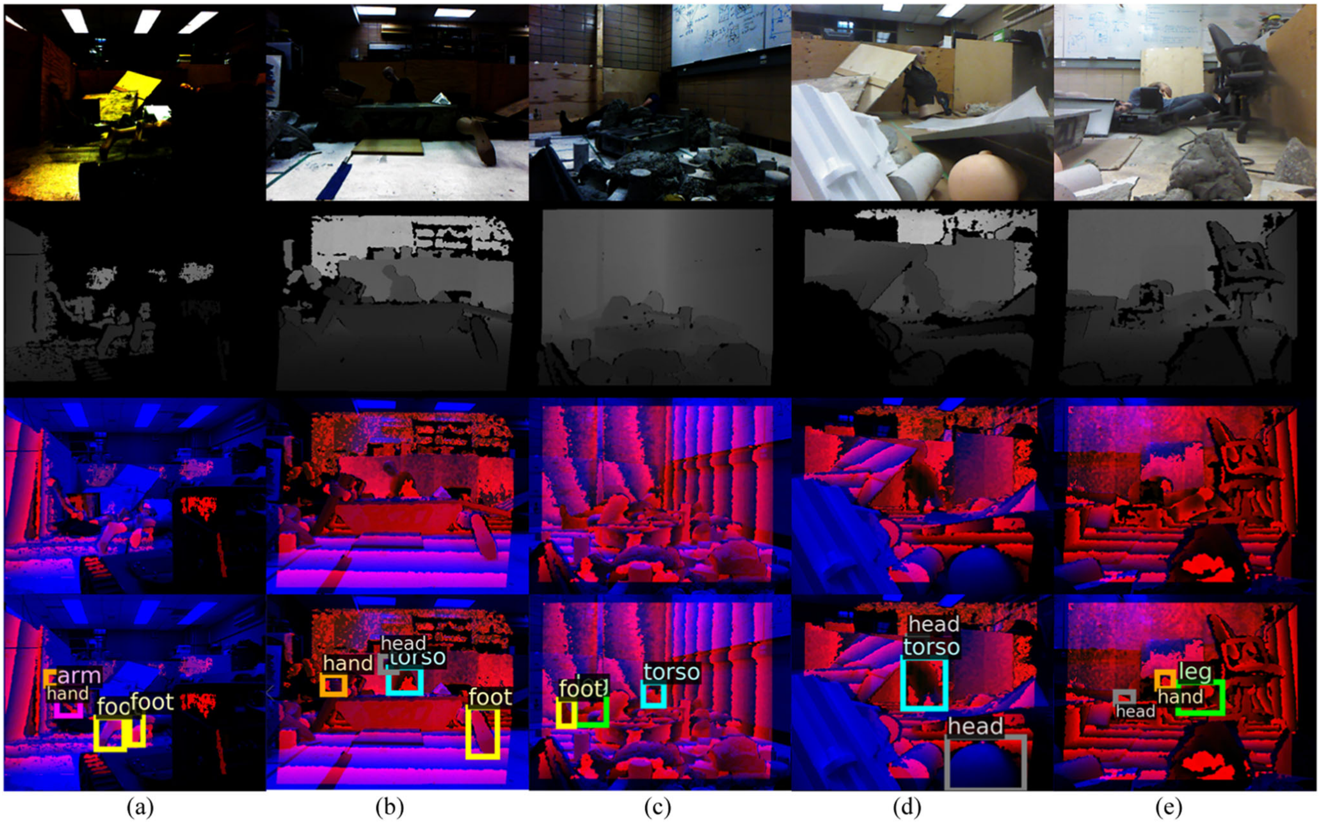

Fig. 5 Test results from RetinaNet. Each sub-figure contains, from top to bottom, the RGB input image, the depth input image, the combined RGB-D image, and the detection output. Gamma values are 0.1, 0.2, 0.4, 0.8, and 1.0 from **a** to **e**, respectively

results for the networks using RGB-only or depth-only images are also presented for comparison.

Table 2 presents the results for each individual body part. In general, RetinaNet had higher overall precision-recall for both the fully visible and partially occluded datasets, demonstrating its robustness to occlusion. The main advantage of RetinaNet is its focal loss which allows the network to focus on harder training examples by down-weighting the contribution of easier examples (e.g., fully visible body parts) in the loss function [30•]. This allows the network to focus on harder examples (e.g., occluded body parts) and harder classes (e.g., body parts that are more difficult to detect). Namely, the focal loss allows RetinaNet to significantly outperform the other networks, in some cases with up to 43% performance improvement on the most difficult body parts to detect: a hand and a foot. Therefore, despite being a single-stage detector, RetinaNet is able to outperform the two-stage detector here. FPN with Faster R-CNN, being the two-stage detector, outperforms the other single-stage detectors such as YOLOv2, YOLOv3, and SSD overall when using the RGB-D and RGB datasets. The YOLO networks generally performed better than SSD as they used higher resolution input images. One possible reason that FPN with Faster R-CNN was able to outperform the YOLO and SSD detectors is due to its lateral connections that produce high-resolution high-level semantic feature maps, allowing it to detect small body parts. For example, being able to capture small features such as fingers on hands can result in more accurate hand detection, especially when the hand is partially occluded.

