Sim-to-Real Learning for Casualty Detection from Ground Projected Point Cloud Data

Roni Permana Saputra[1,2], Nemanja Rakicevic[1], Petar Kormushev[1]

***Abstract*— This paper addresses the problem of human body detection—particularly a human body lying on the ground (a.k.a. casualty)—using point cloud data. This ability to detect a casualty is one of the most important features of mobile rescue robots, in order for them to be able to operate autonomously. We propose a deep-learning-based casualty detection method using a deep convolutional neural network (CNN). This network is trained to be able to detect a casualty using a point-cloud data input. In the method we propose, the point cloud input is pre-processed to generate a depth image-like ground-projected heightmap. This heightmap is generated based on the projected distance of each point onto the detected ground plane within the point cloud data. The generated heightmap—in image form—is then used as an input for the CNN to detect a human body lying on the ground. To train the neural network, we propose a novel sim-to-real approach, in which the network model is trained using synthetic data obtained in simulation and then tested on real sensor data. To make the model transferable to real data implementations, during the training we adopt specific data augmentation strategies with the synthetic training data. The experimental results show that data augmentation introduced during the training process is essential for improving the performance of the trained model on real data. More specifically, the results demonstrate that the data augmentations on raw point-cloud data have contributed to a considerable improvement of the trained model performance.**

I. INTRODUCTION

Detecting injured people, i.e. casualties, during search and rescue (SAR) missions is a key challenge faced in the area of SAR robotics. To be able to detect a casualty, a SAR robot needs to perceive one or more physical properties of the casualty, such as visual features, temperature, scent and 3D body shape. A wide range of research studies have been conducted relating to searching and detecting human bodies and leveraging the advances in development for various sensors that can perceive these physical properties [1, 2].

Recently, significant progress has been made in developing human presence detection techniques for general cases, such as pedestrian detection and human activity capturing [3, 4]. Most of these techniques perform optimally when detecting people who are fully visible and presented in certain pose variations, such as standing, walking and sitting. Moreover, in most cases, the sensors or cameras are used to detect the persons face perpendicularly with respect to the object of interest, because otherwise additional image processing is needed to align them properly [5]. The following particular cases are still under-explored problems:

[1]Authors are with Robot Intelligence Lab, Dyson School of Design Engineering, Imperial College London, UK {r.saputra, n.rakicevic, p.kormushev}@imperial.ac.uk

[2]Roni P. Saputra is also with the Research Center for Electrical Power and Mechatronics, Indonesian Institute of Sciences - LIPI, Indonesia

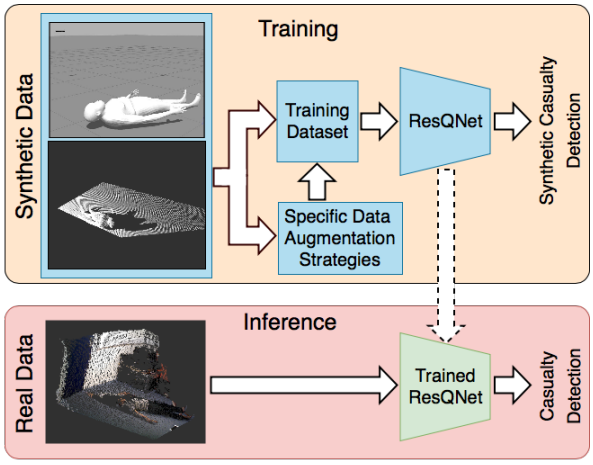


Fig. 1: Sim-to-real casualty detection learning pipeline. **Top (learning stage)**: The proposed ResQNet framework is trained using synthetic (augmented) data obtained from simulation; **Bottom (inference stage)**: The trained ResQNet uses real sensor data to perform casualty detection.

1) Detecting injured people, usually a human body lying on the ground. This case leads to significant difference in pose variation compared to with standard human presence detection techniques. Moreover, the fact that the floor is right under the body makes depth segregation extremely challenging;
2) Using an onboard sensor or camera on ground robots (facing forward) to detect a human body lying on the ground. This scenario introduces a limited viewing angle and permits the camera to only partially observe the object of interest.

On the other hand, the data-driven approach, such as deep convolutional neural networks (CNNs), is currently popular in various research areas. Particularly for visual perception tasks, leveraging various advanced techniques using CNNs has achieved considerable progress [6–12]. However, to achieve such good performance, it is well known that this data-driven approach requires an extensive amount of data.

In practice, a large amount of training data is not always available, especially for cases that are problem specific. Also, generating such new datasets is expensive and non-trivial work. To the best of our knowledge, there is currently no available dataset for detecting a human body lying on the floor using sensors onboard a mobile ground robot.

In this study, we propose a data-driven approach for detecting a casualty using a 3D point cloud data input. We propose a sim-to-real learning technique (see Fig. 1), in which we use

a synthetic dataset—generated from simulations—to train a deep CNN and then apply the trained model to real sensor data. To achieve a good performance in inferring from real data, we introduce specific data augmentation strategies, such as noise augmentation, down-sampling and segment removal, for augmenting the simulated training data.

