Pedestrian Proximity Detection using RGB-D Data

Adam Tupper
Department of Computer Science
and Software Engineering
University of Canterbury
Christchurch, New Zealand
atu31@uclive.ac.nz

Richard Green
Department of Computer Science
and Software Engineering
University of Canterbury
Christchurch, New Zealand
richard.green@canterbury.ac.nz

***Abstract*—This paper presents a novel method for pedestrian detection and distance estimation using RGB-D data. We use Mask R-CNN for instance-level pedestrian segmentation, and the Semiglobal Matching algorithm for computing depth information from a pair of infrared images captured by an Intel RealSense D435 stereo vision depth camera. The resulting depth map is post-processed using both spatial and temporal edge-preserving filters and spatial hole-filling to mitigate erroneous or missing depth values. The distance to each pedestrian is estimated using the median depth value of the pixels in the depth map covered by the predicted mask. Unlike previous work, our method is evaluated on, and performs well across, a wide spectrum of outdoor lighting conditions. Our proposed technique is able to detect and estimate the distance of pedestrians within 5m with an average accuracy of 87.7%.**

I. Introduction

In 2017, there were 39 pedestrian fatalities and 281 serious injuries as a result of vehicle-related accidents in New Zealand alone [1]. Furthermore, there were 243 workplace fatalities in New Zealand between 2010 and 2018 which were related to vehicles and machinery [2]. In total over 50% of all workplace fatalities over the same period were vehicle or machinery related [2]. These statistics highlight the need for increased safety measures for vehicles and machines operating in proximity to humans.

The safety envelope metaphor is used to describe the margin around a machine which should be free from obstructions, and more importantly humans, for safe operation. Though the responsibility has traditionally fallen to the human or operator to ensure that this margin is respected, as machines become more aware of their surroundings and become more autonomous this responsibility is shifting to the machine. Take for example a situation involving a pedestrian and an autonomous vehicle. While the pedestrian shares the responsibility to maintain a safe distance, the vehicle, just as a human driver would have, has a responsibility to be aware of pedestrians in its proximity and behave appropriately. Detecting the proximity of humans is not only beneficial for autonomous machines, but also for assisting human operators of machines, similar to reversing cameras and existing proximity sensors.

To improve safety around vehicles and machines, we propose a method for detecting and monitoring the distance of humans from a machine within a narrow safety envelope using an RGB-D camera. Our approach uses human instance segmentation and stereo vision to achieve this. The final output of our method is shown in Fig. 1.

Fig. 1. An example output from our proposed method, showing multiple pedestrians identified with individual distance estimates.

This paper is structured as follows: Section II gives a brief overview of the current methods for combined pedestrian detection and distance estimation, and discusses their limitations. Sections III and IV detail our proposed approach, experimental results and the limitations of our method. Finally, Section V concludes the paper and includes a discussion on future work.

II. Related Work

Most of the research surrounding automated or assistive pedestrian safety systems has focused on pedestrian detection, a specialisation of object detection. The task for pedestrian detection is to identifying pedestrians, or more generally people, in images using bounding boxes. For this task, neural networks have been used extensively, thanks to the availability of large amounts of data and annotated images. Popular datasets commonly used for training and testing include the CalTech Pedestrian [3], KITTI [4] and CityPersons [5] datasets. Zhang et al.'s recently published review on pedestrian detection [6] provides a thorough examination of different methods, the current state of the art and how far we are away from achieving human-level performance.

Despite the high level of research activity in pedestrian detection, research beyond pure detection or segmentation

has received less attention. However, the availability and affordability of RGB-D cameras (such as the Microsoft Kinect and Intel RealSense devices) has led some researchers to exploring the use of RGB-D data for detecting and measuring the distances to pedestrians.