The hand, due to self-occlusion, size, and its similarities with the foot, was difficult to detect for a number of the networks, especially, for instances where the spacing between fingers is less distinct. In contrast, the head and torso were easier to detect with higher precision-recall for the majority of the networks. Using the RGB-D information resulted in higher overall precision and recall for the majority of the networks compared to only using RGB or depth data. The RGB-D data incorporates color, geometry, and scale information, while being invariant to illumination. By further analyzing failure cases, it was observed that the other single-stage detectors, the two YOLO and the SSD detectors could not handle changes in illumination such as dim lighting conditions as well as RetinaNet. As depth is invariant to lighting, this resulted in better precision in these networks for the depth-only dataset over the RGB dataset, especially for the YOLO detectors. The robustness of RetinaNet to illumination also suggests that the network has encoded stronger illumination-invariant features (i.e., using focal loss).

Figure 5 shows the performance of RetinaNet using the RGB-D dataset under the different illumination conditions. In Fig. 5a, with the lowest lighting condition, two feet, an arm, and a hand of a potential victim were detected. In Fig. 5b with the second lowest lighting condition, the partially occluded torso and a head of one potential victim were detected along with a hand and foot of other potential victims. Both Fig. 5 c and e exhibit large body part occlusions, with self-occlusion as well as by clutter, while Fig. 5d presents partially occluded heads at different viewpoints and scales. RetinaNet was able to detect these body parts, demonstrating its ability to not only deal with occlusion, but also different illumination conditions and body parts of varying viewpoints and scale.

Conclusions

In this paper, we investigated, for the first time, the use of deep learning networks to address the victim identification problem in cluttered USAR environments. By providing the first feasibility and comparison study of state-of-the-art detectors, our results showed that deep networks can be trained to perform in dark cluttered environments by including RGB-D information, and we can use deep learning to detect partially occluded body parts. In general, using RGB-D information resulted in higher precision-recall compared to only using RGB or depth data. With respect to the individual detectors, the single-stage detector RetinaNet had both higher recall and mean average precision than the other detectors. By adopting such end-to-end deep networks, we can eliminate the time-consuming process of manually defining features to extract from such complex environments.

Authors' Contributions The authors Angus Fung and Long Yu Wang contributed equally to this work.

Funding Information This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada Research Chairs (CRC) Program, and the NVIDIA GPU grant.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Human and Animal Rights and Informed Consent This article does not contain any studies with human or animal subjects performed by any of the authors.

References

Papers of particular interest, published recently, have been highlighted as:

- Of importance

1. Louie W-YG, Nejat G. A victim identification methodology for rescue robots operating in cluttered USAR environments. Adv Robot. 2013;27:373–84. https://doi.org/10.1080/01691864.2013.763743.