The main contributions of this work include:

1) Introducing ResQNet, a novel framework for deep-learning-based casualty detection using a point-cloud input;
2) Exploring specific data augmentation strategies to improve the sim-to-real performance of the proposed casualty detection approach;
3) Making publicly available our novel dataset consisting of synthetic and real 3D point-cloud data, containing human bodies lying on the ground upon publication. [1]

II. RELATED WORK

Rapid developments in advanced machine learning, especially in the area of deep learning techniques—such as deep CNNs—has increased the use of these techniques in a wide range of computer vision research areas [6–12]. These CNN-based techniques rely significantly on a large amount of training data to achieve an acceptable performance and generalisation. However, in many cases, such large datasets are not always openly available. For instance, to the best of our knowledge, currently, there is no available existing dataset that is suitable for the particular purpose of our study, which utilises point cloud data inputs to detect a human body lying on the ground. Generating datasets from real sensor data is not trivial and manually labelling this dataset is prohibitively expensive as well.

Work focusing on learning tasks from synthetic images has been extensively studied in recent years [13–20]. For instance, in [16], the researchers present studies on learning from synthetic images for 2D human pose estimation tasks. Tasks involving 3D pose estimations learned from synthetic data is also explored in [17], and in [18, 19] the authors investigate the subject by presenting action recognition tasks using a similar sim-to-real technique. Particularly in the task of human presence detection, in the works presented in [13, 20], learning from synthetic data is used to complete pedestrian detection tasks.

In more recent work on domain adaptation [21], the authors use a mixture of synthetic, real and realistic generated images to extract invariant features that enable a robot to learn a policy in simulation which would successfully transfer on a real robot. Alternatively, in the domain randomisation approach [22], a robust policy is learned by exposing the learner to a large number of various simulated scenarios which might encompass cases similar to the real ones.

To increase the resemblance between the synthetic image and the real image and increase the learning performance test with the real image, several post-processing scenarios can be injected onto the synthetic image. Rozantsev et al [23] describe their study on the technique of rendering synthetic images in their post-processing scenarios. These post-processing scenarios include: object boundary blurring (BB), motion blurring (MB), random noise (RN) and material properties (MPs).

[1] https://sites.google.com/view/sim-to-real-resqnet/

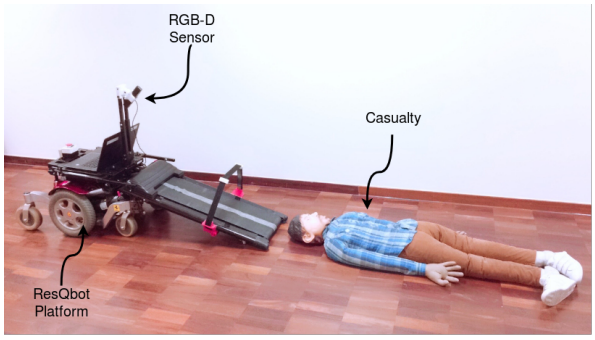


Fig. 2: ResQbot platform equipped with an RGB-D camera for detecting a casualty [25, 26]

Planche et al in [24] generate synthetic depth data from 3D models. In this work, they try to understand the causes of the noise in the real depth data and mimic this noise in synthetic depth data. Similarly, in the work in [23], RN and MB are added to the synthetic data. In addition to these noises, they also include radial and tangential lens distortion, lens scratching and lens graining into the data.

III. ROBOT PLATFORM AND HARDWARE

In previous work [25, 26], we have developed and tested a mobile rescue robot called ResQbot (as shown in Fig. 2). This robot is designed so that it can safely perform a casualty extraction task with human victims in rescue scenarios. To perform this casualty extraction procedure, a loco-manipulation approach is used, in which the robot uses its locomotion system—wheeled locomotion—to perform a manipulation task. An example of such a task used in SAR missions is loading the victim from the ground.

ResQbot is equipped with perception devices, including an RGB-D camera, that provides the perceptive feedback required during the operation. The RGB-D camera is also designed to further enable ResQbot to perform autonomous casualty detection. In this study, in particular, we used real data—i.e. point cloud data—obtained from the ResQbot sensor as part of our experiments. The work conducted in this study is also designed so that it can be implemented in real-time as part of the ResQbot system in future work. We refer readers to [25, 26] for more details about the ResQbot design and specification.

IV. METHODOLOGY

A. Problem Definition

The problem of casualty detection addressed in this study concerns when the body is lying on the ground. In contrast, most state-of-the-art human detection studies usually concern human bodies in upright poses. Moreover, in most cases the image is captured by the camera being faced perpendicularly towards the object (i.e. the person).