Xi, Chen and Aggarwal [7] used the depth map generated by a Microsoft Kinect camera to segment humans from their surroundings. Their approach uses template matching to identify human heads, followed by a region growing algorithm to obtain whole body contours. Though there method achieves a 98.4% detection accuracy in indoor lighting conditions, their approach has several limitations. Firstly, if a persons head is occluded, their method is unable to detect them because of the importance placed on head detection. Secondly, their method was only evaluated in indoor lighting conditions and unlikely to be robust to interference caused by outdoor lighting. If patchy or noisy depth values are present, a head may not be able to be identified, or the region growing algorithm may not be able to extract a whole body contour. The limitations of the Kinect sensor in outdoor lighting conditions are well documented, depth estimation is severely hampered by sunlight because of interfering infrared light [8]. Xi, Chen and Aggarwal [7] do not report on the accuracy of their depth estimates.

Similarly, Spinello and Arras [9] also used the Kinect camera for people detection. For their approach, they combined the Histogram of Gradients (HOG) feature descriptor algorithm with their own variant, the Histogram of Depths (HOD) feature descriptor algorithm, to extract features in the colour and depth images respectively and classify pedestrians using a linear Support Vector Machine (SVM). Once again their method was evaluated only under indoor lighting conditions, and they do not report the accuracy of their depth estimates.

Sharma and Green [10] took a slightly different approach to estimating pedestrian distance, using background subtraction to identify pedestrians. The pixels identified as pedestrians were mapped to the depth map computed by an Intel RealSense R200 camera to estimate distance. However, there method also detected other moving objects in the scene and no distinction between these and pedestrians is made. Their method only achieves 45% accuracy when the camera was mobile and was unable to operate in bright outdoor conditions.

Nimmo and Green [11] also used an Intel RealSense R200 camera. However, instead of background subtraction to detect pedestrians, they used the Single Shot Detector (SSD) network to predict bounding boxes. Depth values for pedestrians are estimated using the depth value at the centre of the bounding box. This poses several problems, if the depth estimates at the centre of the bounding box are inaccurate or missing, then the distance is unable to be estimated. This is particularly likely in outdoor conditions. Their depth estimation was only tested under controlled, indoor conditions at a range of up to 1.3m. Under these conditions, their method proved to be accurate to 0.1 m.

To summarise, there are several limitations in the current research into pedestrian detection and distance estimation, these include the following:

- Poor performance and a lack of evaluation, particularly of the quality of distance estimates, under outdoor lighting conditions.
- A reliance on dense, high quality depth information.

Our method aims to address these issues by using instance segmentation to identify pedestrians and distance estimation techniques more robust to less dense depth information caused by variations in outdoor lighting conditions.

III. Proposed Method

Our proposed method consists of two main stages: segmenting pedestrians using a Mask R-CNN model [12] and estimating the distance to each pedestrian using the depth map generated by an RGB-D camera. In the first stage, colour images are first fed through the Mask R-CNN model to identify the pixels associated with pedestrians in the scene. In the second stage, the pixels in the predicted instance masks are then mapped to a filtered depth map, and the median distance is computed for each pedestrian.

A. Tools and Apparatus

To capture images and depth data, we use an Intel RealSense D435 camera, the specifications for which are listed in Table I. Our software is implemented in Python, using the RealSense SDK 2.0, OpenCV and PyTorch. Although the D435 is capable of capturing colour and depth images at a range of higher resolutions, we use 640 $\times$ 480 pixel images to enable faster processing at inference time. The system our method is trained and evaluated on uses an AMD Ryzen 2500X CPU, 16 GB of RAM and an Nvidia RTX 2070 GPU.

TABLE I
SPECIFICATIONS AND SETTINGS OF THE INTEL REALSENSE D435 CAMERA USED.

