Training a convolutional neural network for transportation sign detection using synthetic dataset

Huy Le*, Minh Nguyen†, Wei Qi Yan‡, and Saide Lo§
School of Engineering, Computer & Mathematical Sciences,
Auckland University of Technology
Email: *robert01011991@gmail.com, †minh.nguyen@aut.ac.nz, ‡wyan@aut.ac.nz, §saide.lo@aut.ac.nz

***Abstract*—A sufficient amount of training data must be prepared when training a neural network for transportation signs, which requires extensive time and labour work to complete as it is usually done manually. This paper presents a real-time object detection neural network to predict the transportation signs in different environmental conditions and be sufficiently trained by a synthetic generated dataset. Our proposed method is based on the concept of domain randomisation, where a massive amount of images is generated by randomly orienting the transportation sign objects to a computer graphics rendered virtual scene. The trained model could achieve over 80% precision and over 90% recall and mAP against the real-world test dataset. Our proposed method showed that the generated synthetic dataset is adequate for predicting various transportation signs. Preparing and utilising the synthetic data can be an efficient solution to minimise the labour cost and human error during the manual data annotation step.**

***Index Terms*—Artificial intelligence, YOLO, Synthetic dataset generation**

I. Introduction

Deep learning [1], [2] has captured vast attraction across both research and industrial domains. In many computer vision-based applications, the deep learning techniques using convolutional neural network (CNN) [3] give significant performance results in various tasks, such as image classification and object detection. These beneficial techniques have also been applied to the field of transportation. These tasks involve vehicles detection [4], traffic-sign detection (TSD) [5], [6] or real-time traffic incident classification [7]. Those studies have also shown that they can help to reduce the labour cost and response time during emergencies. In recent years, there is many datasets of the traffic signs have been collected in different countries, such as the Chinese Traffic Sign Database (TSRD) [8], German Traffic Sign Dataset (GTSRB) or Tsinghua-Tencent 100K (TT100K) [9]. These datasets are used for the TSD task and contain many transportation signs images in different countries and natural environment conditions. However, the major drawback of using these datasets for deep neural network training is preparing many labelled data. The GTSRB dataset consists of more than 40 classes and over 50000 classified images in the train/test dataset. Meanwhile, the TT100K dataset contributes 100000 images, but only 30000 of those contain traffic-sign instances. The dataset samples often require extensive data annotation as they are done manually and are generally difficult to prepare for the unique research purpose. The sim2real transfer is the most popular method that uses synthetic images to train the deep neural network and minimise manually annotated instances. The significant advantage of sim2real is that we can easily control the ground-truth annotation without human labour. The successfully uses of sim2real can be found in the 3D human pose estimation [10] and the industrial robotics control system [11]. One disadvantage of using the sim2real method is the non-realistic appearance issue between the synthetic images and the natural scene. Many researchers have used generative adversarial networks (GAN) to generate realistic images from synthetic datasets to overcome this drawback [12], [13]. However, we still need to prepare an initial real dataset to train the GAN model and generate the larger synthetic dataset.

The graphics processing unit (GPU) significantly improves both gaming and deep neural network training purposes. The latest Nvidia GPUs with the built-in compute unified device architecture (CUDA) allow direct access to the GPU's instruction set and parallel computational elements for kernel-related tasks execution [14]. It means that more large blocks of data will be processed in parallel at the same time to reduce the deep neural network training time [15]. The CUDA units are also responsible for real-time high-resolution graphics rendering tasks. Nowadays, the GPU can be easily enabled with few popular game engines, such as Unity [16], to render realistic images more petite than a second. The game engine's advantage is domain randomisation, which allows us to generate nearly unlimited synthetic images with random parameters. An early successful example of using domain randomisation was using various camera positions, object locations and lighting conditions to generate synthetic images [17]. In reality, the transportation signs generally appear with similar shapes and can only be distinguished by comparing their textures. However, unexpected natural illumination or camera angles can easily shift the original texture information and reduce model prediction performance. Hence, the data annotation process is the most critical step of training the deep neural network model for transportation signs prediction where the most accurate data labelling is required. In such a situation, using the game engine's power and automatic digital graphic

978-1-6654-0645-1/21/$31.00 ©2021 IEEE

rendering techniques can be a suitable solution to overcome the manual data annotation limitations and improve the deep-learning-based prediction accuracy.