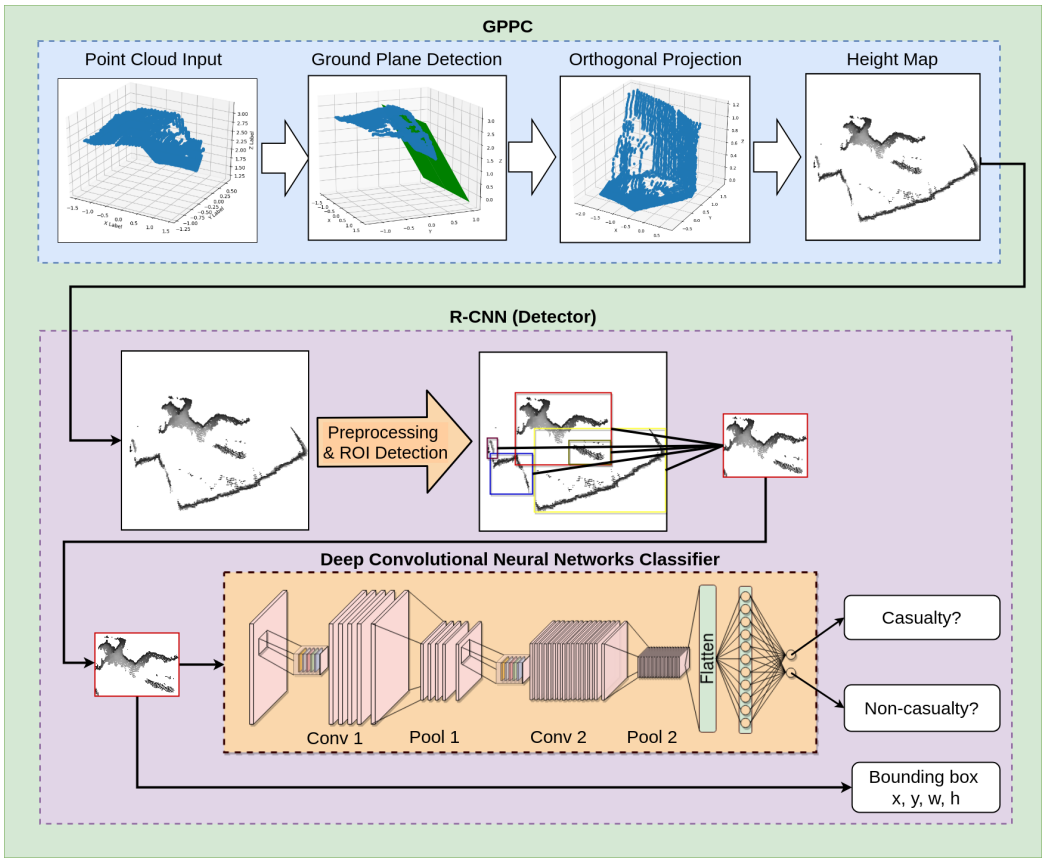


Fig. 3: ResQNet - proposed deep-learning-based casualty detection framework. It consists of the ground projected point cloud (GPPC) part (above) and the object detection and classification part (below).

In our study we aim to address a number of problems, which include:

1) Detecting a human body lying on the ground;
2) Using an onboard sensor on a mobile robot that is not facing perpendicularly towards the ground and the body;
3) Using a 3D point cloud as the data input.

To deal with this problem, we propose a data-driven approach based on training a deep neural network to detect a human body lying on the floor using a ground projected point cloud (GPPC) input. We introduce a novel framework called ResQNet for this deep-learning-based casualty detection from point cloud data. To ensure this data-driven approach performs well, it is essential to have a substantial amount of training data.

To the best of our knowledge, currently, there is no dataset available for training neural networks for our particular setup (i.e. input and output data types). However, collecting and labelling a real dataset using real sensor data is very expensive. Therefore, in this study, we propose to use synthetic data for training purposes. This data is generated using Gazebo simulations. In each simulation we use a virtual RGB-D sensor that produces point-cloud data and loads human body models into the simulation, in different positions and orientations with respect to the sensors.

To ensure that the ResQNet trained on synthetic data is transferable to real sensor data, in addition to the standard data augmentation usually implemented in deep learning we propose additional specific augmentation strategies presented in section V.D.

B. ResQNet Architecture

In this section, we present our proposed approach to the casualty detection problem defined in section IV.A. The full architecture of ResQNet is presented in Fig. 3. The architecture is based on integrating the proposed GPPC casualty detection technique presented by Saputra et al. [27, 28] and adopting a region-based convolutional network (R-CNN) for object detection [7]. The classification part is adopted from the LeNet architecture, where we refine several hyper-parameters for our own particular purposes (i.e. the input and output structure). The LeNet architecture is simple but expressive enough for our classification task. Our hypothesis is that the proposed method can still produce good results on detecting casualties, even though we are using a basic CNN architecture.

1) Generating GPPC heightmap: The initial step in the proposed approach is generating a heightmap from the point cloud data input projected onto the ground plane. In order to generate this heightmap, first we estimate a ground plane from the point cloud data input. We refer readers to [28] for more details about this GPPC approach. Contrary to the previous work in [28], here we generate a heightmap representation from the projected points instead of a discrete

grid-map to preserve 3D information of the objects, including casualties. The heightmap generated from GPPC is in the form of greyscale grid cells representing the maximum normalised distance of point pairs occupying each cell.

Let $\mathbf{C}$ represent an $m \times m$ 2D grid cells on the ground plane. Then find the point pair that corresponds to each cell component. If there is at least one point pair that corresponds to the cell, the greyscale value of this cell is the maximum normalised distance of the point pairs occupying the cell. On the other hand, if there are no point pairs occupying the cell, we assign the maximum greyscale value to this cell.