Range	0.2m to 10m
RGB FOV	69° (H) × 42° (V)
Depth FOV	74° (H) × 62° (V)
RGB Image Resolution	640 × 480
Depth Image Resolution	640 × 480
Frames per Second	30
Weight	73 g
Dimensions (W × H × D)	90 mm × 25 mm × 25 mm
Data Connection	USB 3.0
Supported Operating System	Linux, Windows

B. Pedestrian Instance Segmentation

To classify and predict bounding boxes and binary masks for pedestrians we use a Mask R-CNN model [12]. Mask R-CNN is a convolutional neural network (CNN) which extends the Faster R-CNN [13] architecture. The network consists of three stages: 1) a backbone CNN, 2) a region proposal network (RPN) and a third stage which classifies objects and predicts bounding boxes and instance masks. The backbone CNN is responsible for extracting features from raw image and generating a feature map. We use a ResNet-50 [14] CNN as the backbone in our model. The second stage RPN

scans the resultant feature map from the backbone network to identify potential regions of interest (ROIs) which may contain objects. This is represented by the first block in the architecture diagram shown in Fig. 3. One difference to Faster R-CNN is that the RoIs pass through an RoIAlign layer as opposed to an RoIPool layer. The RoIAlign layer resizes identified RoIs to a fixed size, but does so using bi-linear interpolation to avoid the discretisation which occurs using RoIPool layers. This helps to create more accurate instance masks, which are more sensitive to misalignment than bounding boxes. The final stage predicts the class labels for for identified RoIs and performs bounding box regression to create bounding boxes, the same as Faster R-CNN. However, in parallel to this, another branch of the model uses a fully convolutional network (FCN) to predict binary masks for each object. This branch is represented by the final two convolution blocks parallel to the classification and bounding-box regression branch shown in Fig. 3.

Fig. 2. The coarse mask annotations included with the COCO dataset [15] (left) compared to the fine mask annotations included with the Supervisely Persons dataset [16] (right).

The Mask R-CNN model we use is pre-trained on the Microsoft Common Objects in Context (COCO) dataset [15]. To specialise our model for pedestrian detection and segmentation, we apply transfer learning and train our model on the Supervisely Persons dataset [16]. This has two main advantages, firstly, it focuses our network on recognising only people, as opposed to balancing performance across the 80 object categories in the COCO dataset. Secondly, the Supervisely dataset includes more fine-grained instance masks than the coarse annotations included with the COCO dataset, allowing for more accurate predictions. The difference in the quality of the annotated masks provided with the COCO and Supervisely Persons datasets is shown in Fig. 2.

The Supervisely dataset contains 5711 images with 6884 fine instance-level annotations. These are split 70-30% for training and testing (3998 and 1713 images respectively). Since the images are of variable sizes and the images processed at inference time are fixed at a size of 640 $\times$ 480 pixels, each image is cropped to the central 640 $\times$ 480 pixel region.

C. Depth Estimation

Depth maps are computed from the left and right stereo infrared images captured by the RealSense D435 using the

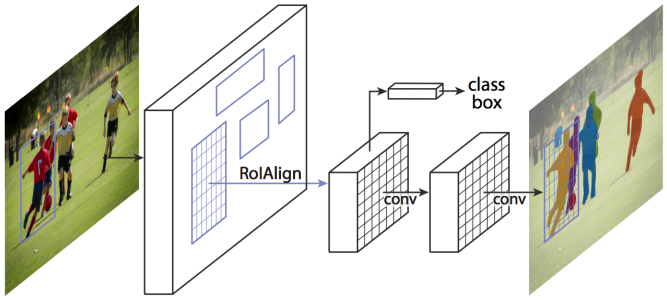

Fig. 3. A high-level architecture diagram of the Mask R-CNN framework [12].

Semiglobal Matching algorithm [17]. To add extra features to the images, an infrared projector shines a dotted grid on the scene.

To estimate the distance to each pedestrian, the depth map is aligned to the RGB image. This allows us to overlay each instance mask over the depth image to extract depth values for each pedestrian. We use the median depth value of the instance mask for our estimate of the distance to each pedestrian. In preliminary testing, we found that using the mean depth value often produced large distance estimates, as it was influenced by depth values far away in the background around the edges of pedestrians which were incorrectly classified as belonging to them.

Because our method does not require dense, high quality depth values across the entire field of view [9] or high quality depth values in very specific locations, such as the centre of each bounding box [11], our method is poised to be more robust to outdoor lighting conditions than previous work.