This paper proposes a novel method using the auto-generated synthetic dataset to train an object detection deep neural network. Moreover, our proposed method allows us to generate nearly unlimited images representing the most natural variation in different perspectives with minimal labour cost. It is also helpful for preparing the deep-learning-based application prototype when time constraints are the priority. Our proposed method uses the domain randomisation approaches, where we generate the training images by randomly locating the synthetic transportation signs with the natural environment backgrounds and applying different augmentation filters. In short, this paper produces two main contributions. Firstly, this is the attempt of using the game engine and GPU to generate the synthetic dataset for transportation signs detection that can minimise the manual data annotation labour intensive and improve dataset quality (Fig. 1). Second, we proposed a method that can be used to predict the actual images and quickly apply them to the prototype or production process.

II. Methods

Transportation sign materials. We used the New Zealand standard road signs in this paper that are categorised into three different types: (1) compulsory, (2) warning, (3) information. All the details of the transportation signs can be obtained at the NZ Transport Agency official website (https://www.nzta.govt.nz/roadcode/). In general, it has hundreds of different transportation signs in New Zealand; but we only used the top 50 common road signs for the deep neural network training: speed limit (R1-1, R1-1.1, R1-1.2); Speed Limit Derestriction (R1-3); Temporary (R1-8); No Stopping (R6-10.1); Disabled Parking (R6-55); Bus Parking (R6-53); Motorcycle Parking (R6-51); No Parking: Bus Stop (R6-71); No Parking (R6-70); No Parking: Taxi Stand (R6-72); Attention (TW-2); Road Works (T1A); Stop (R2-1); Turn Left or Right (R3-11); Turn Left (R3-8); Turn Right (R3-10); Go Straight (R3-9); Priority Over Oncoming Vehicles (R2-8); Wrong Way (R3-7); No Turn Left (R3-1); No Turn Right (R3-2); No U-Turn (R3-3); Road Closed (R3-6); One-way traffic (R3-12); Traffic lights ahead (W10-4); Must Turn Left/Right (R4-1); May Proceed Straight or Turn Left/Right (R4-3); Must Proceed Straight (R4-2); Give Way (R2-2); Give Way at Roundabout (R2-3); Give Way to Oncoming Vehicles (R2-7); Except Bus (R3-5.1); Bus Lane (R4-7); Buses Only (R4-7.1); No Entry (R3-4); School; No Exit (A40-1).

Synthetic dataset generation as shown in Fig. 2. The transportation image pool contains the transportation sign images, saved as individual image files. Each of those sign images will be cropped to the fixed size of 1280 $\times$ 1280. For the background images, over 1400 outdoor landscape images, including day and night time, are used as the "background images pool", and they are cropped to the sized of 1500 $\times$ 2250. The process of synthetic data generation is described as follows. Firstly, one or multiple images will be randomly selected from the "transportation images pool" and carefully pasted to the transparent canvas of size 1500 $\times$ 2250. The probability of displaying single or multiple signs on the same background is set to 50. Secondly, another image of the background is selected from the "background images pool". The selected transportation sign images then will be augmented by applying random filters such as rotation, the x and y coordinates, scaling, blur, noise. However, the coordinate values are restricted to a specific range so that the signs do not exceed the background size. They are dependent on the size of the sign image after scaling and its rotation angle. Lastly, the transportation signs are pasted to the transparent canvas according to the described parameters, and the canvas will be pasted on the top of the selected background. During the synthetic dataset generation, the bounding box with the same size as the sign image is created along with its label name. The size and the coordinates of the bounding boxes are normalised between 0 and 1 by diving by the width and height of the background. This procedure can generate an image size of 1500 x 2250 with the transportation signs randomly oriented inside a random outdoor background. It takes 6 hours to produce over 100000 trainable images and their corresponding labels files. While in the real world, it would take weeks or months to complete a similar process.