2) Region of interest (ROI) based on contour detection: The next part of the ResQNet architecture is detecting the ROI from the GPPC heightmap. Regarding the ROI detection presented in the R-CNN paper, a variety of sophisticated methods have been proposed for generating region proposals from the RGB image. However, in our case in particular, the heightmap produced using the GPPC process consists of greyscale images where the majority of the pixels not belonging to objects are white. Therefore, we hypothesise that simple contour detection is sufficient to generate a region proposal in this case. We use the OpenCV library to find contours through discretising an image. To minimise the number of region proposals, we filter the detected contours within the image. We select the detected contours to be considered as candidate objects of interest, based on the contour size being above a certain threshold.

3) CNN for classifying the ROIs: The final part of ResQNet is the CNN classifier. The particular CNN architecture used in ResQNet is adopted from the LeNet architecture presented in [29]. The networks consist of feature extraction parts with two layers of convolutions with max-pooling and downsampling at each output of the convolution layer. The output of the feature extraction part is then flattened and passed to a fully connected layer and softmax classifier layer. The output of this CNN is a binary classifier, that classifies each candidate region passed to the network into two classes, casualty and non-casualty.

V. OBTAINING THE DATASET

A. Generating Synthetic Human Body Models from an On-line Human Body Shape Modeller

The synthetic human bodies used in our experiments are created via an on-line human body modeller developed by the University of Michigan Transportation Research Institute (UMTRI) [30]. Fig. 4 illustrates the four different human body shapes generated by a set of different parameters in the on-line modellers [31]. For the purpose of generating training datasets, we create 216 synthetic human bodies by combining different value settings, including the stature, body mass index (BMI), the ratio of erect sitting height to stature (SHS) and age. We refer readers to [30] for more details about this on-line modeller and the parameter setting for this modeller. In addition to this, we also vary the orientation of the body w.r.t. the camera. Table I shows the combination of value settings that we use for generating the synthetic bodies.

TABLE I: The combination of value settings for generating the synthetic bodies.

Parameter	Value Range
Stature [cm]	1500, 1600, 1700, 1800, 1900, 2000
BMI $[kg/m^2]$	20, 25, 30
SHS [ratio]	0.4, 0.5, 0.6
Age [years]	20, 40, 60, 80
Orientation [deg]	$0, \pm 45, \pm 90, \pm 135, 180$

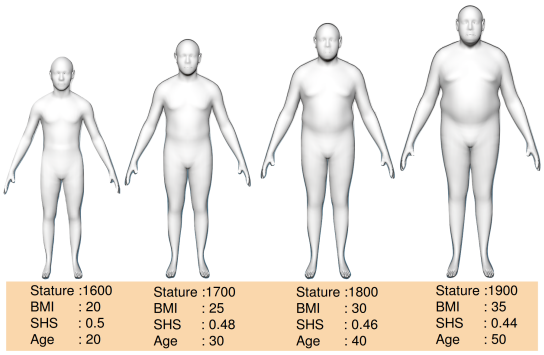


Fig. 4: Various human body models created via on-line human body modeller from UMTRI.

B. Generating Simulated Point Cloud Data Using Gazebo Simulation

To obtain synthetic point cloud data, we use Gazebo simulation. In this simulation, we simulate an RGB-D camera that can produce point cloud data simulation. The human body models created are then loaded into the Gazebo and set into various positions and orientations with respect to the RGB-D camera. In total we generate 10 thousand synthetic point cloud data with synthetic human body inside and another 10 thousand synthetic point cloud data with no synthetic human body inside. Fig. 5 shows the simulated RGB-D camera and human body in the Gazebo, and the synthetic point cloud produced from the simulation.

C. Obtaining a Real Dataset from the ResQbot sensor

The final aim of this work is to test and implement the ResQNet casualty detector using real sensor data from the ResQbot. Fig. 6 shows the real point-cloud data obtained from the ResQbot sensor. The point-cloud dataset that we collect for the experiments includes point clouds that contain a casualty (human body lying on the ground) and point-clouds that contain other objects that are not casualties (furniture, boxes, wall, stairs etc).

D. Augmenting the Dataset

One of our hypotheses is that introducing additional specific augmentation strategies to the synthetic training data can increase the similarity between the simulation and reality. Hence, we expect that by incorporating these augmenting strategies with the training data, the trained model would also perform well with real sensor data inputs.

In addition to the standard data augmentation—rotation, scaling and shifting—performed on the synthetic GPPC

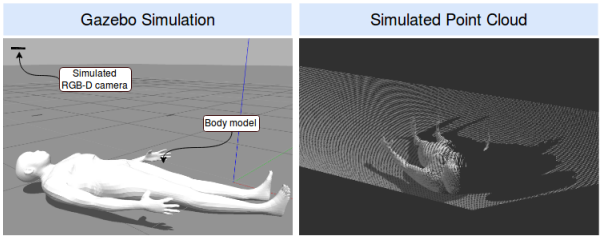


Fig. 5: Obtaining synthetic data from the Gazebo simulator. **Left**: Loading human body model into Gazebo and simulating a virtual RGB-D sensor. **Right**: Visualisation of synthetic point-cloud data generated within the Gazebo simulator.