D. Depth Post-Processing

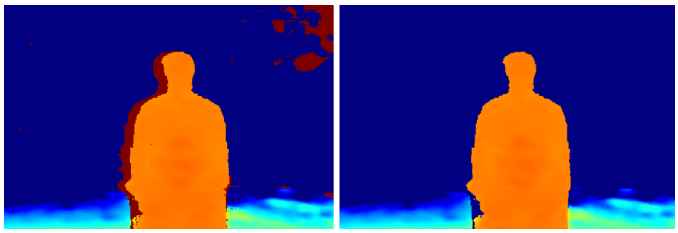

Fig. 4. The raw (left) and filtered (right) depth map.

To improve the accuracy of our system, we perform several post-processing procedures on the raw depth information obtained. The results of which are shown in Fig. 4. To obtain our final depth image, we perform edge-preserving filtering, spatial hole-filling and temporal filtering. Each of these are explained below.

1) Edge-preserving filtering: Edge-preserving filtering [18] attempts to smooth depth noise while preserving edges. This is achieved by scanning each row and column of the depth map bidirectionally and applying a modified exponential moving average (EMA) filter. Using an EMA filter, the smoothed

depth value, S_t, for a pixel t is calculated using the recursive equation defined in Eq. 1.

$$S_t = \begin{cases} Z_1 & t = 1 \\ \alpha Z_t + (1-\alpha) Z_{t-1} & t > 1 \;\&\; |Z_t - Z_{t-1}| < \delta \\ Z_t & t > 1 \;\&\; |Z_t - Z_{t-1}| > \delta \end{cases} \quad (1)$$

In Eq. 1, α determines the level of smoothing and δ sets the threshold for which two neighbouring pixels are defined as edge pixels. The values of α and δ are empirically determined, we use the recommended values 0.6 and 8 respectively [18]. δ is measured in units of 1/32 disparities [18].

2) Spatial hole-filling: Spatial hole-filling [18] is applied to fill holes in the depth map where there is either no data (e.g. occlusion) or low confidence in a depth value. This is achieved by convolving the kernel shown in Table II over the depth map. When a missing value coincides with the centre of the kernel, it is assigned the minimum value of pixels above, below and to the left. The minimum valid depth value is chosen to bias closer depth readings. In the interest of safety, it is better to have a false positive (an object reported closer than it really is) than a false negative (an object reported further away than it really is). The kernel considers only left pixels as the stereo algorithm is left-referenced [18].

TABLE II
THE KERNEL USED FOR SPATIAL HOLE-FILLING.

1	1	0
1	0	0
1	1	0

3) Temporal filtering: Temporal filtering is used to smooth depth values with respect to time. As with the edge-preserving spatial filtering a modified EMA filter is used [18]. For temporal filtering the same formulation as Eq. 1 is used, except t denotes the current frame as opposed to the current pixel. The α and δ values are used are 0.5 and 20 respectively. Again, these values are recommended by [18].

E. Evaluation

We perform a number of evaluations on our method to test performance and robustness to external factors such as lighting and distance. To be deployed in outdoor environments, our method must demonstrate that it can perform well in non-ideal situations. We briefly explain each of our evaluation procedures below.

1) Pedestrian Segmentation: To evaluate the pedestrian segmentation performance of our trained Mask R-CNN model, we evaluate our model on a set of test images from the Supervisely Persons dataset. As we were unable to find any competing results for segmentation on the Supervisely Persons dataset, we also evaluated our model on the set of COCO dataset test images and compared the results to the person segmentation results achieved by the top three models in the COCO 2018 Challenge [19]. To measure the performance of our pedestrian segmentation model on each of the test sets, we use the set of standard COCO evaluation performance metrics [15]. The metrics are the mean average precision (AP) across a variety of intersection over union (IoU) thresholds and the mean average precision for small, medium and large objects (AP_S, AP_M and AP_L respectively).