Real-world and synthetic datasets for model evaluation. In order to compare the deep neural network training performance, we prepare a real-world dataset consisting of 350 images of transportation signs taken using different photography equipment such as the digital camera or build-in mobile camera. Each image has a size of 1500 x 2250 and contains single or multiple images of the transportation signs. We use the same values to determine the bounding box boundaries described in the previous step. The labelling process is done manually with the online open-source RoboFlow (https://blog.roboflow.com/labelme/). This manual process takes more 2 weeks to complete with the help of over 30 volunteers. We also prepare another synthetic dataset generated using the proposed method, consisting of the same number of images as the real dataset.

Deep neural network model. We use the "You only look once" – version 3 (YOLOv3) [18] model to train the datasets and the build-in ImageNet [19] pre-trained weights for the convolutional layers. The state of the art of YOLOv3 is classifying the entire input image and directly outputting the bounding box coordinates and class names. YOLOv3 is claimed to significantly improve over its predecessors by using the residual blocks, skip connections and upsampling (Fig. 3). These changes aim to resolve the small object detection issues in YOLOv2 and increase the accuracy rate. The YOLOv3 model can detect objects in three different scale levels at layer 82^{nd}, 94^{th}, and 106^{th}. This type of network design is inspired by the Feature-Pyramid Network (FPN) developed in

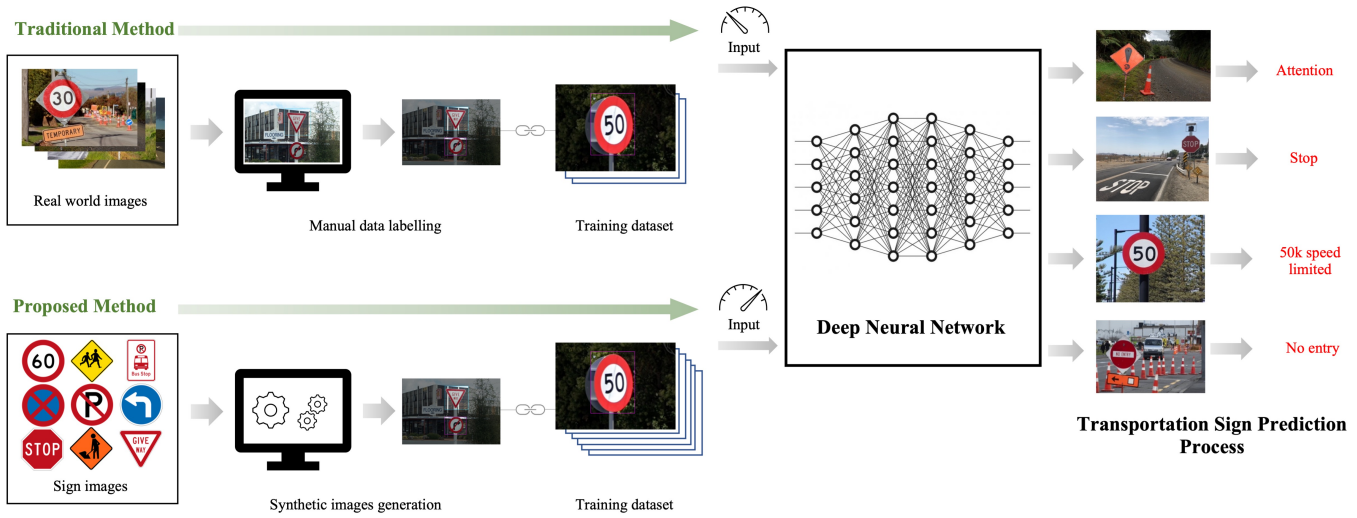

Fig. 1. **Overview of the proposed training process of transportation signs detection.** The traditional method is done with manual data labelling that requires more time and labour cost to complete, while our proposed method can reduce these drawbacks by using the auto rendered computer graphics to generate the synthetic dataset for the deep neural network model.