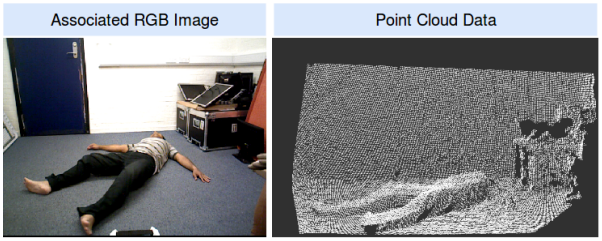


Fig. 6: Obtaining real point-cloud sensor data from ResQbot. **Left**: Associated RGB image in which the point-cloud data is taken. **Right**: Visualisation of real point-cloud data from the RGB-D sensor.

heightmap (or image), we also observe the effect of introducing additional augmenting strategies on the detectors performance. These special augmentation strategies include:

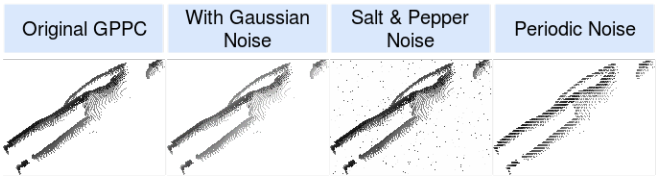
1) Introducing various noise to the GPPC image (i.e. direct input of the neural network), including Gaussian noise, salt-and-pepper noise (SnP) and periodic noise;
2) Introducing noise to the raw point-cloud sensor data readings (i.e. the input of ResQNet);
3) Reducing the resolution of the point-cloud data (i.e. down-sampling the numbers of points);
4) Partially removing segments of the point-cloud data (simulating partial observability or occlusions).

Fig. 7 illustrates the effects of various of augmentation strategies to the GPPC image (i.e. heightmap) as well as to the raw point cloud data.

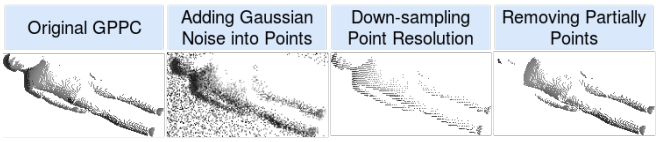
VI. EXPERIMENTAL SETUP AND RESULTS

To quantitatively evaluate the performance of our proposed approach, we conducted several experiments and performed the ablation study to analyse the contribution of each of the data augmentation strategies and the number of corresponding augmented data samples. The network implementation and training in our experiments was done using Keras. The hyperparameters we used in the experiments were adopted from LeNet implementation for image classification [29].

We trained the classification part of ResQNet (the CNN adopted from LeNet) on all cropped images, i.e. candidate regions of the synthetic GPPCs, classified as casualty or non-casualty. For the purpose of performance analysis, we independently trained multiple CNN instances, one for each data augmentation strategy and the number of augmented



(a) Adding noise into GPPC heightmap



(b) Adding noise into raw point clouds

Fig. 7: Illustration of various data augmentations strategies used within the proposed framework.

TABLE II: The summary of the parameter settings used for each of the data augmentation strategy.

	Augmentation Type	Parameters & Values
GPPC Images	Gaussian noise	μ: 0.2 σ: 10, 25, 50
	Salt & pepper noise	s-p ratio: 0.5 noise amount: 0.01-0.05
	Periodic noise	set row value to 255 - every 3, 5, 10 rows
Point Cloud Data	Sensor noise	μ: 0 σ: 0.1, 0.2, 0.25
	Down-sampling	scale: 1/50, 1/20, 1/10
	Removing segments	size: 50, 100, 150

samples it generated (incrementally increasing from 1000 to 10000 samples, with a step of 1000), and then evaluated the trained model. Parameter settings for each augmentation strategy can be found in Table II.

The training data is randomly shuffled and the class labels are equally distributed. The evaluation is done on the leave-one-out synthetic test set to analyse the trained model's accuracy, as well as on the real point-cloud data obtained from the ResQbot's RGB-D sensor to evaluate the model's generalisation capabilities. It is important to emphasise that the real sensor data is not used during any of the training instances, but only for evaluation purposes. To quantify the performance of the trained model on both the synthetic test data and real sensor test data, we calculate the classification accuracy as the performance measure, as well as the F_1 score, summarised in the Table III. Fig. 8 shows the classification accuracy, for each of the instances mentioned above, as a function of the number of augmented data samples.

Performance on the synthetic test data: We evaluate the trained models on 1000 previously unseen synthetic data samples. The results shown in Fig. 8a demonstrate that most of the trained models result in accurate predictions on the synthetic test data with up to 99% accuracy (see Table III).