2) Fill Rate: Since our method is designed to operate outdoors, it is essential that it is robust to different lighting conditions. We preform qualitative and quantitative evaluations to assess performance. We use the set of lighting conditions defined by Vit and Shani [20]. These are given in Table III. Essential to accurate depth estimation is a high fill rate across the region of interest. The fill rate is defined at the percentage of pixels in the region of interest for which there are depth values. For our task, the region of interest is defined as the pixels which belong to the predicted instance masks. The fill rate is calculated before any post-processing. For each of the lighting conditions listed in Table III, a person is placed at 1m intervals within the range of 1m to 5m. The distance to each pedestrian is measured using a tape measure. For each distance and lighting combination, the mean fill rate across 10 consecutive frames is measured.

TABLE III
LIGHTING CONDITIONS TESTED IN OUR SYSTEM EVALUATION.

Lighting Condition	Lux Range
Dawn/Dusk	$<$ 1000 lux
Overcast	1000 - 10,000 lux
Full Daylight	10,000 - 32,000 lux
Direct Sunlight	$>$ 32,000 lux

3) Distance Estimation: The Intel RealSense D435 is factory-calibrated, and even though we do not require sub-centimetre depth accuracy, we still assess the accuracy of the depth readings from start-up in controlled lighting conditions. The interest here is to check that the depth readings are reasonably accurate and to assess how long it takes after a cold start-up for the depth readings to stabilise. Placing the camera 0.60 m from a wall, the furthest away possible with the wall still consuming the entire field of view, we measured the mean depth reading each second for the first 15 minutes of the device warming up. The distance from the wall was measured using a laser measure.

Of more importance is the predicted distance accuracy of pedestrians in real environments, this is evaluated following the same procedure as our fill rate evaluations. For each lighting combination and distance combination, a pedestrian is placed at the measured distance from the camera. The average pedestrian distance is estimated over 10 consecutive frames.

IV. RESULTS

A. Pedestrian Segmentation

The pedestrian segmentation results for our Mask R-CNN model are summarised in Table IV. Our model was trained for 2000 iterations, after which point performance plateaued. On the Supervisely test set, our model performed very well, achieving AP scores consistently higher than those achieved by the top three models submitted for the COCO 2018

challenge. The only exception to this is the mean average precision achieved on small scaled down images (AP_S). Our model achieved only 10% for AP_S, indicating poor performance segmenting small pedestrians. However, since small pedestrians in the image represent those further away, this metric is of the lowest practical significance.

Despite hopes that the fine annotations provided with the Supervisely dataset might lead to better performance on the COCO test set, our model achieves a lower mean average precision than the top 3 models from the COCO 2018 challenge. One possible reason for this is because our model was explicitly trained on images of 640×480 pixels (as this is the resolution of images captured by the D435) and therefore might suffer on images of larger and varying sizes, like those included in the COCO set.

It is worth mentioning that comparisons across datasets between our model and the other models are not completely fair, since 1) the respective models are trained on different datasets and 2) the quality and difficultly of the annotations may differ across datasets. In future work, a comparison of the models with the training and test sets held constant will be evaluated, and tested to see whether the results translate to real application performance improvements.

B. Fill Rate

With respect to fill rate, the D435 camera was able to collect depth values for upwards of 90% of the region of interest across all lighting conditions. Only at distances of 1 m and 2 m in direct sunlight did the average fill rate fall below 90%, averaging 84.6% and 88.9% respectively. The reason for this is likely because in these conditions, the pedestrian covers more of the field of view, resulting in more pixels for which depth values are required. The average fill rates for each distance and lighting condition are shown in Table V.

C. Distance Estimates

For our start-up depth accuracy test, the mean estimated depth for the duration of the 15 minutes from start-up was near-constant, between 0.59m and 0.58m. This was less than 2cm from the measured depth, and well within the accuracy requirements for estimating pedestrian proximity. The near-constant readings also demonstrate that there is no required warm-up period for our approach.

The results for our outdoor distance evaluations are displayed in Fig. 5. Across all lighting conditions and all distances, our method estimated pedestrian distances with 87.7% accuracy. Under dusk, overcast, full daylight and direct sunlight conditions, our method estimated pedestrian distances with 92.1%, 91.3%, 86.9% and 80.3% accuracy respectively.