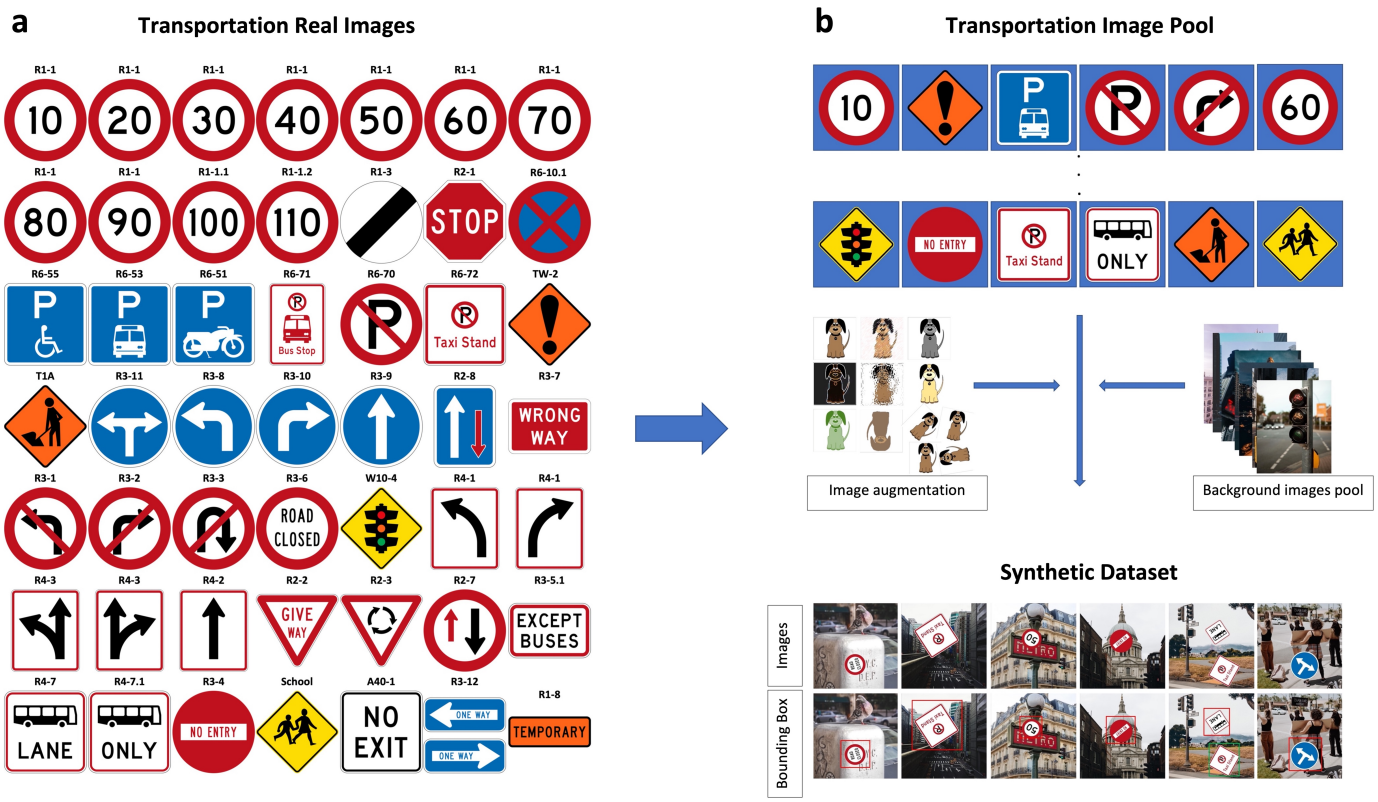

Fig. 2. **Synthetic dataset preparation process. a** The real-world images of different New Zealand transportation signs with their code are described below each image. These signs were also used for the real-world dataset. **b** Synthetic dataset generation where the images are generated by combining the transportation signs with random background images together with various augmentation filters. The details of generated ground-truth label (bounding box) are shown at the bottom.

2017 [20]. FPN allows the model to learn objects of different sizes. The smaller detection blocks detect the objects with lower resolution or large objects, and the greater detection blocks are used for smaller objects. In the training process of this paper, we keep the same training parameters as described in the YOLO original paper. The network is trained for 80 iterations, with 8 is the batch size, and four is used for sub-division.