Performance on the real sensor test data: We further evaluate the trained models on the real sensor data obtained from the ResQbot's sensor reading. This dataset is split into two groups: validation data consisting of 300 samples of point-cloud data containing a casualty and 300 samples that

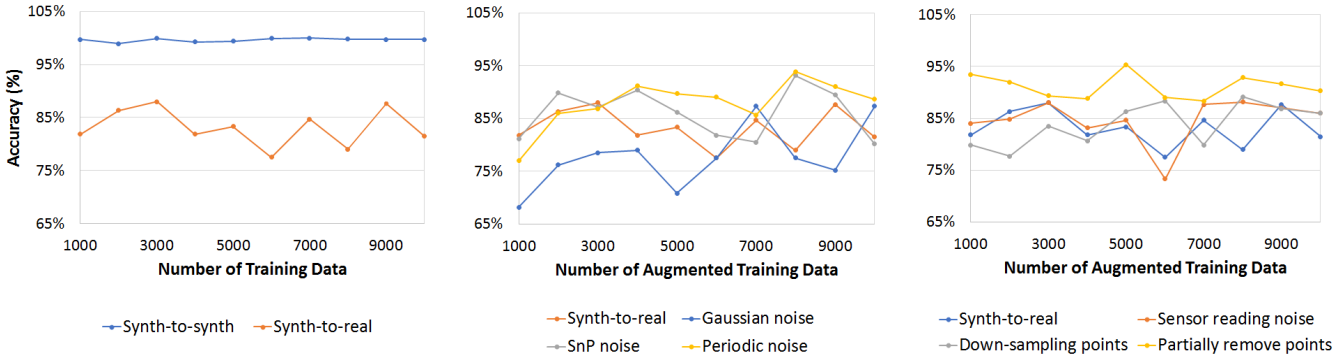


Fig. 8: Effects of different augmentation strategies on the classification accuracy of models trained on synthetic data; **Left**) comparison of the performance on the synthetic and real test datasets, of models trained without data augmentation; Effects of different augmentation strategies: **Centre**) applied to GPPC, **Right**) applied directly on point-cloud data.

do not contain a casualty; and testing data which consists of 100 samples of point-cloud data containing a casualty and 100 samples that do not contain a casualty. Fig. 8a, shows that the prediction accuracy of the trained models significantly drops (down to 80%) when classifying real sensor data inputs (see Table III).

The effect of GPPC image data augmentation: We hypothesised that augmenting synthetic training data could help improve the performance of the trained models in classifying real sensor data. To show this, we first performed data augmentation strategies to the GPPC images as direct inputs of the classifier network. We injected Gaussian, salt and pepper, as well as periodic noise to the image. The noise settings used in this experiments are described as follows:

1) *Injected Gaussian noise*: In this experiments we injected random Gaussian noise to each pixel of the GPPC images. We used random noises with mean 0.2 and variance 10-50.
2) *Salt and pepper noise*: We introduced a combination between salt (i.e. set pixel value to max.) and pepper (i.e. set pixel value to min.) in random pixels with a constant salt and pepper ratio (1:1) and with the noise contribution of 0.01-0.05.
3) *Periodic noise*: For introducing periodic noise to the GPPC images, we set the value of all pixels in one row to 255 (i.e. max value), and we repeat it every 3, 5 and 10 rows.

The results demonstrated in Fig. 8b and Table III indicate that in these experiments, the noise introduced to the GPPC images have only a small impact on improving the performance of the trained model classification of real sensor data. Salt and pepper (SnP) noise introduced into the GPPC training images bring the highest improvement to the accuracy (up to 84%). However, in our conducted experiments, Gaussian noise injected in the images remarkably reduces the performance of the networks. One reason for this might be because Gaussian noise is not a good approximation of the noise occurring on the real GPPC images.

The effect of raw point-cloud data augmentation: Since we proposed to use point cloud data as inputs, we also hypothesised that augmenting the raw point cloud data inputs could directly affect the performance of the trained models. To simulate imperfections of real point cloud data produced by the RGB-D sensor and the disturbances from environmental conditions, we experimented with adding Gaussian noise to the point clouds, down-sampling the points and partially removing point segments. The augmentation settings applied to the point cloud data used in this experiments are the following:

1) *Simulated sensor noise*: In real applications, the sensor measurements are not perfect and they contain measurement noise. The most common noise occurring in sensor measurements is white noise, which can be modelled as Gaussian noise with zero mean and a certain variance $\mathcal{N}(0, \sigma)$. To model this noise, we use $\sigma = 0.1, 0.2, 0.25$ based on the experiment results reported in [32].
2) *Down-sampling point cloud data*: We introduce the down sampling augmentation strategy applied to the point cloud data, in order to model different sensor resolutions. In our experiments, we introduced down-sampling to the original point cloud data with a scale 1:50, 1:20 and 1:10 to augment the data.
3) *Partially removing point segments*: In real life the point cloud data obtained from RGB-D sensor often returns NaN values when detecting a surface with bad reflective properties, or in bad lighting conditions. To model this imperfection, we introduce an augmentation strategy that partially removes random rectangular segments of the synthetic point cloud data used for the training process.

The results demonstrated in Fig. 8c show that in contrast with the previous augmenting strategies that target the GPPC images, these strategies significantly affect the performance of the trained models. Specifically, the results of the models trained on data augmented by partially removing point-cloud segments indicate that this strategy could considerably improve the model's performance to up to 91% accuracy when applied to real sensor data (see Table III).