Fig. 5 also shows that in general, the detector has a tendency to overestimate pedestrian distance, which is something that will be investigated in future work. An unsurprising trend is that the distance measurements degrade with distance and brightness. The degradation with distance can be explained by the reduced disparity between the left and right images at greater distances from the camera. For lighting, the brighter

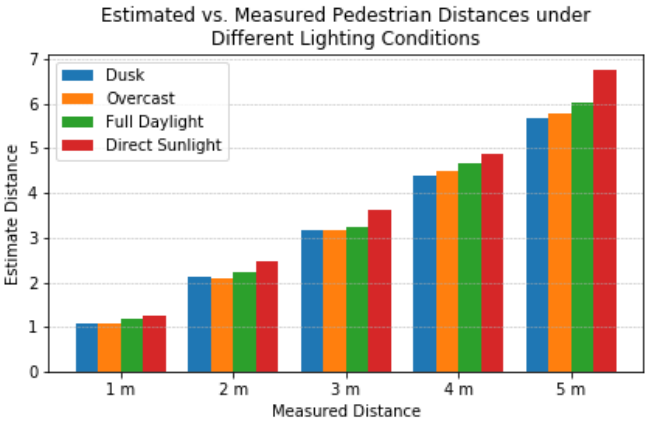

Fig. 5. Mean estimated pedestrian distances under different lighting conditions compared to the measured distances.

the sunlight the more infrared interference is introduced by the sun.

Crucially, our system is able to estimate distances to pedestrians within reasonable bounds across lighting conditions at close distances (within a few metres). This is something which has not been achieved in prior work [7], [9], [10], [11]. Nimmo and Green [11] were only able to achieve approximately 93% accuracy at a distance of 1.3m in indoor lighting conditions.

D. Limitations

As mentioned in the above sections, our system does posses some limitations. Most notably, our detector has a tendency to overestimate distances to pedestrians and performance degrades with both lighting and distance. There is also room for improvement in pedestrian segmentation, which itself impacts distance estimates. Although our segmentation model achieves a 94.6% AP with a 0.5 IoU threshold on the Supervisely Persons dataset, AP_S (mean average precision on small objects is only 10%.

Another limitation of our system is that it only runs at an average of 15 fps. Other methods discussed [10], [11], [9] achieve speeds of 30 fps. Improvements in speed may be able to be achieved with more efficient algorithms or software implementations, such as using C++ or implementing our model in TensorFlow.

V. Conclusion

In this paper, we presented a novel method for detecting pedestrians and estimating their distances using RGB-D data, based on Mask R-CNN and depth information captured using an infrared stereo RealSense D435 camera. Unlike previous methods, tested only in controlled indoor environments, our approach performs well across the full range outdoor lighting conditions and distances, achieving an an average distance estimate accuracy of 87.7%. For pedestrian segmentation, our model achieves an AP_{50} score of 94.6% on the Supervisely Persons dataset. Our method outperforms existing pedestrian detection and distance estimation techniques [10], [11], [9],

TABLE IV
COCO METRIC SCORES OF THE TOP THREE COCO CHALLENGE 2018 MODELS [19] AND OUR MODEL FOR PERSON SEGMENTATION.

	Backbone	Dataset	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MMDet	FishNet	COCO	**0.546**	0.856	0.611	**0.394**	**0.586**	0.727
MegDet	ShuffleNet v2	COCO	0.544	**0.867**	**0.621**	0.37	0.582	0.723
FirstShot	Mask R-CNN	COCO	0.528	0.822	0.584	0.316	0.571	**0.746**
Our Model	Mask R-CNN	COCO	0.362	0.639	0.372	0.17	0.434	0.568
Our Model	Mask R-CNN	Supervisely	0.679	0.946	0.768	0.1	0.609	0.69

TABLE V
MEAN FILL RATES (AND STANDARD DEVIATION) FOR DIFFERENT LIGHTING CONDITIONS AND DISTANCES.