2. Hui N, Li-gang C, Ya-zhou T, Yue W. Research on human body detection methods based on the head features on the disaster scenes. In: 2010 3rd International Symposium on Systems and Control in Aeronautics and Astronautics. China: Harbin; 2010. p. 380–5.
3. Nguyen DT, Li W, Ogunbona PO. Human detection from images and videos: a survey. Pattern Recogn. 2016;51:148–75. https://doi.org/10.1016/j.patcog.2015.08.027.
4. Kadkhodamohammadi A, Gangi A, Mathelin M de, Padoy N (2017) A multi-view RGB-D approach for human pose estimation in operating rooms. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). Santa Rosa, USA, pp 363–372.
5. Li H, Liu J, Zhang G, et al. Multi-glimpse LSTM with color-depth feature fusion for human detection. Beijing: IEEE International Conference on Image Processing; 2017.
6. Pishchulin L, Insafutdinov E, Tang S, et al (2016) DeepCut: joint subset partition and labeling for multi person pose estimation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Las Vegas, pp 4929–4937.
7. Insafutdinov E, Pishchulin L, Andres B, et al. DeeperCut: a deeper, stronger, and faster multi-person pose estimation model. In: Computer Vision – ECCV 2016. Cham, Amsterdam: Springer; 2016. p. 34–50.
8. Iqbal U, Gall J. Multi-person pose estimation with local joint-to-person associations. In: Hua G, Jégou H, editors. Computer Vision – ECCV 2016 Workshops. Cham: Springer International Publishing; 2016. p. 627–42.
9. Papandreou G, Zhu T, Kanazawa N, et al. Towards accurate multi-person pose estimation in the wild. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 3711–9.
10. Liu Y, Nejat G. Multirobot cooperative learning for semiautonomous control in urban search and rescue applications. J Field Robot. 2016;33:512–36. https://doi.org/10.1002/rob.21597.
11. Doroodgar B, Liu Y, Nejat G. A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims. IEEE Trans Cybern. 2014;44:2719–32. https://doi.org/10.1109/TCYB.2014.2314294.
12. Zhang K, Niroui F, Ficocelli M, Nejat G. Robot navigation of environments with unknown rough terrain using deep reinforcement learning. In: 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR); 2018. p. 1–7.
13. Zhang Z, Nejat G, Guo H, Huang P. A novel 3D sensory system for robot-assisted mapping of cluttered urban search and rescue environments. Intell Serv Robot. 2011;4:119–34. https://doi.org/10.1007/s11370-010-0082-3.
14. Zhang Z, Nejat G. Intelligent sensing systems for rescue robots: landmark identification and three-dimensional mapping of unknown cluttered urban search and rescue environments. Adv Robot. 2009;23:1179–98. https://doi.org/10.1163/156855309X452511.
15. Zhang Z, Guo H, Nejat G, Huang P. Finding disaster victims: a sensory system for robot-assisted 3D mapping of urban search and rescue environments. In: Proceedings 2007 IEEE International Conference on Robotics and Automation; 2007. p. 3889–94.
16. Shamroukh R, Awad F. Detection of surviving humans in destructed environments using a simulated autonomous robot. In: 2009 6th International Symposium on Mechatronics and its Applications. Sharjah; 2009. p. 1–6.
17. Shu G, Dehghan A, Oreifej O, et al. Part-based multiple-person tracking with partial occlusion handling. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence; 2012. p. 1815–21.
18. Liu J, Zhang G, Liu Y, Tian L, Chen YQ. An ultra-fast human detection method for color-depth camera. J Vis Commun Image Represent. 2015;31:177–85. https://doi.org/10.1016/j.jvcir.2015.06.014.
19. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Conference on Neural Information Processing Systems.
20. Johnson S, Everingham M. Clustered pose and nonlinear appearance models for human pose estimation. In: Procedings of the British Machine Vision Conference 2010. Aberystwyth: British Machine Vision Association; 2010. p. 12.1–12.11.
21. Pishchulin L, Andriluka M, Schiele B (2014) Fine-grained activity recognition with holistic and pose based features. arXiv:14061881 [cs].
22. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016. p. 770–8.
23. Lin T-Y, Maire M, Belongie S, et al (2014) Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision – ECCV 2014. Springer International Publishing pp 740–755.
24. Wang X, Hu J, Jin Y, et al. Human pose estimation via deep part detection. In: Zhai G, Zhou J, Yang X, editors. Digital TV and Wireless Multimedia Communication. Singapore, Springer Singapore; 2018. p. 55–66.
25. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer Vision – ECCV 2016. Cham: Springer International Publishing; 2016. p. 21–37.
26. Panteleris P, Oikonomidis I, Argyros A (2017) Using a single RGB frame for real time 3D hand pose estimation in the wild. arXiv:171203866 [cs].
27. Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 248–55.
28. Güler RA, Neverova N, Kokkinos I. Densepose: dense human pose estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 7297–306.
29. Li X, Yang L, Song Q, Zhou F (2019) Detector-in-detector: multi-level analysis for human-parts. arXiv:190207017 [cs].
30.• Lin T-Y, Goyal P, Girshick RB, et al. Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV); 2017. p. 2999–3007. **This paper introduces RetinaNet, a single stage object detector that uses focal loss to focus training on hard examples.**
31. Liu L, Ouyang W, Wang X, et al (2018) Deep learning for generic object detection: a survey. arXiv:180902165 [cs].
32.• Lin T-Y, Dollar P, Girshick R, et al. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE; 2017. p. 936–44. **This paper introduces Feature Pyramid Networks, a two stage detector that uses lateral connections to build high-level semantic feature maps at all scales.**
33. Girshick R, Donahue J, Darrell T, Malik J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society; 2014. p. 580–7.
34. Gkioxari G, Girshick R, Dollár P, He K (2017) Detecting and recognizing human-object interactions. arXiv:170407333 [cs].
35. Lan X, Zhu X, Gong S (2018) Person search by multi-scale matching. arXiv:180708582 [cs] 18.
36. Redmon J, Farhadi A (2016) YOLO9000: better, faster, stronger. arXiv:161208242 [cs].
37.• Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. CoRR abs/1804.02767: **This paper introduces YOLOv3, an improvement of YOLOv2 using residual blocks and feature pyramids**.

38. Kato S, Takeuchi E, Ishiguro Y, Ninomiya Y, Takeda K, Hamada T. An open approach to autonomous vehicles. IEEE Micro. 2015;35:60–8. https://doi.org/10.1109/MM.2015.133.
39. (2018) Open-Source To Self-Driving. Contribute to CPFL/Autoware development by creating an account on GitHub. Computing Platforms Federated Labratory.
40. Thakar V, Saini H, Ahmed W, et al (2018) Efficient single-shot Multibox detector for construction site monitoring. arXiv: 180805730 [cs].
41. Simonyan K, Zisserman A (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 14091556.
42. Everingham M, Van Gool L, Williams CKI, et al. The Pascal visual object classes (VOC) challenge. Int J Comput Vis. 2010;88:303–38. https://doi.org/10.1007/s11263-009-0275-4.
43. COCO - Common Objects in Context. http://cocodataset.org/#detection-leaderboard. Accessed 9 Oct 2018.
44. Non-max Suppression - Object detection. In: Coursera. https://www.coursera.org/lecture/convolutional-neural-networks/non-max-suppression-dvrjH. Accessed 14 Nov 2018.