The effect of the augmentation combination strategy: In addition to the independent augmentation experiments, we also investigated the effect of possible combination of

TABLE III: Summary of the experimental results, which shows the average performance of each data augmentation strategy used in the experiments.

Training & Augmentation	Accuracy	F1 Score
Synth-to-synth	99.63	96.78
Synth-to-real	83.16	79.85
+ Gaussian noise	77.75	71.48
+ SNP noise	84.68	82.38
+ Periodic noise	83.81	80.93
+ Sensor reading noise	85.98	83.78
+ Down-sampling points	87.88	86.28
+ Removing points	91.11	90.65

each augmentation strategy to improve the sim-to-real performance. We used the Bayesian Optimisation (BO) approach to find the optimal combination of the augmentation strategies that yields the highest model performance. To find this, we used a validation set of real sensor data that includes 300 point cloud data with a casualty in the data and another 300 point cloud data with a non-casualty.

In these experiments, we adapted the implementation from [33]. We used the default BO hyperparameters and performed the optimisation process in 25 iterations. The result of this hyperparameter optimisation process can be found in Figs. 9-10. We found that the optimal combination of the augmentation strategies consisted of: 1) 2000 samples augmented by adding sensor data noise, 2) 2000 samples augmented by down-sampling points, and 3) 6000 samples augmented by partially removing point segments. The results show that this combination can improve the performance of sim-to-real learning in term of detection accuracy in real sensor data up to 93% in validation data. We then tested the trained network from this optimal combination using unseen testing data consisting 100 point cloud data with casualty in it and another 100 point cloud data with no-casualty in it. The result shows that the trained network achieves 95% accuracy on real sensor data test set, with a balance of false positive and false negative errors.

VII. CONCLUSIONS AND FUTURE WORK

In this study, we have addressed the problem of casualty detection from point cloud data. More specifically, we proposed a novel sim-to-real domain-randomisation-based learning technique in which we trained our model to perform casualty vs non-casualty classification from synthetic point-cloud data. We introduced several data augmentation strategies for synthetic data, and observed the effect of each strategy, as well as their combinations, on the performance of the models trained with this data, when applied to the real sensor data test dataset.

The experimental results demonstrate that the data augmentation strategies on raw point-cloud data have contributed to superior model testing performance when compared to the data augmentation done using GPPC heightmap images. In particular, augmenting synthetic point-cloud data by partially removing point segments, has shown considerable effect on improving the trained model's classification performance on the real sensor test data, compared to other strategies. This can be explained as such a strategy simulates scenarios which occur in reality; e.g. the RGB-D sensor often gives NaN values when detecting a surface with bad reflective properties, or when bad lighting conditions also affect the quality of the point-cloud data from the RGB-D sensor.

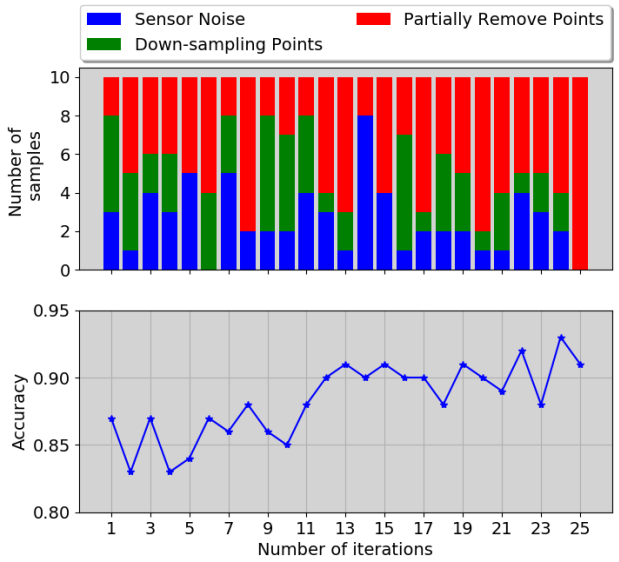


Fig. 9: Plots showing the contributing sample proportions coming from each data augmentation approach (legend on top) and the corresponding accuracy achieved on the synthetic point cloud training data. The y axes of the top and the bottom plots show the number of samples in thousand and the accuracy, respectively. The 25 iterations shown are obtained via Bayesian Optimisation [33].

As future work, we would like to extend this approach and examine possible implementations using point-cloud data produced by 3D LIDAR sensors, which have different properties than RGB-D sensors. Moreover, we will investigate other factors that can be randomized and would contribute to the robustness of the estimator. Some of the techniques to explore would be transfer learning and fine tuning to improve the final performance.

ACKNOWLEDGMENT

Roni Permana Saputra would like to thank the Indonesia Endowment Fund for Education – LPDP, for the financial support of the PhD program. The authors also would like to show our gratitude to James Paul Foster for sharing his comments and feedback on this work.

REFERENCES

[1] A. Ivanovs, A. Nikitenko, M. Di Castro, T. Torims, A. Masi, and M. Ferre, "Multisensor low-cost system for real time human detection and remote respiration monitoring," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, 2019, pp. 254–257.