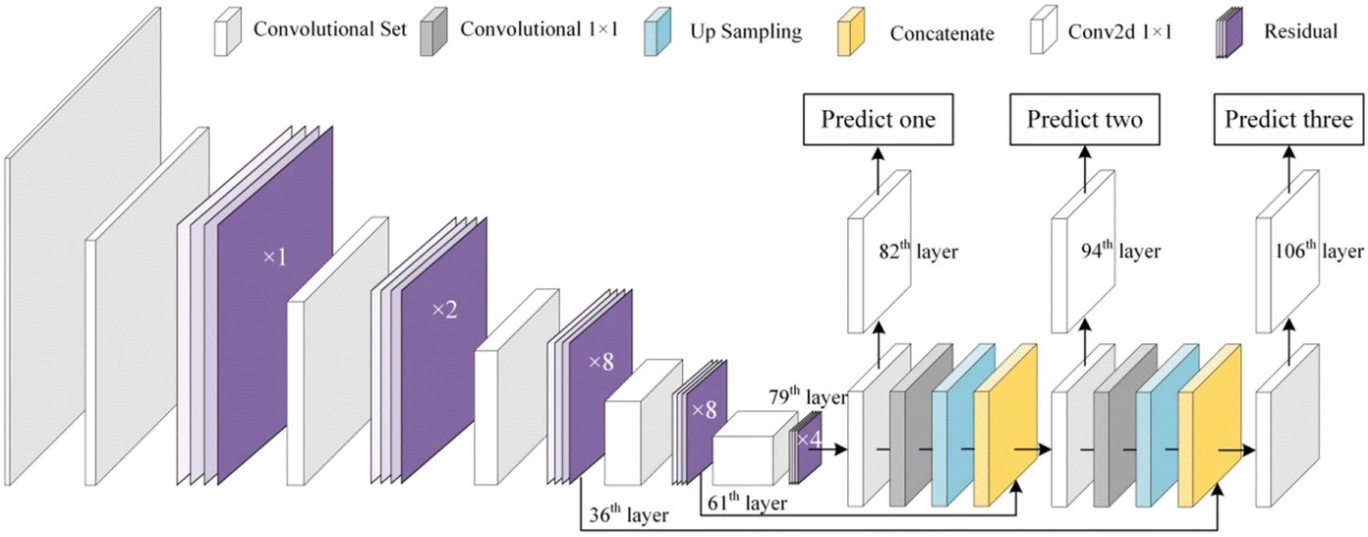

Fig. 3. YOLOv3 neural network architecture.

Model evaluation metrics. To measure the accuracy of the object detection model, we use the two common metrics which are used in MS COCO [21] and Pascal VOC [22] dataset.

The **recall** metric evaluates how well the objects have been detected (Equation 1) and can be calculated by obtaining the intersection-over-union (IoU) value. The IoU determines the overlap between the predicted bounding and the ground truth over their area union. If the IoU value is greater than a threshold value (generally set to 0.5), it is considered as true-positive (TP), otherwise false-positive (FP). The false-negative (FN) is made when no prediction is made for a particular ground truth bounding box.

$$recall = \frac{TP}{TP + FN} \tag{1}$$

The mean average precision (mAP) metric calculates the mean value of average precision over the IoU thresholds, where the precision can be obtained using Equation 2. In precision equation, the value of false-negative is replaced by false-positive (FP). The Pascal VOC used a static threshold of 0.5 for the mAP, commonly called mAP0.5. Another mAP metric is mAP[0.5:0.95], where the threshold is from 0.5 to 0.95.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

Software and hardware requirements. We use Python 3.8 and PyTorch (ver.1.7.1) for the deep neural network training back-end. Single GPU (GeForce RTX 2080 Super) is used for the model training, which has 384 tensor cores that could produce up to 11.15 TFLOPS for FP32. Each epoch takes approximately 32 minutes to complete, or it requires an average of 19 milliseconds to process a single image.