	Dusk (556 lux)	Overcast (7,380 lux)	Full Daylight (22,279 lux)	Direct Sunlight (102,837 lux)
1 m	90.7 % (0.3)	92.1 % (0.6)	91.3 % (0.4)	84.6 % (1.4)
2 m	92.3 % (0.2)	93.5 % (0.3)	92.7 % (0.3)	88.9 % (1.3)
3 m	94.1 % (0.2)	96.3 % (0.3)	96.3 % (0.2)	91.1 % (1.0)
4 m	97.3 % (0.2)	98.5 % (0.4)	99.0 % (0.1)	96.6 % (0.9)
5 m	98.5 % (0.2)	98.9 % (0.4)	99.1 % (0.1)	97.9 % (0.5)

[7], none of which have been proven to perform in outdoor lighting conditions. Our method show promise for use in automated or assistive driving technologies, or for dynamic safety envelopes around industrial, agricultural or construction equipment.

A. Future Work

Our method for pedestrian detection and distance estimation is a promising approach which has demonstrated encouraging preliminary results. However, as is evident by the limitations of our system, there is still room for improvement. Some avenues for future work are listed below.

1) Model improvements: Our model was only trained using 3998 images, a comparatively small amount compared to the millions of examples commonly used to train top performing neural networks. In future work, we will investigate performance improvements which can be gained by training our Mask R-CNN model on more examples and performance enhancements which might be gained using different models.

2) Harnessing depth data for segmentation: In the current form, our method segments pedestrians and predicts instance masks using the RGB data alone. In future work, investigation into also using the depth map for refining the predicted instance masks will also be investigated. [9] used both RGB and depth data for their segmentation with positive results.

3) Estimating depth: As mentioned in our results, our method consistently overestimates distances to pedestrians. Further investigation into other methods for distance estimates, other than the median mask value may yield less biased results.

REFERENCES

[1] Ministry of Transport, "Pedestrian Crashes," 2018. [Online]. Available: https://www.transport.govt.nz/mot-resources/new-road-safety-resources/pedestrians/

[2] WorkSafe New Zealand, "WorkSafe Fatalities Detail," 2019. [Online]. Available: https://worksafe.govt.nz/data-and-research/ws-data/fatalities/

[3] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2009, pp. 304–311.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, Aug. 2013. [Online]. Available: https://doi.org/10.1177/0278364913491297

[5] S. Zhang, R. Benenson, and B. Schiele, "CityPersons: A Diverse Dataset for Pedestrian Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 4457–4465.

[6] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards Reaching Human Performance in Pedestrian Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 973–986, Apr. 2018.

[7] L. Xia, C. Chen, and J. K. Aggarwal, "Human detection using depth information by Kinect," in *CVPR 2011 WORKSHOPS*, Jun. 2011, pp. 15–22.

[8] S. Zennaro, M. Munaro, S. Milani, P. Zanuttigh, A. Bernardi, S. Ghidoni, and E. Menegatti, "Performance evaluation of the 1st and 2nd generation Kinect for multimedia applications," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2015, pp. 1–6.

[9] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 3838–3843.

[10] U. Sharma and R. Green, "Anti-Collision System for Pedestrian Safety," Computer Vision Lab, University of Canterbury, Tech. Rep., 2017.

[11] J. Nimmo and R. Green, "Pedestrian Avoidance in Construction Sites," Computer Vision Lab, University of Canterbury, Tech. Rep., 2017.

[12] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.90

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, pp. 740–755.

[16] Supervisely, "Supervisely - Web platform for computer vision. Annotation, training and deploy," 2018. [Online]. Available: https://supervise.ly/

[17] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[18] Intel Corporation, "Depth Post-Processing for Intel RealSense D400 Depth Cameras," 2019. [Online]. Available: https://dev.intelrealsense.com/docs/depth-post-processing

[19] C. Consortium, "COCO (Common Objects in Context) Detection Leaderboard," 2018. [Online]. Available: http://cocodataset.org/detection-leaderboard

[20] A. Vit and G. Shani, "Comparing RGB-D Sensors for Close Range Outdoor Agricultural Phenotyping," *Sensors (Basel, Switzerland)*, vol. 18, no. 12, p. 4413, Dec. 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/30551636