[2] M. Andriluka, P. Schnitzspan, J. Meyer, S. Kohlbrecher, K. Petersen, O. V. Stryk, S. Roth, and B. Schiele, "Vision based victim detection from unmanned aerial vehicles," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 1740–1747.

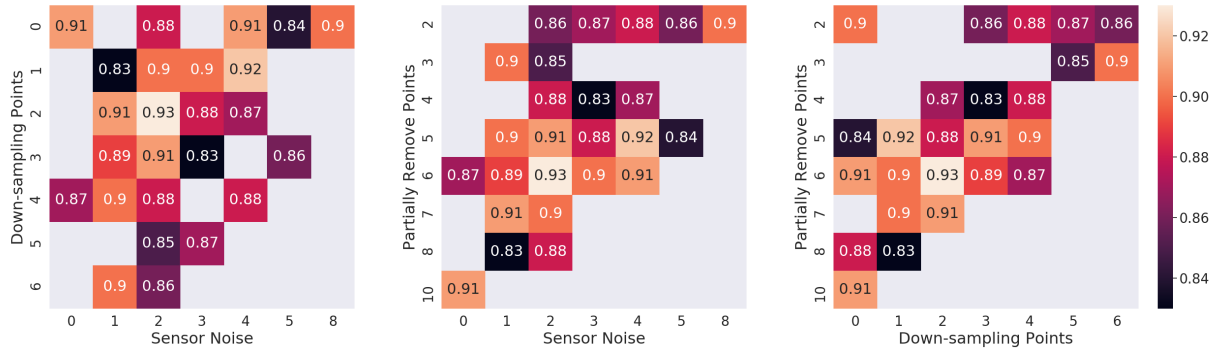


Fig. 10: Heatmaps showing the accuracy achieved for different combinations of data augmentation techniques. Each plot is a projection on a 2-dimensional plane corresponding to a pair of data augmentation techniques. The axes show the number of samples in thousand, and the color intensities represent the accuracy, as indicated in the colormap on the right.

[3] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 34, no. 3, pp. 334–352, 2004.

[5] X. Cao, Y. Wei, F. Wen, and J. Sun, "Face alignment by explicit shape regression," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2887–2894.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[8] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1137–1149, 2017.

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[12] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, "Megdet: A large mini-batch object detector," in *2018 IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6181–6189.

[13] J. Marin, D. Vázquez, D. Gerónimo, and A. M. López, "Learning appearance in virtual scenarios for pedestrian detection," in *2010 IEEE computer society conference on computer vision and pattern recognition (CVPR)*, 2010, pp. 137–144.

[14] J. Papon and M. Schoeler, "Semantic pose using deep networks trained on synthetic rgb-d," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 774–782.

[15] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *2017 IEEE computer society conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 109–117.

[16] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele, "Articulated people detection and pose estimation: Reshaping the future," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 3178–3185.

[17] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen, "Synthesizing training images for boosting human 3d pose estimation," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 479–488.

[18] H. Rahmani and A. Mian, "Learning a non-linear knowledge transfer model for cross-view action recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 2458–2466.

[19] H. Rahmani and A. Mian, "3d action recognition from novel viewpoints," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, 2016, pp. 1506–1515.

[20] L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormählen, and B. Schiele, "Learning people detection models from few training samples," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1473–1480.

[21] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 95–104.

[22] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[23] A. Rozantsev, V. Lepetit, and P. Fua, "On rendering synthetic images for training an object detector," *Computer Vision and Image Understanding*, vol. 137, pp. 24–37, 2015.

[24] B. Planche, Z. Wu, K. Ma, S. Sun, S. Kluckner, O. Lehmann, T. Chen, A. Hutter, S. Zakharov, H. Kosch, *et al.*, "Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 1–10.

[25] R. P. Saputra and P. Kormushev, "Resqbot: A mobile rescue robot for casualty extraction," in *2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2018, pp. 239–240.

[26] R. P. Saputra and P. Kormushev, "Resqbot: A mobile rescue robot with immersive teleperception for casualty extraction," in *Annual Conference of Towards Autonomous Robotic Systems (TAROS)*, 2018, pp. 209–220.

[27] R. P. Saputra and P. Kormushev, "Casualty detection for mobile rescue robots via ground-projected point clouds," in *Annual Conference of Towards Autonomous Robotic Systems (TAROS)*, 2018, pp. 473–475.

[28] R. P. Saputra and P. Kormushev, "Casualty detection from 3d point cloud data for autonomous ground mobile rescue robots," in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2018, pp. 1–7.

[29] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] M. P. Reed, U. Raschke, R. Tirumali, and M. B. Parkinson, "Developing and implementing parametric human body shape models in ergonomics software," in *Proceedings of the 3rd international digital human modeling conference*, 2014.

[31] "Human shapes - realistic human body shape modeler based on real data," http://humanshape.org/, accessed: 2018-07-30.

[32] M. Mirdanies and R. P. Saputra, "Experimental review of distance sensors for indoor mapping," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 8, no. 2, pp. 85–94, 2017.

[33] Fernando, "Bayesian optimization," https://github.com/fmfn/BayesianOptimization, 2018.