III. Results Evaluation

Model evaluation. As described in the previous Section, we trained four different datasets with the same models. We used 3 model versions of YOLOv3 (tiny, 416, SPP) for this experiment. The training parameters were kept the same in order to minimise the potential training bias. We generated two synthetic datasets using the proposed method; one has 50 different class names (100K), and the other consists of 35 class names (350). Table 1 summarises the quantitative evaluation using values of recall, precision, and mean average precision (mAP). Our proposed method achieved over 98% for the 100K dataset and over 70% for the 350 dataset. Significantly, the synthetic dataset could produce an average of 42% higher than the real-world dataset. It means that the synthetic dataset can predict the bounding box position with a very low false-negative rate. There is the same result with the false-positive value found in the precision records where our synthetic dataset could achieve over 87% and an average of 18.7% higher than the real-world dataset. The mAP values, which were calculated based on the IoU thresholds, comparable mAP0.5 values scored over 90% for the synthetic dataset. The higher IoU threshold (mAP[0.5:0.95]) scored over 57% for the synthetic dataset. The experiment results suggest that the model's ability to predict the transportation sign and its bounding box is better in the case of synthetic than the real-world images.

Predictions under different conditions. We further experimented with examining how well the model predicted under different lighting and weather conditions (Fig. 4). In order to complete this experiment, we collected the real-life images taken from either personal cameras or by vehicles driving video streams. The images and video streams are conducted under different conditions: morning, night-time, raining or foggy. The results indicated that the models could predict most

transportation signs with an average of over 50% accuracy rate. Significantly, the rates could reach 70% under poor lighting conditions.

IV. Conclusion

In this paper, we showed that utilising the synthetic dataset can successfully train the object detection deep neural network model to classify the real-world images of New Zealand transportation signs. Our proposed method uses the power of the graphic processing unit (GPU) to generate a nearly unlimited image within a short period. It is labour cost-efficient and practical compared with the manually labelling techniques. The experimental results indicated that our synthetic dataset could score similar prediction accuracy rates as the real-world dataset. Significantly, our dataset could achieve promising performance under poor conditions, such as low-lighting or foggy weather. Our method is strongly recommended for low-cost applications or prototypes for transportation or deep neural network-related projects. However, we are only implementing and test our solution on a desktop environment for this stage. In practice, the commercial application should be able to run on different platforms with various hardware restrictions. Furthermore, we aim to expand the project to mobile platforms such as iOS and Android, where we can generate the synthetic dataset and train the models using on-mobile GPU.

References

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

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

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[4] S. Roy and M. S. Rahman, "Emergency vehicle detection on heavy traffic road from cctv footage using deep convolutional neural network," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019, pp. 1–6.

[5] C. Dewi, R.-C. Chen, Y.-T. Liu, Y.-S. Liu, and L.-Q. Jiang, "Taiwan stop sign recognition with customize anchor," in *Proceedings of the 12th International Conference on Computer Modeling and Simulation*, 2020, pp. 51–55.

[6] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *Journal of Big Data*, vol. 7, no. 1, pp. 1–26, 2020.

[7] H. Nguyen, C. Cai, and F. Chen, "Automatic classification of traffic incident's severity using machine learning approaches," *IET Intelligent Transport Systems*, vol. 11, no. 10, pp. 615–623, 2017.

[8] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2110–2118.

[9] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[10] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3d human pose estimation: motion to the rescue," *arXiv preprint arXiv:1907.02499*, 2019.

[11] M. Kaspar, J. D. M. Osorio, and J. Bock, "Sim2real transfer for reinforcement learning without dynamics randomization," *arXiv preprint arXiv:2002.11635*, 2020.

[12] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2107–2116.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[14] F. Abi-Chahla, "Nvidia's cuda: The end of the cpu?" *Toms Hardware*, pp. 1954–7, 2008.

[15] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," in *2019 IEEE high performance extreme computing conference (HPEC)*. IEEE, 2019, pp. 1–9.

[16] J. Haas, "A history of the unity game engine," *Diss. WORCESTER POLYTECHNIC INSTITUTE*, 2014.

[17] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[22] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

TABLE I

FULL MODEL EVALUATION. THE TABLE DESCRIBES THE EVALUATION RESULTS OF OUR SYNTHETIC DATASETS (**PRESENTED IN BOLD**) AND THE OTHER TWO DATASETS CONDUCTED USING THE MANUAL DATA ANNOTATION PROCESS. THE RECALL AND PRECISION VALUES ARE SET AT THE IOU THRESHOLD OF 0.5. THE MEAN AVERAGE PRECISION (**MAP**) VALUES ARE SET AT THE IOU THRESHOLD OF 0.5 (**MAP0.5**) AND FROM 0.5 TO 0.95 (**MAP[0.5:0.95]**).

Model	Dataset	Number of class	Recall	Precision	mAP0.5	mAP[0.5:0.95]
YOLOv3 tiny	NZ Signs 350 (Real)	35	0.476	0.298	0.482	0.304
	NZ Signs 350 (Synthetic)	**35**	**0.733**	**0.386**	**0.579**	**0.323**
	NZ Signs 100K (Synthetic)	**50**	**0.973**	**0.876**	**0.903**	**0.559**
YOLOv3 416	NZ Signs 350 (Real)	35	0.6	0.356	0.538	0.342
	NZ Signs 350 (Synthetic)	**35**	**0.843**	**0.359**	**0.735**	**0.432**
	NZ Signs 100K (Synthetic)	**50**	**0.983**	**0.873**	**0.907**	**0.576**
YOLOv3 SPP	NZ Signs 350 (Real)	35	0.528	0.318	0.496	0.295
	NZ Signs 350 (Synthetic)	**35**	**0.693**	**0.403**	**0.595**	**0.396**
	NZ Signs 100K (Synthetic)	**50**	**0.983**	**0.872**	**0.908**	**0.576**

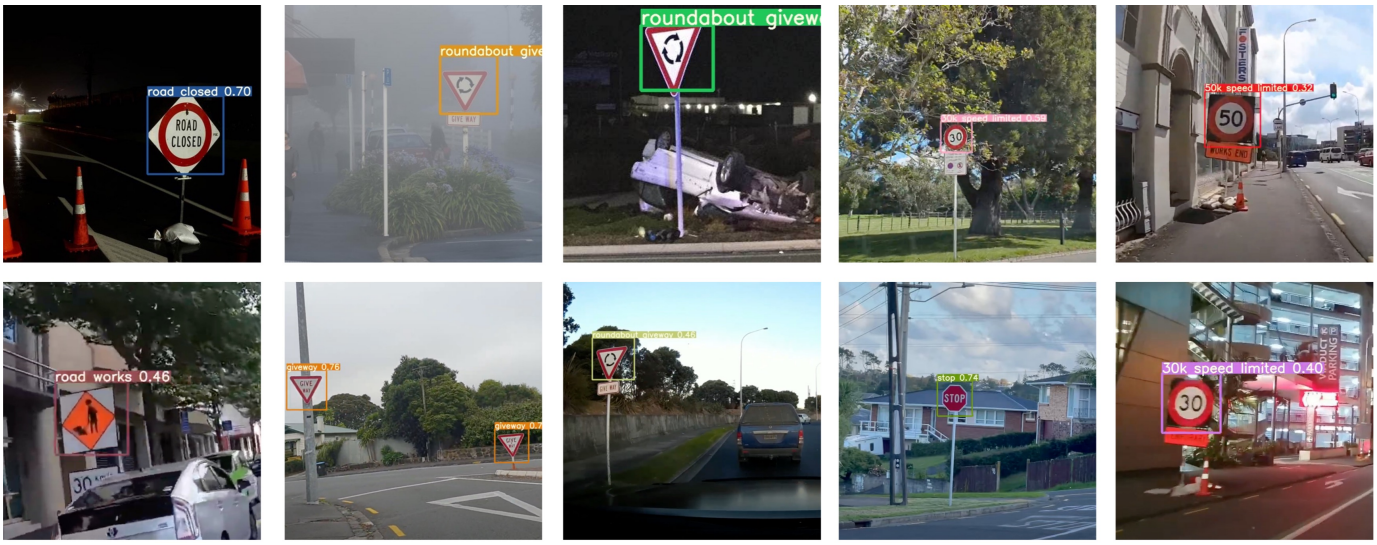

Fig. 4. YOLOv3 model predictions using our synthetic dataset under various light and weather conditions.